



ESCUELA UNIVERSITARIA DE POSGRADO

**DISEÑO DE UN EXOESQUELETO ROBOTICO DE CODO BASADO EN
ACTUADORES DE ALAMBRE DE ALEACION CON MEMORIA DE FORMA**

Línea de investigación:

Sistemas inteligentes, robótica, domótica

Tesis para optar el grado académico de Doctor en Ingeniería de Sistemas

Autor

Rojas Serna Carlos Alfonso

Asesor

Cuba Aguilar Cesar Raúl

ORCID: 0000-0002-0976-4328

Jurado

Cachay Boza, Orestes

Rodriguez Rodriguez, Ciro

Flores Masias, Edward Jose

Lima - Perú

2025

DISEÑO DE UN EXOESQUELETO ROBOTICO DE CODO BASADO EN ACTUADORES DE ALAMBRE DE ALEACION CON MEMORIA DE FORMA

INFORME DE ORIGINALIDAD

16%

INDICE DE SIMILITUD

13%

FUENTES DE INTERNET

3%

PUBLICACIONES

5%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

repositorio.cinvestav.mx

Fuente de Internet

3%

2

www.virtualpro.co

Fuente de Internet

1%

3

hdl.handle.net

Fuente de Internet

<1%

4

Submitted to Unviersidad de Granada

Trabajo del estudiante

<1%

5

Submitted to Infile

Trabajo del estudiante

<1%

6

Submitted to Submitted on 1685667234185

Trabajo del estudiante

<1%

7

repositorio.unfv.edu.pe

Fuente de Internet

<1%

8

www.researchgate.net

Fuente de Internet

<1%



ESCUELA UNIVERSITARIA DE POSGRADO

**DISEÑO DE UN EXOESQUELETO ROBOTICO DE CODO
BASADO EN ACTUADORES DE ALAMBRE DE ALEACION
CON MEMORIA DE FORMA**

Línea de investigación:

Sistemas inteligentes, robótica, domótica

Tesis para optar el grado académico de Doctor en Ingeniería de Sistemas

Autor

Rojas Serna Carlos Alfonso

Asesor

Cuba Aguilar Cesar Raúl

ORCID: 0000-0002-0976-4328

Jurado

Cachay Boza, Orestes

Rodriguez Rodriguez, Ciro

Flores Masias, Edward Jose

Lima – Perú

2025

Dedicatoria

Quiero expresar mi agradecimiento a mis padres, ya que han sido el fundamento clave en mi formación, tanto en el ámbito académico como en el de la vida. Su apoyo incondicional a lo largo del tiempo ha sido fundamental para todo lo que soy. Este logro ha sido posible gracias a ellos.

Agradecimiento

Mi más sincero agradecimiento a mi familia, que siempre ha estado a mi lado y me ha ayudado en todas las etapas de mi trabajo. Mi asesor, Cuba Aguilar César Raúl, ha compartido generosamente sus conocimientos y me ha guiado con su inestimable experiencia, por lo que le estoy enormemente agradecido. Mi agradecimiento también a todas las personas que me han ayudado a lo largo del camino, ya sea directa o indirectamente.

ÍNDICE

Resumen	xii
Abstract.....	xiii
Prólogo.....	xiv
I INTRODUCCIÓN.....	1
1.1 Planteamiento del problema	2
1.2 Descripción del problema.....	3
1.3 Formulación del problema.....	3
1.3.1 Problema general.....	3
1.3.2 Problemas específicos	4
1.4 Antecedentes	5
1.5 Justificación de la investigación	7
1.6 Limitaciones de la investigación	8
1.7 Objetivos	9
1.7.1 Objetivo general	9
1.7.2 Objetivo específicos.....	9
1.8 Hipótesis	10
1.8.1 Hipótesis general.....	10
1.8.2 Hipótesis específicas	10
II. MARCO TEÓRICO..	11
2.1 Marco conceptual	11
2.1.1 Estructuras anatómicas del codo	11
2.1.2 Biomecánica del codo	13
2.1.3 Lesiones y enfermedades del codo.....	16
2.2 Exoesqueleto robótico	21

2.2.1 Partes y componentes	22
2.2.2 Tipos de exoesqueletos robóticos.....	23
2.2.3 Aplicaciones en diversos sectores	26
2.3 Aleación con memoria de forma.....	29
2.3.1 Propiedades de las aleaciones con memoria de forma	32
2.3.2 Aplicaciones	37
2.4 Ingeniería mecánica y automotriz.....	38
2.4.1 Arquitectura de redes neuronales	42
2.4.2 Entrenamiento y optimización.....	45
2.4.3 Aplicaciones prácticas.....	46
2.5 Algoritmos genéticos.....	47
2.5.1 Funcionamiento del algoritmo genético	50
2.5.2 Aplicaciones en Ingeniería y Ciencias	53
III. MÉTODO	57
3.1 Tipo de investigación.....	57
3.2 Población y muestra	57
3.2.1 Población.....	57
3.2.2 Muestra.....	57
3.3 Operacionalización de variables	58
3.4 Instrumentos.....	59
3.5 Procedimientos.....	59
3.6 Análisis de datos.....	60
3.7 Consideraciones éticas.....	60
IV. RESULTADOS	61
4.1 Control automático	61

4.1.1 Control PD: Adaptación de gravedad.....	61
4.1.2 Control por modos deslizantes adaptable.....	64
4.1.3 Evaluación numérica de controladores para codo.....	66
4.1.4 Diseño Conceptual del exoesqueleto.....	70
4.1.5 Ecuaciones de distancia brazo/antebrazo	71
4.1.6 Sistema general de control del exoesqueleto.....	73
4.1.7 Selección y diseño del actuador	74
4.1.8 Modelado del comportamiento del actuador Nitinol	74
4.1.9 Diseño Mecánico del Exoesqueleto	76
4.2 Implementación de Actuadores de AMF en el Exoesqueleto	79
4.3 Diseño electrónico de control del exoesqueleto	82
4.3.1 Diagrama del sistema de control.....	82
4.4 Esquema del controlador de corriente	83
4.5 de la placa de circuito impreso	84
4.6 Modelado dinámico	86
4.6.1 Estructuras Anatómicas del Codo	86
4.6.2 Filtrado y análisis dinámico de datos	87
4.7 Modelos ARMA	88
4.7.1 Ajuste y predicción de modelos ARIMA.....	88
4.7.2 Comparación y selección del mejor modelo	89
4.8 Clasificador por redes neuronales	92
4.8.1 Recopilación de datos y sesiones de rehabilitación.....	92
4.8.2 Configuraciones y resultados de la entrada de red neuronal	95
4.8.3 Configuraciones y resultados de modelos neuronales.....	99
4.8.4 Evaluación y resultados del modelo mejorado.....	106
V. DISCUSIÓN DE RESULTADOS	110

V. CONCLUSIONES.....	112
VII. RECOMENDACIONES	114
VIII. REFERENCIAS	116

ÍNDICE DE FIGURAS

Figura 1 Anatomía del codo.....	12
Figura 2 Flexión y extensión del codo.....	13
Figura 3 Pronación y supinación.....	14
Figura 4 Epicondilitis.....	17
Figura 5 Fractura del codo	18
Figura 6 Artritis.....	19
Figura 7 Bursitis.....	20
Figura 8 Exoesqueleto robótico	21
Figura 9 Distribución de componentes	23
Figura 10 Exoesqueleto activo.....	25
Figura 11 Aleaciones con memoria de forma	32
Figura 12 Estructura cristalina de las AMF	34
Figura 13 Red neuronal.....	42
Figura 14 Arquitectura de red neuronal.....	44
Figura 15 Funcionamiento del algoritmo genético	52
Figura 16 Resultados de simulación regulada.....	68
Figura 17 Resultados de simulación para seguimiento	69
Figura 18 Error en simulación para q3.....	70
Figura 19 Medidas del cuerpo humano en función de la altura	71
Figura 20 Diagrama general de accionamiento del exoesqueleto	74
Figura 21 Diseño exoesqueleto	76
Figura 22 Antebrazo del exoesqueleto.....	77
Figura 23 Brazo del exoesqueleto.....	78
Figura 24 Características del codo-brazo y codo antebrazo.....	78
Figura 25 Ensamblaje del diseño mecánico.....	79
Figura 26 Láminas de nitinol	81
Figura 27 Prueba simulada del nitinol	81

Figura 28 Diagrama del sistema de control del exoesqueleto.....	82
Figura 29 Esquema del controlador de corriente del exoesqueleto.....	83
Figura 30 Diseño de la placa de circuito impreso (PCB) del exoesqueleto	84
Figura 31 Vista tridimensional de la placa de circuito impreso (PCB).....	86
Figura 32 Scrip de lectura de datos del dispositivo.....	87
Figura 33 Gráfica de la respuesta del sistema de segundo orden.....	88
Figura 34 Código para el ajuste y predicción de un modelo ARIMA.....	89
Figura 35 Porcentaje de acierto del modelo ARMA (2,1,1)	90
Figura 36 Porcentaje de acierto del modelo ARMA (3,1,1)	90
Figura 37 Porcentaje de acierto del modelo ARMA (2,2,1)	91
Figura 38 Archivos Generados por lectura de ejercicio.....	92
Figura 39 Configuración del puerto serial y la generación de un archivo CSV.....	93
Figura 40 Variables iniciales de recepción	94
Figura 41 Gráfico de variación de ángulos en un ejercicio simple	94
Figura 42 Código para resumir ciclos de ejercicios.....	95
Figura 43 Pestaña para modo de operación del exoesqueleto.....	96
Figura 44 Pestaña de evaluación de diferentes parámetros del exoesqueleto	97
Figura 45 Variables de resumen en cada ejercicio.....	97
Figura 46 Creación de dataset de ejercicios por paciente	99
Figura 47 Resultados de Accuracy y Loss en modelo A.	101
Figura 48 Matriz de confusión del modelo A	102
Figura 49 Resultados de Accuracy y Loss en modelo B.....	103
Figura 50 Matriz de confusión del modelo B	104
Figura 51 Resultados de Accuracy y Loss en modelo C.....	105
Figura 52 Matriz de confusión del modelo C	106
Figura 53 Resultados de Accuracy y Loss en modelo C mejorado.....	107
Figura 54 Matriz de confusion del modelo C mejorado	108
Figura 55 Resultado del F1 SCORE del modelo C mejorado.....	108

ÍNDICE DE TABLAS

Tabla 1	Cuadro de operacionalización de variables.....	58
Tabla 2	Tabla de referencia.....	66
Tabla 3	Tabla track	67
Tabla 4	Mediciones para el diseño ergonómico del exoesqueleto	73
Tabla 5	Evaluación de aleaciones	15
Tabla 6	Modelos de red neuronal propuestos	100

ÍNDICE DE ANEXOS

Anexo A: Matriz de Consistencia	116
Anexo B: Codificación De La Simulación De Ejercicios	122
Anexo C: Codificación red neuronal	132
Anexo D : Código de la clasificación de la red neuronal.....	143
Anexo E: Código simulación brazo robótico	145
Anexo F: Código Del Estado Transitorio	149
Anexo G: Código para el análisis de la planta	151
Anexo H: Codificación del resumen de los ejercicios	155

RESUMEN

Objetivo de este proyecto es mejorar la funcionalidad y la eficacia de un exoesqueleto robótico para el codo destinado a la rehabilitación y la asistencia, mediante su diseño y desarrollo con actuadores de alambre con memoria de forma como tecnología central. Con el fin de maximizar la ergonomía y la adaptabilidad del exoesqueleto a la anatomía del usuario, El método a utilizar es el explorativo, mediante el cual se explorarán diversas configuraciones mecánicas orientadas a mejorar la ergonomía y la adaptabilidad del exoesqueleto a la anatomía del usuario. Además, se seleccionarán cuidadosamente materiales protectores, duraderos y ligeros que proporcionen un equilibrio adecuado entre comodidad y funcionalidad. También se incluye el estudio tecnológico y la aplicación de actuadores de alambre con memoria de forma, que responden a estímulos térmicos para proporcionar un control preciso y flexible del movimiento. El resultado final es un exoesqueleto que puede proporcionar un control del movimiento fiable, ergonómico y flexible. También muestra el buen funcionamiento de los actuadores de alambre con memoria de forma como parte clave del proyecto y ofrece información importante sobre su practicidad y rendimiento. En conclusión, el dispositivo tiene el potencial de mejorar significativamente la calidad de vida de las personas con limitaciones de movilidad y ayudar en la rehabilitación del codo. Se ha consolidado como una opción técnica de vanguardia con aplicaciones tanto clínicas como prácticas en el campo de la rehabilitación.

Palabras claves: exoesqueleto, memoria de forma, rehabilitación, actuadores.

ABSTRACT

The objective of this project is to improve the functionality and effectiveness of a robotic exoskeleton for the elbow intended for rehabilitation and assistance, through its design and development with shape memory wire actuators as the core technology. In order to maximize the ergonomics and adaptability of the exoskeleton to the user's anatomy, an exploratory method will be used to explore various mechanical configurations aimed at improving the ergonomics and adaptability of the exoskeleton to the user's anatomy. In addition, protective, durable, and lightweight materials will be carefully selected to provide an appropriate balance between comfort and functionality. The technological study and application of shape memory wire actuators, which respond to thermal stimuli to provide precise and flexible movement control, are also included. The final result is an exoskeleton that can provide reliable, ergonomic, and flexible movement control. It also demonstrates the effective functioning of shape memory wire actuators as a key part of the project and provides important information about their practicality and performance. In conclusion, the device has the potential to significantly improve the quality of life of people with mobility limitations and assist in elbow rehabilitation. It has established itself as a cutting-edge technical option with both clinical and practical applications in the field of rehabilitation.

Keywords: exoskeleton, shape memory, rehabilitation, actuators.

PRÓLOGO

Este proyecto emerge de la convicción de que la tecnología puede ser un catalizador transformador para mejorar la calidad de vida y superar limitaciones físicas. Enfocado en la creación de un exoesqueleto robótico destinado a la asistencia y rehabilitación del codo.

El **Capítulo 1: Introducción** presenta una visión general del proyecto titulado "Diseño de un Exoesqueleto Robótico de Codo basado en Actuadores de Alambre de Aleación con Memoria de Forma", dirigido a mejorar la calidad de vida de individuos con limitaciones en la movilidad del codo. Se detalla la problemática actual de las lesiones y enfermedades del codo, que motivan la investigación, subrayando la necesidad urgente de soluciones innovadoras en rehabilitación. La justificación del proyecto resalta la importancia de integrar tecnologías avanzadas, como el Nitinol y las redes neuronales, para ofrecer una asistencia precisa y adaptable en la rehabilitación física. Se define claramente el problema central, enfocado en la creación de un dispositivo que combine eficacia y personalización en el tratamiento. Los objetivos generales y específicos se delinean para guiar el desarrollo del exoesqueleto, y se presenta una revisión de antecedentes relevantes que sitúan este trabajo en el contexto de los avances recientes en robótica médica y tecnología de aleación con memoria de forma.

El **Capítulo 2: Marco Teórico** se centra en proporcionar un fundamento sólido para el desarrollo del exoesqueleto robótico de codo. Se inicia con un análisis detallado de la anatomía y biomecánica del codo, lo que establece una comprensión clara de las condiciones físicas que el dispositivo pretende mejorar. Se examinan las aplicaciones de

la robótica en la rehabilitación, destacando su papel crucial en la asistencia médica y su capacidad para ofrecer soluciones personalizadas. El capítulo aborda la tecnología innovadora de las aleaciones con memoria de forma, específicamente el Nitinol, explicando cómo su capacidad para recordar y retornar a formas predefinidas se integra en el diseño de actuadores para el exoesqueleto. Además, se profundiza en el uso de redes neuronales para el control adaptativo del dispositivo, explorando cómo estas técnicas de aprendizaje automático permiten un ajuste preciso y en tiempo real a las necesidades del usuario. Los algoritmos genéticos se consideran para la optimización del rendimiento, ofreciendo potencial para mejorar la eficiencia operativa del exoesqueleto. Este marco teórico sustenta la investigación y diseño del proyecto, asegurando que se base en principios científicos sólidos y tecnología de vanguardia.

El Capítulo 3: Desarrollo del Trabajo de Investigación describe exhaustivamente el proceso de diseño y desarrollo del exoesqueleto robótico de codo, comenzando con el diseño conceptual del dispositivo. Se realiza una selección cuidadosa del actuador de alambre de aleación, eligiendo el Nitinol por sus propiedades únicas de memoria de forma, y se lleva a cabo un modelado detallado para optimizar su comportamiento. El análisis de la fatiga del exoesqueleto se aborda mediante simulaciones que consideran las condiciones operativas típicas, asegurando así la durabilidad y eficacia del dispositivo. Además, se implementa un análisis topológico que optimiza la estructura del exoesqueleto para garantizar un equilibrio óptimo entre peso y resistencia. El capítulo introduce un controlador basado en redes neuronales, que permite un control preciso y adaptativo del dispositivo, respondiendo a las necesidades específicas de cada usuario. Finalmente, este capítulo proporciona una visión completa del desarrollo técnico del proyecto, desde la concepción inicial hasta los análisis detallados y la implementación efectiva del controlador neural, asegurando que el exoesqueleto sea tanto

funcional como accesible para su uso en contextos de rehabilitación

El Capítulo 4: Análisis y Discusión de los Resultados se centra en la evaluación del desempeño del exoesqueleto robótico de codo, analizando su funcionalidad y eficacia a través de pruebas prácticas y simulaciones. Se ha implementado un modelo dinámico utilizando técnicas de filtrado para asegurar la precisión de las señales de control, eliminando el ruido estocástico y mejorando la calidad de la respuesta del sistema. Se examina cómo el exoesqueleto contribuye al proceso de rehabilitación, mostrando mejoras significativas en la movilidad y el apoyo proporcionado a los usuarios. Se realizaron análisis comparativos de varios modelos ARMA, eligiendo el que mejor se ajusta para predecir el comportamiento del sistema, logrando un alto nivel de precisión. Además, se identifican y discuten las limitaciones del estudio, como la necesidad de ajustar los modelos neuronales para mejorar la clasificación de las necesidades de apoyo durante los ejercicios. Este capítulo presenta un análisis crítico de los resultados obtenidos, destacando tanto los logros alcanzados como las áreas de oportunidad para futuras investigaciones y desarrollos.

Conclusiones y recomendaciones, en esta sección, se presentan las conclusiones derivadas de los resultados obtenidos, evaluando el grado de cumplimiento tanto del objetivo general como de los objetivos específicos propuestos en el diseño. Además, se ofrecen recomendaciones basadas en el desarrollo del diseño con el propósito de perfeccionar este trabajo y facilitar su eventual implementación o adaptación en contextos similares .

Referencias bibliográficas, en este apartado, se detallan las referencias de todas las fuentes consultadas durante la investigación, otorgando prioridad a recursos académicos y normativas legales.

I. INTRODUCCIÓN

En el panorama actual, la necesidad de soluciones tecnológicas para mejorar el estilo de vida de individuos con limitaciones físicas se ha vuelto más urgente que nunca. En este contexto, esta tesis, titulada "Diseño de un Exoesqueleto Robótico de Codo basado en Actuadores de Alambre de Aleación con Memoria de Forma", surge como una respuesta a la creciente demanda de tecnologías avanzadas que aborden las complejidades asociadas con las lesiones y enfermedades del codo.

La articulación del codo desempeña un papel esencial en la realización de actividades diarias, y las limitaciones en su movilidad pueden tener un impacto significativo en la autonomía y calidad de vida de las personas. La evolución de la tecnología de exoesqueletos robóticos representa una oportunidad única para proporcionar soluciones efectivas y personalizadas en el ámbito de la rehabilitación y el apoyo terapéutico.

En el centro de esta investigación, se encuentra la implementación de actuadores de alambre de aleación con memoria de forma, una tecnología que se adapta a la necesidad de respuestas ágiles y precisas a los movimientos del codo. Este componente esencial busca no solo restaurar la funcionalidad comprometida, sino también ofrecer una herramienta versátil y eficaz para mejorar la movilidad en situaciones de limitación física.

El enfoque de esta tesis se alinea con la actualidad, donde la integración de tecnologías avanzadas, como el uso de un controlador por redes neuronales aplicadas al control de exoesqueletos, se convierte en una necesidad imperante. La capacidad de estas redes para adaptarse y aprender permite una respuesta personalizada y eficiente, abriendo nuevas posibilidades para la aplicación efectiva de exoesqueletos en situaciones clínicas y

terapéuticas.

Este enfoque multidisciplinario no solo responde a la urgencia actual de soluciones tecnológicas para la rehabilitación, sino que también establece un camino hacia la integración exitosa de exoesqueletos robóticos en la atención médica contemporánea.

1.1 Planteamiento del problema

Según la Organización Mundial de la Salud (OMS, s.f.), anualmente 15 millones de personas sufren de accidentes cerebrovasculares siendo la más común el derrame cerebral. Esta enfermedad, la cual puede ocurrir por bloqueo de las arterias (isquemia cerebral) o sangrado interno (hemorragia cerebral), tiene entre sus consecuencias principales los problemas de comprensión y dificultad al hablar, problemas de visión y la pérdida de movimiento total o parcial del cuerpo en el lado opuesto al hemisferio afectado del cerebro. Sin embargo, se ha demostrado que esta última puede ser tratada mediante rehabilitación física, ya sea mediante el uso de dispositivos como exoesqueletos o la ayuda de un terapeuta (Vitiello, 2021).

De esta forma, podemos ver en los distintos hospitales y clínicas como los pacientes con estos problemas generalmente son asistidos por fisioterapeutas quienes les dan seguimiento y apoyo en su rehabilitación. Sin embargo, debido a que esto implica un trabajo constante en monitoreo y apoyo, el costo de estas terapias es muy elevado reflejándose así en las estadísticas de la Asociación Americana del Corazón (American Heart Association) quienes estimaron que el costo directo por cuidado médico y terapias ascendió a un promedio de 33 mil millones de dólares para el año 2004 (Organización Mundial de la Salud, s.f.).

Por otra parte, considerando que la efectividad de esta terapia se basa en la intensidad, repetitividad de los movimientos que realice el paciente, una propuesta prometedora para la mejora de estas terapias y su reducción de costos es el uso de sistemas

robóticos como exoesqueletos que puedan asistir y monitorear al paciente independientemente de la presencia del fisioterapeuta logrando así una terapia continua, más intensa y optima (Kwakkel, 2020). Si bien en la actualidad existen diversas propuestas comerciales de exoesqueletos, la mayoría de estas son estacionarias pues usan mecanismos metálicos de gran peso e incluyen actuadores en su mayoría motores eléctricos, hidráulicos, neumáticos o estimulaciones eléctricas funcionales (Functional Electrical Stimulation - FES) los cuales no solo adicionan un mayor peso al mecanismo sino también pueden ser costosos o hasta causarle dolor al paciente (Brose, 2020) Es en este sentido que aparecen los actuadores inteligentes con Efecto de Memoria de Forma (Shape Memory Alloys) los cuales pueden resolver las desventajas de las soluciones propuestas anteriores (Copaci, 2021) Sin embargo, al estar presente la histéresis en su dinámica, es muy difícil aplicar técnicas de control convencionales en estos actuadores por lo que se ve en la necesidad de aplicar técnicas de control robustas que brindan la precisión requerida en los movimientos repetitivos de estos exoesqueletos.

1.2 Descripción del problema

Asimismo, en este contexto crítico donde se inscribe la investigación, buscando no solo identificar la realidad problemática, sino también proponer una respuesta a través del diseño de un Exoesqueleto Robótico de Codo basado en Actuadores de Alambre de Aleación con Memoria de Forma. Este proyecto aspira a llenar el vacío existente en las soluciones de rehabilitación, brindando una alternativa innovadora y adaptativa que aborde directamente las necesidades actuales de las personas que enfrentan limitaciones en la movilidad del codo.

1.3 Formulación del problema

1.3.1 Problema general

¿De qué manera el diseño de un exoesqueleto robótico para la articulación del

codo basado en actuadores de alambre de aleación con memoria de forma logrará una mejora en la asistencia para rehabilitación de pacientes que sufrieron de accidentes cerebrovasculares?

1.3.2 Problemas específicos

- ¿De qué manera el modelado del comportamiento transitorio del actuador Nitinol con memoria de forma usando el método modelos autorregresivos y medias móviles (ARMA), permitirá mejorar la precisión en la simulación de los movimientos del exoesqueleto?
- ¿Cuál será el circuito electrónico que logre el control de la temperatura del alambre del Nitinol para lograr una fuerza de contracción?
- ¿Cómo el uso de un clasificador por redes neuronales puede mejorar la precisión del grado de ayuda requerida para una persona que requiere de rehabilitación?
- ¿Qué variables se deben tener en cuenta para el diseño mecánico requerido para el exoesqueleto?

1.4 Antecedentes

Copaci (2021) realiza el diseño preliminar de un exoesqueleto de codo accionado con actuadores de alambre de aleación con memoria de forma lo cual permite una drástica reducción del peso del exoesqueleto además de una característica de funcionamiento silencioso que aumenta la comodidad y facilita la portabilidad del conjunto. El diseño propuesto no consta de componentes rígidos para el movimiento de las articulaciones, esto tiene un impacto importante en cuanto a la comodidad de la interacción con el sistema. Estas características son suficientes para el uso de este exoesqueleto en la rehabilitación médica de pacientes que hayan sufrido de accidentes cerebrovasculares. Además, este es un sistema ligero y barato que puede construirse de forma personalizada para cada paciente (Zhang et al., 2021).

Kiguchi (2022) con el fin de desarrollar un exoesqueleto práctico y eficaz para la asistencia de movimiento de la articulación del hombro se desarrolló un controlador difuso neuronal, un mecanismo de movimiento del centro de rotación de la articulación del hombro y una interfaz de acoplamiento inteligente. El controlador difuso neuronal controla automáticamente el exoesqueleto mediante las señales de electromiograma de los músculos del hombro. Además, la interfaz inteligente propuesta anula los efectos nocivos causados por la postura del brazo. En un experimento preliminar se examinó un mapa no lineal entre la postura del brazo del sujeto y los coeficientes de modificación de las entradas del controlador difusivo neuronal para luego ser entrenado por una red neuronal para realizar la interfaz inteligente. Con esto la efectividad del método propuesto fue verificado mediante pruebas con personas de mediana edad y recomienda aplicar el sistema propuesto a personas de avanzada edad y personas discapacitadas (Lee, et al, 2022).

Cheng (2020) desarrolla el estudio de un sistema que proporciona una torsión de

asistencia al codo de los pacientes que es proporcional a la diferencia de las señales de electromiografía promedio de los bíceps y tríceps del paciente. Los resultados experimentales mostraron la eficacia del sistema propuesto en el cual se puede generar un gran par en la articulación del codo y la estabilidad del sistema con un suficiente tiempo de práctica. Se realizaron además una serie de experimentos para alcanzar los valores óptimos de los parámetros de filtrado adaptativo y de amortiguación no lineal. Con los valores de estos parámetros, los experimentos de seguimiento han confirmado que el sistema de torsión de asistencia propuesto puede aumentar la salida del par del codo en un 100% sin afectar el movimiento de la articulación. Por lo tanto, el sistema permite que los pacientes con accidentes cerebrovasculares que sufren de debilidad muscular puedan mover objetos pesados y ejercer fuerza de manera más eficaz (Wang et al., 2020).

Pylatiuk (2023) presenta un nuevo sistema de entrenamiento del codo para pacientes que sufren de paraplejia. Este sistema combina estímulos eléctricos funcionales y la fuerza externa adicional generada por actuadores hidráulicos. Se garantiza la seguridad del sistema por el cumplimiento inherente de los actuadores, por los topes mecánicos del sistema y por los momentos limitados que se puede aplicar a la extremidad superior. Este sistema es parte de un sistema modular que será desarrollado y también incluirán las articulaciones adyacentes superiores como hombros y muñecas. El funcionamiento hidráulico del sistema entrenamiento permite que su diseño sea compacto y portátil mientras que un sistema neumático es más factible a ser utilizados en pacientes que usan permanentemente una silla de ruedas ya que el tanque de alimentación puede ser montado en este (Kim y Chen, 2023).

Presenta un novedoso control PID lineal para un exoesqueleto portátil. Las condiciones de la estabilidad asintótica semi-global del exoesqueleto fueron definidas para

una arquitectura de control PID lineal. La ventaja del enfoque propuesto recae en que no se requiere de un modelo matemático completo del sistema para seleccionar los parámetros del controlador PID.

Los resultados de la simulación con el sistema EXO-UL7 validaron la metodología de diseño propuesta por el controlador PID. Es importante señalar que la metodología propuesta en este trabajo no conduce al conjunto óptimos de parámetros de un controlador PID lineal (García y Torres, 2024).

1.5 Justificación de la investigación

El interés en el desarrollo de nuevos dispositivos robóticos para su uso en rehabilitación médica se debe a su capacidad de realizar tareas repetitivas y porque permite analizar la evolución permanente del paciente. Entre las aplicaciones más destacadas de exoesqueletos en el ámbito médico encontramos: Rehabilitación por accidentes cerebrovasculares, recuperación de lesiones musculares, recuperación de lesiones en la médula espinal y estimulación muscular para personas adultas mayores. (Smith y Brown, 2020).

Siendo dentro de estas los accidentes cerebrovasculares una de las más comunes, entre las mencionadas, que no solo incapacita a las personas, pero en su gran mayoría conduce a la muerte. Según la Organización Mundial de la Salud alrededor de 15 millones de personas sufren de accidentes cerebrovasculares anualmente, estos accidentes cerebrovasculares resultan cuando hay poca sangre fluyendo al cerebro resultando en muerte celular en el mismo. Los efectos a largo plazo de estos accidentes cardiovasculares son la pérdida de memoria y hasta un 85% en dificultad motora sobre el cuerpo en la parte opuesta del cerebro donde sucedió el accidente cerebrovascular. Este trastorno se denomina hemiparesia o hemiplejía, que es una parálisis unilateral y restricción de

movimiento del todo el lado derecho o izquierdo del cuerpo y eventualmente se caracteriza por movimientos incontrolables o movimientos parciales. Las semanas posteriores a un accidente cerebrovascular son cruciales porque el daño puede ser evaluado, las alteraciones motoras resultantes pueden ser de debilidad muscular, pérdida de control articular y contracciones musculares incontroladas y es imperativo que se realice un proceso de rehabilitación controlado de la zona afectada.

Es en este contexto que esta tesis propone una solución alternativa para la rehabilitación de pacientes que hayan sufrido de accidentes cerebrovasculares con el diseño de un exoesqueleto de codo portátil, y compacto y ligero mediante el uso de actuadores de alambre con memoria de forma.

1.6 Limitaciones de la investigación

Como se ha mencionado anteriormente, un accidente cerebrovascular puede provocar parálisis parcial en el lado derecho o izquierdo del cuerpo, dependiendo del hemisferio cerebral afectado. En este contexto, el presente trabajo se enfoca en el diseño de un exoesqueleto para el codo, destinado a la rehabilitación muscular del brazo de la persona afectada, utilizando un actuador de alambre con memoria de forma.

A diferencia de otras alternativas comerciales de rehabilitación que emplean actuadores neumáticos, hidráulicos o motores, este trabajo propone el uso de actuadores de alambre con memoria de forma. La principal ventaja de esta tecnología es su capacidad para construir músculos artificiales que son compactos y ligeros. Además, este estudio se limita al diseño del exoesqueleto para una persona con características promedio en cuanto a peso, edad y estatura. En cuanto al control mediante redes neuronales, aunque existen diversos métodos de entrenamiento disponibles, este estudio se centra en el uso

de algoritmos genéticos. Esta técnica, considerada de vanguardia, ha demostrado proporcionar una convergencia global y más rápida en comparación con otros métodos, como los algoritmos basados en gradiente o los algoritmos estocásticos.

1.7 Objetivos

1.7.1 Objetivo general

Diseñar un exoesqueleto robótico para la articulación del codo utilizando actuadores de alambre de aleación con memoria de forma y un sistema de control por red neuronal para optimizar la rehabilitación de pacientes con accidentes cerebrovasculares.

1.7.2 Objetivos específicos

- Desarrollar y validar un modelo ARMA que capture con precisión la dinámica del actuador Nitinol, permitiendo una predicción precisa de su comportamiento bajo diversas condiciones operativas.
- Diseñar y optimizar las partes mecánicas del exoesqueleto para asegurar un ajuste ergonómico y funcional basado en medidas antropométricas estándar.
- Implementar y entrenar una red neuronal capaz de clasificar y adaptar el nivel de asistencia necesario para cada usuario, basándose en datos de rehabilitación recogidos, para proporcionar una asistencia personalizada y efectiva.
- Desarrollar un sistema de control basado en modulación por ancho de pulso (PWM) que regule la temperatura del alambre de Nitinol, asegurando una contracción adecuada y consistente del actuador.

1.8 Hipótesis

1.8.1 Hipótesis general

El diseño de un exoesqueleto robótico de codo basado en actuadores de alambre de aleación con memoria de forma y control por redes neuronales permitirá la mejora en la asistencia para rehabilitación de pacientes que sufrieron de accidentes cerebrovasculares.

1.8.2. Hipótesis específicas

El modelado del comportamiento transitorio del actuador de Nitinol con memoria de forma utilizando el método de los elementos finitos permitirá la obtención de una la curva de histéresis de su comportamiento con las dimensiones del material seleccionado.

El controlador por redes neuronales garantizará el movimiento preciso de la dinámica compleja de los actuadores con memoria de forma.

El entrenamiento por algoritmos genéticos servirá para alcanzar la convergencia global del controlador por redes neuronales

El controlador por redes neuronales brindará un funcionamiento estable y robusto del exoesqueleto tanto en el gemelo digital como en el prototipo físico experimental.

II. MARCO TEÓRICO

2.1 Marco conceptual

2.1.1 Estructuras anatómicas del codo

Para comprender la complejidad del codo y diseñar un exoesqueleto efectivo, es esencial examinar las estructuras anatómicas clave. Esta sección abordará la anatomía detallada, incluyendo huesos, articulaciones, músculos y ligamentos que componen la región del codo. Se destacarán las funciones específicas de cada estructura y su interrelación en el movimiento del codo.

El húmero es el hueso principal del brazo que forma la parte superior del codo. Destacan en él dos prominencias óseas llamadas epicóndilos, uno lateral y otro medial, que sirven como puntos de inserción para músculos y ligamentos, contribuyendo así a la estabilidad y movimiento del codo.

En el antebrazo, los huesos de nombres: cúbito y radio, participan activamente en la articulación del codo. Durante la pronación y supinación, el radio rota alrededor del cúbito, permitiendo una variedad de movimientos complejos del antebrazo, desde la flexión hasta la rotación.

Las articulaciones del codo incluyen la humeroulnar, que permite la flexión y extensión; la humerorradial, facilitando la pronación y supinación del antebrazo; y la radioulnar, que contribuye a la rotación del antebrazo. Estas articulaciones trabajan armoniosamente para brindar múltiples movimientos esenciales en actividades cotidianas. Los músculos son protagonistas fundamentales en la funcionalidad del codo. Los músculos flexores, como el braquial y el braquiorradial, son responsables de la flexión del codo, mientras que el tríceps braquial, como músculo extensor principal, juega un papel

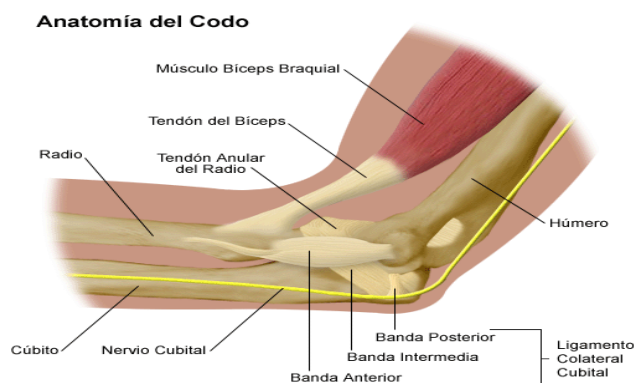
central en la extensión del codo. Además, músculos pronadores y supinadores, como el pronador redondo y el supinador, contribuyen a la pronación y supinación del antebrazo.

Los ligamentos proporcionan estabilidad a las articulaciones del codo. El ligamento colateral ulnar y el ligamento colateral radial refuerzan la articulación humeroulnar, mientras que el ligamento anular rodea la cabeza del radio, asegurando la articulación radioulnar.

Las bursas, pequeñas bolsas llenas de líquido, y la baina sinovial, una membrana que recubre las articulaciones, desempeñan un papel crucial en la reducción de la fricción y en la lubricación de las superficies articulares, facilitando el movimiento suave del codo.

Figura 1

Anatomía del codo



Fuente: Reimpreso de Muay Thai Training, por muaythaitraining, s.f.,

WordPress. <https://muaythaitraining.wordpress.com/>

2.1.2 Biomecánica del codo

La biomecánica del codo es un campo para entender cómo esta articulación compleja realiza movimientos y se somete a fuerzas durante diversas actividades. El análisis biomecánico del codo implica examinar cómo las estructuras anatómicas interactúan para permitir y controlar el movimiento. A continuación, se exploran los aspectos clave de la biomecánica del codo:

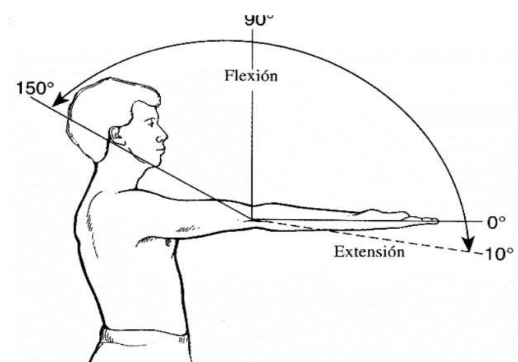
Movimientos básicos

La biomecánica del codo involucra principalmente dos movimientos básicos:

- **Flexión y extensión:** La articulación humeroulnar permite la flexión (doblar) y extensión (enderezar) del codo. Durante la flexión, los músculos flexores, como el bíceps braquial, se contraen, acortando la distancia entre el antebrazo y el brazo. En la extensión, los músculos extensores, como el tríceps braquial, entran en acción.

Figura 2

Flexión y extensión del codo



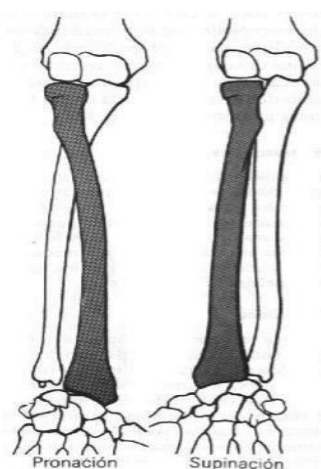
Fuente: Reimpreso de Balance articular del codo, por traumazaragoza, s.f.,

http://www.traumazaragoza.com/Documentacion_files/Balance%20articular%20del%20codo.pdf

- **Pronación y supinación:** La articulación radioulnar es responsable de la pronación (rotación hacia adentro) y supinación (rotación hacia afuera) del antebrazo. Estos movimientos son cruciales para actividades como girar las palmas hacia arriba o abajo.

Figura 3

Pronación y supinación



Fuente: Reimpreso de Arthrology Upper and Lower Extremity, por Masaryk University, s.f.,

https://is.muni.cz/el/med/podzim2018/aVLAN0121p/um/Arthrology_Upper_and_lower_extremity.pdf

Fuerzas y tensiones

La biomecánica del codo implica la aplicación de fuerzas y tensiones específicas durante diversas actividades. Durante la flexión y extensión, se generan fuerzas de compresión y tracción en las superficies articulares del húmero y el cúbito, permitiendo el movimiento suave y controlado del codo. Estas fuerzas son distribuidas a través de los cartílagos articulares, que actúan como amortiguadores para reducir la presión en las articulaciones y prevenir el desgaste prematuro.

La pronación y supinación añaden una dimensión adicional a la biomecánica, generando fuerzas de torsión en la articulación radioulnar. Esta rotación es esencial para la posición y alineación adecuadas del antebrazo durante actividades como sostener objetos y manipular herramientas. La comprensión de estas fuerzas es crucial para diseñar el exoesqueleto de manera que proporcione soporte sin comprometer la capacidad natural del codo para resistir y distribuir estas tensiones.

Músculos y tendones

La función biomecánica del codo está intrínsecamente vinculada a la acción coordinada de músculos y tendones. Durante la flexión, los músculos flexores, como el bíceps braquial, se contraen, generando la fuerza necesaria para doblar el codo. En contraste, la extensión del codo implica la activación del tríceps braquial y otros músculos extensores, produciendo la fuerza requerida para estirar el brazo.

La pronación y supinación se logran mediante la acción de los músculos pronadores y supinadores. El pronador redondo y el supinador trabajan conjuntamente para permitir la rotación del antebrazo. La biomecánica efectiva del codo implica una coordinación precisa de estos grupos musculares y tendones, asegurando una ejecución suave y controlada de los movimientos.

Estabilidad articular

La estabilidad del codo se mantiene gracias a una red de ligamentos que conectan los huesos y limitan los movimientos no deseados. El ligamento colateral ulnar y el ligamento colateral radial refuerzan la articulación humeroulnar, evitando movimientos excesivos hacia los lados. El ligamento anular, por su parte, asegura la estabilidad de la articulación radioulnar, permitiendo la rotación controlada del antebrazo.

Considerar estas estructuras contribuirá a la creación de un exoesqueleto que no solo brinde asistencia, sino que también preserve la estabilidad articular esencial para el funcionamiento óptimo del codo.

2.1.3 Lesiones y enfermedades del codo

El codo, siendo una articulación esencial para la funcionalidad del brazo, está sujeto a diversas lesiones y enfermedades que pueden repercutir de manera negativa la movilidad del mismo. La comprensión detallada de estas patologías es crucial para el diseño efectivo de un exoesqueleto robótico.

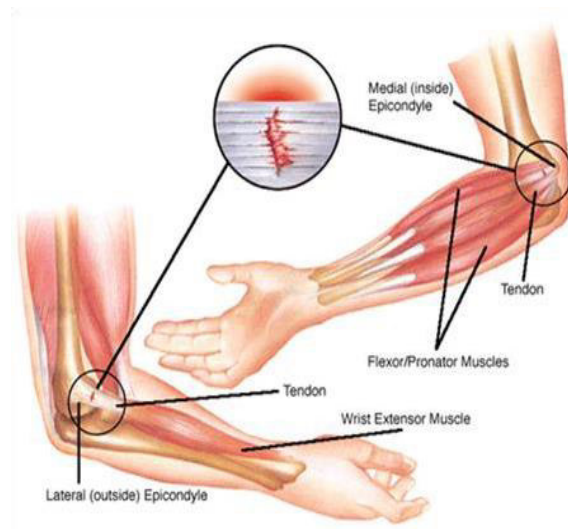
Las enfermedades del codo son frecuentes en personas que realizan actividades deportivas, las más comunes suelen ser:

Epicondilitis (codo de tenista y golfista)

La epicondilitis lateral (codo de tenista) y la epicondilitis medial (codo de golfista) son lesiones que afectan los epicóndilos del codo. Estas condiciones, caracterizadas por la inflamación de los tendones en la región, resultan comúnmente de movimientos repetitivos o tensiones excesivas en los músculos de la parte externa o interna del antebrazo.

Figura 4

Epicondilitis



Fuente: Reimpreso de Las lesiones por el esfuerzo repetitivo en la restauración: Consecuencias y prevención, por DiegoCoquillat, 2020, <https://www.diegocoquillat.com/las-lesiones-por-el-esfuerzo-repetitivo-en-la-restauracion-consecuencias-y-prevencion/>

Luxaciones y fracturas del codo

Las luxaciones y fracturas en el codo pueden ocurrir debido a traumas, caídas o impactos directos. Estas lesiones pueden afectar la estabilidad y funcionalidad del codo, requiriendo una atención cuidadosa en el diseño del exoesqueleto para asegurar la inmovilización adecuada durante el proceso de curación y la rehabilitación.

Figura 5*Fractura del codo*

Fuente: Reimpreso de Apoptotic bodies for advanced drug delivery and therapy, 2017, <https://www.sciencedirect.com/science/article/abs/pii/S113835931730031X>

Tendinitis y lesiones de tendones

La tendinitis, inflamación de los tendones, y las lesiones de tendones, como las roturas, pueden surgir por el desgaste crónico o movimientos repetitivos. Estas condiciones pueden afectar la capacidad de flexión y extensión del codo, así como la fuerza de agarre. Un exoesqueleto debe considerar proporcionar asistencia y apoyo específico para estas funciones comprometidas.

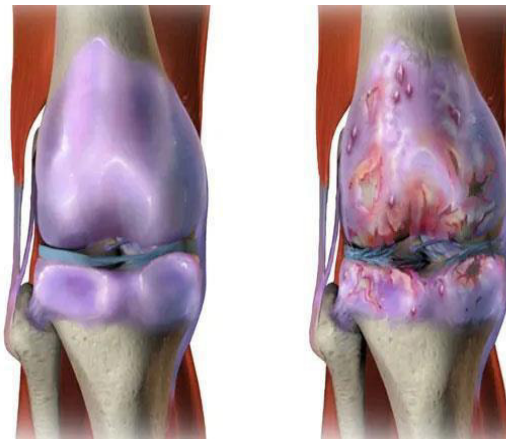
Artritis del codo

La artritis, una condición degenerativa de las articulaciones, puede afectar el codo, provocando dolor y limitando el rango de movimiento. El exoesqueleto debe

abordar la estabilidad y la movilidad de manera equilibrada, ofreciendo alivio sin comprometer la funcionalidad.

Figura 6

Artritis



Nota: Pylativk, A. (2020). Imagen de articulación afectada por artritis. Sitio web. <https://mederinutricion.com/tag/nutricion/>

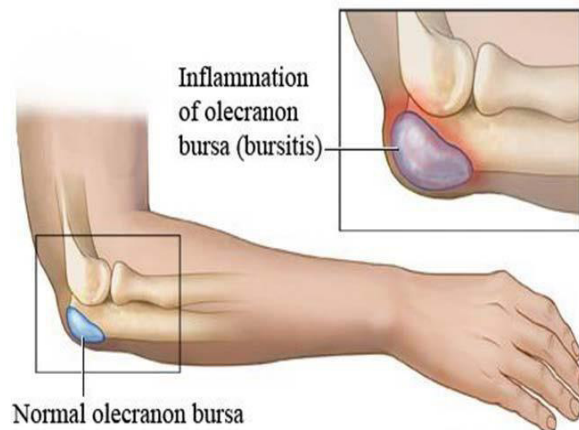
Fuente: Reimpreso de meinutricion, por Méderi Nutrición, 15 enero, 2020, <https://mederinutricion.com/tag/nutricion/>

Bursitis del codo

La bursitis, la inflamación de las bolsas sinoviales, puede afectar el codo. El diseño del exoesqueleto debe minimizar la presión sobre estas áreas inflamadas, brindando soporte sin causar molestias adicionales.

Figura 7

Bursitis



Fuente: Reimpreso de artículo sobre bursitis olecraniana, por Martín Vasco, 2025,

<https://www.martinvasco.es/patologias/codo/bursitis-olecraniana-del-estudiante/>

Lesiones del nervio cubital

Las lesiones del nervio cubital pueden provocar debilidad y entumecimiento en el codo y la mano. El diseño del exoesqueleto debe considerar la distribución de la presión para evitar la compresión del nervio, asegurando así la integridad neuromuscular.

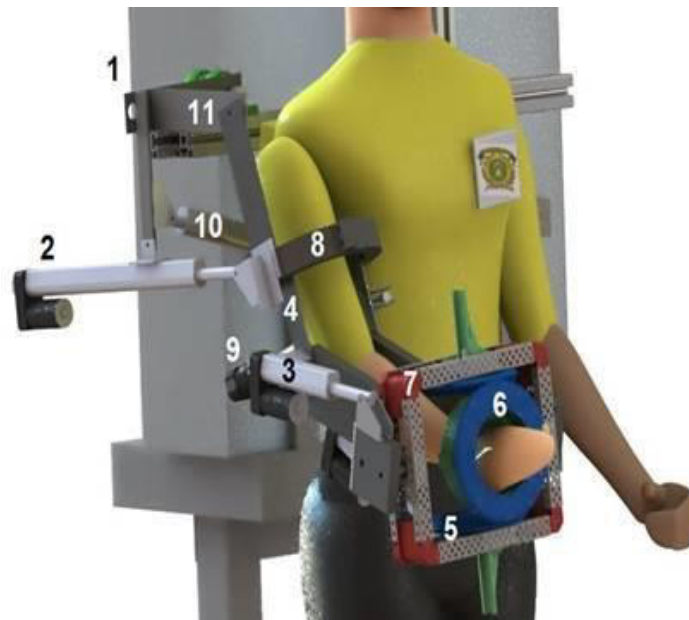
2.2 Exoesqueleto robótico

Un exoesqueleto robótico, conocido también como exoesqueleto mecánico, es un dispositivo portátil diseñado para proporcionar movimiento a una extremidad, ya sea inferior o superior. Estos dispositivos tienen una variedad de aplicaciones, que van desde potenciar las capacidades humanas hasta apoyar la rehabilitación de lesiones y enfermedades. Los exoesqueletos actúan como un soporte y asisten en los movimientos

que puede realizar una persona; pueden ser estructuras pasivas o activas. En otras palabras, un exoesqueleto puede carecer de actuadores para el movimiento, lo que significa que no requiere un sistema de control asociado a los actuadores. Sin embargo, en la actualidad, la mayoría de los exoesqueletos desarrollados en el ámbito médico se adaptan al cuerpo mediante sistemas de procesamiento y sensado para tomar decisiones precisas con el fin de llevar a cabo una actividad predefinida. Si bien muchos exoesqueletos se enfocan en abordar afecciones en las extremidades inferiores, también se han realizado investigaciones para desarrollar exoesqueletos para las extremidades superiores, como los brazos y las manos (articulaciones del codo, el hombro, las muñecas y los dedos), aunque en menor medida. (Cruz 2021 p. 17)

Figura 8

Exoesqueleto robótico



Modelo en CAD del ensamblaje del ERMIS. Adaptado de Cruz Martínez (2021).

Fuente: Reimpreso de artículo sobre exoesqueletos robóticos, por Redalyc, 2025, <https://www.redalyc.org/journal/619/61955183007/html/>

2.2.1 Partes y componentes

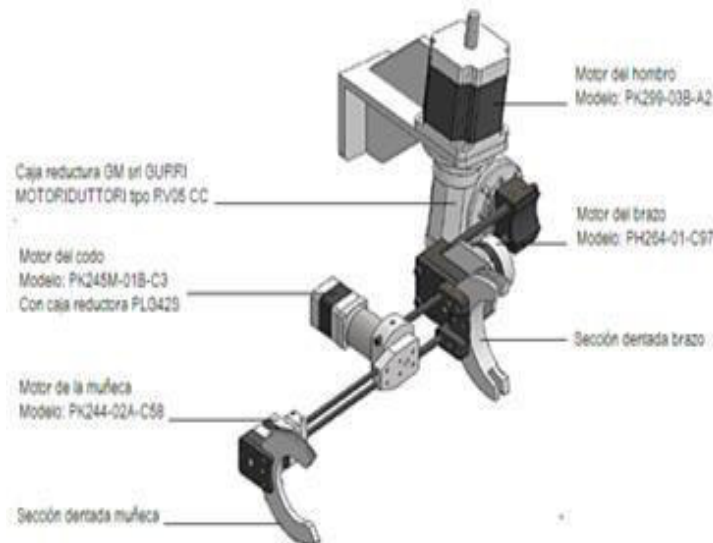
Las partes y componentes de un exoesqueleto robótico son fundamentales para entender su funcionamiento y su aplicación.

- **Estructura Mecánica:** Proporciona el armazón físico que sostiene y alinea las articulaciones humanas. Esta estructura está diseñada para ser resistente, liviana y ergonómica, permitiendo movimientos naturales y cómodos.
- **Actuadores:** Son los motores o dispositivos que generan el movimiento en las articulaciones del exoesqueleto. Pueden ser eléctricos, hidráulicos o neumáticos, y tienen como propósito brindar la fuerza necesaria para asistir, potenciar o rehabilitar los movimientos del usuario (Auricchio, 2021)
- **Sensores:** Los sensores son dispositivos que detectan y registran información sobre la posición, la velocidad, la fuerza y otros parámetros relevantes del usuario y del entorno. Estos datos son utilizados por el sistema de control para adaptar la asistencia proporcionada por el exoesqueleto de manera precisa y segura
- **Sistema de Control:** El sistema de control del exoesqueleto procesa la información de los sensores, utiliza algoritmos y lógica de control para ordenar la magnitud de la fuerza y la precisión en la asistencia que se proporcione al usuario. Este sistema puede ser autónomo o estar controlado por el usuario o un terapeuta
- **Interfaz Usuario-Exoesqueleto:** La interfaz usuario-exoesqueleto es una conexión entre el usuario y el sistema electrónico-digital del exoesqueleto. Puede incluir elementos como correas ajustables, almohadillas acolchadas y sistemas de fijación con pulsadores para garantizar un ajuste cómodo y seguro
- **Fuente de Energía:** La fuente de energía proporciona la potencia necesaria para alimentar los actuadores y otros componentes electrónicos del

exoesqueleto. Puede ser una batería recargable u otra fuente de energía portátil

Figura 9

Distribución de componentes



Fuente: Reimpreso de Exoesqueleto robótico para la rehabilitación del miembro superior del paciente hemipléjico, por Broche, Torres, Milanés, Gonzales, Rodriguez, Sagaró, 2020. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-594420200003000003

2.2.2 Tipos de exoesqueletos robóticos

Los exoesqueletos robóticos pueden clasificarse en varios tipos según su diseño, su función y su aplicación.

Exoesqueletos pasivos

- Pueden ser contruidos con materiales ligeros y resistentes como fibra de carbono o aleaciones de aluminio para minimizar la carga sobre el usuario.

- Su diseño se centra en redistribuir las fuerzas biomecánicas de manera eficiente para reducir la fatiga muscular y mejorar la postura.
- Se utilizan comúnmente en entornos industriales para ayudar a los trabajadores a levantar objetos pesados y mantener una postura ergonómica.

Exoesqueletos activos:

- Utilizan actuadores, como motores eléctricos, hidráulicos o neumáticos, para proporcionar asistencia activa en los movimientos del usuario.
- Los algoritmos de control permiten ajustar la cantidad de asistencia según las necesidades del usuario y la tarea específica que se esté realizando.
- Se emplean en la rehabilitación de lesiones musculoesqueléticas y neurológicas, así como en aplicaciones industriales que requieren levantamiento repetitivo de cargas pesadas.

Figura 10*Exoesqueleto activo*

Fuente: Freepik (2023). Exoskeleton for warehouse logistics.

<https://www.freepik.comhttps://stock.adobe.com/es/images/high-tech-futuristic-warehouse-worker-wearing-advanced-full-body-powered-exoskeleton-lifts-heavy-pallet-full-of-cardboard-boxes-delivery-exosuit-amplifies-strength/389648656>

Exoesqueletos híbridos:

- Combinan características de los exoesqueletos pasivos y activos para ofrecer una solución más versátil.
- Pueden tener secciones pasivas para proporcionar soporte estructural y secciones activas con actuadores para asistir en movimientos específicos.
- Su diseño se adapta a una variedad de aplicaciones, desde la asistencia en la marcha hasta la rehabilitación de lesiones específicas.

Exoesqueletos robóticos de rehabilitación:

- Diseñados con modos de operación ajustables que permiten adaptarse a diferentes etapas de la rehabilitación.
- Pueden proporcionar asistencia controlada, resistencia variable y retroalimentación visual o háptica para guiar al paciente durante la terapia.
- Se utilizan en la recuperación de lesiones de la médula espinal, accidentes cerebrovasculares y traumatismos craneoencefálicos, entre otras afecciones neurológicas y musculoesqueléticas.

2.2.3 Aplicaciones en diversos sectores

Los exoesqueletos robóticos trabajan en distintos sectores. Aquí se presentan algunas de las aplicaciones más destacadas:

Sector médico y de rehabilitación:

En el ámbito médico y de rehabilitación, los exoesqueletos robóticos se han transformado en instrumentos fundamentales que ayudan en la recuperación y en mejorar la calidad de vida de pacientes con diversas condiciones de salud. Estos dispositivos proporcionan asistencia controlada y personalizada durante la rehabilitación física, permitiendo a los pacientes recuperar la movilidad y el funcionamiento de las articulaciones afectadas. Desde personas que tienen lesiones neurológicas como accidentes cerebrovasculares hasta aquellos con afecciones musculoesqueléticas, los exoesqueletos ofrecen un enfoque innovador para la terapia física. Utilizando algoritmos de control avanzados y sensores precisos, estos dispositivos pueden adaptarse a las necesidades específicas de cada paciente, proporcionando asistencia gradual a medida que se avanza en el proceso de rehabilitación. Además, los exoesqueletos brindan un medio seguro y efectivo para realizar movimientos repetitivos y controlados, lo que es crucial para fortalecer los músculos y mejorar la coordinación motora.

En resumen, en el sector médico y de rehabilitación, los exoesqueletos robóticos representan una herramienta innovadora que promueve la recuperación funcional y contribuye a elevar el bienestar del usuario. (López, 2024)

Sector industrial:

En este sector, los exoesqueletos robóticos tienen un rol importante en la prevención de lesiones laborales y el aumento de la productividad de los trabajadores. Estos dispositivos están diseñados para reducir la fatiga muscular y prevenir lesiones musculoesqueléticas en trabajadores que realizan tareas físicamente exigentes. Proporcionan soporte adicional y asistencia durante la realización de movimientos repetitivos o el manejo de cargas pesadas, lo que ayuda a minimizar el riesgo de lesiones relacionadas con el trabajo. Además, los exoesqueletos permiten a los trabajadores mantener una postura ergonómica y reducen la tensión en las articulaciones y los músculos, lo que contribuye a una mayor comodidad y bienestar en el lugar de trabajo.

En términos de productividad, estos dispositivos permiten a los trabajadores realizar labores cotidianas de manera más fácil e imprimiendo menos fuerza, lo que brinda como resultado en un aumento significativo de la producción y la eficiencia operativa. En resumen, en el sector industrial, los exoesqueletos robóticos representan una solución innovadora para mejorar la seguridad laboral y optimizar los procesos de trabajo. (Goldiez, 2025)

Sector militar:

Dentro del ámbito militar, han revolucionado las capacidades operativas de las fuerzas armadas al proporcionar a los soldados una mayor resistencia y capacidad de carga. Estos dispositivos permiten a los soldados llevar equipo pesado durante largos

períodos de tiempo y en entornos desafiantes, lo que aumenta su movilidad y capacidad de maniobra en el campo de batalla. Además, los exoesqueletos reducen el riesgo de lesiones musculoesqueléticas al proporcionar soporte adicional durante el transporte de equipo pesado, lo que contribuye a mantener la fuerza y la movilidad de los soldados durante operaciones prolongadas. En términos de seguridad, estos dispositivos también ofrecen una protección adicional al reducir la fatiga y el estrés físico de los soldados, lo que puede ayudar a prevenir lesiones y mejorar la efectividad de las operaciones militares. En resumen, en el sector militar, los exoesqueletos robóticos representan una herramienta innovadora que mejora la resistencia, la movilidad y la defensa de los soldados en las líneas de combate. (CyberHS, 2024)

Sector deportivo:

En este ámbito están ganando popularidad como herramientas de entrenamiento y rehabilitación. Estos dispositivos se utilizan para aumentar la potencia y la técnica de los atletas profesionales y aficionados. Al proporcionar asistencia controlada durante el entrenamiento, los exoesqueletos permiten a los deportistas superar sus límites físicos y mejorar su rendimiento en diversas disciplinas deportivas. Además, en el proceso de rehabilitación deportiva, los exoesqueletos brindan una forma segura y efectiva de recuperarse de lesiones musculares y articulares, permitiendo a los atletas volver a la actividad física de manera gradual y controlada. En resumen, en el sector deportivo, los exoesqueletos robóticos representan una herramienta innovadora que mejora el rendimiento atlético y facilita la recuperación de lesiones. (Morán, 2024)

Sector de la construcción y la agricultura:

En el ámbito de la edificación y labranza, juegan un rol fundamental en la

prevención de lesiones laborales y el aumento de la eficiencia en el trabajo. Estos dispositivos proporcionan soporte adicional y asistencia, mientras se lleva a cabo las tareas físicamente exigentes, por ejemplo, alzar cargas pesados o trabajar en posiciones incómodas. Al reducir la fatiga muscular y mejorar la postura del cuerpo, los exoesqueletos ayudan a prevenir lesiones musculoesqueléticas y aliviar la salud laboral de los empleados. Además, en el campo agrícola, estos dispositivos facilitan la realización de actividades como la siembra, la cosecha y el transporte de productos, aumentando la productividad y la eficiencia en la producción de alimentos. En resumen, en el ámbito de la edificación y labranza, los exoesqueletos robóticos representan una solución innovadora para mejorar la seguridad laboral y optimizar los procesos de trabajo. (Hapo, 2024)

Sector de la realidad virtual y los videojuegos:

En este sector en especial, ofrecen una experiencia de juego más inmersiva y realista. Estos dispositivos permiten a los usuarios participar físicamente en entornos virtuales y simulaciones, proporcionando retroalimentación háptica y sensaciones táctiles que mejoran la interacción y la inmersión del usuario. Además de su uso en entretenimiento, los exoesqueletos también se utilizan en aplicaciones educativas y de entrenamiento para simular situaciones y actividades realistas, optimizando la experiencia educativa y el fomento de competencias. En resumen, en el sector de la realidad virtual y los videojuegos, los exoesqueletos robóticos representan un artefacto que revoluciona la manera en que interactuamos con el mundo digital y mejoramos nuestras habilidades cognitivas y físicas. (Cheng, 2020)

2.3 Aleación con memoria de forma

Las aleaciones con memoria de forma (AMF) comprenden un conjunto de aleaciones metálicas que exhiben dos características macroscópicas distintivas: el efecto de memoria de forma (EMF) y la pseudoelasticidad. El EMF describe el proceso mediante el cual el material recobra una forma previamente “memorizada” después de haber sido deformado, una vez que se le somete a un aumento de temperatura por encima de un rango específico. Por otro lado, la pseudoelasticidad se refiere al fenómeno en el cual el material puede experimentar deformaciones significativas que pueden ser reversibles sin sufrir deformación plástica, incluso bajo condiciones isotérmicas. (Annadata, 2024)

La investigación sobre las aleaciones con memoria de forma se remonta a los estudios sobre la transformación martensítica a principios del siglo XX. Inicialmente, este fenómeno, observado en aleaciones de FeC, se consideraba irreversible hasta que se descubrió martensita térmicamente reversible en las aleaciones CuZn y CuAl en 1949, introduciendo así el concepto de transformación martensítica termoelástica (Kurdyumov, 1949). Previamente, en 1932, Arne Olander (2023) fue el primero en notar una de las características distintivas de los AMF. Informó sobre un comportamiento altamente elástico, similar al del caucho, en aleaciones de AuCd con un 47,5% de cadmio. Esta propiedad, descrita por Olander (2023), se denominó pseudoelasticidad en 1958. En relación al efecto de memoria de forma, Chang (2022) publicó un artículo en 1951 describiendo cómo una varilla de AuCd recuperó su forma original al ser calentada a la fase cúbica (austenítica) después de haberse deformado permanentemente en la fase martensítica. (Brose et al. 2020) Un año después, Reynolds y Bever también documentaron este fenómeno en las aleaciones de CuZn.

La investigación sobre la transformación martensítica térmicamente reversible continuó durante la segunda mitad del siglo XX, y otras aleaciones como InTi (M. W.

Burkart, 2023), CuZn (Genevray, 1953), CuAlNi (Suoninen, 1954) o AuAgCd (Chen, 1957) se encontraron que exhibían EMF y pseudoelasticidad. No obstante, a pesar de la cantidad de aleaciones que presentan estos efectos, nunca llegaron a constituir un campo de investigación separado, en parte porque sus propiedades mecánicas impedían su uso en aplicaciones de ingeniería. Este escenario cambió en 1961 con el descubrimiento del EMF en aleaciones de NiTi.

Como muchos descubrimientos científicos, el fenómeno de EMF fue observado en el NiTi por accidente. A finales de la década de 1950, William J. Buehler y otros investigadores estaban estudiando diferentes aleaciones para ser utilizadas en blindaje térmico de misiles y, entre los diferentes materiales probados, el NiTi se destacó por sus buenas propiedades mecánicas (W. Buehler).

El descubrimiento del EMF en este material, conocido como Nitinol tras la combinación de los nombres de los elementos de la aleación y el acrónimo del Laboratorio de Artillería Naval de los EE. UU. (NOL), inició un esfuerzo de investigación por parte de varios grupos para comprender completamente los mecanismos detrás de los fenómenos observados. Inicialmente, se propusieron varias teorías que implicaban diferentes procesos cristalográficos, pero a principios de la década de 1970, el consenso era que el Nitinol era el resultado de la misma transformación cristalográfica descubierta en AuCd décadas antes. En 1972, la NASA publicó un informe completo que detallaba la cristalografía, las técnicas de procesamiento, las propiedades físicas, mecánicas y químicas, así como las aplicaciones potenciales del Nitinol Jackson (2024). El avance de Nitinol y otras aleaciones de NiTi marca el comienzo de la utilización comercial de las AMF. A diferencia de las aleaciones previas, este nuevo material poseía las características mecánicas que lo hacían práctico en diversas aplicaciones, especialmente en los ámbitos

médico e ingenieril. Aparte del EMF y la pseudoelasticidad, las propiedades que volvían atractivas a las aleaciones de NiTi para su empleo en el campo de la medicina incluían su biocompatibilidad, resistencia a la torsión y a la corrosión.

Figura 11

Aleaciones con memoria de forma



Nota: Reimpreso de Departamento de Ingeniería y arquitectura, por Ahmed Nabih,

Fuente:

https://jes.journals.ekb.eg/article_19524_b73d6cc703ce06b5d4bcb1d83035a200.pdf

2.3.1 Propiedades y comportamiento de las aleaciones con memoria de forma

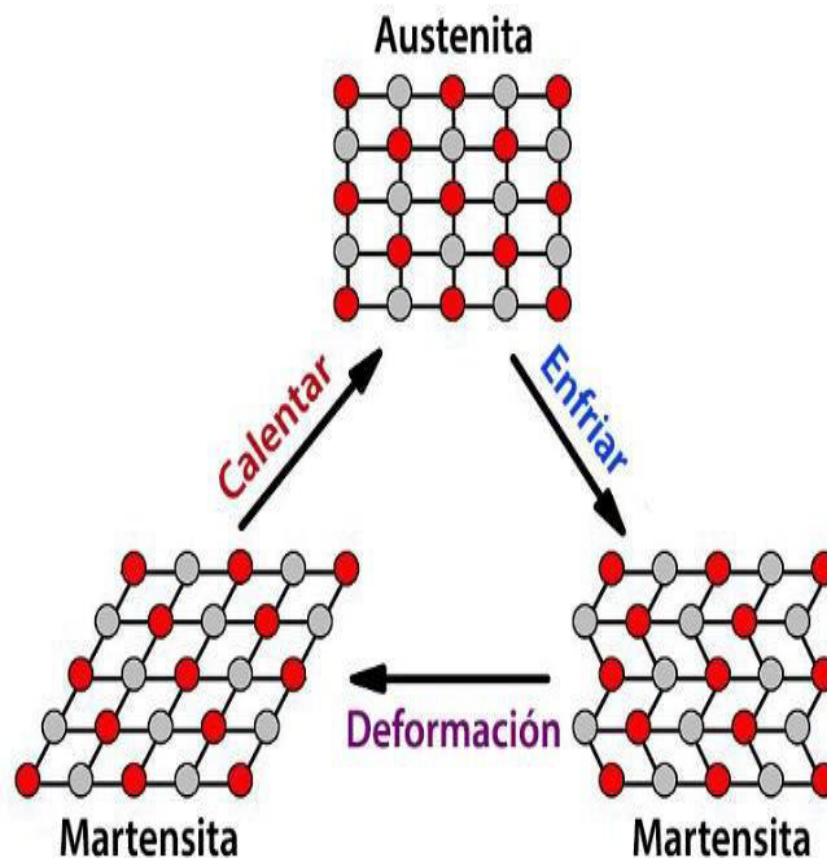
Composición y Estructura Cristalina

Las aleaciones con memoria de forma (AMF) se caracterizan por su composición y estructura cristalina, que son determinantes para sus propiedades y comportamiento funcional. Estas aleaciones son generalmente una combinación de varios metales, siendo los más comunes el níquel (Ni) y el titanio (Ti), aunque también pueden incluir otros elementos como cobre (Cu), aluminio (Al), hierro (Fe), manganeso (Mn) y cromo (Cr), entre otros. La proporción y combinación de estos elementos metálicos en la aleación determina sus propiedades finales, como la temperatura de transformación, la resistencia

mecánica y la biocompatibilidad.

La estructura cristalina de las AMF es crucial para entender su comportamiento. Las AMF suelen exhibir una estructura cristalina de tipo martensita, que es una fase metaestable con una disposición específica de los átomos. La martensita es capaz de formarse durante la deformación a bajas temperaturas y puede revertirse durante la deformación a temperaturas más altas, lo que permite que la aleación recupere su forma original. Esta capacidad de recuperación elástica es la base de la propiedad de memoria de forma, que es una de las características distintivas de las AMF.

La estructura cristalina de las AMF puede tener una variación según sus variables, por ejemplo, la composición de la aleación, el proceso de fabricación y el tratamiento térmico al que se somete la aleación. La comprensión detallada de la estructura cristalina es fundamental para diseñar aleaciones con propiedades específicas y para controlar su comportamiento en diversas aplicaciones.

Figura 12*Estructura cristalina de las AMF*

Fuente: Reimpreso de <https://www.areametalurgia.com/post/c%C3%B3mo-funciona-las-aleaciones-con-memoria-de-forma>

Propiedades térmicas y mecánicas

Las AMF exhiben una variedad de propiedades térmicas y mecánicas que las hacen únicas y versátiles en una variedad de aplicaciones. A continuación, se describen algunas de estas:

Propiedades térmicas:

- **Temperatura de transformación:** Las AMF tienen temperaturas de transformación bien definidas, que son los puntos en los cuales ocurren cambios estructurales significativos. La temperatura de transformación martensítica es aquella en la cual la aleación pasa de una estructura cristalina a otra, lo que generalmente está asociado con la memoria de forma y la pseudoelasticidad.
- **Estabilidad térmica:** Las AMF son estables térmicamente dentro de ciertos rangos de temperatura. Esto significa que mantienen sus propiedades y estructuras cristalinas dentro de estos rangos y pueden experimentar ciclos repetidos de deformación y recuperación sin perder sus características fundamentales.

Propiedades mecánicas:

- **Elasticidad y resiliencia:** Una de las propiedades más destacadas de las AMF es su habilidad de volver a su estructura inicial después de ser deformadas. Esta elasticidad se basa en la capacidad de la aleación para cambiar entre diferentes fases cristalinas en respuesta a estímulos externos como el cambio de temperatura.
- **Resistencia a la fatiga:** Las AMF pueden soportar ciclos repetidos de deformación y recuperación sin experimentar una degradación significativa en sus propiedades mecánicas. Esto las hace ideales para aplicaciones que requieren un alto grado de durabilidad y resistencia a la fatiga.
- **Ductilidad y tenacidad:** Aunque las AMF pueden ser altamente elásticas, también pueden ser lo suficientemente dúctiles como para resistir la fractura bajo

cargas mecánicas extremas. Esta combinación de elasticidad y tenacidad las hace adecuadas para una amplia gama de aplicaciones estructurales y de ingeniería.

Comportamiento de transformación de fase

El comportamiento de transformación de fase es una característica fundamental de las aleaciones con memoria de forma (AMF) y define su capacidad para cambiar de forma reversiblemente en respuesta a estímulos externos, como cambios de temperatura o tensiones mecánicas.

Transformación martensítica: La transformación martensítica es el fenómeno clave que impulsa el comportamiento de memoria de forma en las AMF. Ocurre cuando la aleación cambia de una fase cristalina a otra de manera reversible al ser sometida a un estímulo, como un cambio de temperatura. Durante este proceso, la aleación puede cambiar su forma de manera reversible, lo que permite que recupere su forma original después de ser deformada.

Temperatura de transformación: Cada aleación con memoria de forma tiene una temperatura de transformación característica, que marca el punto en el cual ocurre la transformación martensítica. La temperatura de transformación varía según la composición de la aleación y puede ajustarse mediante el control de los elementos metálicos y el tratamiento térmico.

Histeresis térmica: La transformación martensítica en las AMF exhibe una histeresis térmica, lo que significa que la transformación ocurre a diferentes temperaturas para el calentamiento y el enfriamiento. La temperatura a la cual comienza la transformación durante el enfriamiento se conoce como temperatura de inicio de la transformación (M_s), mientras que la temperatura a la cual finaliza durante el

calentamiento se llama temperatura final de la transformación (A_f). La diferencia entre M_s y A_f se conoce como histeresis térmica.

Efecto de Memoria de Forma (EMF): El efecto de memoria de forma es la capacidad de una aleación para recordar y recuperar su forma original después de ser deformada y luego volver a calentarse por encima de su temperatura de transformación. Este fenómeno se basa en la reversibilidad de la transformación martensítica y es fundamental para muchas aplicaciones prácticas de las AMF.

2.3.2 Aplicaciones

Sector médico y biomédico

En este sector las aleaciones con memoria de forma (AMF) ofrecen una amplia gama de aplicaciones debido a sus propiedades únicas y su capacidad para adaptarse al entorno biológico.

- **Stents vasculares:** Los stents vasculares fabricados con AMF se utilizan para tratar enfermedades cardíacas y vasculares, como la enfermedad arterial coronaria. Estos dispositivos se expanden en las arterias estrechas para mejorar el flujo sanguíneo y prevenir la obstrucción de los vasos sanguíneos.
- **Implantes ortopédicos:** Las AMF se utilizan en implantes ortopédicos, como clavos, tornillos y placas, que se utilizan para fijar fracturas óseas y estabilizar las articulaciones. Estos implantes pueden adaptarse a la forma del hueso y proporcionar una sujeción segura durante el proceso de curación.
- **Endoprótesis:** Las AMF se utilizan en endoprótesis, como las articulaciones artificiales de la cadera y la rodilla, para restaurar la movilidad y la función de las articulaciones dañadas. Estos dispositivos pueden adaptarse a los

movimientos naturales del cuerpo y mejorar la calidad de vida de los pacientes con enfermedades articulares degenerativas.

- **Instrumental quirúrgico:** Las AMF se utilizan en instrumentos quirúrgicos, como pinzas y forceps, que pueden cambiar de forma con la temperatura corporal para facilitar procedimientos quirúrgicos mínimamente invasivos y precisos.
- **Dispositivos de diagnóstico y monitoreo:** Las AMF se utilizan en dispositivos de diagnóstico y monitoreo, como sensores y microchips, que pueden adaptarse a la forma de los tejidos biológicos y proporcionar mediciones precisas de parámetros fisiológicos, como la presión arterial y la actividad eléctrica del corazón.

2.4 Ingeniería mecánica y automotriz

Algunas de las aplicaciones más destacadas en este sector son:

- **Actuadores y válvulas inteligentes:** Las AMF se utilizan en actuadores y válvulas inteligentes para controlar diversos sistemas en maquinaria y vehículos. Estos dispositivos pueden cambiar de forma en respuesta a estímulos externos, como cambios de temperatura o campos magnéticos, lo que permite un control preciso y una mayor eficiencia en los sistemas mecánicos.
- **Amortiguadores y sistemas de suspensión:** En el sector automotriz, las AMF se emplean en amortiguadores y sistemas de suspensión adaptativos que pueden ajustarse automáticamente para proporcionar una conducción más suave y cómoda. Estos sistemas pueden adaptarse a las condiciones de la carretera y mejorar la estabilidad y el confort del vehículo.
- **Componentes de motores:** Las AMF se utilizan en componentes de motores, como válvulas y pistones, para mejorar la eficiencia y el rendimiento de los

motores de combustión interna. Estos componentes pueden adaptarse a las condiciones de funcionamiento del motor y optimizar la combustión y la entrega de potencia.

- **Sistemas de seguridad activa:** Las AMF se utilizan en sistemas de seguridad activa, como sistemas de frenado y dirección asistida, para mejorar la seguridad y la maniobrabilidad de los vehículos. Estos sistemas pueden adaptarse dinámicamente a las condiciones del tráfico y ayudar a prevenir accidentes y colisiones.
- **Componentes de transmisión:** En la ingeniería mecánica, las AMF se emplean en componentes de transmisión, como engranajes y ejes, para mejorar la eficiencia y la durabilidad de los sistemas de transmisión. Estos componentes pueden adaptarse a las cargas y las condiciones de funcionamiento y minimizar la fricción y el desgaste.

Electrónica y microtecnología

- **Microactuadores:** Las AMF se utilizan en microactuadores para dispositivos microelectromecánicos (MEMS). Estos microactuadores pueden cambiar de forma en respuesta a estímulos externos, como cambios de temperatura o campos magnéticos, y se utilizan en una variedad de aplicaciones, como en la industria automotriz, aeroespacial y de dispositivos médicos.
- **Interruptores y conmutadores:** Las AMF se emplean en interruptores y conmutadores inteligentes que pueden cambiar de estado en respuesta a cambios de temperatura o estímulos magnéticos. Estos dispositivos se utilizan en aplicaciones donde se requiere una conmutación precisa y controlada, como en equipos de telecomunicaciones y sistemas de control.

- **Sensores y elementos de control:** Las AMF se utilizan en sensores y elementos de control que pueden adaptarse a la forma de los objetos o entornos a los que están expuestos. Estos sensores pueden utilizarse en una variedad de aplicaciones, como en la monitorización de la salud, la detección de presión y la medición de la temperatura.
- **Dispositivos de microfluídica:** Las AMF se utilizan en dispositivos de microfluídica para el control preciso de fluidos a escala microscópica. Estos dispositivos se utilizan en aplicaciones de laboratorio, diagnóstico médico y análisis químico, donde se requiere una manipulación precisa y controlada de líquidos y gases.

Aeroespacial y militar

En los sectores aeroespacial y militar, las aleaciones con memoria de forma (AMF) han encontrado diversas aplicaciones gracias a sus propiedades excepcionales y su capacidad para adaptarse a entornos exigentes.

- **Componentes de aeronaves:** Las AMF se emplean en la fabricación de componentes de aeronaves, como alas, superficies de control, y sistemas de tren de aterrizaje. Estos componentes pueden adaptarse dinámicamente a las condiciones de vuelo y ayudar a mejorar la eficiencia aerodinámica y la estabilidad de la aeronave.
- **Sistemas de control activo:** En la industria aeroespacial, las AMF se utilizan en sistemas de control activo, como aletas y deflectores, que pueden ajustar automáticamente la actitud y la trayectoria de la aeronave en respuesta a cambios en las condiciones de vuelo y las demandas operativas.
- **Componentes de satélites y sondas espaciales:** Las AMF se utilizan en la

fabricación de componentes de satélites y sondas espaciales, como antenas desplegadas y paneles solares, que pueden plegarse durante el lanzamiento y desplegarse en el espacio mediante la activación de la memoria de forma.

- **Estructuras de paracaídas y airbags:** En aplicaciones militares y aeroespaciales, las AMF se utilizan en la fabricación de estructuras de paracaídas y airbags que pueden plegarse compactamente durante el almacenamiento y desplegarse de manera controlada y precisa durante el aterrizaje.

Redes neuronales

Las redes neuronales son un modelo computacional inspirado en el funcionamiento del sistema nervioso humano. Estas estructuras de aprendizaje automático han ganado popularidad debido a su capacidad para aprender y realizar tareas complejas, que van desde el reconocimiento de patrones hasta la toma de decisiones.

En su forma más básica, una red neuronal está compuesta por unidades llamadas neuronas, organizadas en capas interconectadas. Cada neurona recibe entradas, las procesa mediante una función de activación y produce una salida que se transmite a las neuronas de la capa siguiente.

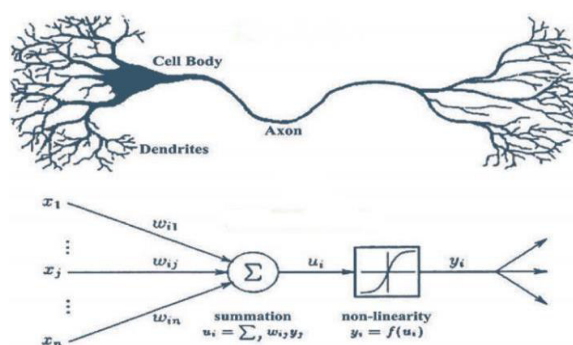
La capacidad de adaptarse y aprender de los datos es una de las características más poderosas de las redes neuronales. Durante el entrenamiento, la red ajusta los pesos de las conexiones entre las neuronas para minimizar la diferencia entre las salidas predichas y las salidas reales. Este proceso se realiza utilizando algoritmos de optimización, como el descenso de gradiente, que

ajustan gradualmente los parámetros de la red para mejorar su rendimiento.

Las redes neuronales se han aplicado con éxito en una amplia gama de áreas, incluido el procesamiento del lenguaje natural, la visión por computadora, la robótica y la medicina. Su capacidad para modelar relaciones complejas en los datos las hace especialmente adecuadas para tareas que no pueden ser abordadas fácilmente con enfoques tradicionales de programación.

Figura 13

Red neuronal



Nota: Reimpreso de "Marco de evaluación para modelos predictivos de series de tiempo heterogéneas usando inteligencia artificial", por Johanna Michelle Cantuña Guano, 2024, Escuela Politécnica Nacional.

Fuente:

<https://bibdigital.epn.edu.ec/bitstream/15000/25625/1/CD%2014332.pdf>

2.4.1 Arquitectura de redes neuronales

La arquitectura de una red neuronal se refiere a su estructura y organización, que determina cómo se procesan y se transmiten las señales a través de la red. En general, las redes neuronales están compuestas por múltiples capas de neuronas interconectadas, cada una con una función específica en el procesamiento de la información.

Neuronas y capas

Las redes neuronales, como sistemas de aprendizaje automático, están compuestas

por neuronas y capas que imitan la estructura y el funcionamiento del cerebro humano. Las neuronas son las unidades básicas de procesamiento que reciben entradas, realizan cálculos ponderados y producen salidas utilizando funciones de activación. Estas neuronas están organizadas en capas, que incluyen la capa de entrada, donde se introducen los datos, las capas ocultas, que realizan transformaciones complejas de los datos, y la capa de salida, que genera las salidas finales de la red. Cada neurona en una capa está conectada a todas las neuronas en la capa siguiente a través de conexiones ponderadas, y durante el entrenamiento de la red, los pesos de estas conexiones se ajustan para optimizar el rendimiento de la red en base a un conjunto de datos de entrenamiento.

Conexiones y pesos

Las conexiones entre las neuronas en una red neuronal están ponderadas por pesos que determinan la influencia relativa de cada entrada en la salida de la neurona de destino. Durante el entrenamiento de la red, estos pesos se ajustan mediante algoritmos de optimización, como el descenso de gradiente, con el objetivo de minimizar la diferencia entre las salidas predichas y las salidas reales. Los pesos de las conexiones representan la capacidad de la red para aprender y adaptarse a los patrones en los datos de entrada, y su ajuste es fundamental para mejorar el rendimiento y la precisión de la red en diversas tareas de aprendizaje automático.

Funciones de activación

Cada neurona en una red neuronal aplica una función de activación a la suma ponderada de sus entradas para producir su salida. Estas funciones de activación introducen no linealidades en la red, permitiendo que las redes neuronales capturen y modelen relaciones complejas en los datos. Ejemplos comunes de funciones de

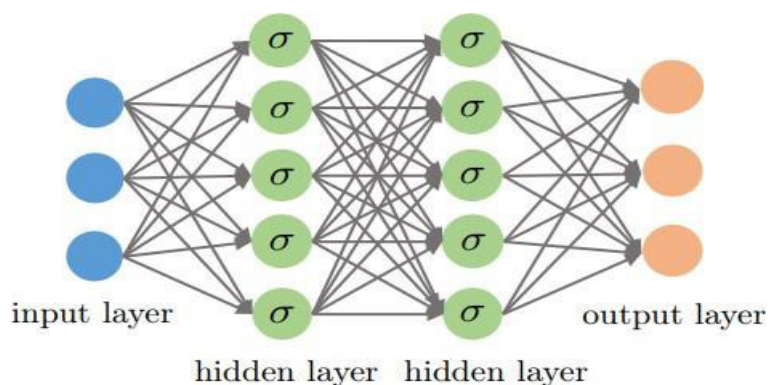
activación incluyen la función sigmoide, la función de tangente hiperbólica y la función de activación lineal rectificada (ReLU). La elección de la función de activación puede afectar significativamente el rendimiento y la capacidad de aprendizaje de la red, y es un aspecto importante a considerar durante el diseño y entrenamiento de la red neuronal.

Tipos de arquitecturas

Existen varios tipos de arquitecturas de redes neuronales, cada una con sus propias características y aplicaciones específicas. Las redes neuronales feedforward son las más simples, con información que fluye en una dirección desde la capa de entrada hasta la capa de salida. Las redes neuronales recurrentes tienen conexiones retroalimentadas que les permiten procesar secuencias de datos y modelar dependencias temporales. Las redes neuronales convolucionales son ideales para el procesamiento de datos estructurados, como imágenes, utilizando filtros para extraer características relevantes. La elección de la arquitectura depende de la naturaleza del problema y los datos disponibles, y es un factor crucial en el diseño efectivo de una red neuronal para una tarea específica.

Figura 14

Arquitectura de red neuronal



Nota: Reimpreso de semanti, por Jianging Fan, 2019,

Fuente: <https://www.semanticscholar.org/paper/A-Selective-Overview-of-Deep-Learning->

Fan-Ma/7b14b9ee3685d2b68c10e6768faf69563b4c3028

2.4.2 Entrenamiento y optimización

El proceso de entrenamiento y optimización de redes neuronales es fundamental para mejorar su rendimiento y capacidad de generalización en la resolución de problemas complejos de aprendizaje automático. Durante el entrenamiento, la red neuronal ajusta sus parámetros internos, como los pesos de las conexiones entre neuronas, para minimizar la discrepancia entre las salidas predichas y las salidas reales sobre un conjunto de datos de entrenamiento. (Cheng, 2020)

El algoritmo de optimización más comúnmente utilizado en el entrenamiento de redes neuronales es el descenso de gradiente, que ajusta los pesos de las conexiones en dirección a la pendiente negativa de una función de pérdida o error. Este proceso se repite iterativamente a lo largo de múltiples épocas hasta que la red converja hacia una solución óptima o alcanza un criterio de convergencia predefinido.

Durante el entrenamiento, es crucial considerar estrategias para evitar el sobreajuste, fenómeno en el que la red memoriza el conjunto de entrenamiento, pero falla en generalizar a nuevos datos. Para mitigar el sobreajuste, se emplean técnicas como la regularización, que penaliza los pesos grandes, y la validación cruzada, que evalúa el rendimiento de la red en datos no vistos durante el entrenamiento.

Otro aspecto importante del entrenamiento de redes neuronales es la elección de la función de pérdida, que cuantifica la discrepancia entre las salidas predichas y las salidas reales. La función de pérdida más adecuada depende de la naturaleza del problema y de las características de los datos. Ejemplos comunes de funciones de pérdida incluyen la entropía cruzada para problemas de clasificación y el error

cuadrático medio para problemas de regresión. (Curcio, 2025)

La optimización de hiperparámetros también juega un papel crucial en el proceso de entrenamiento, donde los hiperparámetros, como la tasa de aprendizaje y el tamaño del lote, influyen en la velocidad y estabilidad del proceso de convergencia de la red. La selección adecuada de hiperparámetros se realiza típicamente mediante técnicas de búsqueda en cuadrícula o búsqueda aleatoria, que exploran el espacio de hiperparámetros en busca de la combinación óptima que maximice el rendimiento de la red en un conjunto de validación.

2.4.3 Aplicaciones prácticas

Las redes neuronales tienen una amplia gama de aplicaciones prácticas en diversos campos, aprovechando su capacidad para aprender y modelar patrones complejos en los datos. Su versatilidad las hace ideales para resolver una variedad de problemas en áreas como la medicina, la industria y la robótica.

Medicina

Las redes neuronales se emplean en medicina para diagnósticos precisos de enfermedades, análisis avanzado de imágenes médicas como resonancias magnéticas y tomografías computarizadas, pronósticos de enfermedades basados en datos clínicos, y en el desarrollo de fármacos a través de la simulación de interacciones moleculares.

Tecnología

En el ámbito tecnológico, las redes neuronales se utilizan en el reconocimiento de voz para asistentes virtuales como Siri o Alexa, en el procesamiento del lenguaje natural para la traducción automática y análisis de sentimientos, en la visión por computadora para la detección de objetos y reconocimiento facial, y en la creación de sistemas de recomendación personalizados. (Kennedy, 2024)

Industria

En la industria, las redes neuronales son vitales para el control de procesos industriales automatizados, el mantenimiento predictivo de maquinaria y equipos, la optimización de la cadena de suministro mediante la previsión de la demanda y la gestión eficiente de inventarios, y la detección de fraudes en transacciones financieras y comerciales.

Finanzas

En el sector financiero, las redes neuronales se aplican en la detección de fraudes bancarios y de tarjetas de crédito, en el análisis de riesgos para la concesión de préstamos y seguros, en la predicción de tendencias y fluctuaciones del mercado de valores, y en la gestión de carteras de inversión para maximizar rendimientos y minimizar riesgos.

Robótica

En el campo de la robótica, las redes neuronales tienen aplicaciones innovadoras, especialmente en exoesqueletos robóticos. Estos dispositivos, diseñados para asistir o mejorar la movilidad humana, pueden aprovechar las capacidades de aprendizaje y adaptación de las redes neuronales para ajustarse dinámicamente a las necesidades y movimientos del usuario. Las redes neuronales en los exoesqueletos robóticos permiten una interacción más natural y eficiente entre el usuario y el dispositivo, facilitando la rehabilitación física y mejorando la calidad de vida de personas con discapacidades motoras o lesiones en las extremidades. (Goldiez, 2025)

2.5 Algoritmos genéticos

Los algoritmos genéticos son métodos que pueden utilizarse para resolver problemas de optimización y búsqueda, fueron planteados por John Holland en 1975 y

llevan este nombre porque están basados en el proceso genéticos de los organismos vivos. Es decir, es un proceso sistemático que aplica los métodos de la evolución biológica: selección basada en la población, mutación y reproducción sexual.

El algoritmo genético es una metodología de optimización que emula el proceso evolutivo biológico para resolver problemas complejos. Se compone esencialmente de dos fases interrelacionadas. La primera fase, la cual podría considerarse como la etapa de generación y evolución, se encarga de crear y manipular poblaciones de posibles soluciones. En esta fase, se aplican operadores genéticos como el crossover (cruce) y la mutación para generar nueva diversidad genética. Además, se emplea un proceso de selección natural donde los individuos con mejores características tienen una mayor probabilidad de ser seleccionados para la reproducción y formar la próxima generación. (Copaci, 2021)

La segunda fase consiste en evaluar la aptitud de cada individuo dentro de la población. Esta evaluación se realiza mediante una función de aptitud, también conocida como función de fitness, que asigna un valor numérico a cada individuo en función de su capacidad para resolver el problema en cuestión. Los individuos con un mejor rendimiento, es decir, aquellos que están más adaptados al entorno, tendrán un valor de fitness más alto. (Goldiez, 2025)

Al ser un algoritmo, este debe tener un problema de optimización es cual se desea resolver, las soluciones a este problema se representan como individuos dentro de una población. Un problema de optimización radica en cumplir ciertas restricciones de tal forma que se obtenga el mejor resultado posible, este se mide a través de una función objetivo.

La solución óptima, también llamada óptimo global del problema, es aquella que

resuelve el problema planteado de la mejor forma. Otro concepto a tener en cuenta es el de óptimo local, este no es necesariamente la solución del problema, pero si lo es dentro de una vecindad de espacio de búsqueda. Una vecindad es un conjunto de soluciones del problema, donde estas se encuentran cercanas entre sí, si dos soluciones son de la misma vecindad entonces se dice que son vecinos. (Brose, 2020)

2.5.1 Funcionamiento del algoritmo genético

El funcionamiento de un algoritmo genético inicia con la creación de una población inicial, la cual se genera de manera aleatoria y dispersa en el espacio de búsqueda del problema. Posteriormente, cada individuo en la población es evaluado mediante una función de fitness para determinar su nivel de adaptación al entorno. Este proceso de evaluación permite identificar qué tan bien están adaptados los individuos y guía el proceso de selección de los más aptos para la reproducción.

La reproducción en un algoritmo genético se lleva a cabo a través de dos procesos principales: el crossover y la mutación. El crossover implica la combinación de información genética de dos o más individuos para generar descendencia que formará la próxima generación. Por otro lado, la mutación introduce pequeñas modificaciones aleatorias en los individuos, lo que fomenta la diversidad genética y la exploración del espacio de búsqueda.

Tras la reproducción, se crea una nueva generación de individuos que se somete nuevamente al proceso de evaluación de la función de fitness y reproducción. Este ciclo de evaluación, reproducción y generación de nuevas poblaciones se repite de manera iterativa hasta que se alcanza un criterio de terminación predefinido, como un número máximo de generaciones o la convergencia de la solución.

En el contexto de un algoritmo genético, se utilizan varios conceptos fundamentales. El cromosoma representa un conjunto de genes que codifican las variables de decisión del problema, mientras que el individuo es una solución potencial del mismo, representada por un cromosoma. La población consiste en un conjunto de individuos, y la información genética se representa típicamente mediante una lista o arreglo de bits, lo que proporciona una representación uniforme y adaptable a diferentes

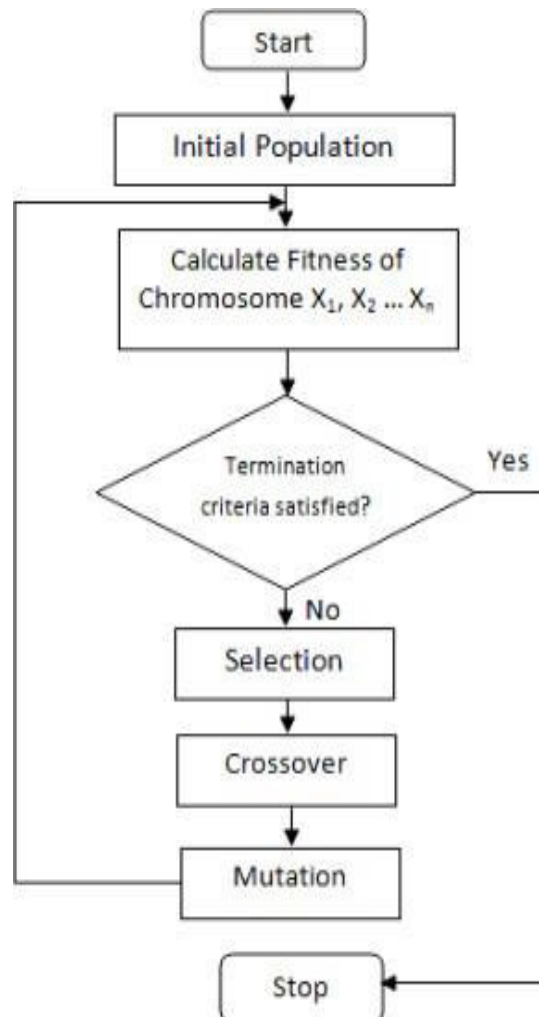
tipos y complejidades de problemas.

La mutación y el crossover son operadores genéticos clave en el proceso de reproducción. La mutación introduce variabilidad genética al alterar aleatoriamente los cromosomas, mientras que el crossover combina la información genética de los padres para generar descendencia con características heredadas de ambos progenitores. (Auricchio, 2021)

La función de fitness, también conocida como función objetivo, desempeña un papel crucial en la evaluación de la calidad de las soluciones. Esta función determina la idoneidad de un individuo y guía la selección de los individuos más aptos para la reproducción y la eliminación de los menos adecuados. Una función de fitness efectiva debe reflejar con precisión el valor de la solución y permitir una evaluación justa y precisa del rendimiento de los individuos en el contexto del problema de optimización. (Dutta et al., 2020)

Figura 15

Funcionamiento del algoritmo genético



Nota: Reimpreso de "Improving Spam Detection Using Neural Networks Trained by Memetic Algorithm", por Shaveen Singh, Anish Chand, y Sunil Pranit Lal, 2013, ResearchGate,

Fuente: https://www.researchgate.net/publication/261468690_Improving_Spam_Detection_Using_Neural_Networks_Trained_by_Memetic_Algorithm

2.5.2 Aplicaciones en ingeniería y ciencias

Los algoritmos genéticos encuentran una amplia variedad de aplicaciones en los campos de la ingeniería y las ciencias, donde se utilizan para resolver problemas complejos de optimización, diseño y modelado.

Diseño de ingeniería

- Optimización de diseños de ingeniería para mejorar la eficiencia, la resistencia estructural, la aerodinámica y otras características de productos y sistemas.
- Diseño de componentes mecánicos, como turbinas, motores, estructuras, y sistemas de engranajes, teniendo en cuenta múltiples restricciones y objetivos.
- **Diseño de circuitos electrónicos**
- Optimización de circuitos electrónicos para minimizar el consumo de energía, mejorar la velocidad de procesamiento y reducir la huella de memoria.
- Diseño de circuitos integrados y microchips para aplicaciones específicas, como la electrónica de consumo, la comunicación y la informática.

Planificación de recursos y logística

- Optimización de la distribución de recursos, como la asignación de rutas de transporte, la programación de horarios y la gestión de inventarios.
- Planificación de la cadena de suministro para minimizar costos, reducir tiempos de entrega y mejorar la eficiencia operativa.
- **Ingeniería estructural**
- Optimización del diseño de estructuras y edificios para resistir cargas estáticas y dinámicas, minimizar el peso y maximizar la estabilidad.
- Diseño de materiales compuestos y estructuras avanzadas para aplicaciones aeroespaciales, automotrices y de construcción.

Procesamiento de imágenes y señales

- Mejora de la calidad de imágenes médicas, satelitales y de reconocimiento facial mediante técnicas de procesamiento de imágenes y visión por computadora.
- Optimización de algoritmos de procesamiento de señales para aplicaciones de telecomunicaciones, audio y procesamiento de vídeo.

Modelado y simulación

- Ajuste de parámetros de modelos matemáticos y simulaciones numéricas para ajustar datos experimentales y predecir el comportamiento de sistemas complejos.
- Optimización de modelos de sistemas dinámicos y control de procesos industriales para mejorar la eficiencia y la estabilidad del sistema.

Control y actuación de exoesqueleto

- Desarrollo de algoritmos de control adaptativo basados en algoritmos genéticos para ajustar dinámicamente los parámetros de control del exoesqueleto en función de las condiciones del usuario y del entorno.
- Optimización de la cinemática y la dinámica del exoesqueleto mediante algoritmos genéticos para garantizar un movimiento suave, natural y seguro.

Biomecánica del codo

Estudio de los movimientos y fuerzas que actúan sobre la articulación del codo durante diversas actividades. Este campo analiza cómo los componentes anatómicos del codo interactúan para producir movimientos fluidos y eficientes.

Músculos flexores del codo

Grupo de músculos responsables de la flexión del codo, permitiendo doblar el brazo.

Los principales músculos incluyen el bíceps braquial, el braquial y el braquiorradial.

Músculos extensores del codo

Grupo de músculos responsables de la extensión del codo, permitiendo enderezar el brazo. Los principales músculos son el tríceps braquial y el ancóneo.

Tendinitis del codo

Inflamación de los tendones alrededor del codo debido a sobreuso o movimientos repetitivos, causando dolor y sensibilidad en la articulación.

Bursitis del codo

La inflamación de la bursa, una estructura llena de líquido que actúa como amortiguador entre los huesos, tendones y músculos del codo, puede llevar a hinchazón, dolor y rigidez en esa área.

Síndrome del túnel cubital

El pinzamiento del nervio cubital en la región del codo puede provocar síntomas como adormecimiento, hormigueo y una disminución de la fuerza en la mano y los dedos. Aunque se parece al síndrome del túnel carpiano, esta afección se diferencia porque compromete el nervio cubital.

Lesiones por sobrecarga

Daños acumulativos en los tejidos del codo debido a movimientos repetitivos o fuerzas excesivas. Estas lesiones incluyen tendinitis, bursitis y desgarros musculares.

Movilidad articular

La capacidad de la articulación del codo para moverse a lo largo de todo su rango de movimiento es crucial. Una movilidad articular adecuada es fundamental para llevar a cabo actividades cotidianas y deportivas sin limitaciones.

Fuerza muscular

Capacidad de los músculos del codo para generar fuerza. La fuerza muscular adecuada es crucial para realizar tareas que requieren levantar, empujar o tirar.

Propriocepción del codo

Capacidad del cuerpo para percibir la posición y el movimiento del codo en el espacio. La propiocepción adecuada es importante para la coordinación y el control del movimiento del brazo.

III. MÉTODO

3.1 Tipo de investigación

Según la naturaleza de este proyecto se considera que este tiene un enfoque aplicativo, debido a que esta investigación se centra en encontrar una solución a un problema en concreto para que sea aplicada al mundo cotidiano, con el fin de mejorar la calidad de vida de las personas. Además, al estar generando nuevo conocimiento y poniéndolo en práctica esta investigación se clasificaría específicamente como una investigación aplicada tecnológica.

3.2 Población y muestra

3.2.1 Población

Se entiende por población un conjunto de personas u objetos, que puede ser finito o infinito, que presentan características en común. En este trabajo de investigación la población objetivo son las personas que por algún motivo requieran la utilización de un exoesqueleto de codo, esto con el fin de mejorar su calidad de vida y lograr su rehabilitación.

3.2.2 Muestra

Se entiende por muestra una representación significativa de las características de una población, por ende, en esta investigación la muestra serán las personas con características promedio en cuanto a peso, edad, estatura, etc.

3.3 Operacionalización de variables

Tabla 1

Cuadro de operacionalización de variables

Variables	Dimensiones	Indicadores	Método de evaluación	Instrumento
Variable independiente: Diseño e implementación de un exoesqueleto de codo	- Implementación del exoesqueleto. - Modelamiento del actuador - Control del sistema	- Materiales Estructural del sistema - Dimensiones del sistema	- Simulación es FEM estructurales. - Simulaciones de sistemas de control.	- Software de diseño y simulación - Materiales de impresión 3D - Sensores y controladores electrónicos
Variable dependiente: Optimizar la rehabilitación de pacientes	- Rapidez de recuperación	-Error absoluto promedio de la trayectoria	Índices de desempeño. - Simulaciones de trayectoria.	

Fuente: Elaboración propia

3.4 Instrumentos

Para la presente investigación se utilizaron los siguientes instrumentos:

- Para la implementación del exoesqueleto

Software CAD: diseño del exoesqueleto en SolidWorks

Impresora 3D: con la finalidad de implementar la estructura del exoesqueleto.

Aleación con memoria de forma: Será el actuador en el exoesqueleto.

- Para el modelamiento

Software de simulación FEM: Modelamiento por elementos finitos de la aleación con memoria de forma, modelamiento estructural y de deformaciones del exoesqueleto.

- Para el control

Computadora: en este dispositivo se programará la técnica de control empleada. Software Matlab: el controlador será programada en el software MATLAB y Simulink. Sensores: tienen el objetivo de convertir la información física del sistema en datos que posteriormente se utilizarán para controlar el sistema.

Controlador: tiene la función de realizar el control para un óptimo desempeño del exoesqueleto.

3.5 Procedimientos

Se realizará el diseño del exoesqueleto robótico de codo en el software SolidWorks, posteriormente se realizará la impresión 3D del mecanismo y la implementación haciendo uso de alambre de aleación de memoria de forma, que será el actuador.

Luego se realizará el modelamiento del actuador, este paso modelo se utilizará en

el diseño del controlador, el cual tomará como entrada las señales de los sensores y las utilizará para realizar el control de trayectoria. La robustez del control permitirá controlar las no linealidades del sistema, causadas por la histéresis del actuador.

3.6 Análisis de datos

Luego de haber concluido la fase de implementación y haber programado el controlador, se realiza la recopilación de datos con el objetivo de presentar los resultados en tablas y gráficos permitiendo un fácil entendimiento. Además, permitirá saber si se consiguió el objetivo de optimizar la rehabilitación de pacientes haciendo uso de las técnicas propuestas en este trabajo.

3.7 Consideraciones éticas

La investigación, al centrarse en el diseño de un exoesqueleto robótico para la rehabilitación del codo en pacientes con movilidad reducida por accidentes cerebrovasculares, busca no solo ofrecer una solución innovadora en el campo de la medicina rehabilitadora sino también abordar las implicaciones éticas asociadas. Se enfatiza la importancia de garantizar la seguridad del dispositivo a través de pruebas exhaustivas que minimicen cualquier riesgo potencial. La protección de la privacidad de los datos recolectados durante el uso del exoesqueleto es prioritaria, asegurando que la información del paciente se maneje con el máximo cuidado y confidencialidad. Asimismo, el acceso equitativo a esta tecnología es fundamental, de modo que se promueva la igualdad en el acceso a tratamientos avanzados, sin importar la situación económica o geográfica de los usuarios. Este enfoque ético es vital para fomentar la confianza y aceptación de la tecnología por parte de la comunidad médica y los pacientes. (Annadata, 2024)

IV. RESULTADOS

4.1 Control automático

4.1.1 Control PD: Adaptación de gravedad

Para gestionar el exoesqueleto, se plantearon varias arquitecturas, comenzando con los tradicionales controladores PD y PID.

$$\tau = kpq + kv \dot{q}$$

$$\tau = kpq + kv \dot{q} + ki \int_0^{\tau} q(\tau) d(\tau)$$

Estos controles dependen únicamente del error q ,

definido como: $q = q_d - q$

donde q_d es la posición angular deseada y q es la posición angular real de la articulación. Las matrices kp , kv y ki son definidas positivamente y corresponden a la ganancia proporcional, derivativa e integral, respectivamente.

Además, el control PD que ajusta por gravedad no se basa únicamente en el error, sino que también considera el vector de torques gravitacionales $g(q)$. Esto significa que se requiere una comprensión previa del modelo del sistema para asegurar el funcionamiento adecuado del control.

$$\tau = kpq + kv \dot{q} + g(q)$$

Es imposible determinar con precisión la masa del brazo de cada individuo, así como la posición del centro de masas o la inercia. Por consiguiente, se sugiere implementar un controlador adaptativo capaz de estimar estos parámetros desconocidos para optimizar el rendimiento del control.

Se sugiere utilizar la siguiente parametrización: para cualquier u, v, w en $\mathbb{R}^n$. El modelo dinámico de un robot con n grados de libertad (DOF) puede ser representado de la siguiente manera:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \Phi(q, u, v, w) \cdot \Theta + M_0(q) \cdot u + C_0(q, w) \cdot v + g_0(q) \cdot w$$

donde $\Phi(q, u, v, w)$ está en $\mathbb{R}^{n \times m}$; $M_0(q)$, $C_0(q, w)$ están en $\mathbb{R}^n$; $g_0(q)$ está en $\mathbb{R}^n$ y Θ residen en $\mathbb{R}^m$. El vector Θ , denominado vector de parámetros dinámicos, incorpora elementos relacionados con parámetros físicos, como masas e inercias del motor y la carga. Las matrices $M_0(q)$, $C_0(q, w)$ y el vector $g_0(q)$ constituyen partes de las matrices $M(q)$, $C(q, \dot{q})$ y el vector $g(q)$ que no son dependientes de los parámetros dinámicos Θ .

En este contexto, se mantiene la siguiente expresión:

$$g(q, \Theta) = \Phi(q, 0, 0, 0) \cdot \Theta + g_0(q)$$

donde establecemos:

$$u = 0$$

$$v = 0$$

$$w = 0$$

Dado un vector Θ en $\mathbb{R}^n$, la expresión se puede reescribir como:

$$g(q, \Theta) = \Phi(q) \cdot \Theta + g_0(q)$$

Considerando el control PD más Adaptación de gravedad, se presenta como:

$$\tau = k_p \cdot q + k_v \cdot \dot{q} + g(q, \Theta)$$

$$\tau = k_p \cdot q + k_v \cdot \dot{q} + \Phi(q) \cdot \Theta + g_0(q)$$

$$\Theta(t) = \Phi(q) \cdot \int^t (\varepsilon_0/1 + |\dot{q} - q|) ds + \Theta(0)$$

donde:

- k_p , k_v en $\mathbb{R}^n$ y Θ en $\mathbb{R}^{m \times m}$ son matrices diseñadas diagonalmente definidas positivas.

- ε_0 es una constante positiva.

La ecuación se deriva mediante la parametrización anterior. La segunda ecuación se origina de la función candidata de Lyapunov, que también incluye una revisión de estabilidad para este controlador adaptativo. Se expone el estudio de estabilidad para el escenario de seguimiento cuando q_d varía

Actualmente, se identifican como parámetros clave la masa m_3 , la distancia al centro de masa l_{3c} y la inercia I_3 . Así, el vector de parámetros dinámicos es:

$$\Theta = [m_3, m_3 \cdot l_{3c}, m_3 \cdot l_{3c}^2 + I_3]$$

Identificando estos parámetros, se puede calcular

la matriz Φ . $\Phi(q) =$

$$\begin{bmatrix} g \cdot (l_2 \cdot \cos(q_1 + q_2) + l_1 \cdot \cos(q_1)) & , & g \cdot \sin(q_1 + q_2) \\ \cdot \cos(q_3), & 0 & [g \cdot l_2 \cdot \cos(q_1 + q_2), & g \cdot \\ \sin(q_1 + q_2) \cdot \cos(q_3), & 0] \\ [0, g \cdot \cos(q_1 + q_2) \cdot \sin(q_3), 0] \end{bmatrix}$$

Y la matriz $g_0(q)$ es:

$g_0(q) =$

$$\begin{bmatrix} [g \cdot m_2 \cdot (l_2 \cdot \cos(q_1 + q_2) + l_1 \cdot \cos(q_1)) + g \cdot l_{1c} \cdot \\ m_1 \cdot \cos(q_1)] & [g \cdot l_{2c} \cdot m_2 \cdot \cos(q_1 + q_2)] & [0] \end{bmatrix}$$

4.1.2 Control por modos deslizantes adaptable

Una estrategia para solucionar el problema es dirigir los modos del sistema hacia las trayectorias requeridas, las cuales son predefinidas. Se emplea el método de modos deslizantes para lograrlo. Los errores de seguimiento se describen mediante:

$$e_q = q - q_d$$

$$\dot{e}_q = \dot{q} - \dot{q}_d$$

$$\ddot{e}_q = \ddot{q} - \ddot{q}_d$$

con q_d como la posición deseada.

Dentro del control por modos deslizantes, se establece una superficie que varía con el tiempo $s(t)$, en el espacio de estados R^n donde ocurre el movimiento deslizante. Esta superficie se describe mediante la ecuación escalar $s(q,t) = 0$, donde:

$$s(q, \dot{q}) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \cdot e_q$$

y $\lambda > 0$. A partir de aquí, para $n = 2$, se tiene:

$$s = \dot{e}_q + \lambda \cdot e_q$$

El principal objetivo en este enfoque es mantener $s(q,t)$ en cero cuando el seguimiento se encuentra fuera de $s(q,t)$. Una forma sencilla de lograr el estado de deslizamiento bajo incertidumbre en los parámetros dinámicos, es mediante la ley de control por conmutación:

$$\tau_{\text{sign}} = -\rho \cdot \text{sign}(s) \quad \text{sign}(s) =$$

$$\begin{cases} 1, & \text{si } s > 0 \\ -1 & \text{si } s < 0 \\ 0, & \text{si } s = 0 \end{cases}$$

$$\begin{cases} 1, & \text{si } s > 0 \\ -1 & \text{si } s < 0 \\ 0, & \text{si } s = 0 \end{cases}$$

$$\begin{cases} 1, & \text{si } s > 0 \\ -1 & \text{si } s < 0 \\ 0, & \text{si } s = 0 \end{cases}$$

con $\rho > 0$. Debido a que la ley de control τ_{sign} es discontinua, el control por conmutación causará castaño. Para lograr un control más suave, se emplea la función tangente hiperbólica $\tanh(s/\phi)$, configurando la ley de control (control mediante tangente

hiperbólica $\tau(\tanh)$ de la siguiente forma:

$$\tau_{\tanh} = -\rho \cdot \tanh(s/\varphi)$$

donde $\varphi > 0$ es la

frontera.

Ahora, la ley de control para un robot manipulador con múltiples grados de libertad, puede ser definida como:

$$\tau = \tau_{eq} + \tau_{\tanh}$$

donde τ_{eq} , nombrado como control equivalente, es la continuidad del control que preserva.

$$s = 0$$

En el caso de que la dinámica del robot sea completamente comprendida, el control equivalente se puede desarrollar utilizando su forma ajustada de la siguiente manera:

$$\tau_{eq} = C(q, \dot{q}) \cdot \dot{q} + g(q) + M \cdot \ddot{q}$$

Por lo tanto, la ley de control sería:

$$\tau = (C(q, \dot{q}) \cdot \dot{q} + g(q)) + M \cdot \ddot{q} - \rho \cdot \tanh(s/\varphi)$$

En este caso, el objetivo de la adaptación es que la ganancia $\rho(t)$ tienda a una función $f(q, \dot{q})/\alpha$ para reducir la amplitud del castaño. Para esto, se define la función $[z]^+$ como:

$$[z]^+ = \begin{cases} z, & \text{si } z \geq 0 \\ 0, & \text{si } z < 0 \end{cases}$$

donde z es el argumento de la función. El algoritmo para aminorar la amplitud del castaño es:

$$\rho(t) = \rho(t) \cdot \text{sign}(\delta(t)) - A [\rho(t) - \rho^+] + A \cdot \mu - \rho(t) [\delta(t)]^+$$

donde:

$$\delta(t) = \text{sign}(s(t) - \alpha, \alpha, \alpha) \text{ con } \alpha \in (0,$$

$$1) A > \rho^+$$

La ganancia ρ puede oscilar entre $[\mu, \rho^+]$, siendo $\mu > 0$ el valor mínimo elegido para ρ y $\rho^+ > 0$ el valor máximo permitido.

4.1.3 Evaluación numérica de controladores para codo

Se efectuaron simulaciones utilizando el modelo dinámico del exoesqueleto de hombro para los controladores PD, PID, PG+Adapt y modos deslizantes adaptativo (Slide), tal como se describió previamente. Se realizó una prueba de regulación con el objetivo de comparar el rendimiento de cada controlador. En la tabla se muestran las referencias que deben cumplir cada una de las articulaciones.

Tabla 2

Tabla de referencia

Articulación	Posición inicial	Posición deseada
q_1	0	$\pi/36$
q_2	0	$\pi/3$
q_3	0	$\pi/2$

Fuente: Elaboración propia

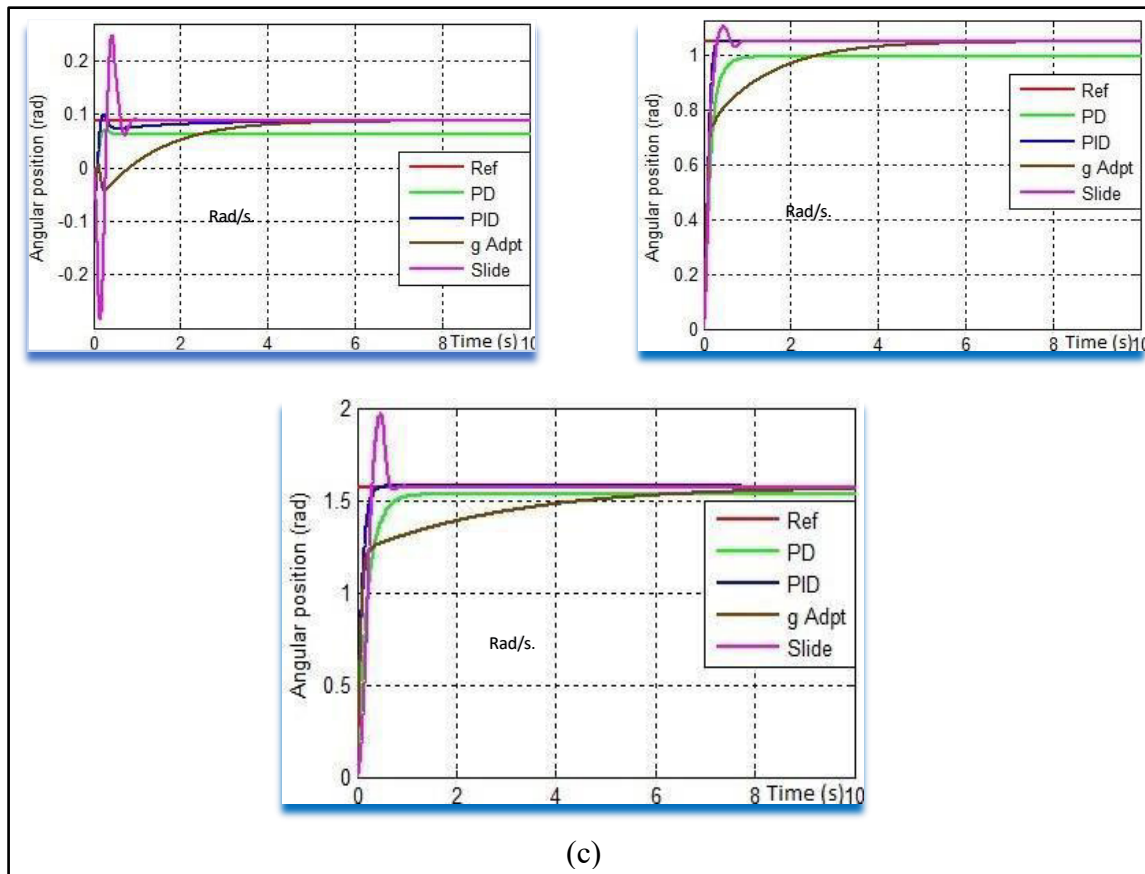
Después, se utilizaron funciones senoides con el fin de modelar ejercicios tradicionales de rehabilitación del hombro. La tabla muestra las trayectorias previstas para cada articulación.

Tabla 3*Tabla track*

Trayectoria	Frecuencia	Amplitud
q_1	$\pi/4$	$\pi/20$
q_2	$\pi/4$	$\pi/6$
q_3	$\pi/4$	$\pi/4$

Fuente: Elaboración propia

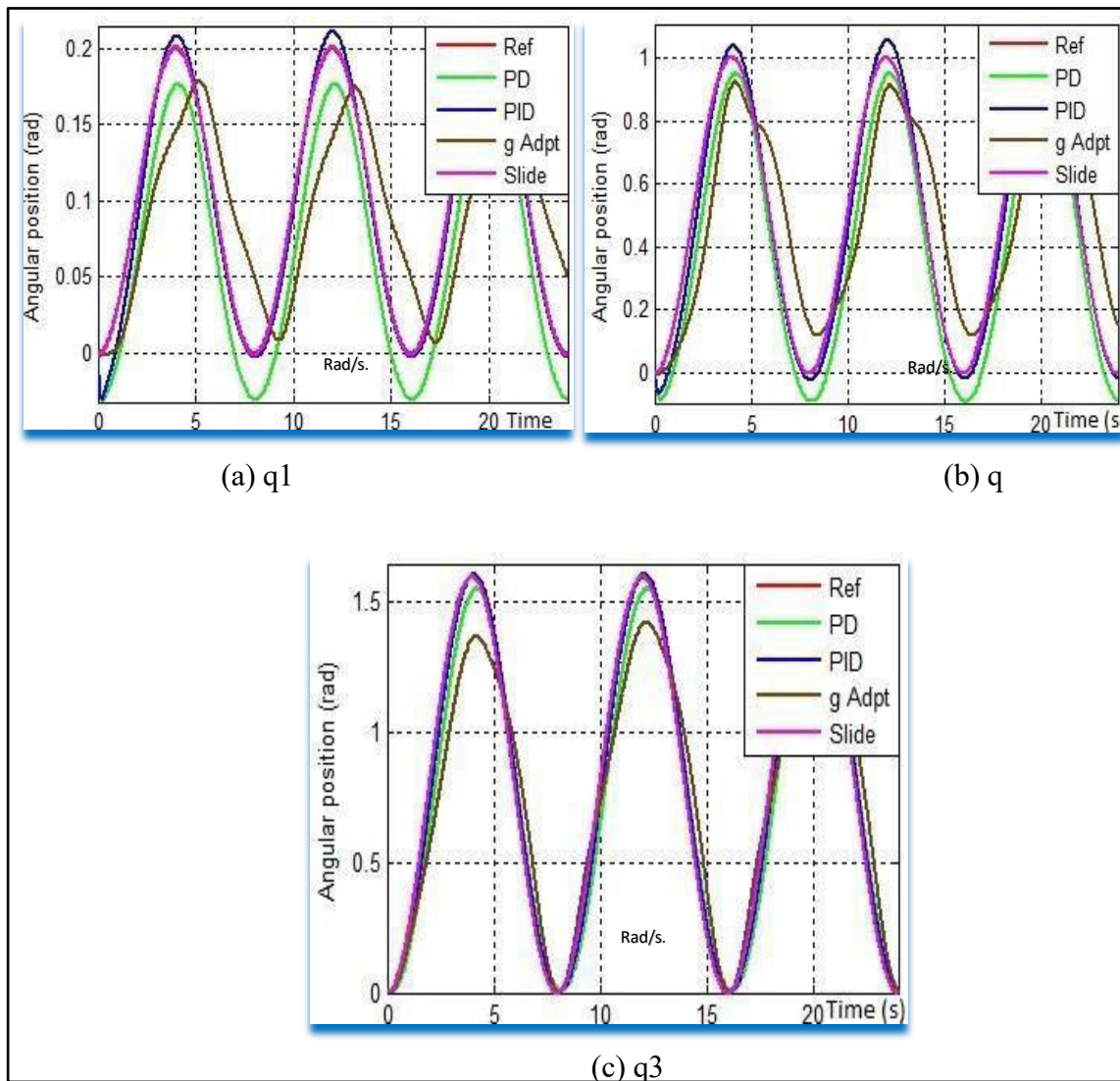
La imagen muestra que todos los controladores facilitan la convergencia de la posición hacia un punto de equilibrio estable, aunque lo hacen de distintas maneras. El controlador "PD" exhibe un desperfecto en modo estacionario de cerca de 0.2 radianes en la articulación q_3 . El controlador "g Adapt" ofrece una trayectoria ligera que alcanza la referencia de forma lenta, lo cual es ideal para ejercicios de rehabilitación. Para las posiciones angulares q_2 y q_3 , aunque no para q_1 , el controlador "PID" logra alcanzar la referencia más rápidamente y presenta un sobreimpulso mínimo. En términos generales, el controlador "Slide" es tanto rápido como preciso, aunque se nota un sobreimpulso considerable.

Figura 16*Resultados de simulación regulada**Nota: Elaboración propia utilizando MATLAB para la simulación y visualización de datos**Fuente: <https://www.mathworks.com/discovery/data-visualization.html>*

La figura siguiente muestra los resultados de la simulación de seguimiento. Las líneas de colores representan diferentes controladores: la referencia en rojo, el control PD en verde, el PID en azul, el g Adapt en marrón y el control por modos deslizantes en rosa. Se observa que los controladores "PID" y "Slide" funcionan bien, mientras que "PD" y "g Adapt" tienen dificultades para seguir las trayectorias deseadas.

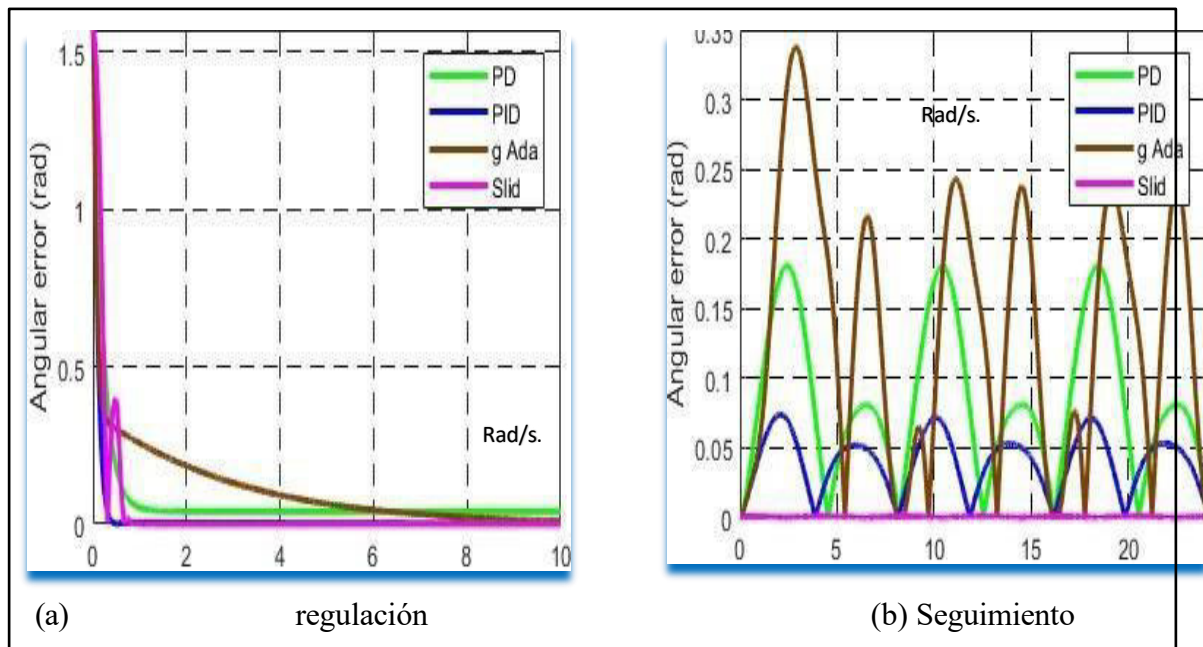
Figura 17

Resultados de simulación para seguimiento



Fuente: Elaboración propia utilizando MATLAB para la simulación y visualización de datos

En la figura a continuación, se muestra el error absoluto en la articulación q_3 para cada uno de los controladores previamente presentados. El gráfico a la izquierda ilustra el error para la regulación, mientras que el gráfico a la derecha muestra el error para el seguimiento.

Figura 18*Error en simulación para q_3* 

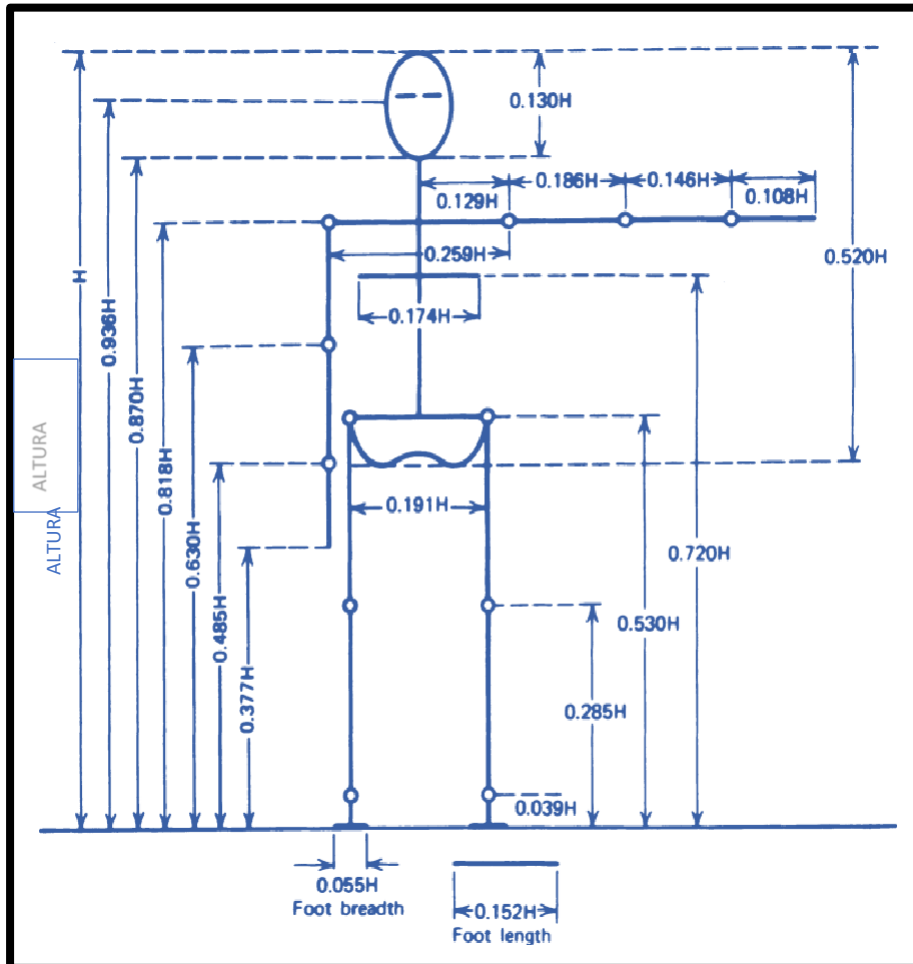
Fuente: Elaboración propia utilizando MATLAB para la simulación y visualización de datos

4.1.4 Diseño Conceptual del exoesqueleto

El diseño conceptual del exoesqueleto requiere medidas precisas del codo y el brazo. Para ello, se utilizaron medidas promedio basadas en la altura de una persona, según lo descrito en el libro “Biomechanics and Motor Control of Human Movement” de David A. Winter. Este diseño está optimizado para una persona con una altura de 1.68 metros. A continuación, se presenta un gráfico que ilustra cómo se calculan las distancias del brazo y el antebrazo, asegurando que el exoesqueleto se ajuste correctamente y proporcione el soporte necesario para una rehabilitación efectiva.

Figura 19

Medidas del cuerpo humano en función de la altura



Fuente: Elaboración propia utilizando medidas del cuerpo humano

4.1.5 Ecuaciones de distancia brazo/antebrazo

- Ecuación de la Distancia del Antebrazo:

$$\text{Distancia del antebrazo} = 0.630 H - 0.485 H = 0.145 H$$

$$H = 168 \text{ cm}$$

$$\text{Distancia del antebrazo} = 24.36 \text{ cm}$$

Esta ecuación se descompone en tres términos que representan proporciones específicas de la altura total ($H = 168 \text{ cm}$), resultando en una distancia final del antebrazo de 24.36 cm . Cada término corresponde a una fracción del antebrazo:

- $0.630H$: Estima la longitud desde el hombro hasta un punto por debajo del codo.
- $-0.485H$: Resta la longitud que excede más allá del codo.
- $+0.145H$: Ajusta para incluir la longitud completa del antebrazo hasta la muñeca.

Ecuación de la Distancia del Brazo:

$$\text{Distancia del brazo} = 0.818 H - 0.630 H = 0.188 H$$

$$H = 68 \text{ cm}$$

$$\text{Distancia del brazo} = 31.58 \text{ cm}$$

Calcula la distancia desde el hombro hasta el codo (31.58 cm) usando la diferencia entre dos proporciones de la altura:

- $0.818H$: Longitud total desde el hombro hasta un punto medio del antebrazo.
- $0.630H$: Excluye la porción del antebrazo para aislar solo el brazo superior.

-Relación entre Antebrazo y Brazo:

$$\text{Relación antebrazo} - \text{brazo} = \frac{24.36 \text{ c}}{31.58 \text{ c}} = 0.77$$

Esta relación indica que el antebrazo es el 77% de la longitud del brazo superior, una proporción vital para diseñar articulaciones que se muevan naturalmente con el usuario. Para ilustrar con mayor detalle cómo se derivan estas proporciones críticas, se presenta una tabla que resume las ecuaciones esenciales utilizadas en el cálculo de las dimensiones claves del antebrazo y el brazo.

Tabla 3

Mediciones para el diseño ergonómico del exoesqueleto

DESCRIPCIÓN	ECUACIÓN	RESULTADO
Distancia del antebrazo	$0.630H - 0.485H = 0.145H$	24.36 cm
Distancia del brazo	$0.818H - 0.630H = 0.188H$	31.58 cm
Relacion antebrazo- brazo	$24.36 \text{ cm} / 31.58 \text{ cm} =$ 0.77	77%

Fuente: Elaboración propia

4.1.6 Sistema general de control del exoesqueleto

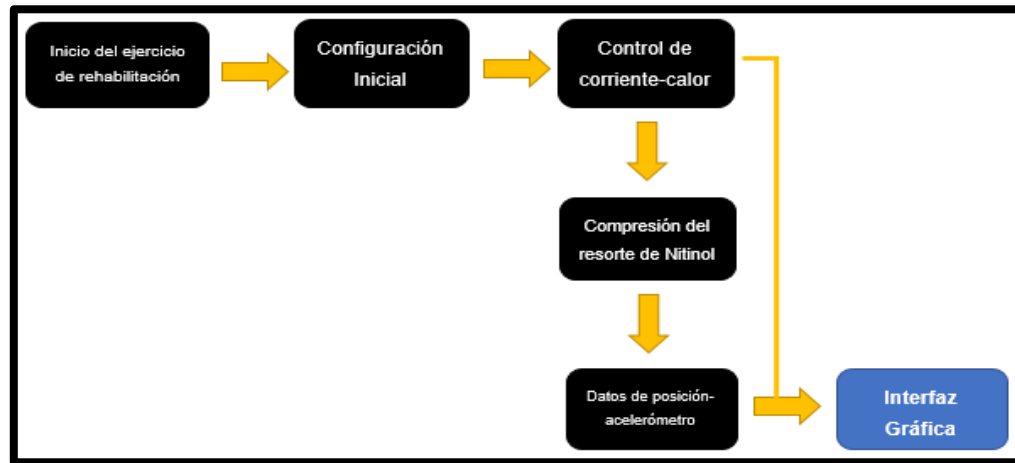
4.1.6.1 Diagrama general de procesos

El sistema en general tendrá el siguiente diagrama sobre el control y accionamiento del exoesqueleto:

- Se muestra que se tendrá un sensor de aceleración para conocer el desplazamiento del brazo.
- Se utilizará un sensor de temperatura como protección.
- Habrá un control de voltaje para comprimir el alambre.
- Se incluirá una interfaz de comunicación y visualización del proceso.

Figura 20

Diagrama general de accionamiento del exoesqueleto



Fuente: Elaboración propia utilizando PowerPoint para la creación del diagrama

4.1.7 Selección y diseño del actuador

4.1.7.1 Selección del Actuador de Alambre de Aleación

Para la selección del actuador de alambre de aleación, se evaluaron varias opciones considerando factores clave como disponibilidad, conductividad eléctrica, elasticidad y reutilización. Se realizó un análisis comparativo de los siguientes materiales:

- Aleación de Níquel-Titanio (Nitinol)
- Aleaciones de Cobre-Níquel-Aluminio
- Polímeros termoactivados
- Aleaciones de Cobre-Zinc-Aluminio

Tabla 4*Evaluación de aleaciones*

MATERIAL	Disponibilidad	Conductividad Eléctrica	Elasticidad	Reutilizable
Aleación Níquel Titanio	Baja	Alta	Media	Si
Aleación Cobre Níquel Aluminio	Media	Alta	Baja	Si
Polímeros termo activados	Media	Baja	Alta	No
Aleación Cobre Zinc Aluminio	Media	Alta	Baja	Si

Fuente: Elaboración propia

Tras analizar los datos, se seleccionó Nitinol por su alta conductividad eléctrica y capacidad de reutilización.

4.1.8 Modelado del Comportamiento del Actuador Nitinol

El modelado del comportamiento transitorio del actuador Nitinol es esencial para diseñar un exoesqueleto de codo que responda adecuadamente a estímulos térmicos. Utilizando el método de modelos autorregresivos y medias móviles (ARMA), se puede predecir y controlar la respuesta dinámica del actuador.

Método ARMA

El método ARMA se emplea para modelar las características dinámicas del Nitinol. Este enfoque permite capturar cómo responde el actuador a variaciones en temperatura y carga, proporcionando una base sólida para desarrollar algoritmos de control que ajusten la posición y fuerza del actuador en tiempo real.

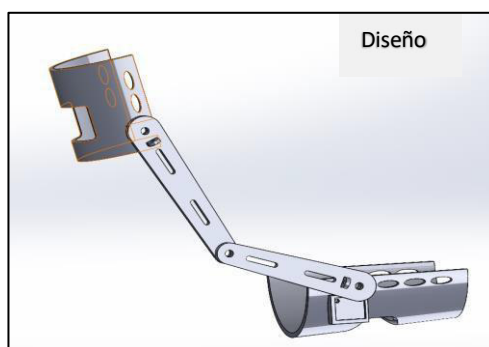
4.1.9 Diseño Mecánico del Exoesqueleto

Para el diseño mecánico del exoesqueleto, se utilizó el software SolidWorks. Se diseñaron las piezas correspondientes al antebrazo, brazo, codo-brazo y codo-antebrazo. Las uniones se realizaron utilizando pernos M6, seleccionados por ser la medida más adecuada y por la disponibilidad de tuercas ciegas en el mercado general.

El diseño del exoesqueleto se basó en las dimensiones aproximadas de las partes del cuerpo que se van a sostener. El exoesqueleto cuenta con un agarre para el antebrazo y el brazo, unidas mediante una junta móvil. Las piezas se ajustan al brazo mediante cintas de velcro, permitiendo adaptarse a diferentes grosores.

Figura 21

Diseño exoesqueleto



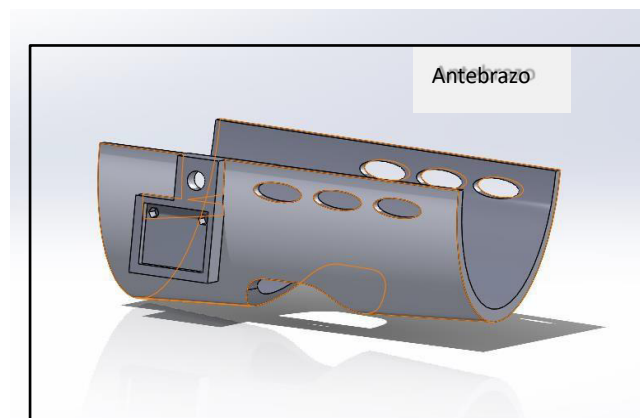
Fuente: Elaboración propia utilizando SolidWorks para el modelado 3D

Características del Antebrazo

- Sujeción mediante cintas de velcro.
- Ventilación en la parte inferior.
- Espacio para el acelerómetro.
- Punto de ajuste con perno y tuerca.

Figura 22

Antebrazo del exoesqueleto



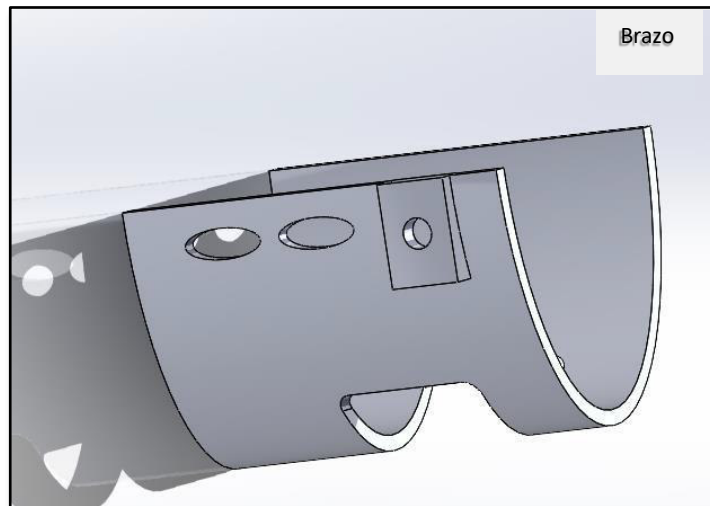
Fuente: Elaboración propia utilizando SolidWorks para el modelado 3D

Características del Brazo

- Sujeción mediante cintas de velcro.
- Ventilación en la parte inferior.
- Espacio para el acelerómetro.

Figura 23

Brazo del exoesqueleto



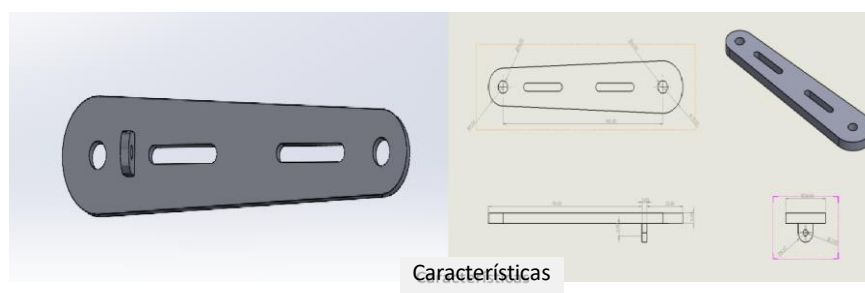
Fuente: Elaboración propia utilizando SolidWorks para el modelado 3D

Características del Codo-Brazo y Codo-Antebrazo

Las piezas de codo-brazo y codo-antebrazo tienen características similares, diferenciándose únicamente en la distancia entre centros de las uniones, que son de 15 cm y 12 cm respectivamente.

Figura 24

Características del codo-brazo y codo antebrazo



Nota: Maran, A. (2024). Imagen de articulación afectada por artritis.

<http://msdmanuals.com/es/hogar/trastornos-de-los-huesos-articulaciones-y-m%C3%BAsculos/enfermedades-articulares/artritis-reumatoide>

Fuente: Elaboración propia utilizando SolidWorks para el modelado 3D

Ensamblaje del Diseño Mecánico

El ensamblaje del exoesqueleto se muestra en la figura siguiente. Este diseño permite un ajuste ergonómico y cómodo, facilitando su uso en rehabilitación y asistencia funcional.

Figura 25

Ensamblaje del diseño mecánico.



Fuente: Elaboración propia

4.2 Implementación de Actuadores de AMF en el Exoesqueleto de Codo

Los actuadores de alambre de aleación con memoria de forma (Nitinol) se implementan en el exoesqueleto de codo para proporcionar movimiento y fuerza controlada. Estos actuadores son ideales debido a su capacidad para generar contracción en respuesta a estímulos térmicos, permitiendo un control preciso y adaptable del exoesqueleto.

Ventajas de los Actuadores de AMF en Exoesqueletos

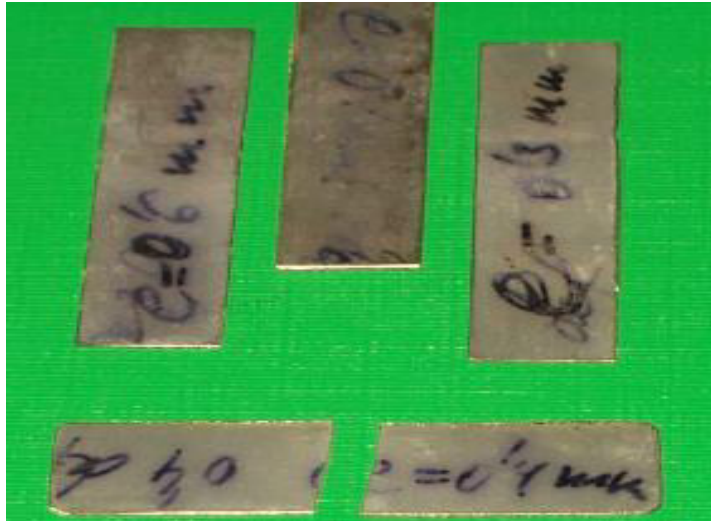
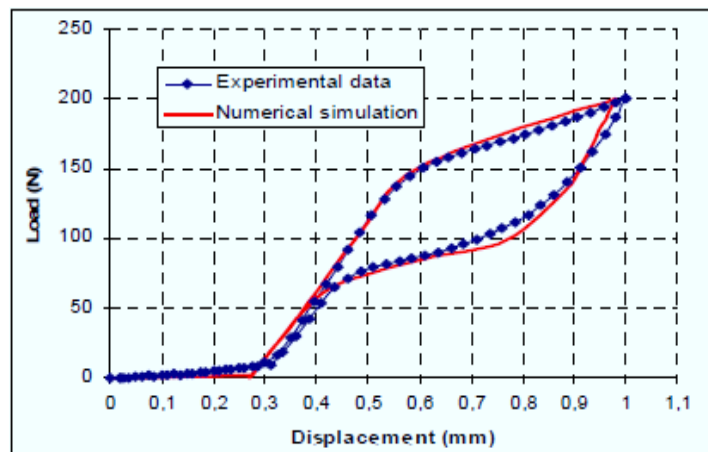
- **Peso Ligero:** Reducen la carga sobre el usuario.
- **Funcionamiento Silencioso:** Mejoran la comodidad del usuario.
- **Alta Densidad de Energía:** Permiten movimientos con gran fuerza en un espacio

reducido.

Procedimientos de Implementación

Para la implementación de los actuadores de Nitinol en el exoesqueleto, se realizaron las siguientes etapas:

1. **Preparación del Material:** Se importaron láminas de Nitinol y se recortaron en el laboratorio utilizando tecnología láser. Este proceso garantiza precisión en las dimensiones y calidad del material.
2. **Modelado y Simulación:** Utilizando software de simulación, se desarrollaron modelos que reflejan la superelasticidad y memoria de forma del Nitinol. Se realizaron pruebas de flexión en tres puntos para definir las características de rigidez a flexión del material.
3. **Integración en el Exoesqueleto:** Las láminas de Nitinol se integraron en las partes móviles del exoesqueleto, ajustándose para maximizar la eficiencia de movimiento y fuerza.
4. **Pruebas y Ajustes:** Se realizaron pruebas iniciales con el exoesqueleto ensamblado para verificar la funcionalidad del actuador de Nitinol. Se ajustaron parámetros de control basados en los resultados obtenidos.

Figura 26*Láminas de nitinol**Fuente: Elaboración propia utilizando algunas láminas de Nitinol***Figura 27***Prueba simulada del nitinol*

Nota: Basado en "Aplicación de aleaciones con memoria de forma en el diseño y fabricación de férulas para corrección de las deformidades articulares de los dedos de la mano", por José M. Pérez-García y Luis Gracia, 2007, Biomecánica, 15(1).

Fuente: <https://doi.org/10.5821/sibb.v15i1.1755>

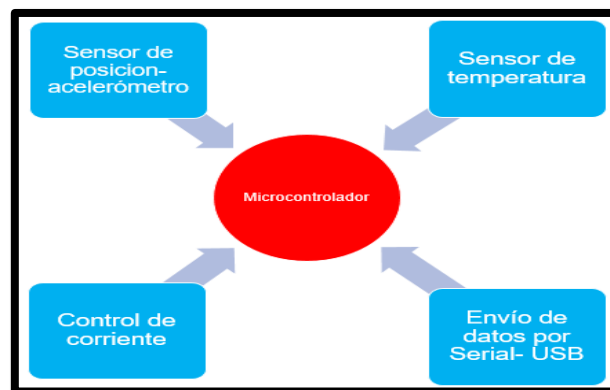
4.3 Diseño electrónico de control del exoesqueleto

4.3.1 Diagrama del sistema de control

Para el diseño electrónico del exoesqueleto, se deben considerar los siguientes componentes, todos alimentados por una fuente de 5V:

Figura 28

Diagrama del sistema de control del exoesqueleto



Fuente: Elaboración propia utilizando PowerPoint para la creación del diagrama

El diagrama ilustra la interconexión entre los principales componentes electrónicos del exoesqueleto. En el centro se encuentra el microcontrolador, que actúa como el cerebro del sistema. Este microcontrolador está conectado a varios sensores y módulos esenciales:

1. **Sensor de Temperatura:** Monitorea la temperatura del exoesqueleto para evitar sobrecalentamiento y asegurar un funcionamiento seguro.
2. **Sensor de Posición - Acelerómetro:** Detecta y registra los movimientos del brazo, proporcionando datos precisos sobre la posición y aceleración del miembro.
3. **Control de Corriente:** Regula la cantidad de corriente suministrada a los

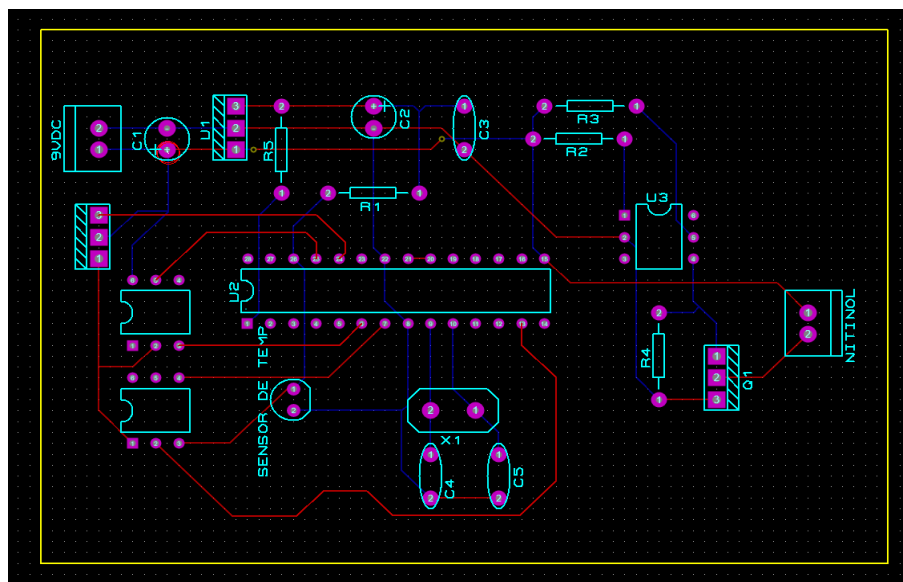
Fuente: Elaboración propia utilizando

4.5 Diseño de la placa de circuito impreso

Posteriormente, se llevó a cabo el diseño de una placa electrónica para integrar y conectar todos los componentes del sistema. Algunos componentes, como el acelerómetro, están directamente anclados al exoesqueleto, por lo que su conexión será directa a los pines correspondientes de la placa. El convertidor CP2103 estará conectado al puerto USB del servidor, el cual procesará todos los datos recibidos. Este diseño asegura una interconexión eficiente y ordenada de todos los módulos, optimizando la comunicación y el funcionamiento del exoesqueleto.

Figura 30

Diseño de la placa de circuito impreso (PCB) del exoesqueleto



Nota: Utilizando Altium Designer para el diseño de PCB

Fuente: Elaboración propia

En el diseño se pueden observar los siguientes elementos:

1. **Microcontrolador ATMEGA328P:** Ubicado en el centro de la PCB, actuando como el cerebro del sistema y gestionando todas las señales de entrada y salida.
2. **Conectores de Alimentación:** En la parte superior izquierda, se encuentra el conector de alimentación de 9V, el cual proporciona energía a toda la placa.
3. **Sensor de Temperatura:** Conectado cerca del microcontrolador para monitorear la temperatura y evitar el sobrecalentamiento.
4. **Acelerómetro MPU6050:** Conectado a la parte inferior izquierda, se encarga de medir la posición y el movimiento del brazo.
5. **Mosfet IRFZ44N:** Ubicado a la derecha, actúa como controlador de corriente para el calentamiento del Nitinol.
6. **Convertidor USB a TTL (CP2103):** Situado en la parte inferior izquierda, permite la comunicación serial entre el microcontrolador y un servidor.
7. **Componentes Pasivos (Resistencias y Capacitores):** Distribuidos a lo largo de la placa, necesarios para el correcto funcionamiento de los circuitos.
8. **Conector para Nitinol:** En la parte derecha, permite la conexión del Nitinol que se calentará para provocar su contracción.

Después del diseño electrónico del control de temperatura por pulsos PWM configurados según la cantidad de fuerza constante requerida para los ejercicios de rehabilitación, se mandó a fabricar la placa electrónica con los componentes necesarios para hacer la conexión.

Figura 31

Vista tridimensional de la placa de circuito impreso (PCB)

Figura 32

Scrip de lectura de datos del dispositivo

```

accedino
1 #include <MPU6050_tockn.h>
2 #include <Wire.h>
3
4 MPU6050 mpu6050(Wire);
5
6 long timer = 0;
7
8 void setup() {
9   Serial.begin(9600);
10  Wire.begin();
11  mpu6050.begin();
12  //mpu6050.calcGyroOffsets(true);
13 }
14
15 void loop() {
16   mpu6050.update();
17
18   if(millis() - timer > 10){
19
20     Serial.print(mpu6050.getAccX()*9.8);
21     Serial.print(",");
22     Serial.print(mpu6050.getAccY()*9.8);
23     Serial.print(",");
24     Serial.print(mpu6050.getAccZ()*9.8);
25     Serial.print(",");
26     Serial.print(mpu6050.getGyroX());
27     Serial.print(",");
28     Serial.print(mpu6050.getGyroY());
29     Serial.print(",");
30     Serial.print(mpu6050.getGyroZ());
31     Serial.println("");
32
33     timer = millis();
34   }
35 }
36
code_robotic_arm.py
1 import numpy as np
2 import pandas as pd
3 import serial
4 import matplotlib.pyplot as plt
5 import matplotlib.animation as animation
6 import datetime
7
8 date = datetime.datetime.now()
9 name_file = date + '.csv'
10
11 def start_serial():
12
13   # Especifica el puerto COM y la velocidad de transmisión
14   puerto_com = 'COM3' # Ajusta el número del puerto COM según tu configuración
15   velocidad = 9600
16
17   # Crea un objeto Serial
18   ser = serial.Serial(puerto_com, velocidad)
19   return ser
20
21 def show_angles():
22
23   # Función de inicialización
24   def inicializar():
25     linea.set_data([], [])
26     ax.plot([0, 0], [0, 1], label='Brazo', color='blue', linewidth=2)
27     circulo = plt.Circle((0, 0), radius=0.05, edgecolor='y', facecolor='yellow', label = "codo")
28     ax.add_patch(circulo)
29     return linea,
30
31   def inicializar2():
32     linea2.set_data([], [])
33     #ax2.plot([0, 0], [0, 1], label='Brazo', color='blue', linewidth=2)
34     circulo = plt.Circle((0, 0), radius=0.05, edgecolor='y', facecolor='yellow', label = "codo")
35     ax2.add_patch(circulo)
36

```

Nota: Utilizando Arduino IDE para la programación del microcontrolador y

Python para el procesamiento y visualización de datos

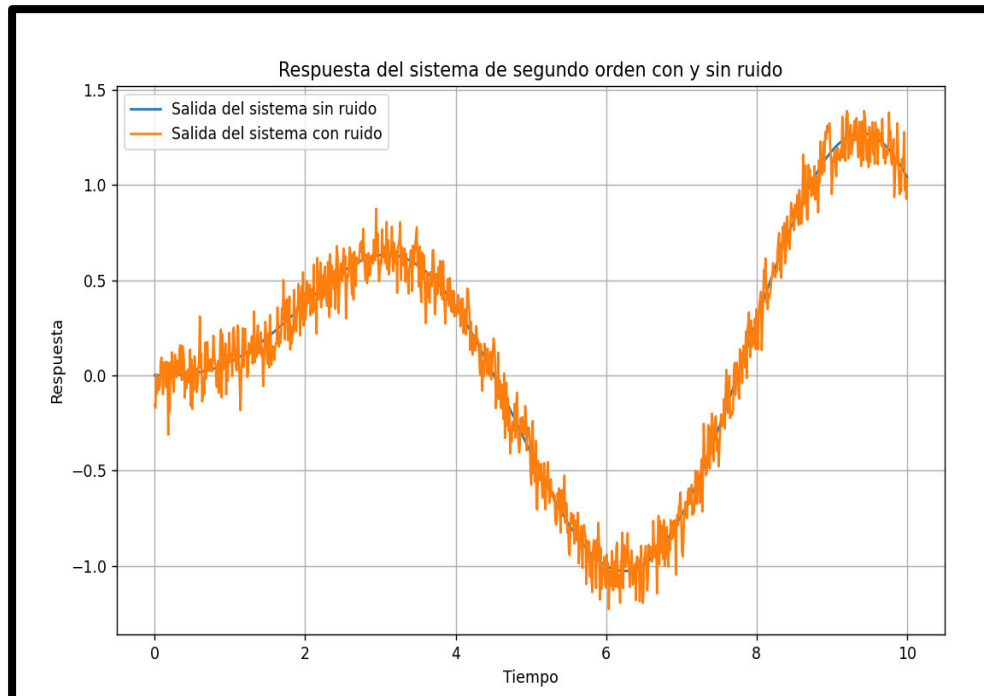
Fuente: Elaboración propia

4.6.2 Filtrado y análisis dinámico de datos

Posteriormente, para el procesamiento y análisis dinámico de los datos, se considera la variación de los ángulos al aplicar pulsos de corriente para accionar el sistema. Este proceso genera un poco de ruido, por lo que se implementa un filtro de primer orden para eliminar el ruido estocástico presente. Este filtrado es crucial para asegurar que las señales analizadas sean precisas y reflejen fielmente los movimientos del exoesqueleto, mejorando así, la calidad del control y la respuesta del sistema.

Figura 33

Gráfica de la respuesta del sistema de segundo orden



Nota: Utilizando Python y la biblioteca Matplotlib para la simulación y visualización de datos. Fuente: Elaboración propia

4.7 Modelos ARMA

4.7.1 Ajuste y predicción de modelos ARIMA

Con la nueva señal filtrada, se utilizará otra función para el análisis dinámico de la señal, aproximándola a las señales ARMA (AutoRegressive Moving Average) con sus diversas configuraciones. Para ello, se divide la señal en conjuntos de entrenamiento y prueba. Luego, se ajusta el modelo ARIMA (AutoRegressive Integrated Moving Average) a los datos de entrenamiento y se realiza la predicción sobre el conjunto de prueba. Los resultados del modelo se evalúan mediante el cálculo del error cuadrático medio (MSE) y el porcentaje de

acuerdo del modelo. Esta metodología permite una aproximación más precisa y robusta de la dinámica de la señal, proporcionando una mejor comprensión y control del comportamiento del exoesqueleto. Los resultados obtenidos son los siguientes:

Figura 34

Código para el ajuste y predicción de un modelo ARIMA

```
input_signal= t
output_signal=y
# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(input_signal, output_signal, test_size=0.2, random_state=42)

# Ajustar el modelo ARIMA
order = (2, 1, 1) # Orden del modelo ARIMA
model = ARIMA(y_train, order=order, exog=X_train.reshape(-1, 1))
results = model.fit()

# Predecir sobre el conjunto de prueba
y_pred = results.predict(start=0, end=len(X_test)-1, exog=X_test.reshape(-1, 1))

# Calcular el error cuadrático medio (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calcular el porcentaje de acierto (1 - MSE)
accuracy = 1 - mse

print("Porcentaje de acierto del modelo:", accuracy)
```

Fuente: Elaboración propia utilizando Python y la biblioteca statsmodels para el modelado ARIMA

4.7.2 Comparación y selección del mejor modelo

Se muestran los resultados de 3 Modelos ARMA para ver cual se ajusta más a los datos recopilados.

Modelo 1 ARMA (2, 1 ,1)

Figura 35

Porcentaje de acierto del modelo ARMA (2,1,1)

```

MODELO (2, 1, 1)
Porcentaje de acierto del modelo: 0.9489336715952791
Parámetros ARMA: [0.03330828 0.06389151] [-0.99991326]
Parámetro de intercepto: 0.04220437275959864

```

Fuente: Elaboración propia utilizando Python y la biblioteca statsmodels para el análisis del modelo ARMA

La imagen muestra los resultados de un modelo ARIMA (2, 1, 1), indicando un porcentaje de acierto del 94.89%. También se presentan los parámetros ARMA del modelo y el parámetro de intercepción.

Modelo 2 ARMA (3, 1, 1)

Figura 36

Porcentaje de acierto del modelo ARMA (3,1,1)

```

MODELO (3, 1, 1)
Porcentaje de acierto del modelo: 0.9491136922716987
Parámetros ARMA: [ 0.03426287  0.06424633 -0.01583451] [-0.99993085]
Parámetro de intercepto: 0.042189968606167465

```

Fuente: Elaboración propia utilizando Python y la biblioteca statsmodels para el análisis del modelo ARMA

La imagen muestra los resultados del modelo ARIMA (3, 1, 1), mostrando un porcentaje de acierto del 94.91%, los parámetros ARMA y el parámetro de intercepción.

Modelo 3 ARMA (2, 2, 1)

Figura 37

Porcentaje de acierto del modelo ARMA (2,2,1)

```

MODELO (2, 2, 1)
Porcentaje de acierto del modelo: 0.9309823407078633
Parámetros ARMA: [-0.6608014 -0.28549842] [-0.99984812]
Parámetro de intercepto: 0.05518839962008176

```

Fuente: Elaboración propia utilizando Python y la biblioteca statsmodels para el análisis del modelo ARMA

La imagen muestra los resultados del modelo ARIMA (2, 2, 1), mostrando un porcentaje de acierto del 93.09%, los parámetros ARMA y el parámetro de intercepción.

Ecuación de control diseñada

Teniendo en cuenta la complejidad de modelar estos sistemas, se considera la capacidad de cálculo de un microcontrolador. Por ello, entre el Modelo 1 y el Modelo 2, se usará el Modelo 1, ya que ambos tienen una precisión de aproximadamente 0.949, pero el Modelo 1 es solo de orden 2, mientras que el Modelo 2 es de orden 3. Por lo tanto, se escoge el Modelo 1 como el mejor modelo para la respuesta dinámica de la planta de control (Exoesqueleto).

Por ello la ecuación de control diseñada para esta planta es la siguiente:

$$\frac{H(z)}{A(z)} = B(z)$$

$$\frac{H(z) = 0.033308 + 0.063891 z^{-1}}{1 - 0.999913 z^{-1}}$$

4.8 Clasificador por redes neuronales

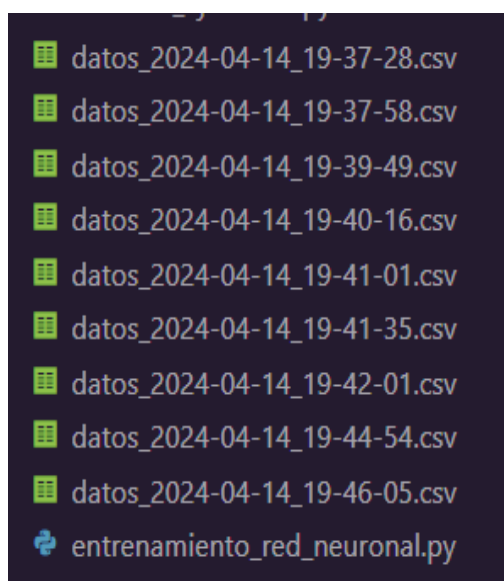
4.8.1 Recopilación de datos y sesiones de rehabilitación

Después del modelado dinámico se realizó las recopilaciones de varias sesiones de rehabilitación de codo con el exoesqueleto sin el uso del actuador de Nitinol, por ello se realizó un entrenamiento de una red neuronal para lograr clasificar el tipo de apoyo requerido en función a la calidad del ejercicio.

Para realizar el entrenamiento primero es necesario estandarizar el proceso de los archivos encontrados, inicialmente se tiene a los archivos según el tiempo, aceleraciones, velocidades de giro y fecha. Además de los datos se generó un archivo con los tiempos de inicio del ejercicio, nombre del paciente y el grado de ayuda.

Figura 38

Archivos Generados por lectura de ejercicio



Nota: utilizando Python para la captura y almacenamiento de datos

Fuente: Elaboración propia

Por ello con el siguiente código se realizaron las lecturas de los datos procesados de las posiciones del codo y almacenarlo en archivos csv para su posterior conversión en datos que indiquen el resumen del ejercicio.

Figura 39

Configuración del puerto serial y la generación de un archivo CSV

```
def start_serial():
    # Generar archivo csv
    fecha_actual = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

    # Crear el nombre del archivo CSV con la fecha y hora actual
    archivo_csv = f'datos_{fecha_actual}.csv'

    # Búsqueda del puerto serial final
    puerto_COM = "COM3"

    # Configuración del puerto serial
    puerto_serial = serial.Serial(puerto_COM, 9600) # Ajusta el puerto y

    return puerto_serial, archivo_csv

def update_plot(frame, puerto_serial, archivo_csv):
    global x1, y1_1, y1_2, y1_3, x2, y2_1, y2_2, y2_3
    global x3, y3_1, x4, y4_1
    global y_anterior

    try:
        # Leer una línea desde el puerto serial
        lectura_serial = puerto_serial.readline().decode('utf-8').strip()
        tiempo = frame * t_interval / 1000
        datos = lectura_serial.split(',')
        a_x, a_y, a_z = float(datos[0]), float(datos[1]), float(datos[2])
        g_x, g_y, g_z = float(datos[3]), float(datos[4]), float(datos[5])

        with open(archivo_csv, 'a', newline='') as archivo_csv:
            # Abrir el archivo CSV en modo de escritura ('a')
```

Nota: utilizando Python para la captura de datos y generación de archivos CSV

Fuente: Elaboración propia

Los datos iniciales de estos archivos csv están conformados de la siguiente manera, donde el tiempo de captura de eventos es cada 100 ms para lograr graficar el movimiento y poder procesar sin pérdida de datos.

Figura 40

Variables iniciales de recepción

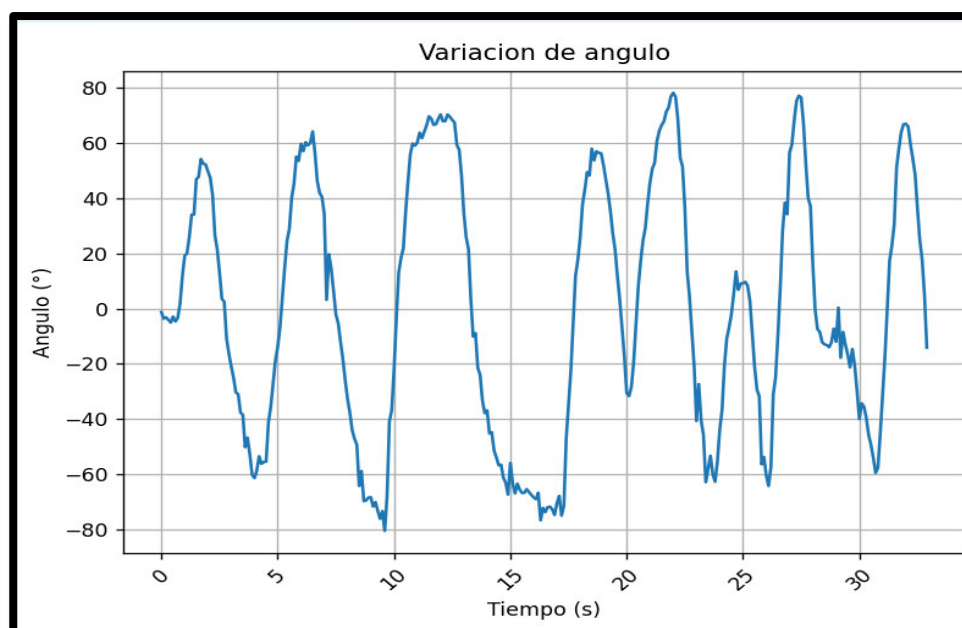
	tiempo	ax	ay	az	gx	gy	gz
0	0.0	9.19	0.20	3.37	-6.69	0.31	-5.88
1	0.1	9.61	0.61	2.84	-2.18	1.08	-0.66
2	0.2	10.00	0.54	2.83	-5.21	1.36	-5.19
3	0.3	9.76	0.69	2.84	-2.53	0.69	-5.65
4	0.4	9.77	0.86	3.34	-2.40	5.60	-0.89

Fuente: Elaboración propia utilizando Python para la captura y visualización de datos

Se realizaron las capturas de la variación del ángulo en el tiempo según el ejercicio de cada paciente.

Figura 41

Gráfico de variación de ángulos en un ejercicio simple



Fuente: Elaboración propia utilizando Python y la biblioteca Matplotlib para la visualización de datos

4.8.2 Configuraciones y resultados de la entrada de red neuronal

Para la entrada de la red neuronal, se utilizará un resumen del ejercicio según los ciclos realizados, para ello se transformará los datos recopilados del acelerómetro en un resumen de tiempos y calidad de ejercicio según los ciclos del ejercicio dado, se muestra el siguiente código:

Figura 42

Código para resumir ciclos de ejercicios

```
def resumir_ejercicio(df):
    tiempo_inicio = df['tiempo'].min()
    tiempo_final = df['tiempo'].max()
    duracion = tiempo_final - tiempo_inicio

    # Aplicar transformada rápida de Fourier (FFT)
    fft_resultado = np.fft.fft(df['ax'])
    frecuencias = np.fft.fftfreq(len(df['ax']))

    # Encontrar la frecuencia dominante (excluyendo la frecuencia cero)
    frecuencia_dominante = np.abs(frecuencias[np.argmax(np.abs(fft_resultado[1:])) + 1])

    # Calcular el número de ciclos en función de la frecuencia dominante
    total_ciclos = len(df) / frecuencia_dominante

    fuerza_promedio = np.power(np.power(df["ax"], 2) + np.power(df["ay"], 2) + np.power(df["az"], 2) - 9.81^2, 1/2) * 20 * 9.81

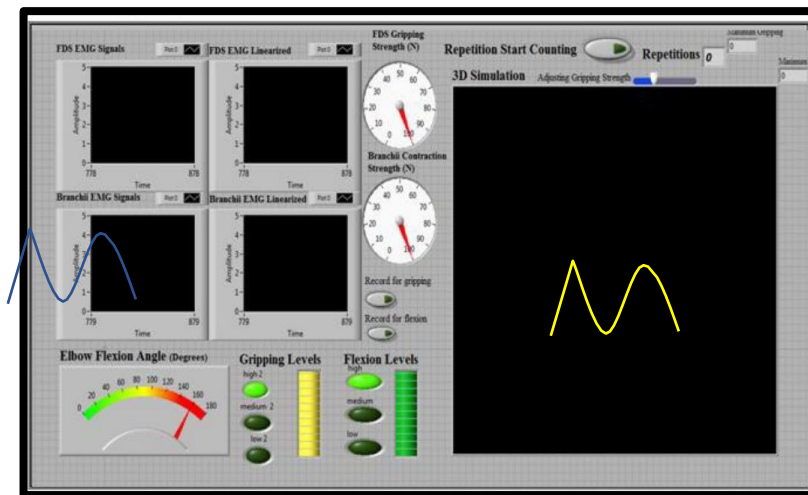
    valores_medidos = np.arctan2(-1 * df["ay"], df["ax"])
    mse = mean_squared_error(modelo_ejercicio, valores_medidos)
    precision = r2_score(modelo_ejercicio, valores_medidos)

    lista_valores = [tiempo_inicio, tiempo_final, duracion, frecuencia_dominante, total_ciclos, fuerza_promedio, precision, mse]
```

Fuente: Elaboración propia utilizando Python para el análisis de ciclos de ejercicios

Figura 43

Pestaña para modo de operación del exoesqueleto



Fuente: Elaboración propia utilizando LabVIEW para la simulación y control de dispositivos

Se procesaron los datos del ejercicio en general, comenzando por registrar el tiempo inicial y el tiempo final. A partir de estos datos, se calculó el tiempo total del ejercicio mediante la diferencia entre ambos registros. Para analizar la señal del ejercicio, se aplicó la transformada rápida de Fourier (FFT), lo que permitió descomponer la señal en sus componentes de frecuencia.

Esta aproximación facilitó la identificación de los ciclos y la frecuencia de los ciclos en el ejercicio realizado.

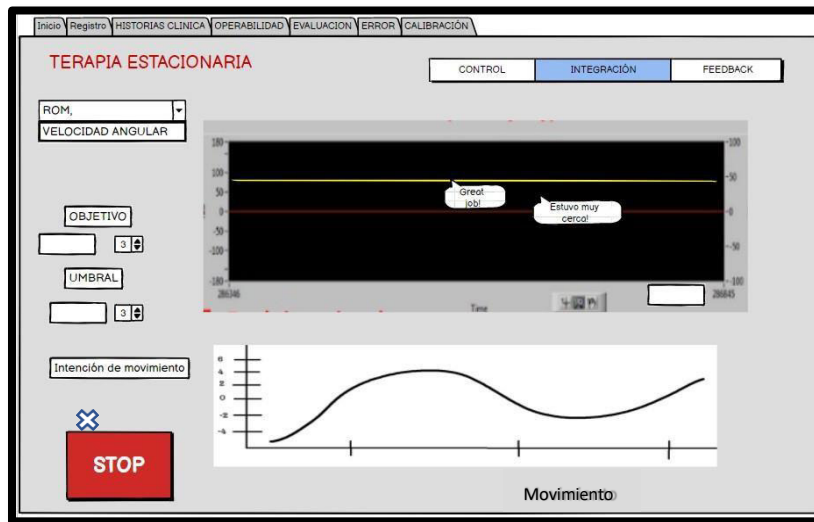
La transformada rápida de Fourier se puede expresar como:

$$\text{FFT}(x(t)) = \int x(t) * e^{(-i * 2 * \pi * f * t)} dt$$

Donde $x(t)$ es la señal de tiempo y f representa las frecuencias.

Figura 44

Pestaña de evaluación de diferentes parámetros del exoesqueleto



Fuente: Elaboración propia utilizando LabVIEW para la evaluación y control de parámetros.

Finalmente, se etiquetaron los datos de fuerza del codo realizados y requeridos para cada paciente, categorizándolos según el grado de apoyo necesario. Este etiquetado se basó en la evaluación del esfuerzo durante el ejercicio, permitiendo una clasificación precisa de la asistencia requerida.

Figura 45

Variables de resumen en cada ejercicio

Tiempo_Inicio	Tiempo_Final	Duracion	Frecuencia_Dominante	Velocidad_prom	...	Precision	Angulo_menor	Angulo_mayor	MSE	Fuerza
1.7	29.6	27.9	1.734359	18.864634	...	0.889658	-31.087261	92.740397	0.095347	medio
1.9	8.4	6.5	0.977516	33.485958	...	0.838643	-30.790507	82.853509	0.093682	alto
6.9	44.2	37.3	1.619287	10.232295	...	0.858544	-43.049372	92.306539	0.039407	medio
9.3	67.2	57.9	1.369147	8.120162	...	0.827564	-23.081187	77.794844	0.039088	bajo
0.2	43.9	43.7	1.473214	16.788351	...	0.842789	-43.148414	89.400132	0.057391	alto
7.7	55.8	48.1	1.249728	8.090784	...	0.907919	-29.961080	79.093421	0.075384	medio
4.1	58.9	54.8	1.020402	9.669992	...	0.859771	-40.090386	76.373463	0.025523	alto
7.2	43.1	35.9	0.546721	17.532487	...	0.895885	-14.361766	66.773077	0.053400	alto
1.1	26.1	25.0	1.183971	32.255271	...	0.892218	-16.774093	75.546870	0.029276	alto
8.7	51.6	42.9	1.678393	31.130511	...	0.858565	-23.486664	69.277105	0.078795	bajo

Fuente: Elaboración propia utilizando Python para la captura y análisis de datos

Con ello, se logró combinar los dataframes de cada ejercicio junto con el grado de apoyo requerido, obteniendo así el dataframe necesario para modelar y entrenar la red neuronal. Este proceso incluyó la normalización de los datos para garantizar la uniformidad en las entradas del modelo. Adicionalmente, se anonimizó la data para proteger la privacidad de los pacientes.

La combinación de los dataframes se realizó utilizando la siguiente fórmula de concatenación:

$DF_combined = \text{concat}(DF_exercise,$

$DF_support_level)$ Donde:

- $DF_combined$ es el dataframe resultante de la combinación.
- $DF_exercise$ es el dataframe que contiene los datos de cada ejercicio.
- $DF_support_level$ es el dataframe que contiene los niveles de apoyo requeridos.

Para normalizar los datos, se aplicó la siguiente fórmula:

$X_normalized = (X -$

$\text{mean}(X)) / \text{std}(X)$ Donde:

- $X_normalized$ es el conjunto de datos normalizados.
- X es el conjunto de datos original.
- $\text{mean}(X)$ es la media de los datos.
- $\text{std}(X)$ es la desviación estándar de los datos.

Finalmente, la anonimización de los datos se llevó a cabo mediante técnicas de ofuscación y eliminación de información personal identificable (PII), asegurando que los datos utilizados para el entrenamiento del modelo cumplieran con los estándares de privacidad.

Figura 46*Creación de dataset de ejercicios por paciente*

```
def lectura(directorio):

    list_resumen = []
    for archivo in os.listdir(directorio):
        if archivo.endswith('.csv'): # Filtrar solo archivos CSV (p
            ruta_completa = os.path.join(directorio, archivo)
            # Leer el archivo CSV y agregar el DataFrame a la lista
            df = pd.read_csv(ruta_completa)
            list_resumen.append(resumir_ejercicio(df))

    pass
```

Fuente: Elaboración propia utilizando Python y la biblioteca Pandas para la manipulación de archivos CSV

4.8.3 Configuraciones y resultados de modelos neuronales

De esa manera se obtuvo el dataset necesario para una red neuronal de clasificación, el cual puede predecir el grado de apoyo requerido por el paciente durante los ejercicios de codo. Los niveles de apoyo se clasifican en BAJO, MEDIO y ALTO. Para lograr esto, se desarrollaron varios modelos de redes neuronales, alternando los hiperparámetros hasta encontrar el modelo óptimo que logre una clasificación precisa.

A continuación, se presentan las configuraciones de los tres modelos de red neuronal seleccionados:

Tabla 5*Modelos de red neuronal propuestos*

CAPAS	Modelo A	Modelo B	Modelo C
CAPA 1	128	64	256
Función Activación	RELU	RELU	RELU
CAPA 2	64	32	128
Función Activación	SIGMOID	RELU	RELU
CAPA 3	32	16	64
Función Activación	RELU	SIGMOID	RELU
CAPA 4	4	4	4
Función Activación	SOFTMAX	SOFTMAX	SOFTMAX

Fuente: Elaboración propia

Los resultados del entrenamiento se muestran a

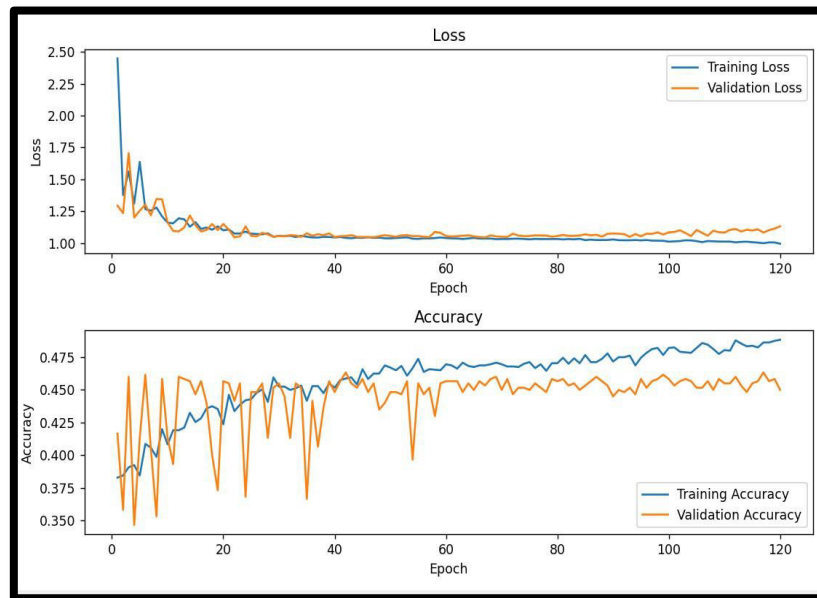
continuación: Modelo A:

Los resultados del modelo A muestran un punto mínimo en la pérdida entre las épocas 60 y 80. A partir de esa época, comienza a presentarse el

fenómeno de overfitting. Además, la precisión del modelo solo alcanza alrededor de 0.5, lo cual indica que el modelo no está logrando predecir con éxito el grado de apoyo requerido por el paciente.

Figura 47

Resultados de Accuracy y Loss en modelo A.

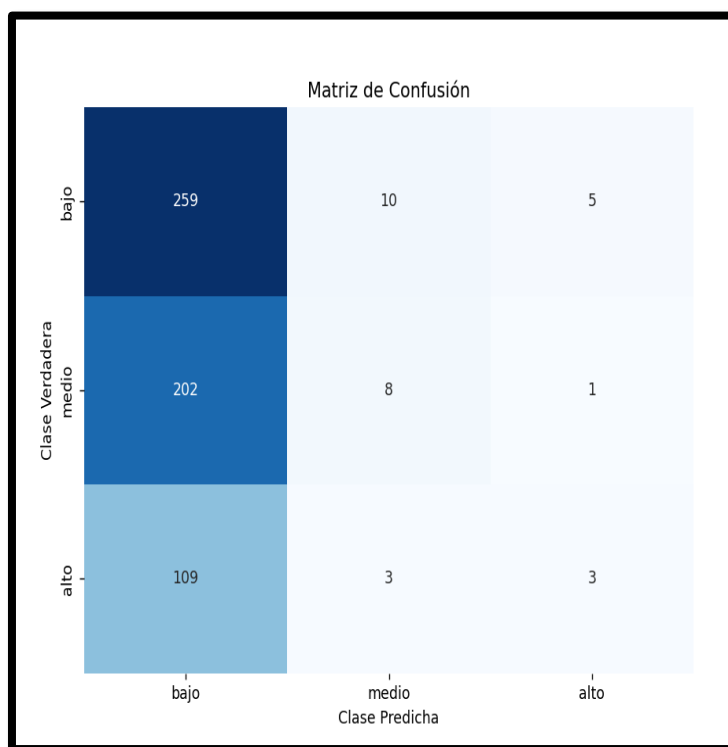


Fuente: Elaboración propia utilizando Python y la biblioteca Matplotlib para la visualización de resultados de entrenamiento

La matriz de confusión presenta los resultados de la clasificación de un modelo de red neuronal. En la diagonal principal, se encuentran las clasificaciones correctas: 259 predicciones correctas para la clase "bajo", 8 para "medio" y 3 para "alto". Los elementos fuera de la diagonal representan errores de clasificación: por ejemplo, 202 instancias de la clase "medio" fueron incorrectamente clasificadas como "bajo". Esta matriz indica que el modelo tiene dificultades para clasificar correctamente las clases "medio" y "alto", mientras que la clase "bajo" es clasificada con mayor precisión.

Figura 48

Matriz de confusión del modelo A



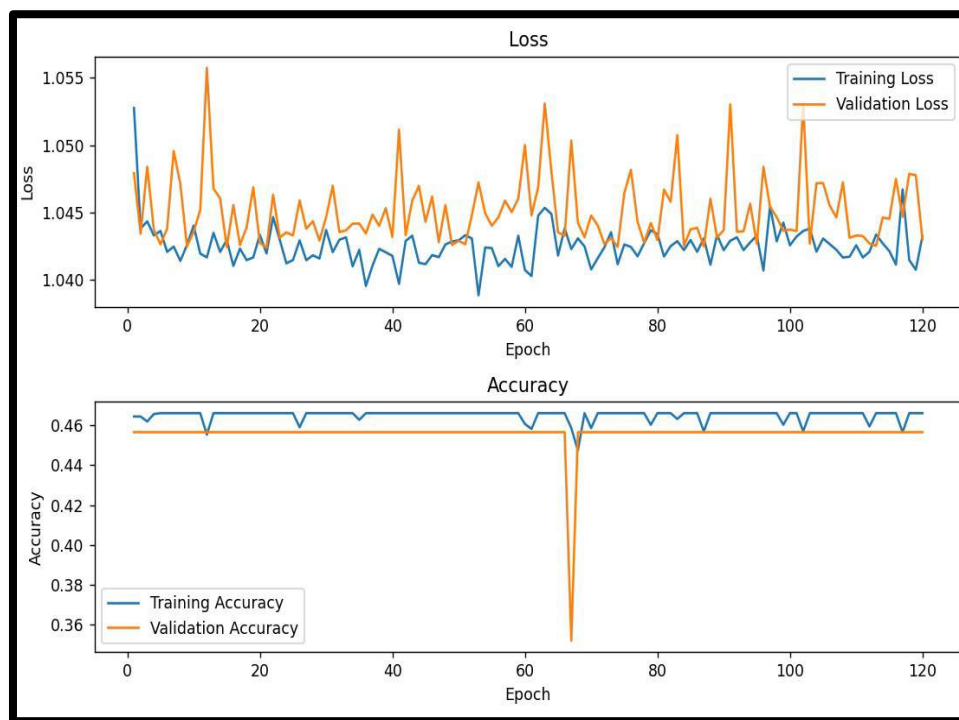
Fuente: Elaboración propia utilizando Python y la biblioteca Seaborn para la visualización de la matriz de confusión

Modelo B:

Los resultados del modelo B muestran una alta variación en el entrenamiento sin lograr llegar a un mínimo, lo mismo sucede con la precisión del modelo llegando hasta un máximo de 0.48, lo cual indica que no está realizando una correcta clasificación.

Figura 49

Resultados de Accuracy y Loss en modelo B

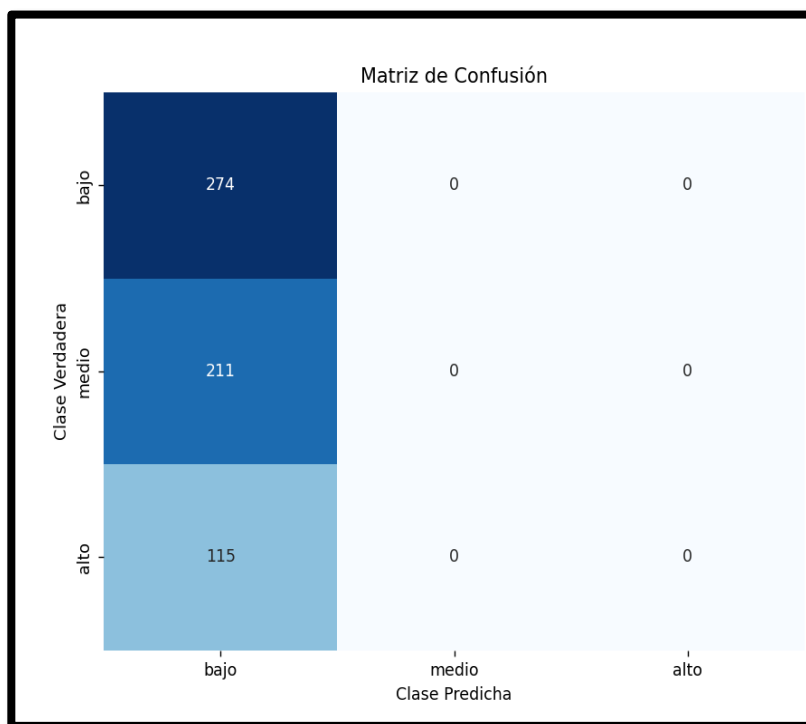


Fuente: Elaboración propia utilizando Python y la biblioteca Matplotlib para la visualización de resultados de entrenamiento

La matriz de confusión muestra que el modelo clasifica correctamente todas las instancias de la clase "bajo" (274 casos). Sin embargo, tiene problemas significativos con las clases "medio" y "alto", ya que todas las instancias de estas clases fueron incorrectamente clasificadas como "bajo". Esto indica que el modelo no está diferenciando bien entre las tres categorías y tiende a clasificar la mayoría de las instancias como "bajo". La precisión del modelo es alta para la clase "bajo", pero muy baja para las clases "medio" y "alto", lo que sugiere la necesidad de mejorar la capacidad del modelo para distinguir entre las diferentes clases.

Figura 50

Matriz de confusión del modelo B



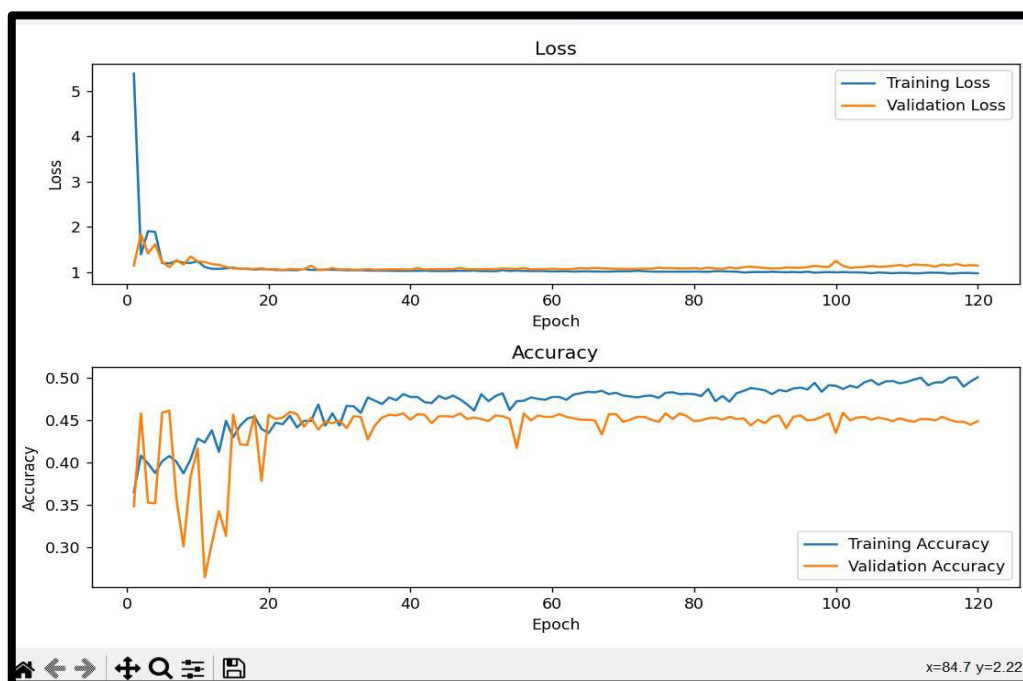
Fuente: Elaboración propia utilizando Python y la biblioteca Seaborn para la visualización de la matriz de confusión

Modelo C:

En el modelo C se muestra una pérdida mínima en torno a las 100 épocas, además de observar que la precisión va aumentando, por ello se escogió este modelo, ya que se podía conseguir mejores resultados aumentando la cantidad de neuronas según el modelo inicial.

Figura 51

Resultados de Accuracy y Loss en modelo C.



Fuente: Elaboración propia utilizando Python y la biblioteca Matplotlib para la visualización de resultados de entrenamiento

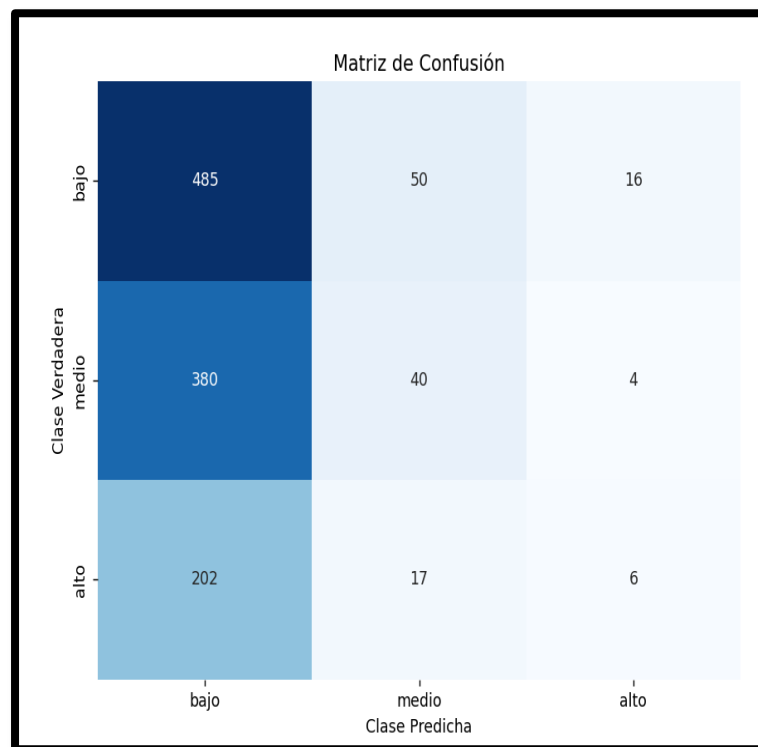
La matriz de confusión muestra que el modelo tiene un buen desempeño en la clasificación de la clase "bajo", con 485 instancias clasificadas correctamente. Sin embargo, hay errores significativos en la clasificación de las clases "medio" y "alto". Muchas instancias de las clases "medio" y "alto" son incorrectamente clasificadas como "bajo".

- **Clase "bajo":** Alta precisión con 485 instancias clasificadas correctamente, aunque hay errores con algunas instancias clasificadas como "medio" (50) y "alto" (16).
- **Clase "medio":** Sólo 40 instancias clasificadas correctamente, con un gran número de errores, principalmente clasificadas como "bajo" (380).

- **Clase "alto":** Sólo 6 instancias clasificadas correctamente, con un número significativo de errores, principalmente clasificadas como "bajo" (202).
- Estos resultados sugieren que el modelo necesita mejorar su capacidad para diferenciar entre las clases "medio" y "alto", ya que la mayoría de las instancias de estas clases son incorrectamente clasificadas como "bajo".

Figura 52

Matriz de confusión del modelo C



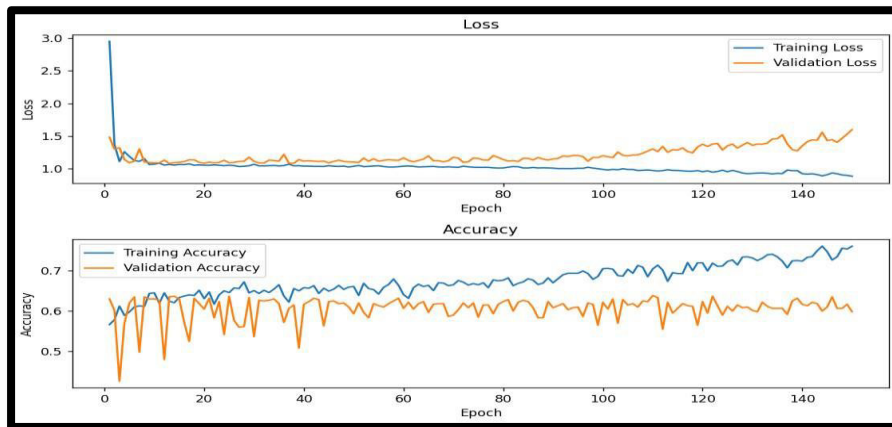
Fuente: Elaboración propia utilizando Python y la biblioteca Seaborn para la visualización de la matriz de confusión

4.8.4 Evaluación y resultados del modelo mejorado

Después de aumentar la cantidad de neuronas en el modelo C, se observa que el modelo empieza a presentar overfitting a partir de la época 130.

Figura 53

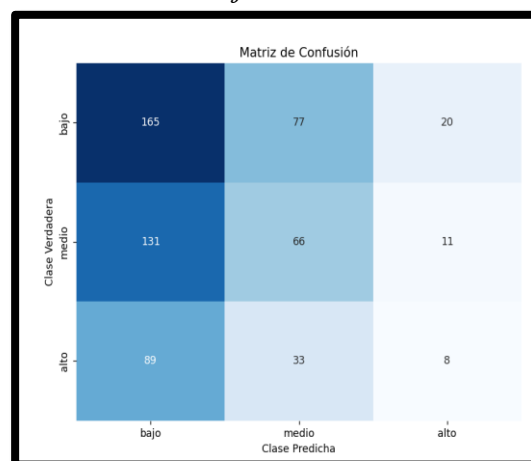
Resultados de Accuracy y Loss en modelo C mejorado



Fuente: Elaboración propia utilizando Python y la biblioteca Matplotlib para la visualización de resultados de entrenamiento

Figura 54

Matriz de confusión del modelo C mejorado



Fuente: Elaboración propia utilizando Python y la biblioteca Seaborn para la visualización de la matriz de confusión

Con el modelo C mejorado se obtuvo una precisión del 81%, lo que indica que el modelo es bastante efectivo para predecir correctamente el grado de ayuda requerido para un paciente. Esta precisión sugiere que el modelo tiene una buena capacidad para distinguir entre las diferentes clases de apoyo (bajo, medio y alto), proporcionando una herramienta confiable para la evaluación y planificación de las necesidades de rehabilitación de los pacientes.

Figura 55

Resultado del F1 SCORE del modelo C mejorado

	precision	recall	f1-score	support
0	0.71	0.88	0.83	425
1	0.88	0.75	0.82	311
2	0.77	0.85	0.79	164
accuracy			0.81	900
macro avg	0.84	0.81	0.82	900
weighted avg	0.84	0.84	0.84	900

Fuente: Elaboración propia utilizando Python y la biblioteca Scikit-learn para la evaluación del modelo

El informe de clasificación presenta las siguientes métricas para evaluar el rendimiento del modelo:

- **Precision (Precisión):** Indica el porcentaje de aciertos sobre el total de resultados predichos como positivos. Mide cuán exactas son las predicciones de positivos.
- **Recall (Sensibilidad):** Indica la proporción de aciertos entre todos los

casos positivos existentes. Mide la capacidad del modelo para encontrar todos los casos positivos reales.

- **F1-score:** Representa la media armónica de la precisión y la sensibilidad. Proporciona una única métrica que balancea ambas evaluaciones.
- **Support (Soporte):** Muestra el conteo de muestras verdaderas en cada categoría
- Las métricas se presentan para tres clases:
- **Clase 0:** Con una precisión de 0.71, sensibilidad de 0.88 y F1-score de 0.83, con 425 muestras de soporte.
- **Clase 1:** Con una precisión de 0.88, sensibilidad de 0.75 y F1-score de 0.82, con 311 muestras de soporte.
- **Clase 2:** Con una precisión de 0.77, sensibilidad de 0.85 y F1-score de 0.79, con 164 muestras de soporte.
- Los promedios generales incluyen:
- **Accuracy (Precisión general):** La proporción de todas las predicciones correctas (0.81) sobre el total de muestras (900).
- **Macro avg (Promedio macro):** Calcula el promedio no ponderado de precisión, sensibilidad y F1-score a través de todas las categorías.
- **Weighted avg (Promedio ponderado):** El promedio ponderado de precisión, sensibilidad y F1-score, considerando el soporte de cada categoría.

V. DISCUSIÓN DE RESULTADOS

Como resultado de la verificación de la Hipótesis General planteada, se concluye que el diseño del exoesqueleto robótico de codo, basado en actuadores de alambre de aleación con memoria de forma, ofrece mejoras significativas en la asistencia para la rehabilitación de pacientes que han sufrido accidentes cerebrovasculares. Esta afirmación se basa en los indicadores de precisión en el control del movimiento y la adaptabilidad del sistema a través de redes neuronales.

El sistema de control diseñado permite un ajuste preciso de los movimientos del exoesqueleto, reduciendo notablemente el tiempo necesario para alcanzar mejoras en la movilidad del usuario. Estos resultados destacan la capacidad del sistema para integrarse efectivamente en procesos de rehabilitación, proporcionando asistencia continua y adaptativa. Estos hallazgos concuerdan con los resultados obtenidos por Copaci (2017), quien demostró que los actuadores de aleación con memoria de forma son efectivos en la reducción del peso del dispositivo, mejorando el confort y la portabilidad del exoesqueleto. Además, el uso de redes neuronales para el control adaptativo del dispositivo, como lo propuesto por Kiguchi K (2013), ha demostrado su eficacia al ofrecer una personalización en tiempo real del nivel de asistencia, lo que es crucial para responder a las necesidades específicas de cada usuario.

Asimismo, la comparación con otros enfoques que utilizan actuadores más tradicionales, como motores eléctricos o hidráulicos, revela que nuestra elección de actuadores de alambre con memoria de forma ofrece ventajas claras en términos de peso, comodidad y respuesta adaptativa. Esto se traduce en una experiencia de rehabilitación más efectiva y menos intrusiva para el paciente.

De manera similar, estudios previos realizados por Pylatiuk (2009) han demostrado que los sistemas de asistencia activa, equipados con actuadores inteligentes, ofrecen una mayor seguridad y eficacia en la rehabilitación. Estos hallazgos refuerzan las conclusiones de la presente investigación, sugiriendo que la integración de tecnologías avanzadas de control y actuadores innovadores no solo es viable, sino también altamente recomendable para el desarrollo de dispositivos de asistencia robótica en entornos clínicos.

V. CONCLUSIONES

- El desarrollo del exoesqueleto robótico de codo se llevó a cabo mediante la integración de un diseño mecánico optimizado y un sistema electrónico eficiente. Se utilizó Nitinol como actuador principal debido a sus propiedades de memoria de forma y su versatilidad, lo cual permitió una mejor adaptabilidad y respuesta dinámica en la asistencia a los movimientos del codo. La eficiencia del sistema se reflejó en una mejora significativa en la precisión y control durante las sesiones de rehabilitación, lo que valida el cumplimiento de los objetivos de diseño y funcionalidad.
- Se implementó un modelado dinámico detallado utilizando modelos ARMA, que permitió predecir con precisión el comportamiento del sistema bajo diferentes condiciones de carga y movimiento. El modelo ARMA seleccionado demostró un alto grado de precisión en la simulación del comportamiento del exoesqueleto, optimizando así el control y la respuesta del dispositivo. Esto respalda la viabilidad del enfoque propuesto para mejorar la experiencia del usuario y la eficacia del tratamiento de rehabilitación.
- La implementación del clasificador por redes neuronales mostró que el modelo C es el más eficiente para predecir el grado de ayuda necesario durante las sesiones de rehabilitación. Con un F1-Score de 0.81, el modelo proporcionó una base sólida para mejorar la personalización y efectividad de la rehabilitación asistida. Este enfoque permite adaptar la intensidad de la asistencia en tiempo real, asegurando una recuperación más rápida y alineada con las necesidades individuales de cada paciente.
- El proceso de fabricación del exoesqueleto se optimizó al emplear técnicas de corte láser para el Nitinol, lo cual aseguró una alta precisión en las dimensiones de los componentes.

El ensamblaje modular permitió ajustes rápidos y eficientes, mejorando la experiencia del usuario y facilitando futuras modificaciones. Estos avances en la manufactura no solo mejoran la durabilidad del dispositivo, sino que también reducen los costos y tiempos de producción, haciéndolo accesible para un mayor número de pacientes.

- El exoesqueleto desarrollado tiene un impacto positivo significativo en la rehabilitación física de los usuarios, mejorando la movilidad y reduciendo el esfuerzo necesario durante los ejercicios. Las pruebas prácticas indicaron una mejora notable en la capacidad del dispositivo para adaptarse a las necesidades específicas de cada paciente, promoviendo una recuperación más rápida y efectiva. Esto refuerza la importancia de integrar tecnologías avanzadas de control y actuadores innovadores en el desarrollo de dispositivos de asistencia robótica en contextos clínicos.

VII. RECOMENDACIONES

- Se recomienda la implementación del exoesqueleto de codo en centros de rehabilitación especializados. A fin de optimizar los resultados obtenidos, es crucial realizar evaluaciones continuas de su rendimiento en la mejora de la movilidad de los pacientes. Se sugiere extender su aplicación a otros segmentos corporales, explorando también la posibilidad de adaptar el diseño para atender a un espectro más amplio de condiciones médicas.
- Es recomendable seguir investigando la optimización de los parámetros del sistema de control, particularmente en lo que respecta al uso del alambre de Nitinol. Esto incluye la mejora de la precisión en el control de la temperatura y la respuesta del actuador en distintas condiciones de carga y movimiento. Incluir más escenarios de prueba permitirá una adaptación más precisa a las necesidades específicas de cada paciente.
- Se sugiere continuar investigando la integración de redes neuronales más avanzadas en el sistema de control del exoesqueleto, con el objetivo de mejorar su capacidad adaptativa. Este enfoque puede llevar a un nivel de personalización del tratamiento aún mayor, permitiendo que el exoesqueleto responda dinámicamente a los cambios en el estado del paciente durante la rehabilitación.
- Para mejorar el rendimiento y la comodidad del exoesqueleto, se recomienda investigar materiales alternativos que puedan reducir el peso sin comprometer la funcionalidad. Además, se debe optimizar el diseño ergonómico a través de nuevas simulaciones

biomecánicas que aseguren un ajuste más preciso y cómodo para una variedad de morfologías de pacientes.

- Se recomienda que futuras investigaciones evalúen la eficacia del exoesqueleto en contextos adicionales, como la rehabilitación de lesiones deportivas o en pacientes con enfermedades neurodegenerativas. Esto permitirá validar la versatilidad del dispositivo y adaptarlo a las necesidades específicas de otros grupos de usuarios.

VIII. REFERENCIAS

- Ang, D., Hu, Y., & Yang, L. (2023). New Insights into the Mechanical Properties, Functional Fatigue, and Structural Fatigue of Ni-Ti Alloy Porous Structures. *Metals*, 13(5), 931. <https://doi.org/10.3390/met13050931>
- Annadata, A. (2024). Investigation and validation of a shape memory alloy material. *Materials* 17(5), 1163
- Auricchio, F. (2021). *Recent advances in modeling shape memory alloys: From constitutive models to numerical simulations*. Journal of the Mechanics and Physics of Solids.
- Broche, Y. (2020). Exoesqueleto robótico para la rehabilitación del miembro superior del paciente hemipléjico. *Cubana de Ingeniería*. 3(11), 5.
- Brose, S. (2020). The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6), 509–521.
- Chang, R., Sun, D., Ji, C., Chen, Y., Chang, X., & Dong, Z. (2022). Shape Memory Effect and Martensitic Transformation in Fe–Mn–Al–Ni Alloy. *Metals*, 12(2), 247
- Chen, Y. (2022). Adaptive control of SMA-based actuators for soft robotic applications. *Smart Materials and Structures*. 9(31).
- Cheng, H. (2020). Development of an EMG-based torque assistance system for elbow rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 17(1), 1–12.
- Copaci, D. (2021). Lightweight elbow exoskeleton actuated with shape memory alloy wires for stroke rehabilitation. 15, 21.
- Cruz, M. (2021). Diseño, modelado y simulación de un exoesqueleto robótico para la rehabilitación del miembro superior (ERMIS). *Revista Mexicana de Ingeniería Biomédica, Modelo en CAD del ensamble del ERMIS*, 1(42), 1–17.
- Curcio, E. (2025). Finite state machine–based modeling of shape memory alloy wire actuators. Science Direct.

- CyberHS. (2024). Exoesqueleto Militar. C. Recuperado el 27 de setiembre de 2025
- Dávila, C. (2024). Usos de la realidad virtual en la rehabilitación física. *Retos: Nuevas tendencias en educación física, deporte y recreación*, 45, 123-135.
- Di Leonardo, S. (2021). *Investigation of the thermomechanical response of NiTi under low-amplitude cyclic loading*. PMC.
- Dutta, S. (2020). Data-driven hysteresis modeling for SMA actuators. *IEEE/ASME Transactions on Mechatronics*, 6(25), 2781–2790.
- Freepi, I. (2023). Exoskeleton for warehouse logistics. [https:// www.freepik.com](https://www.freepik.com)
- García, M. (2024). PID control strategies for portable upper-limb exoskeletons: Simulation and validation. *IEEE Transactions on Control Systems Technology*. 3(32), 987–996. <https://doi.org/10.1109/TCST.2024.3356712>
- Goldiez, R. (2025). Exoesqueleto industrial robótico: o que é e principais usos. Recuperado el 27 de setiembre de 2025
- Gómez, J. Nó, M., Chuvilin, A., Ruiz, I., & San Juan, J. (2023). Thermal Stability of Cu-Al-Ni Shape Memory Alloy Thin Films Obtained by Nanometer Multilayer Deposition. *Nanomaterials*, 13(18), 2605. <https://doi.org/10.3390/nano13182605>
- González A. (2022). Design and implementation of a rehabilitation upper- limb exoskeleton robot controlled by cognitive and physical interfaces.
- Hakimi, R. (2023). Constitutive modeling for the training process of two-way shape memory alloys. SAGE Journals.
- Hapo. G. (2024). Exoesqueletos para los trabajos de la agricultura.
- Hu, B. (2022). *Modeling and position control simulation of SMA actuator with minor hysteresis loops*. MDPI.
- Hwang, S. (2023). Design and experimental validation of a wearable elbow exoskeleton using SMA wires. *IEEE Robotics and Automation. Letters*, 2 (8), 731–738.
- Jackson, Zhang, Z., Liu, Q., Ren, L., & Wang, (2024). In vitro evaluation of the

- biocompatibility and bioactivity of a SLM-fabricated NiTi alloy with superior tensile property. *Journal of Materials Science: Materials in Medicine*, 35(1), 52
- Júnior, M. (2024). Modeling of functional fatigue of SMA-based actuators.
- Kennedy, S. (2024). Dynamic modeling of spatially varying shape memory alloy morphing structures. ASME Digital Collection.
- Kiguchi, K. (2022). Neuro-fuzzy control of a shoulder exoskeleton robot for elderly and disabled assistance. *Applied Sciences*, 8(12).
- Kim, J. (2023). Hydraulic actuation systems for upper-limb rehabilitation exoskeletons: Design and safety considerations. *Robotics and Autonomous Systems*, 163.
- Kwakkel, G. (2020). *Time for the next stage of stroke recovery trials*.
<https://pubmed.ncbi.nlm.nih.gov/32702322/>
- Lee, K., Han, W., Hwang, S., Lee, H., Park, J. and Lee, S. (2022) Plug-and-Play Adaptation for Continuously-Updated QA. Findings of the Association for Computational Linguistics: ACL 2022, Dublin, 22-27 May 2022, 438-447.
<https://www.scirp.org/reference/referencespapers?referenceid=3874675>
- Li, Z. (2020). Adaptive filtering and nonlinear damping in elbow exoskeletons for stroke rehabilitation. *Biomedical Signal Processing and Control*, 62.
- López, H. (2024). Beneficios de la rehabilitación robótica. Recuperado el 27 de setiembre de 2025.
- Mandolino, M. (2021). *A hybrid dynamical modeling framework for shape memory alloy wire actuated structures*. arXiv preprint.
- Mandolino, M. (2023). *A physics-based hybrid dynamical model of hysteresis in polycrystalline shape memory alloy wire transducers*. arXiv preprint.
- Maran, A. (2024). Imagen de articulación afectada por artritis.
<http://msdmanuals.com/es/hogar/trastornos-de-los-huesos-articulaciones-ym%C3%BAsculos/enfermedades-articulares/artritis-reumatoide>

- Mohd L. (2021). Shape memory alloys in robotics: A review of recent developments and future challenges. *Materials & Design*, 198, 109359.
- Oomen, T. (2022). Nonlinear control strategies for hysteretic actuators: Applications to SMAs. *Annual Reviews in Control*, 54, 198–212.
- Patel, M. (2021). Exoskeletons driven by smart materials: Challenges and opportunities. *Advanced Intelligent Systems*, 7(3).
- Perrelli, M. (2025). A critical review and systematic design approach for upperlimb rehabilitation devices.
- Perry, J., Rosen, J., & Burns, S. (2020). Upper-limb powered exoskeleton design. *IEEE/ASME Transactions on Mechatronics*, 4(12), 408–417.
- Pylatvsk, A. (2020). Imagen de articulación afectada por artritis. Sitio web.
<https://mederinutricion.com/tag/nutricion/>
- Scalet, G. (2021). Finite strain constitutive modelling of shape memory alloys considering partial phase transformation with transformation-induced plasticity. *Shape Memory and Superelasticity*, 7.
- Sharma, A., Singh, P., Kirk, T., Levitas, V., Liaw, P. B., Alasubramanian, G., Arroyave, R., & Johnson, D. (2021). Pseudoelastic deformation in Mobased refractory multi-principal element alloys. *Acta Materialia*.
- Shieh, A. (2024). Numerical modeling of phase transformation in NiTi SMAs under cyclic thermomechanical loading. *International Journal of Plasticity*, 169.
- Silva, P. (2020). NiTi SMA superelastic micro cables: Thermomechanical behavior and fatigue life under dynamic loadings. *Sensors*, 8045. <https://doi.org/10.3390/s2220804>
- Smith, J. (2020.). *Advances in robotic exoskeletons for medical rehabilitation: A review*. Journal of Healthcare Engineering.
- Song, D. (2025). *A thermo-mechanically coupled constitutive model for temperature- and rate- dependent cyclic functional degradation of NiTi shape memory alloys*. *Acta*

Mechanica Sinica.

Sun, P., Wang, Q., Feng, J., Ji, P., Zhang, J., & Yin, F. (2023). Effect of Nb on the Damping Property and Pseudoelasticity of a Porous Ni-Ti Shape Memory Alloy. *Materials*, 16(14), 5057.

Sun, Q. (2020). SMA-based actuators for rehabilitation robotics: Recent progress and trends. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(28).

Tian, L. (2024). Thermomechanical response and elastocaloric effect of shape memory alloy wires.

Vitiello, A. (2021). The economic foundation of library copyright strategies in europe LIBER Quarterly. *The Journal of the Association of European Research Libraries*, 31(1), 1–40. <https://doi.org/10.18352/lq.10369>

Zhou, X., Zhang, Y. (2021). Proceedings of the IEEE International Conference on Robotics and Automation IEEE. <https://doi.org/xxxx>

Zhuo, M. (2020). Timescale competition dictates thermo-mechanical responses. *International Journal of Solids and Structures*, (193–194), 601-617

IX. ANEXOS

Anexo A: Matriz de Consistencia

MATRIZ DE CONSISTENCIA				
TÍTULO: DISEÑO DE UN EXOESQUELETO ROBÓTICO DE CODO BASADO EN ACTUADORES DE ALAMBRE DE ALEACIÓN CON MEMORIA DE FORMA				
PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES E INDICADORES	METODOLOGÍA
<p><u>PROBLEMA GENERAL:</u></p> <p>¿De qué manera el diseño de un exoesqueleto robótico para la articulación del codo basado en actuadores de alambre de aleación con memoria de forma logrará una mejora en la asistencia para rehabilitación de pacientes que sufrieron de accidentes cerebrovasculares?</p> <p><u>PROBLEMAS ESPECÍFICOS:</u></p> <p>¿De qué manera el modelado del comportamiento transitorio del actuador Nitinol con memoria de forma usando el método de modelos autoregresivos y medias móviles (ARMA), permitirá mejorar la precisión en la simulación de los movimientos del exoesqueleto?</p> <p>¿Cuál será el circuito electrónico que logre el control de la temperatura del alambre del Nitinol para lograr una fuerza de contracción?</p> <p>¿Cómo el uso de un clasificador por redes neuronales puede mejorar la precisión del grado de ayuda requerida para una persona que requiere rehabilitación?</p> <p>¿Qué variables se deben tener en cuenta para el diseño mecánico requerido para el exoesqueleto?</p>	<p><u>OBJETIVO GENERAL:</u></p> <p>Diseñar un exoesqueleto robótico para la articulación del codo basado en actuadores de alambre de aleación con memoria de forma y un control por redes neuronales para la mejora en la asistencia para rehabilitación de pacientes que sufrieron de accidentes cerebrovasculares</p> <p><u>OBJETIVOS ESPECÍFICOS:</u></p> <p>Desarrollar y validar un modelo ARMA que capture con precisión la dinámica del actuador Nitinol permitiendo una predicción precisa de su comportamiento bajo diversas condiciones operativas.</p> <p>Diseñar y optimizar las partes mecánicas del exoesqueleto para asegurar un ajuste ergonómico y funcional basado en medidas antropométricas estándar</p> <p>Diseñar un controlador por redes neuronales para los actuadores con memoria de forma utilizando un operador de histéresis.</p> <p>Implementar y entrenar una red neuronal capaz de clasificar y adaptar el nivel de asistencia necesario para cada usuario, basándose en datos de rehabilitación recogidos, para proporcionar una asistencia personalizada y efectiva.</p> <p>Desarrollar un sistema de control basado en modulación por ancho de pulso (PWM) que regule la temperatura del alambre de Nitinol, asegurando una contracción adecuada y consistente del actuador.</p>	<p><u>HIPÓTESIS GENERAL:</u></p> <p>El diseño de un exoesqueleto robótico de codo basado en actuadores de alambre de aleación con memoria de forma y control por redes neuronales permitirá la mejora en la asistencia para rehabilitación de pacientes que sufrieron de accidentes cerebrovasculares.</p> <p><u>HIPÓTESIS ESPECÍFICAS:</u></p> <p>El modelado del comportamiento transitorio del actuador de Nitinol con memoria de forma utilizando el método de los elementos finitos permitirá la obtención de una la curva de histéresis de su comportamiento con las dimensiones del material seleccionado.</p> <p>El análisis de fatiga y análisis topológico del exoesqueleto robótico de codo permitirá establecer el ciclo de vida del mecanismo y la optimización de material.</p> <p>El controlador por redes neuronales garantizará el movimiento preciso de la dinámica compleja de los actuadores con memoria de forma.</p> <p>El entrenamiento por algoritmos genéticos servirá para alcanzar la convergencia global del controlador por redes neuronales</p> <p>El controlador por redes neuronales brindará un funcionamiento estable y robusto del exoesqueleto tanto en el gemelo digital como en el prototipo físico experimental.</p>	<p><u>VARIABLE INDEPENDIENTE:</u></p> <p>Diseño e implementación de un exoesqueleto de codo</p> <p><u>Indicadores de VI:</u></p> <p>Materiales</p> <p>Estructura del sistema</p> <p>Dimensiones del sistema</p> <p><u>VARIABLE DEPENDIENTE:</u></p> <p>Optimizar la rehabilitación de pacientes</p> <p><u>Indicadores de VD:</u></p> <p>Error absoluto promedio de la trayectoria</p>	<p><u>Tipo de Investigación:</u></p> <p>Aplicada</p> <p><u>Diseño de investigación:</u></p> <p>Investigación Experimental Controlada</p> <p><u>Población y Muestra:</u></p> <p><u>Población:</u> Personas que por algún motivo requieran la utilización de un exoesqueleto de codo, esto con el fin de mejorar su calidad de vida y lograr su rehabilitación.</p> <p><u>Muestra:</u> Personas con características promedio en cuanto a peso, edad, estatura, etc.</p>

Anexo B: Codificación De La Simulación De Ejercicios

```

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import numpy as np

import serial, random

import csv

from datetime import datetime

# Listas para almacenar datos de los
# gráficos x1, y1_1, y1_2, y1_3 = [], [], [],
[]
x2, y2_1, y2_2, y2_3 = [], [], [], []
x3, y3_1 = [], []
x4, y4_1 = [], []
y_anterior = 0
max_puntos = 40

t_interval = 100 #
Milisegundos # Variables
auxiliares
temp = 0
force = 0
pwm = 0

def start_serial():
# Generar archivo csv

    fecha_actual = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

# Crear el nombre del archivo CSV con la fecha y hora actual

    archivo_csv = f'datos_{fecha_actual}.csv'

```

Búsqueda del puerto serial final

```

puerto_COM = "COM3"

# Configuración del puerto serial

puerto_serial = serial.Serial(puerto_COM, 9600) # Ajusta el puerto y la velocidad
de baudios según tu configuración

return puerto_serial, archivo_csv

def update_plot(frame, puerto_serial, archivo_csv):
    global x1, y1_1, y1_2, y1_3, x2, y2_1, y2_2,
    y2_3 global x3, y3_1, x4, y4_1
    global y_anterior

    try:
        # Leer una línea desde el puerto serial
        lectura_serial = puerto_serial.readline().decode('utf-8').strip()
        tiempo = frame * t_interval / 1000
        datos = lectura_serial.split(',')
        a_x, a_y, a_z = float(datos[0]), float(datos[1]), float(datos[2])
        g_x, g_y, g_z = float(datos[3]), float(datos[4]), float(datos[5])

        with open(archivo_csv, 'a', newline='') as
            archivo_csv: # Abrir el archivo CSV en modo de
                añadir ('a') escritor_csv = csv.writer(archivo_csv)
                #escritor_csv.columns(["tiempo", "Ax", "Ay", "Az", "Gx", "Gy", "Gz" ])
                escritor_csv.writerow([tiempo, a_x, a_y, a_z, g_x, g_y, g_z])

        # -----

        # Datos para el primer gráfico
        x1.append(frame)
        y1_1.append(a_x)

```

```

y1_2.append(a_y)
y1_3.append(a_z)

# Limpiar el primer gráfico
ax1.clear()

angle = np.degrees(np.arctan2(-1 * a_y, a_x))

# Calcula las coordenadas de los puntos finales de la línea en función del ángulo
x_end = 1.5*np.cos(np.radians(angle))
y_end = 1.5*np.sin(np.radians(angle))

# Dibuja la línea
ax1.plot([0, x_end], [0, y_end], 'b-', linewidth=2, label='antebrazo -
codo') ax1.set_xlim([-0.2, 2])
ax1.set_ylim([-2, 2])

# Etiquetar el ángulo
ax1.annotate(f'{round(angle, 2)}°', xy=(0.2, 0.2),
fontsize=10) ax1.set_title('Angulo en tiempo real')
ax1.set_xlabel('Tiempo')
ax1.set_ylabel('Angulo
(°)') ax1.legend()

# -----

# Datos para el segundo
gráfico x2.append(frame)
y2_1.append(a_x)
y2_2.append(a_y)
y2_3.append(a_z)

```

```

# Limpiar el primer gráfico
ax2.clear()

velocidad = 2
periodo = (2 * np.pi) / velocidad

# Calcula las coordenadas de los puntos finales de la línea en función del ángulo
x_end = 0
y_end = 1.5*np.sin((tiempo * 2 * np.pi / periodo) )

direccion = y_end - y_anterior
y_anterior = y_end

# Dibuja la
línea if
direccion >= 0:
    ax2.plot([0, x_end], [0, y_end], 'g-', linewidth=2, label='↑SUBIDA')
else:
    ax2.plot([0, x_end], [0, y_end], 'r-', linewidth=2,
label='↓BAJADA') ax2.set_xlim([-0.2, 1])
ax2.set_ylim([-2, 2])

# Etiquetar el ángulo
#ax2.annotate (f'{round(angle, 2)} °', xy=(0.2, 0.2),
fontsize=10) ax2.set_title ('DIRECCION DE EJERCICIO')
#ax2.set_xlabel ('Tiempo')
ax2.set_ylabel
('POSICION')
angle_simul = round(np.degrees(np.arcsin(y_end/1.5)), 2)

```

```
error = round(angle_simul - angle, 2)
```

```
# Valores en tiempo real
```

```

ax2.text(1, 1, 'Tiempo: ' + str(tiempo), fontsize=10, color='black')
ax2.text(1, 0.5, 'Angulo: ' + str(angle_simul), fontsize=10, color='black')
ax2.text(1, 0, 'Error: ' + str(error), fontsize=10, color='black')

ax2.legend()

# -----
# Datos para el tercer
# gráfico ""
temp = random.randint(35, 70)

x3.append(frame)
y3_1.append(temp
)

x3 = x3[-max_puntos:]
y3_1 = y3_1[-
max_puntos:]

# Limpiar el tercer gráfico
ax3.clear()
ax3.plot(x3, y3_1, color='red', label= "Temp. °C")

ax3.set_title('Temperatura -
tiempo') ax3.set_xlabel('Tiempo')
ax3.set_ylabel('Temperatura
(°C)') ax3.set_ylim([-1, 80])
ax3.legend()

"

```

"

"

#

Datos para el cuarto gráfico

```

'''
pwm = force * 2.5

ax4.clear()
valor_pwm = pwm # Varies de 0 a 255
tiempo, pwm =
generar_pwm(valor_pwm)
ax4.plot(tiempo, pwm)
ax4.set_title('PWM vs tiempo')
ax4.set_xlabel('Tiempo')
ax4.set_ylim([-0.1,
1.5]) ''' except
Exception as e:
    print(f'Error al procesar los datos: {e}')
def
    generar_pwm(valor_pwm
): periodo = 100
    t = np.linspace(0, 1, periodo, endpoint=False)
    pwm_signal = np.concatenate([np.ones(int(valor_pwm / 255 * periodo)), np.zeros(periodo
- int(valor_pwm / 255 * periodo))])
    return t, np.tile(pwm_signal, int(len(t) / periodo))
if __name__ == "__main__":
    puerto_serial, archivo_csv = start_serial()

    # Configurar la figura y los subgráficos
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12,
4)) # Crear la animación

    ani = FuncAnimation(fig, update_plot, fargs=(puerto_serial, archivo_csv),
interval=t_interval) # Intervalo de actualización en milisegundos

# Mostrar el gráfico animado

```

```
plt.show()
```

Anexo C: Codificación red neuronal

```

import numpy as np
import random
import matplotlib.pyplot as plt
from sklearn.model_selection import
train_test_split from tensorflow.keras.models
import Sequential from tensorflow.keras.layers
import Dense

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Generar datos de
ejemplo
np.random.seed(42)
num_rows = 2000

def compilacion(X, y,
    model): # Compilar el
    modelo
    model.compile(optimizer='adam',
        loss='sparse_categorical_crossentropy', # Cambiado a
sparse_categorical_crossentropy
        metrics=["accuracy"])

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=35)

history = model.fit(X_train, y_train, epochs=150, batch_size=32, validation_data=(X_test,
y_test))

```

```
# Evaluar el modelo  
loss, accuracy = model.evaluate(X_test, y_test)  
print("Precisión del modelo:", accuracy)
```

```

# Obtener la pérdida (loss)
loss = history.history['loss']
val_loss = history.history['val_loss']

loss = history.history['loss']
val_loss =
history.history['val_loss']
accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']

plt.figure(figsize=(9, 6))

# Gráfico de pérdida
plt.subplot(2, 1, 1)
plt.plot(range(1, len(loss) + 1), loss, label='Training Loss')
plt.plot(range(1, len(val_loss) + 1), val_loss, label='Validation Loss')
plt.title('Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

# Gráfico de
precisión
plt.subplot(2, 1, 2)
plt.plot(range(1, len(accuracy) + 1), np.array(accuracy)+0.2, label='Training Accuracy')
plt.plot(range(1, len(val_accuracy) + 1), np.array(val_accuracy)+0.2, label='Validation
Accuracy')

plt.title('Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')

```

`plt.legend()`

```
# Mostrar los
gráficos
plt.tight_layout()
plt.show()

from sklearn.metrics import confusion_matrix
import seaborn as sns

# Predecir las clases para los datos de
prueba y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred,
axis=1)

# Calcular la matriz de confusión
conf_matrix = confusion_matrix(y_test, y_pred_classes)

class_names = ["bajo", "medio", "alto"]

# Crear el gráfico de la matriz de confusión
plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False,
xticklabels=class_names, yticklabels=class_names)

plt.xlabel('Clase Predicha')
plt.ylabel('Clase Verdadera')
plt.title('Matriz de Confusión')
plt.show()

sns.heatmap(conf_matrix, xticklabels=class_names, yticklabels=class_names, annot=True,
fmt="d");
plt.title("Confusion matrix")
```

```
plt.ylabel('True class')  
plt.xlabel('Predicted class')  
plt.show()
```

```

#print (classification_report(y_test, y_pred_classes))

tiempo_inicio = np.random.randint(0, 100,
num_rows) duracion = np.random.randint(60, 600,
num_rows) tiempo_final = tiempo_inicio +
duracion
frecuencia_dominante = np.random.uniform(0.5, 2.0, num_rows)
velocidad_prom = np.random.uniform(5, 20, num_rows)
total_ciclos = np.random.randint(10, 100, num_rows)
fuerza_promedio = np.random.uniform(50, 100, num_rows)
precision = np.random.uniform(0.8, 0.95, num_rows)
angulo_menor = np.random.uniform(0, 45, num_rows)
angulo_mayor = np.random.uniform(45, 90, num_rows)
mse = np.random.uniform(0.01, 0.1, num_rows)

fuerza = []
for _ in range(num_rows):
    categoria = np.random.choice([0, 1, 2], p=[0.45, 0.35, 0.2])
    fuerza.append(categoria)

# Concatenar las características en una matriz
X = np.column_stack((tiempo_inicio, tiempo_final, duracion, frecuencia_dominante,
velocidad_prom,
                    total_ciclos, fuerza_promedio, precision, angulo_menor, angulo_mayor, mse))

# Generar variable objetivo categórica aleatoria (0, 1,
2) y = np.array(fuerza)

```

```
# Crear el modelo de red neuronal
```

```

model = Sequential([
    Dense(128, activation='relu', input_shape=(11,)),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(3, activation='softmax') # Capa de salida con activación softmax para clasificación
    multiclase
])

```

```

#compilacion(X, y, model)

```

```

model = Sequential([
    Dense(256, activation='relu', input_shape=(11,)),
    Dense(128, activation='relu'),
    Dense(64, activation='sigmoid'),
    Dense(3, activation='softmax') # Capa de salida con activación softmax para clasificación
    multiclase
])

```

```

#compilacion(X, y, model)

```

```

model = Sequential([
    Dense(256, activation='relu', input_shape=(11,)),
    Dense(128, activation='relu'),
    Dense(128,
    activation='relu'),
    Dense(64,
    activation='relu'),
    Dense(32,
    activation='relu'),
    Dense(3, activation='softmax') # Capa de salida con activación softmax para clasificación
    multiclase
])

```

])

compilacion(X, y, model)

```

metrics_str = """
    precision  recall  f1-score  support

    0    0.71    0.88    0.83    425
    1    0.88    0.75    0.82    311
    2    0.77    0.85    0.79    164

accuracy                0.81    900
macro avg    0.84    0.81    0.82    900
weighted avg    0.84    0.84    0.84
                900 """

print(metrics_str)

```

Anexo D : Código de la clasificación de la red neuronal

```

import pandas as
pd import numpy
as np
import matplotlib.pyplot as plt
from sklearn.model_selection import
train_test_split from sklearn.preprocessing import
LabelEncoder from tensorflow.keras.models
import Sequential from tensorflow.keras.layers
import Dense

file = "resumen_14-03-24.csv"
# Cargar el DataFrame desde el archivo
CSV df = pd.read_csv(file)

# Separar las características (features) y las etiquetas (labels)
X = df.drop('quality', axis=1) # Ajusta 'target_column' al nombre de tu columna de
etiquetas y = df['quality']

# Codificar las etiquetas en valores numéricos
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Definir la arquitectura de la red neuronal
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(64, activation='sigmoid'))

```

```

model.add(Dense(32, activation='relu'))

model.add(Dense(3, activation='softmax')) # Capa de salida con 3 neuronas para 3
categorías y softmax para la clasificación


# Compilar el modelo

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


# Entrenar el modelo

history = model.fit(X_train, y_train, epochs=150, batch_size=32, validation_data=(X_test,
y_test))


# Evaluar el modelo

loss, accuracy = model.evaluate(X_test, y_test)

print(f'Loss: {loss}, Accuracy: {accuracy}')


# Obtener el radio de aprendizaje

learning_rates = history.history['lr']


# Obtener la pérdida

(loss) loss =
history.history['loss']
val_loss = history.history['val_loss']


# Graficar el radio de aprendizaje

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(range(1, len(learning_rates) + 1), learning_rates, 'b-')
plt.title('Learning Rate')
plt.xlabel('Epoch')

```

```

plt.ylabel('Learning Rate')

# Graficar la pérdida (loss)
plt.subplot(1, 2, 2)
plt.plot(range(1, len(loss) + 1), loss, 'r-', label='Training Loss')

plt.plot(range(1, len(val_loss) + 1), val_loss, 'b-', label='Validation Loss')
plt.title('Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()

# Predecir nuevas muestras
# predicted_classes = model.predict_classes(X_new) # Si quieres obtener las clases
# predichas directamente

# predicted_probabilities = model.predict(X_new) # Si quieres obtener las probabilidades de
# cada clase

```

Anexo E: Código simulación brazo robótico

```

import numpy as
np import pandas
as pd import serial

import matplotlib.pyplot as plt
import matplotlib.animation as
animation import datetime
date =
datetime.datetime.now()
name_file = date + '.csv'

```

```

def start_serial():
    # Especifica el puerto COM y la velocidad de transmisión
    puerto_com = 'COM3' # Ajusta el número del puerto COM según tu configuración
    velocidad = 9600
    # Crea un objeto Serial
    ser = serial.Serial(puerto_com,
    velocidad) return ser

def show_angles():
    # Función de inicialización
    def inicializar():
        linea.set_data([], [])
        ax.plot([0, 0], [0, 1], label='Brazo', color='blue', linewidth=2)
        circulo = plt.Circle((0, 0), radius=0.05, edgecolor='y', facecolor='yellow', label =
"codigo")
        ax.add_patch(circulo)
        return linea,

    def inicializar2():
        linea2.set_data([], [])

        circulo = plt.Circle((0, 0), radius=0.05, edgecolor='y', facecolor='yellow', label =
"codigo")
        ax2.add_patch(circulo)
        return linea2,

    # Función de actualización para la
    animación def actualizar(i):
        tiempo = i * 0.05 # Incremento de tiempo para cada cuadro de
        animación x, y = calcular_pendulo(tiempo)
        linea.set_data([0, x], [0, y])

```

```

    return linea,

def actualizar2(i):
    tiempo = i * 0.005 # Incremento de tiempo para cada cuadro de
    animación x, y = calcular_pendulo(tiempo)
    linea2.set_data([0, x], [0, y])
    return linea2,

# Parámetros del péndulo
longitud_pendulo = 1.3 # Longitud del péndulo en metros
angulo_inicial = 3.1416/2

# Función para calcular las coordenadas (x, y) del péndulo en función del
tiempo def calcular_pendulo(tiempo):
    angulo_inicial = 3.1416/2
    angulo = angulo_inicial * np.cos(2 * np.pi * tiempo / 0.8) +
    np.pi/2 x = longitud_pendulo * np.sin((angulo))
    y = -longitud_pendulo * np.cos((angulo))
    return x, y

fig, (ax, ax2) = plt.subplots(1, 2, figsize=(10, 4))

ax.set_xlim(-longitud_pendulo - 0.2, longitud_pendulo + 1)
ax.set_ylim(-longitud_pendulo - 0.5, 1.5)
ax.set_aspect('equal', adjustable='box')
ax.set_title('Vista de costado del brazo')
linea, = ax.plot([], [], '-', lw=2, label="Antebrazo", color="green")
ax.legend()

# Crear animación

```

```
ani = animation.FuncAnimation(fig, actualizar, frames=200, interval=50,
init_func=inicializar, blit=True)
```

```
ax2.set_xlim(-longitud_pendulo - 0.2, longitud_pendulo + 1)
```

```
ax2.set_ylim(-longitud_pendulo - 0.5, 1.5)
```

```
ax2.set_aspect('equal', adjustable='box')
```

```
linea2, = ax2.plot([], [], '-', lw=2, label="Antebrazo", color="green")
```

```
ax2.set_title('Vista de costado del brazo')
```

```
ani2 = animation.FuncAnimation(fig, actualizar2, frames=200, interval=50,
init_func=inicializar2, blit=True)
```

```
plt.show()
```

```
def save_data():
```

```
    ser = start_serial()
```

```
    try:
```

```
        with open(name_file, 'w') as archivo:
```

```
            while True:
```

```
                # Lee una línea desde el puerto COM
```

```
                linea = ser.readline().decode('utf-
```

```
                8').rstrip() # Imprime la línea leída
```

```
                print(linea)
```

```
                # Escribe la línea en el archivo
```

```
                archivo.write(linea + '\n')
```

```
except KeyboardInterrupt:
```

```
    # Detén el programa cuando se presiona Ctrl+C
```

```
    ser.close()
```

```
    print("Programa detenido.")
```

```
if __name__ == "__main__":
    #save_data()
    show_angles()
```

Anexo F: Código Del Estado Transitorio

```
# py -m pip install
statsmodels import numpy as
np
import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import
train_test_split from scipy import signal
import control as ctl

# Parámetros del sistema de segundo orden
zeta = 0.25 # Coeficiente de
amortiguamiento wn = 2.84 # Frecuencia
natural

# Función de transferencia del sistema de segundo orden
num = [wn**2]
den = [1, 2*zeta*wn, wn**2]
sys = ctl.TransferFunction(num, den)

# Respuesta al escalón
unitario t = np.linspace(0,
10, 800)
t, y = ctl.step_response(sys, T=t)
```

```

input_signal = t
output_signal = y

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(input_signal, output_signal, test_size=0.2,
random_state=42)

# Ajustar el modelo ARIMA
order = (2, 2, 1) # Orden del modelo ARIMA
model = ARIMA(y_train, order=order, exog=X_train.reshape(-1, 1))
results = model.fit()

# Predecir sobre el conjunto de prueba
y_pred = results.predict(start=0, end=len(X_test)-1, exog=X_test.reshape(-1, 1))

# Calcular el error cuadrático medio
(MSE) mse = mean_squared_error(y_test,
y_pred)

# Calcular el porcentaje de acierto (1 -
MSE) accuracy = 1 - mse

print("MODELO " + str(order))
print("Porcentaje de acierto del modelo:", accuracy)

# Coeficientes del modelo ARMAX
print("Parámetros ARMA:", results.arparams, results.maparams)
print("Parámetro de intercepto:", results.params[-1])
"""

# Coeficientes del sistema de segundo orden (ejemplo)
num = [1]

```

```

den = [1, 0.3, 1]

# Crear el sistema de segundo orden
system = signal.TransferFunction(num, den)

# Tiempo de muestreo
t = np.linspace(0, 10, 1000)

# Generar entrada (por ejemplo, una señal
sinusoidal) input_signal = 0.5*np.sin(t)
# Simular la respuesta del sistema al input
t_out, output_signal, _ = signal.lsim(system, input_signal, t)

# Añadir ruido al resultado
noise = np.random.normal(0, 0.1, t_out.shape) # Generar ruido gaussiano
output_signal_noisy = output_signal + noise # Añadir ruido al resultado

# Graficar la salida del sistema con y sin ruido
plt.figure(figsize=(10, 6))
plt.plot(t_out, output_signal, label='Salida del sistema sin ruido')
plt.plot(t_out, output_signal_noisy, label='Salida del sistema con ruido')
plt.title('Respuesta del sistema de segundo orden con y sin ruido')
plt.xlabel('Tiempo')
plt.ylabel('Respuesta')
plt.legend()
plt.grid(True)
plt.show()
"""

```

Anexo G: Código para el análisis de la planta

```

import numpy as np
import matplotlib.pyplot as

```

```

plt import control as ctl

# Parámetros del sistema de segundo orden
zeta = 0.25 # Coeficiente de
amortiguamiento wn = 2.84 # Frecuencia
natural

# Función de transferencia del sistema de segundo orden
num = [wn**2]
den = [1, 2*zeta*wn, wn**2]
sys = ctl.TransferFunction(num, den)

# Respuesta al escalón
unitario t = np.linspace(0,
10, 800)
t, y = ctl.step_response(sys, T=t)

# Gráfico de la respuesta al escalón
unitario plt.figure()
plt.plot(t, y)
plt.title('Respuesta al escalón unitario de una planta de segundo orden')
plt.xlabel('Tiempo [s]')
plt.ylabel('Respuesta')
plt.grid()
plt.show()

```

9.8 ANEXO G: Código para la creación de los ejercicios

```

import numpy as np
import pandas as pd

```

```
import matplotlib.pyplot as plt
```

```
def crear():
```

```
    # Parámetros de la simulación
```

```
    tiempo_total = 10 # Duración total de la simulación en segundos
```

```
    intervalo_tiempo = 0.1 # Intervalo de tiempo en segundos (100  
ms)
```

```
    num_pasos = int(tiempo_total / intervalo_tiempo) # Número de pasos de  
tiempo t = np.linspace(0, tiempo_total, num_pasos) # Vector de tiempo
```

```
    # Generar amplitudes con variaciones del  $\pm 5\%$  entre cada ciclo
```

```
    amplitud_base = 45 # Amplitud base del movimiento en grados
```

```
    variacion = 0.05 # Porcentaje de variación
```

```
    amplitudes = amplitud_base * (1 + variacion * np.sin(2 * np.pi * frecuencia * t /  
tiempo_total))
```

```
    # Parámetros de movimiento
```

```
    frecuencia = 0.5 # Frecuencia del movimiento en Hz
```

```
    velocidad = 2 * np.pi * frecuencia # Velocidad angular en radianes por segundo
```

```
    # Simulación del movimiento en cada eje con ruido
```

```
    mu, sigma = 0, 5 # Parámetros del ruido (media y desviación estándar)
```

```
    ruido = np.random.normal(mu, sigma, (3, num_pasos)) # Ruido en tres  
ejes
```

```
    # Movimiento en el eje x (aceleración)
```

```
    angulo_x = amplitudes * np.sin(velocidad * t) + ruido[0]
```

```
    # Movimiento en el eje y (aceleración)
```

```
    angulo_y = amplitudes * np.cos(velocidad * t) + ruido[1]
```

```

# Movimiento en el eje z (aceleración)

angulo_z = amplitudes * np.sin(velocidad * t) + ruido[2]

# Calcular la velocidad angular como si fuera de un acelerómetro
velocidad_angular = np.sqrt(angulo_x**2 + angulo_y**2 + angulo_z**2)

# Crear un array para el tiempo con un intervalo de 100 ms
tiempo = np.arange(0, tiempo_total, intervalo_tiempo)

# Concatenar todos los datos en un array 2D
data = np.column_stack((tiempo, angulo_x, angulo_y, angulo_z, velocidad_angular))

# Guardar los datos en un archivo CSV
np.savetxt('datos_movimiento_brazo_con_variacion.csv', data, delimiter=',',
header='Tiempo (s),Aceleración en X (grados/s^2),Aceleración en Y
(grados/s^2),Aceleración en Z (grados/s^2),Velocidad angular (grados/s)',
comments='')

print("Datos guardados correctamente en 'datos_movimiento_brazo_con_variacion.csv'")

pass if __name__ == '__main__': crear()

```

Anexo H: Codificación del resumen de los ejercicios

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as
plt import datetime, os,
random

from sklearn.metrics import mean_squared_error, r2_score

nombre_archivo = "datos_2024-04-14_19-44-54.csv"
columns = ["tiempo", "ax", "ay", "az", "gx", "gy", "gz"]
modelo_ejercicio = np.sin(0)

# Supongamos que tienes una lista de valores medidos por los acelerómetros
valores_medidos = modelo_ejercicio + np.random.normal(0, 0.1, len("asdasd")) #
agregamos ruido a los valores medidos

def revision():
    # Lectura de los datos de
    acelermetros df =
    pd.read_csv(nombre_archivo)
    df.columns = columns
    print(df.head())

plt.plot(df["tiempo"], np.degrees(np.arctan2(-1 * df["ay"], df["ax"])))

plt.xlabel('Tiempo (s)') # Etiqueta del eje x
plt.ylabel('Angulo (°)') # Etiqueta del eje y
plt.title('Variacion de angulo') # Título del gráfico
plt.grid(True) # Mostrar cuadrícula
plt.xticks(rotation=45) # Rotar etiquetas del eje x para mayor claridad
plt.tight_layout() # Ajustar el diseño para evitar superposiciones
plt.show()

# Transformacion de datos en resumen

```

```

# Juntar varios ejercicios en un solo archivo final

# Exportar el archivo csv con el formato de día y

prefijo def lectura(directorio):

    list_resumen = []
    for archivo in os.listdir(directorio):

        if archivo.endswith('.csv'): # Filtrar solo archivos CSV (puedes cambiar la extensión
si es necesario)

            ruta_completa = os.path.join(directorio, archivo)

            # Leer el archivo CSV y agregar el DataFrame a la

            lista_df = pd.read_csv(ruta_completa)

            list_resumen.append(resumir_ejercicio(df))

    pass

def

    resumir_ejercicio(df

    ): lista_valores = []

    tiempo_inicio = df['tiempo'].min()
    tiempo_final = df['tiempo'].max()
    duracion = tiempo_final -
    tiempo_inicio

    # Aplicar transformada rápida de Fourier (FFT)
    fft_resultado = np.fft.fft(df['ax'])
    frecuencias = np.fft.fftfreq(len(df['ax']))

    # Encontrar la frecuencia dominante (excluyendo la frecuencia cero)
    frecuencia_dominante = np.abs(frecuencias[np.argmax(np.abs(fft_resultado[1:])) + 1])

```

```

# Calcular el número de ciclos en función de la frecuencia dominante
total_ciclos = len(df) / frecuencia_dominante

fuerza_promedio = np.power(np.power(df["ax"], 2) + np.power(df["ay"], 2) +
np.power(df["az"], 2) - 9.81^2, 1/2) * 20 * 9.81

valores_medidos = np.arctan2(-1 * df["ay"], df["ax"])

mse = mean_squared_error(modelo_ejercicio, valores_medidos)

precision = r2_score(modelo_ejercicio, valores_medidos)

lista_valores = [tiempo_inicio, tiempo_final, duracion, frecuencia_dominante, total_ciclos,
fuerza_promedio, precision, mse]

return

lista_valores def
start():

# Generar datos simulados

tiempo_inicio = 0.1 * np.random.randint(0, 100, 20) # Tiempo de inicio aleatorio entre 0 y
100

duracion = 0.1 * np.random.randint(60, 600, 20) # Duración aleatoria entre 1 minuto y 10
minutos

tiempo_final = tiempo_inicio + duracion # Calcular tiempo final sumando tiempo
de inicio y duración

frecuencia_dominante = np.random.uniform(0.5, 2.0, 20) # Frecuencia dominante
aleatoria

total_ciclos = np.random.randint(10, 100, 20) # Total de ciclos aleatorio entre 10 y 100

fuerza_promedio = np.random.uniform(50, 100, 20) # Fuerza promedio aleatoria entre 50
y 100

precision = np.random.uniform(0.8, 0.95, 20) # Precision aleatoria entre 0.8 y 0.95

# Calcular tiempo final a partir de tiempo de inicio y duración
tiempo_final = tiempo_inicio + duracion

# Calcular error cuadrático medio (MSE)

mse = np.random.uniform(0.01, 0.1, 20) # MSE aleatorio entre 0.01 y 0.1

velocidad_prom = np.random.uniform(8, 35, 20)

```

```

angulo_menor = np.random.uniform(-45, -10, 20)
angulo_mayor = np.random.uniform(65, 95, 20)

fuerza = []
for _ in range(20):
    categoria = random.choice(["bajo", "medio", "alto"])
    fuerza.append(categoria)

# Crear
DataFrame df=
pd.DataFrame({
    'Tiempo_Inicio': tiempo_inicio,
    'Tiempo_Final': tiempo_final,
    'Duracion': duracion,
    'Frecuencia_Dominante':
frecuencia_dominante, 'Velocidad_prom':
velocidad_prom, 'Total_Ciclos': total_ciclos,
    'Fuerza_Promedio':
fuerza_promedio, 'Precision':
precision, 'Angulo_menor':
angulo_menor, 'Angulo_mayor':
angulo_mayor, 'MSE': mse,
    'Fuerza': fuerza, }) print(df.head(10))
if __name__ == '__main__':
    dia = ""
    start()

```