



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería Electrónica Industrial y Automática**

**Desarrollo de un Algoritmo en Tiempo Real  
para la Detección de Gestos de Mano y  
Muñeca mediante Técnicas de  
Aprendizaje Automático**

**Autor:**

**Gómez Sánchez, Pablo**

**Tutor(es):**

**Fraile Marinero, Juan Carlos**

**Ingeniería de Sistemas y Automática**

**Cisnal de la Rica, Ana**

**Ingeniería de Sistemas y Automática**



## **Agradecimientos**

Me gustaría agradecer a mis tutores Juan Carlos Fraile y Ana Cisnal por darme la oportunidad de llevar a cabo este proyecto y por toda su dedicación y ayuda, así como a otros compañeros de ITAP que también ha dedicado su tiempo a que completará este proyecto.

Finalmente, quería agradecerse a mi familia por acompañarme durante esta etapa.



## Resumen

La electromiografía (EMG) es una técnica de registro de la actividad eléctrica producida por los músculos esqueléticos mediante electrodos intramusculares o superficiales. Este TFG aborda el diseño e implementación de reconocimiento de gestos de la mano basado en el procesamiento de señales EMG y de orientación obtenidas del sensor Trigno Duo de Delsys. Para la diferenciación de movimientos se han utilizado modelos de aprendizaje máquina o *Machine Learning* como *Random Forest*, *KNN*, *Linear SVC*, *Logistic Regression* o *Gradient Boosting*. Se han entrenado los modelos mediante la validación cruzada anidada para la selección de hiperparámetros junto a un esquema LOSO para evaluar la capacidad de generalización intersujeto. Los clasificadores lineales, *Linear SVC* y *Logistic Regression* obtuvieron los mejores resultados con 75% de precisión.

## Palabras clave:

Aprendizaje automático, Electromiografía, Orientación, Validación cruzada anidada, LOSO

## Abstract

Electromyography (EMG) is a technique used to record the electrical activity produced by skeletal muscles using intramuscular or surface electrodes. This Final Degree Project focuses on the design and implementation of a hand gesture recognition system based on processing of EMG signals and orientations data acquired from the Trigno Duo Sensor by Delsys. To distinguish between different hand movements, various machine learning models were employed, included Random Forest, Linear SVC, Logistic Regression and Gradient Boosting. The models were trained using nested cross-validation for hyperparameter tuning, combined with a Leave-One-Subject-Out (LOSO) scheme to asses inter-subject generalization. Linear classifiers, specifically Linear SVC and Logistic Regression, achieved the best performance with an accuracy of 75%

## Key words

Machine Learning, Electromyography, Orientation, Nested cross-validation, LOSO



# Índice

---

<b>1</b>	<b>Introducción.....</b>	<b>1</b>
1.1	Electromiografía.....	2
1.2	Estado del arte .....	3
1.3	Objetivos del proyecto.....	12
<b>2</b>	<b>Hardware empleado.....</b>	<b>15</b>
<b>3</b>	<b>Metodología del proyecto.....</b>	<b>19</b>
3.1	Extracción de datos del sensor .....	21
3.2	Filtrado de señales .....	22
3.3	Recogida de datos.....	24
3.4	Extracción de características de las señales sEMG y orientación.....	27
3.5	Selección de características .....	34
3.6	Selección del mejor Modelo de <i>Machine Learning</i> .....	38
<b>4</b>	<b>Presentación de resultados .....</b>	<b>51</b>
4.1	Selección de características .....	51
4.2	Selección del tamaño de ventana.....	56
4.3	Selección del mejor modelo .....	58
<b>5</b>	<b>Discusión de los resultados.....</b>	<b>63</b>
<b>6</b>	<b>Conclusiones .....</b>	<b>65</b>
<b>7</b>	<b>Referencias .....</b>	<b>67</b>





## Índice de figuras

---

Figura 1. Unidad motora (UM) [2].	2
Figura 2. Señal EMG detectada en el antebrazo durante la contracción de la mano (A), reposo (B) y la extensión de la mano (C).	3
Figura 3. Myo Armband [6].	5
Figura 4. Los primeros pasos para la clasificación de movimientos de la mano utilizando señales sEMG [11].	9
Figura 5. Prótesis de mano [6].	10
Figura 6. Sensor Trigno Duo Lite de Delsys [9].	15
Figura 7. Referencia señales sEMG y IMU, Sensor Trigno [9].	16
Figura 8. Dispositivo Delsys Trigno Duo Lite.	16
Figura 9. Base de carga del sistema Delsys Trigno Lite.	17
Figura 10. Diagrama de bloques realizado para este trabajo.	20
Figura 11. Filtrado de las señales EMG.	24
Figura 12. Interfaz gráfica diseñada para visualizar las señales sEMG y de orientación.	25
Figura 13. Músculo flexor superficial de los dedos (A) y músculo extensor de los dedos (B).	26
Figura 14. Colocación del sensor para la recogida de datos.	27
Figura 15. Comparación de las técnicas de ventanización disjunta y superpuesta [6].	29
Figura 16. Logo de la librería TSFEL [21].	30
Figura 17 Gráfica de los métodos de envoltura [24].	36
Figura 18. Diagrama conceptual de un modelo KNN [34].	39
Figura 19. Diagrama conceptual de un modelo Random Forest [36].	40
Figura 20. Una visualización del esquema LOSO CV y CV en un escenario de diagnóstico [45].	47
Figura 21. Ejemplo de matriz de confusión [47].	48
Figura 22. Diagrama de Venn de las características compartidas por las ventanas temporales.	54

Figura 23. Evaluación comparativa de métricas de rendimiento en función del tamaño de ventana de extracción. ....	57
Figura 24. Diagrama de barras comparando las métricas de evaluación por modelo con ventana $w_3$ . ....	60
Figura 25. Matriz de confusión para el modelo <i>Logistic Regression</i> . ....	62

## Índice de tablas

---

Tabla 1. Funciones Básicas Aeropy.py. ....	22
Tabla 2. Características de la librería TSFEL.....	34
Tabla 3. Modelos de <i>Machine Learning</i> utilizados en este trabajo para la determinación de los gestos de la mano. ....	41
Tabla 4. Características seleccionadas para cada tamaño de ventana .....	53
Tabla 5. Características comunes a las ventanas $w_1$ , $w_2$ y $w_3$ .....	55
Tabla 6. Características comunes a las ventanas $w_1$ y $w_2$ .....	55
Tabla 7. Características comunes a las ventanas $w_1$ y $w_3$ .....	55
Tabla 8. Características comunes a las ventanas $w_2$ y $w_3$ .....	56
Tabla 9. Características únicas a $w_1$ .....	56
Tabla 10. Características únicas a $w_3$ .....	56
Tabla 11. Resultados de las métricas obtenidos con cada tamaño de ventana .....	57
Tabla 12. Resultados de las métricas de cada modelo con cada escalador con ventana $w_3$ .....	59



# 1 Introducción

---

En los últimos años, el desarrollo de sistemas de control basados en señales biológicas ha adquirido gran relevancia para el desarrollo de interfaces hombre-máquina. El reconocimiento de gestos de la mano presenta una de las estrategias destacadas para la comunicación entre hombre y máquina. Esta estrategia se utiliza en diferentes aplicaciones como prótesis inteligentes, reconocimiento del lenguaje de signos, dispositivos de rehabilitación o dispositivos de control.

Los sistemas para el reconocimiento de gestos de la mano pueden utilizar dispositivos como guantes, cámaras, unidades de medición inercial (IMUs), sensores de señales electromiográficas (EMGs) o combinaciones de estos sensores, como de señales EMG y IMU [1].

Los sistemas de adquisición de datos como guantes o cámaras de visión tienen limitaciones. Por ejemplo, no pueden ser usados en casos de pacientes con amputaciones o con lesiones nerviosas que no puedan realizar movimientos con la mano. En cambio, los sistemas de reconocimiento que gestos que se basan en señales de EMG pueden ayudar a la rehabilitación de estos pacientes, aunque también tienen sus inconvenientes, como el ruido que generan [1].

La electromiografía es una técnica de registro de la actividad eléctrica producida por los músculos esqueléticos, cuyo uso más extendido se da en el ámbito de la medicina para detectar señales neuromusculares y seguir su evolución. La adquisición de esas señales se consigue mediante electrodos intramusculares (método invasivo) o superficiales, (sEMG, método no invasivo) [2].

En este proyecto se utilizarán las señales EMGs y de orientación proporcionadas por el sensor “Trigno Duo” de Delsys para desarrollar un algoritmo de *Machine Learning* (ML) que pueda detectar tres estados fundamentales de la mano: reposo, mano abierta y mano cerrada. Este enfoque pretende contribuir al desarrollo de sistemas de control más accesibles y eficientes para personas con discapacidades motoras, así como a la mejora de interfaces hombre-máquina en general.

A lo largo del presente documento se describirán las diferentes etapas para la realización del proyecto, como las fases de adquisición de datos y procesamiento de señales, la extracción y selección de características, la construcción y validación de modelos de clasificación de *Machine Learning* y

## Introducción

su posterior implementación en tiempo real. También se presentarán los dispositivos y el software utilizados durante el proyecto.

### 1.1 Electromiografía

La electromiografía es la disciplina que trata con la detección, análisis y el uso de las señales eléctricas que se generan cuando los músculos se contraen de forma voluntaria. A esta señal se le conoce como señales electromiográficas (EMGs) [3]. Específicamente, las señales sEMG (electromiografía de superficie) proporcionan información sobre la actividad muscular de manera no invasiva, lo que las convierte en una fuente eficaz para rehabilitación de paciente amputados [4].

En el ámbito clínico la electromiografía es una metodología para registrar y analizar la actividad bioeléctrica del tejido muscular esquelético con el fin de diagnosticar y tratar enfermedades neuromusculares.

La base de las señales EMG reside en las propiedades bioeléctricas del tejido muscular estriado. Los músculos estriados son los efectores de la motilidad voluntaria [5].

La unidad motora (UM), que es la unidad anatómica y funcional del músculo, se forma por una motoneurona alfa y las fibras musculares (FMs) que inerva. El sistema nervioso central codifica el grado de contracción de las FMs mediante la frecuencia de los impulsos nerviosos de las motoneuronas, que se transmiten como potenciales de acción (PA) a las células musculares. Estos PA son despolarizadores de la membrana de las FMs, que se transmiten a lo largo de la fibra muscular a una velocidad de entre 3 y 5 m/s [5].

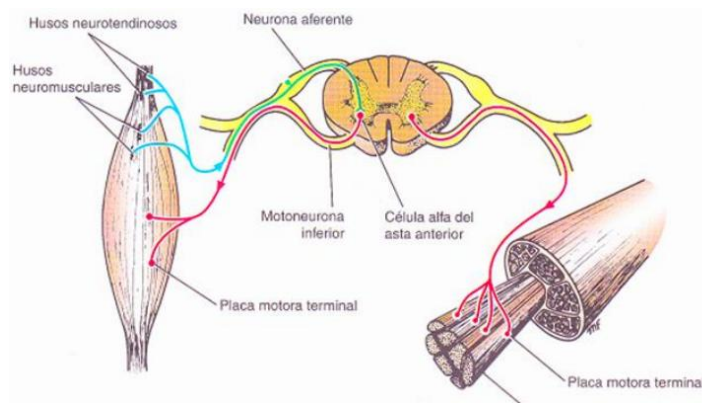


Figura 1. Unidad motora (UM) [2].

## Introducción

Una contracción voluntaria débil activa un número escaso de UM, descargando potenciales de acción a bajas frecuencias. El aumento de fuerza de contracción implica un aumento progresivo de la frecuencia de descarga y el reclutamiento o activación de más UMs, por tanto la amplitud de la señal EMG aumenta correlativamente [5].

El registro de los cambios producidos por la descarga de las FMs de una UM se conoce como Potencial de Acción de Unidad Motora (PAUM). En condiciones normales, la amplitud media de los PAUM es de unos 0,5 mV y la duración varía entre 8 y 14 ms [5].

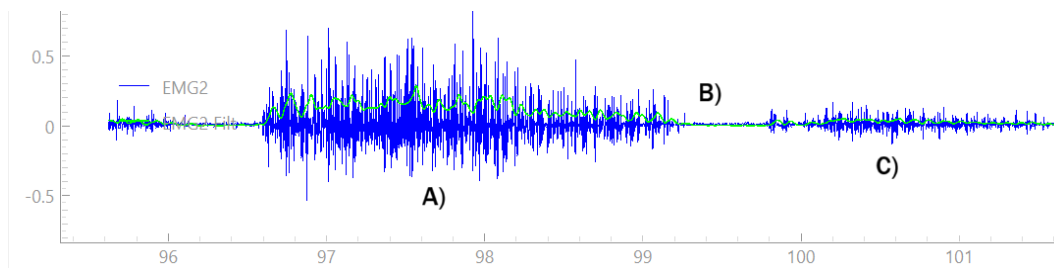


Figura 2. Señal EMG detectada en el antebrazo durante la contracción de la mano (A), reposo (B) y la extensión de la mano (C).

Las señales EMG siempre están acompañadas de ruido, que puede ser de origen técnico o biológico, y generalmente supera la magnitud de la señal de interés en varios órdenes [5].

Algunas fuentes de ruido de origen técnico pueden ser del propio equipo de adquisición, la corriente alterna de la red eléctrica (50 Hz en Europa, 60 Hz en América) o las interferencias electromagnéticas del entorno. Mientras que otras fuentes de ruido de origen biológico pueden ser señales biológicas indeseadas, como actividades de UM lejanas, o la propia impedancia de la piel [5].

Por tanto, el filtrado de este ruido es crucial para obtener una representación inteligible de la actividad muscular.

## 1.2 Estado del arte

Una vez introducido el tema sobre el que va a tratar este trabajo y haberse introducido en el campo de la electromiografía, se procede a analizar proyectos similares de detección de movimientos de la mano ya existentes. En este

## Introducción

apartado se revisan los principales enfoques y líneas de investigación relacionados con el uso de sEMG en el reconocimiento de gestos, así como sus aplicaciones prácticas en el ámbito médico, asistencial y tecnológico. Además, se analizan las oportunidades y desafíos actuales que enfrenta esta tecnología en función del estado de la literatura científica y los proyectos recientes desarrollados en este contexto.

### 1.2.1 Machine Learning para la caracterización y reconocimiento de gestos con señales electromiográficas

El reconocimiento de gestos a partir de señales EMG representa un desafío significativo, dada la naturaleza intrínsecamente compleja de los procesos fisiológicos subyacentes y el comportamiento ruidoso inherente a la señal. A pesar de la existencia de modelos matemáticos que describen la fisiología de la EMG, estos no han proporcionado una solución analítica exacta para el problema del reconocimiento de gestos [6].

En este contexto, los algoritmos de *Machine Learning* se han establecido como herramientas robustas y eficientes para estimar la distribución subyacente a la generación de gestos, permitiendo el desarrollo de estrategias más eficientes y precisas a nivel computacional para la caracterización de señales EMG [7].

Un modelo típico para el reconocimiento de gestos en tiempo real mediante EMG y *Machine Learning* comprende cuatro etapas fundamentales.

- Adquisición de señales EMG:

Se utilizan sensores, comúnmente de superficie (sEMG), para registrar la actividad eléctrica de los músculos. Dispositivos como el Myo Armband, que incorpora ocho sensores EMG de superficie y una IMU son ejemplos de herramientas utilizadas para este fin, midiendo hasta 200 muestras por segundo de actividad eléctrica muscular [6].



## Introducción

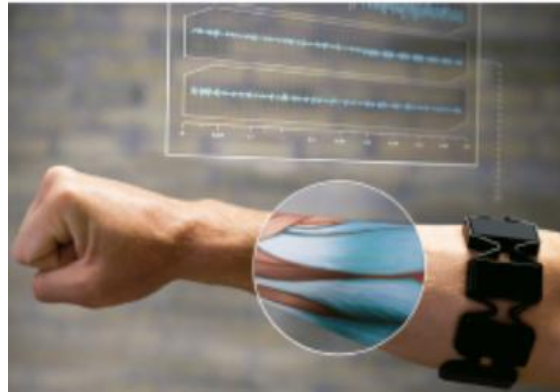


Figura 3. Myo Armband [6].

- Preprocesamiento:

Esta etapa es crucial para aislar la información relevante de la señal EMG del ruido, incluye técnicas como la rectificación, necesaria debido a los valores positivos y negativos de las señales EMG, y el filtrado para extraer información esencial y descartar el ruido [6].

- Extracción de características:

Se aplican diversas técnicas en los dominios del tiempo, frecuencia y tiempo-frecuencia para obtener información significativa que represente cada clase de gesto. En el dominio del tiempo se exploran características como el valor absoluto medio, coeficientes autorregresivos de  $n$ -ésimo orden, *zero cross*, longitud de la señal o el valor eficaz. En el dominio de la frecuencia, se consideran el espectro de potencia, las frecuencias media y mediana y los momentos espectrales. Para el dominio tiempo-frecuencia se utiliza la transformada de *wavelet* [6].

- Clasificación:

En esta etapa, el vector de características extraído de las señales EMG se asigna a una clase de gesto específica. Se emplean dos categorías principales de clasificadores:

Los clasificadores paramétricos tienen una complejidad constante respecto al número de ejemplos de entrenamiento, son fáciles de entender y rápidos. Entre estos se encuentra *Logistic Regression*, análisis discriminante lineal (LDA),

## Introducción

redes neuronales artificiales (ANNs) y máquinas de vectores de soporte (SVMs) [7], [6].

Clasificadores no paramétricos tienen un número potencialmente infinito de parámetros y no asumen una función subyacente. Entre estos se encuentran los árboles de decisión y *k Nearest Neighbours* (KNN) [6].

Las utilidades de los modelos de reconocimiento de gestos basados en EMG son extensas y de gran impacto, abarcando campos de la medicina y la ingeniería. Algunas de las utilidades de estos modelos pueden ser las siguientes [6].

- Prótesis inteligentes:

Permiten a personas con amputaciones controlar dispositivos prostéticos a través de la actividad muscular residual, ya que las señales EMG pueden capturarse incluso en ausencia de la mano.

- Dispositivos de rehabilitación:

Facilitan terapias y es seguimiento del progreso en la recuperación de funciones motoras.

- Sistemas de interacción persona-ordenador (*Human-Computer Interaction*, HCI):

Posibilitan nuevas formas de control para diversas aplicaciones y entornos, mejorando la interfaz entre usuarios y tecnología.

### 1.2.2 Aplicaciones clínicas de las señales EMG.

Las sEMG han demostrado ser de gran utilidad en diversas aplicaciones clínicas, tanto para el sistema nervioso central (SNC) como para el periférico (SNP) [8].

Debido a la alta resolución temporal de las mediciones electrofisiológicas, sEMG se ha utilizado durante mucho tiempo para evaluar la activación del SNC sobre los músculos. Algunas aplicaciones incluyen la medición del momento de activación y duración de la señal del SNC, por ejemplo, en trastornos del

## Introducción

movimiento como la disfonía y los temblores. Este mismo tipo de análisis puede aplicarse a patrones de movimiento complejos, incluyendo la marcha normal y los trastornos asociados a la locomoción [8].

La combinación de múltiples señales de sEMG junto, por ejemplo, un acelerómetro, proporciona una buena representación visual de trastornos del movimiento difíciles de observar [8].

El uso de sEMG permite una caracterización de diversas patologías neuromusculares asociadas al SNP descritas en los siguientes casos.

Las sEMG han evolucionado como herramienta de apoyo en el diagnóstico de trastornos neuromusculares [9]. Aunque revisiones como la de Haig et al [10] concluyó que la literatura era insuficiente para justificar el uso de sEMG en el diagnóstico y manejo rutinario de enfermedades nerviosas y musculares, esta conclusión ha sido objeto de debate. Estudios posteriores sugieren que las sEMG pueden complementar eficazmente a la electromiografía invasiva, proporcionando información adicional relevante [8].

Asimismo, las sEMG derivan una métrica conocida como la velocidad de conducción de la fibra muscular (MFCV), esta métrica es considerada de las más valiosas debido a su vínculo directo con la fisiología muscular [8], [9]. Se ha demostrado su valor en el diagnóstico en miopatías con función de membrana alterada, como la parálisis periódica hipokalémica (HOPP) y la miotonia congénita, donde se han encontrado alteraciones severas en la MFCV, incluso permitiendo la detección de portadores asintomáticos [8]. En el caso de la Esclerosis Lateral Amiotrófica (ALS), las sEMG han mostrado una MFCV significativamente aumentada en las unidades motoras sobrevivientes, complementando la información de las señales electromiográficas invasivas que muestran la pérdida de fibras [8].

Por otro lado, las sEMG se consideran la herramienta principal para el estudio de los determinantes electrofisiológicos de la fatiga muscular patológica. Se ha demostrado un comportamiento anormal de la MFCV durante las pruebas de fatiga en la enfermedad de McArdle y la dominancia de fibras tipo I. También se ha observado un aumento anormal en la frecuencia de potencia mediana de las sEMG en deficiencia de carnitina [8].

Además, las sEMG pueden detectar las fasciculaciones en algunas neuropatías mejor que las señales de electromiografía invasivas, debido a la posibilidad de grabar durante periodos más largos [9].

Finalmente, a través de las sEMG es posible obtener información detallada sobre la localización de las unidades motoras, la dirección de las fibras musculares, la posición de la placa motora y la transición fibra-tendón [8].

## Introducción

A pesar de los avances, la aplicación de las señales sEMG en el entorno clínico presenta desafíos significativos como la complejidad de las señales sEMG y el gran número de variables que la influyen exigen un análisis crítico y cuidadoso para evitar sesgos en las conclusiones. Estas variables pueden ser de carácter técnico, experimental, fisiológico o descriptivo [9]

A nivel técnico las condiciones ambientales y especificaciones de los equipos, así como, la imposibilidad de ajustar las modificaciones de la señal causadas por los tejidos entre la fuente y la superficie de la piel son limitaciones inherentes que pueden afectar a la calidad de las señales sEMG [9]

A nivel experimental, el procedimiento de medición (preparación de la piel, configuración y la localización y orientación de electrodos) y condiciones de contracción pueden influir en la lectura de las señales sEMG [9].

A nivel descriptivo, el procesamiento de las señales y el análisis estadístico a menudo requieren la adhesión a hipótesis de estacionalidad o normalidad, lo que puede llevar a resultados erróneos si se aplican incorrectamente [9].

A nivel fisiológico, las características estructurales y funcionales del sistema neuromuscular son distintas en cada individuo, lo que afecta directamente a la lectura de las señales sEMG al introducir variabilidad interindividual en su forma y comportamiento [9].

Finalmente, la falta de estandarización en sensores, configuraciones, colocación de electrodos y protocolos de registro ha dificultado la integración de las sEMG en el uso clínico rutinario [9],

Sin embargo, la utilidad de una técnica no invasiva para la evaluación de la función neuromuscular es clara. Iniciativas como el proyecto SENIAM (Surface EMG for the Non-Invasive Assessment of Muscles) en Europa y las recomendaciones de la International Society of Electromyography and Kinesiology (ISEK) buscan establecer una base de conocimiento robusta y estandarizada [9].

### **1.2.3 Señales EMG para sistemas HCI**

Las señales EMG son consideradas una fuente principal para nuevas formas de desarrollar sistemas HCI. Un dispositivo de entrada basado en señales EMG permite interpretar las contracciones musculares como comandos que pueden ser procesados por un sistema computacional. Una de las principales ventajas de este enfoque es que la captación de las señales puede realizarse de forma no invasiva mediante electrodos colocados sobre la superficie de la piel, lo que

## Introducción

facilita su integración en aplicaciones prácticas sin generar incomodidad en el usuario.

La implementación de un HCI efectivo requiere clasificar las señales EMG procesadas, lo cual es la parte más difícil del sistema. La elección de la metodología de clasificación depende del campo de aplicación. La mayoría de los clasificadores en el campo están basados en redes neuronales, debido a sus numerosas ventajas en el procesamiento y clasificación de bioseñales [11].

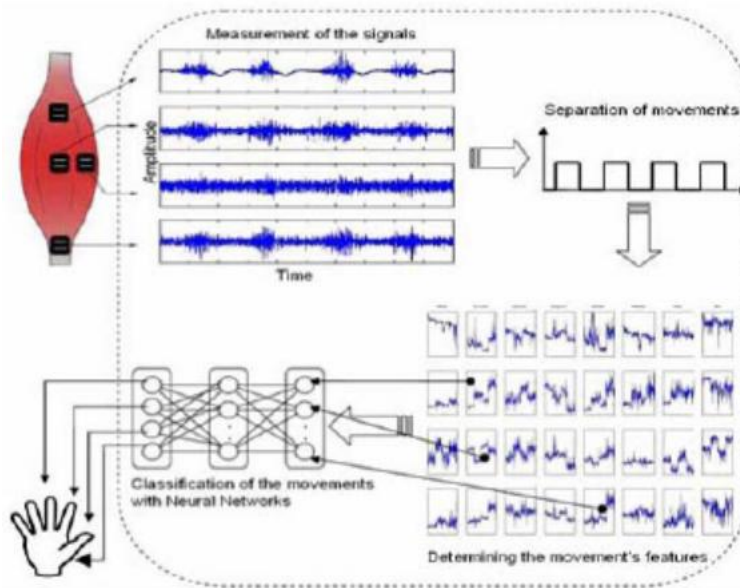


Figura 4. Los primeros pasos para la clasificación de movimientos de la mano utilizando señales sEMG [11].

Las aplicaciones de la clasificación de las señales sEMG abarcan una amplia gama de campos.

Se ha integrado en sistemas que facilitan el uso de software y ordenadores personales por parte de personas con discapacidades motoras, posibilitando la escritura, la navegación web o el manejo de interfaces mediante actividad muscular [11].

En el área del entretenimiento y la realidad virtual, las señales sEMG actúan como entrada para videojuegos inmersivos y equipos de ejercicio físico, mejorando la experiencia del usuario [11].

También han surgido aplicaciones innovadoras como los sistemas de conversión de gestos de voz, diseñados para mejorar la comunicación de personas con capacidad auditiva o del habla. También se han desarrollado

## Introducción

interfaces basadas en EMG para el control del puntero del ratón, así como sistemas avanzados de reconocimientos de gestos [11].

### 1.2.4 Control de prótesis.

En este estudio se propone el control de una prótesis antropomórfica controlada mediante señales de sEMG [12].

La prótesis de mano utiliza una estructura compacta hecha de una aleación de aluminio y que cuenta con 5 dedos, con tamaño similar a los dedos de una mano humana, 15 articulaciones y 4 grados de libertad controlados por cuatro diferentes actuadores (Figura 3). Con esta prótesis, el paciente sería capaz de realizar hasta ocho movimientos diferentes con lo que sería capaz de realizar acciones para manipular objetos.

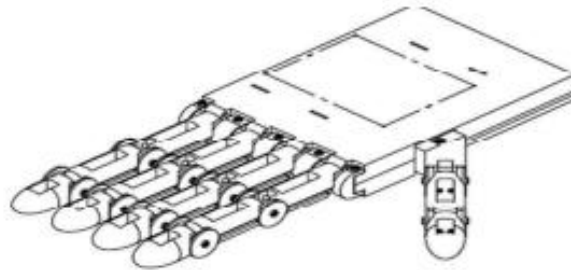


Figura 5. Prótesis de mano [12].

En este estudio se procesan las señales de EMG para reconocer los patrones de movimiento. Las partes fundamentales para el reconocimiento de estos movimientos por las señales EMG incluyen un preprocesamiento de los datos, el segmentado en ventanas de esos datos, la extracción y selección de características.

Las características correspondientes se extraen de las diferentes activaciones de los músculos. Estas características se utilizan para representar las señales EMGs originales y están divididas en tres categorías: Características en el dominio del tiempo, en el dominio de la frecuencia y en el dominio del tiempo-frecuencia.

Finalmente, se utiliza un método de clasificación lineal conocido como Análisis Discriminante Lineal (LDA) basado en aprendizaje supervisado para identificar los patrones correspondientes a distintos gestos o movimientos. El LDA es un clasificador lineal que busca maximizar la separación entre clases proyectando los datos en un espacio de menor dimensión.

### **1.2.5 Control de juegos mediante reconocimiento de gestos de la mano (Hand Gesture Recognition, HGR)**

En este estudio desarrollado por Zhang et al. en 2009 [13], se utilizó un acelerómetro 3D y sensores EMG para desarrollar un sistema de reconocimiento de gestos de la mano. Este sistema busca habilitar interfaces naturales para videojuegos y otras aplicaciones.

Los autores desarrollaron un sistema portátil basado en una unidad que incorpora cuatro sensores EMG superficiales y un acelerómetro 3D, ambos fijados en el antebrazo del usuario.

Durante este estudio se puso como objetivo reconocer 18 movimientos diferentes, entre los que se incluyen movimientos de los dedos y movimientos circulares de la mano en diferentes orientaciones.

La detección de gestos se realizó mediante un modelo multistream de *Hidden Markov Models*, fusionando las señales a nivel de decisión.

La detección de gestos fue validada mediante una aplicación de control de un cubo de Rubik virtual, demostrando el potencial de estos sensores para interfaces naturales en aplicaciones interactivas.

### **1.2.6 Sistema de reconocimiento de lenguaje de signos mediante señales EMG superficiales**

En el estudio de Savur y Sahin [14] se introduce un sistema de reconocimiento en tiempo real del alfabeto del lenguaje de signos americano utilizando exclusivamente señales de electromiografía superficiales.

Para este experimento se utilizaron ocho electrodos secos colocados en el antebrazo derecho del sujeto, simulando una Myo Armband, capturando los 26 gestos correspondientes a las letras del alfabeto. En el procesamiento de las señales se aplicó un filtro digital Butterworth pasabanda de orden 3 entre 5-500 Hz y un filtro notch en 50 Hz para eliminar ruido de línea. Además, se segmentó la señal en ventanas continuas de duración 2 segundos cada una.

En este proyecto se sigue la siguiente metodología:

De cada ventana de los datos EMG se calcularon exclusivamente características en el dominio temporal. En concreto las características fueron medidas simples como valor absoluto medio (MAV), RMS, desviación estándar (SD), media modificada del valor absoluto (MMAV), integral cuadrática (SSI), detector logarítmico (LOG), cambio de amplitud medio (AAC), longitud fractal

## Introducción

(MFL), valor mínimo y máximo, etc... No se capturaron ninguna característica en el dominio de la frecuencia ni transformadas espectrales.

Para la clasificación de los movimientos se utilizó un clasificador SVM (Support Vector Machine) para asignar cada ventana de características al gesto correspondiente.

El sistema se probó en condiciones de tiempo real con un solo sujeto voluntario conocedor del lenguaje de señas americano. Se definieron 26 clases gestuales diferentes que incluían tanto gestos estáticos como dinámicos.

El reconocimiento se dividió en dos etapas. Una primera etapa consistió en pruebas offline sobre datos grabados, donde el SVM alcanzó un 91,1% de precisión. En la segunda etapa, se realizaron pruebas en tiempo real, con el sujeto ejecutando los gestos y el sistema procesándolos con un retardo mínimo. En tiempo real se obtuvo un 82.3 % de precisión.

### 1.3 Objetivos del proyecto

El presente Trabajo de Fin de Grado tiene como propósito el diseño de un sistema inteligente capaz de detectar y clasificar en tiempo real distintos gestos de la mano a partir del procesamiento de señales sEMG y de orientación provenientes del sensor Trigno Duo de Delsys. Esta propuesta se enmarca en el contexto de interfaces naturales de usuario, aportando una solución no invasiva y de bajo coste para aplicaciones en el ámbito de la rehabilitación, accesibilidad y sistemas interactivos basados en reconocimiento de gestos.

El objetivo general del proyecto se basa en desarrollar un sistema de reconocimiento de gestos de la mano mediante técnicas de aprendizaje automático, utilizando como entradas señales EMG y de orientación.

Para ello, se han definido los siguientes objetivos específicos:

- Diseñar y configurar un entorno de adquisición que permita la recogida sincronizada y fiable de señales sEMG y datos de orientación proporcionados por el sensor Trigno Duo.
- Implementar técnicas de filtrado digital para atenuar el ruido y las interferencias presentes en las señales sEMG y de orientación, garantizando su calidad y estabilidad para su posterior análisis.
- Diseñar un protocolo de recogida de datos con múltiples sujetos y movimientos controlados, permitiendo el etiquetado preciso de las muestras para su uso en tareas de clasificación supervisada.



## Introducción

- Aplicar métodos de extracción de características a partir de ventanas temporales solapadas, calculando métricas relevantes en el dominio temporal, frecuencial y del tiempo-frecuencia.
- Evaluar técnicas de selección de características para optimizar el conjunto de atributos, minimizando la redundancia y mejorando el rendimiento computacional.
- Entrenar diversos modelos de clasificación con distintos escaladores de datos y ventanas de tiempo y comparar cual ofrece un mejor rendimiento.
- Obtener una precisión (*accuracy*) aproximada del 80%.



## 2 Hardware empleado

Durante este apartado del proyecto, se describe el hardware empleado para la adquisición de señales de EMG y de orientación, detallando sus características técnicas y su modo de funcionamiento.

Delsys es una empresa estadounidense dedicada a la fabricación de sensores de electromiografía, orientación y aceleración [15]. Cuentan con diferentes sensores con diferentes números de electrodos (todos ellos secos, sin geles conductores). Entre ellos se encuentra el sensor Trigno Duo con el que se desarrollara este proyecto. A continuación, se muestra una imagen del dispositivo (Figura 6).

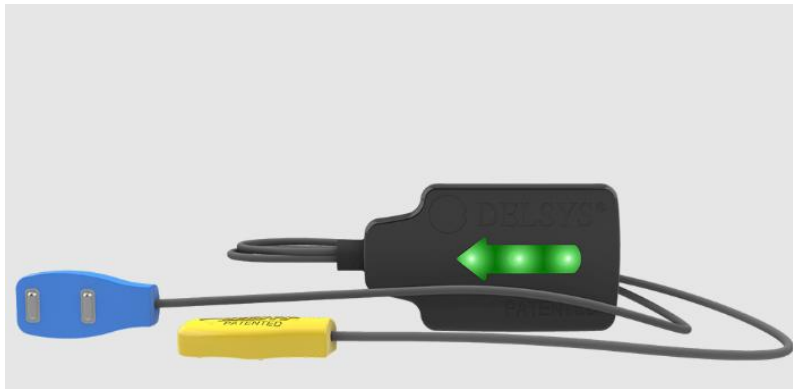


Figura 6. Sensor Trigno Duo Lite de Delsys [15].

Este dispositivo cuenta con dos electrodos de plata independientes capaz de enviar señales de sEMG independientes con una frecuencia de muestreo máxima de 2148 Hz por canal y con un rango máximo de  $\pm 11$  mV. La empresa Delsys especifica un ancho de banda de frecuencias comprendidas entre 10 Hz y 850 Hz o entre 20 Hz y 450Hz. Los dispositivos Trigno Duo pueden ser utilizados para captar las señales de diferentes músculos o de diferentes regiones del mismo músculo.

Además, los sensores Trigno cuentan con una Unidad de Medición Inercial (IMU) acoplada al dispositivo Trigno Avanti (Figura 3) que cuenta con nueve grados de libertad, capaz de proporcionar señales de aceleración, rotación e información del campo magnético terrestre. El acelerómetro puede enviar señales desde el rango de  $\pm 2$  g hasta  $\pm 16$  g con un ancho de banda de entre 24 Hz hasta 450 Hz y el giróscopo puede dar información desde  $\pm 250^\circ/\text{s}$  hasta los  $\pm 2000^\circ/\text{s}$  con un ancho de banda de 24 Hz hasta 360 Hz. El sensor es

## Hardware empleado

capaz de estimar la orientación en el espacio 3D desde nueve canales distintos de información. Esta IMU puede enviar información de las señales de orientación a 10 Hz [16].



Figura 7. Referencia señales sEMG y IMU, Sensor Trigno [15].

La comunicación del sensor Trigno Duo se realiza a través de una estación base conectada al PC mediante el protocolo de comunicación inalámbrico personalizado de Trigno, o a través del protocolo estándar Bluetooth Low Energy (BLE) con dispositivos Android.

Para recibir las lecturas de datos del sensor se realiza mediante el dispositivo Trigno Lite. Este se trata de un receptor USB cuyo máximo número de sensores conectados al mismo es cuatro.



Figura 8. Dispositivo Delsys Trigno Duo Lite.

El sensor Trigno Duo cuenta con una batería recargable que tiene una autonomía de entre 2 y 6 horas dependiendo de las condiciones de uso. Para la recarga del sensor se cuenta con una base de carga (Trigno Lite) que tiene capacidad para recargar hasta cuatro sensores a la vez.

## Hardware empleado



Figura 9. Base de carga del sistema Delsys Trigno Lite.

El sensor Trigno Duo está encapsulado en una carcasa de policarbonato totalmente hermética para evitar cualquier filtración que pueda dañar la electrónica y así poder aumentar la duración del dispositivo. Al ser un dispositivo que suele estar en contacto con la piel de los pacientes, debe de ser resistente al sudor y ser fácilmente lavable.

Para evitar estas filtraciones, el sensor no cuenta con botón de encendido, ni enchufe de carga. Para el encendido del dispositivo, este cuenta con un interruptor magnético interno, el cual se activa al colocarse en el candado magnético localizado en la base de carga. El apagado del sensor se realiza por tiempo o por comunicación inalámbrica.

Para poder trabajar con el sensor es conveniente saber en qué modo se utiliza, para ello, el fabricante añade un sistema de LED en el sensor que indica el estado del sensor con diferentes colores. El código de colores viene especificado en el manual de usuario.



### 3 Metodología del proyecto

---

Una vez introducido el proyecto y haberse familiarizado con los conceptos de señales de sEMG y diferentes estudios realizados con *Machine Learning*, se procederá a realizar una breve explicación de los pasos a seguir para la realización de este trabajo.

Para ello se ha realizado un diagrama de bloques mediante el cual se esquematizará el proyecto basándose en los estudios anteriormente presentados.

Este proyecto presentará siete diferentes etapas que estructuran de forma lógica y secuencial el flujo de trabajo. Estas etapas comprenden: la extracción de datos del sensor, el filtrado de datos de las señales sEMG, la recogida de datos de distintos sujetos, la extracción de características y la selección de las más relevantes, el entrenamiento y validación del mejor modelo de *Machine Learning* y la implementación de este modelo en tiempo real.

El diagrama de bloques propuesto se muestra en la Figura 10.

## Metodología del proyecto

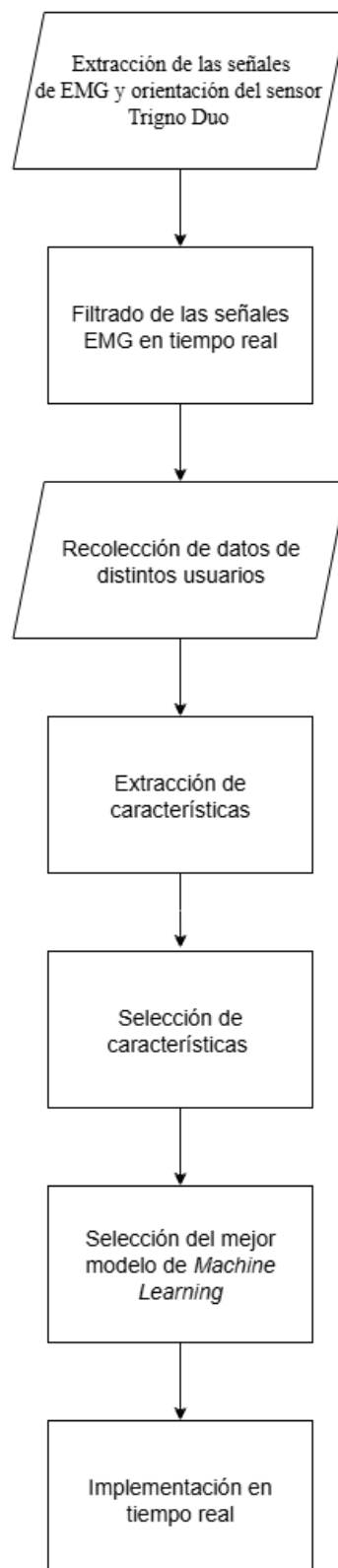


Figura 10. Diagrama de bloques realizado para este trabajo. Los bloques rectangulares indican procesos, mientras que los bloques con formas de paralelogramo indican entrada de datos.



### 3.1 Extracción de datos del sensor

La primera etapa de este proceso será la extracción de datos del sensor Trigno Duo de Delsys descrito anteriormente.

Para ello la propia empresa Delsys proporciona una librería llamada Aeropy.py para configurar el uso del sensor Trigno Duo. El software de Aeropy se desarrolló para ser usado junto con Trigno Wireless Biofeedback System, que en este trabajo se trata del dispositivo USB Delsys Trigno Lite (Figura 4).

Aeropy encapsula la llamada e inicialización de un objeto de la parte precompilada de la librería *BaseInstance*, la cual representa la base a la cual se conectan los sensores, en este caso al dispositivo USB Delsys Trigno Lite [17].

Las funciones más importantes utilizadas para la configuración de este sensor son las de la tabla 1:

Función	Descripción
<b>Conectar a Trigno USB</b>	
<b>ValidateBase(...)</b>	Llamada principal a Trigno Base.
<b>Administración de sensores</b>	
<b>ScanSensors()</b>	Escanea sensores ya emparejados.
<b>GetSensorsNames()</b>	Devuelve el nombre de los sensores.
<b>GetAllSampleModes(...)</b>	Devuelve una lista de los modos de funcionamiento del sensor. (datos, frecuencia)
<b>SetSampleMode( ... )</b>	Establece un modo de toma de datos.
<b>Configuración previa a la recolección de datos.</b>	
<b>Configure( ... )</b>	Configura el Pipeline de transmisión de datos
<b>IsPipelineConfigure( ... )</b>	Devuelve 'True' si el Pipeline está correctamente configurado.
<b>Gestión de colección de datos.</b>	
<b>Start( ... )</b>	Comienza la recolección de datos.

<b>CheckYTDataQueue()</b>	Devuelve <i>'True'</i> si hay nuevos datos <i>YT**</i> en el buffer interno.
<b>PollYTData()</b>	Extrae los datos <i>YT</i> de la cola (si hay)
<b>Stop()</b>	Detiene la recolección de datos.

Tabla 1. Funciones Básicas Aeropy.py.

Los datos YT que se extraen del sensor hacen referencia a un conjunto de muestras temporales organizadas como pares de valores (T, Y) donde la estructura de los datos es la siguiente:

*Dictionary<Guid, List<(double T, double Y)>>*

*Guid*: Es un identificador único de cada canal del sensor (por ejemplo, un canal de EMG).

*List<(double T, double Y)>*: Es una lista de tuplas donde cada tupla contiene:

- T: Indica el instante de tiempo en segundos donde se tomó la muestra.
- Y: Indica el valor de la muestra. Por ejemplo, el valor en voltaje de la señal EMG.

A través de esta librería se ha realizado un programa para extraer los datos del sensor de forma que se obtiene las dos señales EMGs (1926 Hz) y las señales de orientación de la IMU (74 Hz).

### 3.2 Filtrado de señales

Como ya se ha comentado anteriormente, uno de los principales inconvenientes de las señales sEMG es lo sensibles que son al ruido.

Las señales EMGs superficiales contienen la señal que origina el músculo y varios artefactos ruidosos que son endémicos e inevitables. Estos artefactos contaminan las señales EMGs y pueden llevar a malinterpretaciones de la señal. Estos artefactos ocurren especialmente cuando la señal se obtiene durante contracciones dinámicas y cuando está destinada a proporcionar información sobre la fisiología y anatomía de los músculos [18].

Para evitar que estos artefactos afecten al análisis y procesamiento posterior de los datos, es fundamental aplicar un preprocesamiento adecuado a las señales brutas de sEMG.

En este trabajo, el filtrado de señales sEMG se ha implementado a través de una clase personalizada en Python denominada *EMG\_filters*. Esta clase aplica

## Metodología del proyecto

un conjunto de filtros de forma secuencial para limpiar las señales de forma eficiente en tiempo real. El proceso consta de las siguientes etapas:

En primer lugar, se elimina el ruido que se puede generar por las interferencias eléctricas. Para ello se han empleado dos filtros de notch (rechaza banda), uno centrado en 50 Hz y otro en 150 Hz, ambos con un factor de calidad (Q) de 30. El primer filtro elimina directamente la componente de 50 Hz típica de la red europea, y el segundo atenúa posibles armónicos que puedan aparecer a 150 Hz [5].

Una vez suprimido el ruido de red, se aplica un filtro pasa banda de 4º orden, con frecuencias de corte de entre 20 Hz y 450 Hz. Este filtro permite conservar únicamente la banda útil de las señales EMG, ya que la actividad muscular se concentra típicamente en estas frecuencias. Las componentes de baja frecuencia (<20 Hz) suelen estar asociadas a artefactos de movimiento, mientras que las de alta frecuencia (>450 Hz) suelen estar asociadas con el ruido electrónico y térmico [18].

Finalmente, para facilitar la interpretación de la actividad muscular y preparar la señal para posteriores etapas, como la extracción de características, se calcula la envolvente de la señal EMG filtrada. Esto se realiza en dos pasos.

Primero se aplica la transformada de Hilbert para obtener la envolvente instantánea de la señal. Luego, esta señal se pasa por un filtro pasa bajo de 3º orden con una frecuencia de corte de 12 Hz, que actúa como suavizador. Esto elimina oscilaciones rápidas y permite obtener una señal más estable que representa el nivel de activación muscular de forma clara [19].

El resultado del filtrado de las señales se puede ver en la figura 11, donde se puede apreciar las señales roja y azul que pertenecen a las señales sEMG que se reciben de los electrodos en bruto, y la señal verde es la señal sEMG filtrada.

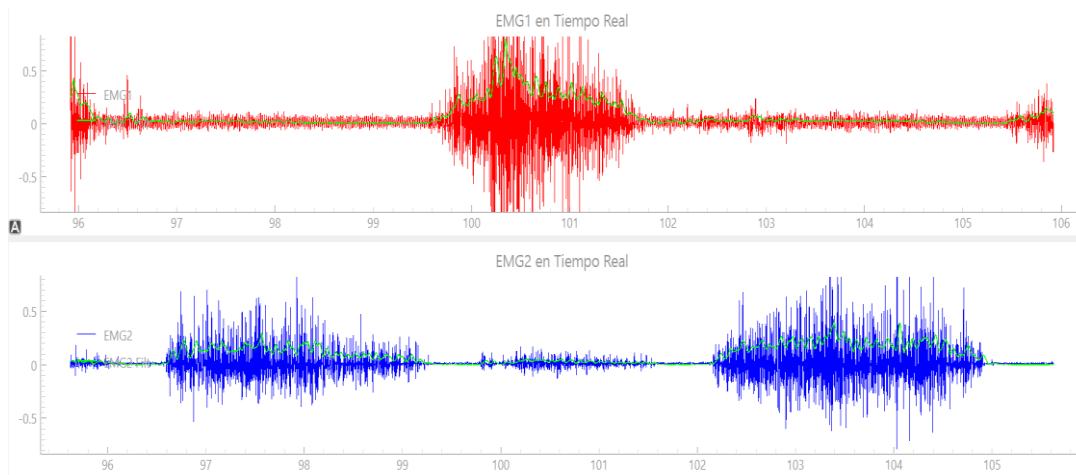


Figura 11. Filtrado de las señales EMG.

### 3.3 Recogida de datos

Para el desarrollo del presente trabajo, se llevó a cabo un proceso de recogida de datos de las señales sEMG y de orientación del sensor Trigno Duo de Delsys.

Para ello se realizó un programa propio en Python utilizando las funciones que ofrece Delsys mediante su librería Aeropy.py descritas en la tabla 1. Este programa gráfico en tiempo real las señales sEMG de los dos electrodos, tanto en crudo como filtradas, y las señales de orientación en ángulos de Euler.

Se hicieron pruebas a 12 participantes de entre 20 y 30 años sanos, sin ninguna lesión. Estos participantes realizaron una serie de movimientos de la mano predefinidos: mano cerrada, reposo y mano abierta, los cuales fueron etiquetados en tiempo real mediante una interfaz gráfica (Figura 12).

## Metodología del proyecto

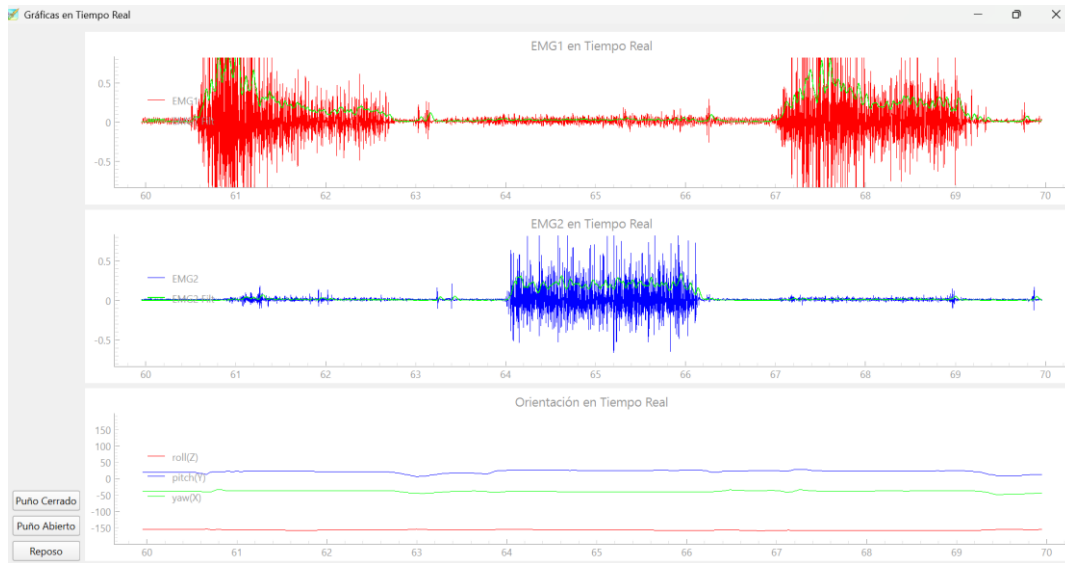


Figura 12. Interfaz gráfica diseñada para visualizar las señales sEMG y de orientación.

Esta interfaz cuenta con botones de mano cerrada, mano abierta y reposo para registrar correctamente los movimientos en la de recogida de datos.

Para medir correctamente las señales electromiográficas se seleccionaron dos músculos del antebrazo cuyas funciones contrarias y complementarias que resultan ideales para detectar los movimientos de apertura y cierre de la mano. El músculo extensor de los dedos y el músculo flexor superficial de los dedos (Figura 13). Estudios de actividad muscular han mostrado que el músculo extensor de los dedos concentra su activación durante la extensión de los dedos (apertura de la mano), mientras que el músculo flexor superficial se activa principalmente durante la flexión de los mismos (gesto de cerrar la mano) [20].

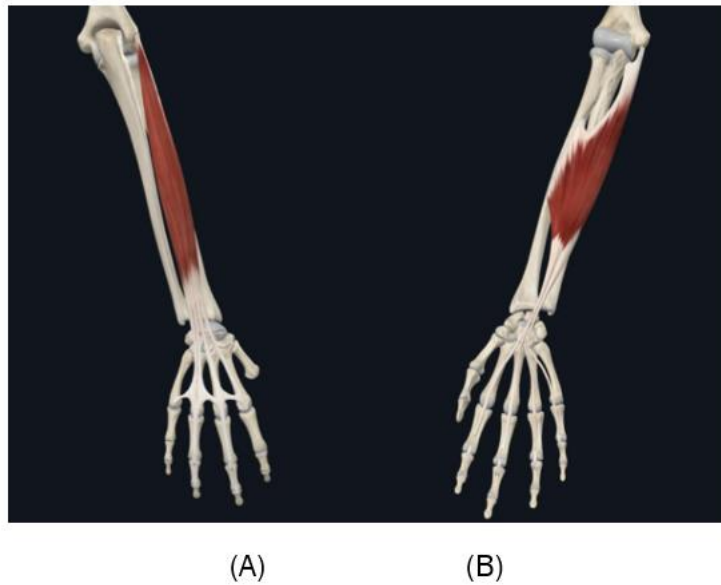


Figura 13. Músculo flexor superficial de los dedos (A) y músculo extensor de los dedos (B).

La colocación de los sensores se realizó siguiendo las recomendaciones del protocolo SENIAM [21], orientando los electrodos de forma longitudinal sobre el eje de las fibras musculares, localizando el punto medio del vientre muscular, y manteniendo una separación adecuada para optimizar la calidad de la señal y evitar la interferencia por ondas tendinosas o estructuras articulares.

La colocación del dispositivo de referencia, que también incluye la IMU, se realizó en la zona posterior de la mano, ya que es una zona eléctricamente neutra en relación con los músculos implicados, con el fin de reducir interferencias en la señal sEMG registrada. Además, es una colocación óptima para la medición de la orientación de la muñeca. La colocación exacta de los electrodos se muestra en la figura 14.

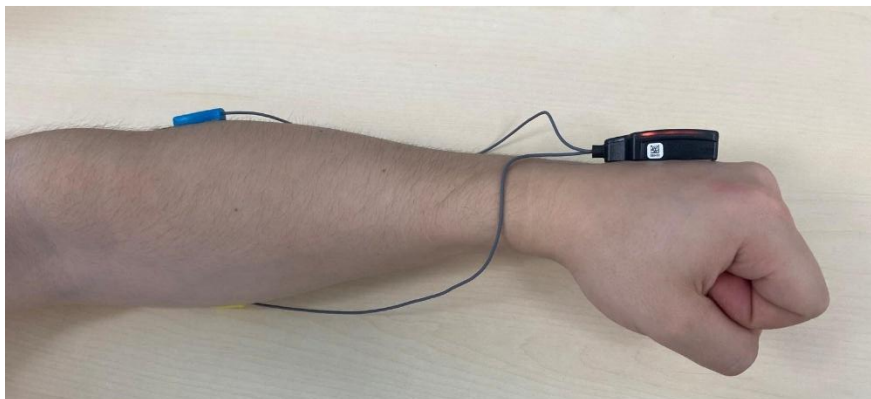


Figura 14. Colocación del sensor para la recogida de datos.

La frecuencia de las señales de sEMG fue de 1926 Hz, mientras que la frecuencia de las señales de orientación fue de 74 Hz. Cada movimiento fue repetido dos veces por cada usuario durante un periodo de aproximadamente 20 s. Por tanto, se recogieron un total de 77040 ( $40 \text{ s} \times 1926 \text{ Hz}$ ) datos de señales sEMG y 2960 ( $40 \text{ s} \times 74 \text{ Hz}$ ) datos de señales de orientación de cada usuario, obteniendo así un conjunto de datos equilibrado y representativo.

Los datos fueron almacenados en formato CSV para su posterior procesamiento, extracción de características y entrenamiento de modelos de clasificación.

### **3.4 Extracción de características de las señales sEMG y orientación.**

#### **3.4.1 Introducción**

Una vez tomadas muestras de los distintos voluntarios para crear una base de datos, se procedió a la extracción de características con el objetivo de transformar los datos temporales en un conjunto de atributos relevantes para la clasificación de los movimientos de la mano.

Las señales sEMG deben de ser previamente segmentadas antes de la extracción de características. Estas señales no pueden ser segmentadas en función de su forma debido a su carácter estocástico, sino que se deben segmentar atendiendo a intervalos de tiempo.

Las señales sEMG son no estacionarias, lo que significa que sus propiedades (como la media o la varianza) cambian con el tiempo. Esta no estacionariedad

causa complejidades y puede afectar catastróficamente los resultados, especialmente el análisis de frecuencia. Sin embargo, un segmento temporal más corto de una señal sEMG puede considerarse localmente estacionario [22].

### 3.4.2 Segmentación de las señales

Esta técnica es conocida como *windowing* o ventanización y, es un proceso crucial en el análisis de señales sEMG, ya sea para aplicaciones de diagnóstico o para el desarrollo de sistemas de control basados en sEMG, como los dispositivos robóticos de rehabilitación y asistencia. Este proceso consiste en dividir las señales sEMG en partes más pequeñas, llamadas segmentos o ventanas, para analizar las características de la señal.

El objetivo es extraer información útil de cada ventana que pueda distinguir una clase de movimiento o actividad muscular de otra [23].

La duración de la ventana es un factor crítico que puede afectar al rendimiento del sistema. El tamaño de la ventana debe ser lo suficientemente larga en el tiempo como para contener información que represente el patrón de la señal original [23].

Un tamaño de ventana demasiado pequeño puede contener mayor varianza y menor sesgo, mientras que ventanas más grandes tienen mayor sesgo y menor varianza. Además, hay que considerar otros factores como el rendimiento en tiempo real, la carga computacional y la precisión del sistema diseñado [24].

Existen dos técnicas diferentes de *windowing* (ventanización) (Figura 15):

- *Disjoint windowing* (Ventanización disjunta): En esta técnica la ventana se caracteriza únicamente por el tamaño de la ventana. Además, las ventanas son distintas y no se superponen.
- *Overlap windowing* (Ventanización superpuesta): En esta técnica la ventana se caracteriza tanto por el tamaño de ventana como por el tamaño del solapamiento (*overlap size*). Esta técnica permite usar tamaños de ventanas más grandes sin introducir demoras significativas en la operación a tiempo real del dispositivo. Sin embargo, la carga computacional es mayor que en la técnica de ventanización disjunta [24]. El número de ventanas al aplicar la ventanización superpuesta se puede calcular como en la siguiente ecuación [12].

$$N^{\circ} \text{ de ventanas} = \frac{\text{longitud de los datos} - \text{tamaño de ventana}}{\text{tamaño del solapamiento}} + 1$$



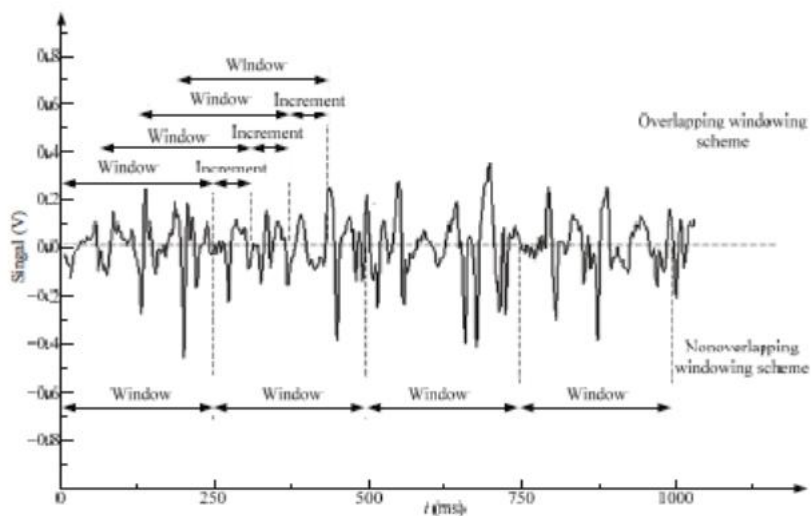


Figura 15. Comparación de las técnicas de ventanización disjunta y superpuesta [12].

### 3.4.3 Características de las señales

Una vez segmentadas las señales sEMG en periodos de tiempo más cortos, se procede a extraer las características de cada ventana.

Aunque los clasificadores pueden entrenarse utilizando las señales filtradas, normalmente se obtienen mejores resultados al extraer características de dichas señales antes del entrenamiento del modelo. La extracción de características también permite reducir la dimensionalidad, lo que simplifica el procesamiento y la clasificación posteriores [25].

Las características se pueden clasificar en tres categorías diferentes: características en el dominio del tiempo, características en el dominio de la frecuencia y características en el dominio tiempo-frecuencia. Comúnmente se utilizan combinaciones de estas para trabajar con modelos de clasificación [25].

Las características del dominio del tiempo (TD) son evaluadas basándose en la amplitud de la señal que varía en el tiempo. La amplitud de la señal depende del estado y tipo de músculo que se estudie en el trabajo. Estas características no requieren ninguna transformación adicional en la señal. Estos elementos hacen que las características TD tengan un coste computacional muy bajo [25].

Las características en el dominio de la frecuencia (FD) son generalmente propiedades estadísticas de la densidad espectral de potencia (PSD) de las señales EMG. Estas se calculan mediante métodos paramétricos o un periodograma [26].

## Metodología del proyecto

Las características FD se utilizan principalmente para la evaluación de la fatiga muscular y el análisis del reclutamiento de unidades motoras. Los cambios en las señales EMG dentro de este dominio están relacionados con variaciones en la frecuencia de potencia mediana (MPF), que varía debido a un cambio hacia frecuencias más bajas [26].

Las características en el dominio del tiempo-frecuencia (TFD) combinan la información del tiempo y la frecuencia, permitiendo caracterizar la información de frecuencia variable en diferentes ubicaciones temporales. Esto es crucial para las señales EMG, ya que son inherentemente estacionarias.

Su principal ventaja es que superan la limitación de las características TD, que solo son adecuadas para señales estacionarias, localizan la energía tanto en el tiempo como en la frecuencia, lo que permite una descripción precisa de fenómenos físicos [26].

### 3.4.4 TSFEL

Para este proyecto se utilizó la librería TSFEL (*Time Series Feature Extraction Library*), una herramienta en Python diseñada para facilitar la extracción automática de características temporales [27] (Figura 16).



Figura 16. Logo de la librería TSFEL [27].

Esta librería organiza las características extraídas en distintas categorías según el dominio del análisis, incluyendo el dominio temporal, el dominio estadístico, el dominio espectral y el dominio fractal, lo que permite capturar patrones relevantes en distintas dimensiones de la señal [28].

- Características temporales: Analizan los cambios y patrones del conjunto de datos a lo largo del tiempo. Capturan información como

## Metodología del proyecto

tendencias, ciclos, correlaciones temporales, la cual es esencial para entender comportamientos dinámicos y predecir valores futuros. En este conjunto se incluyen las características del dominio del tiempo que son sensibles al orden de las observaciones.

En este grupo se incluyen características como *Waveform Length* (WL), *Zero Crossing* (ZC), *Slope Sign Changes* (SSC), y transformaciones como autocorrelación.

- **Características estadísticas:** Resumen los datos utilizando estadísticas descriptivas. Incluyen medidas como la media, la varianza, la asimetría (*skewness*) y *curtosis*, proporcionando una visión general de la distribución de los datos, su tendencia central, dispersión y forma. En este conjunto se incluyen características que no son sensibles al orden de las observaciones.
- **Características espectrales:** Se centran en el dominio de la frecuencia de los datos. Al transformar los datos utilizando técnicas como FFT, la transformada de Fourier o la transformada Wavelet, revelan periodicidades subyacentes, armónicos y componentes de frecuencia, lo cual es crucial para identificar patrones cíclicos y oscilaciones.
- **Características fractales:** Describen la complejidad y la autosimilitud de los datos en diferentes escalas. Se derivan de la teoría fractal e incluyen medidas como la dimensión fractal, capturando los patrones intrincados e irregularidades que suelen estar presentes en sistemas naturales y complejos.

Las características que ofrece esta librería son las mostradas en la tabla 2 [27]:

Característica	Descripción
<b>Temporales</b>	
<b>auc(signal, fs)</b>	Calcula el área bajo de la curva de la señal utilizando la regla del trapecio.
<b>autocorr(signal)</b>	Calcula el primer cruce 1/e de la función de autocorrelación (ACF).
<b>calc_centroid(signal,fs)</b>	Calcula el centroide a lo largo del eje de tiempo.
<b><u>distance(signal)</u></b>	Calcula la distancia recorrida por la señal.
<b><u>lempel_ziv(signal[, threshold])</u></b>	Calcula el índice complejo de Lempel-Ziv (LZ), normalizado por la longitud de la señal.
<b><u>mean_abs_diff(signal)</u></b>	Calcula la diferencia absoluta de la media de la señal.

## Metodología del proyecto

<u>mean_diff(signal)</u>	Calcula las diferencias de la media de la señal.
<u>median_abs_diff(signal)</u>	Calcula la diferencia absoluta mediana de la señal.
<u>median_diff(signal)</u>	Calcula las diferencias medianas de la señal.
<u>negative_turning(signal)</u>	Calcula el número de inflexión negativos de la señal.
<u>neighbourhood_peaks(signal[, n])</u>	Calcula el número de picos en un vecindario definido de la señal.
<u>positive_turning(signal)</u>	Calcula el número positivo de puntos de inflexión de la señal.
<u>slope(signal)</u>	Calcula la pendiente de la señal.
<u>sum_abs_diff(signal)</u>	Calcula la suma de las diferencias absolutas de la señal.
<u>zero_cross(signal)</u>	Calcula la tasa de cruces por cero de la señal.
<b>Estadísticas</b>	
<u>abs_energy(signal)</u>	Calcula la energía absoluta de la señal.
<u>average_power(signal,fs)</u>	Calcula la potencia media de la señal.
<u>calc_max(signal)</u>	Calcula el máximo valor de la señal.
<u>calc_median(signal)</u>	Calcula el valor medio de la señal.
<u>calc_std(signal)</u>	Calcula la mediana de la señal.
<u>calc_min(signal)</u>	Calcula el valor mínimo de la señal.
<u>calc_var(signal)</u>	Calcula la varianza de la señal.
<u>ecdf(signal[,d])</u>	Calcula el valor de ECDF (función distributiva acumulativa empírica) a lo largo del eje del tiempo.
<u>ecdf_percentile(signal[, percentile])</u>	Calcula el valor del percentil de ECDF (función distributiva acumulativa empírica).
<u>ecdf_percentile_count(signal[, percentile])</u>	Calcula las sumas acumulativas de las muestras que son menores que el percentil.
<u>ecdf_slope(signal[,p_init, p_end])</u>	Calcula la pendiente de la ECDF (función distributiva acumulativa empírica) entre dos percentiles.
<u>entropy(signal[, prob])</u>	Calcula la entropía de la señal utilizando la entropía de Shannon.
<u>hist_mode(signal[, nbins])</u>	Calcula la moda de un histograma utilizando un número dado de intervalos (bins) distribuidos linealmente.
<u>interq_range(signal)</u>	Calcula el rango intercuartilico de la señal.
<u>kurtosis(signal)</u>	Calcula kurtosis de la señal.
<u>mean_abs_deviation(signal)</u>	Calcula la desviación absoluta media de la señal.

## Metodología del proyecto

<u>median_abs_deviation(signal)</u>	Calcula la desviación absoluta mediana de la señal
<u>pk_pk_distance(signal)</u>	Calcula la distancia entre los picos.
<u>rms(signal)</u>	Calcula la raíz cuadrática media de la señal.
<u>skewness(signal)</u>	Calcula la asimetría de la señal.
<b>Espectrales</b>	
<u>fundamental_frequency(signal, fs)</u>	Calcula la frecuencia fundamental de la señal.
<u>human_range_energy(signal, fs)</u>	Calcula el ratio de energía en el rango humano.
<u>lpcc(signal[, n_coeff])</u>	Calcula la predicción linear de los coeficientes cepstrales.
<u>max_frequency(signal, fs)</u>	Calcula la máxima frecuencia de la señal.
<u>max_power_spectrum(signal, fs)</u>	Calcula la máxima potencia espectral de la señal.
<u>median_frequency(signal, fs)</u>	Calcula la frecuencia mediana de la señal.
<u>mfcc(signal, fs[, pre_emphasis, nfft,...])</u>	Calcula los coeficientes cepstrales de MEL.
<u>power_bandwidth(signal, fs)</u>	Calcula el ancho de banda de la densidad espectral de potencia de la señal.
<u>spectral_centroid(signal, fs)</u>	Centroide del espectro.
<u>spectral_decrease(signal, fs)</u>	Representa la cantidad de disminución de la amplitud del espectro.
<u>spectral_distance(signal, fs)</u>	Calcula la distancia de la señal espectral.
<u>spectral_entropy(signal, fs)</u>	Calcula la entropía de la señal espectral basándose en la transformada de Fourier.
<u>spectral_kurtosis(signal, fs)</u>	Mide la planitud de una distribución alrededor de su valor medio.
<u>spectral_positive_turning(signal, fs)</u>	Calcula el número de puntos de inflexión positivos de la magnitud fft de la señal.
<u>spectral_roll_off(signal, fs)</u>	Calcula el desplazamiento espectral de la señal.
<u>spectral_roll_on(signal, fs)</u>	Calcula el inicio espectral de la señal.
<u>spectral_skewness(signal, fs)</u>	Mide la asimetría de una distribución alrededor de su valor medio.
<u>spectral_slope(signal, fs)</u>	Calcula la pendiente espectral.
<u>spectral_spread(signal, fs)</u>	Mide la dispersión del espectro alrededor de su valor medio.
<u>spectral_variation(signal, fs)</u>	Calcula cantidad de variación del espectro a lo largo del tiempo.

<u>spectrogram_mean_coeff(signal,fs[, bins])</u>	Calcula la densidad media de la potencia espectral (PSD) para cada frecuencia a través de la duración total de la señal tomada del espectrograma.
<u>wavelet_abs_mean(signal, fs[, wavelet, ...])</u>	Calcula el valor medio absoluto de la Transformada Wavelet Continua (CWT) en cada escala Wavelet.
<u>wavelet_energy(signal, fs[, wavelet, max_width])</u>	Calcula la energía de la Transformada Wavelet Continua (CWT) para cada escala Wavelet.
<u>wavelet_entropy(signal, fs[, wavelet, max_width])</u>	Calcula la entropía de la Transformada Wavelet Continua (CWT) para cada escala Wavelet.
<u>wavelet_std(signal, fs[, wavelet, max_width])</u>	Calcula la desviación estándar de la Transformada Wavelet Continua (CWT) para cada escala Wavelet.
<u>wavelet_var(signal, fs[, wavelet, max_width])</u>	Calcula la varianza de la Transformada Wavelet Continua (CWT) para cada escala Wavelet.
<b>Fractales</b>	
<u>dfa(signal)</u>	Calcula el Análisis de Fluctuación Sin Tendencia (DFA) de la señal.
<u>higuchi_fractal_dimension(signal)</u>	Calcula la dimensión fractal de la señal usando el método de Higuchi (HFD)
<u>hurst_exponent(signal)</u>	Calcula la exponente de Hurst de la señal a través del análisis de rango reescalado (R/S)
<u>maximum_fractal_length(signal)</u>	Calcula la longitud máxima del fractal de la señal, la cual es la longitud media en la escala más pequeña media a partir del gráfico logarítmico que determina la dimensión fractal (FD).
<u>mse(signal[, m, maxscale, tolerance])</u>	Calcula la entropía multiescala (MSE) de la señal, que realiza un análisis de entropía sobre múltiples escalas temporales.
<u>petrosian_fractal_dimension(signal)</u>	Calcula la dimensión Fractal de Petrosian de la señal.

Tabla 2. Características de la librería TSFEL

### 3.5 Selección de características

#### 3.5.1 Introducción

Una vez extraídas todas las características, es crucial abordar la complejidad inherente a la gran cantidad de atributos disponibles en las señales EMG. Un elevado número de entradas en el reconocimiento de patrones puede dificultar

el rendimiento del posterior entrenamiento del modelo clasificador, además, el análisis completo de todas las características disponibles requiere un alto esfuerzo computacional. Por lo tanto, es necesario elegir el enfoque adecuado para la selección de características, ya que cada problema tiene su mejor conjunto de características y seleccionar el conjunto óptimo es esencial para lograr un alto rendimiento del clasificador [29].

A continuación, se analizan los métodos y técnicas que se pueden utilizar para que los modelos funcionen mejor y de manera eficiente.

### 3.5.2 Métodos basados en filtros

Los métodos de filtro se utilizan generalmente como un paso de preprocesamiento de datos, la selección de características es independiente a cualquier algoritmo de Machine Learning [30]. Un ejemplo es encontrar la correlación entre cada característica y el objetivo y descartar aquellas que no alcanzan un umbral. Estos métodos son fáciles y rápidos, pero no son tan eficaces como otros métodos [31]. Algunos métodos basados en filtros son:

- Umbral de varianza: La eliminación de características con varianza nula o casi nula que no aportan información discriminativa es esencial para la construcción de un modelo. Es decir, una característica que sea del mismo valor para todas o gran parte de las ventanas de las señales de EMG es una característica no válida [32].
- Correlación de Pearson: Se usa como una medida para cuantificar la dependencia lineal entre dos variables continuas X e Y, su valor varía entre -1 a 1 [30].
- LDA: El análisis discriminante lineal se usa para encontrar una combinación lineal de características que caracteriza o separa dos o más clases, o niveles, de una variable categórica [30].
- ANOVA: El filtro ANOVA (*Analysis of Variance*) es similar a la eliminación de características por varianza, pero este, en vez de eliminar características de poca o nula varianza, selecciona las características que tienen más significancia estadística con respecto a las clases mediante análisis de varianza [33].
- Chi-cuadrado: Es una prueba estadística que se aplica a los grupos de características categóricas para evaluar la probabilidad de correlación o asociación entre ellos utilizando su distribución de frecuencia [30]. Esta prueba permite eliminar las características que probablemente sean independientes de la clase, y, por lo tanto, irrelevantes para la clasificación [34].

### 3.5.3 Métodos de envoltura (*wrapper*)

Estos métodos necesitan de un algoritmo de Machine Learning y utiliza su rendimiento como criterio de evaluación. Esto significa que el clasificador actúa como la función objetivo para encontrar el subconjunto óptimo de características que minimice el error de generalización. Estos métodos actúan como un preprocesamiento para clasificar las características, seleccionando aquellas con el ranking más alto para aplicarlas a un predictor.

A diferencia de los métodos de filtro, los métodos de envoltura consideran directamente la interacción entre las características y el rendimiento del clasificador.

Los métodos de envoltura tienden a ofrecer un mejor rendimiento de clasificación en comparación con los métodos de filtro. Sin embargo, son computacionalmente más costosos, debido a la re-ejecución del algoritmo de aprendizaje para cada subconjunto de datos [35].



Figura 17 Gráfica de los métodos de envoltura [30].

Algunos de los ejemplos más comunes de métodos de envoltura son los siguientes:

- Selección hacia delante (*Forward Selection*): es un método iterativo en el que se comienza sin tener ninguna característica en el modelo. En cada iteración, se añade una característica que mejora el modelo de clasificación hasta que la adición de una nueva característica no mejore el rendimiento del modelo [30].
- Eliminación hacia atrás (*Backward Selection*): se comienza con un grupo en el que se encuentran todas las características y se va eliminando la característica menos significativa en cada iteración, lo que mejora el rendimiento del modelo. Este método se repite hasta que no se observe ninguna mejora en la eliminación de características. [30]



- Eliminación de características recursivas (*Recursive Feature Elimination*): La eliminación de características recursivas (RFE) es un algoritmo de optimización que busca encontrar el subconjunto de funciones con mejor rendimiento predictivo. Este método crea repetidamente modelos y deja de lado la mejor o la peor característica de rendimiento en cada iteración. El proceso se repite hasta alcanzar el número deseado de características [30].

En este trabajo, RFE se ha implementado utilizando como estimador un *Random Forest Classifier*. Este clasificador permite evaluar la importancia de cada característica de manera robusta, gracias a su capacidad para capturar relaciones no lineales y manejar datos de alta dimensión. Al finalizar el proceso, las características seleccionadas corresponden a aquellas que el modelo ha considerado más relevantes para la tarea de clasificación.

### 3.5.4 Métodos embebidos

Los métodos embebidos (*Embedded Methods*) integran la selección de características directamente en el proceso de entrenamiento del modelo. A diferencia de los métodos de filtro o de envoltura, estos son más eficientes y aprovechan las interacciones entre las variables mientras entrenan el modelo [36]. Algunos ejemplos de estos métodos son los siguientes:

- LASSO: La regularización L1 o LASSO reduce algunos de los coeficientes a cero, lo que indica que ciertos predictores o características serán multiplicados por cero para estimar el valor objetivo. Por lo tanto, no se añadirán a la predicción final del objetivo, esto significa que estas características pueden eliminarse, ya que no están contribuyendo a la predicción final [36], [37].
- RIDGE: La regularización L2 o RIDGE no reduce ningún coeficiente a cero a diferencia de LASSO, sino que solo aproxima a cero los coeficientes [36].
- Elastic Nets: Elastic Nets es una combinación de la regularización L1 y L2. Esta se beneficia de ambas penalizaciones para estabilidad y selección simultánea [36].

También existen algunos algoritmos como SVM (*Support Vector Machine*) con características incorporadas mediante métodos embebidos que también permiten determinar automáticamente qué atributos quedan activos [37].

## 3.6 Selección del mejor Modelo de *Machine Learning*

### 3.6.1 Introducción

Una vez finalizadas las etapas de preprocesamiento de datos, extracción y selección de características, se procede a entrenar y evaluar distintos modelos de clasificación con el objetivo de identificar aquel que ofrezca el mejor rendimiento para la detección de los movimientos de la mano a partir de las señales sEMG y de orientación. Esta etapa es crucial, ya que la elección del modelo más adecuado puede influir significativamente en la precisión y robustez del sistema final.

Para ello, se compararán distintos algoritmos clasificadores de *Machine Learning* bajo un mismo marco experimental aplicando técnicas de validación cruzada anidada y aplicando métricas de rendimiento como la precisión, la sensibilidad y el F1-score.

A continuación, se describen los modelos de clasificación supervisada comparados durante el proyecto.

### 3.6.2 Modelos de *Machine Learning*

- *Linear SVC* (SVM lineal):

*Linear SVC* implementa una SVM de kernel lineal. Busca el hiperplano óptimo que separa las clases maximizando el margen entre ellas. Este hiperplano se obtiene resolviendo un problema de Programación Cuadrática convexo cuyo objetivo es minimizar un error de clasificación regularizado. En la práctica, *Linear SVC* de *scikit-learn* utiliza la pérdida *squared hinge* y una regularización L2 sobre el vector de pesos [38].

- Regresión Logística:

La regresión logística es un modelo lineal probabilístico para clasificación binaria. Se calcula una combinación lineal de las variables de entrada 'x' y se aplica la función sigmoide:

$$p(y = 1|x) = \sigma(w * x + b) = \frac{1}{1 + e^{-(w*x+b)}}$$

El parámetro  $w$  se ajusta típicamente mediante máxima verosimilitud (optimización de la entropía cruzada). Para problemas multiclase se emplea *softmax* (regresión logística multinomial). La regresión logística es conceptualmente similar al SVM lineal, pero usa pérdida logística convexa en lugar de *hinge* [39].

## Metodología del proyecto

- *K-Nearest Neighbors* (KNN):

*K-Nearest-Neighbors* (Figura 18) es un clasificador de aprendizaje supervisado no paramétrico que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Es uno de los algoritmos de clasificación y regresión más populares y sencillos que se utilizan en machine learning a día de hoy [40].

En el contexto de la clasificación, la forma más simple del modelo kNN consiste en predecir la etiqueta de clase objetivo como aquella que está más representada entre los k ejemplo de entrenamiento más similares a un punto de consulta determinado. En otras palabras, la etiqueta de cada clase puede ser considerada como la moda de las k etiquetas de entrenamiento o el resultado de una votación por pluralidad [41].

En el caso de predicciones binarias (problemas de clasificación con dos clases), siempre hay una mayoría o un empate. Por tanto, una votación por mayoría también es automáticamente una votación por pluralidad. Sin embargo, en escenarios de múltiples clases, no se requiere una mayoría para realizar una predicción con KNN [41].

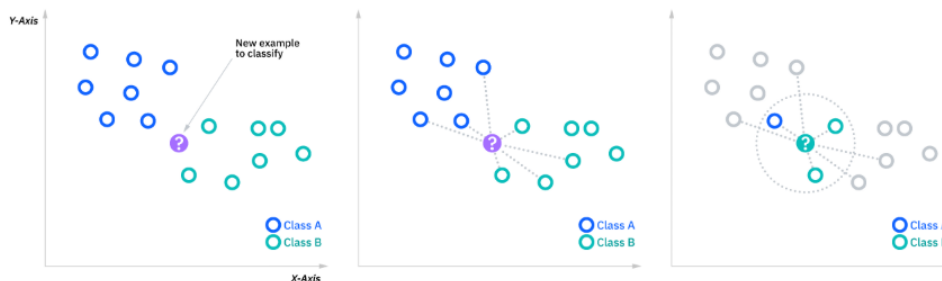


Figura 18. Diagrama conceptual de un modelo KNN [40].

- *Random Forest*:

*Random Forest* (Figura 19) es un ensamblado de árboles de decisión basado en *bagging* o embolsado. Cada árbol se entrena con una muestra *Bootstrap* (muestreo con reemplazo) de los datos y, además, en cada división del árbol se usa un subconjunto aleatorio de características (subespacio aleatorio). Al tener muchos árboles no correlacionados, el bosque promedia las predicciones individuales,

## Metodología del proyecto

reduciendo la varianza sin aumentar el sesgo. Esto suele evitar el sobreajuste de un solo árbol. El resultado es un clasificador robusto que maneja bien tanto clasificación binaria como multiclase [42].

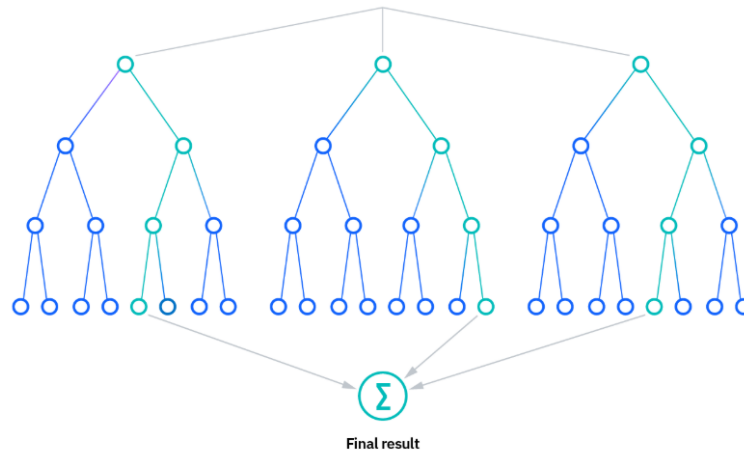


Figura 19. Diagrama conceptual de un modelo Random Forest [42].

- *Gradient Boosting* (árboles de gradiente clásicos):

*Gradient Boosting* construye un modelo aditivo de manera progresiva, etapa por etapa. El funcionamiento de este algoritmo se basa en un modelo fuerte inicial y en cada iteración se entrena un modelo débil (usualmente un árbol pequeño) para corregir los errores del conjunto actual. El modelo resultante es la suma (o votación) de todos los árboles débiles. En particular *AdaBoost* ajusta pesos en instancias mal clasificadas, mientras que *Gradient Boosting Machine* minimiza gradientes de pérdida. En esencia, estos métodos comparten la idea de que cada árbol mejora donde el anterior falla [43].

- *XGBoost* (eXtreme Gradient Boosting):

*XGBoost* es una implementación escalable y optimizada de *Gradient Boosting*. *XGBoost* añade regularización (L1/L2) y técnicas eficientes (uso de gradientes de segundo orden, manejo de datos dispersos, cache inteligente) para mejorar la velocidad y precisión. Matemáticamente, en cada iteración se minimiza una función objetivo que incluye la función de pérdida y un término de complejidad del árbol, usando aproximación mediante Taylor de segundo orden. El resultado es un potente ensamble que suele lograr resultados de vanguardia en muchas tareas de clasificación [44].

## Metodología del proyecto

- *Catboost*

*CatBoost* es otra implementación de *Gradient Boosting* sobre árboles de decisión, con innovaciones para datos reales. Sus dos aportes clave son el “*ordered boosting*” y un tratamiento especial de variables categóricas. El “*ordered boosting*” entrena usando permutaciones de los datos para evitar filtrado de información objetivo (target leakage), logrando estimaciones menos sesgadas. Además, *CatBoost* convierte categóricas de forma innovadora [45].

Como resultado, *CatBoost* suele superar a otros *boosted trees* en calidad de predicción en diversos conjuntos de datos. En esencia, matemáticamente es similar a *XGBoost*, con estas mejoras algorítmicas específicas [45].

A continuación se muestra en la tabla 3 un resumen recogiendo los modelos de clasificación con sus principales ventajas e inconvenientes.

Modelo	Tipo	Ventajas	Inconvenientes
<b>Linear SVC</b>	Lineal	Rápido, efectivo en espacios de alta dimensión, buen rendimiento con características relevantes	No maneja bien datos no linealmente separables sin kernel.
<b>Logistic Regression</b>	Lineal	Interpretable, rápido, útil para datos linealmente separables	Menor rendimiento si los datos no tienen relación lineal
<b>KNN</b>	No paramétrico	Simple, no requiere entrenamiento, útil con datos bien distribuidos.	Lento con grandes datasets, sensible al ruido y escala de datos.
<b>Random Forest</b>	Ensamble (Árboles)	Maneja no linealidades, robusto al sobreajuste, buen rendimiento general	Puede ser más lento, menos interpretable que modelos lineales.
<b>Gradient Boosting</b>	Ensamble (Boosting)	Precisión elevada, captura relaciones lineales no complejas	Más lento que otros modelos, sensible al ruido si no se ajusta bien
<b>XGBoost</b>	Ensamble (Boosting)	Alta precisión, manejo eficiente de valores nulos, regularización incorporada	Complejidad alta, requiere ajuste de hiperparámetros.
<b>CatBoost</b>	Ensamble (Boosting)	Buen rendimiento con datos categóricos, manejo automático de valores nulos	Menos conocido, documentación algo limitada.

Tabla 3. Modelos de *Machine Learning* utilizados en este trabajo para la determinación de los gestos de la mano.

### 3.6.3 Métodos de escalado

Antes de entrenar los modelos de clasificación, es necesario aplicar un escalado a las características extraídas. Este proceso tiene como objetivo garantizar que todas las variables tengan un rango comparable, evitando que aquellas con valores más grandes dominen el aprendizaje del modelo.

En este trabajo se han comparado cuatro estrategias distintas de escalado:

- *Min-Max Scaler*: Este escalador transforma linealmente los datos originales en un rango específico. Sea  $x_{min}$  y  $x_{max}$  el valor mínimo y el máximo de una variable en el conjunto de datos de entrenamiento. El algoritmo *Min-max* escala un valor  $v$  de dicha variable a un nuevo valor  $v'$  utilizando la ecuación siguiente:

$$v' = \frac{v - x_{min}}{x_{max} - x_{min}} + x_{min}$$

Esta transformación mapea los valores originales de las muestras de entrenamiento en el intervalo  $[x_{min}, x_{max}]$  a  $[-1,1]$  o  $[0,1]$ , dependiendo del rango especificado, preservando la distribución relativa de los datos [46].

- *Standard Scaler (Z-score)*: En este algoritmo, el nuevo valor  $v'$  de una variable de escala a partir del valor original  $v$  utilizando la siguiente fórmula:

$$v' = \frac{(v - \bar{x})}{\sigma}$$

Donde  $\bar{x}$  y  $\sigma$  son la media y la desviación estándar de los valores de la variable en las muestras de entrenamiento respectivamente. Tras esta transformación, los nuevos valores tendrán una media de 0 y una desviación estándar de 1 [46].

- *Robust Scaler*: Este algoritmo es un método de escalado de datos que se propone para mejorar la precisión de datos biomédicos. Una de sus características fundamentales es que es intrínsecamente robusto a los valores atípicos.

El algoritmo *Robust Scaler* funciona escalando los datos a un intervalo apropiado, específicamente el intervalo  $[0,1]$ , utilizando una función logística generalizada para ajustar la función de densidad acumulada empírica de los datos [46].

- *None*: Se incluyen también experimentos sin aplicar ningún tipo de normalización, para evaluar si ciertos modelos (como los basados en

árboles, que no se ven afectados por escalas) funcionan adecuadamente sin este preprocesamiento.

### 3.6.4 Selección y validación de modelos.

#### 3.6.4.1 Introducción.

Una vez explicados los modelos de Machine Learning y los escaladores que se van a utilizar en el proyecto, se procede a explicar el proceso de selección y validación de los modelos, así como los criterios empleados para la selección del mejor modelo predictivo.

Uno de los aspectos fundamentales en el desarrollo de modelos de aprendizaje automático supervisado es la estimación precisa de su rendimiento en datos no vistos. Para evitar una evaluación optimista o sesgada del modelo, es necesario emplear técnicas de validación que separen adecuadamente los conjuntos de entrenamiento y prueba.

En este trabajo se ha optado por el uso de validación cruzada *k-fold* anidada (*nested k-fold cross-validation*), por ser una de las estrategias más robustas y recomendadas en escenarios donde se lleva a cabo selección de hiperparámetros, combinada con un esquema *Leave-One-Subject-Out* (LOSO).

El esquema LOSO junto a *nested CV* es el preferido para abordar las diferencias interindividuales en la distribución de las características de las señales sEMG y para garantizar que los modelos de *Machine Learning* desarrollados sean robustos y generalizables a nuevos sujetos.

#### 3.6.4.2 Cross Validation.

La validación cruzada (Cross-validation, CV) es un método de remuestreo utilizado para evaluar modelos de aprendizaje automático con un conjunto de datos limitado [47].

El método CV consiste en dividir los datos en un conjunto de entrenamiento y otro conjunto de prueba. Se entrenan los datos con el conjunto de entrenamiento y luego se evalúa el resultado con el conjunto de prueba. Pero con este método solo evalúas el modelo una vez, entonces no se puede saber si el resultado fue aleatorio o no. Para ello es necesario evaluar el modelo múltiples veces para poder tener más confianza en el diseño del modelo.

Este método tiene un parámetro llamado *k* que se refiere al número de grupos en los que se va a dividir el conjunto de datos. Por eso este procedimiento se suele denominar *k-fold CV*.

## Metodología del proyecto

La validación cruzada se utiliza principalmente para estimar la capacidad del modelo de reconocer datos no conocidos. Es decir, se usa una muestra limitada para estimar como el modelo funcionaría en general a la hora de hacer predicciones de datos no utilizados durante el entrenamiento.

El procedimiento general de la validación cruzada k-fold es el siguiente [47]:

- Mezcla aleatoriamente el conjunto de datos.
- Divide el conjunto de datos en  $k$  grupos.
- Para cada grupo único:
  - Toma el grupo como conjunto de datos de prueba.
  - Toma los grupos restantes como conjunto de entrenamiento.
  - Ajusta el modelo usando el conjunto de entrenamiento y lo evalúa con el conjunto de prueba.
  - Conserva la puntuación de evaluación y descarta el modelo.
- Resume la capacidad del modelo utilizando el conjunto de puntuaciones obtenidas en las evaluaciones.

La selección del valor  $k$  en la validación cruzada k-fold es un aspecto crítico, ya que puede influir significativamente en la estimación del rendimiento del modelo. Una elección inapropiada puede provocar una valoración errónea de la capacidad predictiva del modelo generando resultados con alta varianza (si el rendimiento depende excesivamente de los datos de entrenamiento específicos) o con alto sesgo (si se sobreestima la calidad del modelo) [47].

Existen tres formas comunes para determinar el valor  $k$ :

1. Representatividad estadística: El valor  $k$  se escoge de forma que permita que cada partición de entrenamiento sea lo suficientemente grande como para representar de forma adecuada el conjunto de datos original-
2.  $k=10$ : Es una elección empírica habitual, ya que proporciona un equilibrio aceptable entre sesgo y varianza. Se ha demostrado que, en la práctica,  $k=10$  suele ofrecer estimaciones fiables del rendimiento del modelo.
3.  $k=n$ : En este caso,  $k$  se iguala al tamaño del conjunto de datos ( $k=n$ ), de modo que en cada iteración se utiliza un único ejemplo como conjunto de prueba y el resto como conjunto de entrenamiento. Esta técnica garantiza que cada observación se utilice como dato de prueba, maximizando así el aprovechamiento del conjunto de datos. Este



## Metodología del proyecto

método también se conoce como LOOCV (*Leave One Out Cross Validation*).

### 3.6.4.3 Nested CV

El inconveniente que tiene k-fold CV ocurre cuando se utiliza el mismo conjunto de datos tanto para ajustar como para seleccionar el modelo, se corre el riesgo de obtener una evaluación sesgada positiva de su desempeño [48].

Una forma de mitigar es anidar el procedimiento de optimización de hiperparámetros dentro del procedimiento de selección del modelo. A esta técnica se le conoce como *double cross-validation* o *nested cross-validation* y es una de las técnicas preferidas para evaluar y comparar modelo de Machine Learning.

Nested CV es una técnica que aborda el problema de la optimización de hiperparámetros y la selección del modelo para intentar solucionar un problema de sobreajuste del conjunto de datos [48].

Este procedimiento considera la optimización de hiperparámetros como parte inherente del propio modelo, e integra dicha optimización dentro de un esquema más amplio de validación cruzada k-fold utilizado para evaluar y comparar distintos modelos.

En este enfoque, el procedimiento de CV para la optimización de hiperparámetros se encuentra anidada dentro de otro ciclo de CV utilizado para la selección del modelo. Esta estructura de doble bucle es lo que da nombre al método como double CV.

Como se ha explicado antes, en k-fold CV se entrena un modelo con  $k-1$  subconjuntos de datos, llamado conjunto de entrenamiento, y se evalúa con el subconjunto restante, llamado subconjunto de prueba.

En nested CV, cada conjunto de entrenamiento generado por el bucle externo se utiliza como entrada para un procedimiento de optimización de hiperparámetros, como grid search o random search, que encuentra el mejor conjunto de hiperparámetros para el modelo. La evaluación del conjunto de hiperparámetros se realiza utilizando k-fold CV que divide el conjunto de datos de entrenamiento en  $k$  pliegues [48].

Con este procedimiento, la búsqueda de hiperparámetros no tiene posibilidad de sobreajustarse al conjunto de datos, ya que únicamente se expone a un subconjunto de los datos proporcionado por validación cruzada externa. Esto elimina el riesgo de que el procedimiento de búsqueda sobreentrene le

## Metodología del proyecto

conjunto de datos original, proporcionando así una estimación menos sesgada del rendimiento del modelo.

El mayor inconveniente de nested CV es el gran incremento de número de evaluaciones que se tienen que realizar a los modelos. Si  $n*k$  modelos son entrenados y evaluados en la búsqueda de hiperparámetros de un modelo CV tradicional, entonces en el procedimiento nested CV este número aumenta a  $k*n*k$  [48].

### 3.6.4.4 LOSO

Para la validación externa de los modelos se ha integrado, junto a la técnica nested CV, el esquema LOSO. Esta combinación permite evaluar de manera más rigurosa la capacidad de generalización de los modelos [49].

El procedimiento LOSO es usado para estimar el rendimiento de los algoritmos de *Machine Learning* cuando estos son usados para realizar predicciones sobre datos que no se han utilizado durante la etapa de entrenamiento.

Es un método computacionalmente costoso, ofrece una estimación fiable y no sesgada del rendimiento del modelo.

El esquema LOSO es una estrategia de validación fundamental en el aprendizaje automático, especialmente cuando se trabaja con datos que tienen una estructura jerárquica, como es el caso de las señales biológicas, como las sEMG, donde múltiples mediciones provienen del mismo individuo [50].

El principio de LOSO recae en los datos de un sujeto completo se dejan fuera de la etapa de entrenamiento para conformar el conjunto de prueba, mientras que los datos de todos los demás sujetos se utilizan para entrenar el modelo. Este proceso se repite de manera iterativa hasta que los datos de cada sujeto han sido usados una vez como conjunto de prueba. Si el estudio incluye  $n$  sujetos, se realizarán  $n$  iteraciones, aseguran que el modelo sea evaluado en datos de un sujeto que no ha pertenecido al grupo de sujetos del entrenamiento [50].

La recomendación de usar LOSO en aplicaciones con señales biológicas es debido a varias razones críticas que abordan las complejidades inherentes a este tipo de datos [50].

- La violación de la suposición de independencia: CV asume que todas las muestras son independientes y provienen de una distribución idéntica. Sin embargo, en el procesamiento de señales biológicas, esta suposición no se puede garantizar debido a la posible presencia de

## Metodología del proyecto

estructuras jerárquicas, correlaciones espacio-temporales o procesos no estacionarios [50].

- Estimación del sesgo de la muestra y la generalizabilidad (Argumento de jerarquía): Se recomienda específicamente LOSO CV en lugar de CV para estudios de diagnóstico clínico con múltiples sujetos. Los resultados de CV estándar, deben ser considerados sesgados de forma optimista, lo que significa que se puede sobreestimar el rendimiento real del modelo. Mientras que LOSO CV simula la inclusión de nuevos datos de futuros sujetos en el análisis. Esto es crucial para evaluar la generalizabilidad de la metodología [51].
- Evitar la sobreestimación de la precisión: Métodos como la CV pueden sobreestimar masivamente la precisión de la predicción [50], [51] . Esto ocurre porque el algoritmo puede identificar implícitamente la identidad de la persona basándose en características únicas del sujeto. En cambio, LOSO CV garantiza que los conjuntos de entrenamiento y prueba contengan registros de diferentes sujetos, forzando al modelo a aprender características generalizables de la condición biológica en lugar de características específicas del modelo [51].

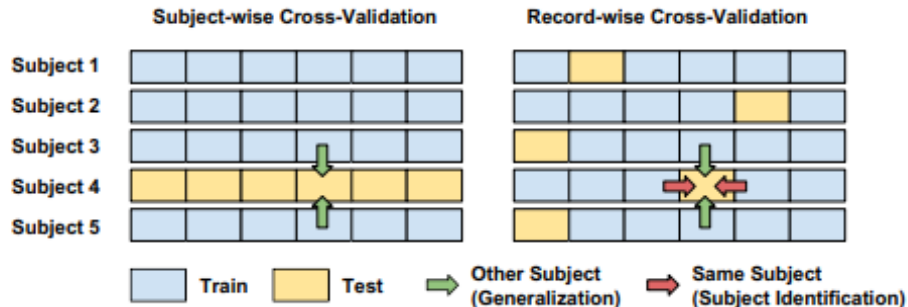


Figura 20. Una visualización del esquema LOSO CV y CV en un escenario de diagnóstico [51].

No obstante, debido a su elevado coste computacional, LOSO no resulta adecuada para conjuntos de datos muy grandes con decenas o miles de sujetos.

### 3.6.5 Métricas de evaluación.

Una vez definidos los esquemas de validación utilizados, LOSO y nested CV, resulta fundamental establecer cómo se evaluará el rendimiento de los

## Metodología del proyecto

modelos de clasificación. Para ello se recurren a una serie de métricas de evaluación que permiten cuantificar la eficacia del modelo en la tarea de clasificación.

Las métricas de evaluación utilizadas en este trabajo han sido la *accuracy* (exactitud), *precision* (precisión), el *recall* (sensibilidad) y el *F1-score*.

Las métricas de evaluación se emplean al menos en tres aplicaciones principales. Evaluar la capacidad de generalización del clasificador entrenado cuando se prueba con datos no vistos, como evaluador para la selección del modelo y como discriminador para seleccionar la solución óptima entre todas las soluciones generadas durante el entrenamiento de la clasificación [52].

Para los problemas de clasificación binaria, la evaluación de la discriminación de la solución óptima durante el entrenamiento se deriva de una matriz de confusión (Figura 21). Esta matriz, donde las filas representan la clase predicha y las columnas la clase real, se define mediante los siguientes componentes [52]:

- *True positive* (TP): Número de instancias positivas correctamente clasificadas.
- *False negative* (FN): Número de instancias positivas incorrectamente clasificadas. (predichas como negativas).
- *False positive* (FP): Número de instancias negativas incorrectamente clasificadas (predichas como positivas).
- *True negative* (TN): Número de instancias negativas correctamente clasificadas.

VALORES DE PREDICIÓN	Verdaderos Positivos (VP)	Falsos Positivos (FP) ERROR tipo 1
	Falsos Negativos (FN) ERROR tipo 2	Verdaderos Negativos (VN)
	VALORES REALES	

Figura 21. Ejemplo de matriz de confusión [53].

## Metodología del proyecto

A partir esta matriz de confusión, se derivan varias métricas ampliamente usadas para evaluar el rendimiento de un clasificador en problemas de clasificación multiclases [52].

- *Accuracy* (Acc):

También conocida como precisión o exactitud. Esta métrica mide la proporción de predicciones correctas sobre el número total de instancias evaluadas. Es la métrica más utilizada en la práctica, tanto para problemas de clasificación binaria como multiclase. Se calcula como:

$$Acc = \frac{TP + TN}{TP + FN + FP + TN}$$

- *Precision* (P):

*Precision* se utiliza para medir la proporción de instancias positivas que son correctamente predichos de entre el total de instancias que el modelo predijo como positivas. Es decir, responde a la pregunta: de todas las instancias que mi modelo clasificó como positivas ¿Cuántas eran realmente positivas? Se calcula como:

$$P = \frac{TP}{TP + FP}$$

- *Recall* (R) o *Sensitivity* (sen):

*Recall*, también conocido como *sensitivity*, mide la fracción de instancias positivas que son correctamente clasificados. Es decir, responde a la pregunta: de todas las instancias que eran realmente positivas, ¿Cuántas logró identificar correctamente el modelo? Se calcula como:

$$R = \frac{TP}{TP + FN}$$

- F1-score:

Esta métrica representa la media armónica entre los valores de *recall* y *precision*. F1-score ha sido reportada como un buen discriminador y ha demostrado un mejor rendimiento que *accuracy* en la optimización de clasificadores para problemas de clasificación binaria. Se calcula como:

$$F1 - score = \frac{2 * P * R}{P + R}$$



## 4 Presentación de resultados

---

Una vez explicada la metodología de este trabajo, en esta sección se presentan y analizan los resultados obtenidos tras el entrenamiento y validación de los modelos de clasificados desarrollados anteriormente.

Para ello se ha llevado a cabo un exhaustivo proceso experimental en el que se han considerado tamaños de ventana deslizante diferentes con un solapamiento del 40% durante la segmentación de los datos, con el objetivo de evaluar el impacto del parámetro temporal en el rendimiento del sistema de reconocimiento de datos.

Se han evaluado tres tamaños de ventana diferentes: 0,25 s con un solapamiento de 0,1 s ( $w_1$ ), 0,5 s con un solapamiento de 0,2 s ( $w_2$ ), y 1 s con un solapamiento de 0,4 s ( $w_3$ ). Cada configuración ha sido sometida al mismo proceso de extracción y selección de características y posterior normalización mediante los métodos de escalado explicados en el apartado 3.6.3.

Posteriormente se han entrenado y validado los modelos utilizando un esquema de Nested CV combinado con una validación intersujeto LOSO, garantizando así una evaluación robusta y generalizable.

A continuación se presentan los resultados obtenidos para cada combinación de modelo, escalador y tamaño de ventana, evaluados mediante métricas estándar para problemas de clasificación multiclase como *accuracy*, *precision*, *recall* y *F1-score* explicadas en el apartado 3.6.4.5.

Este análisis permite identificar no solo que modelo ofrece el mejor rendimiento general, sino también como el preprocesamiento temporal afecta a la capacidad predictiva del sistema en contexto del reconocimiento de movimientos de la mano a partir de señales sEMG y de orientación.

### 4.1 Selección de características

En primer lugar, se comenzará con la selección de características. El objetivo de este paso es identificar y conservar únicamente aquellas variables más relevantes y discriminativas para la tarea de clasificación, eliminando información redundante, irrelevante o ruidosa.

En este trabajo se han extraído las características y seleccionado las características siguiendo el mismo método para las ventanas  $w_1$ ,  $w_2$  y  $w_3$ .

## Presentación de resultados

De cada señal sEMG se han extraído las todas las características proporcionadas por la librería TSFEL presentadas en la Tabla 2 del apartado 3.4.4. Posteriormente se han realizado los mismos métodos de selección de características para las tres ventanas de tiempo diferente. Estos métodos de selección se han aplicado a todo el conjunto de características en común, es decir a las características extraídas de la señal sEMG1 y a las características extraídas a la señal sEMG2.

En primer lugar, se han eliminado las características con varianza nula o casi nula que no aportan información discriminativa para el modelo. Para este método se ha aplicado un umbral de varianza de 0.01.

A partir de estas características se han seleccionado las 50 mejores características mediante el filtro de ANOVA. Finalmente se seleccionan las mejores 20 características con el método de envoltura de RFE utilizando como estimador *Random Forest Classifier*, (apartado 3.5.3).

Después de aplicar estos filtros queda un conjunto de 20 características a las que se añade el valor de la media de los datos de orientación durante la duración de las ventanas expresados en ángulos de Euler.

A continuación, se muestran las características seleccionadas para cada tamaño de ventana diferente (Tabla 4).

<b>W<sub>1</sub></b>	<b>W<sub>2</sub></b>	<b>W<sub>3</sub></b>
absolute_energy_EMG1	absolute_energy_EMG1	absolute_energy_EMG1
average_power_EMG1	average_power_EMG1	average_power_EMG1
ecdf_percentile_0_EMG1	mfcc_1_EMG1	ecdf_percentile_1_EMG1
ecdf_percentile_1_EMG1	mfcc_10_EMG1	maximum_fractal_length_EMG1
max_EMG1	mfcc_2_EMG1	max_EMG1
máximo_fractal_length_EMG1	mfcc_9_EMG1	mean_EMG1
mean_EMG1	maximum_fractal_length_EMG1	median_EMG1
mediant_EMG1	spectral_decrease_EMG1	root_mean_square_EMG1



## Presentación de resultados

root_mean_square_EM G1	spectral_distance_EM G1	spectral_decrease_EM G1
spectral_decrease_EMG 1	sum_absolute_diff_EM G1	spectral_distance_E MG1
spectral_distance_EMG 1	absolute_energy_EMG2	sum_absolute_diff_EM G1
absolute_energy_EMG2	average_power_EMG2	absolute_energy_EMG 2
average_power_EMG2	mfcc_1_EMG2	average_power_EMG2
ecdf_percentile_0_EMG 2	mfcc_10_EMG2	ecdf_percentile_0_EM G2
máximo_fractal_lengt h_EMG2	mfcc_11_EMG2	histogram_mode_EMG 2
mean_EMG2	máximo_fractal_lengt h_EMG2	máximo_fractal_lengt h_EMG2
median_EMG1	spectral_distance_EMG 2	median_EMG2
min_EMG2	spectral_distance_EMG 2	min_EMG2
root_mean_square_EM G1	spectral_positive_turnni ng_points_EMG2	spectral_decrease_EM G2
sum_absolute_diff_EM G2	sum_absolute_diff_EM G2	sum_absolute_diff_EM G2

Tabla 4. Características seleccionadas para cada tamaño de ventana

Con el objetivo de analizar la estabilidad y consistencia de las características seleccionadas durante el entrenamiento del modelo, se realizó una comparación entre las distintas ventanas  $w_1$ ,  $w_2$  y  $w_3$ . A continuación se muestra un diagrama de Venn que ilustra las características compartidas y exclusivas de cada ventana temporal.

## Presentación de resultados

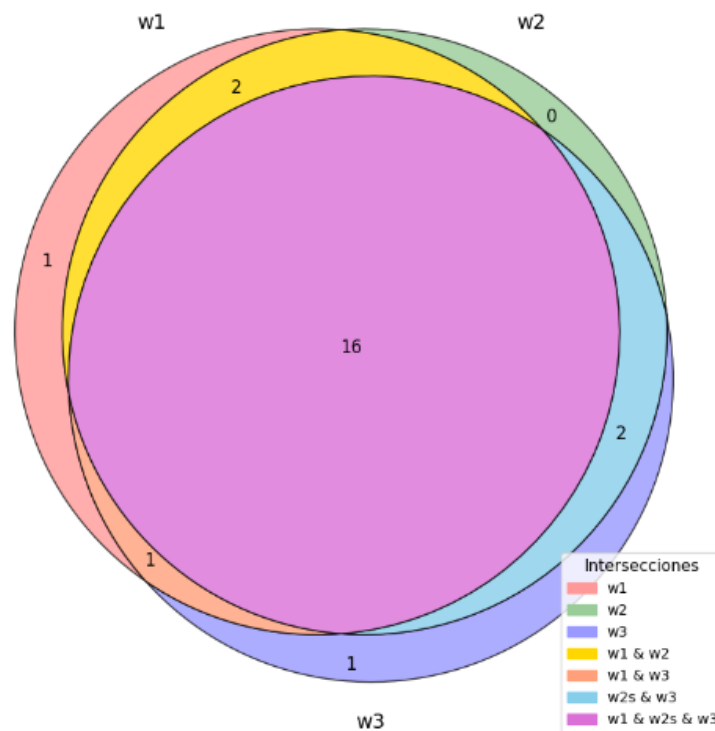


Figura 22. Diagrama de Venn de las características compartidas por las ventanas temporales.

Este diagrama (Figura 22) muestra que existe un conjunto de dieciséis características comunes entre todas las ventanas (zona rosa), lo cual indica una gran robustez frente a cambios en la duración de la ventana de extracción. La ventana  $w_1$  y  $w_3$  presentan una pequeña cantidad de características exclusivas, mientras que la ventana  $w_2$  no aporta ninguna característica exclusiva. También se observan intersecciones parciales entre pares de ventanas, dos características comunes entre  $w_1$  y  $w_2$ , dos entre  $w_2$  y  $w_3$  y una entre  $w_1$  y  $w_3$ .

Estos resultados indican que, aunque algunas características pueden depender de la duración de la ventana, existe un núcleo común amplio que se mantiene estable independiente de la variación del tamaño de la ventana, lo cual es deseable para garantizar la generalización del modelo independientemente de la segmentación temporal empleada.

A continuación se muestran las características que pertenecen al núcleo común de todas las ventanas, aquellas que pertenecen a la intersección entre dos ventanas y aquellas que son exclusivas únicamente de una ventana.

## Presentación de resultados

<b>Características comunes a <math>w_1</math>, <math>w_2</math> y <math>w_3</math></b>
absolute_energy_EMG1
absolute_energy_EMG2
average_power_EMG1
ecdf_percentile_0_EMG2
ecdf_percentile_1_EMG1
max_EMG1
maximum_fractal_length_EMG1
maximum_fractal_lengthn_EMG2
mean_EMG1
median_EMG1
median_EMG2
min_EMG2
root_mean_square_EMG1
spectral_decrease_EMG1
spectral_distance_EMG1
sum_absolutte_diff_EMG2

Tabla 5. Características comunes a las ventanas  $w_1$ ,  $w_2$  y  $w_3$ .

<b>Características comunes a <math>w_1</math> y <math>w_2</math></b>
ecdf_percentile_0_EMG1
mean_EMG2

Tabla 6. Características comunes a las ventanas  $w_1$  y  $w_2$ .

<b>Características comunes a <math>w_1</math> y <math>w_3</math></b>
average_power_EMG2

Tabla 7. Características comunes a las ventanas  $w_1$  y  $w_3$ .

Características comunes a $w_2$ y $w_3$
spectral_decrease_EMG2
sum_absolute_diff_EMG1

Tabla 8. Características comunes a las ventanas  $w_2$  y  $w_3$ .

Características únicas a $w_1$
Root_mean_square_EMG2

Tabla 9. Características únicas a  $w_1$ .

Características únicas a $w_3$
Histogram_mode_EMG2

Tabla 10. Características únicas a  $w_3$ .

## 4.2 Selección del tamaño de ventana

Una vez seleccionadas las características para cada ventana mediante técnicas de extracción y reducción de dimensionalidad, se procedió a determinar el tamaño óptimo de ventana para el análisis de las señales sEMG y de orientación.

Para ello se evaluaron las diferentes combinaciones de tamaño de ventana e incrementos,  $w_1$ ,  $w_1$  y  $w_3$ , midiendo el impacto en el rendimiento de los clasificadores mediante las métricas anteriormente comentadas.

La selección del tamaño óptimo se basó en los resultados obtenidos por el valor medio de las métricas de evaluación de todos los modelos entrenados durante el proceso de entrenamiento.

## Presentación de resultados

Ventanas	Métricas			
	<i>Accuracy</i>	<i>Precision</i>	<i>F1-score</i>	<i>Recall</i>
$w_1$	0,632±0,055	0,655±0,05	0,628±0,056	0,632±0,055
$w_2$	0,657±0,048	0,681±0,042	0,653±0,047	0,657±0,048
$w_3$	0,664±0,052	0,681±0,054	0,659±0,053	0,664±0,052

Tabla 11. Resultados de las métricas obtenidos con cada tamaño de ventana.

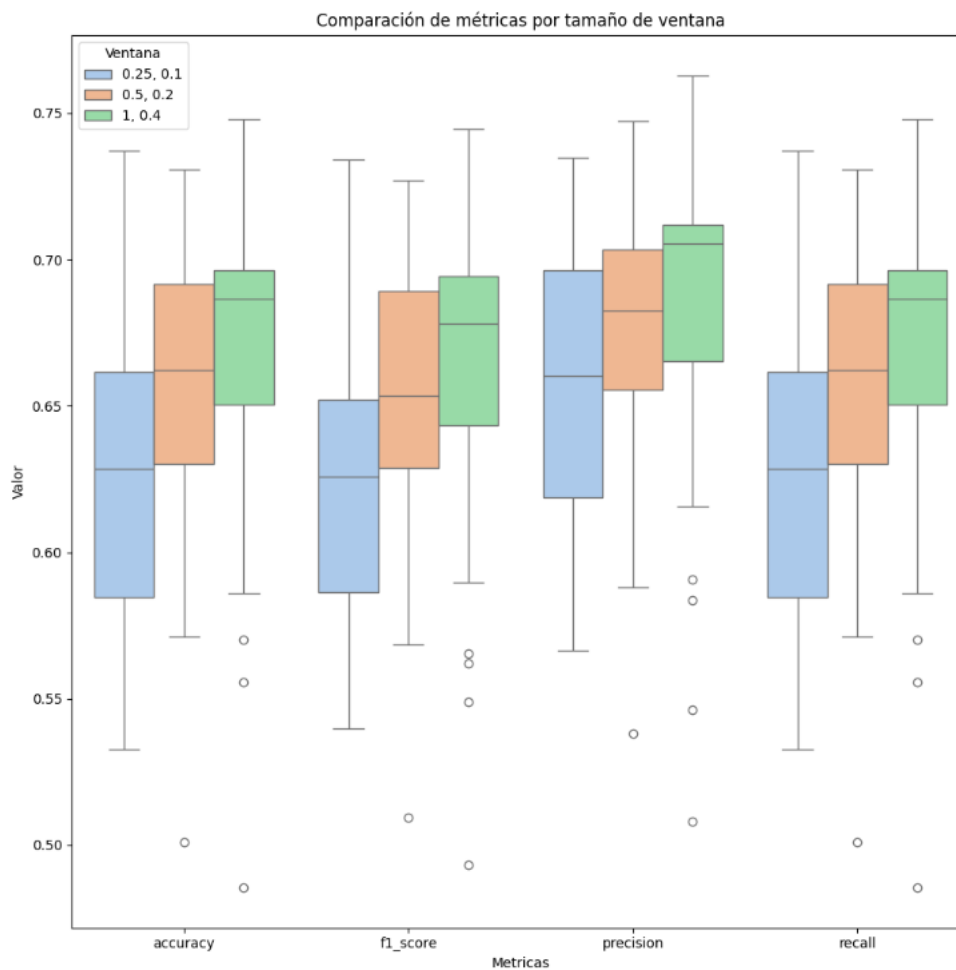


Figura 23. Comparativa de métricas de rendimiento en función del tamaño de ventana de extracción.

En base a los datos mostrados de la tabla 11 y la figura 23 se puede observar que el tamaño de ventana de tiene un impacto directo sobre el rendimiento del modelo de clasificación. En la figura 23 se puede observar como la ventana  $w_3$

## Presentación de resultados

presenta, en general, mejores resultados, con medias más elevadas en las métricas de evaluación, y menor dispersión. Por el contrario, la ventana  $w_1$  tiende a generar un rendimiento más variable y con valores centrales inferiores.

Estos resultados evidencian que el tamaño e incremento de la ventana de extracción influyen significativamente en el desempeño del modelo, lo cual es un factor crítico en tareas de clasificación de señales de sEMG.

### 4.3 Selección del mejor modelo

Una vez se ha seleccionado la mejor ventana para la extracción de datos se llevará a cabo el proceso de seleccionar el mejor modelo de clasificación.

En este apartado se presentan los resultados obtenidos tras el proceso de entrenamiento y evaluación presentados en el apartado 3.6.4 para la ventana de extracción de datos seleccionada anteriormente.

Se muestran los valores de las métricas explicadas en el apartado 3.6.5 para cada combinación de modelo y escalador. El análisis de estos resultados permite identificar la combinación que ofrece mejor equilibrio entre las métricas evaluadas y una mayor estabilidad en el rendimiento, lo que justifica su elección como movimiento más óptimo para la tarea de clasificación de movimientos de la mano.

Modelo	Escalador	Métricas de evaluación			
		Accuracy	F1-Score	Precision	Recall
Ada_boost	Standard	0,6541	0,6444	0,6894	0,6541
	Robust	0,6597	0,6536	0,6597	0,6957
	Min_max	0,6647	0,6588	0,6957	0,6647
	none	0,6510	0,6588	0,6859	0,6510
Cat_boost	Standard	0,6984	0,6964	0,7233	0,6984
	Robust	0,6962	0,6944	0,7237	0,6962
	Min_max	0,6964	0,6947	0,7240	0,6964
	none	0,6964	0,6947	0,7240	0,6964
Decision_tree	Standard	0,6157	0,6019	0,6518	0,6157
	Robust	0,6132	0,5897	0,6280	0,6132

## Presentación de resultados

	Min_max	0,5859	0,5620	0,5859	0,5907
	none	0,6642	0,6524	0,6684	0,6642
<b>Gradient_boosting</b>	Standard	0,6881	0,6782	0,7055	0,6881
	Robust	0,6881	0,6787	0,7164	0,6881
	Min_max	0,6881	0,6782	0,7055	0,6881
	none	0,6881	0,6782	0,7054	0,6881
<b>KNN</b>	Standard	0,6282	0,6309	0,6453	0,6282
	Robust	0,5703	0,5654	0,5837	0,5703
	Min_max	0,6327	0,6395	0,6555	0,6327
	none	0,4857	0,4932	0,5082	0,4857
<b>Linear_SVC</b>	Standard	0,7034	0,6983	0,7073	0,7034
	Robust	0,6488	0,6437	0,6462	0,6488
	Min_max	0,7262	0,7273	0,7304	0,7262
	none	0,5556	0,5490	0,5463	0,5556
<b>Logistic_regression</b>	Standard	0,7017	0,7273	0,7304	0,7017
	Robust	0,7479	0,7445	0,7627	0,7479
	Min_max	0,7070	0,7075	0,7086	0,7070
	none	0,7137	0,7128	0,7139	0,7137
<b>Random_forest</b>	Standard	0,6753	0,6658	0,6959	0,6753
	Robust	0,6931	0,6868	0,7114	0,6931
	Min_max	0,6625	0,6487	0,6709	0,6625
	none	0,6887	0,6812	0,7077	0,6887
<b>Xg_boost</b>	Standard	0,6859	0,6797	0,7100	0,6859
	Robust	0,6895	0,6823	0,7116	0,6895
	Min_max	0,6870	0,6810	0,120	0,6870
	none	0,6814	0,6755	0,7073	0,6814

Tabla 12. Resultados de las métricas de cada modelo con cada escalador con ventana  $w_3$ .

## Presentación de resultados

La tabla 12 muestra los resultados obtenidos por distintos algoritmos de clasificación combinados con las diferentes técnicas de escalado propuestas en el apartado 3.6.3.

Entre los modelos evaluados, *Logistic Regression* con el escalador *Robust* obtiene el mejor rendimiento global alcanzando un *accuracy* del 74,79%, un *F1-score* de 0,7445 y un *precision* del 76,27%, lo que refleja una buena capacidad de discriminación entre las tres clases diferentes.

También destaca el rendimiento de *Linear SVC* con el escalador *Min\_max* logrando un *accuracy* de 72,62%, un *F1-score* de 0,7273 y un *precision* del 73,04%, confirmando la efectividad de los clasificadores lineales.

En contraparte, modelos como *KNN* sin escalado o *Decision Tree* presentan los peores resultados, con valores de *accuracy* inferiores al 60%, lo que evidencia una menor capacidad de generalización en este contexto. Por otro lado, modelos basados en ensamblado, como *CatBoost* y *Gradient Boosting*, muestran un rendimiento sólido y consistente en todas las configuraciones de escalado, con valores de *accuracy* cercanos al 70%.

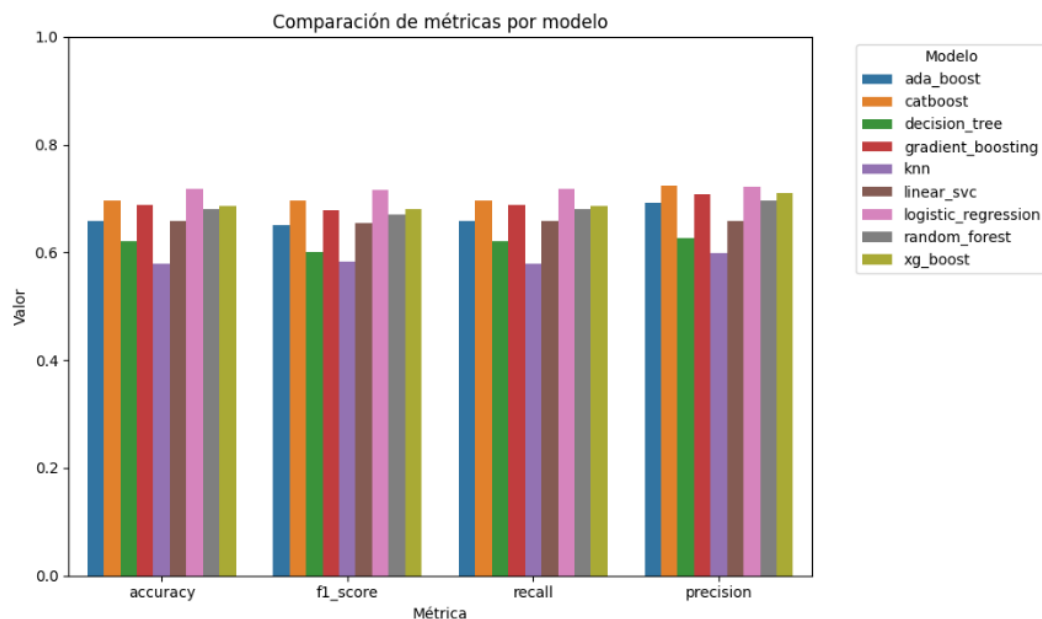


Figura 24. Diagrama de barras comparando las métricas de evaluación por modelo con ventana  $w_3$ .



## Presentación de resultados

En el diagrama mostrado en la figura 24 se muestran las diferentes métricas de evaluación promedio para cada uno de los modelos de clasificación evaluados tras aplicar diferentes escaladores.

Los valores obtenidos corresponden a la media obtenida por cada modelo, considerando todas las combinaciones de escaladores utilizadas durante la validación cruzada.

Se observa que los modelos *Logistic Regression*, *Catboost* y *Linear SVC* muestran un rendimiento superior y consistente en todas las métricas, destacando *Logistic Regression* que alcanza unos valores de *accuracy* de 71,76%, *F1-score* 0,7165 y *precision* de 72,16%.

Por el contrario, modelos como *KNN* y *Decision Tree* presentan un rendimiento más bajo, especialmente como *F1-score* y *recall*, lo que indica una menor capacidad para generalizar correctamente entre clases.

A partir de la tabla 12 y la figura 20 se puede concluir que el escalado de característica influye notablemente de algunos modelos, especialmente en los lineales como *Logistic Regression* y *Linear SVC*.

Los métodos basados en *boosting* (*Catboost*, *xgboost*, *Gradient boosting*) ofrecen buenos resultados de forma consistente, siendo menos sensibles al tipo de escalado que se le aplique.

Finalmente, modelos como *KNN* y *Decision Tree* presentan un rendimiento más limitado, probablemente debido a su menor capacidad para capturar relaciones complejas en los datos sEMG.

Para seleccionar el mejor modelo se escogerá aquel modelo que presente mayor *accuracy* en el rendimiento del entrenamiento tanto con los datos habiendo aplicado un escalado, como en el rendimiento con los valores medios al aplicar todos los métodos de escalado.

El modelo que mejor responde a estos requisitos es el modelo de *Logistic Regression* con una *accuracy* media de aproximadamente el 72% y una *accuracy* de aproximadamente el 75% aplicando el método de escalado *Robust*.

A continuación se muestra la matriz de confusión del modelo *Logistic Regression* normalizada.

## Presentación de resultados

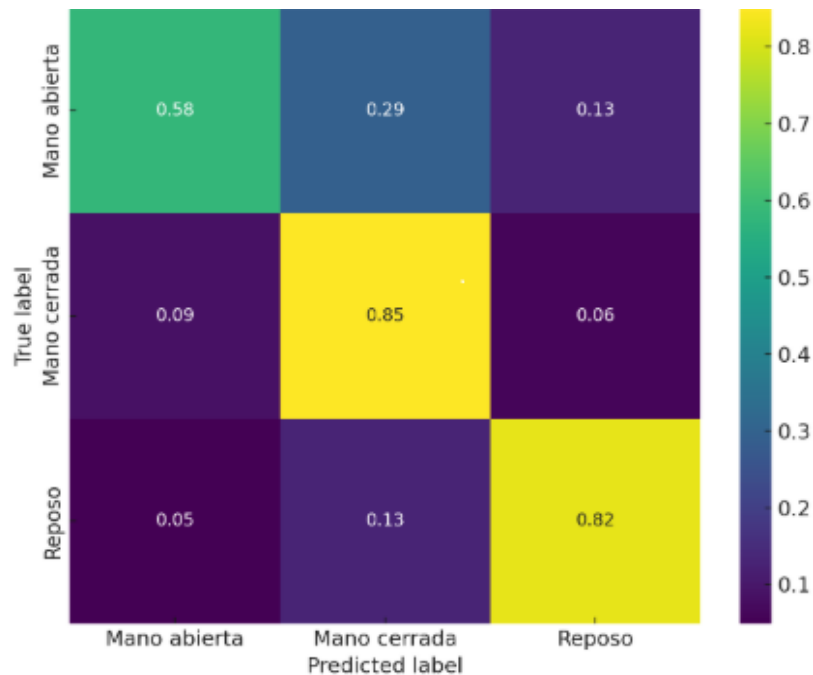


Figura 25. Matriz de confusión para el modelo *Logistic Regression*.

La matriz de confusión normalizada de la figura 25 refleja el desempeño del modelo en la clasificación de los tres estados de la mano: abierta, cerrada y reposo.

El modelo presenta su mejor rendimiento en la detección del movimiento mano cerrada con una tasa del 85%, el modelo también presenta un buen rendimiento en la detección del movimiento reposo con una tasa del 82%.

Cabe destacar que la clase mano abierta es la que presenta mayor ambigüedad, con un 53% de tasa de acierto. Es decir, un 42% de los casos siendo mal clasificados, especialmente confundidos con el estado mano cerrada un 29% de las veces, y un 13% de las veces con el estado de reposo.

Este patrón sugiere que los movimientos mano abierta y mano cerrada comparten características similares que pueden inducir al error del clasificador.

## 5 Discusión de los resultados.

---

El presente trabajo ha permitido evaluar la eficacia de distintos algoritmos de clasificación aplicados al reconocimiento de gestos de la mano mediante señales EMG y de orientación. Durante la fase de validación offline se observaron resultados sólidos en términos precisión y robustez, especialmente en contextos intra-sujeto.

Entre los modelos evaluados, los clasificadores lineales, como *Linear SVC* o *Logistic Regression* destacaron por ofrecer un rendimiento superior en comparación con modelos más complejos como los basados en árboles de decisión o ensambles. Esta tendencia sugiere que el preprocesamiento ha generado un espacio de características bien estructurado y linealmente separable. Esto hace que los modelos más simples funcionen mejor que otros más complejos, que podrían estar sobreajustado o siendo innecesariamente sofisticados para la estructura real de los datos.

Otro aspecto relevante fue el análisis del impacto del tamaño de las ventanas deslizantes utilizadas durante la segmentación de las señales. En este estudio se utilizaron tres configuraciones distintas de ventana:  $w_1$ ,  $w_2$  y  $w_3$ . Los resultados experimentales mostraron que los mejores desempeños clasificatorios se obtuvieron con las ventanas  $w_2$  y  $w_3$ , lo que sugiere que, en este caso, una mayor duración de ventana permitió capturar de forma más completa la evolución temporal de los gestos musculares.

La ventana  $w_1$ , aunque ofrece mayor resolución temporal y tiempos de respuesta más rápidos, tendieron a generar características más inestables y con menor capacidad discriminativa. Por otro lado, las ventanas más largas permiten promediar mejor la actividad muscular a lo largo del tiempo, generando representaciones más robustas de cada gesto.

En consecuencia, en este caso, un mayor tamaño de ventana mejora la separabilidad entre clases, especialmente entre gestos con señales electromiográficas similares. No obstante, esta elección también depende del equilibrio deseado entre precisión y capacidad de respuesta en tiempo real, y sería recomendable en el futuro realizar una optimización del tamaño de ventana adaptada al proceso, especialmente si se desea una implementación interactiva o en entornos clínicos.

Uno de los principales retos identificados fue la alta tasa de confusión entre los gestos mano abierta y mano cerrada. A pesar de que ambos movimientos presentan diferencias biomecánicas claras, sus patrones electromiográficos pueden solaparse, especialmente cuando la intensidad de la contracción muscular no es suficientemente distinta o el gesto no se realiza con consistencia. Además, la similitud en la activación de grupos musculares

## Discusión de los resultados

flexores y extensores durante ambas acciones podría contribuir a esta ambigüedad en la señal EMG, dificultando su correcta clasificación incluso tras la extracción de características.

Por el contrario, el gesto reposo mostró una mayor facilidad de discriminación, presumiblemente debido a la ausencia de activación muscular significativa, lo que genera un patrón de señal distintivo respecto a los gestos activos.

En términos cuantitativos, la *accuracy* alcanzada por el mejor modelo fue aproximadamente del 75%, ligeramente por debajo del objetivo inicial del 80%. Esta diferencia puede atribuirse a la variabilidad intersujeto, la complejidad inherente a las señales EMG y la similitud entre algunos de los gestos considerados. Aun así, los resultados obtenidos son prometedores y sientan una base sólida para futuras mejoras, tanto en la adquisición como en procesamiento de datos.

En caso de querer continuar con el proyecto, y mejorar los resultados, se pueden explorar las siguientes vías.

- Incorporar técnicas de *Deep Learning* podría representar una mejora sustancial respecto a los modelos tradicionales evaluados. Por ejemplo, redes convolucionales (CNN) o recurrentes (LSTM/GRU), permitiría capturar patrones temporales y no lineales difíciles de modelar con clasificadores tradicionales.
- Mejorar la calidad de adquisición de señales mediante un mapeo anatómico individualizado, además también se podría aumentar el número de individuos de los cuales recolectar datos.

Este trabajo ha sido desarrollado dentro de la división de robótica médica del Instituto Universitario ITAP (Instituto de las Tecnologías Avanzadas de la Producción).

Esta división se caracteriza por el desarrollo de tecnologías que aportan soluciones a retos quirúrgicos, sanitarios y de rehabilitación.

Por tanto, una posible línea futura para este proyecto sería la integración en un videojuego serio enfocado en la rehabilitación de pacientes con problemas neuromotores en la mano.

# 6 Conclusiones

---

El objetivo de este trabajo ha sido conseguir un algoritmo capaz de diferenciar los movimientos de mano cerrada, mano abierta y reposo a partir de las señales sEMG y de orientación obtenidas del sensor Trigno Duo de Delsys.

Para ello se ha planteado un enfoque que combina la adquisición de datos en tiempo real del sensor, la recolección de datos con varios sujetos, el procesamiento de las señales de sEMG y de orientación, extracción de características, selección de atributos relevantes y entrenamiento de modelos de clasificación, todo ello con el objetivo de lograr una solución robusta y precisa que permita diferenciar entre movimientos de la mano en condiciones reales.

Durante el proceso se ha realizado un sistema de adquisición de datos con visualización en tiempo real de las señales sEMG y de orientación y con etiquetado manual de las señales, lo que permitió construir una base de datos personalizada con información de diferentes sujetos.

A partir de esta base de datos se ha aplicado una estrategia de ventanas deslizantes solapadas con tamaños e incrementos diferentes, estableciendo la ventana de 1 s con un desplazamiento de 0,4 s como la ventana más óptima para este proyecto.

La extracción de características se realizó empleando la librería TSFEL, que proporcionó un amplio conjunto de descriptores temporales, estadísticos, espectrales y fractales, sobre los que se aplicó posteriormente una selección de características con métodos como eliminación de características por umbral de varianza, ANOVA y RFE, reduciendo la dimensionalidad a un total de 23 características y mejorando la generalización de los modelos.

Los experimentos realizados incluyeron una evaluación exhaustiva de diferentes clasificadores (*Linear SVC*, *Logistic Regression*, *Random Forest*, *Gradient Boosting*, entre otros) combinados con diferentes escaladores. La validación se efectuó mediante un esquema de *nested k-fold* para la selección de hiperparámetros, junto a un sistema LOSO para el entrenamiento del modelo, lo que garantiza una estimación fiable del rendimiento del modelo.

De todos los modelos entrenados, *Logistic Regression* con el escalador *Robust* obtuvo los mejores resultados globales, alcanzando una *accuracy* del 74,79% y un *F1-score* de 0,7445.

El análisis de la matriz de confusión reveló una buena capacidad para detectar correctamente los estados de reposo y mano cerrada, con tasas de acierto

## Conclusiones

superiores al 80%. Sin embargo, también se observó cierta confusión entre las clases mano abierta y mano cerrada.

A partir del trabajo realizado se pueden extraer una serie de conclusiones:

- Trabajar con señales EMG puede ser un proceso complicado debido su comportamiento estocástico y sensibilidad al ruido inherente a este tipo de señales. Por lo tanto se ha tenido que realizar un estudio sobre el correcto procesamiento de este tipo de señales.
- En cuanto a la implementación de modelos de *Machine Learning*, se ha aprendido que para solucionar los problemas es necesario ajustar todos los parámetros de manera adecuada y seleccionar los atributos correctamente para el buen desempeño del modelo. También se ha aprendido que la elección en las técnicas de evaluación resulta clave para obtener unos resultados fiables y no sesgados del modelo.

## 7 Referencias

---

- [1] A. Jaramillo-Yanez, M. E. Benalcázar y E. Mena-Maldonado, «Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review,» *Sensors*, vol. 20, pp. 1-36, 27 April 2020.
- [2] E. Altamira, M. Mayorga y G. Paredes, «Electromiograma (EMG),» *Dpto. de Bioingeniería, Facultad de Ingeniería*, 2017.
- [3] C. De Luca, «Electromyography.,» de *Encyclopedia of Medical Devices and Instrumentation*, John G. Webster, 2006, pp. 98-109.
- [4] A. Altamirano-Altamirano, A. Vera, R. Muñoz, L. Leija y D. Wolf, «Multichannel sEMG signal analysis using Hilbert-Huang Transform to identify time-frequency features,» *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 36, 2014.
- [5] L. Gila, A. Malanda, I. R. Carreño, J. R. Falces y J. Navallas, «Métodos de procesamiento y análisis de señales electromiográficas,» *Anales del Sistema Sanitario de Navarra*, vol. 32, nº 3, pp. 1-17, 2009.
- [6] A. G. Jaramillo y M. E. Benalcázar, «Real-time hand gesture recognition with EMG using machine learning,» *IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pp. 1-5, 2017 .
- [7] J. Yousefi y A. Hamilton-Wright, «Characterizing EMG data using machine-learning tools,» *Computers in Biology and Medicine*, vol. 51, pp. 1-13, 2014.
- [8] M. J. Zwarts, G. Drost y D. F. Stegeman, «Recent progress in the diagnostic use of surface EMG for neurological diseases,» *Journal of Electromyography and Kinesiology*, vol. 10, nº 5, pp. 287-291, 2000.
- [9] J.-Y. Hogrel, «Clinical applications of surface electromyography in neuromuscular disorders,» *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 35, nº 2, pp. 59-71, 2005.
- [10] A. J. Haig, J. B. Gelblum, J. J. Rechten, PHD y A. J. Gitter, «Technology assessment: The use of surface EMG in the Diagnosis and Treatment of

## Referencias

- Nerve and Muscle Disorders,» *American Association of Neuromuscular and Electrodiagnostic Medicine (AANEM)*, vol. 19, nº 3, pp. 392-295, 1996.
- [11] M. R. Ahsan, M. I. Ibrahimy y O. O.Khalifa, «EMG Signal Classification for Human Computer Interaction: A Review,» *European Journal of Scientific Research*, vol. 33, nº 3, pp. 480-501, 2009.
- [12] N. Wang, K. Lao y X. Zhang, «Design and Myoelectric Control of an Anthropomorphic Prosthetic Hand,» *Journal of Bionic Engineering*, vol. 14, nº 1, pp. 47-59, 2017.
- [13] Z. Xu, X. Chen, W. Wen-hui, Y. Ji-hai, V. Lantz y W. Kong-qiao, «Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors,» *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 1-6, 08 February 2009.
- [14] Sahin, C. Savur y Ferat, «Real-Time American Sign Language Recognition System by Using Surface EMG Signal,» *14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 497-502, 2015.
- [15] Delsys, «delsys.com,» Delsys, [En línea]. Available: <https://delsys.com/>. [Último acceso: June 2025].
- [16] Delsys, «Quattro EMG Sensor, Duo EMG Sensor, Mini EMG Sensor User's Guide,» 2019.
- [17] Delsys, «AeroPy Programmer's Guide,» 2020.
- [18] C. J. D. Luca, L. Donald Gilmore, M. Kuznetsov y S. H. Roy, «Filtering the surface EMG signal: Movement artifact and baseline noise contamination,» *Journal of Biomechanics*, vol. 43, nº 8, pp. 1573-1579, 2010.
- [19] J. Chen, Y. Sun, S. Sun y Z. Yao, «Reducing Power Line Interference from sEMG Signals Based on Synchrosqueezed Wavelet Transform,» *Sensors*, vol. 23, nº 5182, pp. 1-20, 2023.
- [20] N. J.Jarque-Bou, J. L. Sancho-Bru y M. Vergara, «A Systematic Review of EMG Applications for the Characterization of Forearm and Hand Muscle Activity during Activities of Daily Living: Results, Challenges, and Open Issues,» *Sensors*, vol. 21, nº 3035, pp. 1-36, 2021.



## Referencias

- [21] i. B. Freriks y D. i. H. Hermens, «SENIAM,» [En línea]. Available: <http://www.seniam.org/>. [Último acceso: July 2025].
- [22] White, J. K. Hammond y P. R., «THE ANALYSIS OF NON-STATIONARY SIGNALS USING TIME–FREQUENCY METHODS,» *Journal of Sound and Vibration*, vol. 190, n° 3, pp. 419-447, 29 February 1996.
- [23] H. Ashraf, A. Waris, S. O. Gilani, A. S. Kashif, M. Jamil, M. Jochumsen y I. K. Niazi, «Evaluation of windowing techniques for intramuscular EMG-based diagnostic, rehabilitative and assistive devices,» *Journal of Neural Engineering*, vol. 18, n° 1, p. 016017, 2021.
- [24] M. A. Oskoei, S. Member, a. H. Hu y S. Member, «Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb,» *IEEE Transactions on Biomedical Engineering*, vol. 55, n° 8, pp. 1956-1965, 2008.
- [25] A. M. Moslhi, H. H. Aly y M. ElMessiery, «The Impact of Feature Extraction on Classification Accuracy Examined by Employing a Signal Transformer to Classify Hand Gestures Using Surface Electromyography Signals,» *Sensors*, vol. 24, n° 4, p. 1259, 2024.
- [26] N. Nazmi, M. A. A. Rahman y S.-I. Yamamoto, «A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions,» *Sensors*, vol. 16, n° 8, p. 1304, 2016.
- [27] F. Portugal, «tsfel.com,» [En línea]. Available: <https://tsfel.readthedocs.io/en/latest/index.html>. [Último acceso: June 2025].
- [28] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Lui, T. Schultz y H. Gamboa, «TSFEL: Time Series Feature Extraction Library,» *SoftwareX*, vol. 11, p. 100456, 2020.
- [29] J. J. A. M. Junior, M. L. Freitas, H. V. Siqueira, A. E. Lazzaretti, S. F. Pichorim y S. L. Stevan, «Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach,» *Biomedical Signal Processing and Control*, vol. 59, p. 101920, 2019.

## Referencias

- [30] AprendeIA, «aprendeia.com,» [En línea]. Available: <https://aprendeia.com/2019/01/04/metodos-de-seleccion-de-caracteristicas-machine-learning/>. [Último acceso: June 2025].
- [31] 4Geeks, «Selección de características, 4geeks.com,» [En línea]. Available: <https://4geeks.com/es/lesson/seleccion-de-caracteristicas>. [Último acceso: July 2025].
- [32] S.-l. developers, «Feature selection, removing features with low variance - scikit.learn.org,» [En línea]. Available: [https://scikit-learn.org/stable/modules/feature\\_selection.html#removing-features-with-low-variance](https://scikit-learn.org/stable/modules/feature_selection.html#removing-features-with-low-variance). [Último acceso: June 2025].
- [33] S.-l. developers, «Feature selection, f\_classif - scikit-learn.org,» [En línea]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.f\\_classif.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html). [Último acceso: June 2025].
- [34] S.-l. developers, «Feature selection, chi2 - scikit-learn.org,» [En línea]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.chi2.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html). [Último acceso: June 2025].
- [35] G. Chandrashekar y F. Sahin, «A survey on feature selection methods,» *Computers and Electrical Engineering*, vol. 20, nº 1, pp. 16-28, 2013.
- [36] A. desconocido, «Fritz AI,» 21 September 2023. [En línea]. Available: <https://fritz.ai/hands-on-with-feature-selection-techniques-embedded-methods/>. [Último acceso: June 2025].
- [37] A. Singh, «Medium.com,» 13 November 2024. [En línea]. Available: <https://medium.com/@abhaysingh71711/efficient-feature-selection-a-guide-to-embedded-methods-for-optimized-machine-learning-45ea3c83e622>. [Último acceso: July 2025].
- [38] S.-l. developers, «Linear SVC - scikit-learn.org,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>. [Último acceso: June 2025].

## Referencias

- [39] S. Ji y Y. Xie, «Logistic Regression: From Binary to Multi-Class,» A\&M University: College Station, Texas , 2024.
- [40] IBM, [En línea]. Available: [https://www.ibm.com/think/topics/knn?mhsrc=ibmsearch\\_a&mhq=knn](https://www.ibm.com/think/topics/knn?mhsrc=ibmsearch_a&mhq=knn). [Último acceso: June 2025].
- [41] R. Sebastian, «STAT: 451 Machine Learning Lecture Notes,» University of Wisconsin, Madison, 2020.
- [42] IBM, [En línea]. Available: <https://www.ibm.com/mx-es/think/topics/random-forest#:~:text=El%20bosque%20aleatorio%20es%20un,problemas%20de%20clasificaci%C3%B3n%20y%20regresi%C3%B3n>. [Último acceso: June 2025].
- [43] Google for developers, [En línea]. Available: <https://developers.google.com/machine-learning/decision-forests/intro-to-gbdt?hl=es-419#:~:text=In%20gradient%20boosting%2C%20at%20each,error%20of%20the%20strong%20model>. [Último acceso: June 2025].
- [44] T. Chen y C. Guestrin, «XGBoost: A Scalable Tree Boosting System,» de *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 785-794.
- [45] P. Liudmila, G. Gleb, V. Aleksandr, D. A. Veronika y G. Andrey, «CatBoost: unbiased boosting with categorical features,» *Advanced in Neural Information Processing System (NeurIPS)*, vol. 31, 2018.
- [46] X. H. Cao, I. Stojkovic y Z. Obradovic, «A robust data scaling algorithm to improve classification accuracies in biomedical data,» *BMC Bioinformatics*, vol. 17, n° 359, pp. 1-10, 2016.
- [47] J. Brownlee, «machinelearningmastery,» 4 October 2023. [En línea]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/>. [Último acceso: June 2025].
- [48] J. Brownlee, «machinelearningmastery,» 20 November 2021. [En línea]. Available: <https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/>. [Último acceso: June 2025].

## Referencias

- [49] J. Brownlee, «machinelearningmastery,» 26 August 2020. [En línea]. Available: <https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/>. [Último acceso: June 2025].
- [50] M. P. Pauli, P. Constantin y G. Martin, «Balanced Leave-One-Subject-Out Cross-Validation for Microsleep Classification,» *Current Directions in Biomedical Engineering*, vol. 7, n° 2, pp. 147-150, 2021.
- [51] S. Saeb, L. Lonini, A. Jayaraman y D. C. Mohr, «The need to approximate the use-case in clinical machine learning,» *GigaScience*, vol. 6, pp. 1-9, 2017.
- [52] Hossin, M. a. Sulaiman y M.N., «Data Classification Evaluations,» *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 5, n° 2, p. 1, March 2015.
- [53] A. J. I. Barrios, «Health Big Data,» 26 Julio 2019. [En línea]. Available: <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>. [Último acceso: July 2025].