
Diseño e implementación de una interfaz biomédica y sensores inerciales para el control de actuadores en tiempo real

Santiago José Rivera Lemus



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de una interfaz biomédica y sensores
inerciales para el control de actuadores en tiempo real**

Trabajo de graduación presentado por Santiago José Rivera Lemus para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



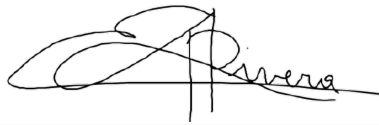
**Diseño e implementación de una interfaz biomédica y sensores
inerciales para el control de actuadores en tiempo real**

Trabajo de graduación presentado por Santiago José Rivera Lemus para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025


Vo.Bo.:



(f)

Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:



(f)

Dr. Luis Alberto Rivera Estrada



(f)

Ing. Kurt Emmanuel Kellner Juárez



(f)

MSc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 23 de junio de 2025.

Esta investigación nació de mi interés en el campo de la medicina y la historia de mi familia en este ámbito. A lo largo de mi vida, he estado expuesto a la medicina en Guatemala, y poder aplicar mis conocimientos como Ingeniero Mecatrónico en este campo fue una experiencia muy significativa durante el tiempo que duró el proyecto.

Quisiera agradecer a todas las personas que me acompañaron durante este recorrido universitario, especialmente a mi familia, Nancy y Jose, quienes me vieron enfrentar dificultades y nunca dudaron en brindarme su apoyo para seguir adelante con mis trabajos. A mis tíos Byron, Patty y Quique, quienes siempre estuvieron a mi lado, dándome aliento y sabias palabras en los momentos clave de mi vida. Y, por supuesto, a mi padre, Carlos, quien me observa desde el cielo y cuyo cariño y apoyo siempre sentí a lo largo de este trayecto.

También quiero agradecer a mis compañeros, con quienes compartí risas, esfuerzos y desafíos durante estos cinco años, y a todos los catedráticos que contribuyeron al fomento del aprendizaje en esta maravillosa carrera. Especialmente, deseo expresar mi agradecimiento al Dr. Luis Alberto Rivera Estrada, mi asesor, cuyo conocimiento, orientación y, en ocasiones, oportunos regaños, fueron fundamentales para la culminación exitosa de este trabajo.

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiii
Resumen	xiv
Abstract	xv
1. Introducción	1
2. Antecedentes	2
2.1. Interfaces biomédicas	2
2.2. Interfaces biomédicas desarrolladas en la UVG	3
2.3. Control de actuadores por medio de señales bioeléctricas	4
2.4. Mano animatrónica UVG	5
3. Justificación	8
4. Objetivos	10
4.1. Objetivo general	10
4.2. Objetivos específicos	10
5. Alcance	11
6. Marco teórico	12
6.1. Señales bioeléctricas	12
6.2. Señales EMG	12
6.3. Características en el tiempo	13
6.3.1. Zero crossing	13
6.3.2. Media del valor absoluto	13
6.3.3. Media cuadrática	13
6.3.4. Varianza	13

6.3.5.	Amplitud de Willison	14
6.3.6.	Curtosis	14
6.4.	BITalino	14
6.5.	Myoware	15
6.6.	Aprendizaje automático	16
6.6.1.	Aprendizaje no supervisado	16
6.6.2.	Aprendizaje supervisado	16
6.7.	Interfaz HMI (Human-Machine Interface)	20
6.8.	Sensores inerciales	20
6.9.	Actuadores	20
6.10.	Prótesis	21
7.	Recolección de señales EMG	22
7.1.	Dispositivos de adquisición para señales EMG	22
7.1.1.	Dispositivo BITalino	22
7.1.2.	Dispositivo MyoWare	23
7.2.	Algoritmos de adquisición	23
7.2.1.	<i>MATLAB Support for BITalino Biosignal Devices Toolbox</i>	24
7.2.2.	<i>BITalino toolbox</i>	24
7.3.	Algoritmo de captura en tiempo real	26
7.3.1.	Implementación BITalino con MyoWare	26
7.3.2.	Implementación BITalino con detección automática	28
7.4.	Procesamiento de señales	30
7.5.	Interfaz de recolección de señales	30
7.5.1.	Primera iteración del módulo	31
7.5.2.	Segunda iteración de interfaz	34
8.	Clasificadores de señales EMG	37
8.1.	Características utilizadas	37
8.2.	Gestos utilizados	38
8.2.1.	Colecciones recopiladas con un sensor	38
8.2.2.	Colección recopilada con dos sensores	40
8.3.	<i>Classification Learner</i>	41
8.4.	Resultados de clasificación de 2 gestos con un sensor	42
8.4.1.	SVM	42
8.4.2.	ANN	46
8.4.3.	KNN	49
8.5.	Resultados de clasificación de 3 gestos con un sensor	53
8.5.1.	SVM	53
8.5.2.	ANN	56
8.5.3.	KNN	60
8.6.	Resultados de clasificación de 3 gestos con dos sensores	63
8.6.1.	SVM	63
8.6.2.	ANN	67
8.6.3.	KNN	70
8.7.	Resultados obtenidos con participantes externos	74
8.7.1.	Resultados participante 1	74
8.7.2.	Resultados participante 2	76

8.7.3. Resultados participante 3	78
8.7.4. Resultados participante 4	80
8.7.5. Análisis de resultados generales	82
9. Sistema de sensores inerciales	83
9.1. Características del IMU	83
9.2. Algoritmo de lectura	84
9.3. Calibración del sensor	84
9.4. Algoritmo de control	84
10. Mano antropomórfica UVG	87
10.1. Características de mano antropomórfica	87
10.2. Algoritmos de control	87
10.3. Movimientos predeterminados	90
11. Interfaz de control de mano antropomórfica	94
11.1. Implementación de clasificadores	95
11.2. Funcionalidades	96
11.2.1. Modo binario	97
11.2.2. Modo 1 sensor	98
11.2.3. Modo 2 sensores	99
12. Conclusiones	102
13. Recomendaciones	103
14. Bibliografía	104
15. Anexos	107
15.1. Anexo A: Manual de usuario	107

Lista de figuras

1.	Primera fase de la interfaz UVG [5].	3
2.	Segunda fase de interfaz [6].	4
3.	Placa Muscle Sensor v3 [9].	5
4.	Módulo Myoware [10].	6
5.	Mano animatrónica antropomórfica [12].	7
6.	Ejemplo de señal EMG [16].	13
7.	Kit de desarrollo BITalino [18].	15
8.	Muscle Sensor MyoWare 2.0 [19].	15
9.	Representación de SVM [24].	17
10.	Representación de funciones de Kernel y representación de hiperplanos [24].	17
11.	Representación SVM <i>One vs All</i> [27].	18
12.	Representación SVM <i>One vs One</i> [27].	18
13.	Representación del algoritmo KNN [30].	19
14.	Representación de capas de una Red neuronal artificial simple [23].	20
15.	Prótesis realizada en el antiguo Egipto [34].	21
16.	Esquema de circuito utilizado en el sensor MyoWare [19].	23
17.	Diagrama de flujo de algoritmo de adquisición en tiempo real con MyoWare.	27
18.	Diagrama de flujo de algoritmo de adquisición en tiempo real con algoritmo de detección automática.	29
19.	Módulo de recolección de señales EMG de proyectos anteriores [6].	31
20.	Segunda versión del módulo de recolección de señales EMG.	32
21.	Diagrama de flujo de algoritmo de módulo de recolección EMG.	33
22.	Diagrama de flujo de algoritmo de segunda iteración de módulo de recolección EMG.	35
23.	Segunda versión de interfaz de recolección de señales EMG.	36
24.	Posición de los electrodos para la captura de los 3 gestos.	38
25.	Gesto 1: Movimiento de puño hacia arriba.	39
26.	Gesto 2: Movimiento de “ola’ con los dedos de la mano.	39
27.	Gesto 3: Movimiento muñeca para arriba dos veces.	40
28.	Posición de electrodos de segundo sensor.	41

29.	Gesto 3: Puño hacia abajo.	41
30.	Matriz de confusión con etapa de validación para SVM lineal con dos gestos.	43
31.	Matriz de confusión con etapa de prueba para SVM lineal con dos gestos.	43
32.	Matriz de confusión con etapa de validación para SVM cuadrática con dos gestos.	44
33.	Matriz de confusión con etapa de prueba para SVM cuadrática con dos gestos.	44
34.	Matriz de confusión con etapa de validación para SVM cúbica con dos gestos.	45
35.	Matriz de confusión con etapa de prueba para SVM cúbica con dos gestos.	45
36.	Matriz de confusión con etapa de validación para ANN de tres capas con dos gestos.	46
37.	Matriz de confusión con etapa de prueba para ANN de tres capas con dos gestos.	47
38.	Matriz de confusión con etapa de validación para ANN de dos capas con dos gestos.	47
39.	Matriz de confusión con etapa de prueba para ANN de dos capas con dos gestos.	48
40.	Matriz de confusión con etapa de validación para ANN ancha con dos gestos.	48
41.	Matriz de confusión con etapa de validación para ANN ancha con dos gestos.	49
42.	Matriz de confusión con etapa de validación para KNN pesada con dos gestos.	50
43.	Matriz de confusión con etapa de prueba para KNN pesada con dos gestos.	50
44.	Matriz de confusión con etapa de validación para KNN fina con dos gestos.	51
45.	Matriz de confusión con etapa de prueba para KNN fina con dos gestos.	51
46.	Matriz de confusión con etapa de validación para KNN cúbica con dos gestos.	52
47.	Matriz de confusión con etapa de prueba para KNN cúbica con dos gestos.	52
48.	Matriz de confusión con etapa de validación para SVM lineal.	53
49.	Matriz de confusión con etapa de prueba para SVM lineal con tres gestos.	54
50.	Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.	54
51.	Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos.	55
52.	Matriz de confusión con etapa de validación para SVM cúbica con tres gestos.	55
53.	Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos.	56
54.	Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.	57
55.	Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.	57
56.	Matriz de confusión con etapa de validación para ANN de dos capas con tres gestos.	58
57.	Matriz de confusión con etapa de prueba para ANN de dos capas con tres gestos.	58
58.	Matriz de confusión con etapa de validación para ANN ancha con tres gestos.	59
59.	Matriz de confusión con etapa de prueba para ANN ancha con tres gestos.	59
60.	Matriz de confusión con etapa de validación para KNN pesada con tres gestos.	60
61.	Matriz de confusión con etapa de prueba para KNN pesada con tres gestos.	61
62.	Matriz de confusión con etapa de validación para KNN fina con tres gestos.	61
63.	Matriz de confusión con etapa de prueba para KNN fina con tres gestos.	62
64.	Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.	62
65.	Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos.	63
66.	Matriz de confusión con etapa de validación para SVM lineal con tres gestos.	64

67.	Matriz de confusión con etapa de prueba para SVM lineal con tres gestos. . .	64
68.	Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.	65
69.	Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos.	65
70.	Matriz de confusión con etapa de validación para SVM cúbica con tres gestos.	66
71.	Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos. . .	66
72.	Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.	67
73.	Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.	68
74.	Matriz de confusión con etapa de validación para ANN de dos capas con dos gestos.	68
75.	Matriz de confusión con etapa de prueba para ANN de dos capas con dos gestos.	69
76.	Matriz de confusión con etapa de validación para ANN ancha con tres gestos.	69
77.	Matriz de confusión con etapa de prueba para ANN ancha con tres gestos. . .	70
78.	Matriz de confusión con etapa de validación para KNN pesada con tres gestos.	71
79.	Matriz de confusión con etapa de prueba para KNN pesada con tres gestos. .	71
80.	Matriz de confusión con etapa de validación para KNN fina con tres gestos. .	72
81.	Matriz de confusión con etapa de prueba para KNN fina con tres gestos. . . .	72
82.	Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.	73
83.	Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos. .	73
84.	Matriz de confusión de modelo SVM con conjunto de prueba de participante 1.	75
85.	Matriz de confusión de modelo ANN con conjunto de prueba de participante 1.	75
86.	Matriz de confusión de modelo KNN con conjunto de prueba de participante 1.	76
87.	Matriz de confusión de modelo SVM con conjunto de prueba de participante 2.	77
88.	Matriz de confusión de modelo ANN con conjunto de prueba de participante 2.	77
89.	Matriz de confusión de modelo KNN con conjunto de prueba de participante 2.	78
90.	Matriz de confusión de modelo SVM con conjunto de prueba de participante 3.	79
91.	Matriz de confusión de modelo ANN con conjunto de prueba de participante 3.	79
92.	Matriz de confusión de modelo KNN con conjunto de prueba de participante 1.	80
93.	Matriz de confusión de modelo SVM con conjunto de prueba de participante 4.	81
94.	Matriz de confusión de modelo ANN con conjunto de prueba de participante 4.	81
95.	Matriz de confusión de modelo KNN con conjunto de prueba de participante 4.	82
96.	Diagrama de flujo de algoritmo implementado para lectura y mapeo de MPU6050.	86
97.	Diagrama de flujo de algoritmo de control de la mano antropomórfica.	89
98.	Conteo del número 0 en mano antropomórfica.	90
99.	Conteo del número 1 en mano antropomórfica.	91
100.	Conteo del número 2 en mano antropomórfica.	91
101.	Conteo del número 3 en mano antropomórfica.	92
102.	Conteo del número 4 en mano antropomórfica.	92
103.	Conteo del número 5 en mano antropomórfica.	93
104.	Interfaz de control de mano antropomórfica UVG, pestaña de instrucciones. .	95
105.	Interfaz de control de mano antropomórfica UVG, pestaña de entrenamiento.	96
106.	Interfaz de control de mano antropomórfica UVG, pestaña de control.	96

107. Posición y conexión de sensores para modo Binario.	97
108. Interfaz de control de mano antropomórfica UVG, en funcionalidad de modo binario.	98
109. Interfaz de control de mano antropomórfica UVG, clasificando señal clase 2 en modo de un sensor.	99
110. Interfaz de control de mano antropomórfica UVG, clasificando señal clase 1 en modo de un sensor.	99
111. Interfaz de control de mano antropomórfica UVG, clasificando señal clase 1 en modo de dos sensor.	100
112. Interfaz de control de mano antropomórfica UVG, clasificando señal clase 2 en modo de dos sensor.	100
113. Interfaz de control de mano antropomórfica UVG, clasificando señal clase 3 en modo de dos sensor.	101

Lista de cuadros

1.	Resultados de las SVM con 2 gestos, recopilados en 1 sensor	42
2.	Resultados de ANN con 2 gestos, recopilados en 1 sensor	46
3.	Resultados de KNN con 2 gestos, recopilados en 1 sensor	49
4.	Resultados de las SVM con 3 gestos, recopilados en 1 sensor	53
5.	Resultados de ANN con 3 gestos, recopilados en 1 sensor	56
6.	Resultados de KNN con 3 gestos, recopilados en 1 sensor	60
7.	Resultados de las SVM con 3 gestos, recopilados en 2 sensores	63
8.	Resultados de ANN con 3 gestos, recopilados en 2 sensores	67
9.	Resultados de KNN con 3 gestos, recopilados en 2 sensores	70
10.	Resultados de los clasificadores con 2 gestos, recopilados de participante externo 1	74
11.	Resultados de los clasificadores con 2 gestos, recopilados de participante externo 2	76
12.	Resultados de los clasificadores con 2 gestos, recopilados de participante externo 3	78
13.	Resultados de los clasificadores con 2 gestos, recopilados de participante externo 4	80

Este trabajo se centra en el diseño y desarrollo de una interfaz biomédica, con distintos módulos: uno para la recolección de señales electromiográficas (EMG) y otro para el control en tiempo real basado en señales EMG y un sistema de sensores inerciales. Las interfaces fueron desarrolladas con el objetivo de implementar un control eficiente de actuadores o sistemas robóticos. Aunque se contaba con algoritmos previos para la recolección de señales, se optó por mejorarlos para permitir la adquisición en tiempo real. Para ello, se implementó un algoritmo de segmentación de señales y se calcularon diversas características, lo que facilitó la implementación y el entrenamiento de varios modelos de aprendizaje automático, como Máquinas de Soporte de Vectores (SVM), Redes Neuronales Artificiales (RNA) y el modelo de *K-Nearest Neighbor* (KNN). Se realizaron diversas pruebas con estos algoritmos para asegurar una clasificación adecuada de las señales. Además, se incorporó un sistema de sensores inerciales, en particular el MPU6050, para combinar las fortalezas de ambos enfoques y lograr un control eficiente de los actuadores. El sistema de control seleccionado fue una mano antropomórfica diseñada en la Universidad del Valle de Guatemala, con la que la interfaz demostró una integración perfecta para su control.

Uno de los objetivos principales del proyecto fue la implementación de modelos de aprendizaje automático. Gracias a la selección de características, como el *zero crossing* y la media del valor absoluto, combinadas con la segmentación de las señales en diferentes tramos, se lograron porcentajes de clasificación superiores al 90 %, alcanzando hasta un 98 % en los mejores casos.

El control de la mano animatrónica fue especialmente eficaz y rápido, permitiendo un control en tiempo real que hizo que la mano se moviera casi instantáneamente al detectar una señal relacionada con el movimiento. Además, el sistema de sensores inerciales proporcionó un control eficiente y constante, reflejando de manera precisa los movimientos observados en la mano.

En el futuro, se espera aprovechar la versatilidad de la interfaz para lograr un control más eficaz y natural al implementarla en prótesis de miembro superior. Esto permitirá una integración más fluida con los movimientos del usuario, mejorando la funcionalidad y la adaptabilidad de la prótesis para ofrecer una experiencia más intuitiva y realista.

This work focuses on the design and development of two biomedical interfaces: one for collecting electromyographic (EMG) signals and another for real-time control based on EMG signals and an inertial sensor system. The interfaces were developed with the aim of implementing efficient control of actuators or robotic systems. Although there were pre-existing algorithms for signal collection, it was decided to improve them to enable real-time acquisition. To achieve this, a signal segmentation algorithm was implemented, and various features were calculated, facilitating the implementation and training of several machine learning models, such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and the K-Nearest Neighbor (KNN) model. Several tests were performed with these algorithms to ensure proper signal classification. In addition, an inertial sensor system, particularly the MPU6050, was integrated to combine the strengths of both approaches and achieve efficient control of the actuators. The selected control system was an anthropomorphic hand designed at Universidad del Valle de Guatemala, with which the interface demonstrated perfect integration for its control.

One of the main objectives of the project was the implementation of machine learning models. Thanks to the feature selection, such as zero crossing and the mean absolute value, combined with signal segmentation into different segments, classification rates above 90 % were achieved, reaching up to 98 % in the best cases.

The control of the animatronic hand was particularly effective and fast, allowing for real-time control that made the hand move almost instantly upon detecting a signal related to movement. Additionally, the inertial sensor system provided efficient and constant control, accurately reflecting the movements observed in the hand.

In the future, it is expected to leverage the versatility of the interface to achieve more effective and natural control by implementing it in upper-limb prosthetics. This will allow for smoother integration with the user's movements, enhancing the functionality and adaptability of the prosthetic to provide a more intuitive and realistic experience.

En los últimos años, la investigación y desarrollo de prótesis se han centrado en el estudio e implementación de señales bioeléctricas para su control. Las señales electromiográficas (EMG) y su integración con algoritmos de aprendizaje automático han demostrado ser beneficiosas para mejorar la calidad de vida de personas afectadas que requieren este tipo de tecnología.

Este trabajo propone el desarrollo de una interfaz capaz de interpretar señales EMG mediante clasificadores de aprendizaje automático, así como la implementación de sensores inerciales para controlar actuadores, enfocándose en el control de prótesis.

El desarrollo de la interfaz se llevó a cabo en varias etapas. Se realizaron pruebas con diferentes sensores para optimizar la extracción de señales EMG y se implementaron algoritmos de clasificación con el fin de encontrar el clasificador más adecuado para las aplicaciones deseadas. Además, se utilizaron fundamentos de proyectos desarrollados para obtener resultados más efectivos y prácticos.

El objetivo principal fue mejorar los resultados de fases anteriores, implementando un control en tiempo real y añadiendo sensores inerciales, tanto para la clasificación como para el control de movimientos. El documento detalla la metodología empleada para la clasificación y control de una mano antropomórfica, así como la validación de los resultados a través del control de dicha prótesis.

2.1. Interfaces biomédicas

En tiempos recientes, el estudio de las señales del cuerpo ha sido un campo explorado para su uso dentro de diferentes aplicaciones para ayudar y mejorar la vida de las personas que lo lleguen a necesitar. Ya sea dentro del campo médico, con la creación de prótesis o en el campo de la robótica, para lograr el control de diferentes actuadores y robots en general, para tener un control sin necesidad de un mando físico, más bien un control con las señales del cuerpo.

En la Universidad del Norte de México se desarrolló un sistema con la capacidad de adquirir y visualizar señales EMG [1]. Por medio del uso de electrodos y ciertos microcontroladores se logró la captación de las señales, por medio de un procedimiento específico de adquisición utilizando circuitos de amplificación y diferentes filtros para lograr tener una señal limpia. Además, el uso de una conversión A/D y el uso de USART fueron implementados para una comunicación estable con un ordenador. La interfaz utilizó LabVIEW para lograr mostrar visualmente las señales EMG producidas por una persona.

En la Universidad de Cauca se realizó el desarrollo de una interfaz gráfica Hombre – Máquina de recolección y diferenciación de señales EMG por medio de tres diodos y diferentes sensores para lograr, analizar y procesar las señales [2]. En el desarrollo se utilizó Aprendizaje Automático para lograr diferenciar las señales por el uso de características. Por ejemplo, en este proyecto se utilizaron las características de media, varianza, histograma de amplitudes y el valor RMS. Los algoritmos de control y la creación del sistema se hicieron utilizando LabVIEW 6i de National Instruments, donde se logró el control de diferentes aplicaciones como lo son un brazo robótico y su simulación, además del control de un robot móvil.

En 2019 en la Universidad Nacional de La Plata se creó una interfaz hombre – máquina con recolecciones en tiempo real de las señales y portátil [3]. Para conseguir una buena adquisición de señales, el proyecto seleccionó el microcontrolador MSP430G2553 de Texas

Instruments y un circuito integrado ADS1299 para la etapa de alimentación del sistema. Los resultados obtenidos fueron satisfactorios y se logró la creación de un dispositivo portátil con vida útil de aproximadamente 38 horas.

2.2. Interfaces biomédicas desarrolladas en la UVG

En la Universidad del Valle de Guatemala se han estado desarrollando herramientas en los últimos años para lograr una mejor obtención y análisis de señales bioeléctricas. Empezando en el año 2020, con el trabajo de María Fernanda Girón [4], se implementaron diferentes algoritmos de reconocimiento de señales electromiográficas (EMG) para su implementación en el control de sistemas robóticos. Realizado por medio de algoritmos de Aprendizaje Automático con Maquinas de vectores de soporte (SVM) y Redes neuronales (RNAs), obteniendo rendimientos de clasificación superiores al 80 %.

En ese mismo año Rodrigo Balda [5], desarrolló una primera iteración de una interfaz con la habilidad de manipular sistemas robóticos con el uso de señales electroencefalográficas obteniendo resultados similares al trabajo de Girón, con porcentajes superiores a un 80 %.

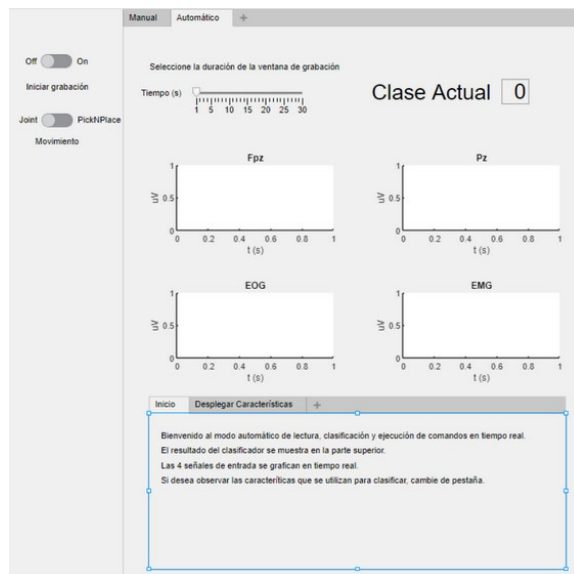


Figura 1: Primera fase de la interfaz UVG [5].

En el año 2021, se continuó la investigación y mejora de los trabajos previos dentro de la Universidad del Valle de Guatemala con el trabajo de Roberto Cáceres [6]. En ese trabajo se implementó una unificación con los trabajos previos creando una sola interfaz con el mismo objetivo de control de sistemas robóticos con la implementación de algoritmos de aprendizaje automático.

En la investigación se obtuvieron rendimientos superiores a trabajos anteriores dentro de la universidad utilizando los dos tipos de señales bioeléctricas siendo las señales EEG como señales auxiliares. El rendimiento varió obteniendo un porcentaje de 85 % solo con características de la señal EMG, incrementando a un 87 % con el uso de las señales EEG como auxiliares.

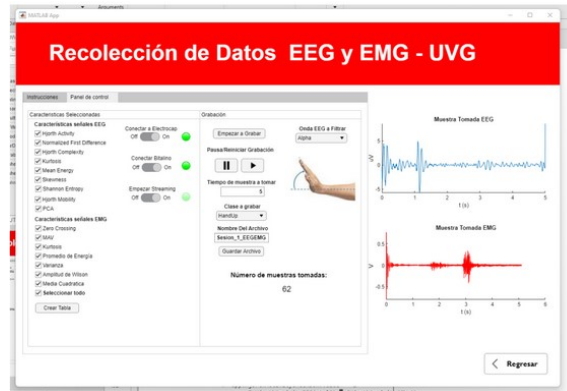


Figura 2: Segunda fase de interfaz [6].

La investigación dentro de la universidad continuó en el año 2023 con el trabajo de Fernando Sandoval [7] solventando una mejora en la disminución del tiempo computacional de los algoritmos y de extracción de señales con un cambio en el dispositivo optando por el sistema BIOPAC y con la base de datos del centro HUMANA. La mejora de los algoritmos llegó a un porcentaje de clasificación positivo mostrando un rendimiento superior a un 90 %.

Dentro de la simulación dentro de la interfaz a pesar de lograr obtener resultados positivos se mostraron porcentajes de clasificaciones erróneas, gracias a una variación en la clasificación de señales de un mismo sujeto las cuales fueron tomadas en diferentes días, esto siendo más pronunciado en las señales EEG.

En los proyectos observados en la UVG podemos analizar limitaciones que existieron con los proyectos anteriores. Estas entran a la falta de un control en tiempo real y rendimientos que pueden llegar a incrementarse para la clasificación de señales.

Las limitaciones mencionadas abren las puertas a una mejora en el reconocimiento de señales EMG y EEG para el uso de la interfaz biomédica UVG, además de una implementación física. Abriendo el campo de una implementación en tiempo real de los algoritmos para una aplicación específica dentro del campo médico con el uso de actuadores para la recreación de movimientos naturales del cuerpo.

2.3. Control de actuadores por medio de señales bioeléctricas

La investigación de la creación y programación de una prótesis o bien actuadores por medio de señales biomédicas ha sido importante durante las últimas épocas donde se han hecho varios avances con equipo capaz no solo de filtrar y obtener las señales si no que también con la capacidad de analizar y procesar las mismas.

Para poder encontrar y reconocer las diferentes señales obtenidas en el cuerpo y poder interpretarlas para un control más sistemático ha sido tema de discusión en tiempos recientes. El análisis de las mismas y la diferenciación de las mismas cae en las características de las diferentes señales.

Es necesario el uso de aprendizaje automático y el uso de características de las diferentes señales para lograr una clasificación. Ejemplos de estas pueden ser el valor medio absoluto, la longitud de forma de onda, cruces por cero, cambio de signo de pendiente, entre otros [8].

En el año 2015 en la Universidad de Zaragoza [9], se inició entorno experimental para pruebas de control con señales bioeléctricas, donde se logró una captación y lecturas de las señales EMG para crear movimientos de un brazo robótico por medio de la placa *Muscle Sensor v3* con la cual se lograron obtener movimientos simples relacionando señales tanto EMG como EEG y sus características.

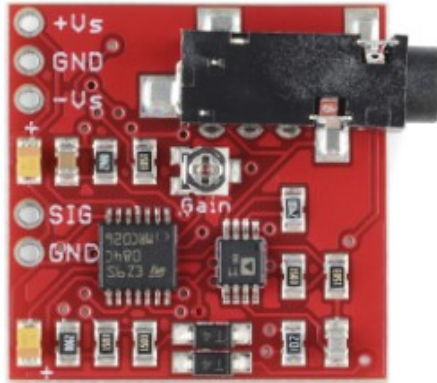


Figura 3: Placa Muscle Sensor v3 [9].

En los proyectos observados, las limitaciones que brindan recaen mucho en el análisis de las señales y el rendimiento de clasificación por medio del Aprendizaje Automático. Los sensores utilizados no brindaron una señal tan limpia y manipulable. La variedad de movimientos se ve limitada y trae consigo cierto porcentaje de error con el reconocimiento de las señales.

Las limitaciones mencionadas brindan campo para mejora de control de los diferentes dispositivos, tanto con una recolección de datos más exacta como una mejora en el procesamiento y análisis de las señales. El control de actuadores por medio de señales bioeléctricas brinda la necesidad de optimizar y mejorar la forma de obtener las señales, los algoritmos de aprendizaje automatizado se muestran esenciales para el control exacto y buscado para aplicaciones específicas.

2.4. Mano animatrónica UVG

En la Universidad del Valle de Guatemala se han desarrollado diversas prótesis con el objetivo de replicar los movimientos naturales de la mano humana. Estas iniciativas buscan fomentar el avance tecnológico, proporcionando beneficios significativos no solo para Guatemala, sino también para todas las personas que necesitan estas prótesis en su vida diaria.

En el año 2020, se realizó un diseño y una prótesis de miembro superior en la Univer-

sidad del Valle de Guatemala hecho por Gonzalo Palarea [10], quien diseñó una prótesis electromecánica de bajo costo por medio de materiales impresos en 3D con la idea de crear una alternativa a los productos y precios de esos tiempos. Palarea logró crear un diseño bajo en costo con una programación basada en el sensor *Myoware* con el cual fue posible realizar movimientos simples reflejando movimientos naturales de la mano.



Figura 4: Módulo Myoware [10].

En el año 2021 se empezó el proyecto de la optimización del diseño y control de una mano animatrónica antropomórfica, realizado por Omar Gálvez. Quien por el uso de motores “Dynamixel” tanto AX-12A como XL-320 logró recrear los movimientos establecidos y suaves aprovechando los motores al máximo. Dentro de la programación utilizada Gálvez utilizó tanto MATLAB como el microcontrolador OpemCM9.04 C para el control de los diferentes actuadores utilizados. Además de utilizar un sensor de captura “LeapMotion” para la obtención de los diferentes movimientos deseados a utilizar en la prótesis [11].

En el año 2022 se continuó con el trabajo del proyecto gracias a Miguel García, quien optimizó aún más el trabajo de Galvez. García trabajó en la implementación del control y visualización de movimientos con el programa “Unity” para el procesamiento de los diferentes movimientos captados por el proyecto [12].

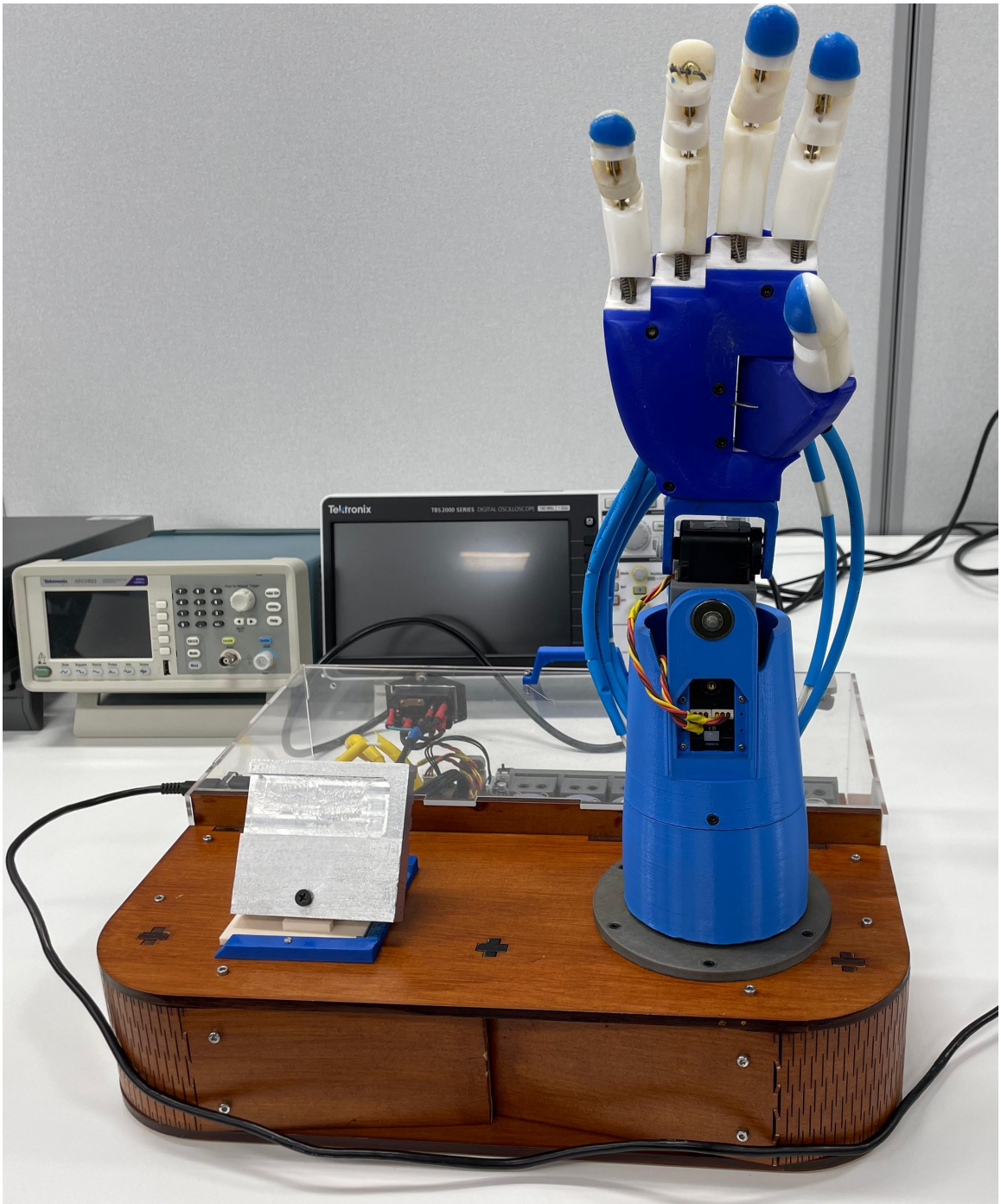


Figura 5: Mano animatrónica antropomórfica [12].

Hoy en día, la tecnología es parte necesaria dentro de nuestras vidas habituales, siendo dentro de diferentes campos de implementación, por ejemplo, dentro del campo médico. La tecnología dentro del ámbito médico muestra ser una clave esencial para el apoyo no solo para el personal médico, sino que brinda ayuda a personas que puedan llegar a tener alguna discapacidad o enfermedad.

El desarrollo y creación de prótesis ha sido un campo de investigación importante no solo en la actualidad, sino que a lo largo de la historia hay registros de usos de prótesis rudimentarias, siendo el primer registro histórico en el antiguo Egipto. La tecnología ha sido importante en la investigación actual de prótesis y un gran campo de desarrollo es el uso de señales bioeléctricas para el control de los mismos.

En la actualidad el desarrollo de prótesis, principalmente de extremidades, se ha enfocado en el uso de señales bioeléctricas para el control de las mismas. El estudio de las señales bioeléctricas ha traído una amplia gama de campos de trabajo no solo para el mencionado control de actuadores, se puede mencionar el análisis de los mismos para lograr un mejor entendimiento de como funciona el cuerpo humano.

El presente trabajo tiene como objetivo utilizar señales EMG y EEG para su uso dentro del control de diferentes actuadores, por medio de una interfaz biomédica y el uso de equipo de extracción de señales. Es importante mencionar que en proyectos desarrollados anteriormente se logró implementar una interfaz capaz de realizar un análisis y diferenciación de diferentes señales bioeléctricas para su uso dentro de simulaciones y el uso en sistemas robóticos.

Este trabajo busca implementar una mejora en el método de recolección desarrollado en trabajos realizados anteriormente y mejorar el análisis de diferenciación entre las diferentes señales creadas por el cuerpo humano. El proyecto también busca implementar una lectura y análisis en tiempo real de las señales para lograr un control en tiempo real.

El control ideal propuesto en la investigación se implementará mediante el uso de apren-

dizaje automático, utilizando clasificadores que se basen en las características de diversas señales. Además de clasificar las señales de manera más precisa, se busca emplear otros tipos de sensores, como sensores de fuerza y giroscopios. Estos sensores no solo permitirán recrear movimientos exactos, sino también detectar y demostrar una señal que indique la intención de realizar un movimiento.

4.1. Objetivo general

Mejorar las interfaces biomédicas para control de sistemas robóticos desarrollada en fases previas, incorporar sensores fuerza e inerciales, implementar un sistema de control en tiempo real y adaptar las interfaces para el control de actuadores.

4.2. Objetivos específicos

- Mejorar las interfaces biomédicas desarrolladas anteriormente, integrando un módulo de detección, muestreo y procesamiento en tiempo real de señales EMG.
- Evaluar sensores de fuerza e inerciales y adaptarlos para su uso dentro de las interfaces.
- Aplicar algoritmos de extracción de características y aprendizaje automático a señales EMG, para el control de actuadores.
- Implementar el control de actuadores por medio de señales EMG y sensores de fuerza y dirección.
- Validar las interfaces desarrolladas reconociendo movimientos comunes y controlando prótesis de miembro superior desarrolladas anteriormente en la UVG.

Este trabajo forma parte de la línea de investigación en interfaces biomédicas, con el principal objetivo de aplicar algoritmos de clasificación basados en aprendizaje automático para el control en tiempo real de actuadores. Además, se implementó el uso de sensores inerciales para mejorar dicho control. De manera más específica, se centró en el desarrollo de una interfaz capaz de integrar estas tecnologías.

El trabajo se estructuró en cinco fases: recolección de señales EMG, desarrollo de algoritmos de clasificación, implementación de sensores inerciales, control de una mano robótica, y desarrollo de la interfaz de control.

El alcance del proyecto se vio afectado por fallas en los sensores MyoWare, por lo que no se logró la implementación deseada. A pesar de esto, se encontraron alternativas para la implementación y el cumplimiento de los objetivos especificados.

A pesar de estas limitaciones, se espera que el desarrollo de este proyecto sienta las bases para futuras investigaciones, ya que la implementación de este tipo de interfaces puede ofrecer un gran apoyo no solo a la tecnología de control por señales bioeléctricas, si no que también un avance en el estudio y desarrollo de prótesis.

6.1. Señales bioeléctricas

Las señales bioeléctricas se refieren a las señales generadas por células nerviosas y células musculares, en específico en la membrana de potencial, las cuales producen una diferencia de potencial y corriente eléctrica para lograr diferentes procesos biológicos. Estas señales son consideradas como las señales biomédicas más importantes dentro del estudio del cuerpo humano, esto gracias a que la mayoría de sistemas dentro del cuerpo utilizan células excitables hace posible que las señales bioeléctricas sean esenciales para conocer cómo funcionan los mismos [13].

Las señales bioeléctricas se pueden clasificar principalmente en el biosistema en el que actúan los mismos, dentro de los más estudiados y utilizados en diferentes campos podemos encontrar: señales electroencefalográficas (EEG), señales electrocardiográficas (ECG) y señales electromiográficas (EMG) [14].

6.2. Señales EMG

Las señales electromiográficas son señales producidas por el sistema muscular esquelético del cuerpo humano o bien el “motor” que mueve al ser humano. Las señales EMG son generadas durante una contracción muscular y la actividad eléctrica que se genera durante este proceso la cual se diferencia en la comparación de voltajes en el proceso. Las señales EMG en lo regular se encuentran con un voltaje entre 70 a 90 mV en reposo. Estas señales son importantes en diferentes campos de investigación e innovación en el mundo científico, siendo parte clave en diferentes estudios sobre el comportamiento del cuerpo humano.

La extracción de estas señales se puede hacer por diferentes formas, principalmente podemos analizar los electrodos de superficie o bien por electrodos intramusculares [15].

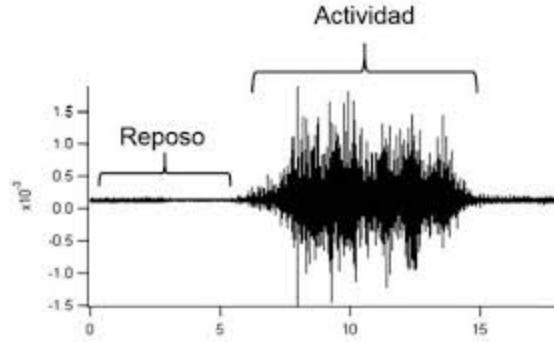


Figura 6: Ejemplo de señal EMG [16].

6.3. Características en el tiempo

6.3.1. Zero crossing

Este parámetro indica cuántas veces una onda cruza el eje cero, un procedimiento común en el procesamiento de señales. Su uso es clave para determinar la frecuencia de la señal de manera precisa [17].

6.3.2. Media del valor absoluto

Estima la media del valor absoluto de un grupo de datos en un tiempo en específico [17].

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_i| \quad (1)$$

6.3.3. Media cuadrática

Su aplicación dentro de las señales EMG, se modela como un proceso aleatorio gaussiano. En el valor calculado se relaciona con la fuerza aplicada [17].

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

6.3.4. Varianza

Es un parámetro que presenta información sobre la dispersión y la variabilidad en un conjunto de datos [17].

$$\text{VAR} = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (3)$$

6.3.5. Amplitud de Willison

Determina las veces que un valor absoluto de una diferencia de amplitud supera un umbral [17].

$$\text{WAMP} = \sum_{n=1}^{N-1} f(|x_{i+1} - x|), f(a) = \begin{cases} 1 & x > \varepsilon \\ 0 & \text{de lo contrario} \end{cases} \quad (4)$$

6.3.6. Curtosis

Es una medida estadística que describe la forma de como se distribuye una variable [17].

$$\text{Curtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)} \quad (5)$$

6.4. BITalino

PLUX Biosignals es una empresa cuyo objetivo es obtener soluciones para el trato y estudio de bioseñales. Cumpliendo con su objetivo, PLUX Biosignals ha desarrollado diferentes herramientas y servicios con el objetivo de apoyar a desarrollo de tecnologías basadas en el uso de bioseñales. Dentro de las herramientas y kits desarrollados podemos encontrar el BITalino siendo una plataforma de bioseñales open source diseñado para captar y analizar señales bioeléctricas [18].

BITalino es compatible con el software OpenSignals y tiene bloques de hardware con sensores capaces de captar señales electrocardiográficas, electromiográficas, electroencefalográficas, actividad electrodermica, entre otros [18].

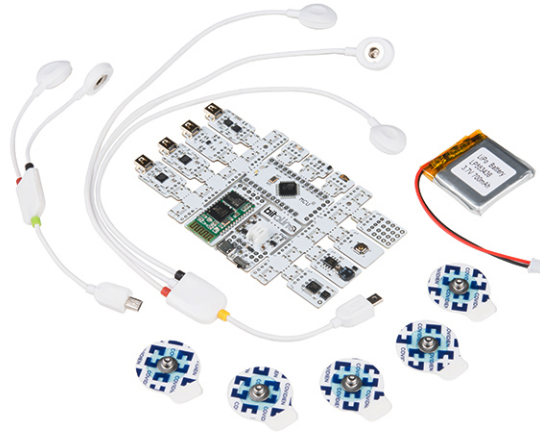


Figura 7: Kit de desarrollo BITalino [18].

6.5. Myoware

MyoWare 2.0 es un sensor creado por la empresa Advancer Technologies, con el objetivo de captar y visualizar señales electromiográficas por medio del uso de tres electrodos. El sensor tiene una variedad de accesorios para la mejora en la captación de señales, el sensor está diseñado para ser trabajado con un microcontrolador Arduino con la capacidad no solo de captar las señales, sino que amplificarlas y rectificarlas para facilitar el análisis y uso de las señales [19].



Figura 8: Muscle Sensor MyoWare 2.0 [19].

6.6. Aprendizaje automático

El aprendizaje automático, también conocido como *Machine Learning*, es una rama de la inteligencia artificial, la cual pretende automatizar la construcción de diferentes modelos analíticos con base en series de ejemplos [20].

Gracias al uso de diferentes datos ejemplares es posible crear una mejora en el “conocimiento” de la máquina para lograr predecir o identificar diferentes series de datos, basándose no solo por analogía o por experiencia.

El aprendizaje automático tiene diferentes ramas en las cuales se han ido creando basándose en diferentes técnicas de aprendizaje, siempre con el objetivo de mejorar con el tiempo y ser adaptable a cambios en los diferentes campos establecidos. Las dos ramas principales del aprendizaje automático se conocen como aprendizaje supervisado y el aprendizaje no supervisado [21].

6.6.1. Aprendizaje no supervisado

El aprendizaje no supervisado se refiere al tipo de aprendizaje sin el uso de etiquetas establecidas en los datos alimentados al sistema, es decir, que el sistema es el encargado de generar y establecer las diferencias entre los diferentes tipos de datos y clasificarlos por su cuenta propia. Principalmente, el aprendizaje no supervisado trata de utilizar técnicas de agrupamiento para lograr la clasificación de la serie de datos [22].

6.6.2. Aprendizaje supervisado

El aprendizaje supervisado es una rama del aprendizaje automático basado en el uso de series de datos con etiquetas establecidas para entrenar a la máquina. Este tipo de aprendizaje puede ser considerado como un aprendizaje guiado por el programador, el objetivo general de este tipo de aprendizaje es generalizar series de datos con base en la experiencia que se le ingresa a la máquina [23].

El aprendizaje automático se puede utilizar en diferentes campos, por ejemplo en la identificación y diagnóstico de una enfermedad por medio de comparación de síntomas, otro ejemplo también es la diferenciación y clasificación de señales eléctricas para su uso práctico o análisis [23].

- **Máquina de vectores de soporte**

Las máquinas de vectores de soporte (SVM) es un método de aprendizaje supervisado utilizado para una clasificación estadística y basándose en un análisis de regresión [21].

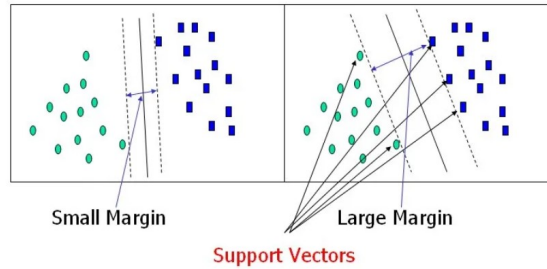


Figura 9: Representación de SVM [24].

Siendo parte del aprendizaje supervisado al momento las SVM al tener 2 o más series de datos etiquetadas, este puede actuar como un clasificador discriminativo, haciendo uso de un hiperplano óptimo para lograr la separación de las clases. Un hiperplano bajo la definición matemática se puede establecer como un subespacio plano en razón a las dimensiones de un plano general, es decir, que para un espacio n-dimensional el hiperplano tiene dimensiones n-1 [25].

Al momento en el que las SVM no pueden dividir claramente los grupos de puntos, existe la opción de utilizar funciones de kernel las cuales ayudan al sistema a ampliar los datos en diferentes grados de dimensiones para encontrar alguna dimensión en la cual, si exista una curva establecida para separar de manera perfecta los grupos de datos, siempre haciendo uso de los hiperplanos para las diferentes y altas dimensiones que se puedan alcanzar.

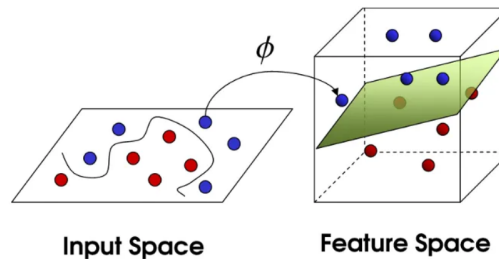


Figura 10: Representación de funciones de Kernel y representación de hiperplanos [24].

Gracias a esto, las SVM son consideradas eficaces en diferentes casos, principalmente en los grupos de datos con una dimensionalidad alta, aunque por preferencia las SVM funcionan con mejor eficacia con grupos de datos no muy extensos y que no posean muchos valores atípicos [25].

Mencionando esto, las SVM son consideradas como clasificadores binarios por lo que su implementación con una clasificación mayor a 2 clases tienden a ser utilizados de diferentes maneras que a la clasificación de 2 clases. Existen diferentes métodos de clasificación dentro de los cuales pueden entrar los métodos *One vs All*(OvA) y *One vs One*(OvO), entre otros.

- One vs All

El método OvA sugiere la creación de un número de máquinas de vectores de soporte equivalente al número de clases existentes. En otras palabras, se genera una SVM para cada clase, comparándola contra todas las demás. Así, el hiperplano resultante de cada modelo SVM describe cómo se diferencia cada clase en relación con el conjunto completo de clases [26].

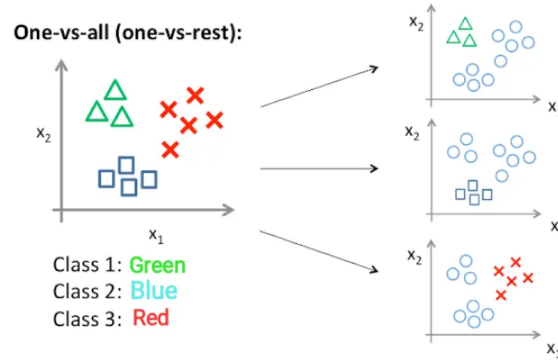


Figura 11: Representación SVM *One vs All* [27].

- One vs One

El método OvO sugiere la creación de un número de modelos SVM correspondiente a todos los pares posibles de clases en el conjunto de datos. En este caso, el número de SVM es igual a la cantidad de comparaciones uno contra uno posibles entre todas las clases del grupo de datos [26].

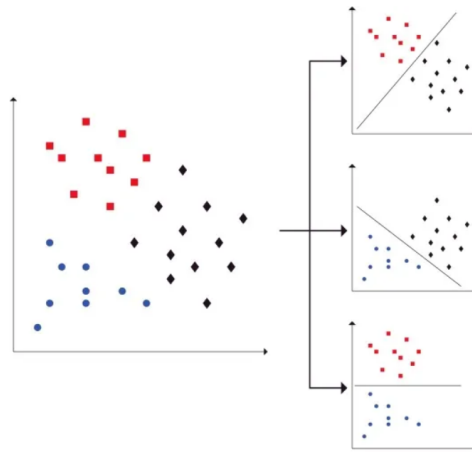


Figura 12: Representación SVM *One vs One* [27].

- **K-Nearest Neighbor**

El algoritmo K-Nearest Neighbor (KNN) es un método de aprendizaje supervisado ampliamente conocido, considerado uno de los algoritmos más antiguos y básicos en el campo de la clasificación. Se destaca por su simplicidad y por ser un algoritmo no paramétrico, lo que significa que no requiere de suposiciones previas sobre la distribución de los datos ni de parámetros predeterminados. En lugar de eso, KNN se

basa únicamente en los datos de entrenamiento disponibles, lo que lo convierte en un modelo flexible y fácil de implementar [28].

El funcionamiento de los algoritmos KNN se fundamenta en el cálculo de distancias entre el punto que se desea clasificar y los puntos ya clasificados durante la fase de entrenamiento. En general, se utiliza la distancia euclidiana para medir la proximidad entre los puntos, aunque existen otros métodos de cálculo que se pueden implementar. Al introducir un nuevo punto o valor, el algoritmo determina a qué clase pertenece identificando los “K” vecinos más cercanos. La clasificación final se realiza en función de la clase predominante entre esos vecinos, es decir, aquella a la que pertenece la mayoría de los puntos más próximos [29].

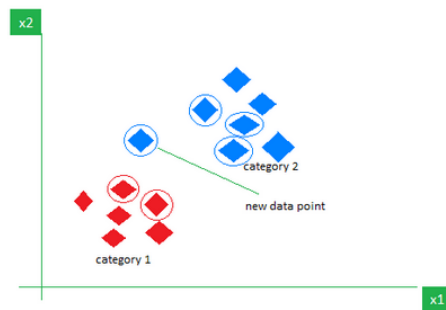


Figura 13: Representación del algoritmo KNN [30].

El uso de algoritmos KNN se limita comúnmente a aplicaciones donde los conjuntos de datos son pequeños y la precisión requerida no es extremadamente alta. Un área popular de aplicación es en los sistemas de recomendación, donde el algoritmo puede sugerir productos o contenido a los usuarios en función de similitudes con otros usuarios o productos. También es empleado en tareas como la detección de anomalías y la clasificación de imágenes [28].

■ Redes neuronales

Las redes neuronales artificiales (RNA) o también conocidas como redes neuronales, son un método de aprendizaje supervisado que se basa en el funcionamiento de las neuronas en el cerebro humano, analizando los datos por medio de diferentes capas de procesamiento [21].

Los RNA se representan por medio de diferentes puntos o nodos, los cuales están interconectados teniendo un valor el cual se cambia mediante más datos ingresen al sistema. Este funciona gracias a los diferentes procesos que ocurren en las diferentes capas, es decir que al momento que ingresa un dato a la primera capa este lo procesa y lo manda a la segunda capa con el propósito de ir descomponiendo los datos originales en valores más abstractos dejando únicamente la información “importante” para el aprendizaje del sistema [23].

En mención, los RNA más simples solo constan de 3 capas, siendo una capa de entrada, una de procesamiento y otra de salida, aprendiendo principalmente sobre el ajuste de

los parámetros que van cambiando por su cuenta gracias al algoritmo común para este método de aprendizaje. Este proceso se conoce como entrenamiento, por lo que al momento que el algoritmo termina de trabajar es posible ingresar valores reales para comprobar el funcionamiento del algoritmo [23].

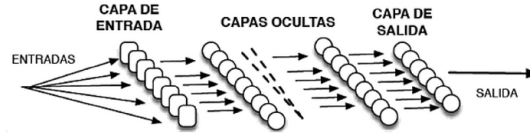


Figura 14: Representación de capas de una Red neuronal artificial simple [23].

Las redes neuronales son los algoritmos más utilizados por las ventajas que traen siendo capaces de reconocer patrones, prototipos y relaciones entre los conjuntos de datos alimentados al sistema [21].

6.7. Interfaz HMI (Human-Machine Interface)

Las interfaces humano-máquina es conocida como la interacción entre las personas con una interfaz diseñada para una interacción con un sistema o herramienta. Las interfaces HMI han sido importantes con el desarrollo de la tecnología para permitir la interacción con el usuario y el producto. Las implementaciones de las interfaces pueden tener diferentes formas, las cuales dependen de las necesidades del sistema y como se desea el control de la misma.

El rol de las interfaces HMI para facilitar las interacciones pueden caer dentro de: una interacción intuitiva del usuario, la capacidad de traer información al usuario, integrar diferentes sistemas, entre otros [31].

6.8. Sensores inerciales

Los sensores inerciales son dispositivos que tienen la capacidad de medir aceleraciones tanto lineales como angulares. Estos dispositivos normalmente están compuestos por medio de acelerómetros, giróscopos y magnetómetros. Además muestran ser útiles en diferentes campos como lo son: la robótica, animación, estudios del movimiento humano, entre otros [32].

6.9. Actuadores

Un actuador es un dispositivo capaz de convertir energía eléctrica en movimiento o torque. Existen diferentes tipos de actuadores, clasificados según sus especificaciones y usos. Estos dispositivos son fundamentales para el desarrollo tecnológico y tienen múltiples aplica-

ciones, tales como el control automatizado, la robótica y el posicionamiento de componentes, entre otros [32].

6.10. Prótesis

De acuerdo a la enciclopedia A.D.A.M. una prótesis es un dispositivo diseñado para remplazar una parte faltante del cuerpo o ayudar a que una parte del cuerpo funcione mejor [33].

Las prótesis han sido parte importante dentro de la historia de la humanidad, donde se han encontrado evidencias que se han realizado prótesis a lo largo de la historia. Siendo de las más antiguas una momia con una prótesis del dedo gordo del pie hecha de cuero para ayudar al movimiento de la persona [34].



Figura 15: Prótesis realizada en el antiguo Egipto [34].

Hoy en día se muestran diferentes investigaciones y desarrollos sobre como mejorar las prótesis para la gente con la necesidad de una, con el desarrollo de prótesis biónicas con el objetivo de lograr que el miembro artificial tenga la capacidad de ser controlado por las señales naturales del cuerpo sin la necesidad de tener movimientos fijos y preestablecidos. La implementación de motores y el desarrollo de tecnologías de inteligencia artificial han permitido una evolución en la creación de prótesis [34].

Con el fin de adquirir un mejor entendimiento de las señales EMG, se decidió emplear distintos tipos de sensores para su captura y procesamiento. Además, se realizaron experimentos con estas señales para lograr su recopilación y procesamiento en tiempo real, con el fin de utilizarlas en el control de un actuador.

7.1. Dispositivos de adquisición para señales EMG

Para la adquisición de las señales EMG, se analizaron diferentes opciones de sensores, destacando el uso del dispositivo BITalino, previamente empleado en fases anteriores en la Universidad del Valle de Guatemala. En particular, se utilizó la placa *BITalino (r)evolution Board*, que cuenta con múltiples canales para la captura de diversas señales bioeléctricas. Con el objetivo de explorar alternativas al BITalino, se experimentó con el sensor MyoWare. Este dispositivo está diseñado específicamente para obtener señales EMG, presentándolas en diferentes modalidades. Además, su compatibilidad con microcontroladores como Arduino lo convierte en una herramienta versátil para el desarrollo de diversos proyectos. Para este estudio, se empleó el modelo *MyoWare 2.0* junto con sus accesorios para comunicación inalámbrica.

7.1.1. Dispositivo BITalino

Para la utilización del sensor BITalino se hizo uso del software de MATLAB ya que cuenta con dos *toolboxes* las cuales poseen diferentes funciones para facilitar el uso del sensor y obtener los mejores resultados con el sensor. Se realizaron pruebas con ambas librerías siendo estas *MATLAB Support for BITalino Biosignal Devices* y *BITalino Toolbox*,

ambas cuentan con diferentes ventajas y desventajas para su uso dentro del proyecto.

Ambas librerías utilizan la función *Read()*, la cual permite la lectura de los n valores anteriores guardados en el *buffer* permitiendo una lectura rápida y exacta de los valores leídos en el sensor. Gracias a esto esta función fue la implementada dentro de la experimentación del proyecto.

En fases anteriores, se determinó que las frecuencias de muestreo en las que funciona el sensor son de 1, 10, 100 y 1000 Hz.

7.1.2. Dispositivo MyoWare

Para utilizar el dispositivo MyoWare, fue necesario integrar un microcontrolador ESP32 para establecer la comunicación entre el sensor y MATLAB mediante comunicación serial. Entre los accesorios disponibles, el accesorio de comunicación inalámbrica fue el más utilizado. Este microcontrolador funciona a través del módulo ESP32-WROOM, el cual utiliza comunicación BLE (*Bluetooth Low Energy*) para enlazar el accesorio con el dispositivo ESP32. El sensor MyoWare ofrece diferentes modos de representación de la señal capturada, gracias a su diseño que incorpora diversas configuraciones y filtros, como se puede observar en la Figura 16.

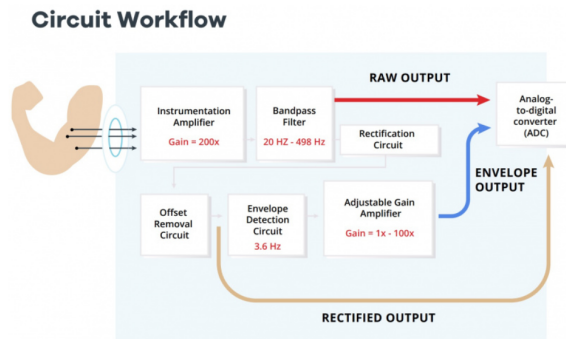


Figura 16: Esquema de circuito utilizado en el sensor MyoWare [19].

Se encontraron diferentes problemas del sensor por lo que se optó por la funcionalidad “Envelope” la cual representa la señal en niveles de intensidad, similar a un pulso. Para obtener este tipo de señal, es necesario cambiar las configuraciones del sensor a las de tipo *ENVELOPE*.

7.2. Algoritmos de adquisición

Como se mencionó dentro del apartado 7.1.1, se utilizaron dos *toolboxes* específicos para las pruebas dentro del proyecto. A continuación se presentan los resultados obtenidos con ambas librerías.

7.2.1. *MATLAB Support for BITalino Biosignal Devices Toolbox*

Como se mencionó anteriormente, ambos *toolboxes* utilizan la misma función *read()* para la adquisición de señales. La principal diferencia radica en que el *MATLAB Support for BITalino Biosignal Devices Toolbox* es la versión más actualizada, lo que le otorga más funcionalidades en comparación con el *BITalino Toolbox*. Esta versión proporciona un control más preciso del sensor, permitiendo ajustar diversos parámetros con mayor facilidad.

Al utilizar la función *read()*, solo es necesario ingresar el nombre del dispositivo y elegir entre el tiempo de captura o el número de muestras. Al ejecutar este comando, el sensor se prepara automáticamente para la captura de datos y realiza las operaciones necesarias para su control.

El principal problema identificado con esta librería es su falta de adecuación para un funcionamiento en tiempo real, lo que genera dificultades al capturar señales con un tiempo establecido o un número fijo de muestras.

Al configurar un tiempo establecido, el sensor espera un segundo completo antes de realizar la captura de datos. Esto genera un desfase respecto a los movimientos realizados en tiempo real. Aunque se intentó reducir este tiempo, el intento no tuvo éxito, ya que la función solo admite valores enteros, impidiendo realizar capturas en intervalos menores a un segundo.

Por otro lado, al emplear la captura basada en un número fijo de muestras, es necesario incluir un tiempo de espera entre lecturas. Si esta pausa se elimina, el sensor comienza a capturar un alto nivel de ruido, lo que degrada la calidad de las señales obtenidas y las vuelve inadecuadas para cualquier tipo de análisis o clasificación.

Debido a estos problemas, se concluyó que el uso de esta librería no es adecuado para aplicaciones en tiempo real. Sin embargo, en contextos donde no se requiera funcionalidad en tiempo real, la librería demostró ser superior gracias a su facilidad de uso y la consistencia en la captura de señales.

7.2.2. *BITalino toolbox*

Como se mencionó anteriormente, se utilizó la función *read()* del *toolbox* de BITalino para la lectura de señales, las cuales se almacenan inicialmente en el *buffer* del sensor. Una vez capturados los valores y gracias a la comunicación con MATLAB, estos se almacenan en una variable. Consecuentemente el sensor vacía el *buffer* y se prepara para recibir nuevos datos.

A diferencia del *MATLAB Support for BITalino Biosignal Devices Toolbox*, la librería *BITalino Toolbox* no incluye la funcionalidad automática para preparar el sensor para la toma de datos. Por esta razón, es necesario realizar una configuración manual que mantenga al sensor en espera de nuevas lecturas.

Esto se logra mediante el uso de la función *startBackground(Objeto)*, que inicializa el sensor y le permite leer datos en segundo plano, sin interrumpir la ejecución de otros comandos

o funciones en MATLAB.

Primera iteración

Para implementar una funcionalidad de lectura en tiempo real, es esencial que el sensor esté constantemente capturando valores. Por esta razón, se implementó un ciclo *while* en el cual el sensor captura muestras durante un lapso de 0.25 segundos llegando a un aproximado de 250 a 300 muestras por captura, esperando la activación de una bandera para almacenar los valores obtenidos a través de la función *read()*.

Durante las pruebas de captura de valores, se evaluaron diferentes intervalos de tiempo, incluyendo 5 segundos, 1 segundo, 0.50 segundos y 0.25 segundos. El lapso de 0.25 segundos resultó ser el más adecuado, permitiendo la captura y clasificación de las señales de manera eficiente, sin causar problemas para la recolección de las mismas.

En las pruebas realizadas con este lapso de tiempo, se determinó que el algoritmo no es adecuado para una implementación en tiempo real. Esto se debe al intervalo entre lecturas del BITalino, lo que ocasiona la pérdida de cierta cantidad de información debido al tiempo de espera entre lecturas. Para optimizar el uso del sistema, se determinó que la opción más simple y efectiva para eliminar esto es la reducción del tiempo de captura.

Al reducir el intervalo entre capturas el algoritmo presentó el inconveniente de dificultar la recolección de señales por parte del usuario, ya que el lapso de tiempo entre capturas es tan breve que complica la reacción del usuario para realizar los gestos deseados entre lecturas.

Es importante destacar que el *toolbox* de BITalino despliega las señales capturadas en múltiples canales. Esto gracias a la existencia de diferentes tipos de señales las cuales puede capturar el dispositivo. Sin embargo, en esta implementación, únicamente se seleccionó el canal correspondiente a las señales EMG.

Segunda iteración

En la segunda iteración del algoritmo se implementaron las mismas funciones utilizadas en la versión anterior, pero esta vez se enfocó en lograr lecturas constantes, sin interrupciones entre ellas.

Para implementar esta funcionalidad, fue importante considerar una desventaja al realizar lecturas rápidas con la librería de *BITalino Toolbox*. Las pruebas mostraron que en intervalos de tiempo muy cortos, las muestras capturadas eran inconsistentes; el sensor no lograba realizar lecturas precisas sin un breve tiempo de pausa, necesario para llenar el *buffer* del sensor. Para resolver este problema, se determinó que era esencial introducir una pequeña pausa entre lecturas. Se eligió un intervalo mínimo de 100 milisegundos entre lecturas para minimizar la pérdida de datos.

Al implementar esta pausa, surgió una nueva complicación: en tiempos cortos, el sensor mostró una variabilidad en la cantidad de muestras recopiladas. En pruebas realizadas con

intervalos menores a un segundo, se observó que, a 100 milisegundos, el sensor recogía entre 100 y 200 muestras, lo cual generaba cierta inconsistencia.

Para gestionar estas limitaciones y optimizar la recolección de datos, se diseñó un *buffer* circular que permite mantener un espacio de almacenamiento constante para las señales, actualizándose tantas veces como sea necesario. Este *buffer* tiene capacidad para almacenar aproximadamente 300 señales y se actualiza mediante un ciclo *while* para asegurar una captura continua y eficiente.

7.3. Algoritmo de captura en tiempo real

Con el objetivo de obtener lecturas en tiempo real el BITalino debe de estar leyendo constantemente, por lo que es necesario un algoritmo que tenga la capacidad de detectar cuando se desea una lectura.

Al igual que en el algoritmo de recolección de señales, se implementaron dos métodos de captura en tiempo real. El primero utiliza la primera iteración del proceso de recopilación de señales junto con un sensor MyoWare, mientras que el segundo emplea una detección automática basada en umbrales en conjunto de la segunda iteración del algoritmo de recopilación.

7.3.1. Implementación BITalino con MyoWare

Dentro de la primera iteración para implementar el funcionamiento de lectura en tiempo real, se utilizó el sensor MyoWare y su funcionalidad *Envelope*. Primero se inicializó la comunicación entre el sensor y sus diferentes accesorios de comunicación con el ESP32 y la misma entre el ESP32 y MATLAB. Utilizando la librería de MyoWare en el ambiente de programación Arduino IDE se debe inicializar el sensor en una propiedad de *ENVELOPE* y realizar el procedimiento de comunicación con el ESP32 dentro de un ciclo *while* para que el MyoWare siempre esté enviando valores. Al recibir la información en el ESP32 se realizó un límite para convertir la señal en una bandera, siendo esta la que recibe MATLAB. Una vez MATLAB recibe esta bandera, ejecuta los comandos necesarios para obtener la señal del dispositivo BITalino.

En la Figura 17, se puede observar el diagrama de flujo del algoritmo implementado.

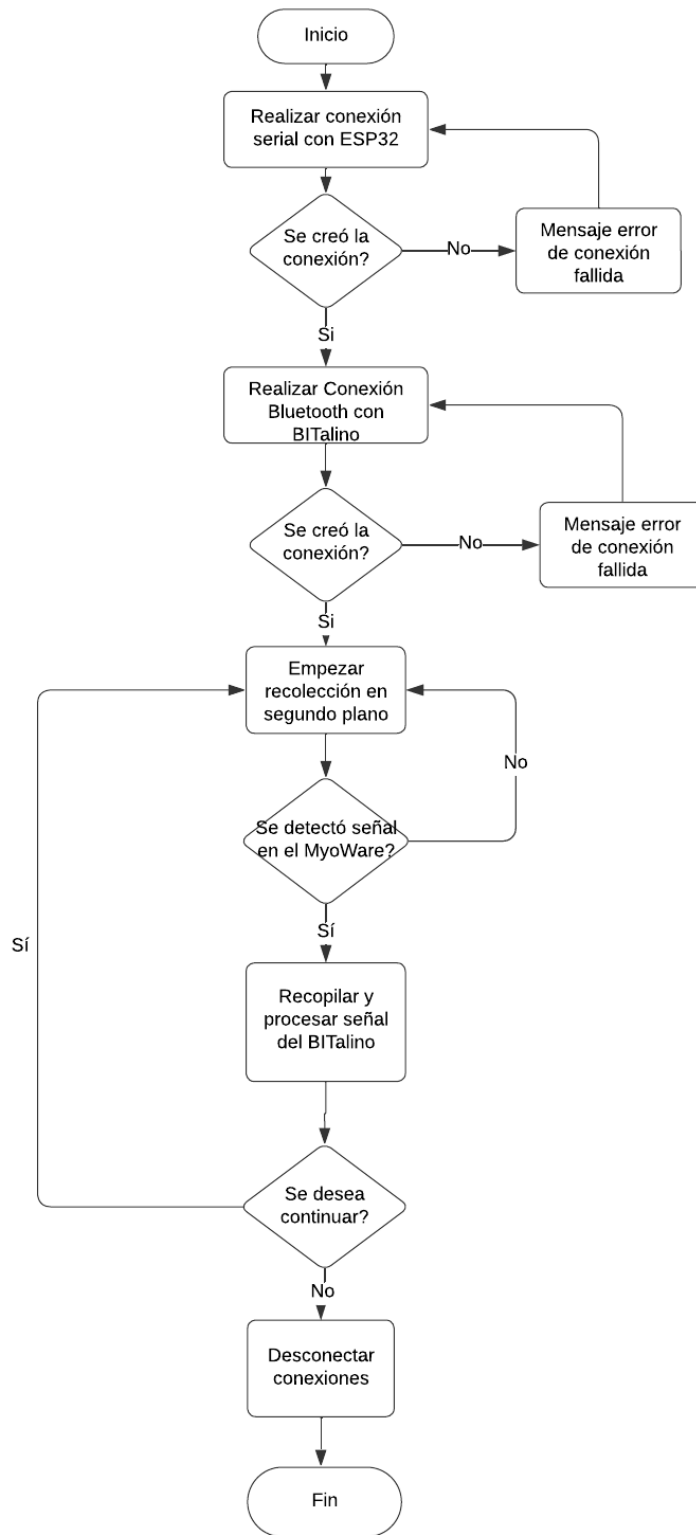


Figura 17: Diagrama de flujo de algoritmo de adquisición en tiempo real con MyoWare.

7.3.2. Implementación BITalino con detección automática

En esta implementación de captura de señales en tiempo real, se utilizó un umbral base para identificar momentos en los que no hay actividad de señal.

Para definir este umbral, fue necesario implementar un algoritmo de calibración. Este algoritmo captura señales durante 5 segundos mientras el usuario se encuentra en reposo, calculando la media del valor absoluto (MAV) de las señales obtenidas. Al finalizar este proceso, se selecciona el valor máximo de las características medidas para establecer el umbral que las señales deben superar.

Con el umbral definido, el algoritmo de recolección comienza a almacenar las señales capturadas, comparando el MAV recolectado en tiempo real con el umbral establecido. Cuando se detecta una señal que supera el umbral, el algoritmo almacena esa señal junto con 300 muestras previas y 300 muestras posteriores al momento en que se registró la detección.

En la Figura 18, se puede observar el diagrama de flujo del algoritmo implementado.

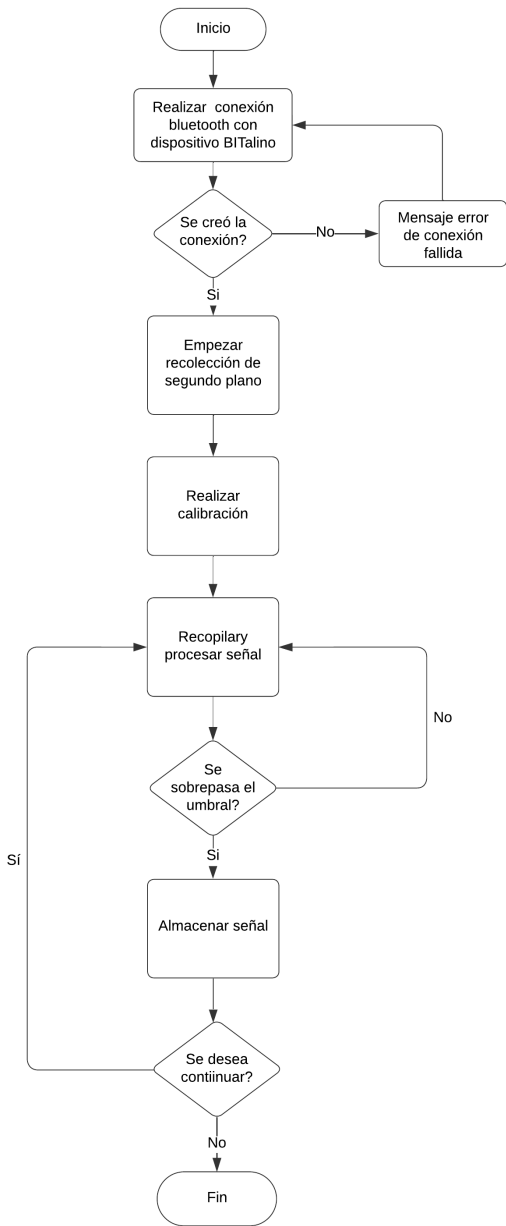


Figura 18: Diagrama de flujo de algoritmo de adquisición en tiempo real con algoritmo de detección automática.

7.4. Procesamiento de señales

Al tener la señal en bruto deseada, se emplearon dos filtros con el objetivo de obtener la mejor señal posible. Esto se hizo con el objetivo de evitar problemas al momento de su clasificación. En investigaciones anteriores se determinó que las señales EMG se encuentran entre el rango de frecuencia de 10 Hz y 450 Hz [4]. En fases anteriores del proyecto se diseñó un filtro IIR de tipo Butterworth pasa bandas entre los rangos de 20 Hz y 450 Hz. Además, se implementó un filtro rechaza bandas entre 59 Hz y 61 Hz para eliminar el ruido causado por la red eléctrica [6].

Para almacenar las señales capturadas de manera más consistente, se implementó un algoritmo diseñado específicamente para este propósito. Este algoritmo identifica la ventana en la que ocurrió el movimiento utilizando una calibración basada en el ruido de fondo. Posteriormente, dentro de una ventana de 500 muestras, centra la señal para garantizar una representación más precisa y uniforme de los datos. Este algoritmo fue proporcionado por el Dr. Luis Alberto Rivera Estrada.

Para mejorar el reconocimiento y clasificación de las señales, se decidió segmentarlas en tres partes. Esta segmentación permitió capturar y analizar distintas fases de cada señal, lo que enriqueció el conjunto de datos y facilitó una mejor identificación de patrones específicos en cada segmento.

7.5. Interfaz de recolección de señales

Antes de iniciar el entrenamiento y clasificación de señales, se implementó un módulo desarrollado en trabajos anteriores del proyecto. Este módulo fue creado utilizando la aplicación integrada de *App Designer* en MATLAB. El módulo incluye diversas opciones con el objetivo de determinar las características de extracción de las señales, además de permitir la conexión con un dispositivo BITalino. La aplicación también muestra las señales adquiridas en gráficos, representando los valores en voltaje. Gracias a esto la interfaz es capaz de mostrar las señales adquiridas en tiempo real durante el tiempo establecido [6]. Este módulo desarrollado anteriormente se puede observar en la Figura 19.

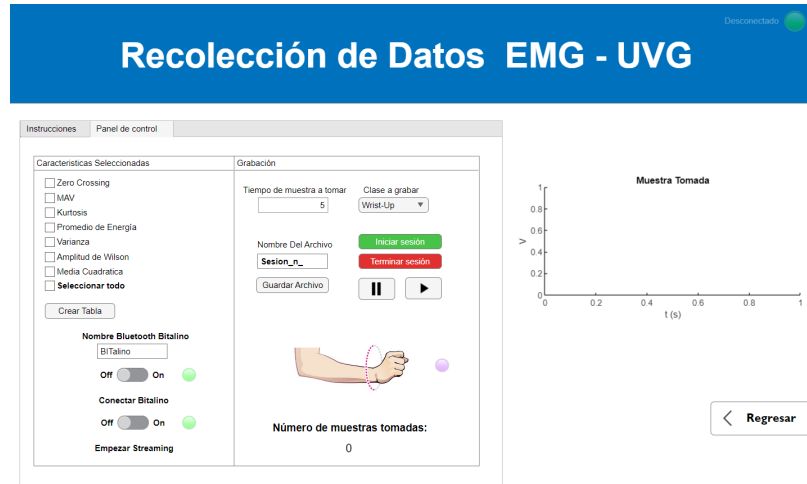


Figura 19: Módulo de recolección de señales EMG de proyectos anteriores [6].

Aunque se contaba con el módulo desarrollado en proyectos anteriores, fue necesario realizar ajustes para adaptarlo a la implementación del proyecto presente. En primer lugar, se eliminó la restricción de clases, ya que el objetivo de este trabajo era recopilar distintos tipos de señales, no solo un grupo en específico.

Gracias a la implementación de distintos algoritmos, se desarrollaron varias iteraciones del módulo para integrar y optimizar los algoritmos creados. Cada iteración permitió ajustar y mejorar la interacción entre el usuario y el sistema de recopilación.

7.5.1. Primera iteración del módulo

Para los propósitos de este trabajo y la implementación del primer algoritmo desarrollado, se incorporó un segundo sensor BITalino, lo cual exigió implementar un sistema adicional de conexión y extender las funcionalidades del módulo. Este cambio permitió añadir un nuevo modo de uso en el módulo, ofreciendo al usuario la opción de operar con un solo sensor o con dos sensores simultáneamente.

La integración del segundo sensor no solo requirió adaptar la conexión, sino también rediseñar aspectos del módulo para gestionar y visualizar los datos capturados de ambos sensores en tiempo real. Este nuevo modo de uso permite una mayor flexibilidad en las aplicaciones del sistema y mejora la precisión en la captura de señales al disponer de un sensor adicional. Los cambios en el módulo se pueden observar en la Figura 20.

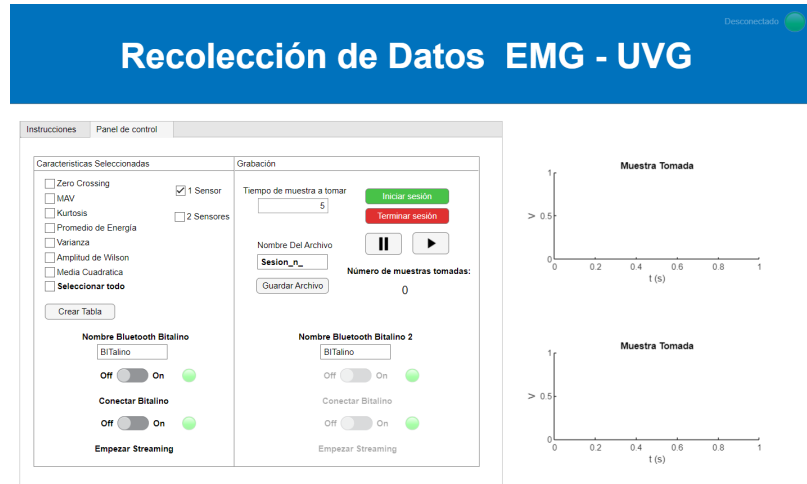


Figura 20: Segunda versión del módulo de recolección de señales EMG.

En la Figura 21 se puede observar el diagrama de flujo del algoritmo implementado, para el módulo de recolección EMG.

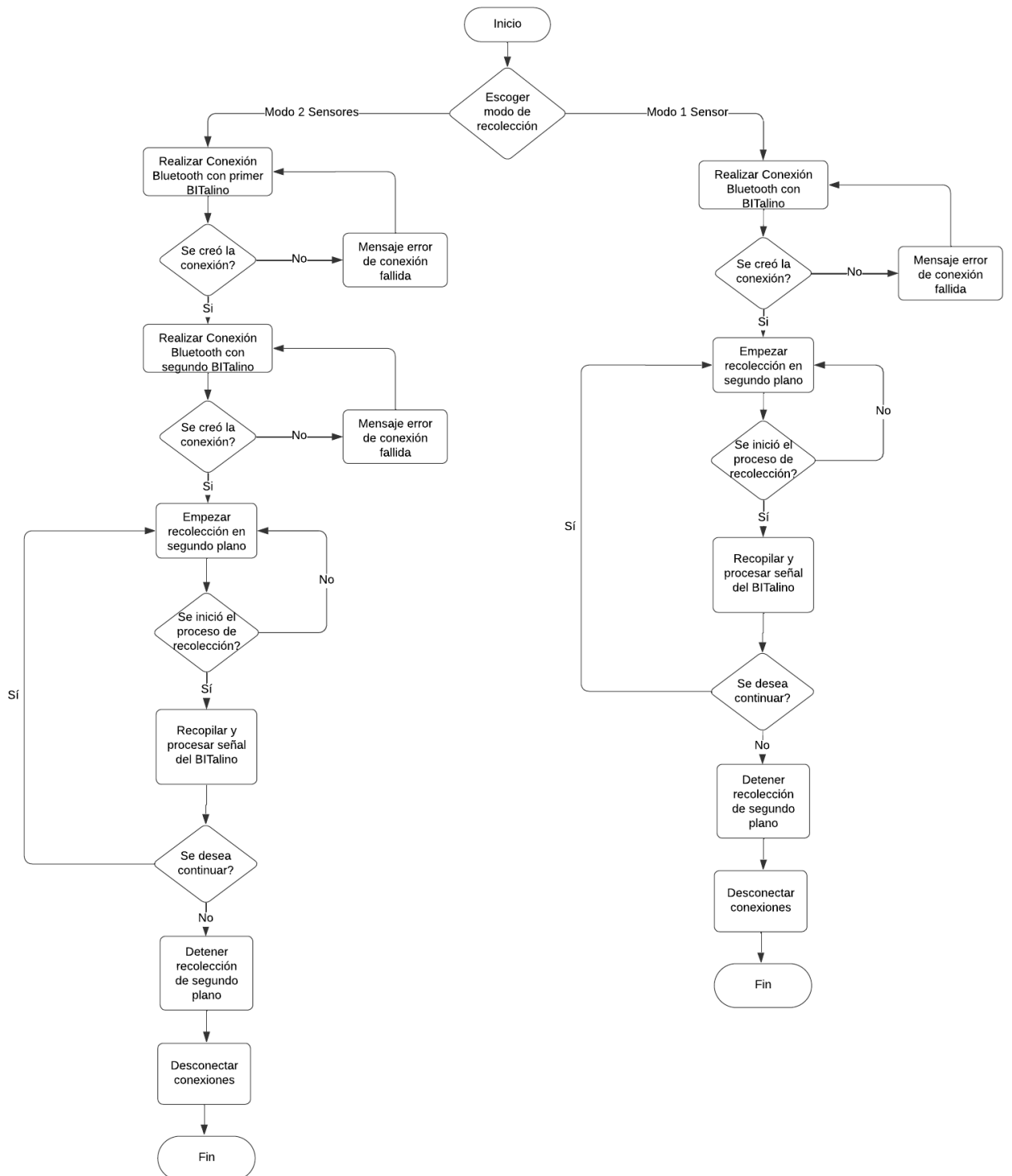


Figura 21: Diagrama de flujo de algoritmo de módulo de recolección EMG.

7.5.2. Segunda iteración de interfaz

Para la implementación de la segunda iteración, que utiliza el algoritmo de detección automática, fue necesario agregar un botón de calibración en el módulo. Este botón permite al usuario iniciar el proceso de calibración de manera sencilla, facilitando la captura de un umbral base en tiempo real.

En la Figura 22, se puede observar el diagrama de flujo del algoritmo implementado, para esta versión del módulo.

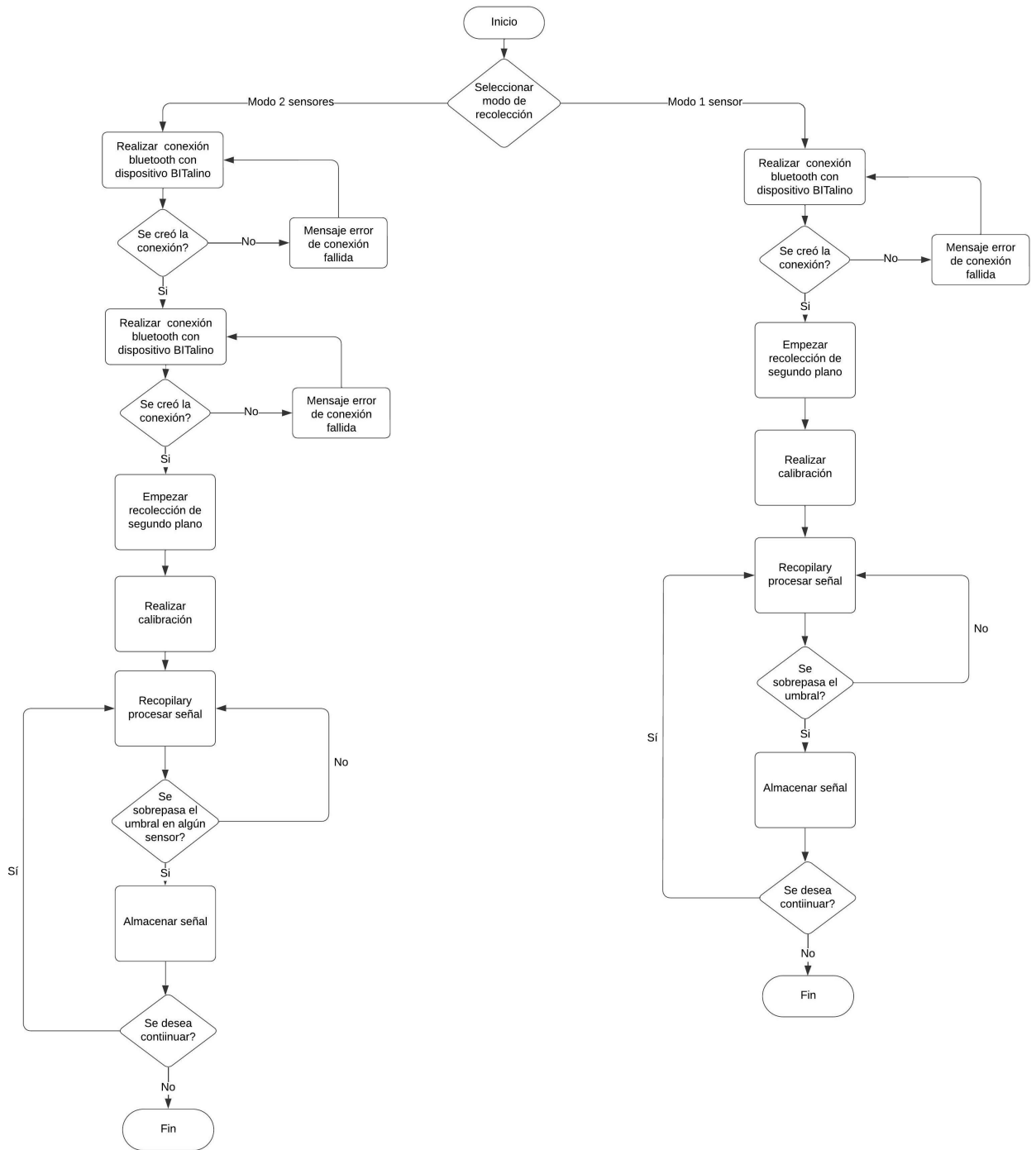


Figura 22: Diagrama de flujo de algoritmo de segunda iteración de módulo de recolección EMG.

Durante las pruebas realizadas con los distintos algoritmos y módulos, se observó que los algoritmos y módulos que incluían el modelo de detección automática demostraron ser los más eficientes y consistentes. Este modelo permitió optimizar la captura de señales en tiempo real, reduciendo tanto los errores de detección como la necesidad de ajustes manuales.

En la Figura 23, se puede observar la versión final del módulo de recolección de señales EMG.

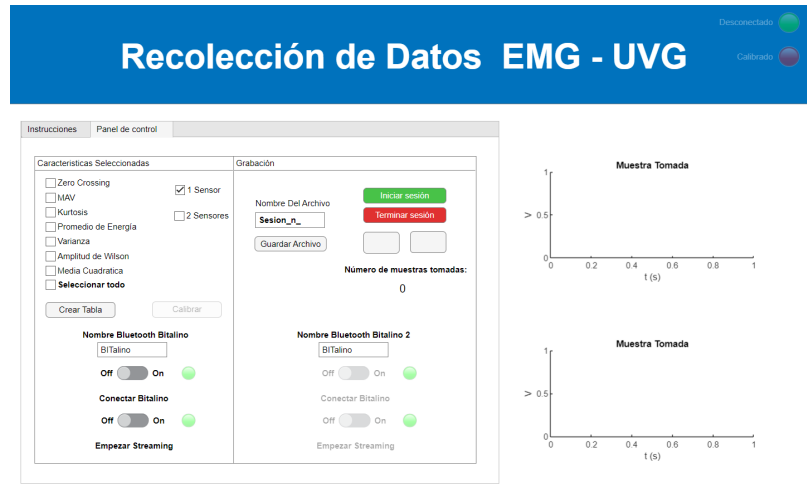


Figura 23: Segunda versión de interfaz de recolección de señales EMG.

Clasificadores de señales EMG

En este capítulo se presentan los clasificadores utilizados junto con sus respectivos resultados. Para ello, los resultados se ilustran mediante matrices de confusión, las cuales permiten evaluar la precisión de los algoritmos. Estos resultados demuestran el desempeño de los distintos métodos frente a diversas pruebas. Para validar su funcionalidad, se realizaron ensayos utilizando una variedad de gestos y diferentes cantidades de sensores.

8.1. Características utilizadas

Con el objetivo de obtener resultados más eficientes y reducir la carga computacional, se realizaron dos intentos principales. En el primer intento, se utilizaron todas las características implementadas dentro de la interfaz de recolección. Estas características son: *Zero Crossing*, Media del Valor Absoluto (MAV), Curtosis, Promedio de Energía, Varianza, Amplitud de Willison y Media Cuadrática.

En el segundo intento, se redujo deliberadamente el número de características, eliminando aquellas que no aportaban significativamente a la precisión del algoritmo. Este enfoque permitió optimizar el rendimiento del sistema, logrando un equilibrio entre eficiencia computacional y precisión en los resultados. Las pruebas realizadas revelaron que, al utilizar únicamente las características *Zero Crossing* y MAV, se obtenían resultados y porcentajes de exactitud comparables a los obtenidos al emplear todas las características. Por ello, se optó por utilizar estas dos características, manteniendo así un estándar consistente en las pruebas realizadas.

8.2. Gestos utilizados

Como se mencionó anteriormente, se llevaron a cabo pruebas utilizando distintas cantidades de gestos, con el objetivo de evaluar el rendimiento de los clasificadores en diferentes situaciones. Estas pruebas brindan una perspectiva más amplia de la capacidad para manejar datos con diferentes números de clases.

En particular, se realizaron experimentos utilizando colecciones de señales recopiladas con un solo sensor y con dos sensores, lo que permitió evaluar la eficacia de los algoritmos al adaptarse a estas configuraciones. Este enfoque permitió analizar cómo varía el rendimiento de los clasificadores al enfrentarse a diferentes volúmenes y complejidades de datos, destacando su capacidad para ajustarse a entornos más simples o más complejos.

Estos experimentos proporcionaron información valiosa sobre la robustez y versatilidad de los clasificadores, ayudando a identificar las configuraciones óptimas y su comportamiento frente a diversas condiciones de captura de señales.

8.2.1. Colecciones recopiladas con un sensor

Para obtener los mejores resultados en la recolección de señales, la posición ideal de los electrodos consiste en colocar dos de ellos sobre el músculo principal, mientras que un tercer electrodo se sitúa a cierta distancia para actuar como referencia. Esta disposición se muestra en la Figura 24.

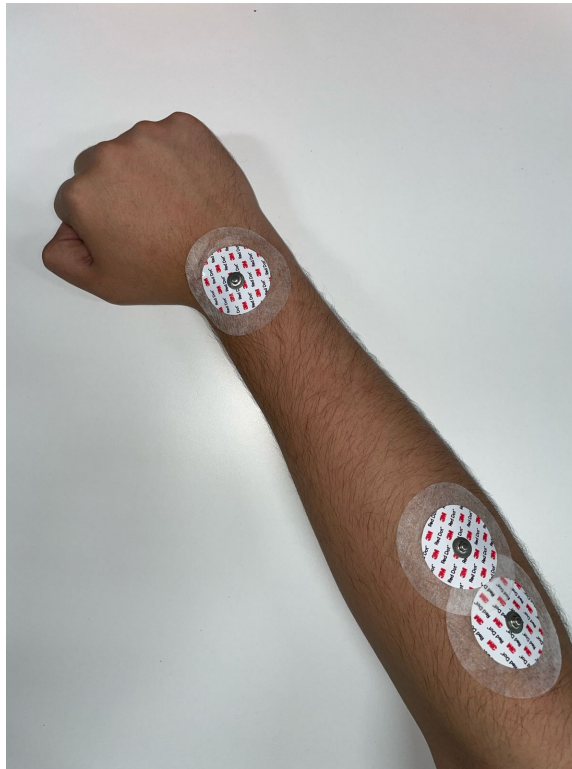


Figura 24: Posición de los electrodos para la captura de los 3 gestos.

Colección de 2 gestos

El primer gesto fue el movimiento del puño para arriba. Este se puede observar en la Figura 25.



Figura 25: Gesto 1: Movimiento de puño hacia arriba.

El segundo gesto fue el realizar un movimiento de una “ola” con los dedos de la mano (Figura 26).

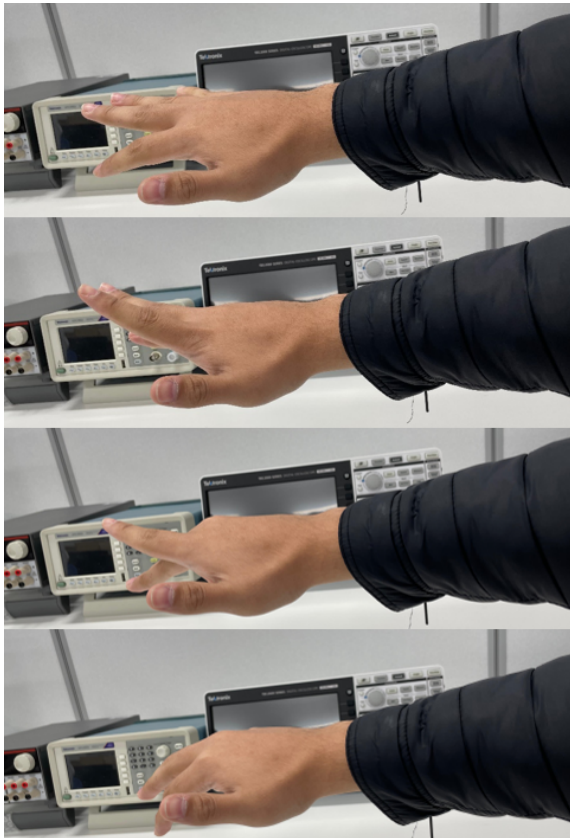


Figura 26: Gesto 2: Movimiento de “ola” con los dedos de la mano.

Colección de 3 gestos

Esta colección se compone de la combinación de los 2 gestos, observados en la sección 8.2.1. Además de la adición de un tercer gesto.

El tercer gesto es el movimiento del puño para arriba realizado dos veces (Figura 27).

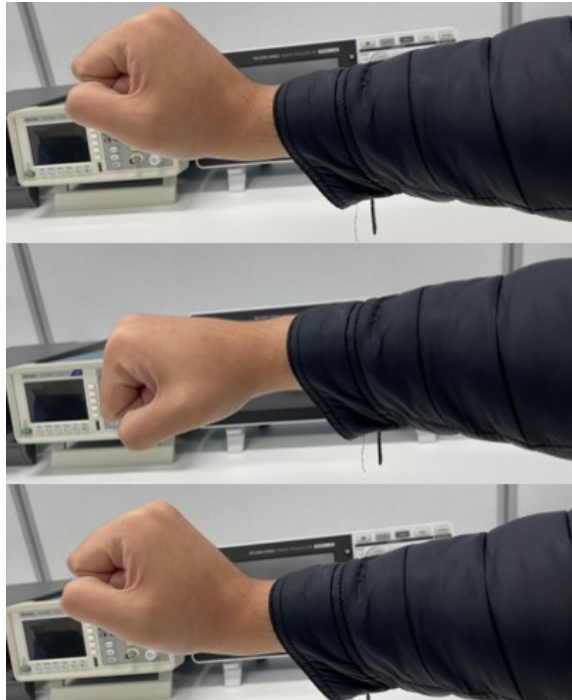


Figura 27: Gesto 3: Movimiento muñeca para arriba dos veces.

8.2.2. Colección recopilada con dos sensores

Para realizar la recopilación de señales con dos sensores, se continuó con la disposición de electrodos mencionada en la Sección 8.2.1. Además, de agregar un segundo conjunto de electrodos en otro músculo. Este posicionamiento de electrodos se puede observar en la Figura

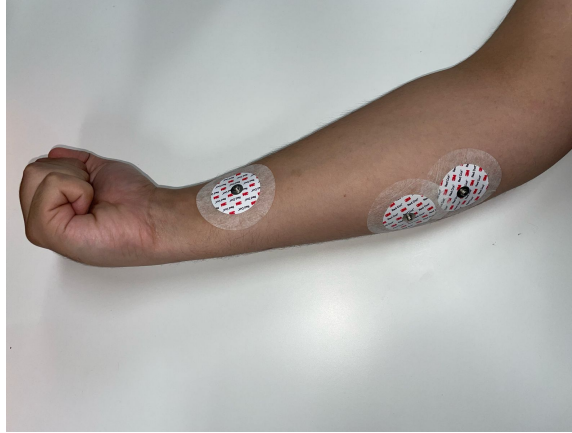


Figura 28: Posición de electrodos de segundo sensor.

Colección 3 gestos

Esta colección se compone de la combinación de los 2 gestos utilizados con un solo sensor, observados en la sección 8.2.1.

El tercer gesto es el movimiento de hacer un puño hacia abajo. Este gesto se puede observar en la Figura 29.



Figura 29: Gesto 3: Puño hacia abajo.

8.3. *Classification Learner*

Para comparar y seleccionar los clasificadores más adecuados, se utilizó la aplicación integrada *Classification Learner* de MATLAB, que permite entrenar una amplia variedad de modelos de clasificación. La aplicación ofrece métodos como: Máquinas de Vectores de Soporte (SVM), Redes Neuronales Artificiales (ANN), y *k-Nearest Neighbor* (KNN), entre otros.

Con el objetivo de evaluar distintas opciones, se seleccionaron tres métodos principales: SVM, ANN y KNN. El entrenamiento se centró en estos modelos, priorizando aquellos que mostraron los mayores porcentajes de precisión durante la validación. Para garantizar un estándar uniforme entre los modelos utilizados, se eligieron aquellos que, en promedio, obtuvieron los mejores resultados. Estos modelos son: SVM Lineal, SVM Cuadrática, SVM Cúbica, ANN Ancha, ANN de dos capas, ANN de tres capas, KNN Fina, KNN Pesada y KNN Cúbica.

De la misma forma se evaluaron los porcentajes de validación en fase de prueba, con el objetivo de comprobar su utilidad con datos diferentes a los que se evaluaron. La aplicación solo requiere una base de datos estructurada según las características deseadas, junto con las etiquetas correspondientes a cada dato. Para este proyecto se hicieron uso de dos características siendo *Zero Crossing* y MAV.

8.4. Resultados de clasificación de 2 gestos con un sensor

Para la evaluación, se utilizó una recolección de 80 muestras, distribuidas en 40 por clase. Adicionalmente, se creó una base de datos con 20 muestras, divididas en 10 por clase, con el objetivo de verificar el rendimiento de los clasificadores.

8.4.1. SVM

El porcentaje de validación obtenido se puede observar en el Cuadro 1.

SVM	Número de clases	Porcentaje de validación
SVM Lineal	2	93.8 %
SVM Cuadrática	2	95 %
SVM Cúbica	2	93.8 %

Cuadro 1: Resultados de porcentaje de validación para las SVM seleccionadas.

En las Figuras 30 - 35 podemos observar las matrices de confusión de los diferentes clasificadores SVM

Resultados SVM linear en fase de validación

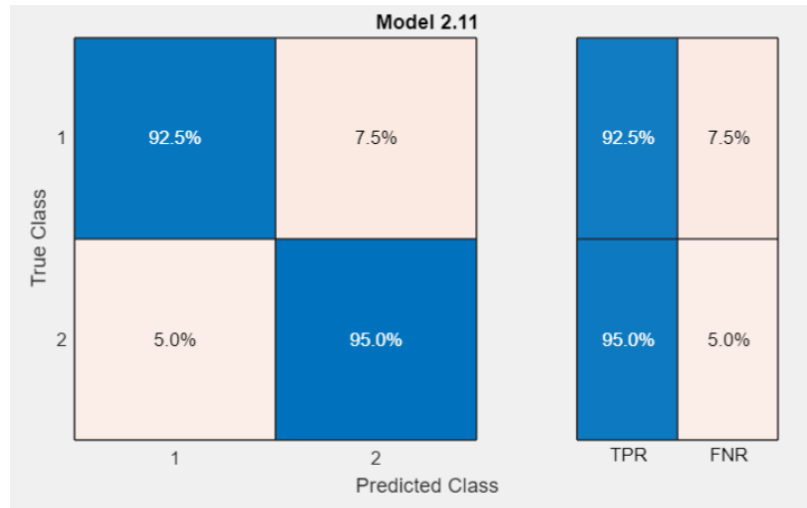


Figura 30: Matriz de confusión con etapa de validación para SVM linear con dos gestos.

Resultados SVM linear en fase de prueba

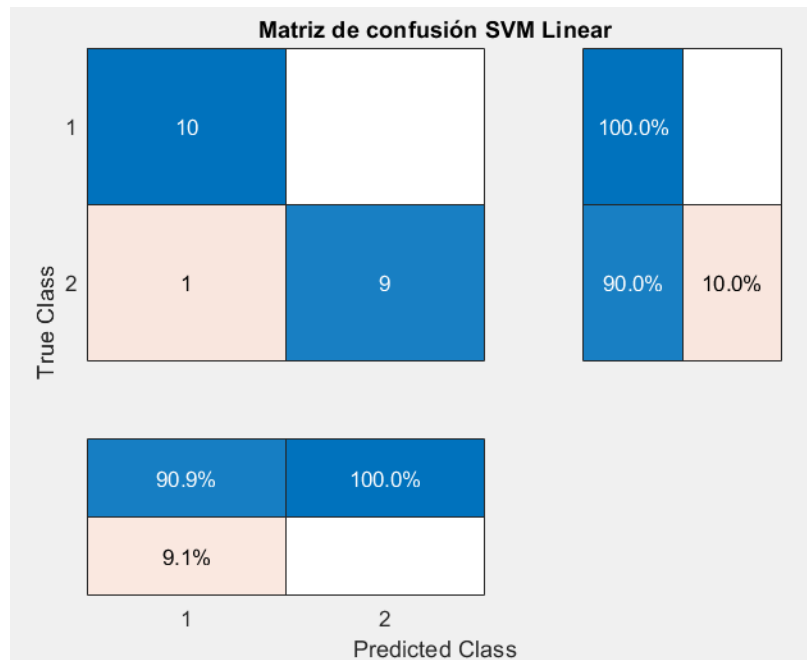


Figura 31: Matriz de confusión con etapa de prueba para SVM linear con dos gestos.

Resultados SVM cuadrática en fase de validación

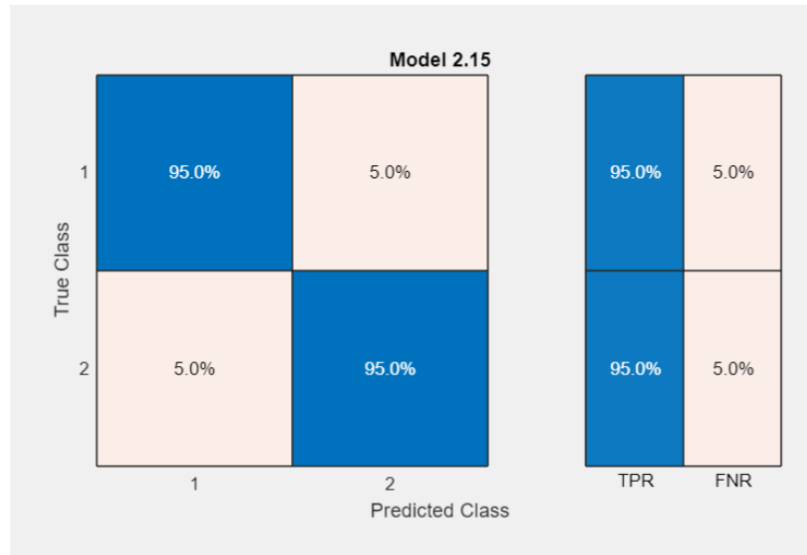


Figura 32: Matriz de confusión con etapa de validación para SVM cuadrática con dos gestos.

Resultados SVM cuadrática en fase de prueba

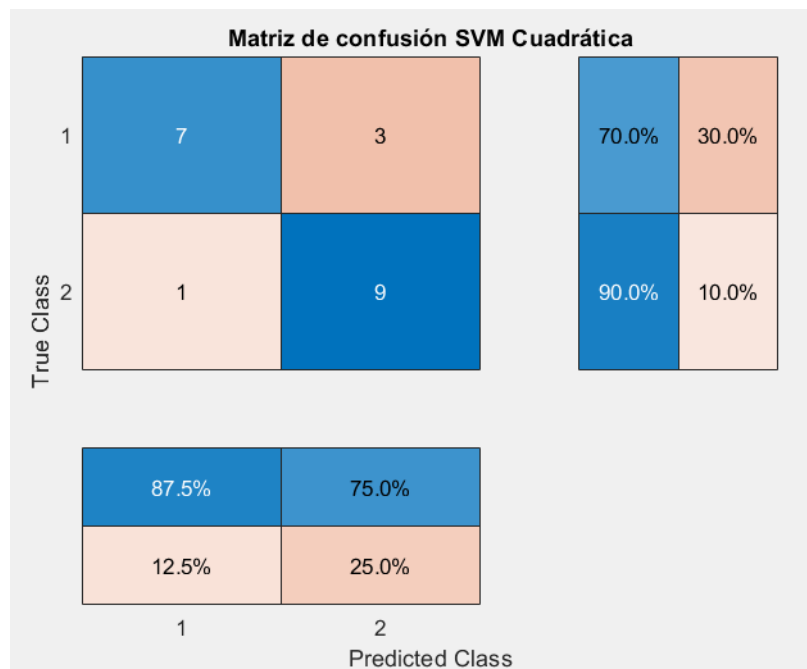


Figura 33: Matriz de confusión con etapa de prueba para SVM cuadrática con dos gestos.

Resultados SVM cúbica en fase de validación

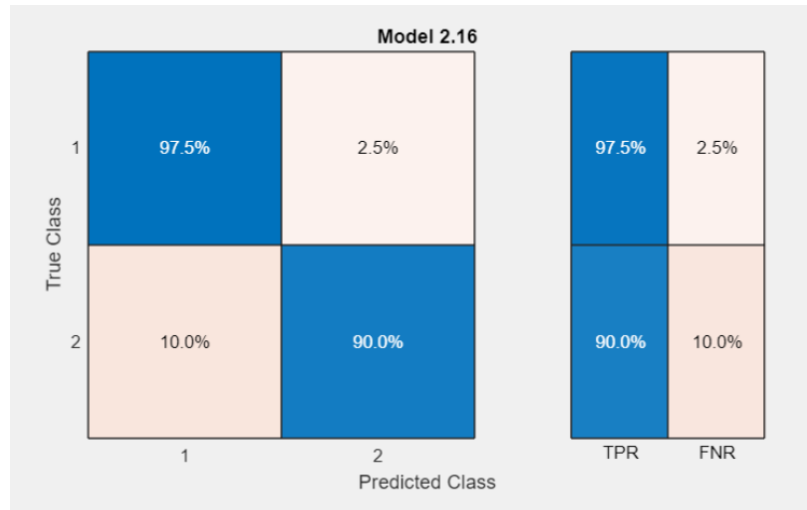


Figura 34: Matriz de confusión con etapa de validación para SVM cúbica con dos gestos.

Resultados SVM cúbica en fase de prueba

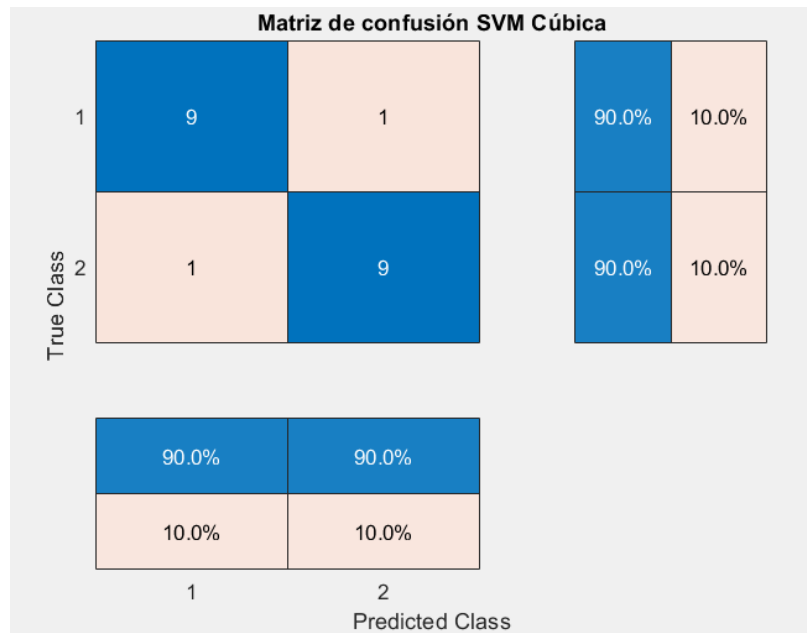


Figura 35: Matriz de confusión con etapa de prueba para SVM cúbica con dos gestos.

Según los resultados obtenidos, en etapa de validación el modelo de SVM Cuadrática produjo la mejor clasificación. Aunque, los porcentajes de predicción en la etapa de prueba fueron mejores la SVM Lineal como se puede ver en la Figura 31.

8.4.2. ANN

El mejor porcentaje de validación calculado fue el de la red neuronal de tres capas y la red de arquitectura ancha como se puede observar en el Cuadro 2.

Red Neuronal Artificial	Número de clases	Porcentaje de validación
ANN tres capas	2	91.2 %
ANN dos capas	2	90 %
ANN ancha	2	91.2 %

Cuadro 2: Resultados de porcentaje de validación para las redes neuronales seleccionadas.

En las Figuras 36 - 41 podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

Resultados ANN de tres capas en fase de validación

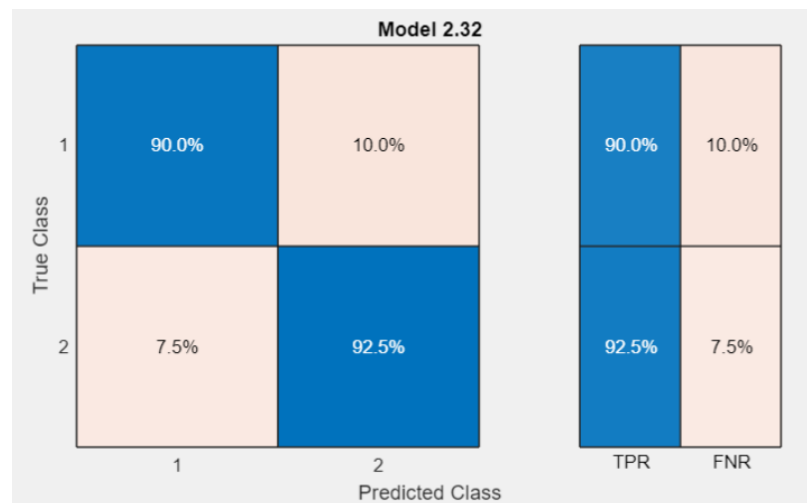


Figura 36: Matriz de confusión con etapa de validación para ANN de tres capas con dos gestos.

Resultados ANN de tres capas en fase de prueba

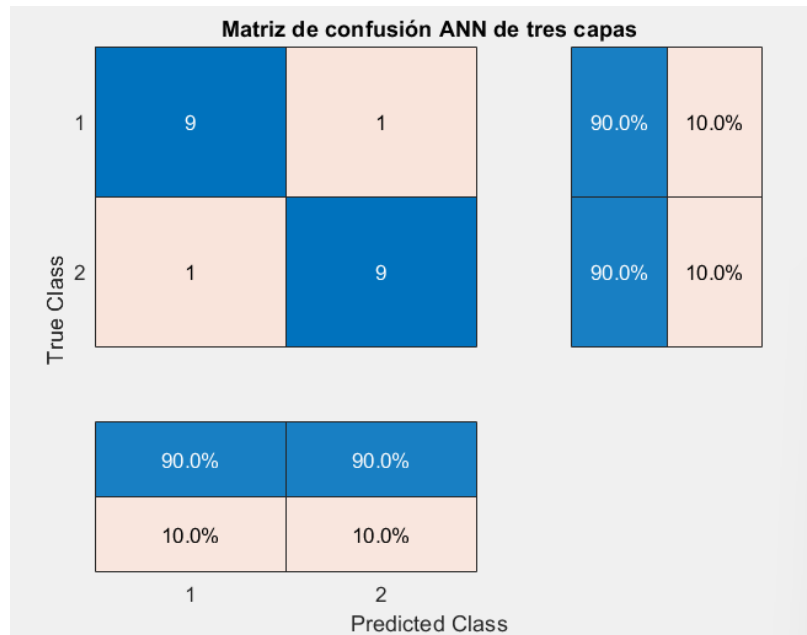


Figura 37: Matriz de confusión con etapa de prueba para ANN de tres capas con dos gestos.

Resultados ANN de dos capas en fase de validación

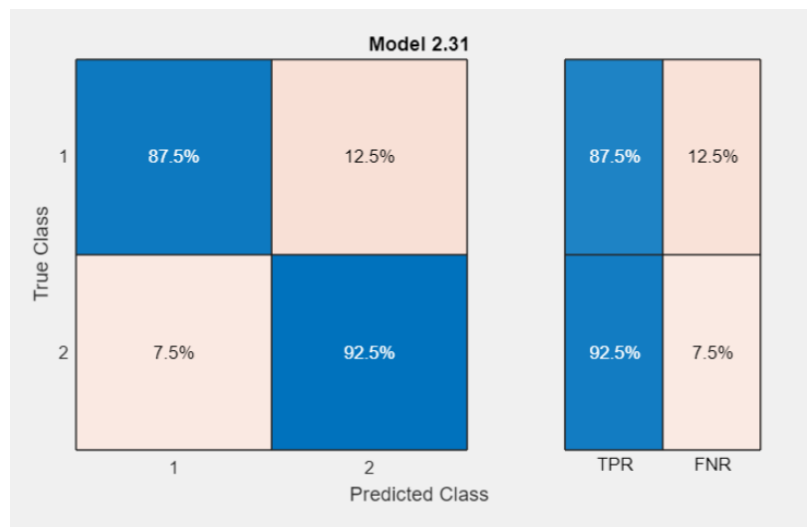


Figura 38: Matriz de confusión con etapa de validación para ANN de dos capas con dos gestos.

Resultados ANN de dos capas en fase de prueba

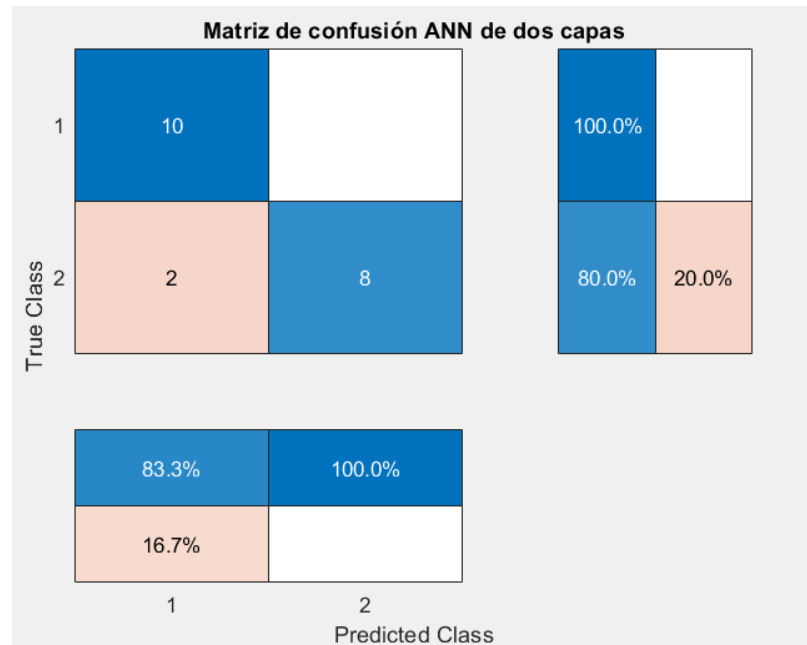


Figura 39: Matriz de confusión con etapa de prueba para ANN de dos capas con dos gestos.

Resultados ANN de arquitectura ancha en fase de validación

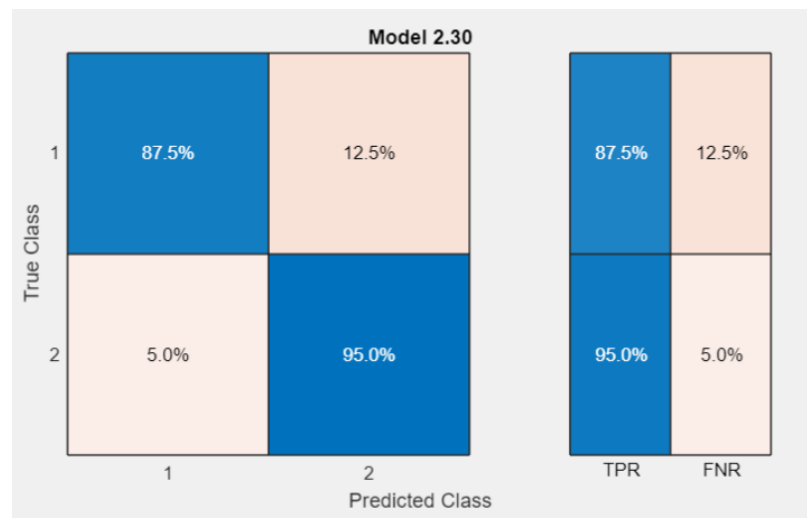


Figura 40: Matriz de confusión con etapa de validación para ANN ancha con dos gestos.

Resultados ANN de arquitectura ancha en fase de prueba

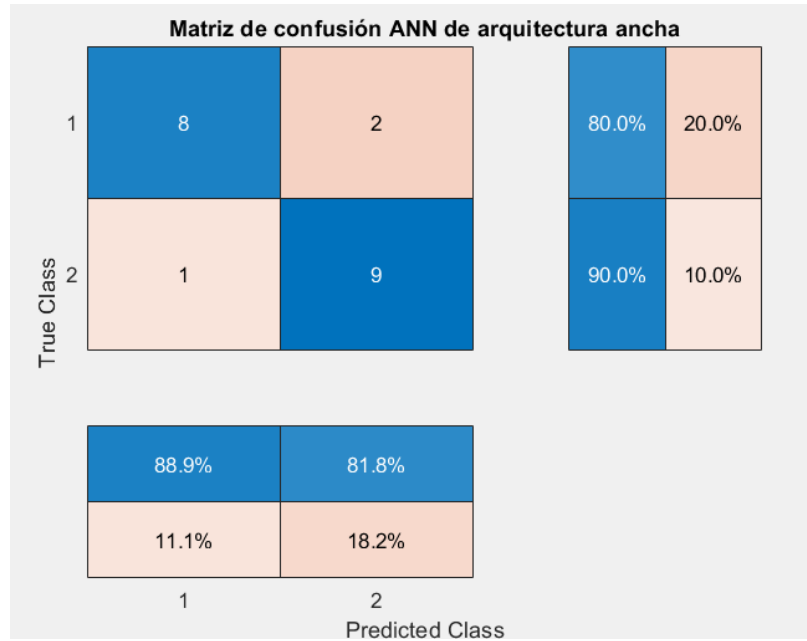


Figura 41: Matriz de confusión con etapa de validación para ANN ancha con dos gestos.

Según los resultados obtenidos, en etapa de validación y de prueba los modelos de una red de tres capas y la red de arquitectura ancha produjeron la mejor clasificación, teniendo el porcentaje más alto.

8.4.3. KNN

El mejor porcentaje de validación calculado fue el del algoritmo KNN Cúbica como se puede observar en el Cuadro 3.

k-Nearest Neighbor	Número de clases	Porcentaje de validación
KNN Pesada	2	95 %
KNN Fina	2	96.2 %
KNN Cúbica	2	97.5 %

Cuadro 3: Resultados de porcentaje de validación para las KNN seleccionadas.

En las Figuras 42 - 47 podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

Resultados del algoritmo KNN Pesada en fase de validación

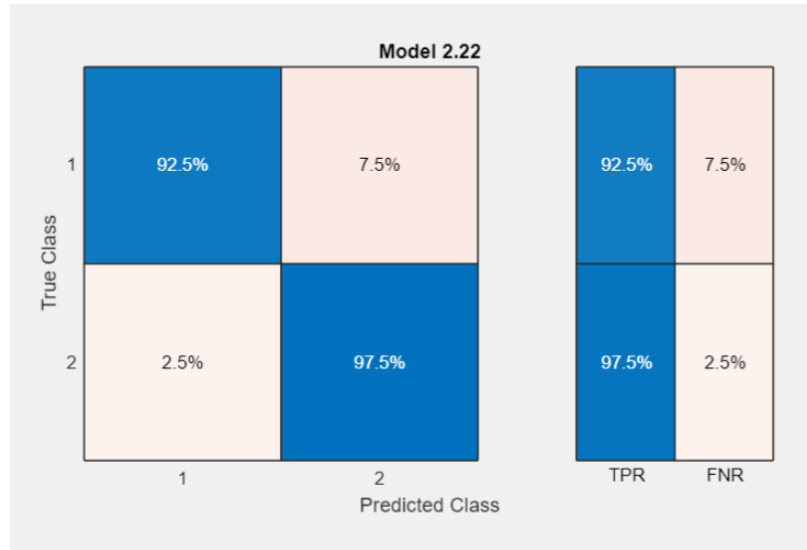


Figura 42: Matriz de confusión con etapa de validación para KNN pesada con dos gestos.

Resultados del algoritmo KNN Pesada en fase de prueba

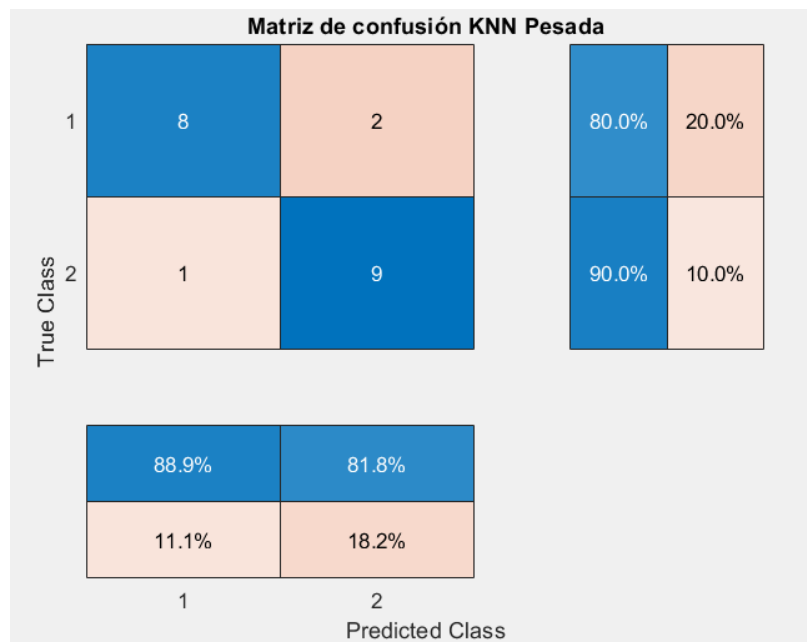


Figura 43: Matriz de confusión con etapa de prueba para KNN pesada con dos gestos.

Resultados del algoritmo KNN Fina en fase de validación

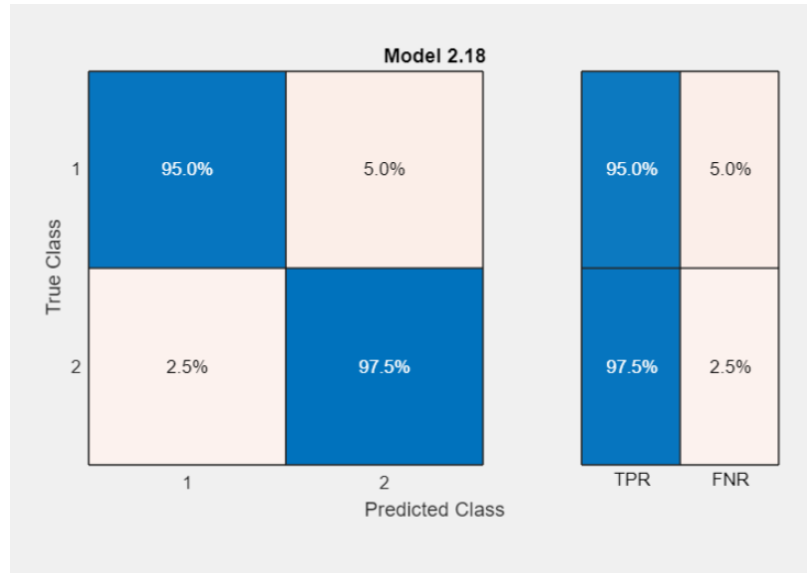


Figura 44: Matriz de confusión con etapa de validación para KNN fina con dos gestos.

Resultados del algoritmo KNN Fina en fase de prueba

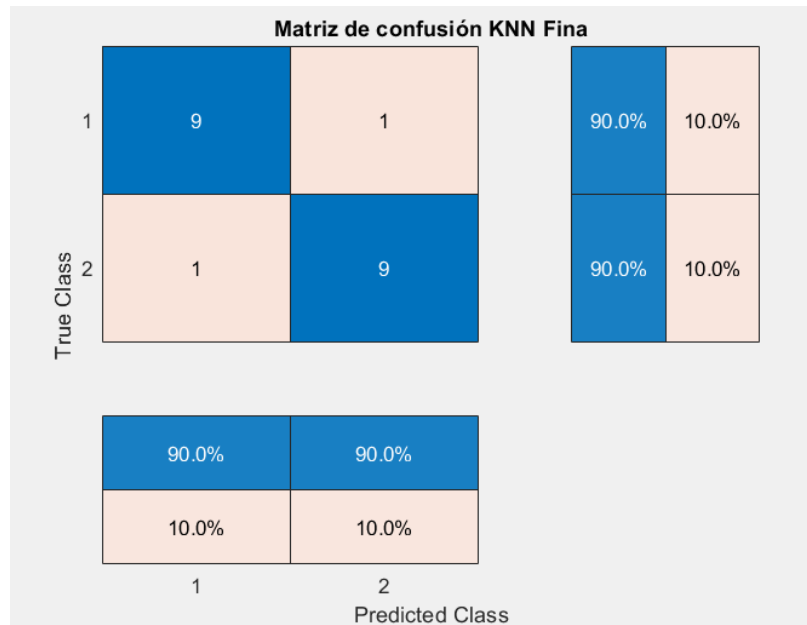


Figura 45: Matriz de confusión con etapa de prueba para KNN fina con dos gestos.

Resultados del algoritmo KNN cúbica en fase de validación

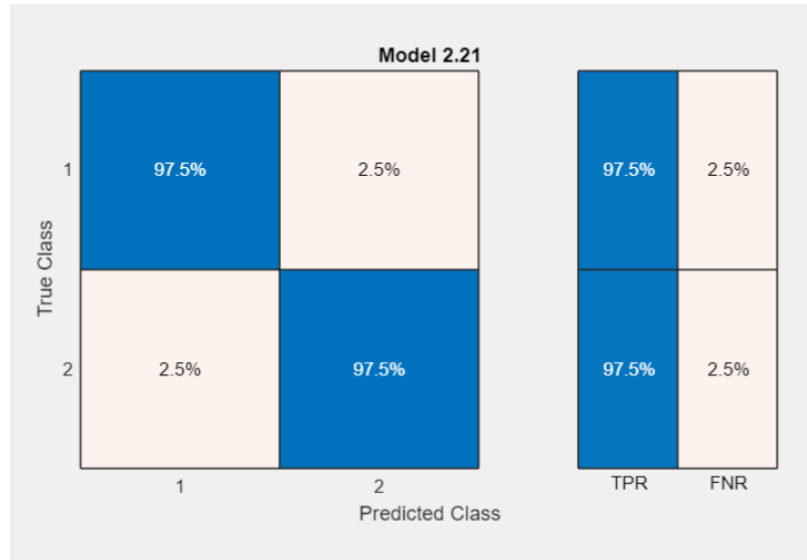


Figura 46: Matriz de confusión con etapa de validación para KNN cúbica con dos gestos.

Resultados del algoritmo KNN cúbica en fase de prueba

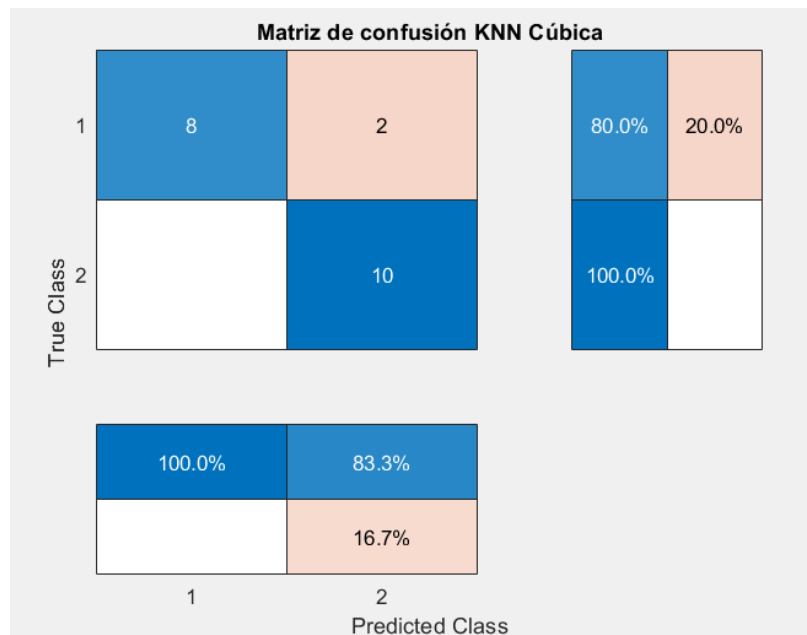


Figura 47: Matriz de confusión con etapa de prueba para KNN cúbica con dos gestos.

En comparación a los otros modelos, los algoritmos KNN resultaron tener los mejores porcentajes en etapa de validación. Además, los resultados en etapa de prueba demuestran su consistencia a la hora de clasificar, manteniendo porcentajes similares en todos los casos.

8.5. Resultados de clasificación de 3 gestos con un sensor

Para la evaluación, se utilizó una recolección de 120 muestras, distribuidas en 40 para cada clase. Adicionalmente, se creó una base de datos de prueba con 60 muestras, divididas en 20 por clase, para verificar el rendimiento de los clasificadores.

8.5.1. SVM

El porcentaje de validación obtenido se puede observar en el Cuadro 4.

SVM	Número de clases	Porcentaje de validación
SVM Linear	3	95.7 %
SVM Cuadrática	3	93.83 %
SVM Cúbica	3	93.83 %

Cuadro 4: Resultados de porcentaje de validación para las SVM seleccionadas.

En las Figuras 48 - 53 podemos observar las matrices de confusión de los diferentes clasificadores SVM.

Resultados SVM linear en fase de validación

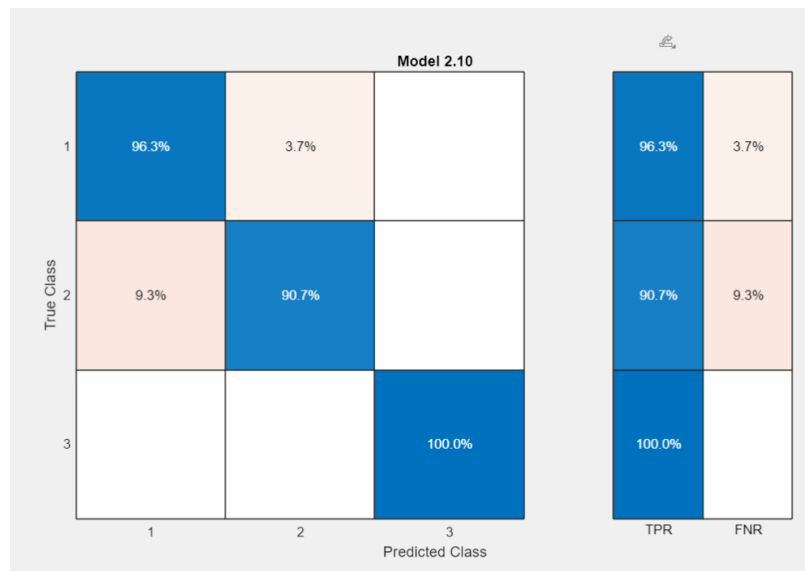


Figura 48: Matriz de confusión con etapa de validación para SVM linear.

Resultados SVM linear en fase de prueba

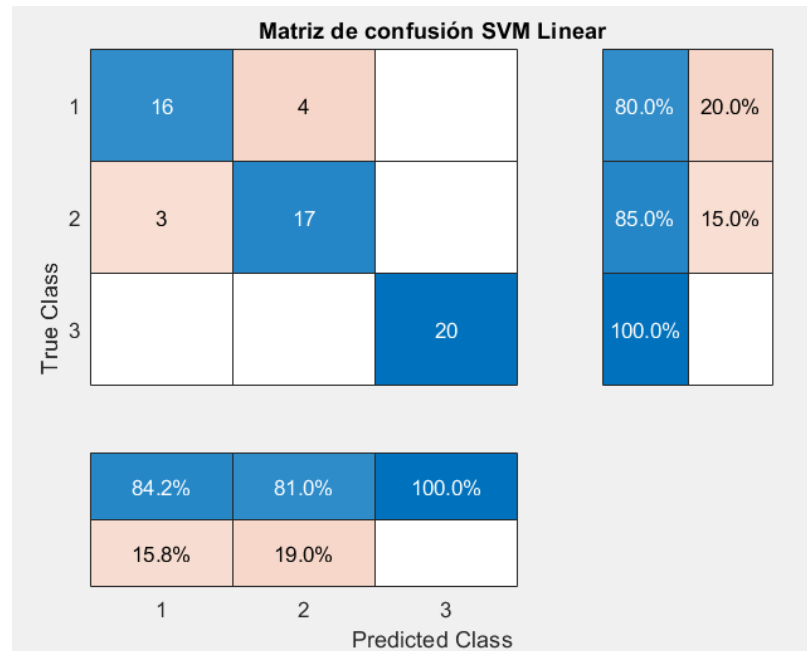


Figura 49: Matriz de confusión con etapa de prueba para SVM linear con tres gestos.

Resultados SVM cuadrática en fase de validación

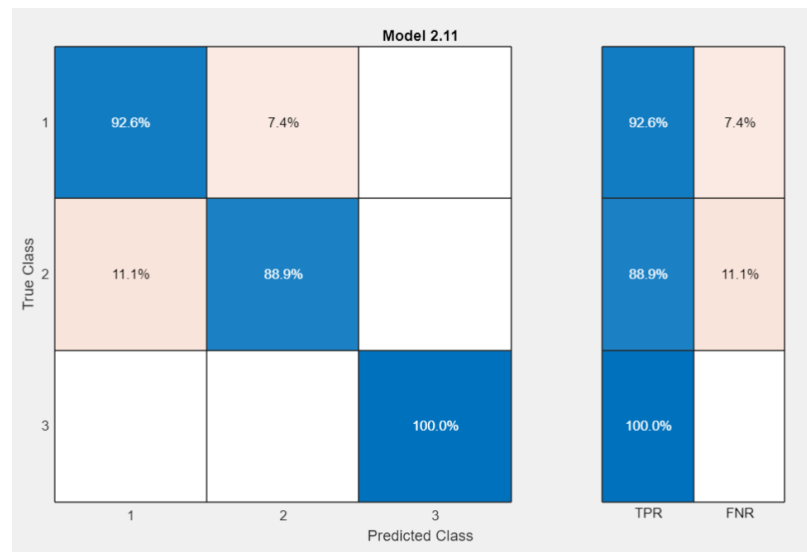


Figura 50: Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.

Resultados SVM cuadrática en fase de prueba

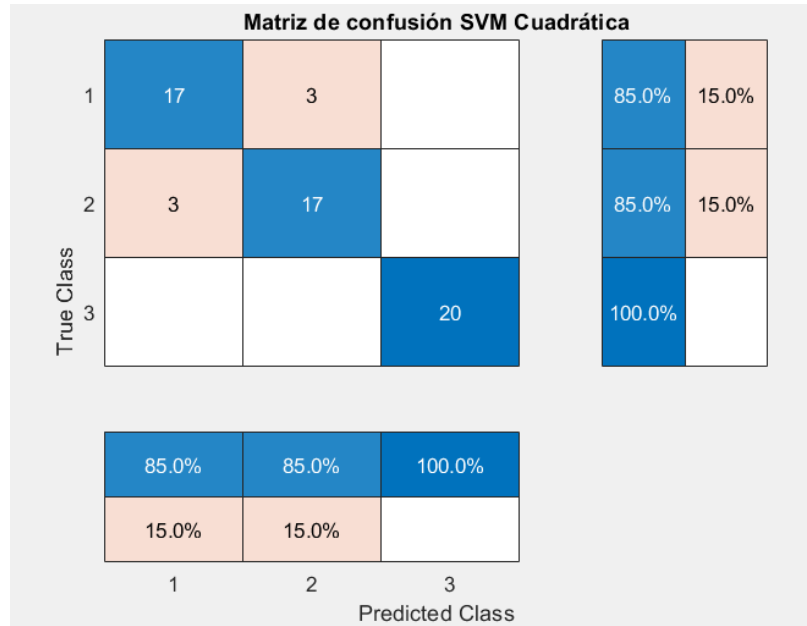


Figura 51: Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos.

Resultados SVM cúbica en fase de validación

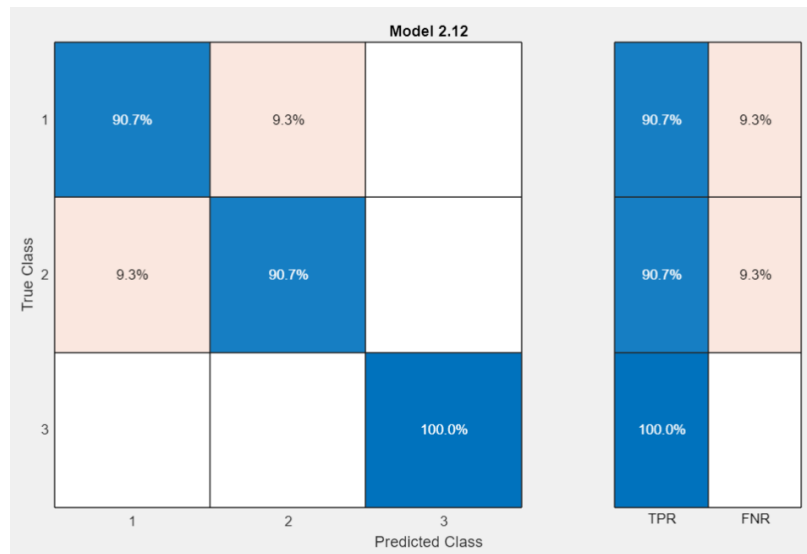


Figura 52: Matriz de confusión con etapa de validación para SVM cúbica con tres gestos.

Resultados SVM cúbica en fase de prueba

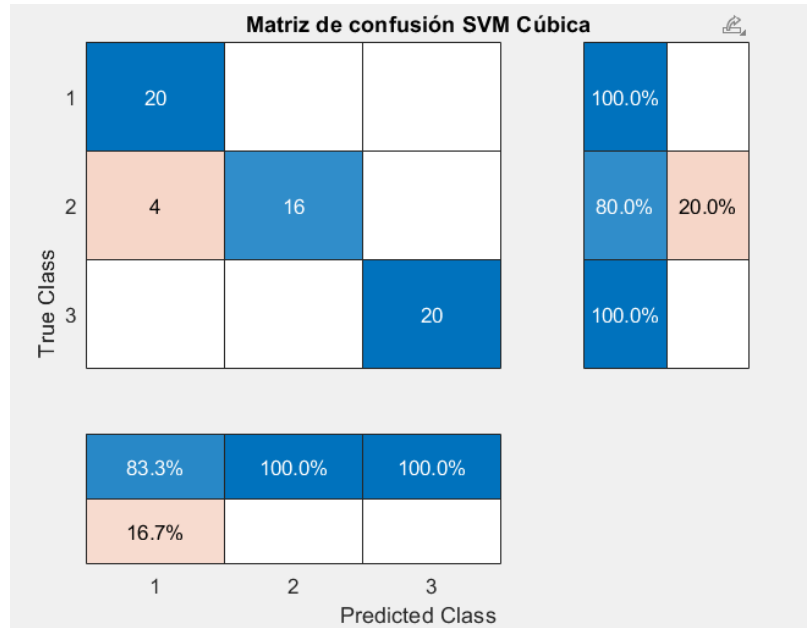


Figura 53: Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos.

Según los resultados obtenidos, en etapa de validación el modelo de SVM Linear produjo la mejor clasificación. Aunque, los porcentajes de predicción en la etapa de prueba fueron mejores la SVM cúbica como se puede ver en la Figura 53.

8.5.2. ANN

El mejor porcentaje de validación calculado fue el de la red neuronal de tres capas como se puede observar en el Cuadro 5.

Red Neuronal Artificial	Número de clases	Porcentaje de validación
ANN tres capas	3	92.6 %
ANN dos capas	3	92 %
ANN ancha	3	92 %

Cuadro 5: Resultados de porcentaje de validación para las redes neuronales seleccionadas.

En las Figuras 54 - 59 podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

Resultados ANN de tres capas en fase de validación

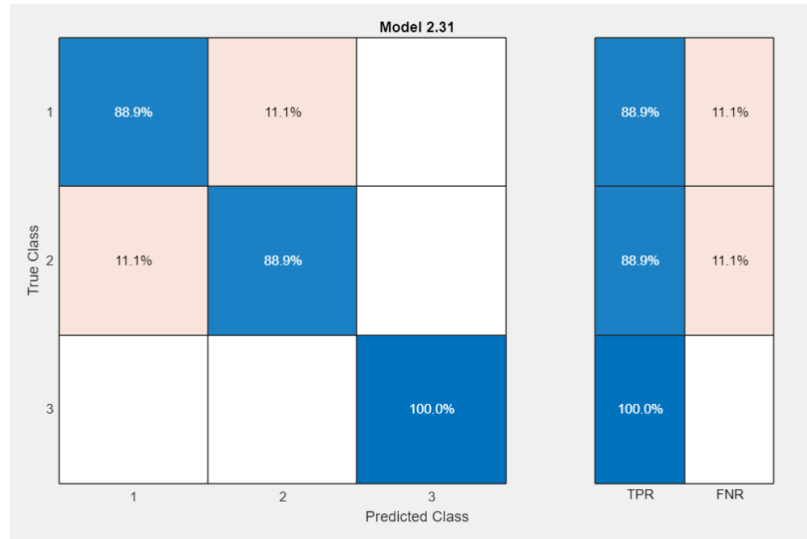


Figura 54: Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.

Resultados ANN de tres capas en fase de prueba

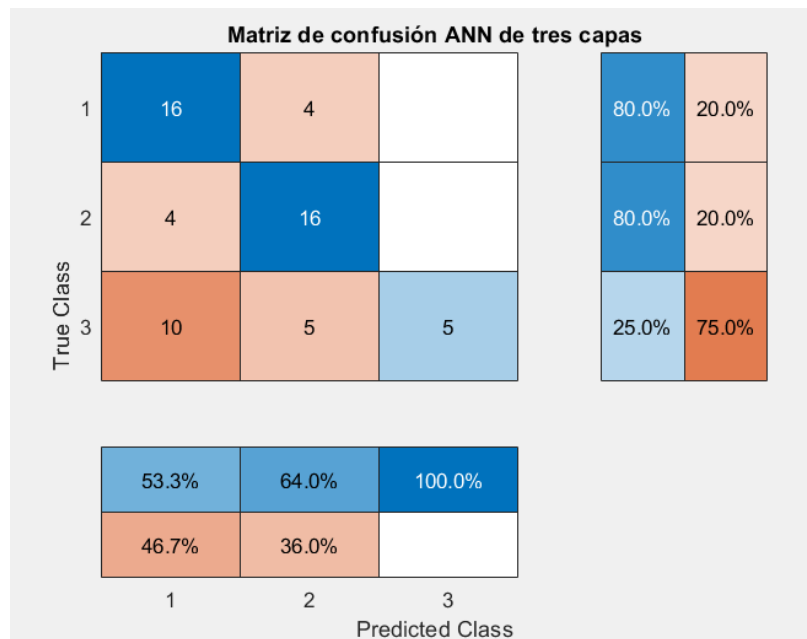


Figura 55: Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.

Resultados ANN de dos capas en fase de validación

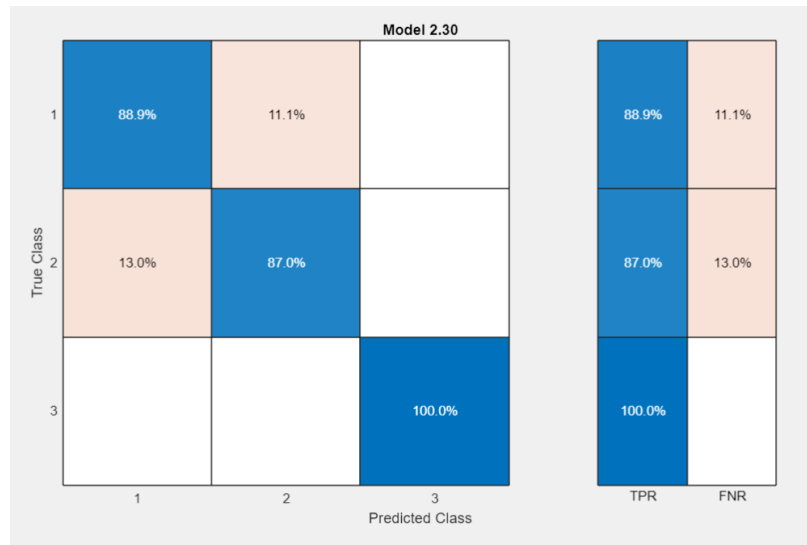


Figura 56: Matriz de confusión con etapa de validación para ANN de dos capas con tres gestos.

Resultados ANN de dos capas en fase de prueba

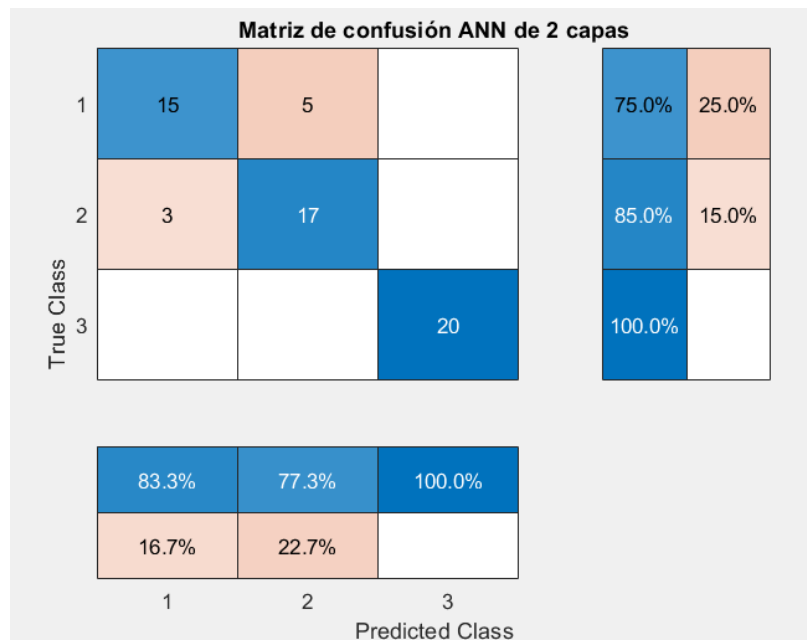


Figura 57: Matriz de confusión con etapa de prueba para ANN de dos capas con tres gestos.

Resultados ANN de arquitectura ancha en fase de validación

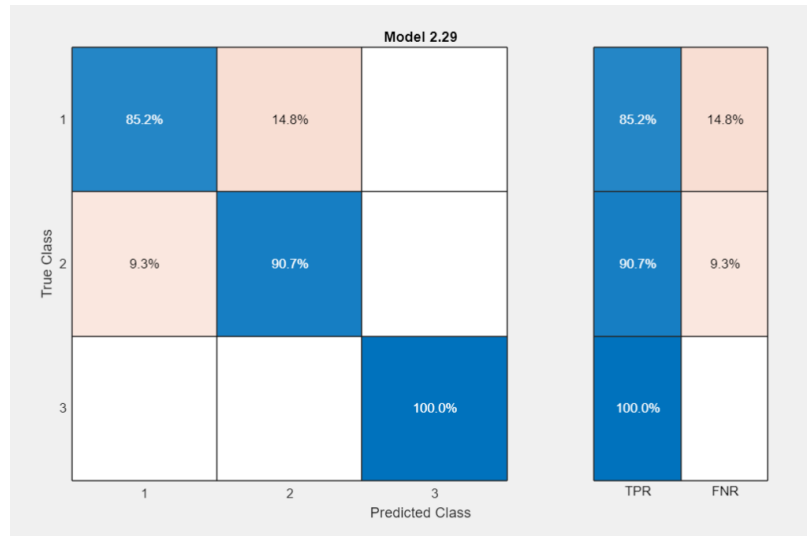


Figura 58: Matriz de confusión con etapa de validación para ANN ancha con tres gestos.

Resultados ANN de arquitectura ancha en fase de prueba

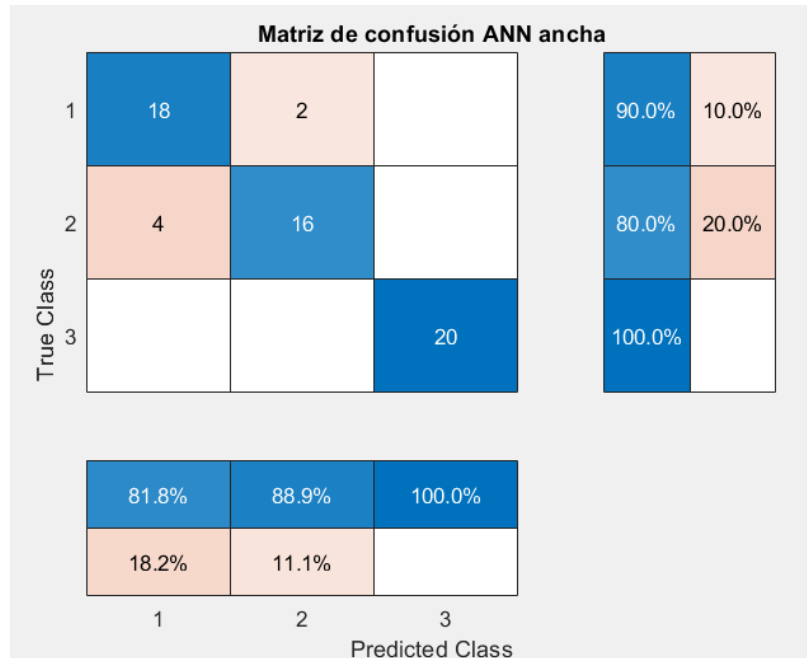


Figura 59: Matriz de confusión con etapa de prueba para ANN ancha con tres gestos.

En el caso de los algoritmos ANN podemos observar que en porcentajes de validación la ANN de tres capas tiene el más alto. Cabe mencionar que los porcentajes de predicción en la etapa de prueba fueron mejores para los porcentajes para la ANN de arquitectura ancha como se puede ver en la Figura 59.

8.5.3. KNN

El mejor porcentaje de validación calculado fue el del algoritmo KNN Pesada como se puede observar en el Cuadro 6.

k-Nearest Neighbor	Número de clases	Porcentaje de validación
KNN Pesada	3	92 %
KNN Fina	3	89.5 %
KNN Cúbica	3	88.9 %

Cuadro 6: Resultados de porcentaje de validación para las KNN seleccionadas.

En las Figuras 60 - 65 podemos observar las matrices de confusión de los diferentes algoritmos creados.

Resultados del algoritmo KNN Pesada en fase de validación

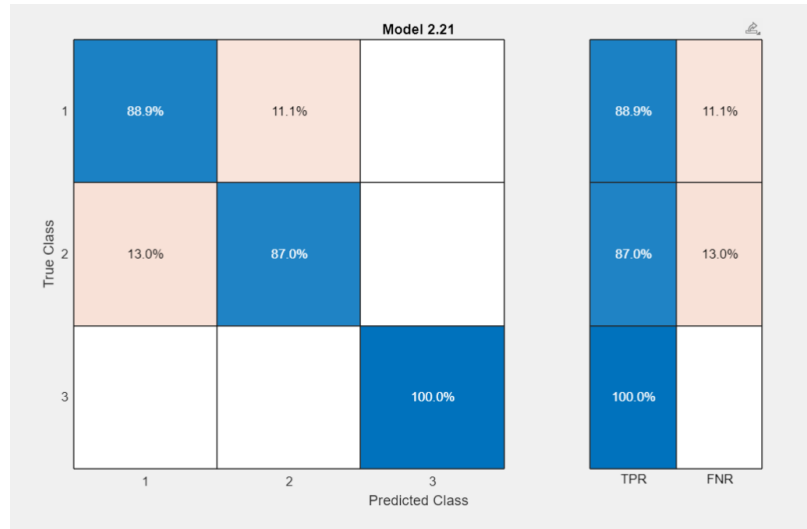


Figura 60: Matriz de confusión con etapa de validación para KNN pesada con tres gestos.

Resultados del algoritmo KNN Pesada en fase de prueba

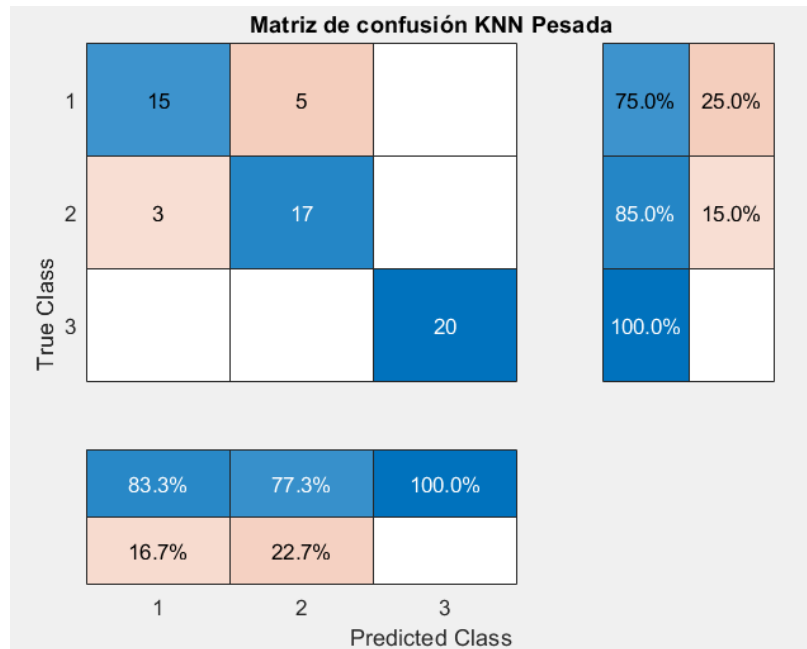


Figura 61: Matriz de confusión con etapa de prueba para KNN pesada con tres gestos.

Resultados del algoritmo KNN Fina en fase de validación

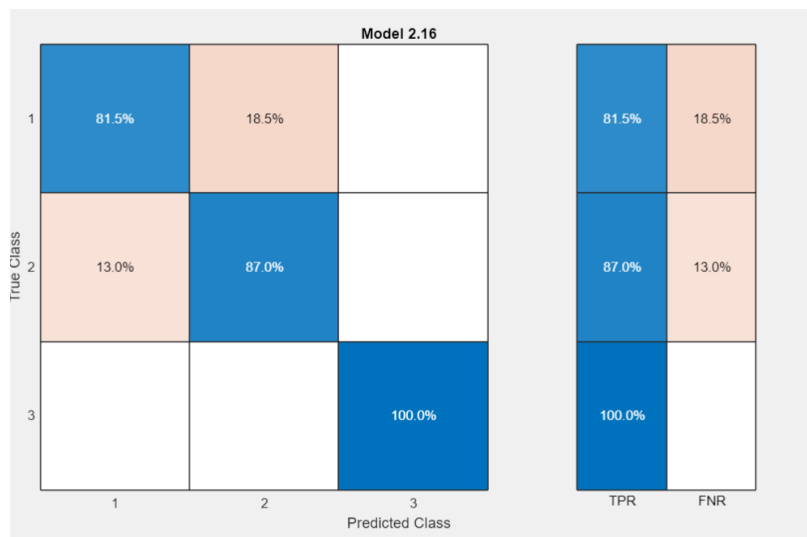


Figura 62: Matriz de confusión con etapa de validación para KNN fina con tres gestos.

Resultados del algoritmo KNN Fina en fase de prueba

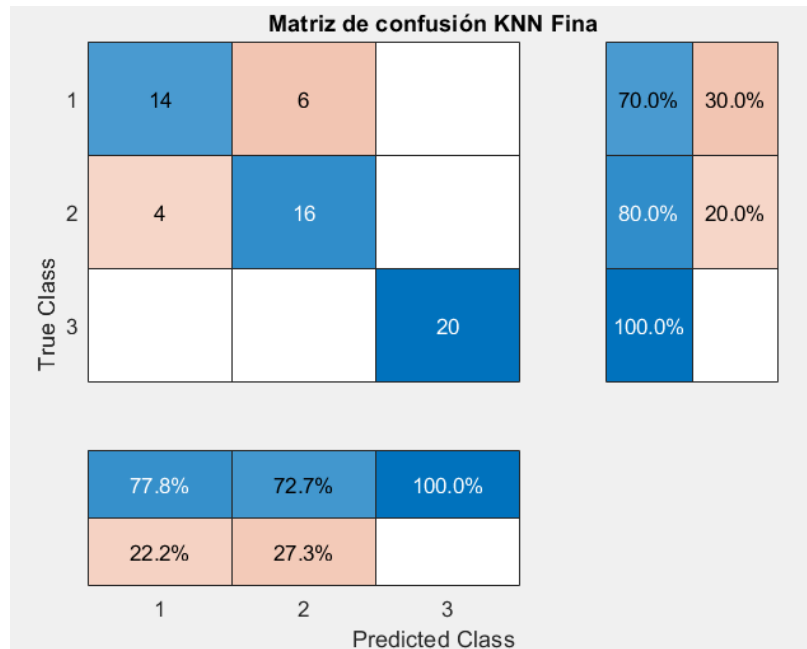


Figura 63: Matriz de confusión con etapa de prueba para KNN fina con tres gestos.

Resultados del algoritmo KNN cúbica en fase de validación

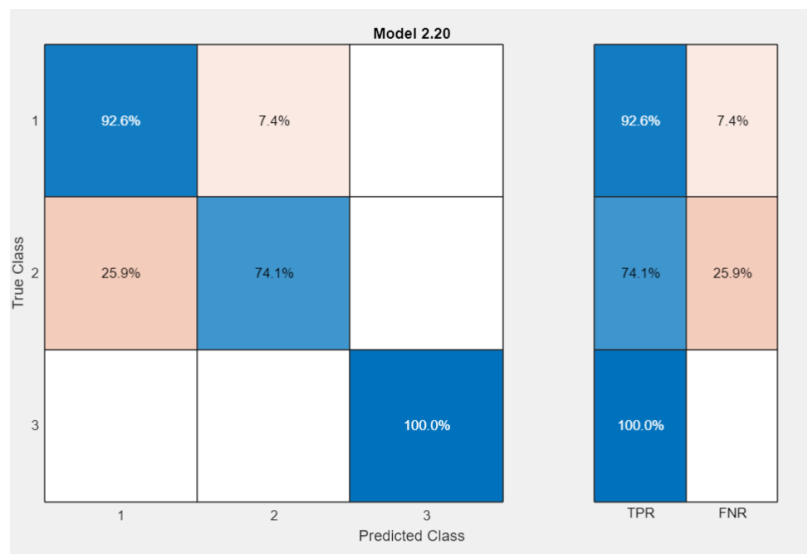


Figura 64: Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.

Resultados del algoritmo KNN cúbica en fase de prueba

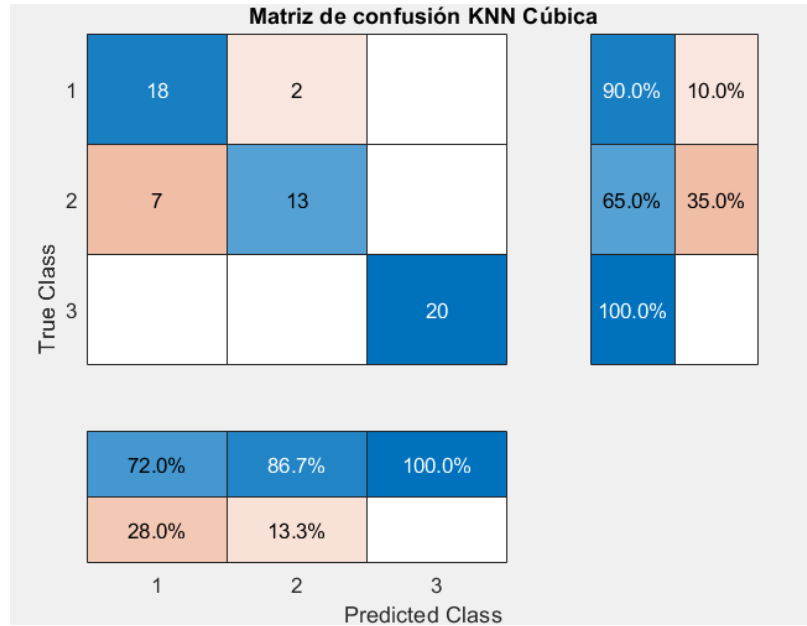


Figura 65: Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos.

En el caso de los algoritmos KNN podemos observar que en porcentajes de validación el modelo KNN Pesado tiene el más alto. Cabe mencionar que los porcentajes de predicción en la etapa de prueba fueron mejores para la KNN cúbica como se puede ver en la Figura 65.

8.6. Resultados de clasificación de 3 gestos con dos sensores

Para la evaluación, se utilizó una recolección de 120 muestras, distribuidas en 40 para cada clase. Adicionalmente, se creó una base de datos de prueba con 60 muestras, divididas en 20 por clase, para verificar el rendimiento de los clasificadores.

8.6.1. SVM

El porcentaje de validación obtenido se puede observar en el Cuadro 7.

SVM	Número de clases	Porcentaje de validación
SVM Linear	3	92.6 %
SVM Cuadrática	3	92.6 %
SVM Cúbica	3	94.2 %

Cuadro 7: Resultados de porcentaje de validación para las SVM seleccionadas.

En las Figuras 66 - 71 podemos observar las matrices de confusión de los diferentes clasificadores SVM

Resultados SVM linear en fase de validación

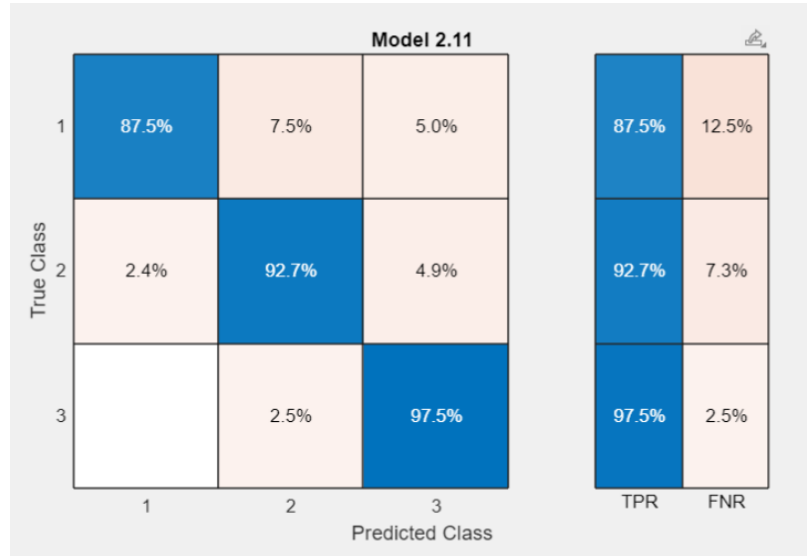


Figura 66: Matriz de confusión con etapa de validación para SVM linear con tres gestos.

Resultados SVM linear en fase de prueba

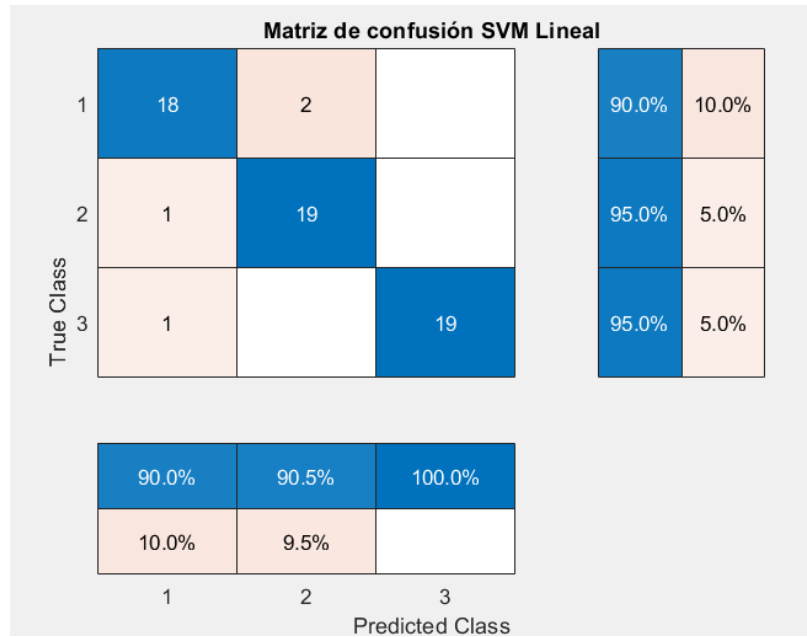


Figura 67: Matriz de confusión con etapa de prueba para SVM linear con tres gestos.

Resultados SVM cuadrática en fase de validación

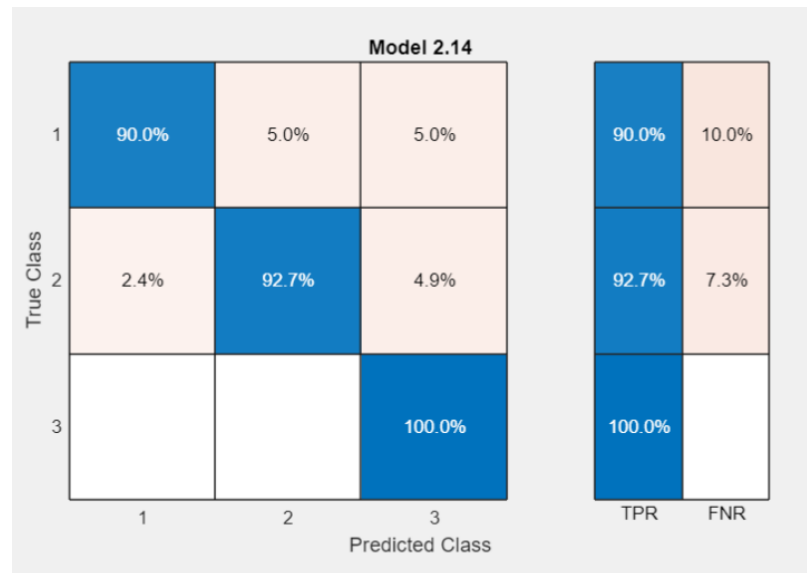


Figura 68: Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.

Resultados SVM cuadrática en fase de prueba

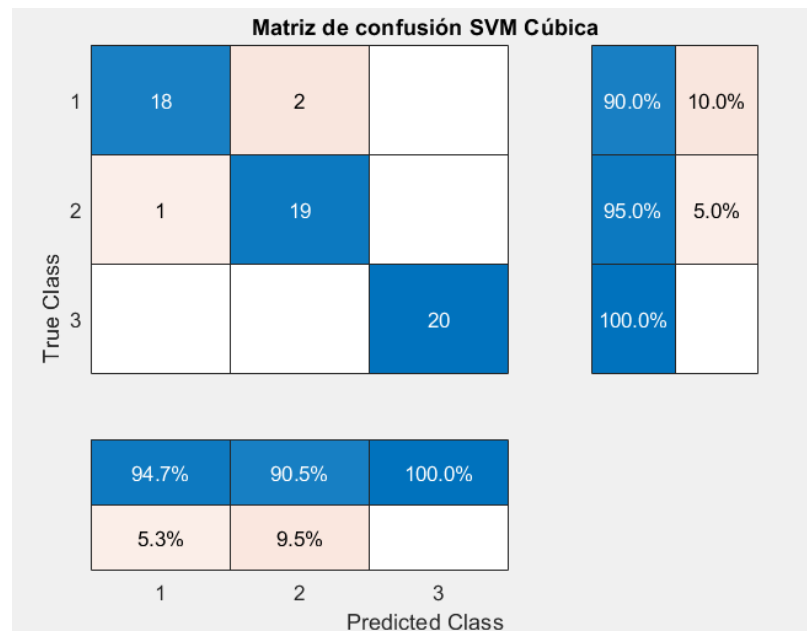


Figura 69: Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos.

Resultados SVM cúbica en fase de validación

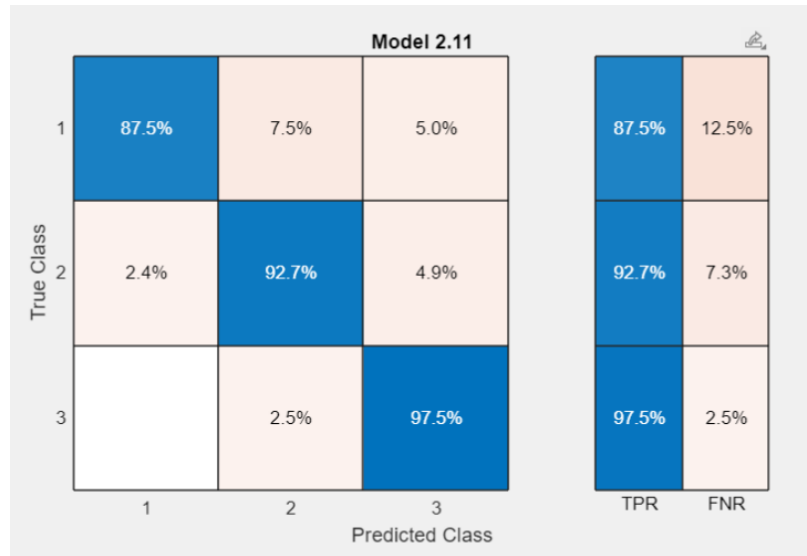


Figura 70: Matriz de confusión con etapa de validación para SVM cúbica con tres gestos.

Resultados SVM cúbica en fase de prueba

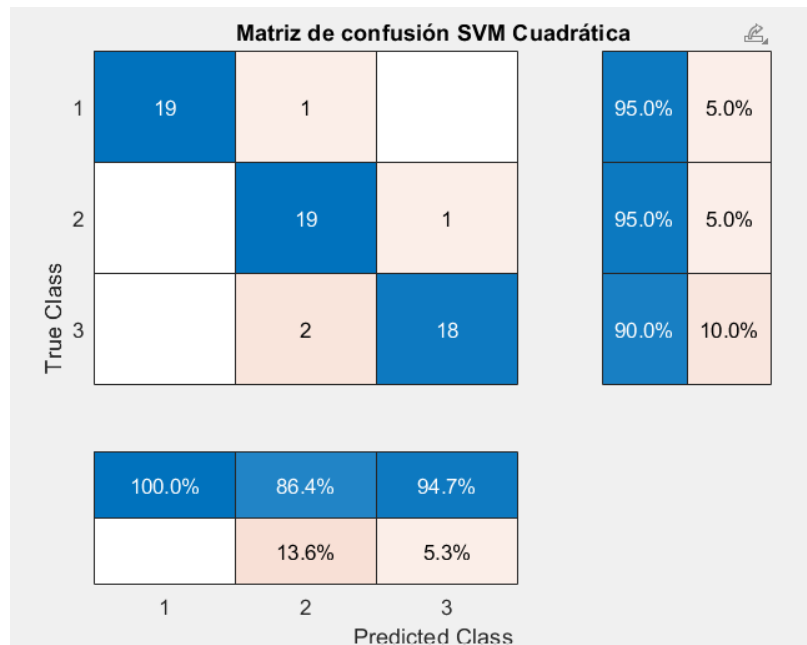


Figura 71: Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos.

Según los resultados obtenidos, en etapa de validación el modelo de SVM Cúbica produjo la mejor clasificación. Aunque, los porcentajes de predicción en la etapa de prueba demuestran que con los resultados entre los modelos es pareja, mostranso similitud en el porcentaje de clasificación.

8.6.2. ANN

El mejor porcentaje de validación calculado fue el de la red neuronal de dos capas como se puede observar en el Cuadro 8.

Red Neuronal Artificial	Número de clases	Porcentaje de validación
ANN tres capas	3	90.1 %
ANN dos capas	3	91.7 %
ANN ancha	3	90.9 %

Cuadro 8: Resultados de porcentaje de validación para las redes neuronales seleccionadas.

En las Figuras 72 - 77 podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

Resultados ANN de tres capas en fase de validación

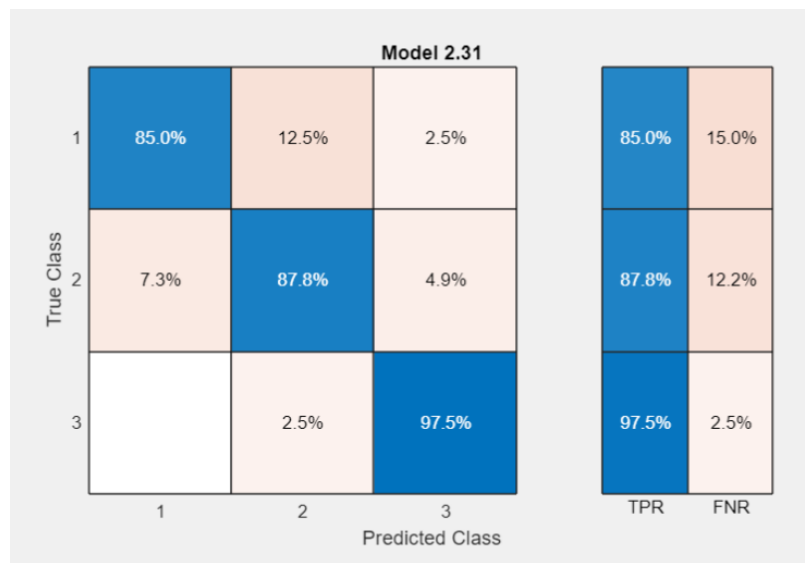


Figura 72: Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.

Resultados ANN de tres capas en fase de prueba

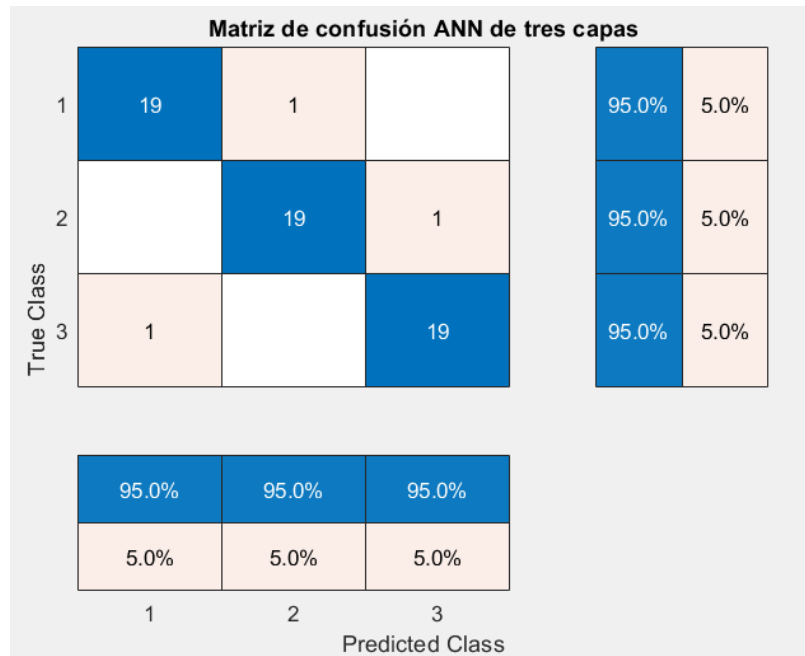


Figura 73: Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.

Resultados ANN de dos capas en fase de validación

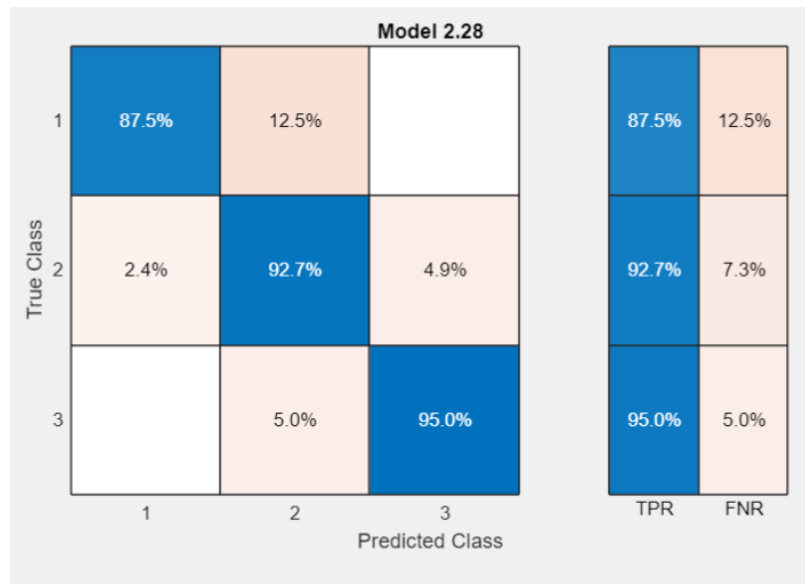


Figura 74: Matriz de confusión con etapa de validación para ANN de dos capas con dos gestos.

Resultados ANN de dos capas en fase de prueba

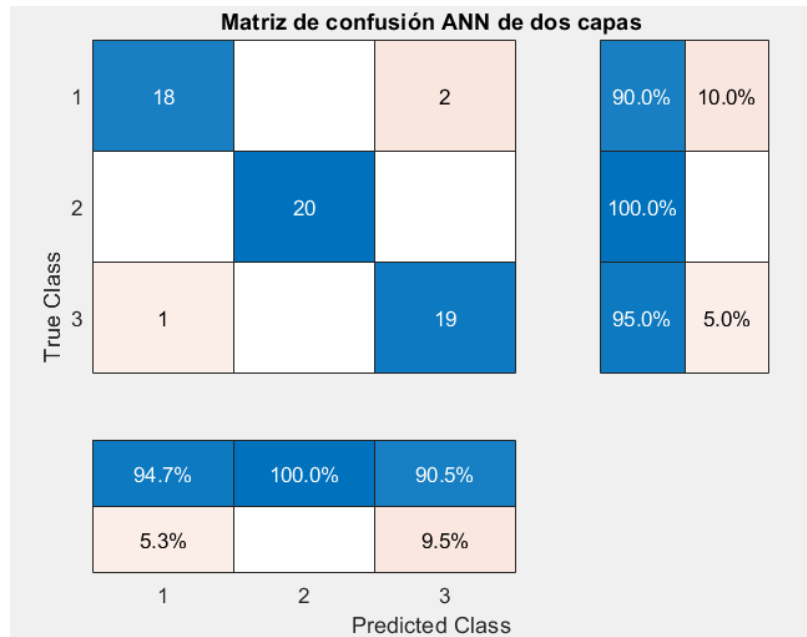


Figura 75: Matriz de confusión con etapa de prueba para ANN de dos capas con dos gestos.

Resultados ANN de arquitectura ancha en fase de validación

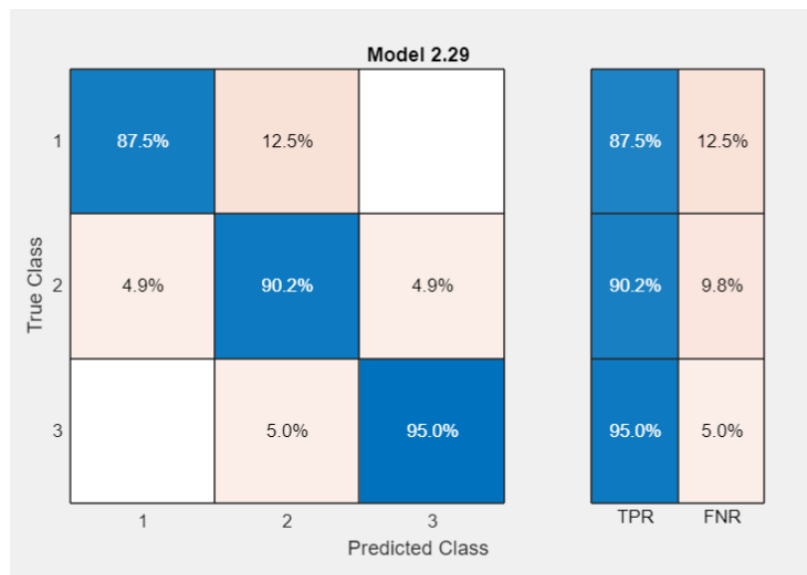


Figura 76: Matriz de confusión con etapa de validación para ANN ancha con tres gestos.

Resultados ANN de arquitectura ancha en fase de prueba

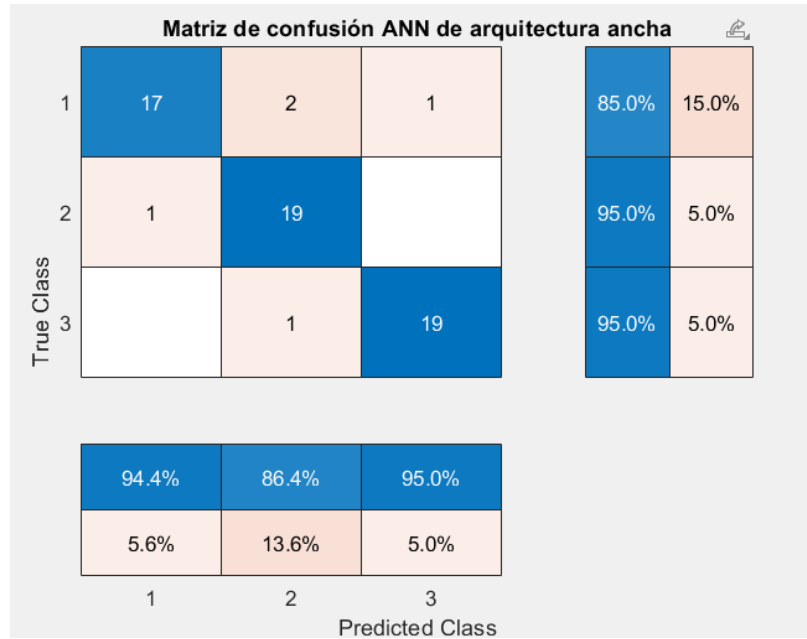


Figura 77: Matriz de confusión con etapa de prueba para ANN ancha con tres gestos.

Según los resultados obtenidos, en etapa de validación el modelo ANN de dos capas produjo la mejor clasificación. Los resultados en fase de prueba apoyan ese resultado, teniendo pocos errores, pero mostrando un gesto con un 100 por ciento de clasificación, como se puede observar Figura 75.

8.6.3. KNN

El mejor porcentaje de validación calculado fue el del algoritmo KNN Pesada como se puede observar en el Cuadro 9.

k-Nearest Neighbor	Número de clases	Porcentaje de validación
KNN Pesada	3	90.1 %
KNN Fina	3	89.3 %
KNN Cúbica	3	88.4 %

Cuadro 9: Resultados de porcentaje de validación para las KNN seleccionadas.

En las Figuras 78 - 83 podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

Resultados del algoritmo KNN Pesada en fase de validación

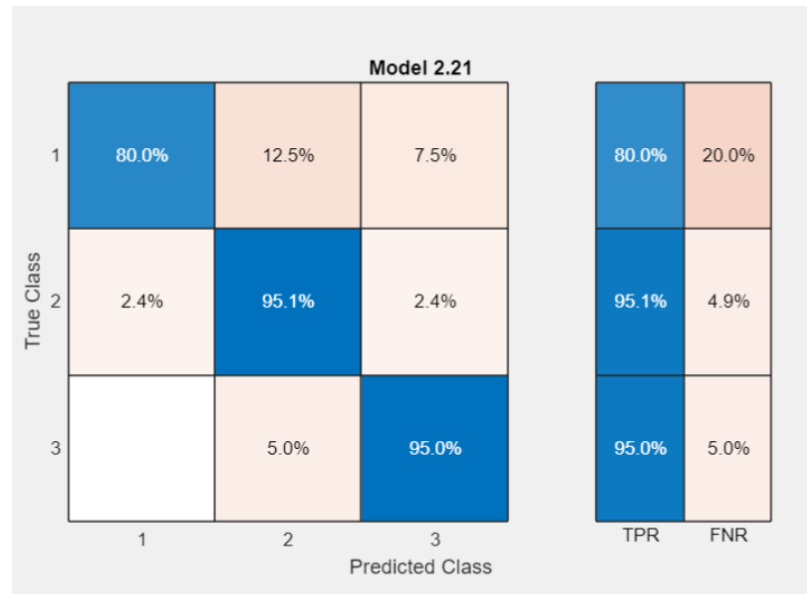


Figura 78: Matriz de confusión con etapa de validación para KNN pesada con tres gestos.

Resultados del algoritmo KNN Pesada en fase de prueba

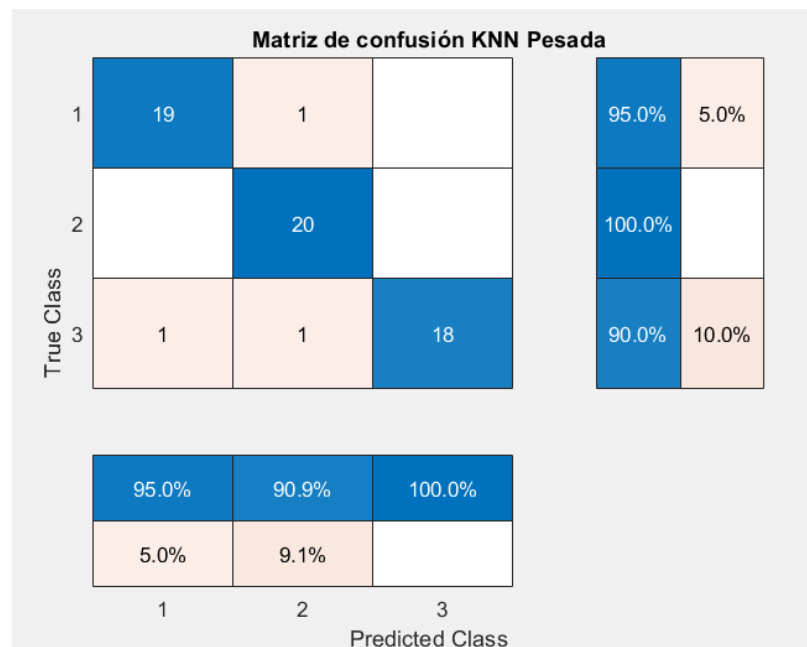


Figura 79: Matriz de confusión con etapa de prueba para KNN pesada con tres gestos.

Resultados del algoritmo KNN Fina en fase de validación

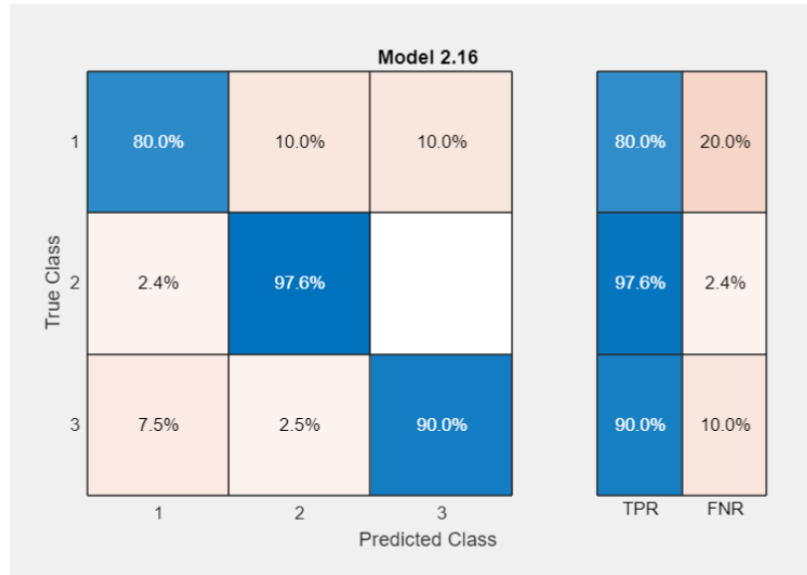


Figura 80: Matriz de confusión con etapa de validación para KNN fina con tres gestos.

Resultados del algoritmo KNN Fina en fase de prueba

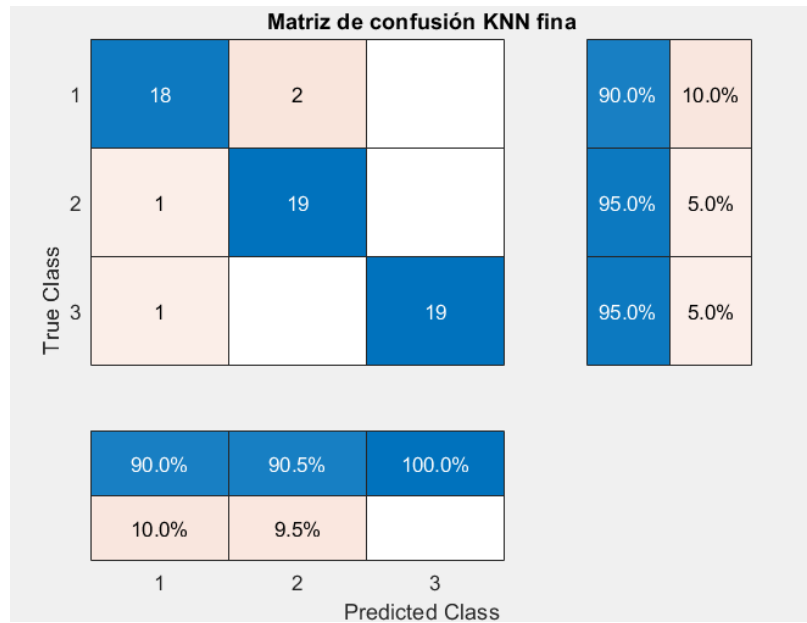


Figura 81: Matriz de confusión con etapa de prueba para KNN fina con tres gestos.

Resultados del algoritmo KNN cúbica en fase de validación

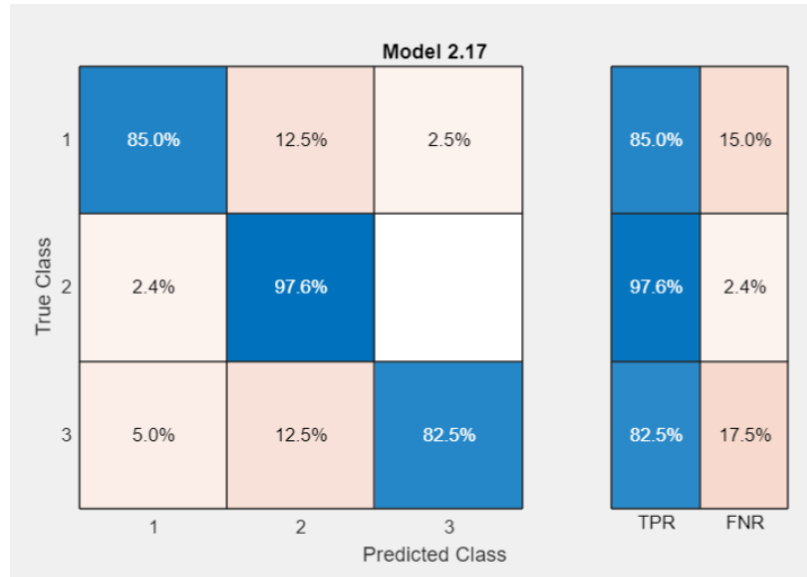


Figura 82: Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.

Resultados del algoritmo KNN cúbica en fase de prueba

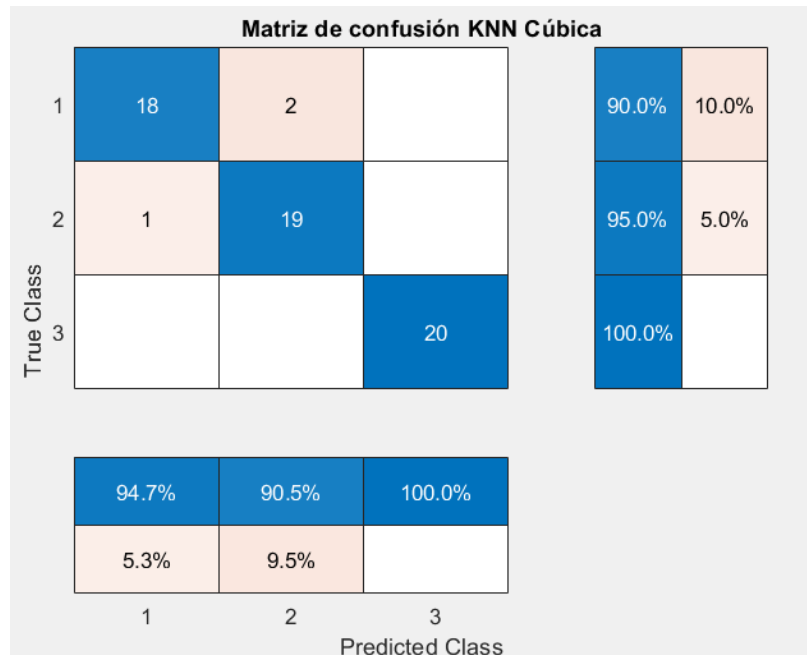


Figura 83: Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos.

Según los resultados obtenidos, en etapa de validación el modelo KNN pesada produjo la mejor clasificación. Aunque, los porcentajes de predicción tienen mejores resultados en

comparación entre los tres modelos. Es bueno mencionar que en este caso los modelos KNN, presentaron más errores a comparación de las SVM y ANN.

8.7. Resultados obtenidos con participantes externos

Considerando los resultados obtenidos en las secciones 8.4 – 8.6, se seleccionaron los siguientes modelos de clasificación para ser evaluados con participantes externos. Los modelos seleccionados para el caso de un solo sensor fueron: SVM lineal, una red neuronal artificial (ANN) de tres capas y KNN cúbica. Para el caso de dos sensores, los modelos seleccionados fueron: SVM cúbica, una ANN de dos capas y KNN pesada.

Con base en estos modelos, se realizaron pruebas adicionales utilizando señales obtenidas de distintos participantes con el fin de evaluar el desempeño de los clasificadores ante los cambios de usuario a usuario.

Las colecciones de señales fueron de 2 gestos y fueron recolectadas con un solo sensor. Los gestos utilizados son los mismos a los gestos mencionados en la sección 8.2.1.

Los modelos de clasificación fueron entrenados utilizando un conjunto de 60 señales, distribuidas equitativamente entre el gesto 1 (30 señales) y el gesto 2 (30 señales). Adicionalmente, se evaluó su desempeño con un conjunto de 20 señales de prueba independientes, no incluidas en el conjunto de entrenamiento.

Al igual que en la sección anterior, los resultados del conjunto de prueba se presentan mediante matrices de confusión.

8.7.1. Resultados participante 1

El primer participante fue una persona adulta de 30 años de género masculino.

Clasificador	Número de clases	Porcentaje de validación
SVM	2	93.8 %
ANN	2	87.2 %
KNN	2	83.6 %

Cuadro 10: Resultados de porcentaje de validación para las clasificadores seleccionados, participante 1.

Las matrices de confusión de los diferentes modelos de clasificación, se pueden encontrar en las Figuras 84 - 86.

Resultados del modelo SVM con conjunto de prueba

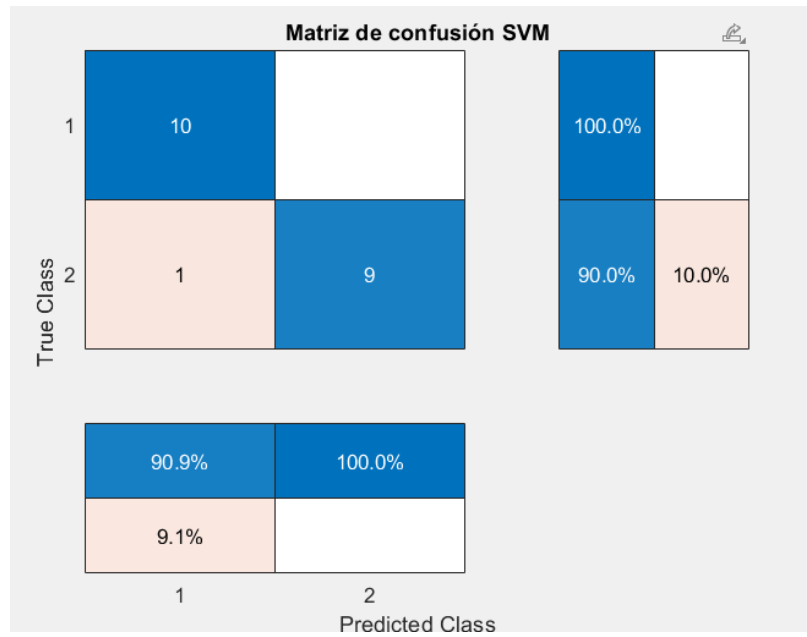


Figura 84: Matriz de confusión de modelo SVM con conjunto de prueba de participante 1.

Resultados del modelo ANN con conjunto de prueba

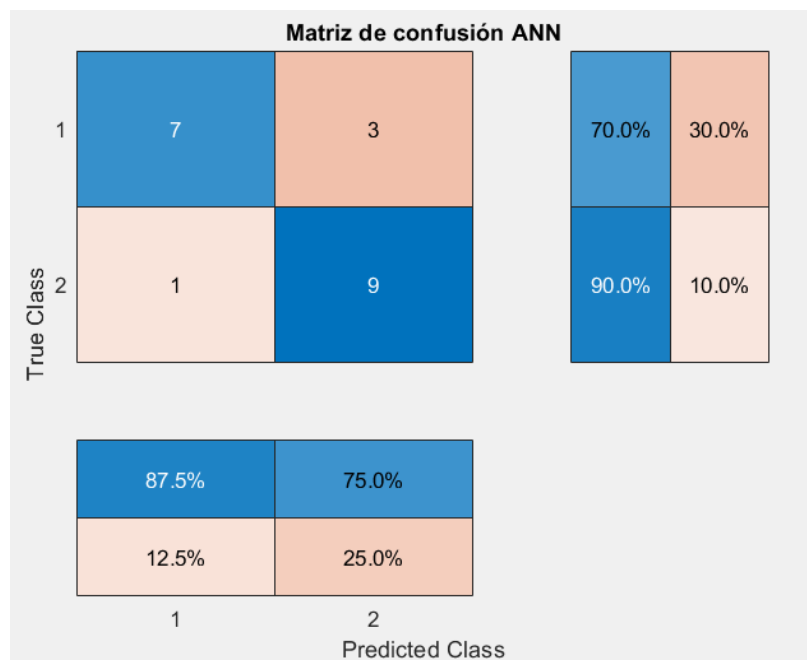


Figura 85: Matriz de confusión de modelo ANN con conjunto de prueba de participante 1.

Resultados del modelo KNN con conjunto de prueba

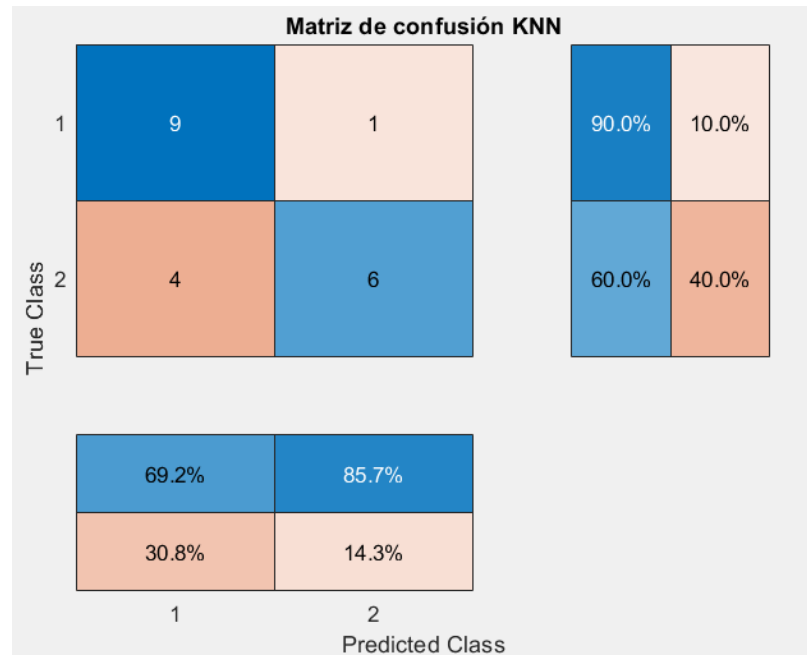


Figura 86: Matriz de confusión de modelo KNN con conjunto de prueba de participante 1.

8.7.2. Resultados participante 2

El primer participante fue una persona adulta de 30 años de género femenino.

Clasificador	Número de clases	Porcentaje de validación
SVM	2	91.5 %
ANN	2	86.4 %
KNN	2	82.0 %

Cuadro 11: Resultados de porcentaje de validación para las clasificadores seleccionados, participante 2.

Las matrices de confusión de los diferentes modelos de clasificación, se pueden encontrar en las Figuras 87 - 89.

Resultados del modelo SVM con conjunto de prueba

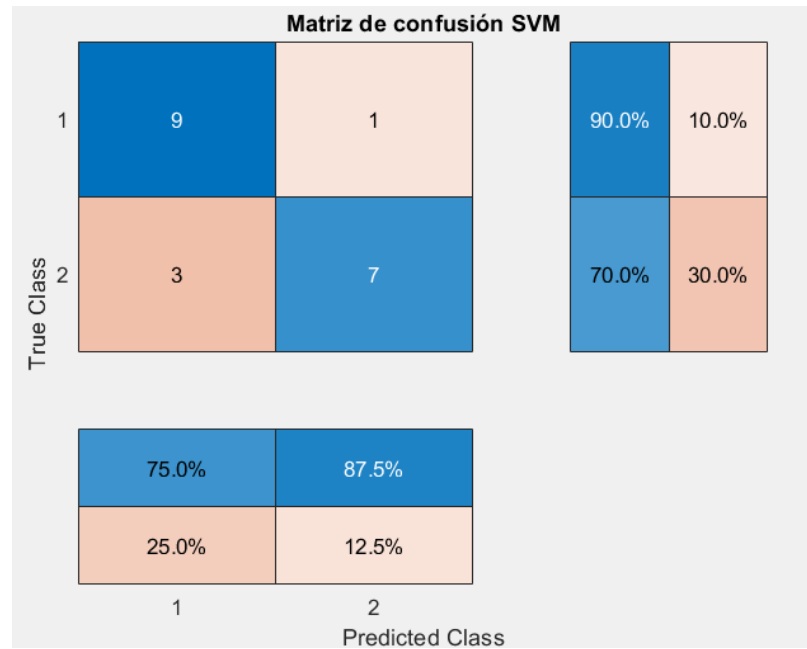


Figura 87: Matriz de confusión de modelo SVM con conjunto de prueba de participante 2.

Resultados del modelo ANN con conjunto de prueba

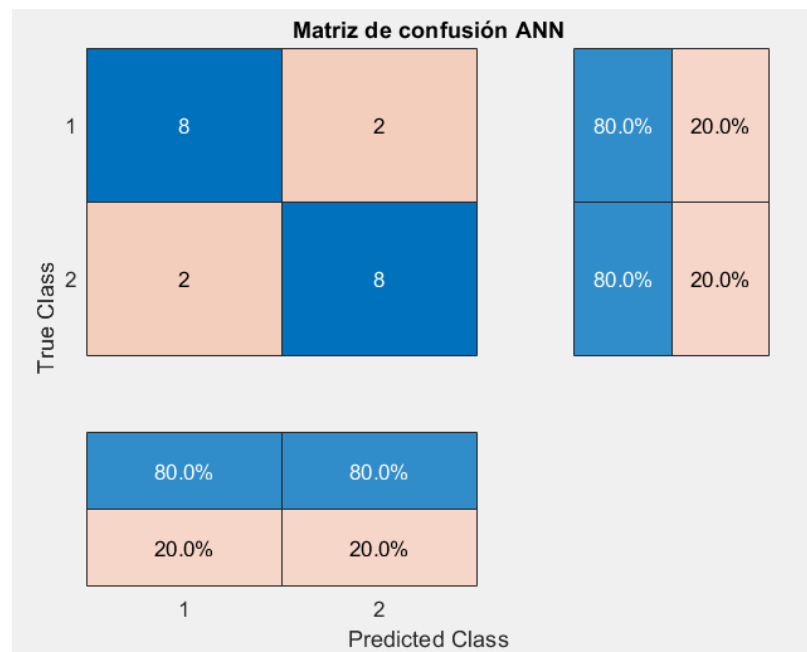


Figura 88: Matriz de confusión de modelo ANN con conjunto de prueba de participante 2.

Resultados del modelo KNN con conjunto de prueba

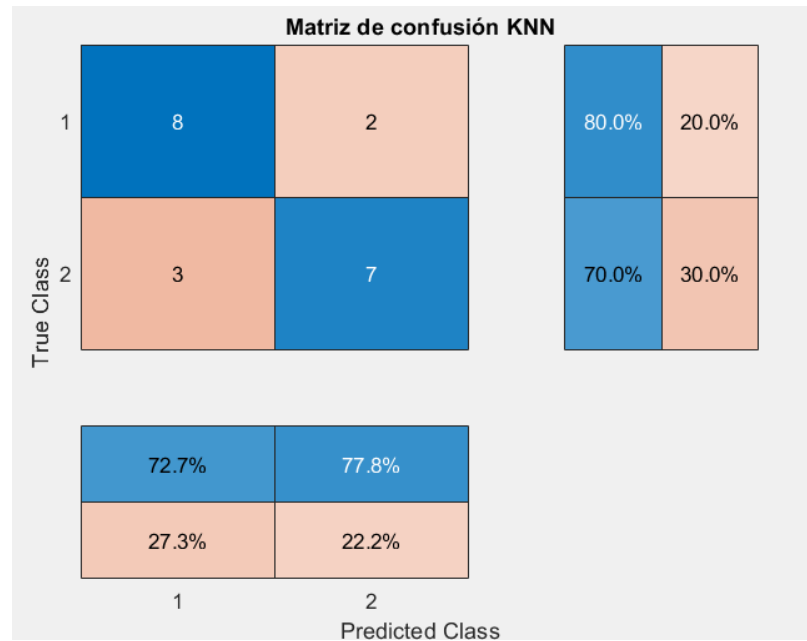


Figura 89: Matriz de confusión de modelo KNN con conjunto de prueba de participante 2.

8.7.3. Resultados participante 3

El primer participante fue una persona adulta de 57 años de género femenino.

Clasificador	Número de clases	Porcentaje de validación
SVM	2	95.2 %
ANN	2	84.5 %
KNN	2	87.3 %

Cuadro 12: Resultados de porcentaje de validación para las clasificadores seleccionados, participante 3.

Las matrices de confusión de los diferentes modelos de clasificación, se pueden encontrar en las Figuras 90 - 92.

Resultados del modelo SVM con conjunto de prueba

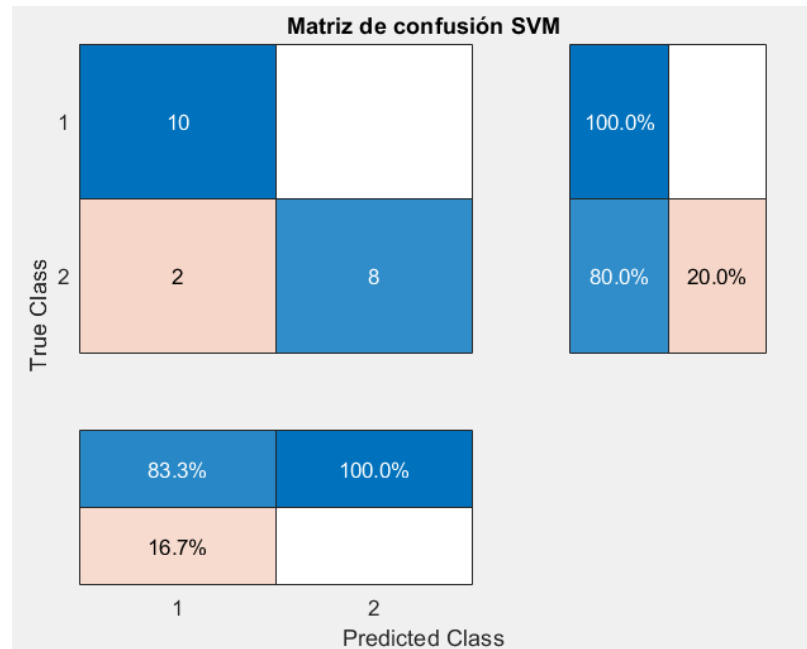


Figura 90: Matriz de confusión de modelo SVM con conjunto de prueba de participante 3.

Resultados del modelo ANN con conjunto de prueba

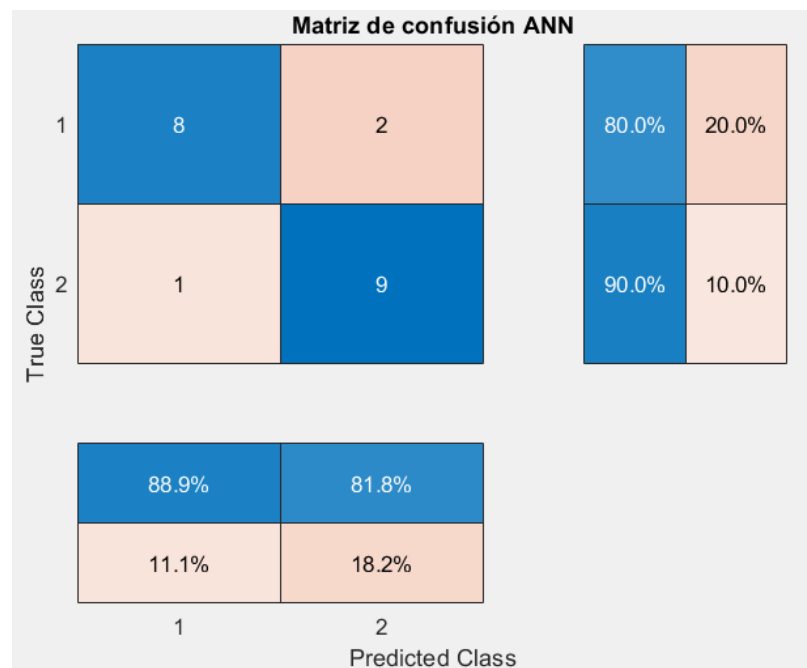


Figura 91: Matriz de confusión de modelo ANN con conjunto de prueba de participante 3.

Resultados del modelo KNN con conjunto de prueba

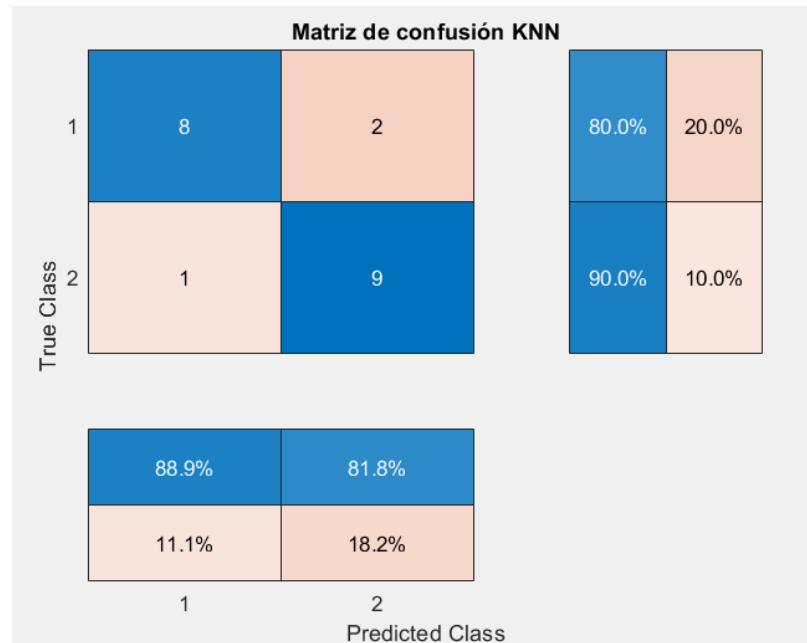


Figura 92: Matriz de confusión de modelo KNN con conjunto de prueba de participante 1.

8.7.4. Resultados participante 4

El primer participante fue una persona adulta de 50 años de género masculino.

Clasificador	Número de clases	Porcentaje de validación
SVM	2	90.0 %
ANN	2	90.0 %
KNN	2	85.0 %

Cuadro 13: Resultados de porcentaje de validación para las clasificadores seleccionados, participante 4.

Las matrices de confusión de los diferentes modelos de clasificación, se pueden encontrar en las Figuras 93 - 95.

Resultados del modelo SVM con conjunto de prueba

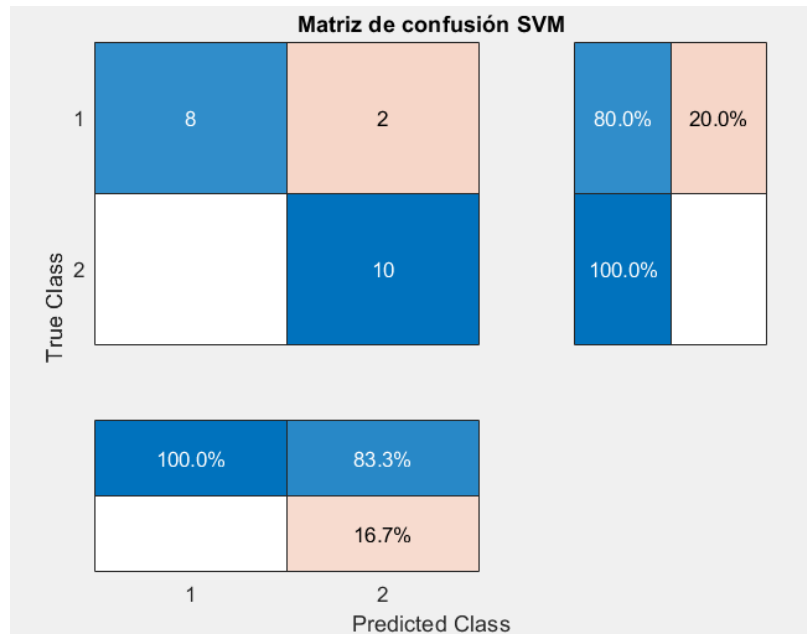


Figura 93: Matriz de confusión de modelo SVM con conjunto de prueba de participante 4.

Resultados del modelo ANN con conjunto de prueba

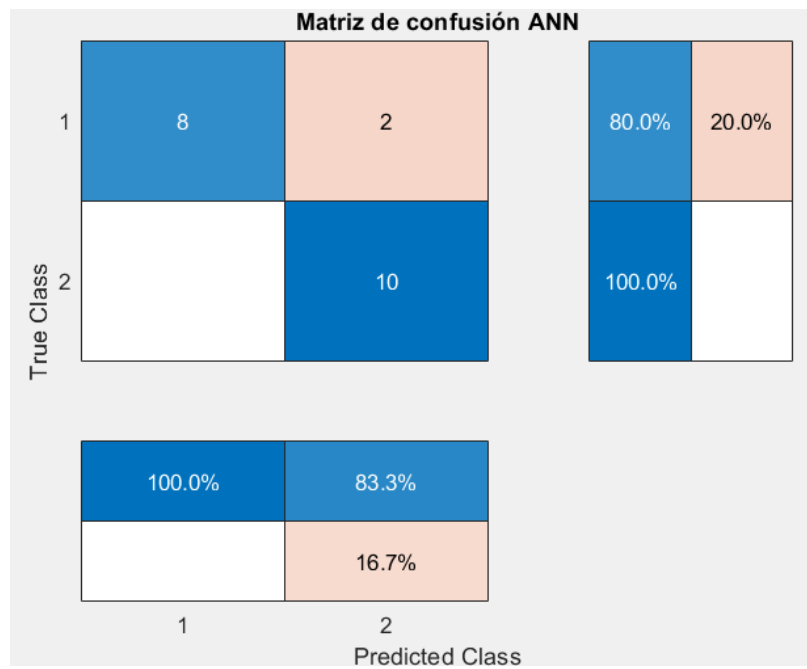


Figura 94: Matriz de confusión de modelo ANN con conjunto de prueba de participante 4.

Resultados del modelo KNN con conjunto de prueba

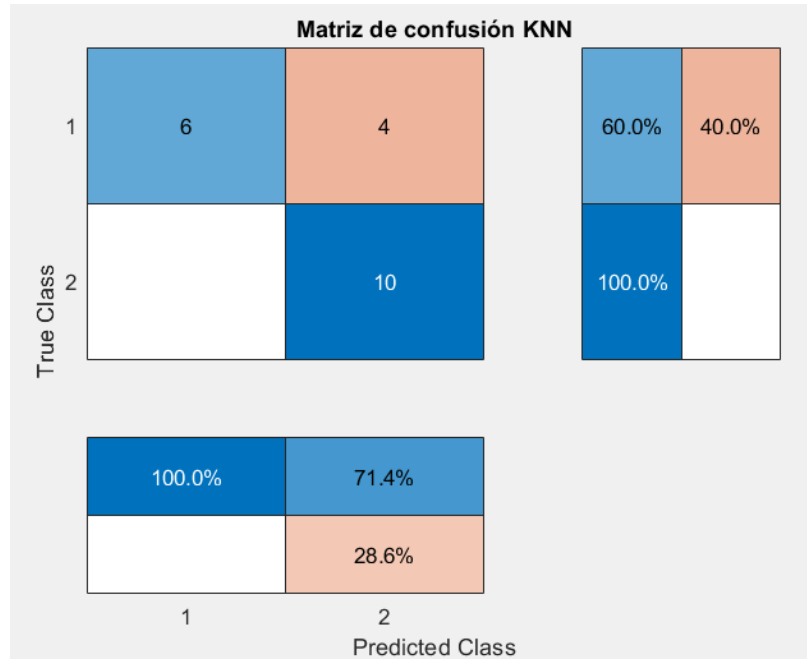


Figura 95: Matriz de confusión de modelo KNN con conjunto de prueba de participante 4.

8.7.5. Análisis de resultados generales

En general, las matrices de confusión correspondientes a los cuatro participantes externos reflejan un desempeño consistente de los modelos de clasificación. En la mayoría de los casos, se observa una alta precisión en la distinción entre los dos gestos, con una cantidad reducida de errores de clasificación.

Aunque existen ligeras variaciones en el rendimiento entre participantes, estas no afectan significativamente la efectividad global del modelo.

No obstante, los resultados obtenidos validan la capacidad del sistema para adaptarse a señales provenientes de diferentes personas, lo cual es un aspecto clave para su aplicación práctica.

Sistema de sensores inerciales

Para implementar un sistema de sensores inerciales, se utilizó una unidad de medición inercial (IMU), para trabajar en conjunto con señales EMG en el control de actuadores. En este capítulo se detallan los algoritmos y el modo de uso empleados para dicho control.

9.1. Características del IMU

Existen diversos sensores inerciales que se consideraron para su implementación en este trabajo. Entre ellos, el MPU6050 destacó como la opción más accesible dentro de los recursos disponibles en la Universidad del Valle de Guatemala. Este módulo IMU (Unidad de Medición Inercial) combina un acelerómetro y un giroscopio en un solo dispositivo, lo que lo hace ideal para aplicaciones relacionadas con el control de movimientos.

El MPU6050 es reconocido no solo por su accesibilidad, sino también por la facilidad de calibración de sus sensores, lo que permite obtener mediciones precisas tanto del acelerómetro como del giroscopio. Para integrarlo en el sistema, se estableció una conexión con un microcontrolador. En este caso, se eligió utilizar el accesorio de comunicación inalámbrica del MyoWare y su módulo ESP32-WROOM, debido a su arquitectura moderna, su capacidad de procesamiento y su compatibilidad con el entorno de desarrollo Arduino IDE, lo que simplifica el desarrollo y la implementación del sistema.

Se implementó la lectura en este microcontrolador con el objetivo de facilitar la implementación del sistema junto con el sensor MyoWare que también trabaja por medio de arquitectura de Arduino.

9.2. Algoritmo de lectura

Al iniciar con los algoritmos para la lectura del MPU6050, es importante destacar que este sensor se comunica mediante el protocolo I2C lo que requiere configurar adecuadamente esta comunicación en el microcontrolador.

En el entorno de Arduino IDE, existen diversas librerías específicas para el módulo ESP32 que facilitan la comunicación con el MPU6050. Sin embargo, en este trabajo se optó por no utilizar dichas librerías. En su lugar, se implementó una comunicación directa utilizando la librería Wire, que es específica para manejar el protocolo I2C.

El algoritmo de lectura se basa en el uso de direcciones asignadas al dispositivo. Para realizar una lectura, es necesario indicar la dirección del registro del MPU6050 donde se encuentra almacenada la información requerida. Esto implica enviar una solicitud a dicha dirección y, posteriormente, recibir los datos correspondientes. Este enfoque permitió un mayor control sobre la comunicación y un entendimiento más profundo del funcionamiento interno del sensor.

9.3. Calibración del sensor

Para calibrar el sensor, es importante destacar que las lecturas del acelerómetro del MPU6050 se miden en términos de gravedades (g). Por esta razón, el sensor se configuró para operar en un rango de $2g$, adecuado para obtener lecturas precisas en este contexto.

El proceso de calibración consistió en mantener el sensor en posición vertical sobre cada uno de los ejes de lectura (X, Y y Z), asegurándose de que permaneciera estacionario. Durante este procedimiento, se registraron los valores medidos por el acelerómetro en cada eje. A partir de estos datos, se realizó una regresión lineal para ajustar las lecturas y compensar cualquier desviación inherente al sensor.

Posteriormente, los parámetros obtenidos de la regresión lineal se aplicaron a las lecturas adquiridas en tiempo real, normalizando los valores entre -1 y 1. Esto permitió obtener una escala consistente y precisa al medir las variaciones en la orientación, especialmente al girar el giroscopio. Este enfoque aseguró una calibración adecuada del sensor para su uso en aplicaciones prácticas.

9.4. Algoritmo de control

Como se mencionó anteriormente, durante la calibración del sensor, las lecturas del acelerómetro se normalizaron a un rango entre -1 y 1. Este rango se utilizó para implementar el control de los actuadores en este proyecto.

El propósito principal del sensor es colocarlo en el antebrazo del usuario, permitiendo simular sus movimientos naturales. Esto facilita la generación de movimiento angular, combinado con el control adicional mediante señales EMG captadas por el MyoWare.

Para alcanzar este objetivo, se realizó un mapeo de los valores del sensor a ángulos predefinidos. Este mapeo se empleó para traducir las lecturas del acelerómetro en comandos de control, creando así un movimiento rotativo preciso en el actuador seleccionado. Este enfoque permitió combinar la detección de movimientos con señales musculares, logrando un control más intuitivo y natural.

En la implementación con una mano antropomórfica, los valores obtenidos del sensor fueron transformados a un rango de 80 a 190. Este ajuste permitió adaptar las lecturas del MPU6050 a los ángulos requeridos para controlar los movimientos de la mano.

En la Figura 96, se puede observar el diagrama de flujo para las lecturas y mapeo del MPU6050.

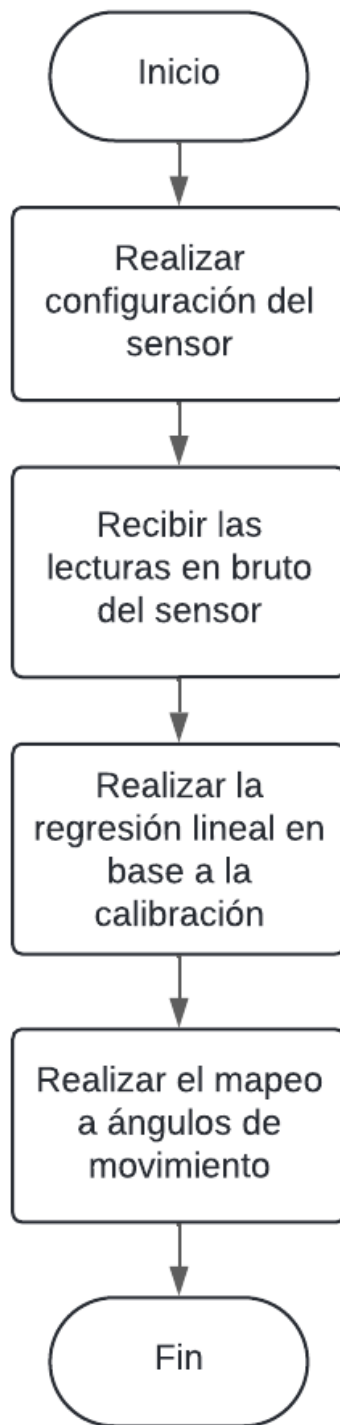


Figura 96: Diagrama de flujo de algoritmo implementado para lectura y mapeo de MPU6050.

Con el fin de presentar el control de actuadores se utilizó una mano antropomórfica desarrollada en la Universidad del Valle de Guatemala [11]. En este capítulo se detalla el uso de la misma y la comunicación entre los diferentes componentes.

10.1. Características de mano antropomórfica

En el caso de la mano antropomórfica UVG, se emplean dos tipos de servomotores para controlar el movimiento de las distintas partes del proyecto. Estos servomotores, de la marca Dynamixel, corresponden a las series AX-12A y XL-320.

Para operar la mano, se utiliza el microcontrolador OpenCM, diseñado específicamente para el control de motores de alta potencia, en particular los servomotores mencionados. El programa para este microcontrolador se basa en el software de Arduino, lo que permite compartir muchas de sus funciones y modos de trabajo, facilitando la programación y control de los motores.

10.2. Algoritmos de control

En investigaciones previas, se desarrolló un programa específico para el control de los servomotores utilizados, diseñado especialmente para integrarse con el software Unity y generar los movimientos deseados en el entorno virtual [12]. El código convierte los ángulos percibidos en Unity a ángulos reconocidos por el microcontrolador, lo que permite reproducir el movimiento exacto en la mano antropomórfica. Además, el programa se comunica con MATLAB, que actúa como intermediario entre la computadora y el controlador.

Para implementar este programa en el contexto de la interfaz biomédica, fue necesario modificar algunas partes del código, optimizando el uso de la mano antropomórfica y eliminando la dependencia de Unity. Aunque la mayor parte del código se mantuvo, se ajustaron los ángulos de entrada para adaptarlos al controlador. A través de pruebas con diferentes ángulos, se identificaron los movimientos adecuados para controlar la mano. En dichas pruebas, se encontró que los dedos operan dentro de un rango de ángulos entre 127° y 190° , por lo que los movimientos se basaron en la apertura y cierre de las distintas partes de la mano.

Dado que ya no es necesaria la integración con Unity, se simplificó el sistema al eliminar la librería y los comandos específicos para la comunicación entre Unity y el dispositivo. Ahora, el control se gestiona únicamente mediante comunicación serial entre MATLAB y el controlador OpenCM. El control se implementa mediante una estructura condicional tipo *switch-case*, donde cada caso corresponde a un conjunto de ángulos de movimiento que se envían al controlador cuando se cumplen las condiciones correspondientes. A través de esta comunicación serial, el controlador ejecuta los movimientos precisos en los servomotores.

El algoritmo de control se ilustra en el diagrama de flujo de la Figura 97.

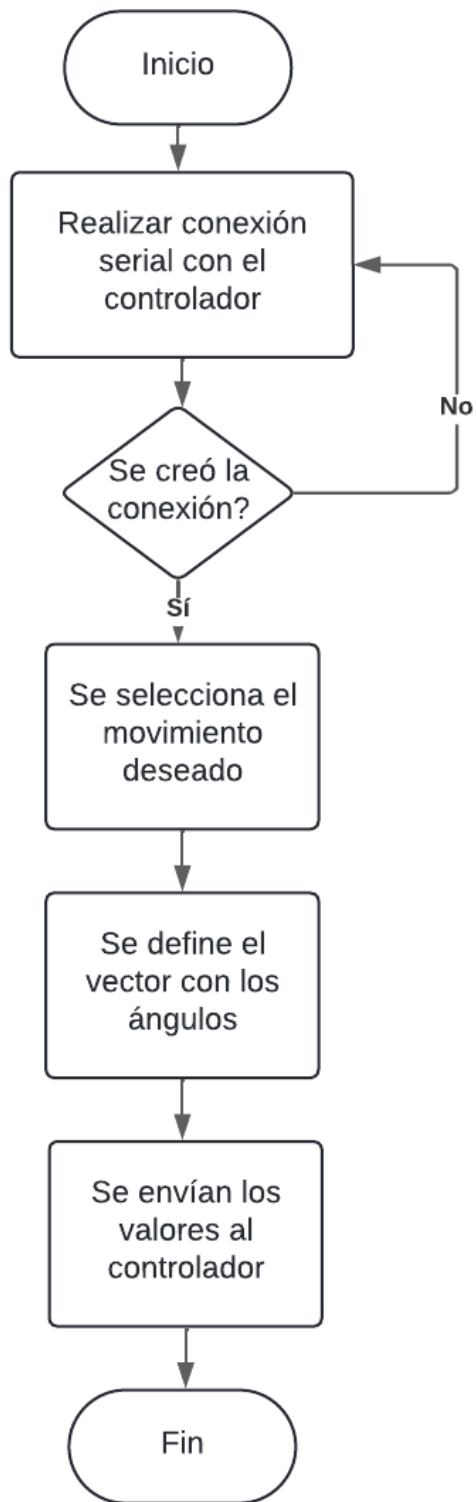


Figura 97: Diagrama de flujo de algoritmo de control de la mano antropomórfica.

10.3. Movimientos predeterminados

Dado que el proyecto requiere la clasificación de diferentes gestos y con el objetivo de recrear movimientos comunes de la mano, se tomó la decisión de implementar el gesto de conteo con los dedos.

Con esto en mente, y utilizando los ángulos de apertura y cierre de los dedos previamente identificados, se definieron variables en MATLAB que contienen los ángulos específicos para cada servomotor de la mano, logrando así el movimiento deseado. Estos movimientos se pueden observar en las Figuras 98 - 103.

Movimiento 0

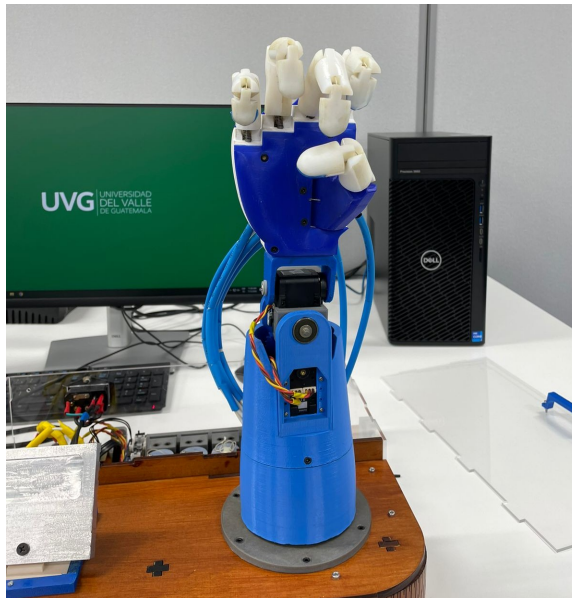


Figura 98: Conteo del número 0 en mano antropomórfica.

Movimiento 1



Figura 99: Conteo del número 1 en mano antropomórfica.

Movimiento 2



Figura 100: Conteo del número 2 en mano antropomórfica.

Movimiento 3



Figura 101: Conteo del número 3 en mano antropomórfica.

Movimiento 4



Figura 102: Conteo del número 4 en mano antropomórfica.

Movimiento 5



Figura 103: conteo del número 5 en mano antropomórfica.

Interfaz de control de mano antropomórfica

Es importante destacar que existen interfaces, realizadas en fases anteriores, capaces de realizar control mediante señales EMG, aunque se basan únicamente en bases de datos preestablecidas. Esto resalta la necesidad de continuar investigando y comprobar su funcionamiento con un enfoque en el control en tiempo real. En este capítulo se presenta la implementación de una interfaz con la capacidad de controlar actuadores en tiempo real, por medio de señales EMG y sensores inerciales.

La interfaz cuenta con tres pestañas de configuración para lograr el control del actuador deseado, estas siendo: Instrucciones, Entrenamiento y Panel de control.

En la pestaña de Instrucciones se encuentran los pasos a seguir para la configuración de los modos deseados a utilizar dentro de la interfaz. Esta pestaña se puede observar en la Figura 104.

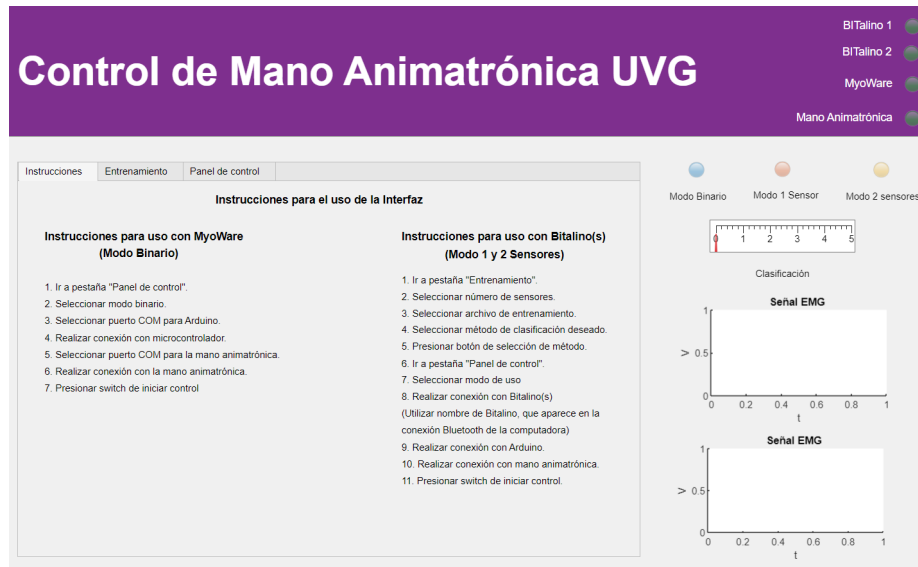


Figura 104: Interfaz de control de mano antropomórfica UVG, pestaña de instrucciones.

11.1. Implementación de clasificadores

Debido a la variedad de clasificadores disponibles en la aplicación de *Classification Learner*, se decidió ofrecer al usuario diferentes opciones. En la interfaz se implementaron los algoritmos de clasificación más efectivos observados en el Capítulo 8, seleccionados principalmente por sus porcentajes de validación en la etapa de prueba.

Si se habilitan las tres opciones, la interfaz selecciona automáticamente el clasificador con el mejor porcentaje de rendimiento durante el entrenamiento, aunque el usuario también tiene la posibilidad de elegir el clasificador de su preferencia. Esta implementación se puede apreciar en la Figura 105, en el cuadro rojo.

Además, la aplicación permite cargar una base de datos personalizada para el entrenamiento. Al ingresar un archivo '.mat', la interfaz selecciona automáticamente una variable llamada 'BaseEntrenamiento', que se utiliza para entrenar los tres clasificadores. Esta implementación se puede apreciar en la Figura 105, en el cuadro verde.

Debido a las funcionalidades de la interfaz, se agregó un selector de cantidad de sensores a utilizar. Esto se implementó gracias a la necesidad de tener diferentes modelos de clasificación para cada modo de uso. Estos se pueden observar en la Figura 105, en el cuadro azul.

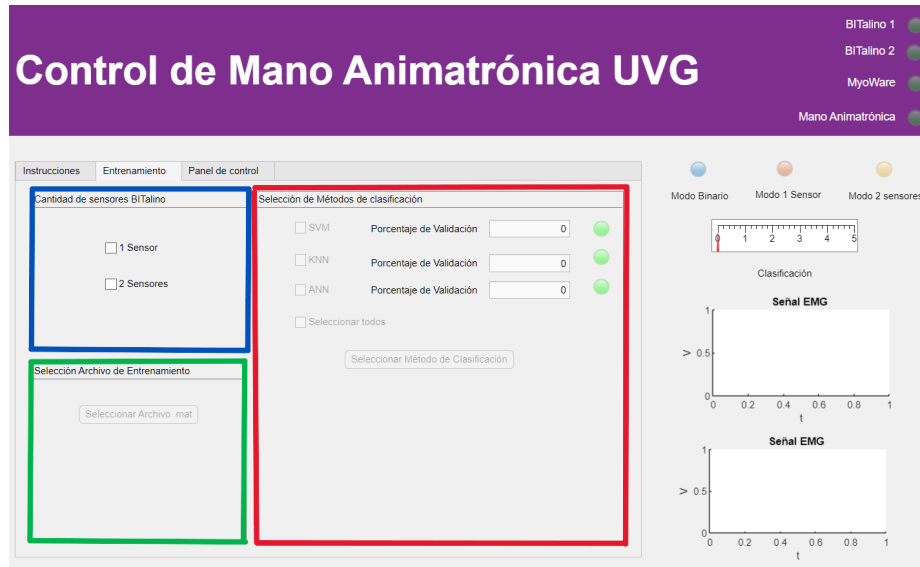


Figura 105: Interfaz de control de mano antropomórfica UVG, pestaña de entrenamiento.

11.2. Funcionalidades

Con la disponibilidad de diferentes sensores, se implementaron tres funcionalidades en la interfaz: modo binario, modo de 1 sensor y modo de 2 sensores. Estas funcionalidades permiten utilizar diversos tipos de sensores y conectarlos a la interfaz de manera efectiva.

La selección de los modos se puede observar en la Figura 106, en el cuadro rojo. De la misma forma se pueden observar las diferentes conexiones a realizar en el cuadro verde.

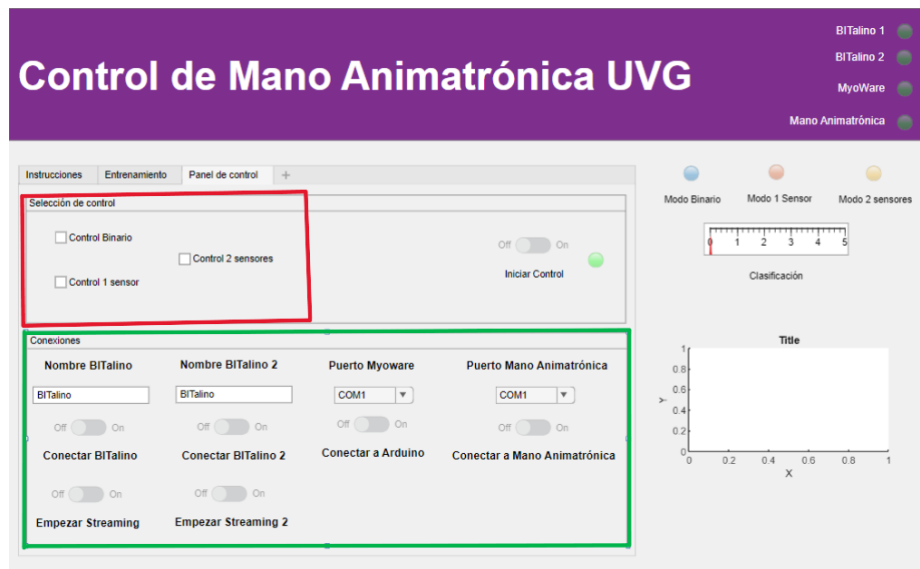


Figura 106: Interfaz de control de mano antropomórfica UVG, pestaña de control.

11.2.1. Modo binario

Esta funcionalidad de la interfaz opera exclusivamente con el sensor MyoWare, utilizando la funcionalidad de *Envelope* descrita en el Capítulo 7.

Una vez establecida la comunicación entre el microcontrolador y MATLAB, se realizan las lecturas correspondientes desde la entrada analógica del microcontrolador. Al recibir la señal, esta se convierte en ángulos de movimiento de la mano, basándose en el nivel de intensidad detectado.

Este proceso permite traducir la intensidad de la señal generada por los músculos en ángulos de operación para controlar el movimiento de la mano antropomórfica.

Además, al estar conectado con el MyoWare, el sistema también recibe las lecturas del MPU6050. Estas lecturas se utilizan para determinar los ángulos de movimiento relacionados con la base rotatoria de la mano, proporcionando una integración eficiente entre ambos dispositivos para el control preciso de los actuadores. En la Figura se puede observar la posición y forma de conexión de los sensores para este modo.

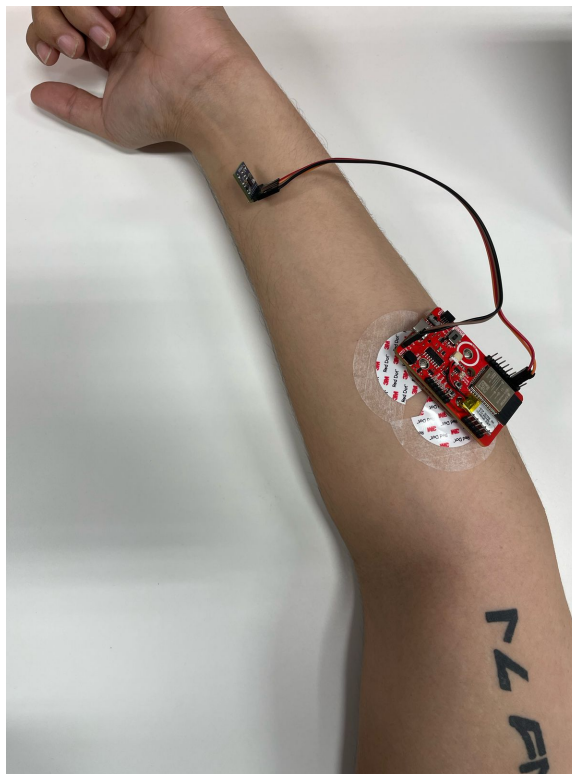


Figura 107: Posición y conexión de sensores para modo Binario.

En la Figura 108, se puede observar la interfaz en la funcionalidad de modo binario mientras esta en uso.

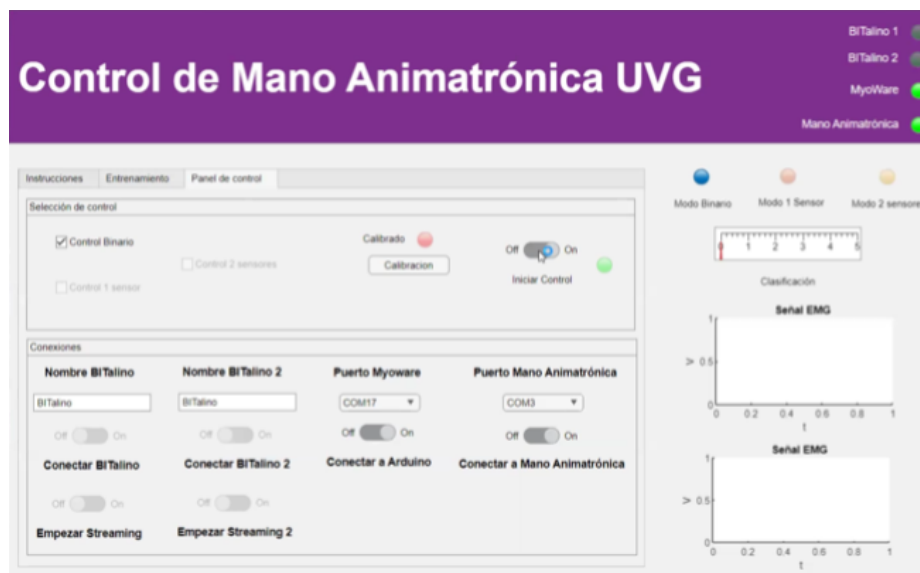


Figura 108: Interfaz de control de mano antropomórfica UVG, en funcionalidad de modo binario.

11.2.2. Modo 1 sensor

Este modo agrega el uso del sensor BITalino para capturar señales EMG, por medio del algoritmo descrito en el Capítulo 7, en la Sección 7.3.2. Las señales adquiridas son clasificadas mediante los algoritmos implementados en la interfaz. Posteriormente, se selecciona una posición predeterminada en el algoritmo, que luego se envía a la mano.

Este modo se diseñó con el objetivo de distinguir entre tres clases de movimientos, aunque también es funcional con un número menor o mayor de clases.

De igual forma que el modo binario, el modo de 1 sensor toma en cuenta las lecturas del MPU6050, para implementar el mismo movimiento rotativo.

Para este modo es necesaria la conexión con el microcontrolador de preferencia, la mano antropomórfica y el sensor BITalino.

En las Figuras 109 y 110, se puede observar el funcionamiento de la interfaz en este modo, desplegando las señales adquiridas como la etiqueta de clasificación.

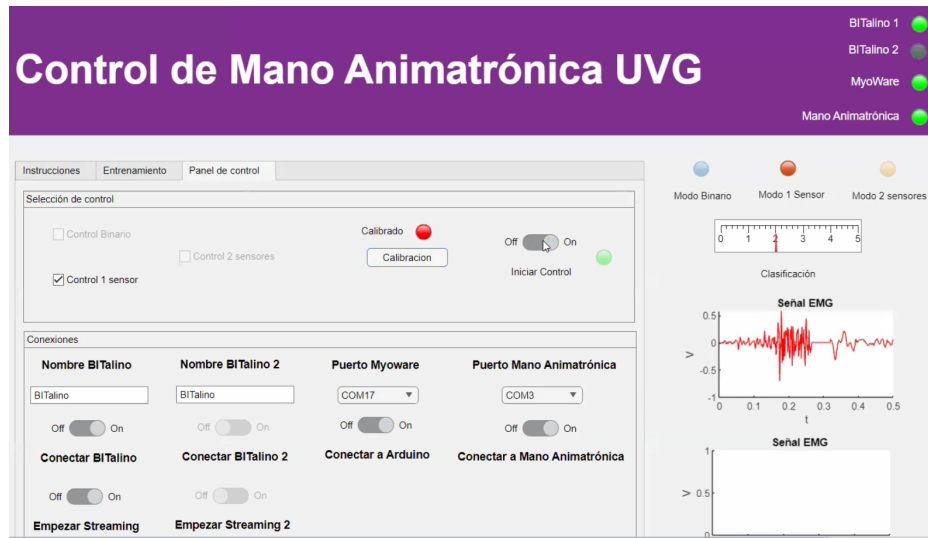


Figura 109: Interfaz de control de mano antropomórfica UVG, clasificando señal clase 2 en modo de un sensor.

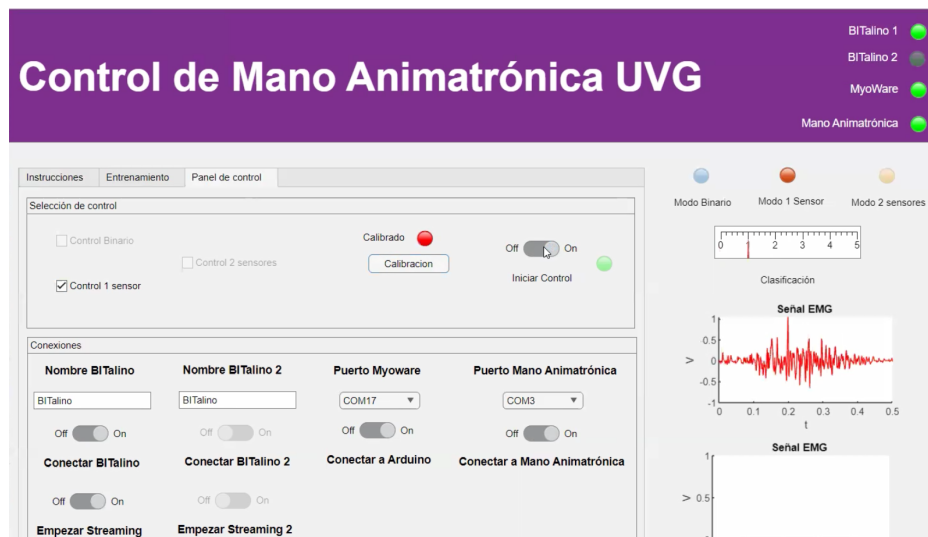


Figura 110: Interfaz de control de mano antropomórfica UVG, clasificando señal clase 1 en modo de un sensor.

11.2.3. Modo 2 sensores

Este modo es similar al modo de 1 sensor, con la única diferencia de que admite un segundo sensor BITalino. Esta funcionalidad se implementó con el objetivo de clasificar entre 4 y 6 clases de movimiento, aprovechando la versatilidad que ofrece la incorporación de dos señales extraídas, lo que permite manejar un mayor número de variables.

En este modo, se utilizan las mismas conexiones mencionadas anteriormente, con la diferencia de que se requiere una conexión adicional al segundo BITalino.

En las Figuras 111 - 113, se puede observar el funcionamiento de la interfaz en este modo, desplegando las señales adquiridas como la etiqueta de clasificación.

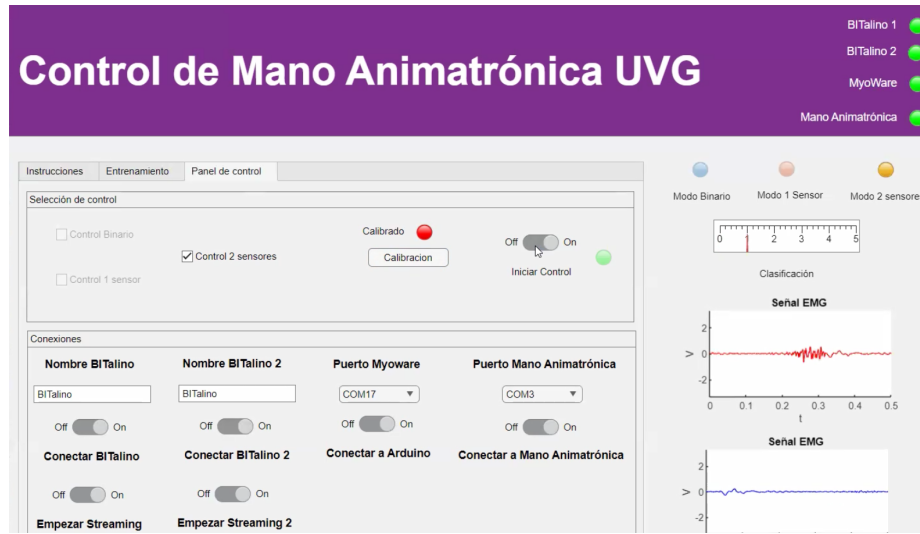


Figura 111: Interfaz de control de mano antropomórfica UVG, clasificando señal clase 1 en modo de dos sensor.

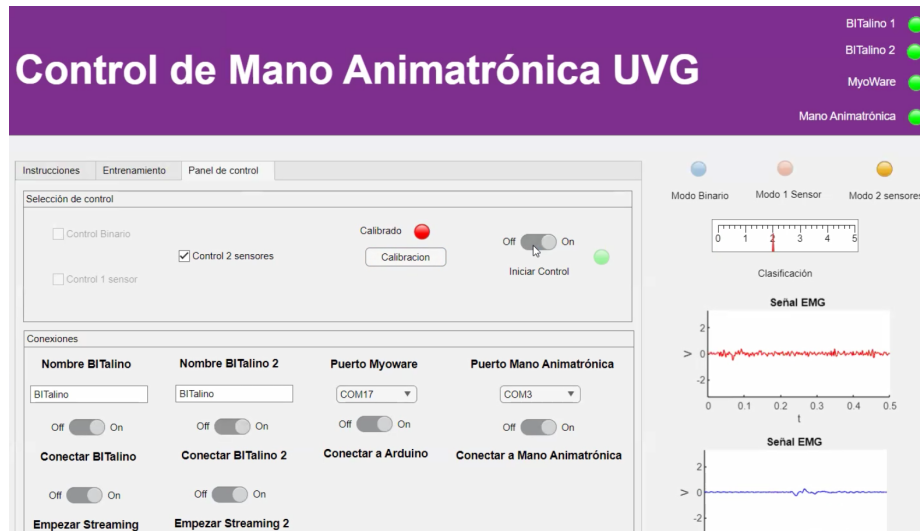


Figura 112: Interfaz de control de mano antropomórfica UVG, clasificando señal clase 2 en modo de dos sensor.

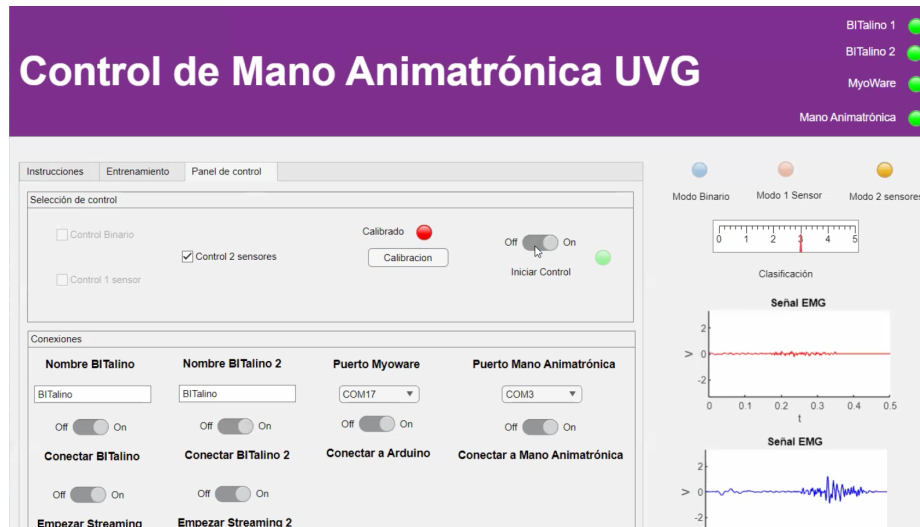


Figura 113: Interfaz de control de mano antropomórfica UVG, clasificando señal clase 3 en modo de dos sensor.

- Los algoritmos de adquisición desarrollados lograron capturar señales EMG de forma precisa y eficiente en tiempo real, garantizando una lectura continua y fiable, esencial para aplicaciones de control robusto.
- El algoritmo de detección automática demostró ser eficaz y consistente al identificar las señales, asegurando una detección precisa y continua a lo largo del proceso de recolección de las señales. Esto permitió una respuesta rápida y fiable del sistema ante la actividad y detección de diferentes gestos.
- Se comprobó que la segmentación de las señales EMG ofrece importantes beneficios durante el proceso de clasificación, permitiendo una mejor organización de los datos y mejorando la precisión de los resultados.
- Los clasificadores implementados demostraron ser altamente eficientes para la clasificación de las señales EMG, alcanzando resultados satisfactorios en cuanto a velocidad y precisión, lo que permite un control efectivo de los actuadores.
- Al seleccionar las características adecuadas de las señales EMG, es posible reducir la carga computacional sin sacrificar la precisión del sistema de clasificación, manteniendo un alto porcentaje de acierto.
- La integración de un sistema de sensores inerciales con un sistema de clasificación de señales EMG proporciona una gran versatilidad en el control de actuadores, ampliando las posibilidades de interacción y movimiento de los dispositivos.
- La comunicación y el control de la mano antropomórfica fueron altamente efectivos, evidenciando la versatilidad de la interfaz para gestionar el funcionamiento de los actuadores. Este sistema permitió una interacción fluida y precisa, adaptándose bien a los diferentes movimientos y señales del usuario.

- En este trabajo se logró implementar el dispositivo MyoWare. Sin embargo, este sensor ofrece otras funcionalidades que no fueron utilizadas en esta ocasión. Se recomienda explorar dichas funciones para mejorar la interfaz, por ejemplo, optimizando la detección de señales musculares o ampliando las aplicaciones del sistema.
- Se recomienda la implementación de más dispositivos para mejorar la clasificación y el control, acercando la interacción al movimiento natural de una prótesis. La incorporación de sensores adicionales, como más unidades de EMG, podría proporcionar una mayor precisión y realismo en el control de los actuadores, mejorando la experiencia del usuario y la funcionalidad de la prótesis.
- Se recomienda la implementación de otros sensores inerciales para lograr un control más preciso y versátil de los actuadores. La integración de sensores adicionales, como giroscopios o acelerómetros de mayor precisión, podría mejorar la detección de movimientos y proporcionar un control más fluido y dinámico, adaptándose mejor a las necesidades del usuario y ampliando las capacidades del sistema.
- Los modelos de clasificación fueron eficaces al momento de diferenciar las señales, pero se recomienda experimentar con diferentes modelos además de los utilizados. En particular, modelos no incluidos dentro de la librería de *Classification Learner* como *Random Forest* y *Gradient Boosting* pueden ser útiles para capturar relaciones complejas entre las características,
- Se recomienda realizar pruebas con distintos tipos de actuadores o prótesis, así como explorar una mayor variedad en el posicionamiento de los electrodos. Por ejemplo, se pueden considerar ubicaciones más simples, como en los músculos de la pierna, o áreas más delicadas, como los músculos de la mano. Esto permitirá aprovechar al máximo la versatilidad de la interfaz.

- [1] J. F. Delgado, E. Vallejo y J. Torres, “Diseño y Construcción de un Sistema de adquisición y Visualización de Señales Electromiográficas,” *Universidad del Norte*, 2007.
- [2] E. Muñoz, O. H. Paruma y J. F. Florez, “Aplicaciones de las Señales Mioelectricas para el Control de Interfaces Hombre-Maquina,” *Universidad del Cauca*, 2004.
- [3] D. Capdevila, E. Spinelli y A. Veiga, “Plataforma Vestible para Interfaces Hombre-Máquina,” *Universidad Nacional de La Plata*, 2019.
- [4] M. F. Girón, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EMG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [5] R. Ralda, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EEG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [6] R. J. Cáceres, “Interfaz para el Control de Sistemas Robóticos Utilizando Combinaciones de Señales Biomédicas,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [7] F. J. Sandoval, “Aplicación sistemática de algoritmos de aprendizaje automático y reconocimiento de patrones a señales bioeléctricas obtenidas de HUMANA y con el sistema Biopac,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2023.
- [8] H. A. Romo, J. C. Realpe y P. E. Jojoa, “Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano,” *Avances en Sistemas e Informática*, 2007.
- [9] C. Alquézar, “Proyecto Roujin: Entorno experimental para pruebas de control con señales bioeléctricas,” Tesis de licenciatura, Universidad de Zaragoza, 2015.
- [10] G. Palarea, “Diseño y Fabricación de una Prótesis Electromecánica de Mano Humana, Controlada por Señales EMG de Superficie,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [11] O. Gálvez, “Optimización de diseño de una mano y muñeca animatrónica antropomórfica de la fase tres e implementación de un control interactivo.,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.

- [12] M. García, “Optimización del proyecto de una mano y muñeca animatrónica antropomórfica y su sistema de control y operación,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
- [13] J. D. Bronzino, *The Biomedical Engineering Handbook*. CRC Press, 2000.
- [14] L. Sörnmo y P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Science, 2005.
- [15] P. A. Parker y R. Merletti, *Electromyography Physiology, Engineering, and Non-Invasive Applications*. Wiley, 2004.
- [16] E. Guzmán-Muñoz y G. Mendéz-Rebolledo, “Electromiografía en las Ciencias de la Rehabilitación,” *Revista Salud Uninorte*, 2018.
- [17] R. Bagherian, M. Esmaceli y M. Nourani, “EMG-Based Feature Extraction and Classification for Prosthetic Hand Control,” 2021.
- [18] PLUX Biosignals, *BITalino: Learning and exploring biosignals everywhere*, <https://www.pluxbiosignals.com/pages/bitalino>, Accessed: 2024-05-11, 2024.
- [19] MyoWare, *MyoWare 2.0 Muscle Sensor*, <https://myoware.com/products/muscle-sensor/>, Accessed: 2024-05-11, 2024.
- [20] J. Adamssen, *Inteligencia Artificial: Cómo el aprendizaje automático, la robótica y la automatización han dado forma a nuestra sociedad*. Efalón Acies, 2020.
- [21] J. Muñoz, *Inteligencia Computacional Inspirada en la Vida*. SPICUM - Universidad de Málaga, 2010.
- [22] C. Pineda, *Aprendizaje automático y profundo en python*. Ediciones de la U, 2022.
- [23] E. Sorias, M. Sánchez-Montañes, R. G. anf Borja Castillo y P. Cano, *Sistemas de Aprendizaje Automático*. Ra-Ma S.A. Editorial y Publicaciones, 2023.
- [24] C. Chique, *Maquina de Soporte Vectorial*, <https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac>, Accessed: 2024-05-12, 2020.
- [25] L. Wang, *Support Vector Machines: Theory and Applications*. Springer, 2005.
- [26] H. Rhys, *Machine Learning with R, the Tidyverse, and Mlr*. Manning, 2020.
- [27] A. Band, *Multi class Classification One vs All One-vs-One*, <https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>, Accessed: 2024-07-27, 2020.
- [28] F. Sabry, *K Nearest Neighbor Algorithm: Fundamentals and Applications*. One Billion Knowledgeable, 2023.
- [29] Y. Akbulut, A. Sengur, Y. Guo y F. Smarandache, “NS-k-NN: Neutrosophic Set-Based k-Nearest Neighbors Classifier,” 2023.
- [30] G. for Geeks, *K-Nearest Neighbor Algorithm*, <https://www.geeksforgeeks.org/k-nearest-neighbours/>, Accessed: 2024-09-11, 2024.
- [31] C. Ltd., *Mastering HMI*. Cybellium Ltd., 2023.
- [32] L. Corona, G. Abarca y J. Mares, *Sensores y Actuadores*. Grupo Editorial Patria, 2014.
- [33] MedlinePlus, *Prothesis*, <https://medlineplus.gov/ency/article/002286.htm>, Accessed: 2024-05-12, 2023.

[34] J.-M. Seeber, *Reconstruyendo el cuerpo*. Teacher Created Materials, 2022.

15.1. Anexo A: Manual de usuario

UNIVERSIDAD DEL VALLE DE GUATEMALA



Figura 1: Selección de características.

Manual de Usuario de Interfaz biomédica EMG de control de actuadores

Santiago Rivera, Estudiante
Contacto: +502 5105-2586
Correo: riv20269@uvg.edu.gt

Fecha: 17 de febrero de 2026

Índice

1. Introducción	2
1.1. Sobre la interfaz	2
2. Requisitos del sistema	2
2.1. BITalino	2
2.1.1. BITalino Toolbox	2
2.1.2. Conexión con BITalino	2
2.2. MyoWare 2.0	3
2.3. Sensor inercial	3
2.4. Módulo inalámbrico	3
3. Interfaz de control de actuadores	3
3.1. Módulo de recolección	4
3.2. Módulo de control	8
3.2.1. Modo binario	9
3.2.2. Modo 1 sensor y Modo 2 sensores	10

1. Introducción

En este manual de usuario, se encontrará detalladamente el uso y operación de la interfaz EMG para el control de actuadores y su módulo de recolección de señales diseñadas dentro de MATLAB. Este documento está dirigido a estudiantes e investigadores que quieran hacer uso de las interfaces, ya sea para la recolección de señales o bien hacer uso de las mismas para el control de actuadores.

1.1. Sobre la interfaz

Esta interfaz cuenta con dos módulos. El módulo principal tiene el enfoque de utilizar aprendizaje automático, señales EMG y sensores inerciales para el control de actuadores. En específico el control de la mano animatrónica hecha en la Universidad del Valle de Guatemala.

El segundo módulo es un módulo de recolección de señales EMG, que trabaja con un algoritmo de detección automática, basada en un umbral en la amplitud de la señal. Además, esta interfaz extrae características en el dominio del tiempo, segmentando la señal en tres partes y extrayendo las mismas de las tres partes.

2. Requisitos del sistema

2.1. BITalino

La interfaz hace uso de un sensor BITalino. Este es un dispositivo muy poderoso basado en arduino. Básicamente es una caja de herramientas y sensores fisiológicos para el desarrollo de software y hardware. Este incluye comunicación bluetooth. El software integrado con el que cuenta el microcontrolador está basado en bajo el lenguaje PHP y Python.

2.1.1. BITalino Toolbox

BITalino Toolbox es básicamente un firmware del software MATLAB para el dispositivo Bitalino. Este permite al usuario conectarse a los dispositivos bitalinos desde MATLAB, así como adquirir y analizar señales. Fue creado con la versión 2017a del software MATLAB y es compatible con cualquier versión después de esta. Al tener una versión de MATLAB compatible con el toolbox se procede a instalar el toolbox desde el software. El primer paso es acceder a la ventana Get Add-Ons en MATLAB, luego en la ventana de búsqueda se coloca el nombre del toolbox y se procede a instalar este.

2.1.2. Conexión con BITalino

Una vez instalado el toolbox, se procede a emparejar el dispositivo BITalino con la computadora, la cual debe contar con conexión Bluetooth. Al detectar el dispositivo, se deben tener en cuenta dos aspectos importantes:

- Se solicitará un PIN para el emparejamiento. Este es “1234”.
- Este emparejamiento solo se realiza una vez; las siguientes conexiones se harán directamente desde el software MATLAB.

El nombre con el que aparece el dispositivo BITalino durante el primer emparejamiento será el que deberá utilizarse en MATLAB para establecer la conexión.

2.2. MyoWare 2.0

MyoWare 2.0 es un sensor creado por la empresa Advancer Technologies, con el objetivo de captar y visualizar señales electromiográficas por medio del uso de tres electrodos. El sensor tiene una variedad de accesorios para la mejora en la captación de señales, el sensor está diseñado para ser trabajado con un microcontrolador Arduino con la capacidad no solo de captar las señales, sino que amplificarlas y rectificarlas para facilitar el análisis y uso de las señales.

Para utilizar el dispositivo MyoWare, fue necesario integrar un microcontrolador ESP32 para establecer la comunicación entre el sensor y MATLAB mediante comunicación serial. Entre los accesorios disponibles, el accesorio de comunicación inalámbrica fue el más utilizado. Este microcontrolador funciona a través del módulo ESP32-WROOM.

2.3. Sensor inercial

En el trabajo se seleccionó un sensor MPU6050, para realizar las lecturas inerciales para el control de los actuadores.

El MPU6050 es reconocido no solo por su accesibilidad, sino también por la facilidad de calibración de sus sensores, lo que permite obtener mediciones precisas tanto del acelerómetro como del giroscopio. Para integrarlo en el sistema, se estableció una conexión con un microcontrolador. En este caso, se eligió utilizar el accesorio de comunicación inalámbrica del MyoWare y su módulo ESP32-WROOM.

Para la conexión del mismo hacemos uso de 3 *jumpers* para la conexión entre los pines SDI, SCL y SCK del MPU6050 y los pines 30, 31 y 32 del módulo inalámbrico del MyoWare.

2.4. Módulo inalámbrico

Dentro de la comunicación entre la computadora y los diferentes sensores se hizo uso de un microcontrolador ESP32. Es importante notar que no es necesario realizar cambios a la programación del mismo. El sensor solo tiene las conexiones del MPU6050 y la conexión inalámbrica se hace automáticamente.

3. Interfaz de control de actuadores

Para utilizar esta interfaz, es necesario contar con el software *MATLAB* desde la versión 2017a, ya que fue para esta versión que se diseñó el *BITalino Toolbox*. Asimismo, es indispensable tener instalados el siguiente *toolbox*:

- **Signal Processing Toolbox**: utilizado para diseñar filtros que permitan procesar las señales.

3.1. Módulo de recolección

Este módulo está diseñado para extraer señales y segmentarlas en tres partes para extraer características en el dominio del tiempo de las mismas

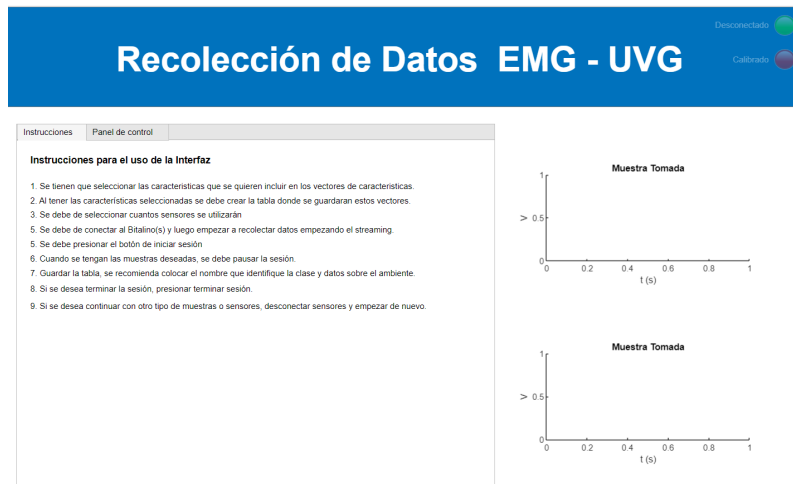


Figura 2: Pantalla inicial de módulo de recolección.

Para iniciar se debe dar click en la pestaña de “Panel de Control”, como se observa en la Figura 3.

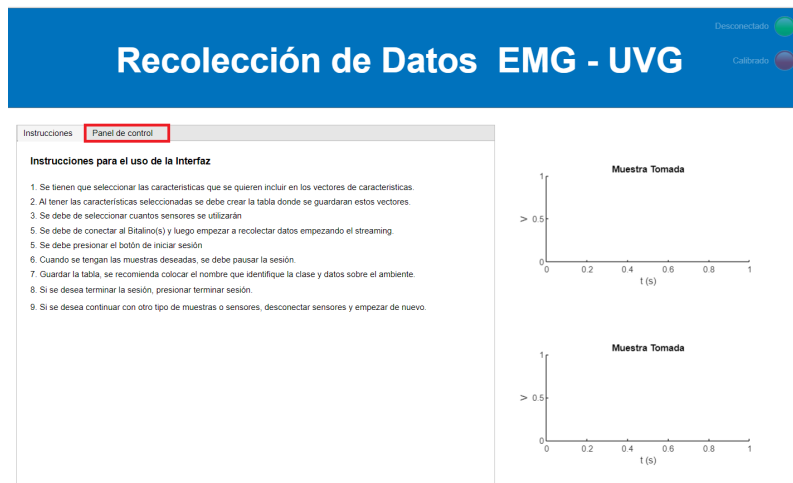


Figura 3: Pestaña de Panel de Control.

Para continuar se deben de seleccionar las características que se desean extraer y el número de sensores que se quiere utilizar. Al realizar esto se debe presionar el botón de crear tabla. Esto se observa en la Figura 4.

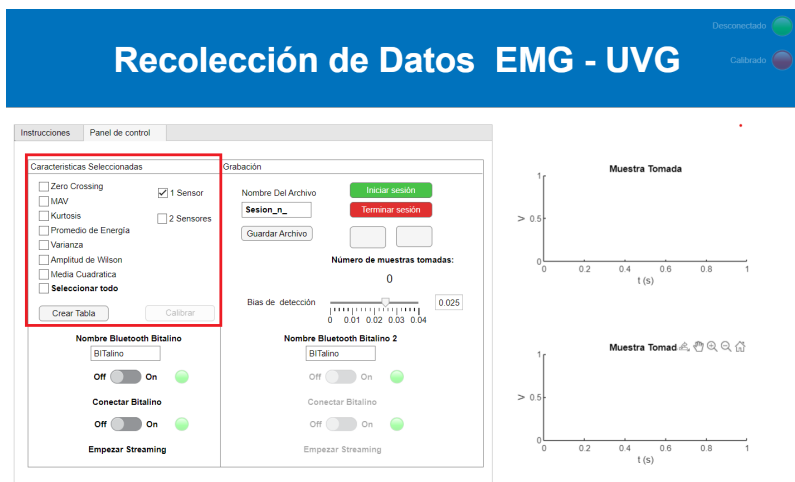


Figura 4: Selección de características.

Al haber realizado la creación de tabla se debe realizar la conexión con el/los BITalino(s). Para esto se debe escribir el nombre del dispositivo que se observó al realizar la conexión bluetooth con la computadora y seleccionar el switch de conectar a BITalino.

Cuando la conexión se complete, el LED a la izquierda del switch se prenderá. “Calibrar” y esperar 5 segundos. Al terminar se prenderá el LED rojo en la esquina superior derecha, significando que el módulo está calibrado.

Consecuentemente se deberá de presionar el switch de “Comenzar Streaming” y de igual forma al completar esto se prenderá el LED a la par del switch. Estos switches y LEDs se pueden observar en la Figura 5.

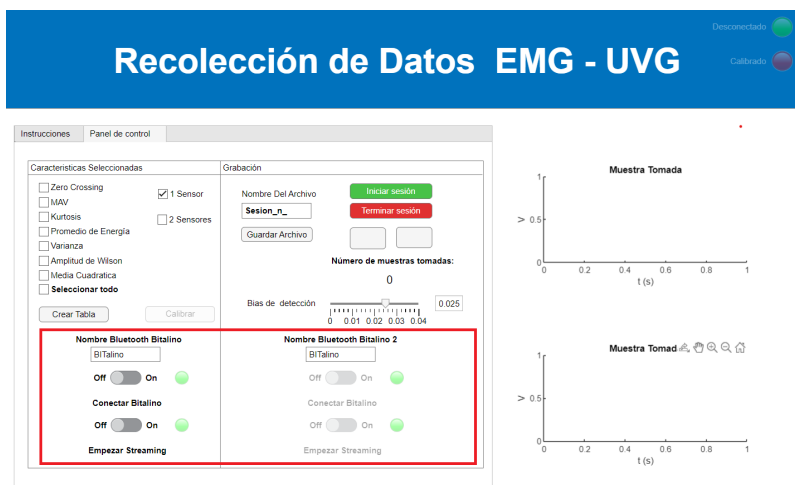


Figura 5: Panel de conexión de BITalino.

Al momento de que el/los sensor(es) comiencen a transmitir datos se debe de calibrar el algoritmo, seleccionando el botón de “Calibrar” y esperar 5 segundos. Al terminar se prenderá el LED rojo en la esquina superior derecha, significando que el módulo está calibrado. Este botón se encuentra en el cuadro en la Figura 6.

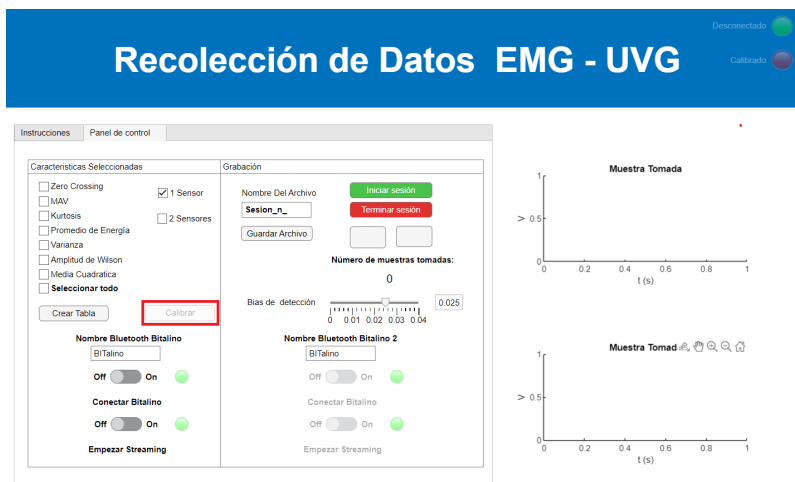


Figura 6: Botón de calibración.

Al tener el módulo calibrado, el módulo ya puede empezar a recolectar señales. Para empezar se debe de presionar el botón de “Iniciar sesión”. En el caso de que se quisiera pausar la recolección se debe de presionar el botón izquierdo, cualquier caso se quiera reanudar, se debe de presionar el botón derecho. Para terminar la sesión se debe de pausar la recolección y presionar el botón de “Terminar Sesión”. Estos se pueden encontrar en la Figura 7.

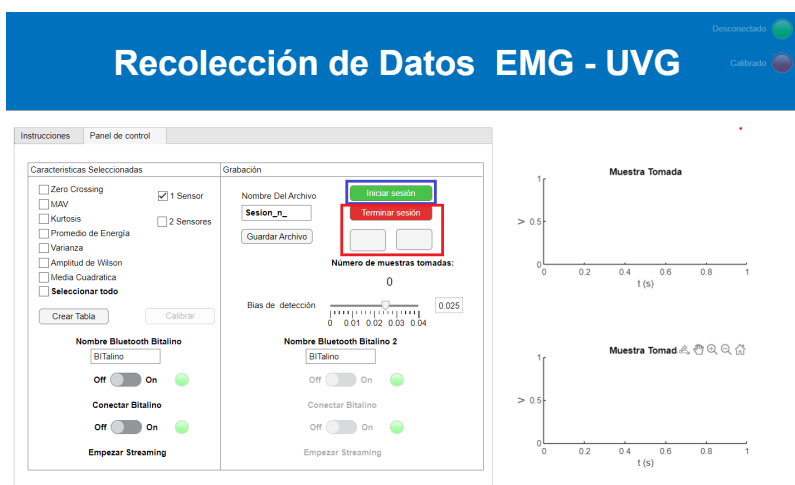


Figura 7: Botones de control de recolección.

Si se quiere modificar el umbral de detección de señales, se puede mover el deslizador observado en el cuadro rojo en la Figura 8.

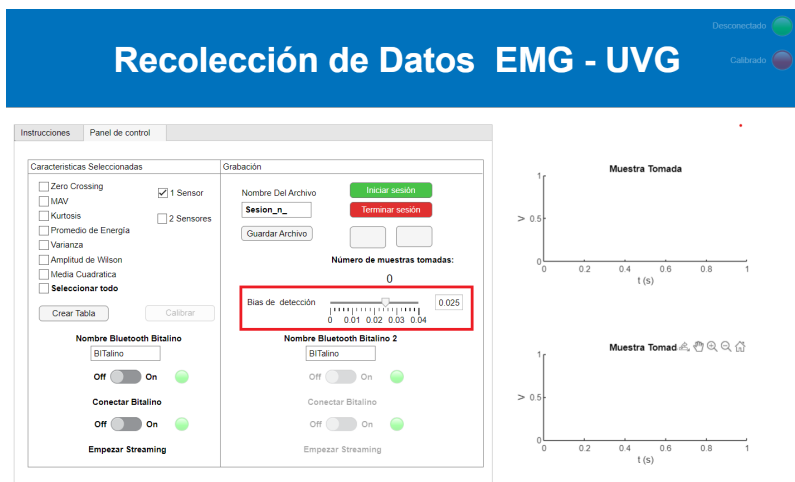


Figura 8: Deslizador de umbral.

Las señales obtenidas se pueden observar en el lado izquierdo del módulo.

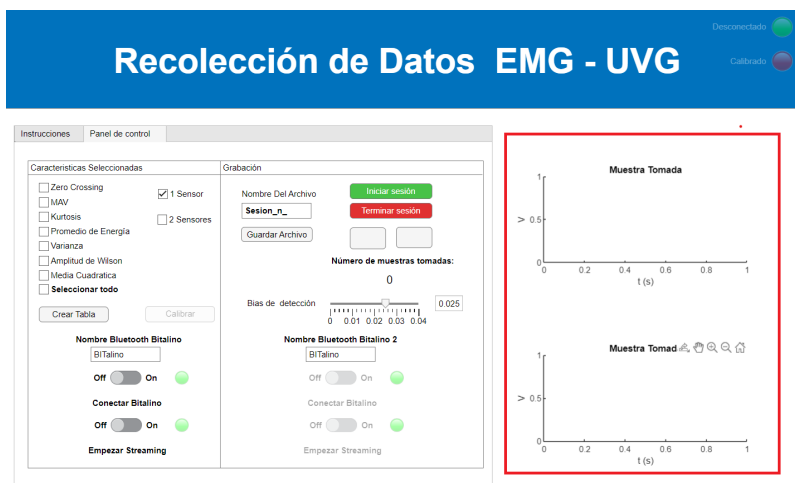


Figura 9: Gráficas de despliegue de señales.

Para guardar las señales recolectadas primero se debe de pausar la recolección, y escribir el nombre deseado del archivo en el espacio proporcionado. Al tener el nombre deseado solo se debe de presionar el botón de “Guardar Archivo”.

NOTA: Tomar en cuenta que al Finalizar la sesión se borrarán las señales recolectadas, si no se han guardado anteriormente.

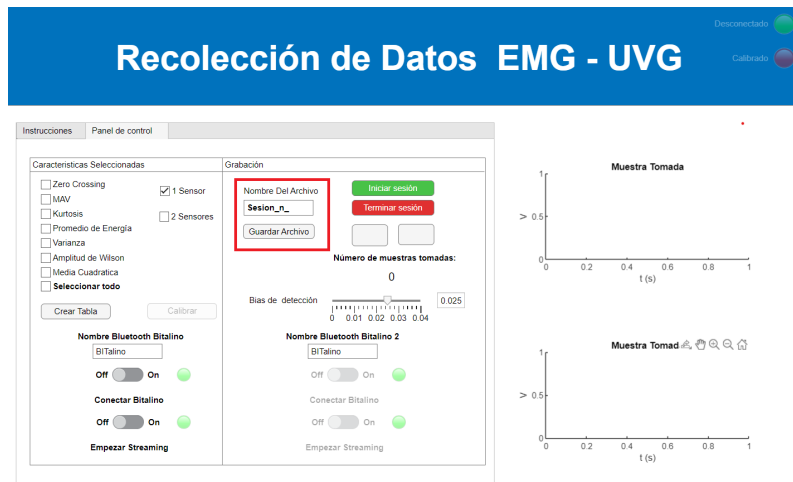


Figura 10: Panel de conexión: Guardado de archivo.

3.2. Módulo de control

Este módulo constituye el núcleo principal de la interfaz de control, integrando tanto las funciones de control como los algoritmos de aprendizaje automático.

Cuenta con distintos modos de operación, todos basados en el uso de sensores inerciales para la interacción. La principal diferencia entre estos modos radica en el tipo y número de sensores empleados.

El **modo binario** (o de intensidad) utiliza el sensor *MyoWare* para ejecutar acciones básicas, mientras que los **modos de uno y dos sensores** emplean el sensor *BITalino* para una clasificación más compleja mediante aprendizaje automático.

La interfaz fue diseñada para trabajar con la mano animatrónica UVG, pero fácilmente puede ser adaptada a trabajar con el actuador deseado. Por facilidad la interfaz posee un modo “*Debug*” con el cual se puede utilizar la misma sin necesidad de un actuador.

Al abrir la interfaz se pueden encontrar instrucciones para el uso de la misma en sus distintos modos, como se puede observar en la Figura 11.

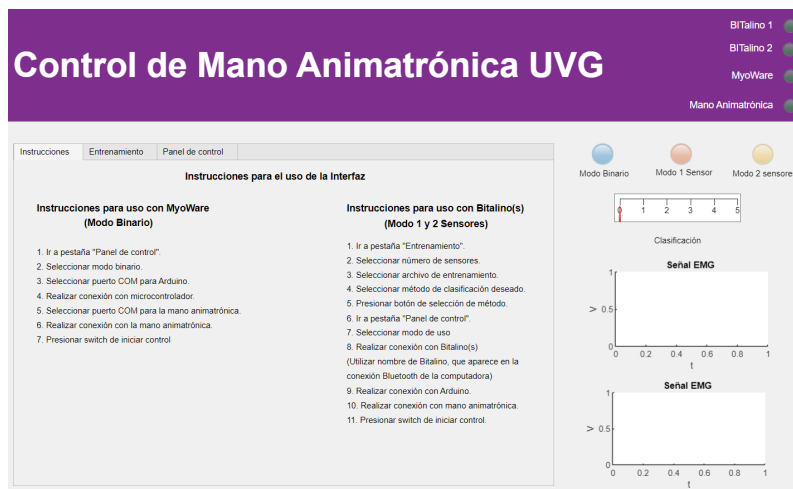


Figura 11: Pantalla inicial de interfaz.

A continuación se mostrara detalladamente el procedimiento de conexión y uso de cada modo.

3.2.1. Modo binario

Como se mencionó anteriormente este modo trabaja por medio del sensor MyoWare y el MPU6050. Además de necesitar la conexión con el actuador (En el caso NO este en modo Debug).

Para iniciar se debe ir a la pestaña de “Panel de Control”. Como se observa en la Figura 12.

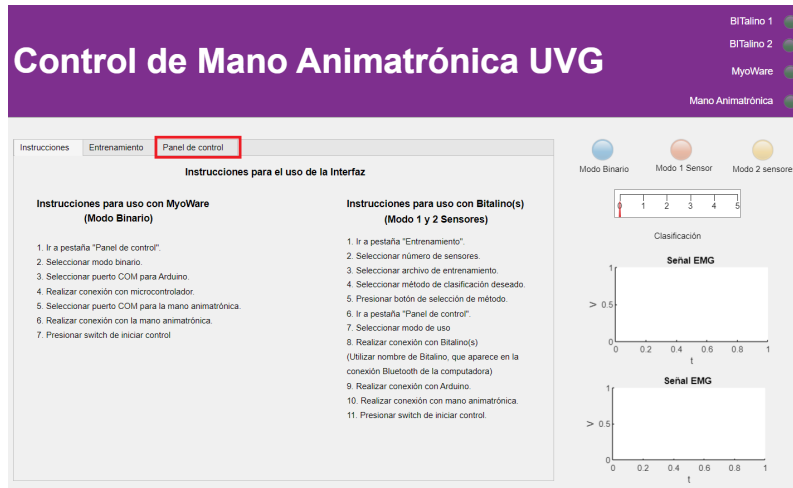


Figura 12: Pestaña Panel de control.

Para continuar se debe de seleccionar la opción de “Modo Binario” en las opciones que se pueden observar en la Figura 13.

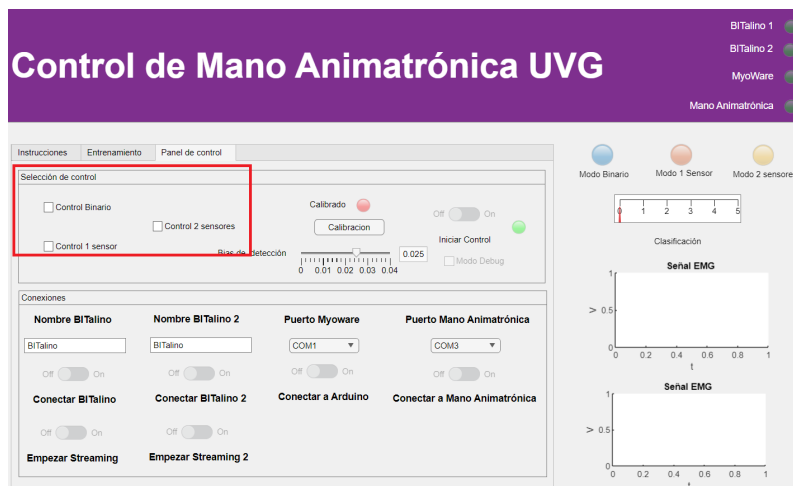


Figura 13: Selección de modo.

Al seleccionar la opción se habilitarán las opciones de conexión, en el caso de este modo la conexión con el MyoWare y la Mano animatrónica. Por lo que se debe seleccionar un puerto COM de conexión y habilitar el switch de encendido.

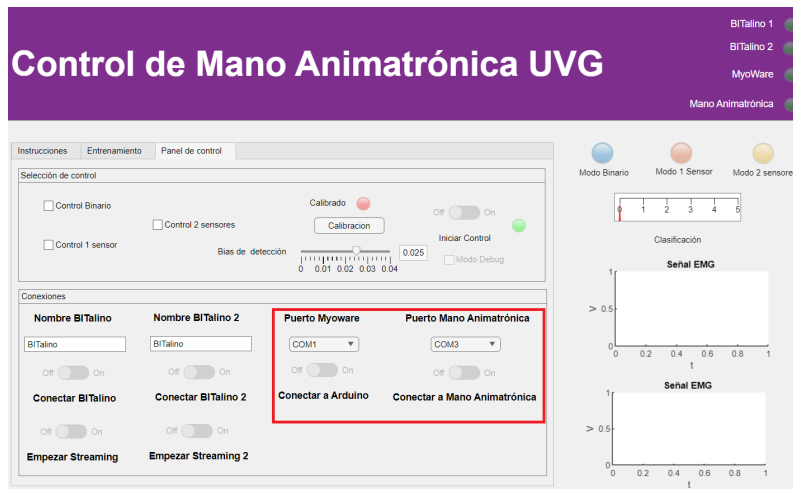


Figura 14: Conexiones modo binario.

Al haber realizado las conexiones se encenderán los LEDs respectivos en la esquina superior derecha.

Para realizar el control solo se debe accionar el switch de “Iniciar Control” y la interfaz empezará a funcionar.

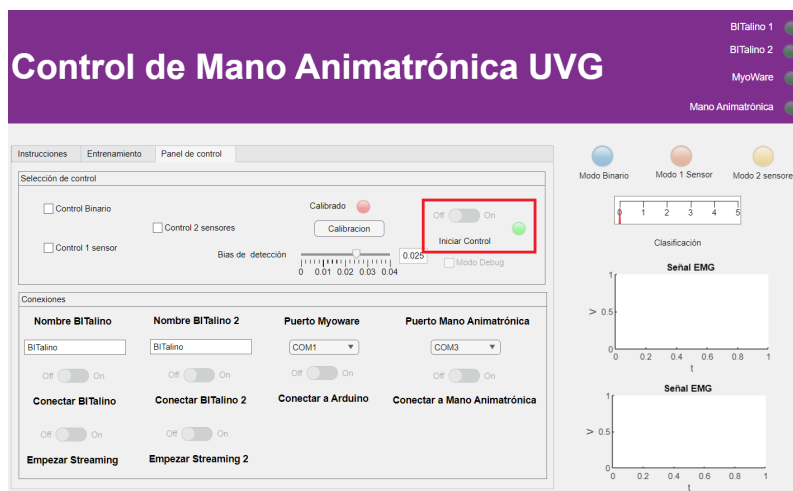


Figura 15: Inicialización de control.

En el caso de querer detener el control, solo se debe de accionar nuevamente el switch de “Iniciar Control”.

3.2.2. Modo 1 sensor y Modo 2 sensores

Estos modos funcionan similar con la única diferencia de la conexiones necesarias para el control del actuador.

Para iniciar nos debemos de ir a la pestaña de “Entrenamiento” para empezar el entrenamiento de nuestro clasificador automático. Esta pestaña se puede ver en la Figura 16

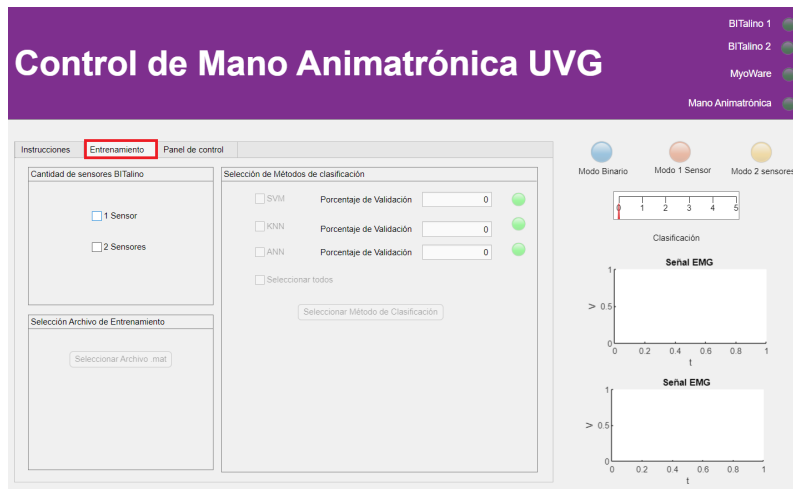


Figura 16: Pestaña de entrenamiento.

Para continuar debemos de seleccionar cuantos sensores queremos utilizar. Esto cambiaría los modelos de clasificación a utilizar.

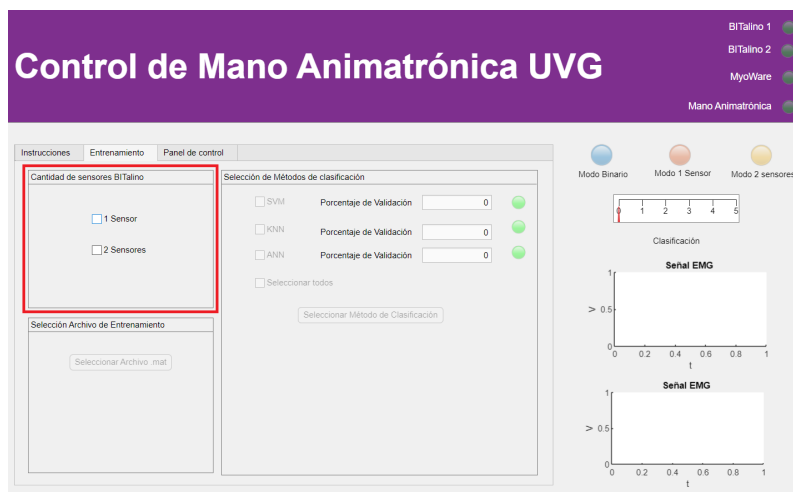


Figura 17: Selección de cantidad de sensores.

Al haber seleccionado la cantidad de sensores se habilitará la opción de seleccionar un archivo de entrenamiento. Este archivo debe de contener los gestos y las etiquetas con las que se deseen entrenar los modelos.

NOTA: Los modelos de clasificación seleccionados utilizan señales extraídas del módulo de recolección de señales utilizando 2 características.

NOTA: Hay que notar que el programa solo acepta datos con una matriz de tamaño específico ($n \times 7$ para 1 sensor y $n \times 13$ para 2 sensores) con el nombre “BaseCuartoTOT”.

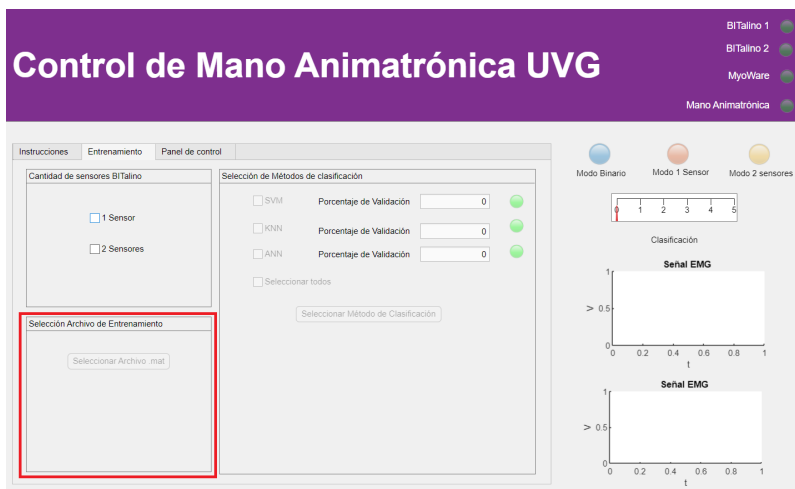


Figura 18: Selección de archivo de entrenamiento.

Al haber seleccionado el archivo, se habilitará la opción de seleccionar un modelo de clasificación. Al seleccionar el deseado la interfaz entrenará el modelo y desplegará el porcentaje de validación. La interfaz automáticamente seleccionará el modelo con porcentaje de validación más alto (En caso se entrene más de uno).

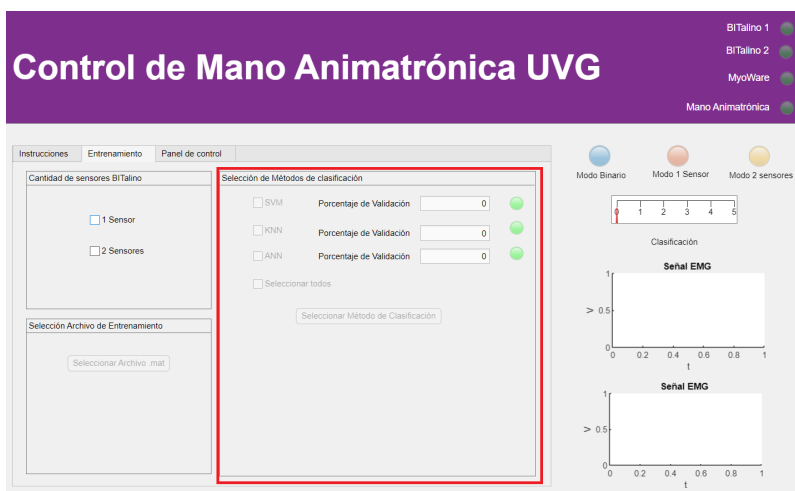


Figura 19: Selección de modelo de clasificación.

Al haber completado esto la interfaz habrá entrenado un modelo de clasificación y estará listo para realizar las conexiones necesarias.

Debemos de ir a la pestaña de “Panel de control”. Esta se puede observar en la Figura 12.

Al seleccionar el modo de uso, como se observa en la Figura 13. Podemos iniciar con las conexiones de los diferentes sensores.

La diferencia entre ambos modos cae en la cantidad de sensores BITalino se hará la conexión.

Para realizar la conexión con el MPU6050 y la Mano Animatrónica, se debe de seleccionar el puerto COM donde este la conexión y accionar el switch de conectar. Las conexiones exitosas activarán los LEDs en la esquina superior derecha.

Para realizar la conexión de los BITalino, se debe empezar con escribir el nombre que demostró la conexión bluetooth en las configuraciones de la computadora y accionar el switch de conectar.

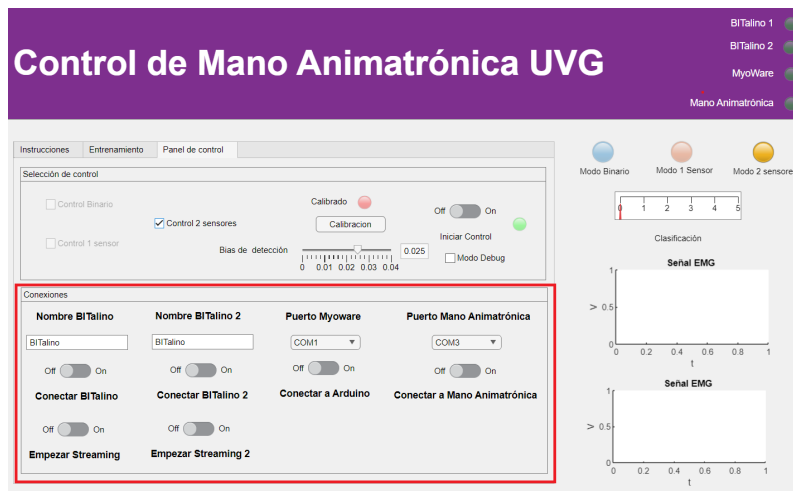


Figura 20: Conexiones de sensores y actuadores.

Al haber realizado la conexión con el BITalino, se debe empezar el *Streaming* de los datos accionando el switch de *Streaming*.

Por lo que se debe proceder a calibrar el algoritmo de detección automática. Esto se hace presionando el botón de calibración que se observa en la Figura 21 y esperando 5 segundos. Al finalizar se encenderá el LED rojo arriba de el botón.

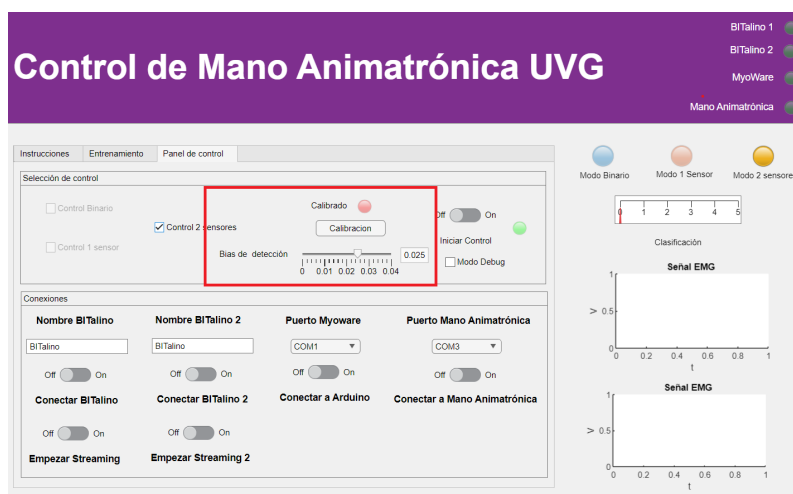


Figura 21: Calibración de control.

Al haber calibrado el algoritmo de detección, se puede proceder a iniciar el control. En caso se detecte que el umbral esta muy alto o bajo, este se puede editar en el deslizador que se encuentra abajo del botón de calibración.

Para iniciar el control se debe de accionar el switch de “Iniciar Control”. Este se observa en la Figura 15.

En el lado derecho de la interfaz se pueden observar las gráficas donde se despliegan las señales detectadas y un indicador que despliega la clasificación de la señal. Esto se observa en la Figura 22.

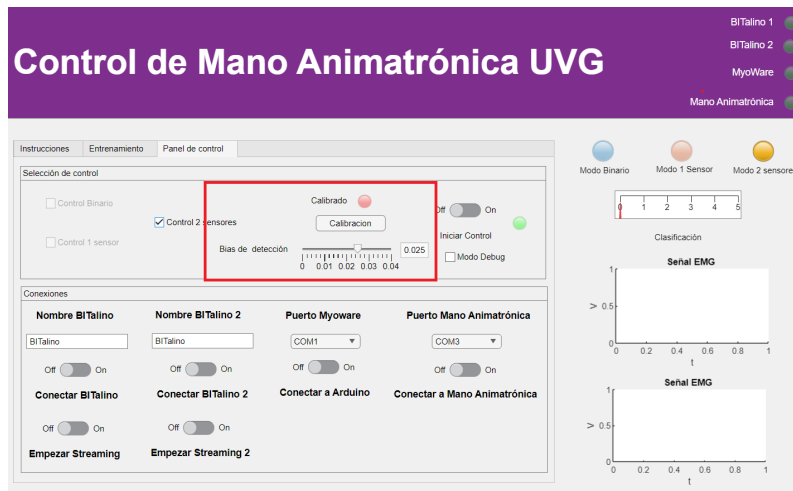


Figura 22: Calibración de control.

