

---

# Diseño e implementación de una interfaz biomédica para el control de agentes robóticos móviles

---

Saby Ximena Andrade Martinez





UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e implementación de una interfaz biomédica para el  
control de agentes robóticos móviles**

Trabajo de graduación presentado por Saby Ximena Andrade Martinez  
para optar al grado académico de Licenciada en Ingeniería Mecatrónica

Guatemala,

2025

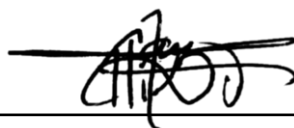
Vo.Bo.:



(f)

M. Sc. Carlos Esquit

Tribunal Examinador:

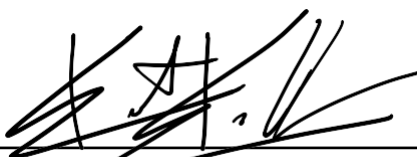


(f) M.Sc. Carlos Esquit



(f)

M. Sc. Miguel Enrique Zea Arenales



(f)

Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 07 de marzo de 2025.

Esta investigación surge de mi interés por la ingeniería biomédica, una disciplina que me ha apasionado desde que alguna vez soñé con ser doctora. También, se debe a la valiosa oportunidad que se presentó en la Universidad del Valle de Guatemala para integrarme a esta línea de investigación. Mi motivación principal ha sido desarrollar interfaces biomédicas que permitan la recolección y clasificación de señales biomédicas para el control de dispositivos robóticos.

Quiero expresar mi gratitud, en primer lugar, a Dios, por brindarme sabiduría, guía y esperanza en los momentos más difíciles, especialmente cuando dudaba de mi capacidad para culminar esta etapa; a mis padres, Fernando y Teresa, por su apoyo incondicional y constante presencia; A mis abuelos, Mario Andrade y Andrés Martínez, y a mis abuelas, Bessie Hernández y Bernandina Rayo, quienes siempre han sido pilares en mi vida; a mi hermano, mis tíos y amigos, quienes con sus palabras de aliento y su confianza me inspiraron, aunque a veces sentía que esa confianza aumentaba mi presión, nunca dejaron de creer en mí y me apoyaron en todo lo que necesitaba. También, quiero agradecer a mi gatita, Menny, quien fue una gran compañera durante este proceso; sus noches a mi lado, desvelándose conmigo, me ayudaron a calmar mi ansiedad y estrés.

Finalmente, agradezco profundamente a la comunidad de la Universidad del Valle de Guatemala. A todos los profesores y personal que contribuyeron a mi formación y aprendizaje, les estaré siempre agradecida. Un especial reconocimiento a mi asesor, el Dr. Luis Alberto Rivera, por su dedicación, sus acertados consejos académicos y sus críticas constructivas, que fueron fundamentales para completar este trabajo de graduación. Reconozco que en los últimos meses me había rendido y sentía que no merecía todo el esfuerzo que usted depositó en mí; sin embargo, nunca dejó de creer en mi capacidad, por eso, estaré eternamente agradecida.

# Índice

<b>Prefacio</b>	<b>III</b>
<b>Lista de figuras</b>	<b>X</b>
<b>Lista de cuadros</b>	<b>XI</b>
<b>Resumen</b>	<b>XII</b>
<b>Abstract</b>	<b>XIII</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Antecedentes</b>	<b>2</b>
2.1 Antecedentes externos a UVG . . . . .	2
2.1.1 Modelo virtual de prótesis de mano controlada mediante señales EEG	2
2.1.2 Teleoperación de un robot terrestre mediante biopotenciales . . . . .	4
2.2 Antecedentes internos de UVG . . . . .	6
2.2.1 Interfaz biomédica para el control de sistemas robóticos utilizando señales EMG . . . . .	6
2.2.2 Interfaz biomédica para el control de sistemas robóticos utilizando señales EEG . . . . .	9
2.2.3 Interfaz para el control de sistemas robóticos utilizando combinaciones de señales biomédicas . . . . .	11
<b>3 Justificación</b>	<b>15</b>
<b>4 Objetivos</b>	<b>17</b>
4.1 Objetivo general . . . . .	17
4.2 Objetivos específicos . . . . .	17
<b>5 Alcance</b>	<b>18</b>
<b>6 Marco teórico</b>	<b>19</b>
6.1 Señales bioeléctricas . . . . .	19
6.2 Señales electromiográficas (EMG) . . . . .	19
6.3 Señales electroencefalográficas (EOG) . . . . .	19
6.4 Características en el tiempo para las señales EMG . . . . .	20
6.4.1 Media del valor absoluto (MAV) . . . . .	20
6.4.2 Zero crossing . . . . .	20

6.5	Bitalino . . . . .	21
6.5.1	Ubicación de electrodos . . . . .	21
6.5.2	Registro de movimientos para el desarrollo de bases de datos de las señales EMG . . . . .	22
6.6	Biopac . . . . .	22
6.6.1	Ubicación de electrodos . . . . .	23
6.7	Conversión de señales analógicas a digitales con biopac . . . . .	24
6.7.1	Configuración de hardware . . . . .	24
6.7.2	Configuración de software . . . . .	24
6.8	Aprendizaje automático . . . . .	25
6.8.1	Aprendizaje supervisado . . . . .	25
6.8.2	Clasificador de máquinas lineales . . . . .	25
6.9	Máquina lineal (Linear machine) . . . . .	25
6.9.1	Análisis discriminante lineal (LDA) . . . . .	26
6.9.2	Aprendizaje bayesiano ( <i>Bayesian learning</i> ) . . . . .	27
6.9.3	Clasificador de máquinas de vectores de soporte (SVM) . . . . .	28
6.9.4	Redes neuronales artificiales . . . . .	31
6.9.5	Clasificador <i>K-Nearest Neighbors</i> (KNN) . . . . .	34
6.10	Robot pololu+3 pi . . . . .	36
6.11	Interfaz hombre-máquina (HMI) . . . . .	36
6.12	Comunicación entre agentes físicos e interfaces de usuario . . . . .	36
<b>7</b>	<b>Adquisición de señales EMG</b>	<b>38</b>
7.1	Obtención de las señales con el bitalino . . . . .	38
7.1.1	Desarrollo del algoritmo de lectura . . . . .	40
7.1.2	Generación de base de datos propia . . . . .	41
7.2	Filtrado y procesamiento de señales . . . . .	42
7.3	Interfaz de recolección de señales EMG . . . . .	43
<b>8</b>	<b>Clasificación de señales EMG</b>	<b>45</b>
8.1	Extracción de características de las señales filtradas . . . . .	45
8.2	Entrenamiento de clasificadores por máquina de vectores de soporte (SVM) con kernels . . . . .	46
8.2.1	Pruebas preliminares de clasificación con ventana fija de la señal . . . . .	46
8.2.2	Clasificación por segmentos de la señal . . . . .	48
8.3	Entrenamiento de clasificadores por máquina lineal . . . . .	50
8.3.1	Pruebas preliminares de clasificación con ventana fija de la señal . . . . .	50
8.3.2	Clasificación por segmentos de la señal . . . . .	51
8.4	Entrenamiento de clasificadores por redes neuronales (ANN) . . . . .	53
8.4.1	Pruebas preliminares de clasificación con ventana fija de la señal . . . . .	54
8.4.2	Clasificación por segmentos de la señal . . . . .	61
8.5	Entrenamiento de clasificadores con el algoritmo de los k vecinos más cercanos ( <i>K- nearest neighbor</i> ) . . . . .	67
8.5.1	Pruebas preliminares de clasificación con ventana fija de la señal . . . . .	68
8.5.2	Clasificación por segmentos de la señal . . . . .	69
8.5.3	Promedio de los resultados . . . . .	70
<b>9</b>	<b>Adquisición y conversión de señales EOG</b>	<b>73</b>

9.1	Adquisición de señales EOG . . . . .	73
9.1.1	Familiarización con el sistema biopac . . . . .	73
9.1.2	Desarrollo del algoritmo de lectura . . . . .	74
9.1.3	Comunicación y control con ESP32 . . . . .	74
9.2	Configuración del sistema . . . . .	75
9.2.1	Canales de adquisición . . . . .	75
9.3	Preprocesamiento de señales . . . . .	75
9.3.1	Filtro de diferencia (C1) . . . . .	75
9.3.2	Filtro de suavizado (C2) . . . . .	75
9.3.3	Umbrales de control (C3 y C4) . . . . .	75
9.4	Proceso de conversión de las señales . . . . .	75
<b>10</b>	<b>Validación de algoritmos por medio de simulación</b>	<b>77</b>
10.1	Verificación de recepción de señales . . . . .	77
10.1.1	Señales digitales del <i>ESP32</i> . . . . .	77
10.1.2	Servomotor . . . . .	78
10.2	Captura de datos con biopac . . . . .	79
10.3	Captura de datos con bitalino . . . . .	81
<b>11</b>	<b>Validación de algoritmos por medio de agente físico</b>	<b>82</b>
11.1	Modo de operación manual . . . . .	82
11.2	Recepción de datos del <i>ESP32</i> . . . . .	83
11.3	Traducción de las señales a comandos de control . . . . .	84
11.3.1	Biopac . . . . .	84
11.3.2	Bitalino . . . . .	85
<b>12</b>	<b>Conclusiones</b>	<b>88</b>
<b>13</b>	<b>Recomendaciones</b>	<b>89</b>
<b>14</b>	<b>Referencias</b>	<b>90</b>

# Lista de figuras

Figura 1.	Gráfica del análisis de ERD [1]. . . . .	3
Figura 2.	Modelo virtual validado [1]. . . . .	3
Figura 3.	Simulación del movimiento en Unity [1]. . . . .	4
Figura 4.	Filtros de paso bajo, alto y banda [2]. . . . .	5
Figura 5.	Programación en LabVIEW [2]. . . . .	5
Figura 6.	Transmisión de datos entre Enobio y LabVIEW [2]. . . . .	5
Figura 7.	Acceso a biblioteca, adquisición, filtrado y puesta en cola de los datos adquiridos. [2]. . . . .	6
Figura 8.	Diseño de filtro (pasa banda tipo Butterworth [2]). . . . .	6
Figura 9.	Diseño de filtro (rechaza banda tipo Butterworth [2]). . . . .	7
Figura 10.	Filtrado de señal de prueba [2]. . . . .	7
Figura 11.	Ventana completa que incluye la señal sin movimiento más la señal con movimiento [2]. . . . .	7
Figura 12.	Señal sin movimiento y detección de umbral. [2]. . . . .	8
Figura 13.	Resultados con SVM polinomial para clasificación de 6 clases, 2 características. [2]. . . . .	8
Figura 14.	Resultados con SVM polinomial para clasificación de 6 clases, 3 características. [2]. . . . .	8
Figura 15.	Resultados con SVM polinomial para clasificación de 6 clases, 5 características [2]. . . . .	9
Figura 16.	Resultados con SVM polinomial para clasificación de 6 clases, 7 características [2]. . . . .	9
Figura 17.	Señal continua y discreta [4]. . . . .	10
Figura 18.	Diagrama de bloques de procesamiento digital de señales [4]. . . . .	10
Figura 19.	Respuesta ideal de cuatro tipos de filtros selectores de frecuencia: (a) filtro pasa bajas; (b) filtro pasa altas; (c) filtro pasa banda; (d) filtro rechaza banda [4]. . . . .	10
Figura 20.	Matrices de confusión para dos tipos de clasificadores con señales propias: (a) SVM polinomio cúbico; (b) redes neuronales artificiales (RNA) [4]. . . . .	11
Figura 21.	Robot R17 [5]. . . . .	12
Figura 22.	Robot R17 (grados de libertad [5]). . . . .	12
Figura 23.	Interfaz recolectora de Datos [5]. . . . .	13
Figura 24.	Interfaz clasificadora EEG-EMG [5]. . . . .	14
Figura 25.	Movimientos realizados en la base de datos [5]. . . . .	14
Figura 26.	Movimientos realizados por los sujetos [5]. . . . .	14
Figura 27.	Bitalino [10]. . . . .	21
Figura 28.	Ubicación de electrodos para señales EMG [10]. . . . .	22

Figura 29. Dispositivo biopac MP36 [8]. . . . .	23
Figura 30. Configuración de los electrodos para capturar señales EOG [12]. . . . .	24
Figura 31. Puerto I/O del dispositivo MP36 [13]. . . . .	24
Figura 32. Configuración de canales digitales en el software biopac student lab para la adquisición de señales [13]. . . . .	25
Figura 33. Hiperplanos de separación en un espacio bidimensional [20]. . . . .	29
Figura 34. Margen asociado a un hiperplano no óptimo frente a un hiperplano óptimo [20]. . . . .	30
Figura 35. Estrategia One-vs-All para clasificación multiclase [21]. . . . .	30
Figura 36. Estrategia One-vs-One para clasificación multiclase [21]. . . . .	31
Figura 37. Representación de un elemento procesador en una red neuronal artificial [23]. . . . .	32
Figura 38. Arquitectura de una red neuronal simple [23]. . . . .	33
Figura 39. Modelo del robot Pololu 3Pi+ en su configuración estándar [25]. . . . .	36
Figura 40. Segmentación de la señal en tres partes identificando cruces por cero. . . . .	39
Figura 41. Cálculo de la media absoluta de valor (MAV) para cada segmento. . . . .	40
Figura 42. Diagrama de flujo del reconocimiento de señales EMG y segmentación. . . . .	41
Figura 43. Primer movimiento incluido en la base de datos: apretar y levantar la muñeca una vez. . . . .	42
Figura 44. Segundo movimiento incluido en la base de datos: levantar cada dedo creando un arco hasta cerrar el puño con suavidad. . . . .	42
Figura 45. Tercer movimiento incluido en la base de datos: apretar y levantar la muñeca dos veces. . . . .	42
Figura 46. Interfaz de recolección de señales EMG (nueva versión.) . . . . .	43
Figura 47. Diagrama de flujo del algoritmo de interfaz de recolección EMG. . . . .	44
Figura 48. Matriz de confusión del clasificador SVM con kernel gaussiano de las pruebas preliminares. . . . .	46
Figura 49. Matriz de confusión del clasificador SVM con kernel polinomial de las pruebas preliminares. . . . .	47
Figura 50. Matriz de confusión del clasificador SVM multiclase de las pruebas preliminares. . . . .	47
Figura 51. Matriz de confusión del clasificador SVM con kernel gaussiano por segmentos de la señal. . . . .	48
Figura 52. Matriz de confusión del clasificador SVM con kernel polinomial por segmentos de la señal. . . . .	49
Figura 53. Matriz de confusión del clasificador SVM multiclase por segmentos de la señal. . . . .	49
Figura 54. Matriz de confusión del clasificador análisis discriminante lineal de las pruebas preliminares. . . . .	50
Figura 55. Matriz de confusión del clasificador SVM lineal de las pruebas preliminares. . . . .	51
Figura 56. Matriz de confusión del clasificador bayesiano lineal de las pruebas preliminares. . . . .	51
Figura 57. Matriz de confusión del clasificador análisis discriminante lineal por segmentos de la señal. . . . .	52
Figura 58. Matriz de confusión del clasificador SVM lineal por segmentos de la señal. . . . .	52

Figura 59. Matriz de confusión del clasificador bayesiano lineal por segmentos de la señal. . . . .	53
Figura 60. Proceso de entrenamiento de red neuronal de 2 capas para las pruebas preliminares. . . . .	54
Figura 61. Matriz de confusión de 2 capas para las pruebas preliminares. . . . .	55
Figura 62. Red neuronal de 2 capas para las pruebas preliminares. . . . .	55
Figura 63. Proceso de entrenamiento de red neuronal de 3 capas para las pruebas preliminares . . . . .	56
Figura 64. Matriz de confusión de 3 capas para las pruebas preliminares. . . . .	57
Figura 65. Red neuronal de 3 capas para las pruebas preliminares. . . . .	58
Figura 66. Proceso de entrenamiento de red neuronal de 50 capas ocultas para las pruebas preliminares . . . . .	59
Figura 67. Matriz de confusión de 50 capas ocultas para las pruebas preliminares. . . . .	60
Figura 68. Red neuronal de 50 capas ocultas para las pruebas preliminares. . . . .	61
Figura 69. Proceso de entrenamiento de red neuronal de 2 capas por segmentos de la señal. . . . .	62
Figura 70. Matriz de confusión de 2 capas por segmentos de la señal. . . . .	62
Figura 71. Red neuronal de 2 capas por segmentos de la señal. . . . .	63
Figura 72. Proceso de entrenamiento de red neuronal de 3 capas por segmentos de la señal. . . . .	64
Figura 73. Matriz de confusión de 3 capas por segmentos de la señal. . . . .	64
Figura 74. Red neuronal de 3 capas por segmentos de la señal. . . . .	65
Figura 75. Proeso de entrenamiento de red neuronal de 50 capas ocultas por segmentos de la señal.c . . . . .	66
Figura 76. Matriz de confusión de 50 capas ocultas por segmentos de la señal. . . . .	66
Figura 77. Red neuronal de 50 capas ocultas por segmentos de la señal. . . . .	67
Figura 78. Matriz de confusión del clasificador KNN estándar con ventana fija de la señal. . . . .	68
Figura 79. Matriz de confusión del clasificador KNN ponderado por distancia con ventana fija de la señal. . . . .	68
Figura 80. Matriz de confusión del clasificador KNN adaptativo con ventana fija de la señal. . . . .	69
Figura 81. Matriz de confusión del clasificador KNN estándar por segmentos de la señal. . . . .	69
Figura 82. Matriz de confusión del clasificador KNN ponderado por segmentos de la señal. . . . .	70
Figura 83. Matriz de confusión del clasificador KNN adaptativo por segmentos de la señal. . . . .	70
Figura 84. Diagrama de flujo del proceso de adquisición y conversión de señales EOG. . . . .	74
Figura 85. Visualización del monitor serial del servomotor con las señales digitales	78
Figura 86. Movimiento del servomotor en función de las señales digitales procesadas.	78
Figura 87. Diagrama de flujo de la configuración entre el biopac y el ESP32 . . . . .	80
Figura 88. Visualización del funcionamiento de la simulación en Webots. . . . .	80
Figura 89. Diagrama de flujo del bitalino. . . . .	81

Figura 90. Interfaz de operación con el agente robótico. . . . .	83
Figura 91. Diagrama de flujo de la configuración entre el biopac y el ESP32. . . . .	85
Figura 92. Visualización de la conexión del biopac. . . . .	85
Figura 93. Diagrama de flujo de la configuración ente el bitalino y el robot móvil. . . . .	87
Figura 94. Visualización de la conxión del bitalino. . . . .	87

# Lista de cuadros

Cuadro 1. Registro del porcentaje de aciertos después de utilizar el sistema de entrenamiento VR. [1]. . . . .	4
Cuadro 2. Comparación de resultados de entrenamiento con diferentes combinaciones y cantidad de características temporales. [5]. . . . .	13
Cuadro 3. Resultados por modelo sin incluir promedios. . . . .	71
Cuadro 4. Promedio de resultados por modelo. . . . .	71
Cuadro 5. Promedio total de los resultados para todas las configuraciones. . . . .	72

Este proyecto consistió en el desarrollo de interfaces biomédicas para la recolección y clasificación de señales electromiográficas (EMG) y electrooculográficas (EOG), con el fin de controlar agentes robóticos móviles en tiempo real. Estas interfaces permiten la adquisición y procesamiento de señales bioeléctricas mediante dispositivos como el bitalino para EMG y el biopac para EOG, generando bases de datos propias que pueden ser utilizadas para desarrollar modelos de clasificación o realizar análisis detallados de las señales.

El proyecto incluyó la implementación de técnicas de aprendizaje automático, utilizando algoritmos como máquinas de soporte de vectores (SVM) y redes neuronales (RN), para entrenar clasificadores con vectores de características extraídas de las señales procesadas. Las interfaces desarrolladas fueron integradas con agentes robóticos para realizar pruebas de control en tiempo real, logrando clasificaciones con altos niveles de precisión. Los modelos entrenados demostraron la efectividad de estas señales bioeléctricas como métodos no invasivos para el control de dispositivos robóticos.

Uno de los objetivos principales fue evaluar la combinación de señales EMG y EOG para mejorar la clasificación de gestos y movimientos. Las pruebas realizadas mostraron que esta combinación incrementa la precisión de los clasificadores en comparación con el uso individual de cada tipo de señal.

En el futuro, se espera que estas interfaces sirvan como herramientas para la creación de bases de datos y para investigaciones relacionadas con dispositivos de asistencia médica, aprovechando técnicas avanzadas de clasificación y el potencial de las señales bioeléctricas para aplicaciones innovadoras en el ámbito de la rehabilitación y la robótica.

**Palabras clave:** Señales electromiográficas, señales electrooculográficas, aprendizaje automático, clasificación de señales, interfaces biomédicas, control robótico.

This project involved the development of biomedical interfaces for the collection and classification of electromyographic (EMG) and electrooculographic (EOG) signals, aimed at controlling mobile robotic agents in real time. These interfaces enable the acquisition and processing of bioelectric signals using devices such as bitalino for EMG and biopac for EOG, generating proprietary databases that can be utilized for developing classification models or conducting detailed signal analyses.

The project included the implementation of machine learning techniques, employing algorithms such as Support Vector Machines (SVM) and Neural Networks (NN) to train classifiers using feature vectors extracted from processed signals. The developed interfaces were integrated with robotic agents to perform real-time control tests, achieving high levels of classification accuracy. The trained models demonstrated the effectiveness of these bioelectric signals as non-invasive methods for controlling robotic devices.

One of the main objectives was to evaluate the combination of EMG and EOG signals to enhance gesture and movement classification. The tests conducted showed that this combination increases classifier accuracy compared to using each type of signal individually.

In the future, these interfaces are expected to serve as tools for creating databases and advancing research related to assistive devices, leveraging advanced classification techniques and the potential of bioelectric signals for innovative applications in rehabilitation and robotics.

**Keywords:** Electromyographic signals, electrooculographic signals, machine learning, signal classification, biomedical interfaces, robotic control.

El estudio de señales biomédicas ha permitido importantes avances en la ciencia, especialmente en el desarrollo de dispositivos para asistencia médica y mejora de la calidad de vida de personas con discapacidades motrices. Estas señales han demostrado ser herramientas valiosas para la creación de sistemas de control que se integran con dispositivos robóticos y de asistencia, abriendo caminos hacia tecnologías más inclusivas y accesibles.

En este proyecto, se plantea el diseño e implementación de una interfaz biomédica capaz de controlar agentes robóticos móviles utilizando señales bioeléctricas, específicamente señales electromiográficas (EMG) y electrooculográficas (EOG). Estas señales son adquiridas, filtradas y procesadas en tiempo real utilizando dispositivos como el biopac, que permite la captura de señales EOG, y el bitalino para las señales EMG. El objetivo es no solo facilitar el control de agentes robóticos, sino también establecer una base sólida para futuros desarrollos en el uso de aprendizaje automático aplicado a señales biomédicas.

El presente trabajo incluye múltiples etapas desde la adquisición de señales hasta su clasificación mediante algoritmos avanzados. También, se desarrollaron interfaces para la recopilación y procesamiento de datos, junto con modelos de clasificación que envían comandos a robots físicos para su validación en tiempo real. Estas interfaces están diseñadas para ser intuitivas y funcionales, lo que permite su uso por otros investigadores y fomentar el crecimiento de esta línea de investigación dentro de la Universidad del Valle de Guatemala (UVG).

Finalmente, este proyecto busca contribuir al avance en el control de sistemas robóticos a través de señales bioeléctricas, con la esperanza de que estos desarrollos sean utilizados por la comunidad científica para innovar en tecnologías de asistencia y rehabilitación.

La presente sección aborda los antecedentes más relevantes relacionados con el uso de señales bioeléctricas en el control de dispositivos robóticos, tanto a nivel internacional como en investigaciones realizadas en la Universidad del Valle de Guatemala. Estos antecedentes reflejan los avances y desafíos en el desarrollo de interfaces biomédicas utilizando tecnologías como señales EEG y EMG para el control de sistemas robóticos. En el apartado de antecedentes externos, se describen proyectos internacionales enfocados en modelos virtuales y dispositivos físicos controlados mediante inteligencia artificial y señales bioeléctricas. Por otro lado, los antecedentes internos destacan los aportes realizados en la UVG, incluyendo el análisis de señales biomédicas y su integración en sistemas robóticos.

## **2.1. Antecedentes externos a UVG**

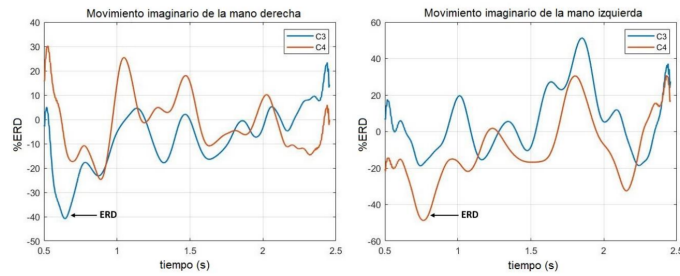
### **2.1.1. Modelo virtual de prótesis de mano controlada mediante señales EEG**

En el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional de Ciudad de México [1], se diseñó un modelo virtual de ambas manos capaz de simular individualmente la apertura y cierre, basándose en la interpretación de movimientos imaginarios de la mano derecha e izquierda. Estos movimientos se extrajeron de una base de datos pública y se capturaron utilizando la técnica de electroencefalografía durante la imaginación motora de las manos [1].

Los registros de EEG obtenidos durante las tareas de imaginación motora se procesaron para diseñar el modelo virtual y las animaciones que incorporan el movimiento. Este procesamiento incluyó la estimación de los Eventos Relacionados con la Desincronización (ERD) de cada participante, un método esencial para identificar patrones en las señales cerebrales relacionados con movimientos específicos. La Figura 1 presenta una gráfica detallada del análisis de ERD, donde se muestran los valores calculados para cada tarea de imaginación

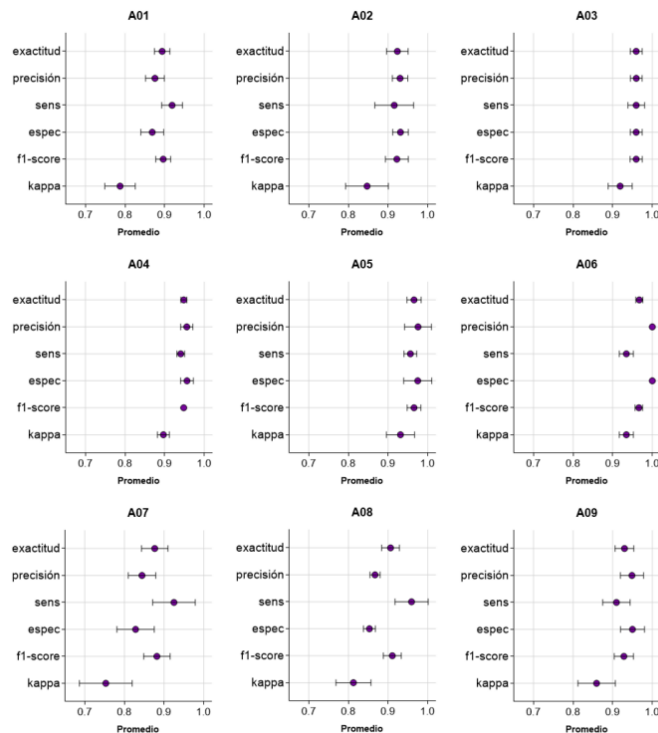
motora, permitiendo validar la relevancia de estos eventos en el diseño del sistema.

Figura 1: Gráfica del análisis de ERD [1].



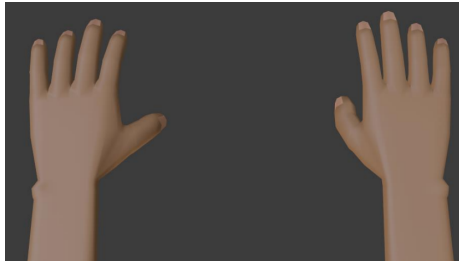
A partir de estos datos, se diseñó un modelo virtual que simula los movimientos de las manos con alta precisión. Este modelo, validado mediante pruebas de clasificación y simulación, se describe en la Figura 2, que muestra la estructura final del modelo, incluyendo las características visuales y funcionales implementadas en Unity. Este modelo representa un hito importante en el desarrollo de tecnologías no invasivas para la simulación de movimientos.

Figura 2: Modelo virtual validado [1].



Los modelos neuronales entrenados lograron clasificar entre un 87.66 % y 96.72 % de los movimientos imaginarios realizados por los participantes en las pruebas. Estos resultados destacan la efectividad de la plataforma desarrollada para interpretar correctamente los movimientos simulados. La Figura 3 detalla cómo Matlab y Unity interactúan para ejecutar estos movimientos virtuales, mostrando la sincronización entre las señales EEG procesadas y las acciones simuladas en tiempo real.

Figura 3: Simulación del movimiento en Unity [1].



Para el control del modelo virtual, se empleó el protocolo TCP/IP entre Matlab y Unity, logrando una transferencia de datos con un tiempo promedio de 0.015 segundos. Sin embargo, el estudio se limitó a un modelo virtual y no avanzó hacia la implementación de un dispositivo físico real.

En el Cuadro 1, se pueden apreciar los distintos gestos de control evaluados, tales como “Descanso”, “Puño”, “Tres dedos”, “Lápiz”, “L”, “Palma arriba”, y “Palma abajo”, y sus correspondientes porcentajes de aciertos, que indican la precisión con la cual el sistema de la prótesis es capaz de interpretar y ejecutar estos gestos correctamente.

Cuadro 1: Registro del porcentaje de aciertos después de utilizar el sistema de entrenamiento VR. [1].

<b>n</b>	<b>Gestos de control</b>	<b>Porcentaje de aciertos (X)</b>
1	Descanso	90%
2	Puño	90%
3	Tres dedos	85%
4	Lápiz	85%
5	L	85%
6	Palma arriba	90%
7	Palma abajo	90%

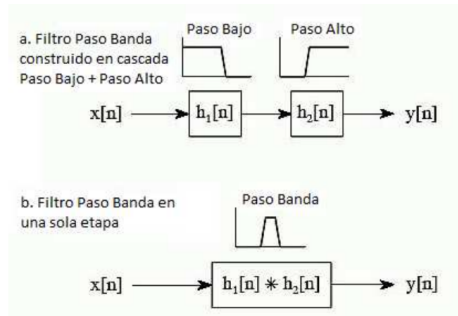
### 2.1.2. Teleoperación de un robot terrestre mediante biopotenciales

En la Universidad Carlos III de Madrid, en colaboración con National Instruments Spain y Starlab, se desarrolló un sistema innovador para el control de un robot terrestre mediante biopotenciales, específicamente señales EOG (Electrooculograma). Este proyecto integró el hardware Enobio, desarrollado por Starlab, con la plataforma LabVIEW Robotics de National Instruments para lograr un control eficaz del robot DaNI 1.0 [2].

El sistema tenía como objetivo principal crear una interfaz hombre-máquina inalámbrica para el control de robots terrestres. Las señales EOG fueron capturadas utilizando el hardware Enobio y procesadas con técnicas avanzadas como filtros paso bajo, paso alto y paso banda. En la Figura 4, se observan las configuraciones de estos filtros, esenciales para

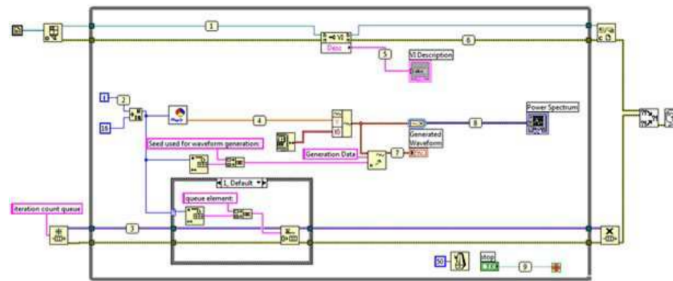
eliminar artefactos y ruido, permitiendo destacar las señales relevantes para el control del robot.

Figura 4: Filtros de paso bajo, alto y banda [2].



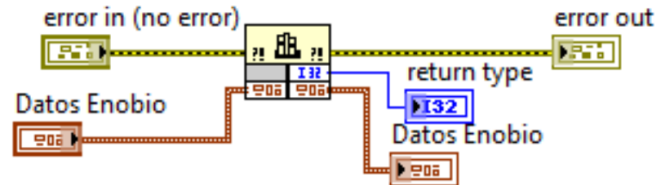
El sistema fue diseñado e implementado utilizando programación gráfica en LabVIEW. En la Figura 5, se presenta el esquema del programa desarrollado, que incluye módulos para la adquisición de datos, el análisis de señales y la generación de comandos de control. Este diseño permitió procesar las señales en tiempo real de manera eficiente.

Figura 5: Programación en LabVIEW [2].



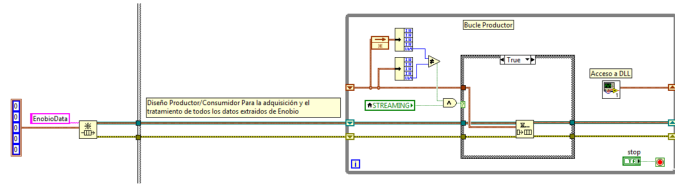
La transmisión de datos entre el hardware Enobio y la plataforma LabVIEW se optimizó para garantizar una adquisición estable y precisa de las señales EOG. En la Figura 6, se ilustra el esquema de transmisión de datos, detallando las mejoras implementadas para reducir la latencia y aumentar la confiabilidad del sistema.

Figura 6: Transmisión de datos entre Enobio y LabVIEW [2].



Además, se integraron procesos como el acceso a bibliotecas específicas, la adquisición y el filtrado de señales, y la puesta en cola de los datos adquiridos para su procesamiento en tiempo real. La Figura 7 muestra este flujo de trabajo, que garantiza un manejo de datos eficiente y adaptable a diferentes condiciones experimentales.

Figura 7: Acceso a biblioteca, adquisición, filtrado y puesta en cola de los datos adquiridos. [2].



Finalmente, el montaje físico del robot DaNI y su integración con el sistema EOG se validaron a través de pruebas experimentales en escenarios con obstáculos simulados y reales. Este proyecto representa un avance significativo en el desarrollo de interfaces cerebro-máquina, especialmente para aplicaciones de robótica asistiva y biomédica.

## 2.2. Antecedentes internos de UVG

### 2.2.1. Interfaz biomédica para el control de sistemas robóticos utilizando señales EMG

En la Universidad del Valle de Guatemala, Maria Fernanda Girón [3] desarrolló una interfaz biomédica para el control de sistemas robóticos utilizando señales electromiográficas de superficie (sEMG). Este trabajo se centró en adquirir, procesar y clasificar señales sEMG en tiempo real para controlar sistemas robóticos mediante algoritmos de aprendizaje automático, destacando el uso de máquinas de vectores de soporte (SVM) y redes neuronales (RN).

La adquisición de señales se llevó a cabo utilizando el dispositivo bitalino y el software OpenSignals, lo que permitió registrar señales electromiográficas con una calidad adecuada para su procesamiento. En la Figura 8, se muestra el diseño del filtro pasa banda tipo Butterworth utilizado para eliminar ruido y resaltar las señales relevantes, mientras que la Figura 9 ilustra el diseño del filtro rechaza banda aplicado para eliminar frecuencias específicas de interferencia.

Figura 8: Diseño de filtro (pasa banda tipo Butterworth [2]).

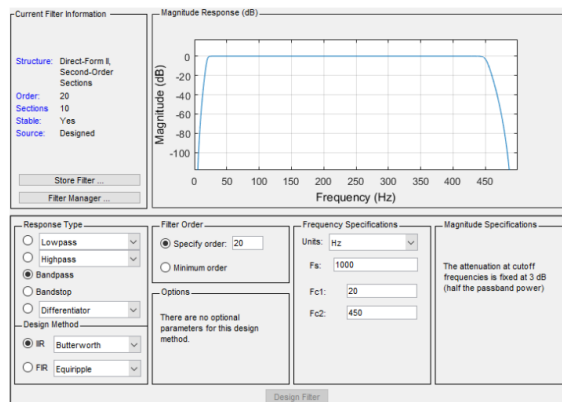
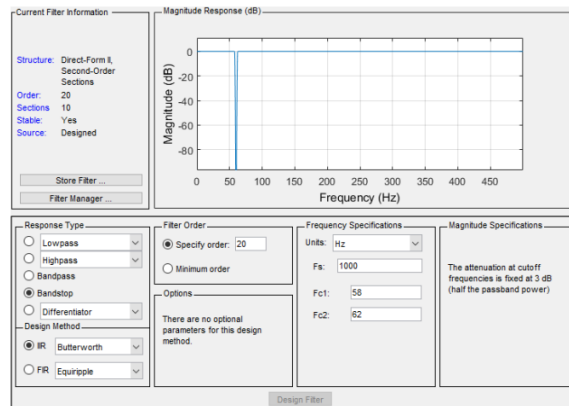
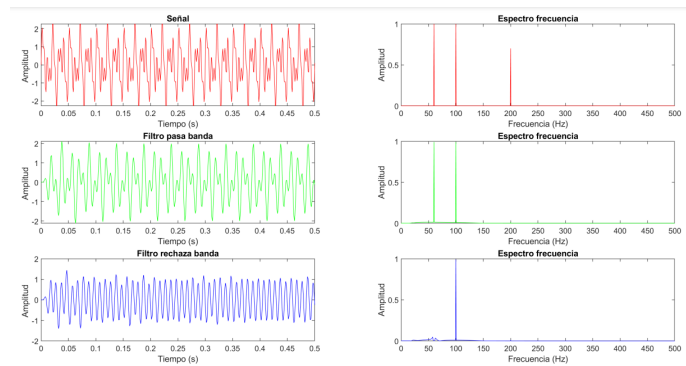


Figura 9: Diseño de filtro (rechaza banda tipo Butterworth [2]).



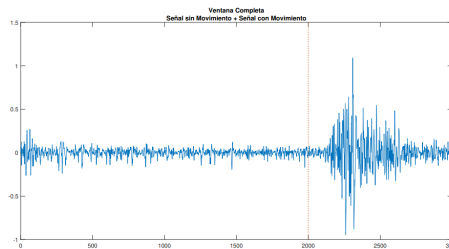
Estos filtros demostraron su eficacia al generar señales limpias para su análisis, como se observa en la Figura 10, donde se presenta el resultado del filtrado aplicado a una señal de prueba.

Figura 10: Filtrado de señal de prueba [2].



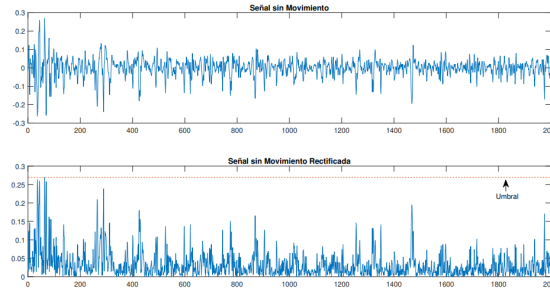
El procesamiento de señales incluyó la detección de umbrales y la segmentación de las señales en ventanas que contenían movimientos específicos. En la Figura 11, se muestra una ventana completa que incluye la señal sin movimiento y la señal con movimiento, lo que permitió establecer una separación clara para su posterior análisis.

Figura 11: Ventana completa que incluye la señal sin movimiento más la señal con movimiento [2].



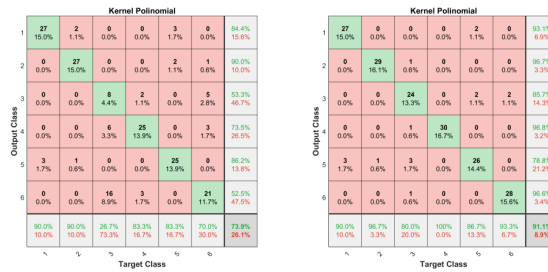
La Figura 12 detalla la detección del umbral en la señal sin movimiento, marcando el inicio de la actividad muscular.

Figura 12: Señal sin movimiento y detección de umbral. [2].



Para la clasificación de movimientos, se utilizaron características extraídas de las señales sEMG, como MAV, ZC, RMS y WL, las cuales se evaluaron en distintos modelos de SVM. En la Figura 13, se presentan los resultados obtenidos con un SVM polinomial para la clasificación de seis clases utilizando dos características, mostrando una precisión inicial moderada.

Figura 13: Resultados con SVM polinomial para clasificación de 6 clases, 2 características. [2].



Posteriormente, en la Figura 14, se observan mejoras en la clasificación al incluir tres características en el modelo. Este patrón de mejora continúa en la Figura 15, donde se utilizaron cinco características, y finalmente en la Figura 16, que muestra los mejores resultados al emplear siete características en el modelo.

Figura 14: Resultados con SVM polinomial para clasificación de 6 clases, 3 características. [2].

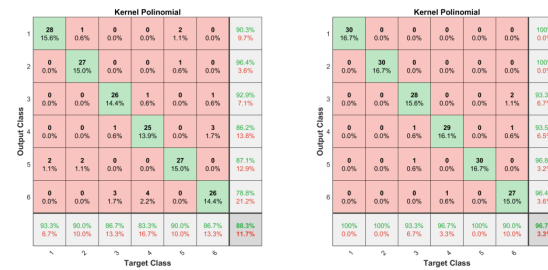


Figura 15: Resultados con SVM polinomial para clasificación de 6 clases, 5 características [2].

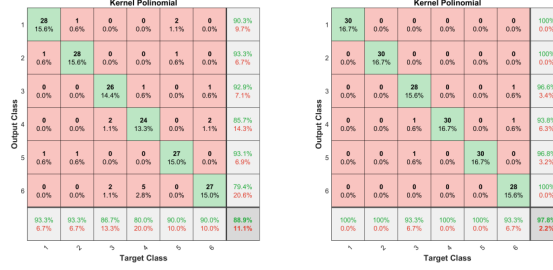
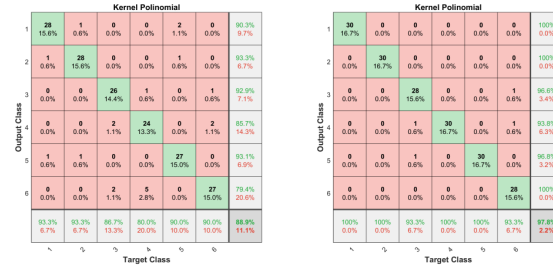


Figura 16: Resultados con SVM polinomial para clasificación de 6 clases, 7 características [2].



Los resultados experimentales mostraron un rendimiento promedio del 90 % con redes neuronales y del 80 % con SVM, destacando la eficacia de las características seleccionadas y los algoritmos empleados. Este desempeño permitió integrar la interfaz con un sistema robótico, logrando el control en tiempo real mediante movimientos clasificados.

Este proyecto representa un avance significativo en el uso de señales electromiográficas y aprendizaje automático para aplicaciones en robótica asistiva, estableciendo una base sólida para futuras investigaciones y desarrollos en este campo.

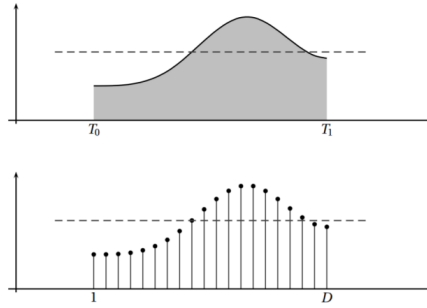
### 2.2.2. Interfaz biomédica para el control de sistemas robóticos utilizando señales EEG

En ese mismo año, Rodrigo Ralda [4] desarrolló una interfaz biomédica basada en señales EEG para el control de sistemas robóticos en la Universidad del Valle de Guatemala. Este trabajo replicó parcialmente la metodología de Girón, integrando señales EEG y evaluando su viabilidad en simulaciones. Aunque la pandemia de COVID-19 limitó el acceso a equipos físicos, los experimentos en simulaciones mostraron un rendimiento combinado superior al 80 %.

El trabajo se estructuró en varias etapas: adquisición y filtrado de señales, extracción de características relevantes como MAV, ZC, curtosis y varianza, y clasificación mediante algoritmos de aprendizaje automático como SVM y RNA. En la Figura 18, se observa la representación de una señal continua y discreta, destacando los principios fundamentales utilizados en el análisis de las señales EEG.

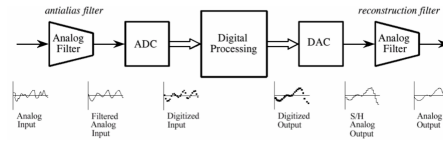
Además, en la Figura ??, se presenta el diagrama de bloques del procesamiento digital

Figura 17: Señal continua y discreta [4].



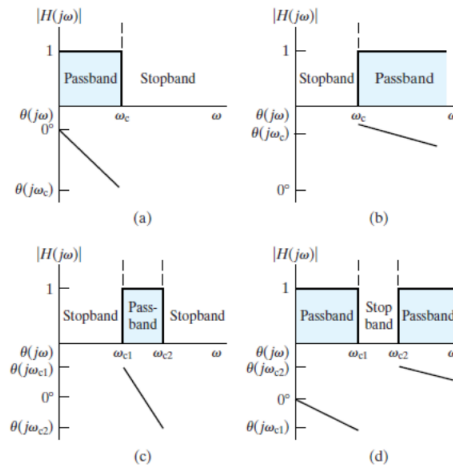
de señales, que detalla las etapas de preprocesamiento necesarias para limpiar las señales antes de la clasificación.

Figura 18: Diagrama de bloques de procesamiento digital de señales [4].



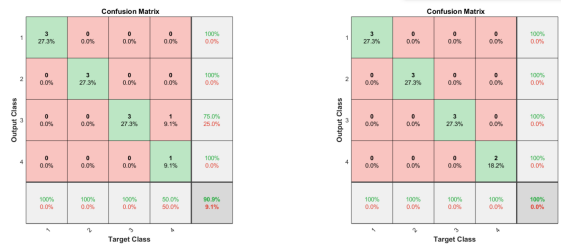
Los filtros utilizados en el procesamiento se describen en la Figura 19, que muestra las respuestas ideales de cuatro tipos de filtros selectores de frecuencia: pasa bajas, pasa altas, pasa banda y rechaza banda. Estos filtros permitieron optimizar la calidad de las señales EEG procesadas para su uso en el sistema de clasificación.

Figura 19: Respuesta ideal de cuatro tipos de filtros selectores de frecuencia: (a) filtro pasa bajas; (b) filtro pasa altas; (c) filtro pasa banda; (d) filtro rechaza banda [4].



Para la clasificación, se evaluaron dos algoritmos principales: SVM polinómico cúbico y Redes Neuronales Artificiales (RNA). Los resultados se analizaron mediante matrices de confusión, como se observa en la Figura 26, donde se presentan los rendimientos de los clasificadores aplicados a las señales propias. Los algoritmos demostraron ser efectivos para distinguir entre las diferentes clases de señales EEG procesadas.

Figura 20: Matrices de confusión para dos tipos de clasificadores con señales propias: (a) SVM polinomio cúbico; (b) redes neuronales artificiales (RNA) [4].



Finalmente, se desarrollaron simulaciones con manipuladores robóticos, incluyendo el R17, UR5 y Puma 560. Aunque no se realizaron pruebas con agentes físicos, los resultados demostraron la viabilidad de la metodología y sentaron las bases para futuras implementaciones en sistemas de control robótico basados en EEG en la Universidad del Valle de Guatemala.

### 2.2.3. Interfaz para el control de sistemas robóticos utilizando combinaciones de señales biomédicas

En 2021, Roberto José Cáceres García [5] llevó a cabo su trabajo de graduación en la Universidad del Valle de Guatemala para el control de sistemas robóticos mediante la combinación de señales EMG y EEG. El objetivo era traducir estas señales en comandos para controlar sistemas robóticos físicos. Se implementaron técnicas de aprendizaje supervisado como redes neuronales artificiales y máquinas de vectores de soporte.

Se determinó que la combinación de cuatro características específicas — MAV (valor absoluto medio), ZC (número de cruces por cero), WL (longitud de onda) y RMS (valor cuadrático medio)— fue la más adecuada para configurar la matriz de características de entrada en el proceso de entrenamiento del clasificador. En la Figura 21, se observa el robot R17 utilizado en este estudio, la Figura 22 detalla sus grados de libertad, esenciales para entender su funcionamiento en este contexto.

Figura 21: Robot R17 [5].

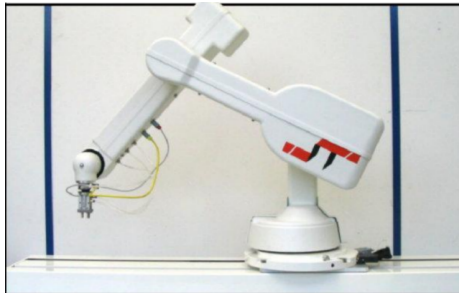
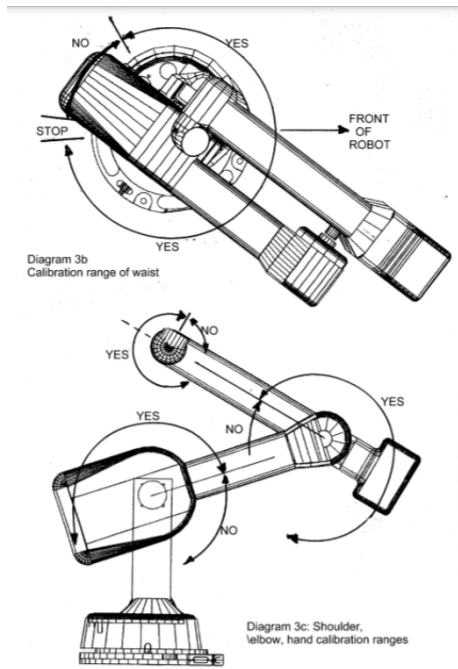


Figura 22: Robot R17 (grados de libertad [5]).



En el Cuadro 2, se observan los resultados de la clasificación utilizando diferentes combinaciones de características temporales para señales EMG y EEG. Se logró un rendimiento del 90 % con Redes Neuronales y del 80 % con Máquinas de Vectores de Soporte, demostrando la eficacia de la combinación de señales biomédicas para mejorar el control de sistemas robóticos.

Cuadro 2: Comparación de resultados de entrenamiento con diferentes combinaciones y cantidad de características temporales. [5].

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	<b>EEG:</b> HA,NFD,HC,KUR, ME,SK,SE,HM,PCA <b>EMG:</b> ZC,MAV,KUR,AE, VAR,AW,SMR	Tilt-right, Turn-left, Palm-Up, Fist-Up	<b>89.5 %</b>
Árbol de decisión Fino	<b>EEG:</b> HA,NFD,HC,KUR, ME,SK,SE,HM,PCA <b>EMG:</b> ZC,MAV,KUR,AE, VAR,AW,SMR	Tilt-right, Turn-left, Palm-Up, Fist-Up	<b>86.6 %</b>
Medium KNN	<b>EEG:</b> HA,NFD,HC,KUR, ME,SK,SE,HM,PCA <b>EMG:</b> ZC,MAV,KUR,AE, VAR,AW,SMR	Tilt-right, Turn-left, Palm-Up, Fist-Up	<b>87.1 %</b>
Discriminante Linear	<b>EEG:</b> HA,NFD,HC,KUR, ME,SK,SE,HM,PCA <b>EMG:</b> ZC,MAV,KUR,AE, VAR,AW,SMR	Tilt-right, Turn-left, Palm-Up, Fist-Up	<b>84.9 %</b>
Bagged Trees	<b>EEG:</b> HA,NFD,HC,KUR, ME,SK,SE,HM,PCA <b>EMG:</b> ZC,MAV,KUR,AE, VAR,AW,SMR	Tilt-right, Turn-left, Palm-Up, Fist-Up	<b>90.5 %</b>

Además, se diseñó una interfaz recolectora de datos, mostrada en la Figura 23, y una interfaz clasificadora para señales EEG y EMG, representada en la Figura 24, que permitieron un control más efectivo del robot.

Figura 23: Interfaz recolectora de Datos [5].

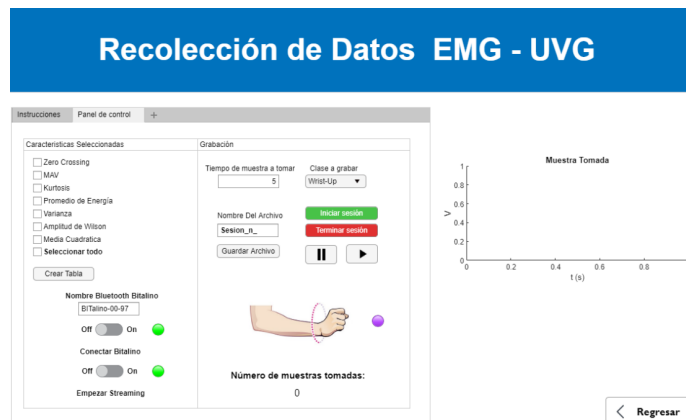
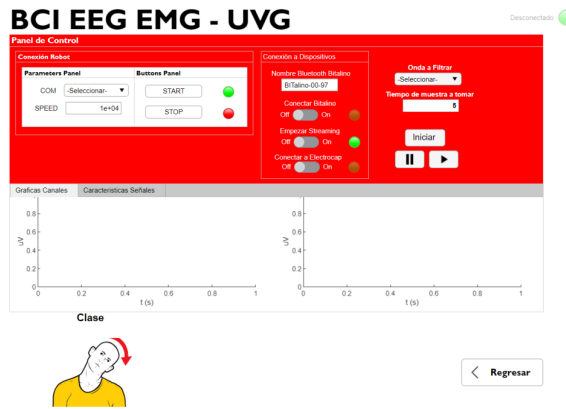


Figura 24: Interfaz clasificadora EEG-EMG [5].



Los movimientos evaluados en la base de datos utilizada se detallan en las Figuras ?? y 26, proporcionando un contexto sobre los gestos procesados.

Figura 25: Movimientos realizados en la base de datos [5].

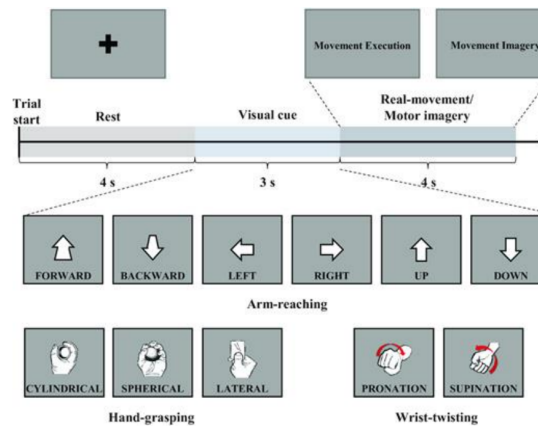
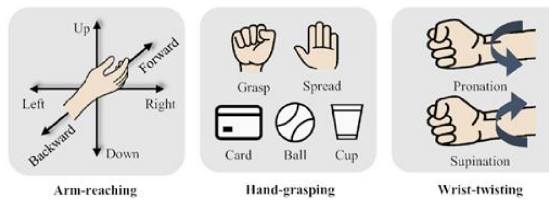


Figura 26: Movimientos realizados por los sujetos [5].



El trabajo de graduación alcanzó la implementación de una interfaz combinada para señales EMG y EEG, logrando un rendimiento alto en la clasificación de movimientos mediante simulaciones. A pesar de las limitaciones impuestas por la pandemia de COVID-19, se demostró la viabilidad de utilizar estas señales combinadas para futuros estudios en el control físico con dispositivos biomédicos.

A lo largo de los años, el desarrollo tecnológico ha permitido que la interacción entre los humanos y las máquinas avance significativamente, facilitando la creación de sistemas más intuitivos y adaptables. En particular, el estudio de señales biomédicas, como las señales electromiográficas (EMG) y electrooculográficas (EOG), ha abierto nuevas oportunidades para el control de dispositivos robóticos, permitiendo su manejo en tiempo real mediante la interpretación de actividades musculares y movimientos oculares.

Este proyecto busca implementar una interfaz biomédica que utilice señales EMG y EOG para controlar robots móviles de manera precisa y eficiente, superando las limitaciones de trabajos previos que se centraron principalmente en simulaciones. Mientras que las señales EMG permiten captar de forma específica los patrones musculares, las señales EOG aportan una referencia adicional basada en los movimientos oculares. Esta combinación permite una interacción más robusta y versátil en entornos controlados.

Las señales EOG, que inicialmente se capturan como señales analógicas, se convierten a digitales para integrarse al sistema de control robótico, eliminando la necesidad de algoritmos complejos como el aprendizaje automático. Este enfoque simplificado no solo facilita la implementación, sino que también garantiza respuestas rápidas y fiables en situaciones dinámicas. Los movimientos detectados a través de las señales EOG, como desplazamientos horizontales y verticales, se tradujeron directamente en comandos digitales para maniobras como giros, avances y detenciones. Este método asegura que el sistema sea capaz de operar en tiempo real con una robustez adecuada para superar obstáculos y responder eficientemente a las condiciones del entorno.

El desarrollo de este sistema también aborda la necesidad de llevar las interfaces biomédicas más allá de las simulaciones, validándolas en sistemas físicos que reflejen aplicaciones prácticas. Este enfoque no solo permite evaluar la eficacia de la interfaz en escenarios reales, sino que también sienta las bases para futuras investigaciones orientadas a optimizar la interacción humano-robot a través de señales biomédicas. Además, este trabajo ofrece una plataforma para explorar nuevas posibilidades en el diseño de sistemas robóticos que combinen señales múltiples para lograr un control más preciso y adaptable.

Este proyecto representa un avance significativo en el uso de señales EMG y EOG para el control robótico, ofreciendo una solución innovadora y eficiente para la interacción en tiempo real con sistemas móviles. Su implementación en un ambiente controlado asegura la viabilidad y la eficacia del sistema, estableciendo un marco para futuros desarrollos en robótica y procesamiento de señales biomédicas.

### 4.1. Objetivo general

Diseñar e implementar una interfaz para el control en tiempo real de agentes robóticos móviles, por medio de señales bioeléctricas.

### 4.2. Objetivos específicos

- Desarrollar una interfaz que permita la detección, muestreo y procesamiento de señales bioeléctricas, particularmente señales EOG y EMG.
- Desarrollar un algoritmo de control de robots móviles a partir de las señales bioeléctricas procesadas.
- Validar la interfaz y los algoritmos desarrollados por medio de simulaciones computarizadas.
- Validar la interfaz y los algoritmos desarrollados con agentes robóticos móviles disponibles en la UVG.

Este proyecto logró implementar un sistema funcional para el control de robots móviles mediante la integración de señales electromiográficas (EMG) y electrooculográficas (EOG). En el caso de las señales EOG, se logró la conversión de señales analógicas capturadas con el dispositivo biopac a digitales, permitiendo su uso en tiempo real para controlar movimientos específicos del robot. Por otro lado, se estableció una conexión efectiva entre el dispositivo bitalino y el robot, lo que permitió responder a comandos derivados de señales EMG en tiempo real.

El sistema alcanzó un margen de precisión del 90 % en la clasificación de señales EMG, demostrando la viabilidad de los algoritmos implementados para aplicaciones de control robótico en tiempo real. Estas conexiones se integraron en interfaces intuitivas desarrolladas para gestionar señales y controlar el robot de manera eficiente, facilitando su operación y análisis.

Sin embargo, el proyecto enfrentó algunas limitaciones importantes. En primer lugar, los experimentos se realizaron únicamente con un sujeto, limitando la generalización de los resultados obtenidos. Además, el uso de App Designer en Matlab introdujo demoras en el procesamiento de datos debido a la naturaleza de la aplicación, lo que impactó la eficiencia del sistema en ciertas tareas.

A pesar de estas limitaciones, este proyecto estableció una base sólida para la implementación de sistemas de control robótico basados en señales biomédicas. Los resultados obtenidos y las interfaces desarrolladas permiten identificar áreas clave de mejora, como la optimización del rendimiento de las interfaces y la validación del sistema con una muestra más amplia. Estos resultados abren la puerta para futuros desarrollos que amplíen las capacidades y la robustez del sistema.

### 6.1. Señales bioeléctricas

Las señales bioeléctricas son aquellas que se generan por la actividad eléctrica de los sistemas biológicos, como los músculos o el cerebro. Estas señales pueden ser medidas mediante electrodos, utilizando un método no invasivo aplicado en la superficie de la piel del área a analizar. Por ejemplo, al provocar la contracción muscular, se capturan las señales eléctricas que registran la actividad muscular. Posteriormente, estas señales se someten a procesos de filtrado para permitir un análisis más preciso, extrayendo las características relevantes que se utilizan en el reconocimiento de patrones [6].

### 6.2. Señales electromiográficas (EMG)

Este tipo de señales pueden medirse mediante electrodos, un método no invasivo aplicado en la superficie de la piel del área a analizar. De esta manera, al provocar la contracción muscular, se capturan las señales eléctricas que registran la actividad eléctrica. Luego, las señales se filtran para permitir un análisis más preciso, extrayendo las características que se utilizan en el reconocimiento de patrones [7].

### 6.3. Señales electroencefalográficas (EOG)

Electrooculograma (EOG) registra las variaciones de voltaje que ocurren con el movimiento angular del ojo. Inicialmente, se consideraba que estos potenciales reflejaban los potenciales de acción generados por los músculos responsables del movimiento ocular. Sin embargo, ahora se sabe que estos potenciales eléctricos son producto de la diferencia de potencial permanente existente entre la córnea y el fondo del ojo. Este método no invasivo

permite capturar con precisión los movimientos oculares y convertir estas señales en comandos utilizables para diversas aplicaciones, como el control de robots móviles y sistemas interactivos [8].

## 6.4. Características en el tiempo para las señales EMG

### 6.4.1. Media del valor absoluto (MAV)

Determina la amplitud promedio de una señal ya que proporciona una estimación del nivel de actividad muscular en un período de tiempo específico [9].

$$\text{MAV} = \frac{1}{N} \sum_{k=1}^N |x_k| \quad (1)$$

Donde:

- MAV es la Media del Valor Absoluto.
- $N$  es el número total de muestras.
- $x_k$  es el valor de la señal en la muestra  $k$ .
- $|\cdot|$  denota el valor absoluto.

### 6.4.2. Zero crossing

Determina el número de veces que la señal cambia de signo mientras se desplaza a través del tiempo [9].

$$\text{ZC} = \sum_{k=1}^{N-1} \text{sgn}(-emg_k \cdot emg_{k+1}) \quad (2)$$

Donde:

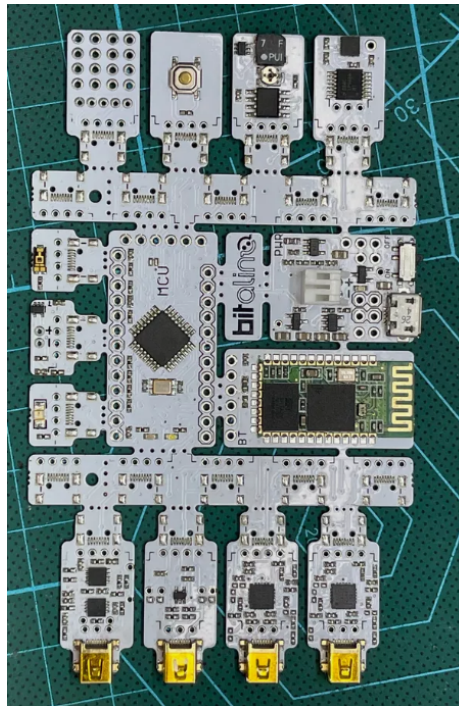
- $\text{ZC}$  es el número de cruces por cero de la señal.
- $N$  es el número total de muestras de la señal.
- $emg_k$  es el valor de la señal en la muestra  $k$ .
- $\text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{en otro caso} \end{cases}$  es la función signo.

## 6.5. Bitalino

El bitalino es una plataforma modular y de bajo costo diseñada para la adquisición de señales biomédicas. Permite capturar datos fisiológicos como señales electromiográficas (EMG), electrocardiográficas (ECG), electrooculográficas (EOG), entre otras, de manera sencilla y portátil. Está compuesto por módulos intercambiables que incluyen sensores, procesadores y módulos de comunicación, lo que facilita su integración en proyectos de investigación, prototipos y aplicaciones educativas.

Gracias a su capacidad para transmitir datos de forma inalámbrica mediante bluetooth y su compatibilidad con múltiples plataformas de desarrollo, el bitalino se ha consolidado como una herramienta accesible y versátil en el campo de la ingeniería biomédica y el diseño de interfaces hombre-máquina. La Figura 27 ilustra los módulos principales y la disposición de sus componentes, destacando su diseño compacto y funcionalidad [10].

Figura 27: Bitalino [10].

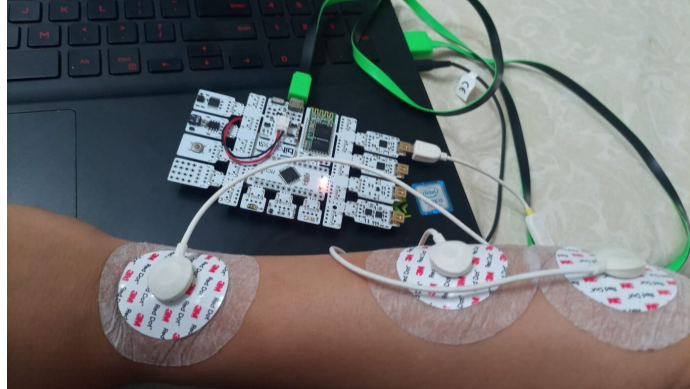


### 6.5.1. Ubicación de electrodos

El concepto de la distancia centro a centro del área conductiva de los electrodos, es la distancia que hay entre cada electrodo para la medición de señales EMG. Sin embargo, no es de vital importancia la forma del electrodo, ya que no afectará en los resultados; su tamaño se encuentra estandarizado entre 1 y 2 mm. En cuanto a la posición del sensor, se sugiere ubicarlo en el músculo de manera que esté alejado de los bordes que comparten con otras áreas musculares, permitiendo así una mayor distancia con respecto a otras áreas musculares [11].

La Figura 28 ilustra un ejemplo de la colocación de electrodos en un brazo para la captura de señales EMG. Esta disposición asegura una adquisición precisa y minimiza las interferencias provenientes de otras regiones musculares.

Figura 28: Ubicación de electrodos para señales EMG [10].



### 6.5.2. Registro de movimientos para el desarrollo de bases de datos de las señales EMG

Para la creación de una base de datos personalizada destinada al entrenamiento de modelos de aprendizaje automático, se definieron y ejecutaron movimientos específicos que permitieron capturar señales EMG asociadas a patrones bien diferenciados. Estos gestos representaron movimientos musculares y oculares controlados, necesarios para establecer comandos en sistemas de control robótico.

Se implementaron tres gestos principales, cada uno repetido entre 25 y 30 veces para garantizar una representación estadística robusta. Cada repetición tuvo una duración de un segundo, lo que permitió registrar señales limpias y consistentes. Estos datos fueron esenciales para construir un conjunto de entrenamiento representativo y fiable, optimizando la capacidad de los modelos para clasificar y reconocer patrones asociados a los movimientos.

## 6.6. Biopac

El biopac es un sistema diseñado para la adquisición, procesamiento y análisis de señales bioeléctricas, ampliamente utilizado en estudios biomédicos y educativos. Este dispositivo permite la captación de señales como electromiografía (EMG), electrooculografía (EOG) y electroencefalografía (EEG), brindando herramientas avanzadas para su análisis en tiempo real [8].

En la Figura 29, se muestra el dispositivo biopac MP36, utilizado para la captura de señales biomédicas. Este equipo cuenta con múltiples canales y conectores que permiten la adquisición simultánea de distintas señales, proporcionando un entorno versátil y eficiente para la investigación biomédica.

Figura 29: Dispositivo biopac MP36 [8].



Para la adquisición de señales EOG con el sistema biopac, los electrodos deben colocarse estratégicamente alrededor de los ojos para captar de manera efectiva los movimientos oculares. En la Figura 2, se muestra la configuración de los electrodos para capturar señales verticales y horizontales, asegurando mediciones precisas.

### 6.6.1. Ubicación de electrodos

Para la adquisición de señales EOG con el sistema biopac, los electrodos deben colocarse estratégicamente alrededor de los ojos para captar de manera efectiva los movimientos oculares. En la Figura 33, se muestra la configuración de los electrodos para capturar señales verticales y horizontales, asegurando mediciones precisas.

#### Electrodos verticales

Un electrodo se coloca encima del ojo derecho, en la frente. Otro electrodo se coloca debajo del ojo derecho, sobre la mejilla. Esta disposición permite registrar los movimientos verticales de los ojos, como mirar hacia arriba o hacia abajo [12].

#### Electrodos horizontales

Un electrodo se coloca en la parte externa del ojo derecho. Otro electrodo se coloca en la parte externa del ojo izquierdo. Esta configuración permite captar los movimientos horizontales, como mirar hacia la derecha o hacia la izquierda [12].

#### Electrodo de referencia (Tierra)

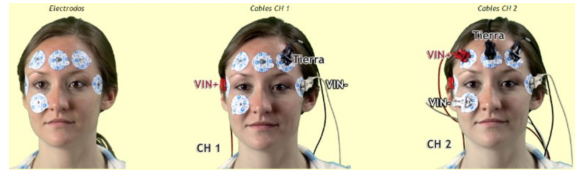
Este se coloca detrás de la oreja o en una zona neutra del cuerpo, como el lóbulo de la oreja. Su propósito es proporcionar una señal base para reducir el ruido y mejorar la calidad de la señal.

Además, los electrodos deben colocarse sobre piel limpia y raspada, aplicando gel conductor para mejorar la conductividad. Es crucial mantener la cabeza inmóvil durante el

registro, limitando el movimiento a los ojos, para obtener mediciones precisas [12].

En la Figura 30, se muestra la disposición completa de los electrodos para capturar señales EOG de manera efectiva.

Figura 30: Configuración de los electrodos para capturar señales EOG [12].



## 6.7. Conversión de señales analógicas a digitales con biopac

La conversión de señales analógicas a digitales es un proceso fundamental para analizar datos fisiológicos, ya que permite el procesamiento computacional de señales bioeléctricas capturadas mediante hardware especializado como el sistema biopac. A continuación, se describe el procedimiento utilizando el equipo MP36 y el software biopac Student Lab [13].

### 6.7.1. Configuración de hardware

Para realizar la conversión, se conecta el sistema biopac a través del puerto I/O del dispositivo MP36. El sistema emplea módulos adicionales como el STP35B para conectar dispositivos externos. Es importante utilizar cables específicos, como el DB25, para garantizar una conexión correcta. Además, el hardware digitaliza las señales mediante niveles lógicos estándar de 0 a 5 volts, como se muestra en la Figura 31.

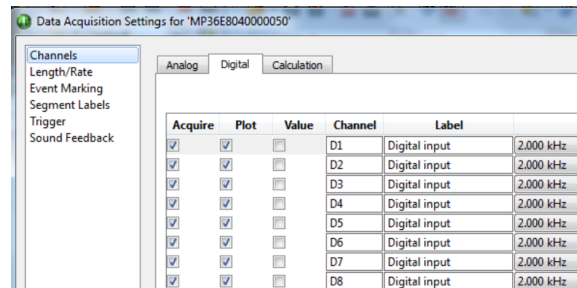
Figura 31: Puerto I/O del dispositivo MP36 [13].



### 6.7.2. Configuración de software

El software biopac student lab permite habilitar los canales digitales para la adquisición de datos. Desde el menú de configuración de adquisición de datos, se selecciona la pestaña de canales digitales y se habilita la adquisición de los ocho canales disponibles, como se muestra en la Figura 32. Esto asegura que las señales digitales se capturen de manera precisa para su posterior análisis.

Figura 32: Configuración de canales digitales en el software biopac student lab para la adquisición de señales [13].



El software biopac student lab permite habilitar los canales digitales para la adquisición de datos. Desde el menú de configuración de adquisición de datos, se selecciona la pestaña de canales digitales y se habilita la adquisición de los ocho canales disponibles. Esto asegura que las señales digitales se capturen de manera precisa para su posterior análisis [13].

## 6.8. Aprendizaje automático

El aprendizaje automático es una disciplina dentro de la inteligencia artificial que se enfoca en resolver problemas mediante la capacidad de aprender a partir de la experiencia. Este enfoque permite a los sistemas computacionales aplicar inferencias sobre información existente para construir representaciones apropiadas de aspectos relevantes de la realidad o procesos específicos. Una metáfora común en el campo describe el aprendizaje automático como la habilidad de reconocer situaciones problemáticas y reaccionar utilizando estrategias previamente aprendidas, mejorando su desempeño en un conjunto amplio de tareas a partir de la resolución de problemas previos [14].

### 6.8.1. Aprendizaje supervisado

El aprendizaje supervisado es una subcategoría del aprendizaje automático y de la inteligencia artificial que utiliza conjuntos de datos etiquetados para entrenar algoritmos. Estos algoritmos aprenden a clasificar datos o predecir resultados basándose en la relación entre las características de entrada y las etiquetas de salida, con el objetivo de generalizar sus predicciones a datos nuevos no vistos previamente [15].

### 6.8.2. Clasificador de máquinas lineales

## 6.9. Máquina lineal (Linear machine)

Una máquina lineal es un clasificador que utiliza funciones discriminantes lineales para separar datos en diferentes categorías. Es ampliamente utilizada en aprendizaje supervisado y análisis estadístico multivariante debido a su simplicidad y eficacia en tareas de clasifica-

ción. Matemáticamente, una máquina lineal define una o más funciones discriminantes de la forma:

$$g_i(x) = w_i^T x + w_{i0}, \quad i = 1, \dots, c$$

donde  $x$  es el vector de entrada que representa las características del dato a clasificar,  $w_i$  es el vector de pesos asociado a la clase  $i$ , definiendo la dirección de separación,  $w_{i0}$  es el término de sesgo o umbral que ajusta la posición del límite de decisión en el espacio, y  $c$  es el número de clases.

La máquina lineal opera dividiendo el espacio de características en regiones mediante hiperplanos definidos por la ecuación  $g(x) = 0$ . Estos hiperplanos actúan como límites de decisión que separan las clases. Un dato  $x$  se asigna a la clase  $i$  si  $g_i(x) > g_j(x)$  para todo  $j \neq i$ . En el caso de múltiples categorías, cada función discriminante define una clase, y el espacio se divide en regiones donde cada una corresponde a la clase cuya función discriminante tiene el valor más alto. En situaciones de empate, la clasificación se considera indefinida o requiere métodos adicionales para resolverla.

Entre las principales ventajas de las máquinas lineales se encuentra su simplicidad y rapidez, ya que los cálculos son computacionalmente eficientes, lo que las hace adecuadas para problemas con grandes volúmenes de datos. Además, sus límites de decisión lineales permiten una fácil interpretación, especialmente en espacios bidimensionales o tridimensionales. También funcionan bien en tareas donde las clases son linealmente separables. Sin embargo, estas máquinas tienen limitaciones, como su incapacidad para resolver problemas no linealmente separables y su sensibilidad a datos ruidosos o valores atípicos, lo que puede afectar significativamente su rendimiento [16].

### 6.9.1. Análisis discriminante lineal (LDA)

El Análisis Discriminante Lineal (LDA) es una técnica estadística que se emplea principalmente para la reducción de dimensionalidad y la clasificación de patrones. Su objetivo principal es encontrar una combinación lineal de las características de un conjunto de datos que permita separar de manera óptima las distintas clases de objetos o eventos. Esta técnica se utiliza ampliamente en aprendizaje automático para mejorar la eficiencia de los modelos de clasificación y como un paso previo en el procesamiento de datos.

El propósito central del LDA es proyectar los datos originales, que pueden tener muchas dimensiones, sobre un espacio de menor dimensión, maximizando la separación entre clases mientras se minimiza la dispersión dentro de cada clase. Este proceso permite representar los datos de manera que las clases sean más fácilmente separables, lo que facilita la clasificación de nuevos puntos [17].

## Proceso del LDA

El proceso de LDA se lleva a cabo en varios pasos estructurados:

1. **Cálculo de la varianza entre clases ( $S_B$ ):** el primer paso consiste en calcular la varianza entre clases, lo que implica medir la distancia entre las medias de las distintas clases para evaluar su separabilidad. Esta varianza proporciona información sobre cuán diferentes son las clases en el espacio de características. Se calcula como la suma ponderada de las distancias entre las medias de las clases y la media global, lo que indica el grado de separación entre ellas.
2. **Cálculo de la varianza dentro de las clases ( $S_W$ ):** el segundo paso es calcular la varianza dentro de las clases, que mide la dispersión de las muestras dentro de cada clase. Este valor refleja cuán homogéneas son las muestras dentro de cada clase. Cuanto menor sea la varianza dentro de la clase, más fácil será separar las clases. La varianza dentro de las clases se obtiene sumando las distancias de cada muestra a la media de su respectiva clase.
3. **Construcción del espacio reducido:** una vez calculadas ambas varianzas, el siguiente paso consiste en construir un espacio dimensional reducido que maximice la varianza entre las clases y minimice la varianza dentro de cada clase. Para lograr esto, se calcula la matriz de transformación  $W$ , que define los coeficientes de discriminación. Este espacio reducido es crucial para proyectar los datos de manera que las clases sean fácilmente separables, optimizando así la clasificación.
4. **Cálculo de la función discriminante lineal ( $g(x)$ ):** finalmente, se calcula la función discriminante lineal  $g(x)$ , que se utiliza para clasificar nuevas observaciones. Esta función es una combinación lineal de las características del nuevo punto de datos y los coeficientes obtenidos en los pasos anteriores. Si el valor de la función discriminante es mayor que un umbral determinado, la observación se clasifica en una clase; de lo contrario, se asigna a la otra clase.

### 6.9.2. Aprendizaje bayesiano (*Bayesian learning*)

El aprendizaje bayesiano es un enfoque probabilístico que se basa en la aplicación del Teorema de Bayes para realizar inferencias sobre hipótesis, a partir de datos observados y conocimiento previo. En este marco, cada ejemplo de entrenamiento observado incrementa o decrementa la probabilidad estimada de que una hipótesis sea correcta. Este método proporciona una mayor flexibilidad en comparación con algoritmos determinísticos que eliminan hipótesis inconsistente con un solo ejemplo.

El aprendizaje bayesiano permite la integración de conocimiento previo, representado por una probabilidad previa ( $P(h)$ ) asignada a cada hipótesis candidata, y una distribución de probabilidad de los datos observados dado que una hipótesis es verdadera ( $P(D|h)$ ). De esta manera, las hipótesis que hacen predicciones probabilísticas pueden ser incorporadas en el modelo. La clasificación de nuevas instancias se logra combinando las predicciones de múltiples hipótesis, ponderadas por sus probabilidades posteriores ( $P(h|D)$ ) [18].

## Teorema de bayes

El Teorema de bayes se utiliza para calcular la probabilidad posterior de una hipótesis ( $P(h|D)$ ) dada una probabilidad previa  $P(h)$ , una probabilidad de los datos observados  $P(D|h)$ , y una probabilidad de los datos  $P(D)$ . Matemáticamente, el Teorema de bayes se expresa como:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Aquí:

- $P(h)$ : Probabilidad previa de la hipótesis  $h$ , antes de observar los datos.
- $P(D|h)$ : Probabilidad de observar los datos  $D$  bajo la hipótesis  $h$ .
- $P(D)$ : Probabilidad de los datos observados.
- $P(h|D)$ : Probabilidad posterior de  $h$ , que refleja la confianza en  $h$  después de observar  $D$ .

## Características del aprendizaje bayesiano

El aprendizaje bayesiano se caracteriza por:

- La habilidad de combinar datos observados con conocimiento previo para estimar la probabilidad de hipótesis.
- Acomodar hipótesis que hacen predicciones probabilísticas.
- Proveer un estándar óptimo para la toma de decisiones bajo incertidumbre.

Sin embargo, este enfoque puede ser computacionalmente costoso, especialmente en espacios de hipótesis grandes. En casos prácticos, los métodos bayesianos como el clasificador Naive Bayes ofrecen una implementación eficiente basada en suposiciones simplificadas, como la independencia condicional entre atributos. El aprendizaje bayesiano se aplica en una amplia variedad de campos, como el diagnóstico médico, la clasificación de textos, el reconocimiento de patrones y la predicción en sistemas dinámicos. Proporciona una base teórica sólida para modelos probabilísticos y métodos como el clasificador Naive Bayes, que es particularmente útil para tareas de clasificación.

### 6.9.3. Clasificador de máquinas de vectores de soporte (SVM)

Las Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés) son un método de aprendizaje supervisado diseñado principalmente para problemas de clasificación y regresión. Su fundamento radica en encontrar un hiperplano óptimo que separe las clases de manera

lineal o no lineal. Este hiperplano maximiza el margen entre las clases, utilizando los puntos más cercanos al límite como "vectores de soporte". Esto permite un buen desempeño de generalización y robustez frente a datos ruidosos o casos atípicos [19].

## Hiperplanos y márgenes

Los hiperplanos son subespacios de dimensión  $n - 1$  en un espacio de características de dimensión  $n$ , utilizados como frontera de decisión en las Máquinas de Vectores de Soporte (SVM). Los márgenes representan la distancia mínima entre el hiperplano y los vectores de soporte más cercanos de cada clase. El objetivo principal de una SVM es maximizar este margen, lo que mejora la capacidad de generalización del modelo [20].

## Características del hiperplano óptimo

El hiperplano óptimo en las SVM cumple con las siguientes características:

- Maximiza el margen  $\tau$ .
- Es equidistante de los vectores de soporte de ambas clases.
- Garantiza una mejor separación entre clases, lo que minimiza la posibilidad de errores.

El margen se calcula utilizando la fórmula:

$$\tau = \frac{1}{\|w\|},$$

donde  $w$  es el vector normal al hiperplano.

## Ejemplo gráfico

La Figura 33 muestra una representación de los hiperplanos de separación en un espacio bidimensional. Además, la Figura 34 ilustra la diferencia entre un margen no óptimo y el margen máximo alcanzado con el hiperplano óptimo.

Figura 33: Hiperplanos de separación en un espacio bidimensional [20].

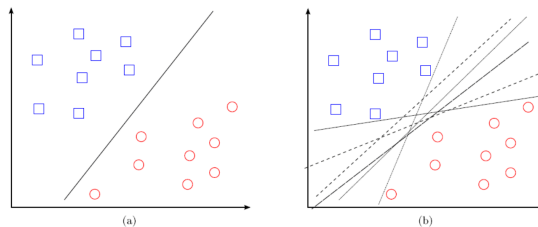
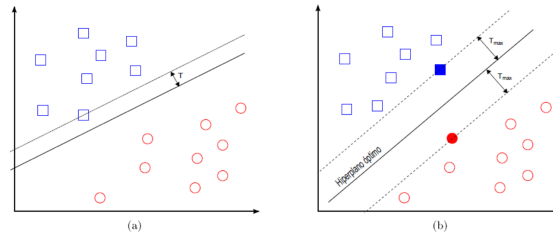


Figura 34: Margen asociado a un hiperplano no óptimo frente a un hiperplano óptimo [20].

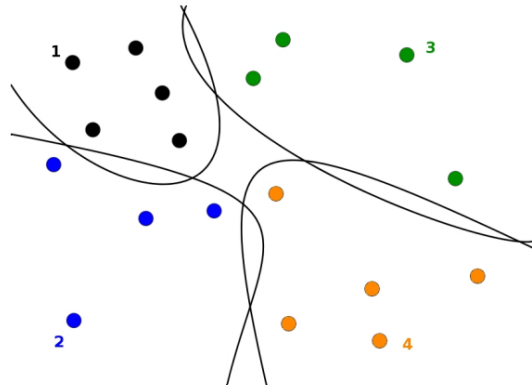


### Clasificación multiclase

- One-vs-All (OvA):** en la estrategia One-vs-All, se entrena una SVM para cada clase, separando los datos de las demás. Durante la fase de predicción, la SVM asigna una puntuación a cada clase y la clase con la puntuación más alta es seleccionada como la predicción final. Este método es sencillo de implementar y computacionalmente eficiente.

La Figura 35 muestra una representación gráfica de esta estrategia, donde una clase es separada del resto mediante un hiperplano específico [21].

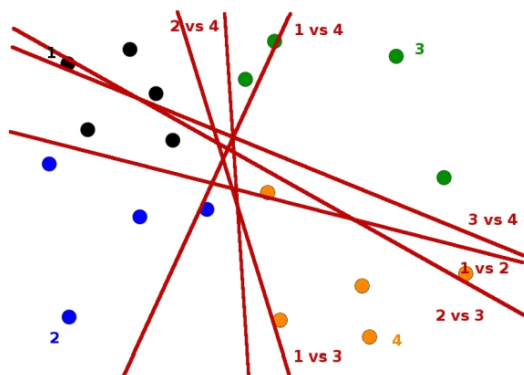
Figura 35: Estrategia One-vs-All para clasificación multiclase [21].



- One-vs-One (OvO):** en la estrategia One-vs-One, se entrena una SVM para cada par de clases posible, resultando en  $C(C - 1)/2$  clasificadores, donde  $C$  es el número total de clases. Durante la clasificación, se utiliza una votación mayoritaria entre todos los clasificadores para determinar la clase final. Este método puede ser más preciso, aunque requiere mayor tiempo de entrenamiento debido al número de clasificadores generados.

La Figura 36 ilustra esta estrategia, mostrando la separación entre múltiples pares de clases mediante varios hiperplanos [21].

Figura 36: Estrategia One-vs-One para clasificación multiclase [21].



## Funciones kernel

Las funciones kernel permiten a las máquinas de vectores de soporte (SVM) transformar los datos de entrada en un espacio de características de mayor dimensión donde las clases pueden ser separables de manera más efectiva. Este método es útil para problemas donde los datos no son linealmente separables en el espacio original. Las funciones kernel más comunes incluyen:

- **Kernel lineal:** Apropiado para datos que son linealmente separables. Representa un caso simple donde no se realiza transformación adicional del espacio de características. La función se define como  $K(x, z) = x^T z$  [22].
- **Kernel polinómico:** Permite capturar relaciones más complejas entre las variables de entrada. Se define como  $K(x, z) = (x^T z + c)^d$ , donde  $c$  es una constante y  $d$  es el grado del polinomio. Este kernel es útil para problemas donde las relaciones no son lineales pero pueden modelarse mediante polinomios [22].
- **Kernel gaussiano:** También conocido como kernel de base radial (RBF), es ideal para problemas donde las fronteras de decisión son no lineales. Su fórmula es  $K(x, z) = \exp(-\gamma \|x - z\|^2)$ , donde  $\gamma$  es un parámetro ajustable. Este kernel transforma los datos en un espacio de características infinito, proporcionando flexibilidad para capturar patrones complejos [22].

### 6.9.4. Redes neuronales artificiales

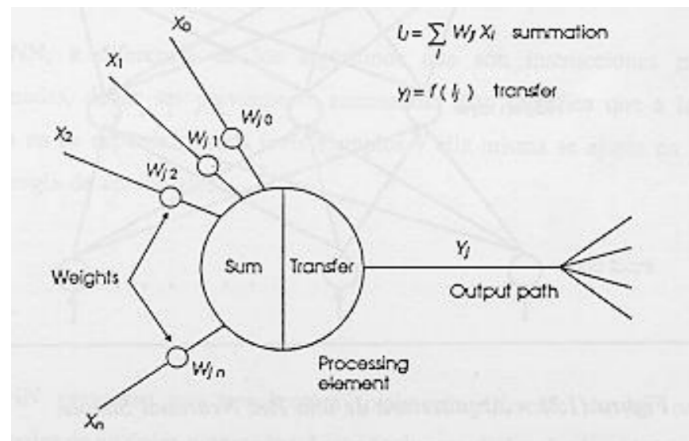
Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés) están inspiradas en el funcionamiento de las neuronas biológicas. El elemento análogo en las ANN es el elemento procesador. Este elemento tiene múltiples entradas y combina los valores mediante una suma ponderada. Posteriormente, la suma es transformada mediante una función de transferencia, que aplica una transformación no lineal antes de producir la salida del PE.

La salida del elemento procesador puede conectarse con las entradas de otros PE mediante conexiones ponderadas, las cuales representan la eficacia sináptica de las conexiones

neuronales. Estas conexiones permiten la propagación de la información dentro de la red, facilitando tareas como la clasificación y el reconocimiento de patrones.

En la Figura 37, se ilustra un elemento procesador típico de una red neuronal artificial, destacando las entradas, la suma ponderada, la función de transferencia y la salida resultante [23].

Figura 37: Representación de un elemento procesador en una red neuronal artificial [23].



### Arquitectura de una red neuronal simple

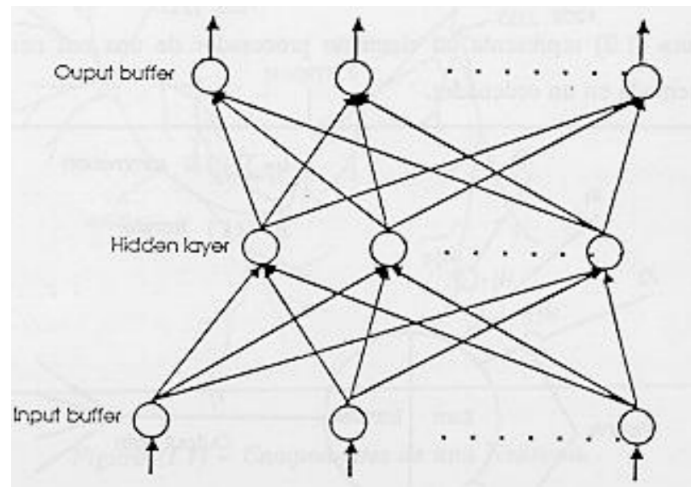
Una red neuronal está formada por un conjunto de unidades elementales, conocidas como elementos procesadores (PE), conectadas de manera estructurada. Estas conexiones no solo son relevantes por el modelo individual de los elementos PE, sino también por la forma en la que interactúan dentro de la red. Generalmente, los PE se organizan en niveles o capas:

- **Capa de entrada:** También denominada buffer de entrada, es donde los datos son introducidos al sistema.
- **Capas ocultas:** Estas capas procesan la información de manera interna, ajustando los pesos y transfiriendo los datos hacia las siguientes capas.
- **Capa de salida:** También conocida como buffer de salida, proporciona los resultados procesados por la red neuronal. Las conexiones entre estas capas son secuenciales y típicamente unidireccionales, fluyendo desde la entrada hacia la salida. La Figura 38 ilustra una red neuronal simple con una única capa oculta y conexiones entre nodos de capas consecutivas.

### Redes neuronales multicapa (MLP - multilayer perceptron)

Las redes neuronales multicapa (*Multilayer Perceptron*, *MLP*) son una generalización de las redes neuronales simples. A diferencia de las redes de una sola capa, las MLP incluyen múltiples capas ocultas que permiten que la red aprenda representaciones más complejas de

Figura 38: Arquitectura de una red neuronal simple [23].



los datos. La capacidad de una red neuronal para aprender relaciones no lineales entre las entradas y las salidas depende en gran medida de la cantidad y configuración de las capas ocultas.

### Número de Neuronas en las Capas Ocultas

El número de neuronas en las capas ocultas es uno de los aspectos clave en la configuración de una red neuronal. En redes con 2 o 3 nodos en las capas ocultas, cada capa realiza transformaciones no lineales sobre los datos, lo que permite modelar relaciones complejas entre las características. Sin embargo, es importante equilibrar el número de neuronas para evitar problemas de *overfitting* (sobreajuste) o *underfitting* (subajuste).

- **Redes con 2 o 3 Neuronas en las Capas Ocultas:** Estas configuraciones suelen ser efectivas para problemas más simples o cuando el conjunto de datos no requiere una representación compleja. La cantidad de neuronas en cada capa determina el poder de representación de la red. Con 2 o 3 neuronas, la red tiene una capacidad limitada para aprender patrones complejos, lo que puede ser adecuado para problemas con menos variabilidad en los datos.

En el caso de un conjunto de datos con características simples, como en la clasificación de carros o defectos en autos, el uso de redes con pocas neuronas en las capas ocultas puede ser suficiente para obtener una precisión aceptable sin introducir la complejidad innecesaria de redes más grandes.

### Ejemplo de Configuración en Redes Neuronales

En el caso de la base de datos de *Evaluación de carros*, la red está configurada con:

- 6 neuronas en la capa de entrada (correspondientes a las características de los datos),
- 3 capas ocultas con diferentes números de neuronas en cada una, y

- 4 neuronas en la capa de salida (correspondientes a las clases).

Los resultados de clasificación fueron satisfactorios con una exactitud del 68 % al 74 %, lo que indica que una red con un número moderado de neuronas ocultas puede ser efectiva para la clasificación en problemas de complejidad media.

En otro ejemplo, la base de datos de *Defectos en autos* utilizó redes con 4 neuronas de entrada, 3 capas ocultas (con un número variable de neuronas) y 3 neuronas de salida, obteniendo una exactitud del 55 %. En este caso, el rendimiento sugiere que la complejidad del problema podría haber excedido la capacidad de una red con 2 o 3 nodos en las capas ocultas.

### Consideraciones al Utilizar Redes con Pocas Neuronas en las Capas Ocultas

Cuando se utilizan redes neuronales con 2 o 3 nodos en las capas ocultas, es fundamental considerar lo siguiente:

- **Complejidad del Problema:** Redes con pocas neuronas pueden ser adecuadas para problemas con relaciones simples, pero para problemas complejos o con datos ruidosos, una mayor cantidad de neuronas en las capas ocultas podría ser necesaria.
- **Capacidad de Generalización:** El número de neuronas en la capa oculta influye en la capacidad de la red para generalizar patrones. Si el número de neuronas es muy pequeño, la red podría no ser capaz de aprender las representaciones adecuadas (*underfitting*). Si el número de neuronas es excesivamente alto, la red podría aprender en exceso los detalles del conjunto de datos de entrenamiento (*overfitting*).

### 6.9.5. Clasificador *K-Nearest Neighbors* (KNN)

El algoritmo *K-Nearest Neighbors* (KNN) es una técnica no paramétrica ampliamente utilizada en problemas de clasificación y regresión. Su premisa básica es asignar a un nuevo punto de datos la clase más frecuente entre las  $K$  observaciones más cercanas en el espacio de características. Este enfoque intuitivo y fácil de implementar se ha extendido a diversas variantes, cada una adaptada para mejorar el desempeño en problemas específicos, como el manejo de valores perdidos o la incorporación de información contextual en la distancia entre puntos [24].

#### KNN estándar

El clasificador KNN estándar asigna la clase de una nueva muestra basándose exclusivamente en la frecuencia de las clases presentes en los  $K$  vecinos más cercanos, calculados mediante una métrica de distancia, comúnmente la distancia euclidiana [24].

Este método asume que:

- Todos los vecinos contribuyen por igual a la decisión de clasificación.

- Las características son igualmente relevantes para determinar la cercanía.

Aunque es efectivo en problemas simples y balanceados, el enfoque estándar presenta limitaciones en casos donde los puntos más cercanos no son igualmente representativos debido a la distribución de los datos o a su proximidad relativa.

### **KNN Ponderado por distancia**

El clasificador KNN ponderado por distancia introduce una mejora significativa al considerar que los vecinos más cercanos tienen una mayor influencia en la decisión de clasificación. Para lograr esto, asigna un peso inversamente proporcional a la distancia entre el punto de consulta y cada vecino. La clase final se determina sumando los pesos de los vecinos para cada categoría y seleccionando aquella con la suma mayor [24].

La función de ponderación puede definirse, por ejemplo, como:

$$\text{Peso} = \frac{1}{\text{Distancia}^2}$$

Este método resulta particularmente útil cuando los vecinos cercanos tienen más probabilidad de pertenecer a la misma clase que el punto en cuestión, aumentando la precisión en problemas con fronteras de decisión complejas o datos desbalanceados.

### **KNN Adaptativo**

El clasificador KNN adaptativo, como el *MI-KNN* (Mutual Information KNN), utiliza información contextual adicional, como la relevancia de las características, para mejorar la clasificación. Este enfoque calcula la **información mutua** (*IM*) entre cada característica y la clase objetivo para evaluar su importancia relativa [24].

La IM mide la dependencia entre dos variables y se utiliza como un peso adicional en la métrica de distancia:

- $\alpha_j = I(x_j, t)$  es la información mutua entre la característica  $x_j$  y la clase objetivo  $t$ .
- $d_j(x_a, x_b)$  es la distancia entre los valores de la característica  $j$  para los puntos  $a$  y  $b$ .

La clave del enfoque adaptativo es que no todas las características tienen la misma relevancia para la clasificación. Al ajustar las distancias de acuerdo con esta relevancia, el modelo mejora su capacidad para clasificar correctamente los datos, especialmente en conjuntos de datos complejos o incompletos.

## 6.10. Robot pololu+3 pi

El robot Pololu 3Pi+ es un robot móvil de alto rendimiento y programable por el usuario, destacando por su versatilidad. Equipado con un microcontrolador ATmega32UA AVR de Microchip y una interfaz USB, este robot viene precargado con un arranque compatible con Arduino, lo que facilita su programación y personalización. Además, el 3Pi+ cuenta con dos controladores de puentes H y una amplia gama de sensores integrados, como codificadores de cuadratura para un control preciso del motor, una unidad de medida inercial completa, y sensores de reflectancia para seguimiento de línea o detección de bordes, así como sensores de impacto en su frente. Su interfaz incluye tres botones para la entrada del usuario y una pantalla OLED gráfica de 128 x 64, un zumbador y LEDs indicadores que proporcionan retroalimentación. La Figura 39 muestra un modelo del robot Pololu 3Pi+ en su configuración estándar [25].

Figura 39: Modelo del robot Pololu 3Pi+ en su configuración estándar [25].



## 6.11. Interfaz hombre-máquina (HMI)

Las Interfaces Hombre-Máquina (HMI) son sistemas que actúan como un medio de comunicación entre los operadores humanos y los sistemas automatizados. Estas interfaces permiten a los usuarios interactuar de manera eficiente con sistemas mecánicos, eléctricos y digitales para supervisar, controlar y optimizar procesos.

Las HMI han evolucionado desde simples paneles con interruptores hasta sistemas avanzados con pantallas táctiles, sensores y software que ofrecen retroalimentación visual en tiempo real. Esto las convierte en herramientas indispensables en aplicaciones industriales y robóticas, mejorando la productividad, la seguridad y la toma de decisiones [26].

## 6.12. Comunicación entre agentes físicos e interfaces de usuario

La comunicación entre un robot y una aplicación móvil, que representa la interfaz gráfica de usuario (GUI), es un proceso mutuo de intercambio de datos y comandos. Este intercambio permite al usuario controlar el robot y recibir información sobre su estado y resultados a través de la aplicación móvil. La comunicación se establece típicamente mediante tecnologías

como Bluetooth, Wi-Fi o USB, facilitando así la interacción intuitiva y remota entre el usuario y el robot para diversas aplicaciones y propósitos [27].

---

## Adquisición de señales EMG

---

En este capítulo se aborda el proceso de adquisición de señales EMG través del sistema bitalino. Además, se incluye un análisis del filtrado y procesamiento de las señales capturadas para mejorar su calidad y preparar los datos para su posterior análisis.

### 7.1. Obtención de las señales con el bitalino

El proceso de reconocimiento de señales se basa en la captura y procesamiento de señales provenientes del bitalino. Este sistema tiene como objetivo segmentar y analizar señales EMG (Electromiografía) en tiempo real, para identificar gestos específicos basados en métricas calculadas a partir de la señal.

El funcionamiento general del sistema se describe a continuación:

- **Configuración inicial:** Se configura el bitalino y se inicializan las variables necesarias para la adquisición y procesamiento de datos.
- **Adquisición de datos:** Las señales se capturan a través de los sensores del bitalino y se convierten a voltaje para su procesamiento.
- **Filtrado:** Se aplican filtros pasa banda y de rechazo de banda (Notch) para eliminar el ruido, como interferencias de la red eléctrica, y optimizar la calidad de la señal.
- **Cálculo de métricas:** Se calcula la Media Absoluta de Valor (MAV) de la señal filtrada, utilizada como criterio para detectar movimientos relevantes. Este cálculo permite identificar cambios significativos en la señal, como se ilustra en la Figura 41.

- **Segmentación de señales:** Una vez detectado un movimiento, la señal se divide en tres segmentos clave, definidos como:
  - **Segmento 1:** Representa la actividad previa al gesto y se utiliza como referencia para establecer el nivel base de ruido.
  - **Segmento 2:** Corresponde al periodo activo del gesto, donde se observan los picos de actividad muscular. Este segmento es crítico para el cálculo de características como la Media absoluta de valor (MAV) y los Cruces por Cero (ZC).
  - **Segmento 3:** Representa la actividad posterior al gesto y permite verificar la estabilización de la señal después del movimiento.

La segmentación y el análisis de cada uno de estos periodos pueden observarse en la Figura 40.

- **Cálculo de características:** Durante el análisis de cada segmento, se calculan características como MAV, Cruces por Cero (ZC) y la duración de la señal.
- **Almacenamiento de datos:** Las características calculadas se guardan en un archivo Excel para futuras evaluaciones o entrenamientos de modelos de aprendizaje automático.
- **Visualización en tiempo real:** Se genera una representación gráfica tanto de la señal en tiempo real como de los segmentos detectados, lo que facilita la supervisión del sistema.

Figura 40: Segmentación de la señal en tres partes identificando cruces por cero.

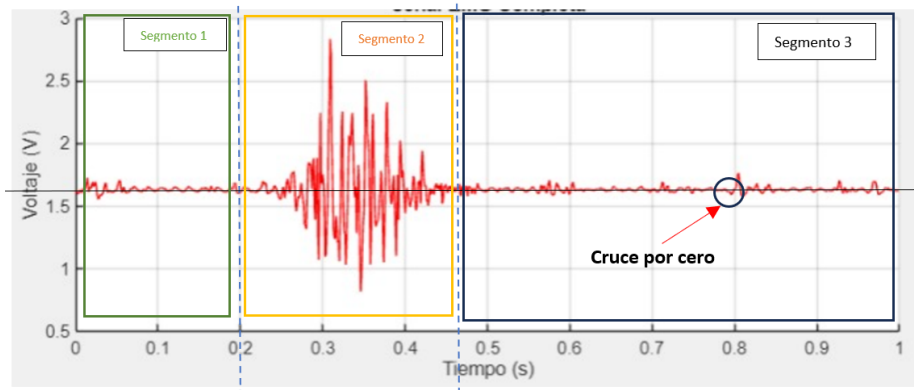
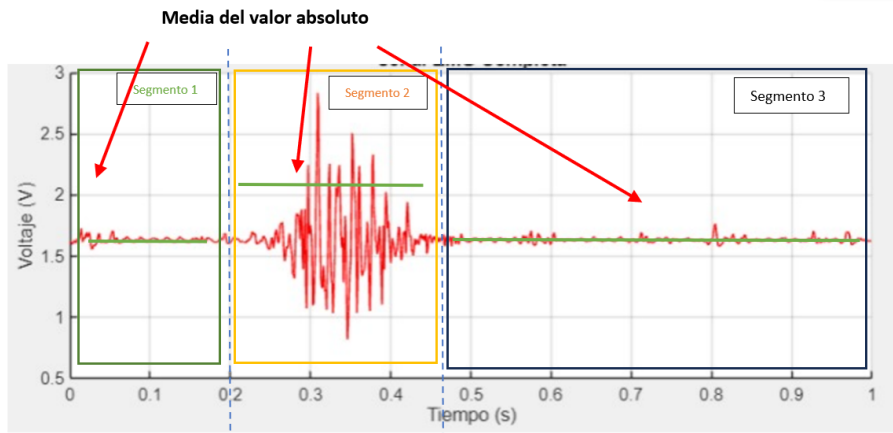


Figura 41: Cálculo de la media absoluta de valor (MAV) para cada segmento.

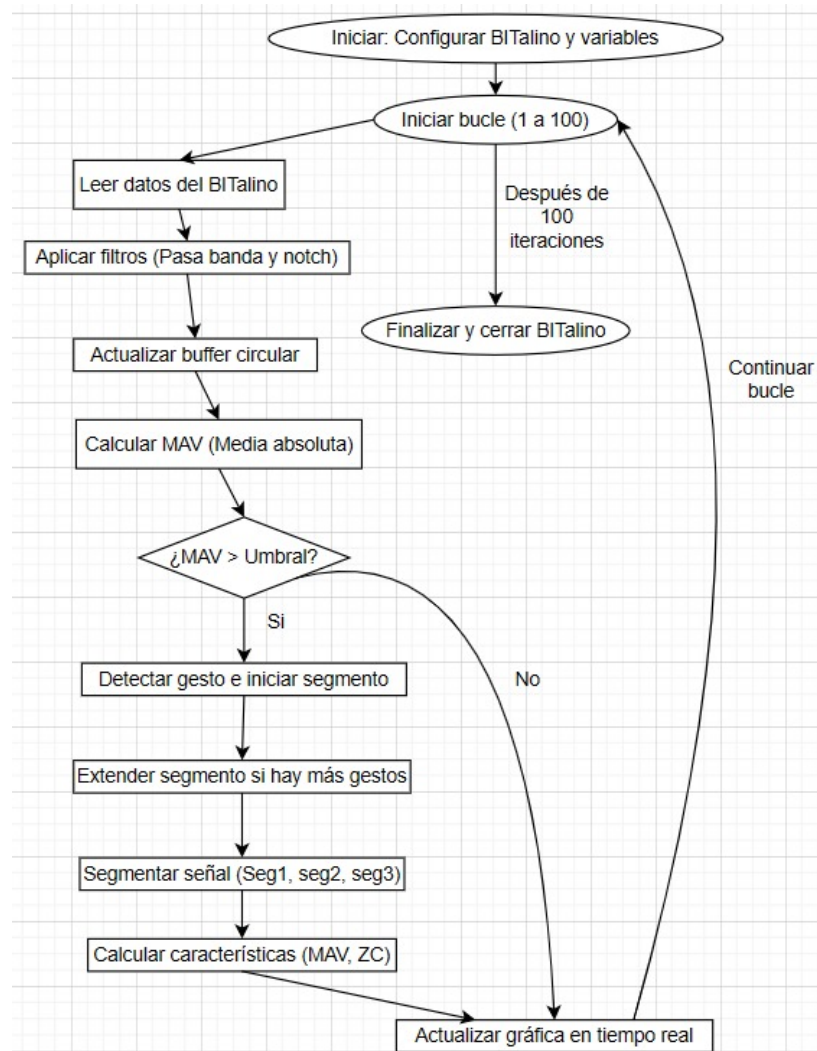


### 7.1.1. Desarrollo del algoritmo de lectura

Para el desarrollo del algoritmo de lectura de señales EMG con el bitalino, se empleó el comando `bitalino()` del bitalino Toolbox en Matlab, el cual permite establecer la conexión con el dispositivo especificando parámetros como la tasa y el tiempo de muestreo de esta forma ajustando la resolución temporal de las señales capturadas.

El diagrama de flujo de la figura 47 ilustra cada una de estas etapas de manera clara y detallada.

Figura 42: Diagrama de flujo del reconocimiento de señales EMG y segmentación.



Este flujo asegura que el sistema pueda procesar los datos en tiempo real, detectar gestos de manera eficiente y almacenar las características para análisis posteriores. La implementación del flujo permite una interacción robusta con el dispositivo bitalino y facilita la extracción de datos relevantes para aplicaciones en interfaces hombre-máquina (HMI) o en sistemas de control robótico.

### 7.1.2. Generación de base de datos propia

Para las siguientes pruebas de entrenamiento, se creó una base de datos propia. Para crear esta base de datos se obtuvieron las señales EMG utilizando el algoritmo de adquisición de señales descrito en el capítulo 7.1. Las señales captadas corresponden a 3 movimientos, descritos en las Figuras 43, 44 y la 45. Se realizaron repetidamente entre 30 a 25 grabaciones de 1 segundo para cada movimiento.

Figura 43: Primer movimiento incluido en la base de datos: apretar y levantar la muñeca una vez.



Figura 44: Segundo movimiento movimiento incluido en la base de datos: levantar cada dedo creando un arco hasta cerrar el puño con suavidad.

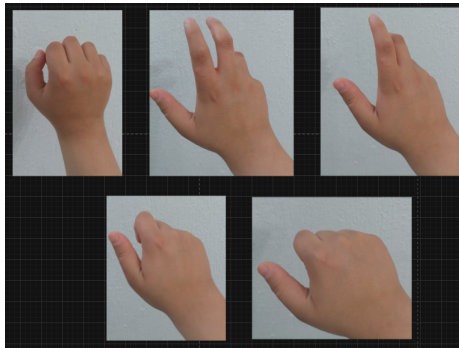
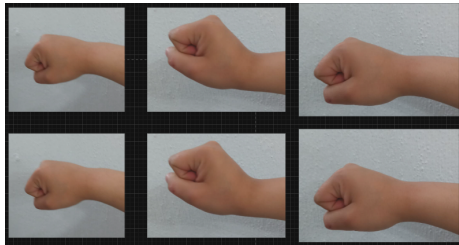


Figura 45: Tercer movimiento incluido en la base de datos: apretar y levantar la muñeca dos veces.



## 7.2. Filtrado y procesamiento de señales

Para garantizar la calidad de las señales electromiográficas (EMG) adquiridas, se implementaron dos filtros digitales diseñados específicamente para mejorar la relación señal-ruido y resaltar las características fisiológicas relevantes. Estos filtros, un pasa banda y un rechaza banda (Notch), fueron diseñados utilizando la metodología Butterworth debido a su respuesta suave en frecuencia, lo que permite preservar las componentes útiles de las señales mientras se eliminan las no deseadas.

El filtro pasa banda se diseñó con frecuencias de corte de 50 Hz y 250 Hz, utilizando un filtro Butterworth de orden 20. Este filtro se aplicó para aislar el rango de frecuencias relevante para las señales EMG, eliminando componentes de baja frecuencia, como el ruido generado por movimientos del cuerpo o artefactos lentos, y de alta frecuencia, como el ruido electrónico o interferencias externas. Este proceso permitió que las señales procesadas

conservaran únicamente la información fisiológica significativa, mejorando la precisión del análisis.

Por otro lado, el filtro rechaza banda (Notch) se diseñó específicamente para mitigar las interferencias de la red eléctrica, centradas en 60 Hz. Este filtro, también de tipo Butterworth y de orden 20, cuenta con frecuencias de corte de 58 Hz y 62 Hz, lo que asegura la eliminación eficaz del ruido de la red eléctrica sin comprometer las componentes útiles de la señal EMG. En conjunto, la implementación de estos filtros resultó esencial para obtener datos de alta calidad y garantizar la fiabilidad de los análisis posteriores.

### 7.3. Interfaz de recolección de señales EMG

La interfaz fue diseñada específicamente para interactuar con un dispositivo bitalino, permitiendo al usuario controlar y monitorear de manera eficiente el proceso de adquisición de datos. Inicialmente, se tomó como referencia la interfaz de Roberto Cáceres [5], que sirvió como base conceptual para el desarrollo. Sin embargo, para garantizar un mayor control sobre todas las funciones y adaptarla a las necesidades específicas del proyecto, se decidió construir una nueva interfaz desde cero, incorporando mejoras significativas en funcionalidad y usabilidad.

En esta nueva implementación, el usuario puede seleccionar el bitalino correspondiente, asignar un nombre al dispositivo y definir el tipo de gesto que desea registrar. Una vez configurada, la interfaz permite iniciar el proceso de adquisición de datos, durante el cual se recopilan señales en tiempo real. Además, la interfaz incluye funcionalidades avanzadas, como la visualización de mensajes que confirman la correcta realización de cada gesto y un contador que muestra la cantidad total de gestos completados en la sesión de prueba. Estas características aseguran un monitoreo constante del progreso y la calidad de los datos adquiridos. Como lo podemos observar en la Figura 46.

Figura 46: Interfaz de recolección de señales EMG (nueva versión.)



Finalmente, para facilitar la comprensión del flujo de trabajo de la interfaz, se incluye un diagrama de flujo en la Figura 47 ,que detalla cada etapa del proceso, desde la configuración inicial del dispositivo hasta la finalización de la sesión de adquisición.

Figura 47: Diagrama de flujo del algoritmo de interfaz de recolección EMG.



---

## Clasificación de señales EMG

---

Este capítulo está dedicado al análisis y clasificación de las señales electromiográficas (EMG) previamente adquiridas. Se detalla el proceso de extracción de características relevantes, seguido por el entrenamiento de diversos clasificadores. Además, se realiza una comparación exhaustiva entre modelos, evaluando sus rendimientos.

### 8.1. Extracción de características de las señales filtradas

Se desarrolló un algoritmo que permitió obtener las características de las señales filtradas, generando un vector de características que comprende 7 variables y 3 clases. Entre las características extraídas se encuentran *Zero crossing*, MAV y duración. Estas características se describen en mayor detalle en el capítulo 6.4.

El proceso consistió en extraer 2000 muestras, correspondientes a una ventana fija de tiempo de 1 segundo durante la ejecución de un movimiento específico debido que eso funcionó correctamente para el entrenamiento, pero esto puede llegar a ser variable en tiempo real. Adicionalmente, se analizó una ventana dividida en tres segmentos (sin señal - señal - sin señal), extrayendo MAV y *Zero crossing* de cada segmento añadiendo el tiempo de duración de cada gesto y concatenando los resultados para obtener un vector de 7 características.

El algoritmo exporta estos vectores de características a un archivo Excel, agregando una columna de etiquetas para su uso en clasificadores. Cada fila corresponde a una muestra individual procesada.

## 8.2. Entrenamiento de clasificadores por máquina de vectores de soporte (SVM) con kernels

Se trabajó con dos clasificadores principales: máquinas de vectores de soporte (SVM) con kernel polinomial y kernel Gaussiano.

Como se explicó anteriormente, para entrenar los clasificadores se utilizó la función **fitcecoc** de Matlab, permitiendo el entrenamiento de un clasificador SVM multiclase. Los datos de entrenamiento fueron leídos desde un archivo Excel que contenía las características extraídas y las etiquetas correspondientes.

El proceso de entrenamiento incluyó la validación cruzada con tres particiones (*five-fold cross-validation*), asegurando la robustez de los modelos generados. Se entrenaron dos clasificadores, uno con kernel polinomial de grado 3 y otro con kernel Gaussiano. La precisión de cada modelo se evaluó mediante la función **kfoldLoss**, calculando el porcentaje de precisión obtenido.

Finalmente, se generaron matrices de confusión para cada modelo, lo que permitió comparar el desempeño entre el kernel polinomial, gaussiano y multiclase.

### 8.2.1. Pruebas preliminares de clasificación con ventana fija de la señal

#### Resultados del clasificador SVM con kernel gaussiano

Los resultados obtenidos para el clasificador con kernel Gaussiano se presentan a través de la matriz de confusión, destacando la precisión global alcanzada en las 3 clases.

Figura 48: Matriz de confusión del clasificador SVM con kernel gaussiano de las pruebas preliminares.

**Matriz de Confusión - SVM Kernel Gaussiano**

	1	2	3
1	9	6	13
2	4	37	3
3	7	1	29

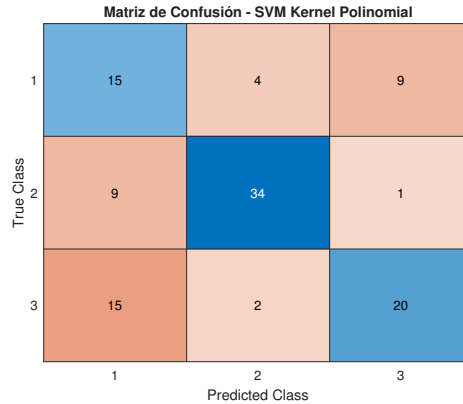
Predicted Class

Precisión de validación cruzada para el modelo SVM Gaussiano: 68.81%

## Resultados del clasificador SVM con kernel polinomial

Para el clasificador SVM con kernel polinomial, también se generó una matriz de confusión, mostrando las predicciones correctas e incorrectas.

Figura 49: Matriz de confusión del clasificador SVM con kernel polinomial de las pruebas preliminares.

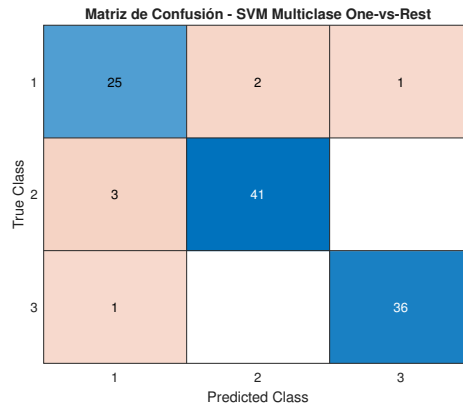


Precisión de validación cruzada para el modelo SVM polinomial de grado 3: 63.30 %

## Resultados del clasificador SVM Multiclase (*One-vs-Rest*)

Los resultados obtenidos para el clasificador multiclase se presentan a través de la matriz de confusión, destacando la precisión global alcanzada en las 3 clases.

Figura 50: Matriz de confusión del clasificador SVM multiclase de las pruebas preliminares.



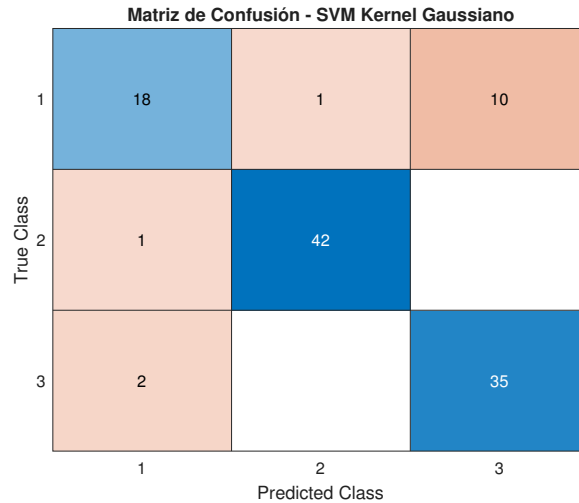
Precisión de validación cruzada para el modelo SVM Multiclase: 93.58 %

### 8.2.2. Clasificación por segmentos de la señal

#### Resultados del clasificador SVM con kernel gaussiano

Los resultados obtenidos para el clasificador SVM con kernel Gaussiano en el caso de segmentación de señales se presentan a través de la matriz de confusión, como se muestra en la Figura 51. Este enfoque permitió analizar la precisión global del modelo considerando las distintas etapas de la señal segmentada. La matriz muestra un desempeño aceptable en las tres clases, destacando la capacidad del modelo para clasificar correctamente la mayor parte de las observaciones, aunque con algunos errores en la clase 3.

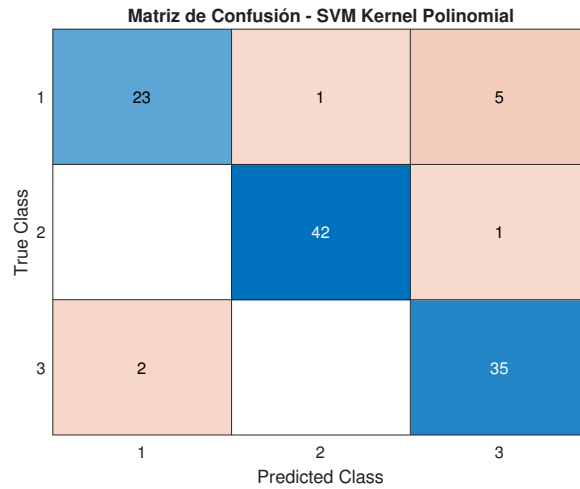
Figura 51: Matriz de confusión del clasificador SVM con kernel gaussiano por segmentos de la señal.



#### Resultados del clasificador SVM con kernel polinomial

La Figura 52 muestra la matriz de confusión para el clasificador SVM con kernel polinomial en el caso de segmentación de señales. Este modelo exhibió un mejor balance entre las clases, especialmente en la clase 2, donde logró una clasificación más consistente. Sin embargo, se observan confusiones significativas en las clases 1 y 3. La precisión global alcanzada por este modelo fue del 74.31 %, lo que indica una mejora en comparación con el modelo Gaussiano

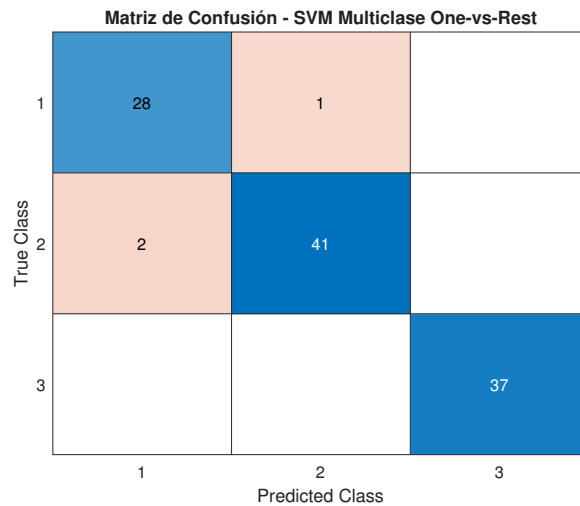
Figura 52: Matriz de confusión del clasificador SVM con kernel polinomial por segmentos de la señal.



### Resultados del clasificador SVM multiclase (*One-vs-Rest*)

En la Figura 53 se presentan los resultados del clasificador SVM multiclase con estrategia One-vs-Rest para la segmentación por señales. Este enfoque logró la mayor precisión global entre los tres clasificadores evaluados, alcanzando un 94.5%. La matriz de confusión muestra una alta consistencia en las predicciones correctas, particularmente en la clase 2. Este desempeño destaca la efectividad del modelo One-vs-Rest en contextos donde la segmentación de señales mejora la separabilidad de las clases. La baja incidencia de errores refleja la capacidad del modelo para manejar distribuciones complejas de datos segmentados.

Figura 53: Matriz de confusión del clasificador SVM multiclase por segmentos de la señal.



### 8.3. Entrenamiento de clasificadores por máquina lineal

Se emplearon tres clasificadores dentro del marco de las máquinas lineales: el Análisis Discriminante Lineal (LDA), la Máquina de Vectores de Soporte Lineal (SVM Lineal) y el Clasificador Bayesiano Lineal. Estos métodos son ampliamente utilizados en problemas de clasificación por su capacidad para encontrar soluciones lineales que separen las diferentes clases en el espacio de características.

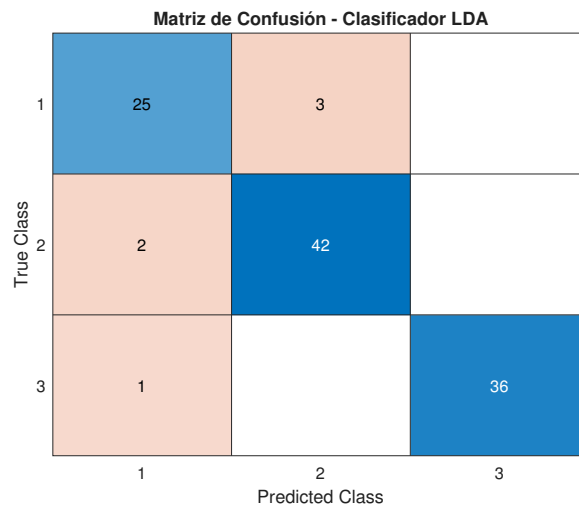
Para entrenar estos clasificadores, se utilizó la validación cruzada con tres particiones (*five-fold cross-validation*), asegurando la robustez de los modelos generados. Los datos de entrenamiento y prueba se dividieron en subconjuntos balanceados, facilitando la evaluación de precisión mediante métricas de desempeño como la matriz de confusión y el porcentaje de precisión.

#### 8.3.1. Pruebas preliminares de clasificación con ventana fija de la señal

##### Resultados del clasificador por análisis discriminante lineal (LDA)

El clasificador LDA mostró un desempeño sólido, como se observa en la Figura 54, donde las clases fueron correctamente clasificadas en la mayoría de los casos. La matriz de confusión refleja que la precisión global para este modelo fue del 94.50 %.

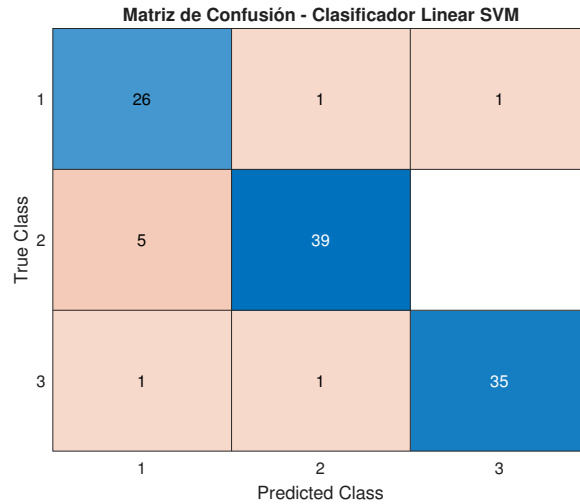
Figura 54: Matriz de confusión del clasificador análisis discriminante lineal de las pruebas preliminares.



##### Resultados del clasificador por máquina de vectores de soporte lineal (SVM)

La SVM Lineal logró un buen rendimiento, como se ilustra en la Figura 55. Este modelo es particularmente efectivo para datos linealmente separables, lo que se refleja en la baja cantidad de errores de clasificación para las clases principales. La precisión global alcanzada fue del 91.74 %.

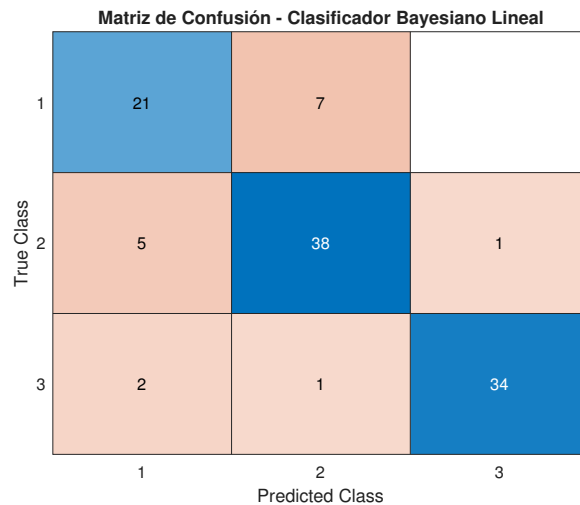
Figura 55: Matriz de confusión del clasificador SVM lineal de las pruebas preliminares.



### Resultados del clasificador por bayesiano lineal

El clasificador bayesiano lineal también fue evaluado, como se observa en la Figura 56. Aunque mostró un rendimiento ligeramente inferior en comparación con los otros clasificadores, sigue siendo una herramienta efectiva para problemas con datos bien distribuidos. La precisión global alcanzada fue del 85.31 %.

Figura 56: Matriz de confusión del clasificador bayesiano lineal de las pruebas preliminares.



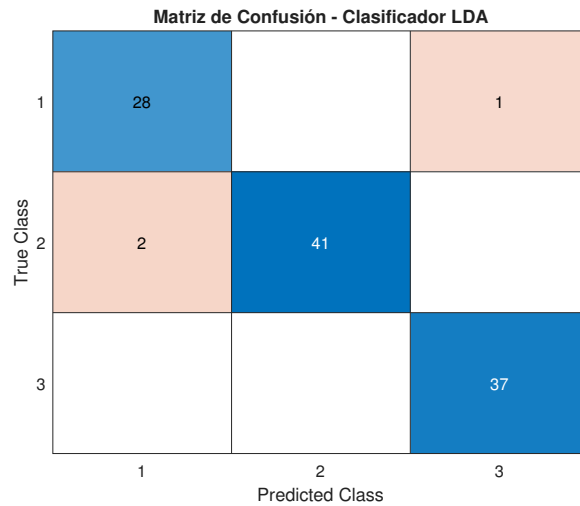
### 8.3.2. Clasificación por segmentos de la señal

#### Resultados del clasificador por análisis discriminante lineal (LDA)

Para la clasificación por segmentos, el modelo LDA fue capaz de identificar las clases con mayor precisión, como se muestra en la Figura 57. La capacidad del modelo para adaptarse

a las diferentes características de los segmentos permitió mantener una buena precisión.

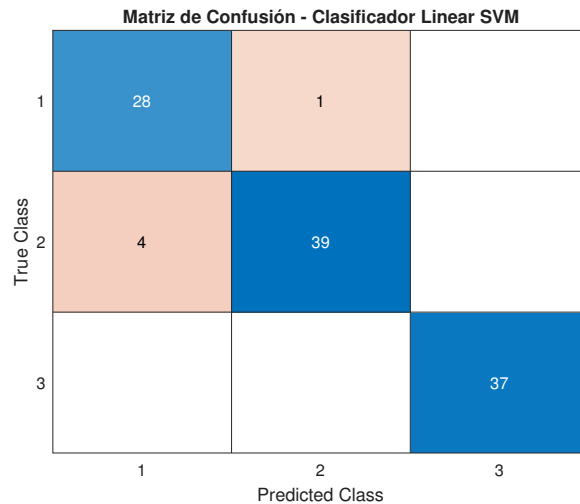
Figura 57: Matriz de confusión del clasificador análisis discriminante lineal por segmentos de la señal.



### Resultados del clasificador por máquina de vectores de soporte lineal (SVM)

La SVM Lineal obtuvo resultados consistentes con los segmentos de señal, como se ilustra en la Figura 58. Este modelo mantuvo una clasificación robusta a pesar de las diferencias en las características de los segmentos.

Figura 58: Matriz de confusión del clasificador SVM lineal por segmentos de la señal.



### Resultados del clasificador por bayesiano lineal

El Clasificador Bayesiano Lineal, aunque eficiente, tuvo dificultades para separar correctamente las clases en algunos segmentos, lo que se refleja en la Figura 59. Sin embargo, su desempeño general fue satisfactorio.

Figura 59: Matriz de confusión del clasificador bayesiano lineal por segmentos de la señal.

**Matriz de Confusión - Clasificador Bayesiano Lineal**

True Class	1	28	1	
	2	1	42	
	3	1		36
		1	2	3
		Predicted Class		

## 8.4. Entrenamiento de clasificadores por redes neuronales (ANN)

Estas redes están inspiradas en el funcionamiento del cerebro humano, utilizando neuronas artificiales organizadas en capas para aprender patrones complejos a partir de los datos de entrada. Los modelos se diseñaron con arquitecturas específicas adaptadas a los datos del problema, destacando tres configuraciones principales: redes de dos capas, tres capas y arquitectura ancha.

Para entrenar las redes neuronales, se emplearon los datos extraídos de las características del conjunto de entrenamiento, y se realizó un preprocesamiento para normalizar los valores de entrada. La configuración del entrenamiento incluyó los siguientes elementos:

Criterio de entrenamiento: Alcanzar un valor mínimo en la función de pérdida. Algoritmo de entrenamiento: Scaled Conjugate Gradient, seleccionado por su eficiencia para problemas con múltiples características. Función de pérdida: Cross Entropy, adecuada para problemas de clasificación multiclase. División de datos: Datos divididos en conjuntos de entrenamiento, validación y prueba para garantizar una evaluación robusta del modelo. El diseño arquitectónico de las redes incluyó la siguiente estructura:

Capa de entrada: Conformada por 18 neuronas, correspondientes a las características extraídas de los datos. Capas ocultas: El número y tamaño de estas capas varió según la arquitectura evaluada (10, 15 y 50 neuronas). Capa de salida: Con 3 neuronas, representando las clases objetivo.

Modelos Implementados Red Neuronal de Dos Capas El primer modelo se configuró con una capa oculta de 10 neuronas. Este diseño permitió capturar patrones básicos en los datos, logrando un equilibrio entre simplicidad y capacidad predictiva. La precisión global fue satisfactoria para las tres clases, como se observa en las matrices de confusión.

Red Neuronal de Tres Capas En este modelo, se incluyó una segunda capa oculta con 5 neuronas, lo que aumentó la capacidad de la red para aprender patrones más complejos.

Este diseño mostró mejoras en la precisión general, especialmente en las clases con mayor variabilidad de características.

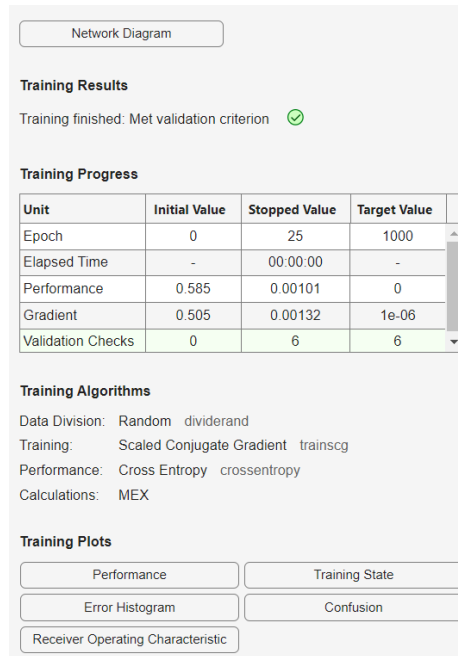
Red Neuronal de Arquitectura Ancha Finalmente, se evaluó un modelo con 50 neuronas en una sola capa oculta. Este enfoque permitió capturar relaciones más complejas entre las características, logrando una precisión global superior en comparación con los modelos anteriores. Sin embargo, este aumento en el número de neuronas incrementó el tiempo de entrenamiento y el riesgo de sobreajuste.

### 8.4.1. Pruebas preliminares de clasificación con ventana fija de la señal

#### Resultados del clasificador por redes neuronales de dos capas

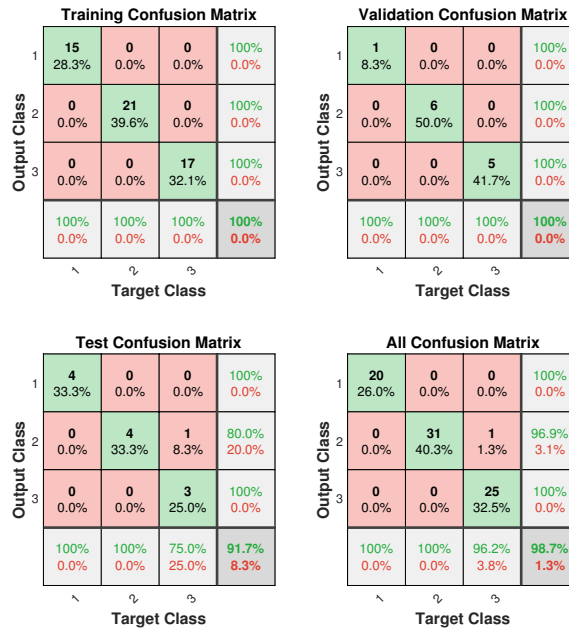
El entrenamiento de la red neuronal de dos capas se llevó a cabo con una configuración inicial que incluye una capa de entrada con 18 características, una capa oculta con 10 neuronas, y una capa de salida con 3 neuronas, correspondiente a las clases del problema. En la Figura 60 , se muestra el progreso del entrenamiento, donde se destaca que el criterio de validación se alcanzó tras 25 épocas. La función de pérdida utilizada fue *Cross entropy*, y el algoritmo de entrenamiento fue el método de *Scaled Conjugate Gradient*.

Figura 60: Proceso de entrenamiento de red neuronal de 2 capas para las pruebas preliminares.



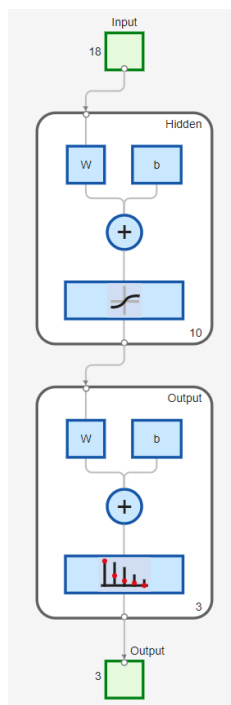
La red obtuvo una precisión global aceptable según las matrices de confusión generadas. En la Figura 61 , se presentan las matrices de confusión para los conjuntos de entrenamiento, validación, prueba y todos los datos combinados. Estas matrices reflejan el desempeño del modelo, destacando que las predicciones fueron consistentes en todas las clases.

Figura 61: Matriz de confusión de 2 capas para las pruebas preliminares.



Además, el diagrama de la red neuronal en la Figura 62 muestra la arquitectura utilizada, donde cada capa realiza transformaciones específicas de los datos de entrada para optimizar la clasificación.

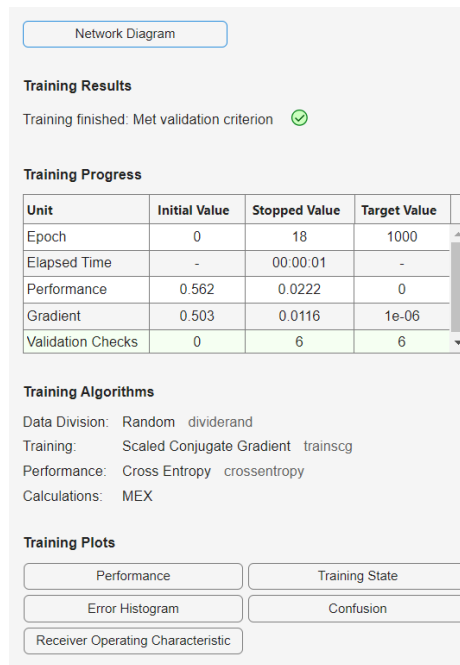
Figura 62: Red neuronal de 2 capas para las pruebas preliminares.



## Resultados del clasificador por redes neuronales de tres capas

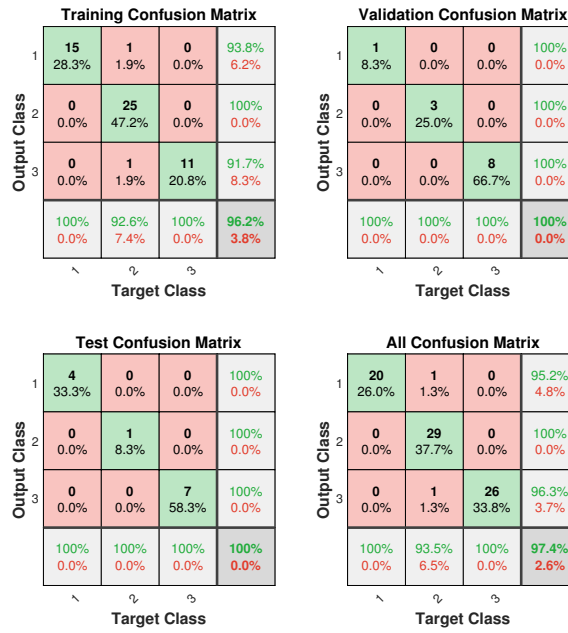
El clasificador de tres capas se configuró añadiendo una segunda capa oculta con 5 neuronas, lo que permitió aprender patrones más complejos en los datos. La Figura 63 muestra que el modelo alcanzó el criterio de validación tras 18 épocas, con un rendimiento ligeramente mejorado respecto al modelo de dos capas.

Figura 63: Proceso de entrenamiento de red neuronal de 3 capas para las pruebas preliminares



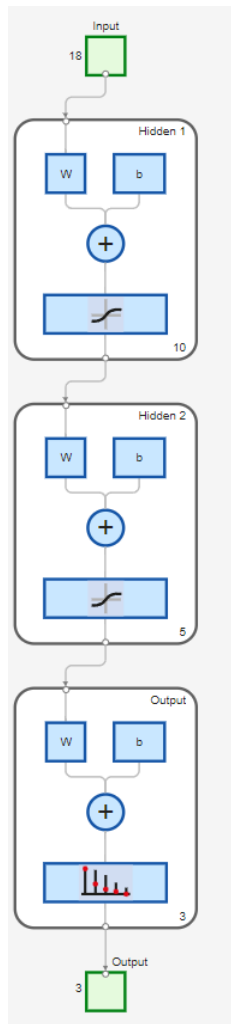
En la Figura 64, se pueden observar las matrices de confusión correspondientes, donde el modelo muestra un buen desempeño general, logrando identificar correctamente las clases en los conjuntos de prueba y validación.

Figura 64: Matriz de confusión de 3 capas para las pruebas preliminares.



El diagrama estructural del modelo se presenta en la Figura 65, mostrando la integración de las capas adicionales y cómo estas contribuyen al procesamiento de las características.

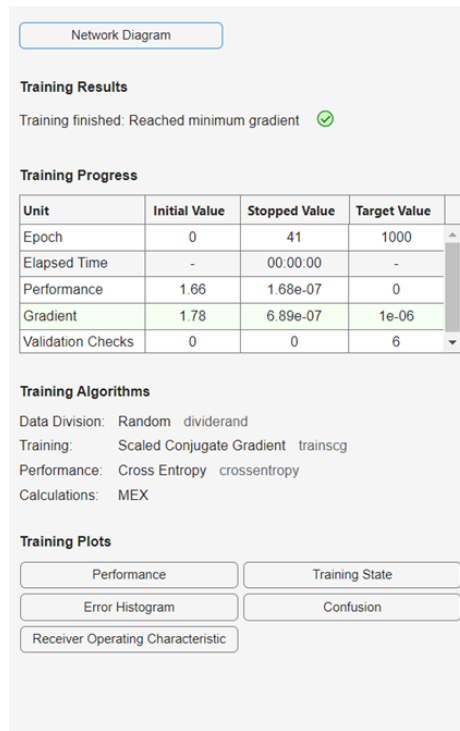
Figura 65: Red neuronal de 3 capas para las pruebas preliminares.



### Resultados del clasificador por redes neuronales de arquitectura ancha

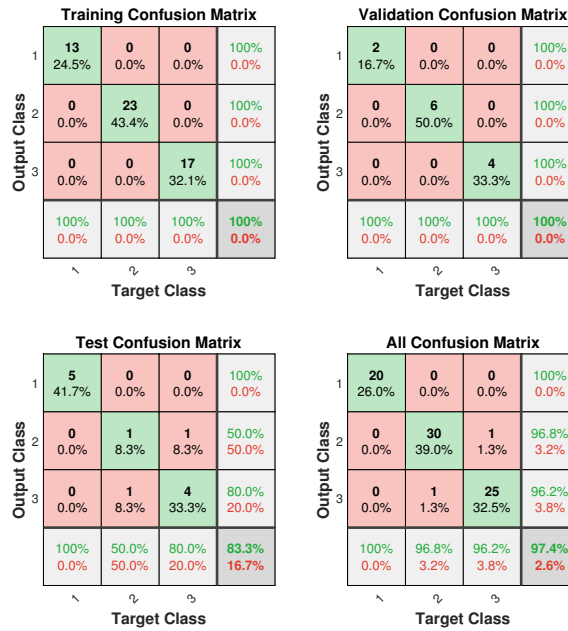
Para el modelo de arquitectura ancha, se incrementó el número de neuronas en la capa oculta a 50, lo que permitió capturar relaciones más complejas en los datos. Como se puede observar en la Figura 66, el modelo alcanzó un criterio de entrenamiento óptimo tras 41 épocas.

Figura 66: Proceso de entrenamiento de red neuronal de 50 capas ocultas para las pruebas preliminares



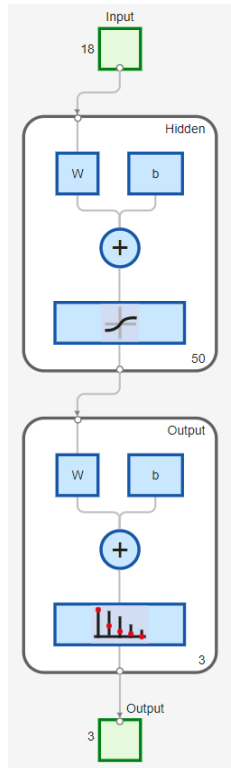
En la Figura 67, las matrices de confusión reflejan un desempeño robusto, destacando una mayor precisión en la identificación de las clases objetivo. Esto se debe al mayor número de parámetros del modelo, que permitió una representación más detallada de las características.

Figura 67: Matriz de confusión de 50 capas ocultas para las pruebas preliminares.



El diagrama estructural de la red neuronal se muestra en la Figura 68, evidenciando la simplicidad en la arquitectura a pesar del aumento en las neuronas de la capa oculta.

Figura 68: Red neuronal de 50 capas ocultas para las pruebas preliminares.

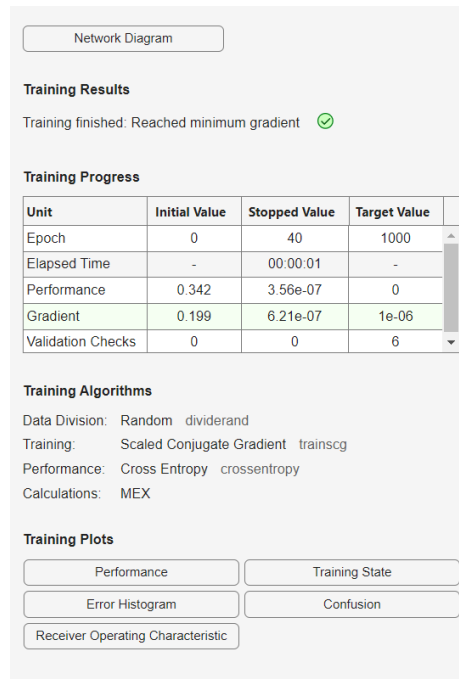


#### 8.4.2. Clasificación por segmentos de la señal

##### Resultados del clasificador por redes neuronales de dos capas

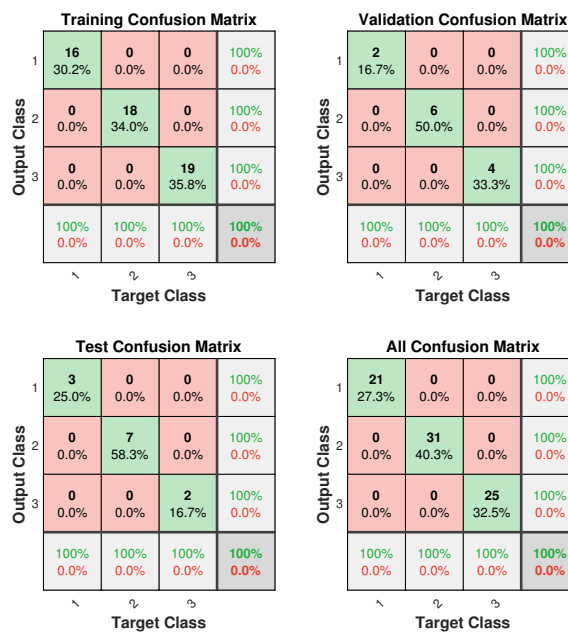
Para el clasificador de redes neuronales de dos capas, el proceso de entrenamiento alcanzó un criterio de validación, finalizando en la época 40. Los resultados del entrenamiento se presentan en la Figura 69 , destacando un rendimiento con una métrica de error de 0.342, un gradiente reducido a  $6.21 \times 10^{-7}$ , sin fallos en la validación cruzada.

Figura 69: Proceso de entrenamiento de red neuronal de 2 capas por segmentos de la señal.



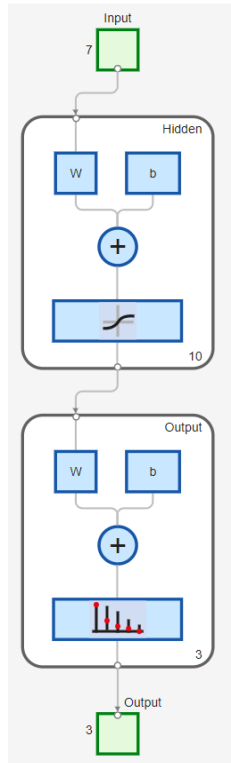
La Figura 70 ilustra las matrices de confusión para el conjunto de entrenamiento, validación y prueba, mostrando que la red logró identificar correctamente la mayoría de las clases objetivo.

Figura 70: Matriz de confusión de 2 capas por segmentos de la señal.



Finalmente, la arquitectura de la red se detalla en la Figura 71 , conformada por una capa de entrada, una capa oculta con 10 neuronas, y una capa de salida para las tres clases.

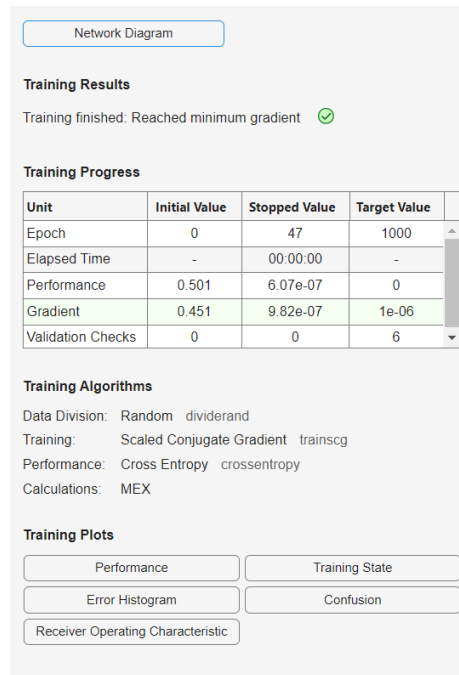
Figura 71: Red neuronal de 2 capas por segmentos de la señal.



### Resultados del clasificador por redes neuronales de tres capas

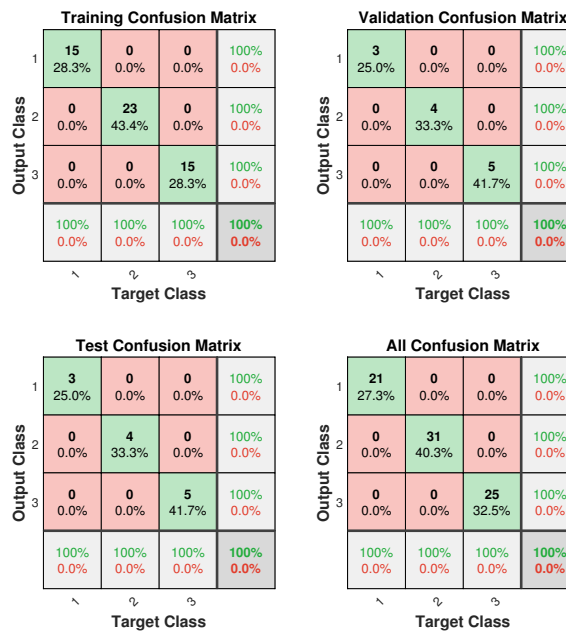
El clasificador de tres capas alcanzó su criterio de validación en la época 47, como se detalla en la Figura 72. El rendimiento obtenido muestra un error de 0.501, con un gradiente mínimo de  $8.02 \times 10^{-7}$ , y una división equilibrada entre las muestras para las particiones de validación.

Figura 72: Proceso de entrenamiento de red neuronal de 3 capas por segmentos de la señal.



Las matrices de confusión, presentadas en la Figura 73 , destacan una precisión razonable en las predicciones, aunque con ligeras confusiones en algunas clases.

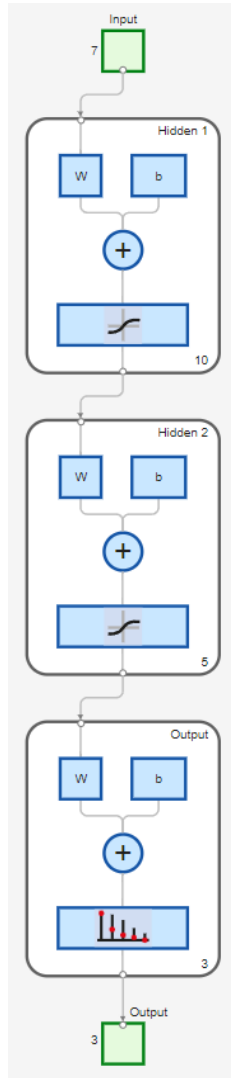
Figura 73: Matriz de confusión de 3 capas por segmentos de la señal.



La estructura de la red, descrita en la Figura 74 , incluye una capa de entrada, dos

capas ocultas con 10 y 5 neuronas respectivamente, y una capa de salida con tres nodos correspondientes a las clases.

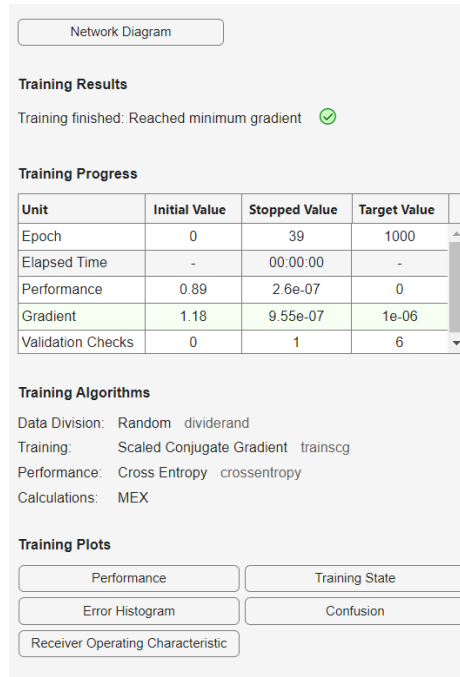
Figura 74: Red neuronal de 3 capas por segmentos de la señal.



### Resultados del clasificador por redes neuronales de arquitectura ancha

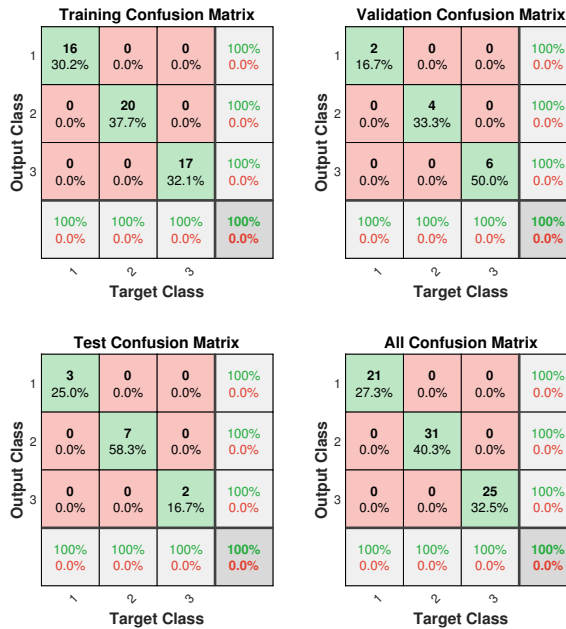
El modelo de arquitectura ancha fue diseñado con 50 neuronas en su capa oculta, alcanzando su criterio de parada en la época 39, como se muestra en la Figura 75. Este modelo presentó un rendimiento con una métrica de error de 1.66 y un gradiente reducido a  $9.55 \times 10^{-7}$ .

Figura 75: Proceso de entrenamiento de red neuronal de 50 capas ocultas por segmentos de la señal.c



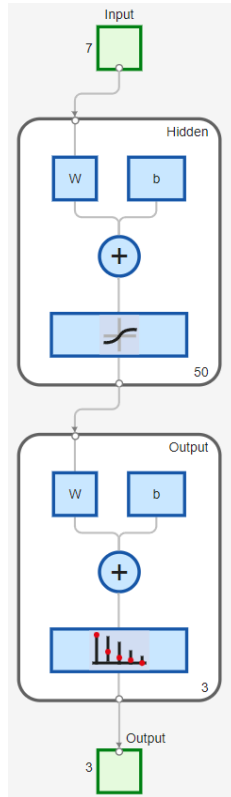
Las matrices de confusión, presentadas en la Figura 76, reflejan un comportamiento consistente en la clasificación de los datos de entrenamiento, validación y prueba.

Figura 76: Matriz de confusión de 50 capas ocultas por segmentos de la señal.



La Figura 77 ,describe la estructura de la red neuronal, conformada por una capa de entrada, una capa oculta con 50 neuronas, y una capa de salida para las clases objetivo.

Figura 77: Red neuronal de 50 capas ocultas por segmentos de la señal.



## 8.5. Entrenamiento de clasificadores con el algoritmo de los $k$ vecinos más cercanos ( $K$ -nearest neighbor)

Se trabajó con tres variantes del clasificador  $k$ -NN: KNN estándar, KNN ponderado por distancia y KNN adaptativo (Weighted KNN). Cada variante fue evaluada utilizando un conjunto de datos estructurado y segmentado para analizar su desempeño en problemas de clasificación.

Como se explicó anteriormente, para implementar estos clasificadores se utilizó la herramienta `fitcknn` de Matlab, que permite el entrenamiento de modelos  $k$ -NN con diferentes configuraciones y funciones de distancia. Los datos de entrenamiento fueron leídos desde un archivo Excel, que contenía las características extraídas y sus correspondientes etiquetas.

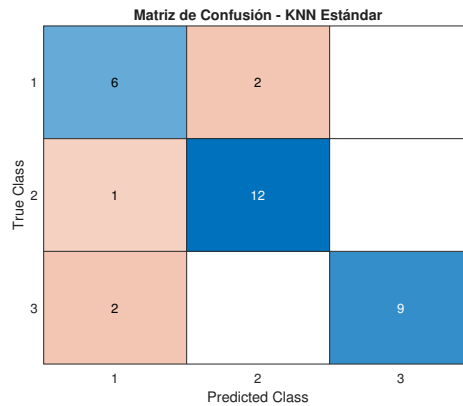
El proceso de entrenamiento incluyó la validación cruzada para asegurar la robustez de los modelos generados, comparando la precisión de clasificación entre las tres variantes. Finalmente, se generaron matrices de confusión para cada variante, lo que permitió analizar el desempeño de cada modelo en términos de precisión global y precisión por clase.

### 8.5.1. Pruebas preliminares de clasificación con ventana fija de la señal

#### Resultados del clasificador por KNN estándar

El clasificador KNN estándar utiliza una estrategia básica donde los puntos de datos se clasifican considerando únicamente la mayoría de las clases presentes en los  $k$  vecinos más cercanos. Este enfoque simple es ideal para problemas en los que la distribución de las clases es homogénea y balanceada. En este caso, la matriz de confusión obtenida se presenta en la Figura 78.

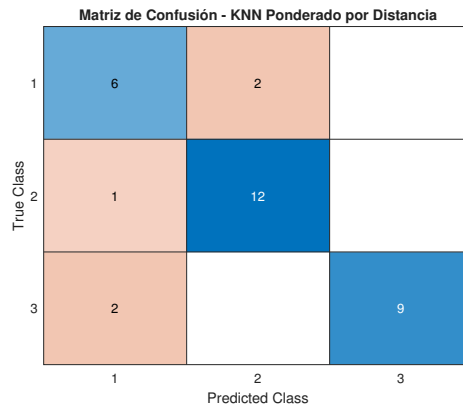
Figura 78: Matriz de confusión del clasificador KNN estándar con ventana fija de la señal.



#### Resultados del clasificador por KNN ponderado por distancia

El clasificador KNN ponderado por distancia mejora la estrategia básica al dar mayor peso a los vecinos más cercanos, considerando que estos son más representativos. La matriz de confusión obtenida con este método se ilustra en la Figura 79.

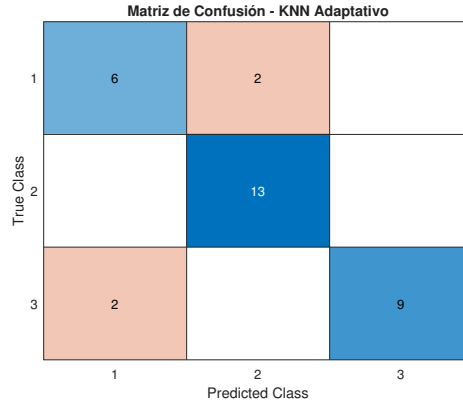
Figura 79: Matriz de confusión del clasificador KNN ponderado por distancia con ventana fija de la señal.



### Resultados del clasificador por KNN adaptativo (*Weighted KNN*)

El KNN adaptativo ajusta dinámicamente los pesos de las características en función de su importancia relativa para la clasificación, utilizando métricas adicionales como la información mutua. La matriz de confusión para este enfoque se presenta en la Figura 80.

Figura 80: Matriz de confusión del clasificador KNN adaptativo con ventana fija de la señal.

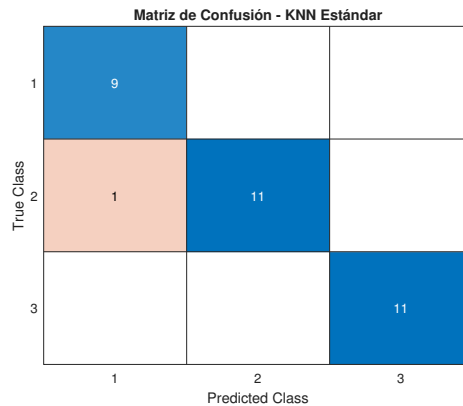


### 8.5.2. Clasificación por segmentos de la señal

#### Resultados del clasificador por KNN estándar

El desempeño del clasificador KNN estándar en la clasificación por segmentos de la señal se evalúa mediante la matriz de confusión mostrada en la Figura 81.

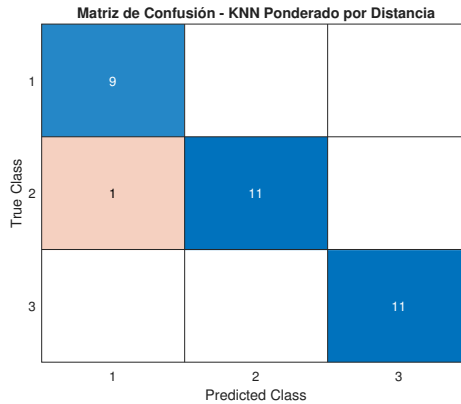
Figura 81: Matriz de confusión del clasificador KNN estándar por segmentos de la señal.



#### Resultados del clasificador por KNN ponderado por distancia

La Figura 82 presenta los resultados del clasificador ponderado por distancia al aplicarse en la clasificación por segmentos.

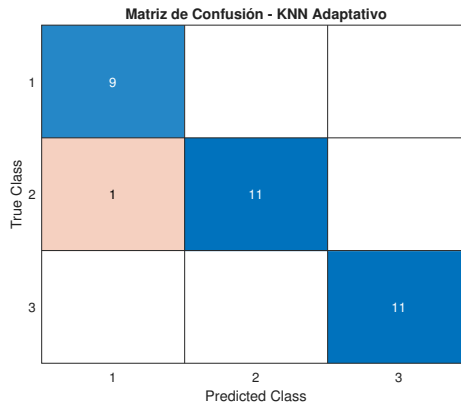
Figura 82: Matriz de confusión del clasificador KNN ponderado por segmentos de la señal.



### Resultados del clasificador por KNN adaptativo (*Weighted KNN*)

Finalmente, la matriz de confusión obtenida para el KNN adaptativo se presenta en la Figura 83, mostrando su capacidad de ajustar dinámicamente las métricas en base a la relevancia de las características.

Figura 83: Matriz de confusión del clasificador KNN adaptativo por segmentos de la señal.



### 8.5.3. Promedio de los resultados

#### Resultados sin promedios

El Cuadro 3, se muestran los valores específicos de precisión alcanzados por cada clasificador. Se pueden observar diferencias importantes entre las configuraciones de ventana fija y tres segmentos, destacando que la mayoría de los modelos presentan un mejor desempeño con la configuración de tres segmentos. Por ejemplo, el clasificador Multiclase alcanza un 93.58 % con ventana fija y un notable 97.25 % con tres segmentos, mientras que el modelo de red neuronal de arquitectura ancha logra un desempeño sobresaliente del 100 % con tres segmentos.

Cuadro 3: Resultados por modelo sin incluir promedios.

<b>Modelo</b>	<b>Ventana fija</b>	<b>Tres segmentos</b>
Gaussiano	68.81 %	87.16 %
Polinomial	63.30 %	91.74 %
Multiclase	93.58 %	97.25 %
LDA	94.50 %	97.25 %
SVM lineal	91.74 %	95.41 %
Bayesiano lineal	85.31 %	97.27 %
2 capas	96.88 %	96.88 %
3 capas	81.25 %	96.88 %
Arquitectura ancha	87.50 %	100.00 %
KNN estándar	84.38 %	96.88 %
KNN ponderado por distancia	84.38 %	96.88 %
KNN adaptativo weighted	87.50 %	96.88 %

### Promedios por modelo

El Cuadro 4, presenta los promedios calculados para cada categoría de modelos: SVM, clasificadores lineales, redes neuronales (ANN) y KNN. Los promedios permiten realizar una comparación general del desempeño de cada modelo. Se destaca que los clasificadores lineales obtuvieron el mayor promedio tanto para ventana fija (90.52 %) como para tres segmentos (96.64 %), mientras que las redes neuronales (ANN) también alcanzaron altos valores y con un promedio de 97.92 % en tres segmentos.

Cuadro 4: Promedio de resultados por modelo.

<b>Modelo</b>	<b>Ventana fija</b>	<b>Tres segmentos</b>
Promedio SVM	75.23 %	92.05 %
Promedio Lineal	90.52 %	96.64 %
Promedio ANN	88.54 %	97.92 %
Promedio KNN	85.42 %	96.88 %

### Promedio total

El Cuadro 5, se resume el desempeño general promedio para todas las configuraciones y modelos. La configuración de tres segmentos supera la de ventana fija, con un promedio total de 95.87 % frente al 84.93 %. Este resultado confirma que la división en tres segmentos favorece la precisión de los clasificadores.

Cuadro 5: Promedio total de los resultados para todas las configuraciones.

<b>Métrica</b>	<b>Promedio Total</b>
Ventana fija	84.93 %
Tres segmentos	95.87 %

En general, los resultados demuestran que la selección de características y el preprocesamiento del señal, en conjunto con la configuración adecuada, tienen un impacto significativo en el desempeño de los clasificadores.

---

## Adquisición y conversión de señales EOG

---

En este capítulo se describe el proceso completo de adquisición, preprocesamiento y conversión de señales electrooculográficas (EOG) utilizando el sistema biopac. Este sistema permite capturar señales fisiológicas con alta precisión y convertirlas en datos digitales utilizables para aplicaciones como el control de robots móviles. Se abordan los aspectos de configuración, procesamiento y calibración necesarios para garantizar la calidad de las señales adquiridas y su correcto uso.

### 9.1. Adquisición de señales EOG

#### 9.1.1. Familiarización con el sistema biopac

Para capturar señales EOG, fue necesario familiarizarse con el sistema biopac y su configuración. Los electrodos fueron colocados en la sien y la frente del participante, permitiendo registrar movimientos oculares en los ejes horizontales.

#### Colocación de electrodos

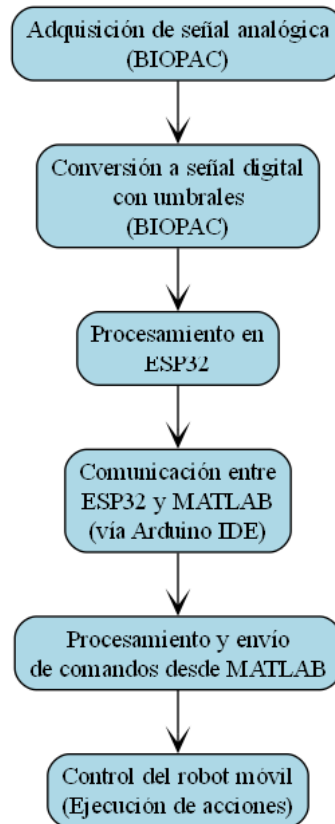
Los electrodos se colocaron estratégicamente en las sienes y la frente para registrar señales EOG, como se muestra en la Figura 30. Esta configuración permitió captar movimientos oculares horizontales y verificar la funcionalidad inicial del sistema.

### 9.1.2. Desarrollo del algoritmo de lectura

El desarrollo del sistema se basa en la integración de tecnologías para capturar, procesar y utilizar señales biomédicas. El flujo del proceso se muestra en la Figura 84 e incluye:

1. Adquisición de señales analógicas desde el sistema biopac.
2. Preprocesamiento de las señales mediante filtros.
3. Conversión a señales digitales en tiempo real.
4. Comunicación con el ESP32 para generar salidas digitales.

Figura 84: Diagrama de flujo del proceso de adquisición y conversión de señales EOG.



### 9.1.3. Comunicación y control con ESP32

El microcontrolador ESP32 fue configurado para recibir las señales EOG digitalizadas y procesarlas. Cada vez que una señal superaba el umbral configurado, el ESP32 generaba una salida digital. Se emplearon reguladores de voltaje para adaptar los niveles de las señales del biopac al rango de operación del ESP32.

## 9.2. Configuración del sistema

### 9.2.1. Canales de adquisición

El sistema biopac fue configurado con cuatro canales (C1 a C4), cada uno destinado a una función específica:

Configurado para medir la tasa de cambio ( $dP/dt$ ) con un rango de frecuencia de 0.05 Hz a 35 Hz. Este canal se utilizó para identificar variaciones rápidas en la señal.

Asignado a las señales suavizadas, eliminando ruido y mejorando su claridad. Se aplicó un promedio móvil con un factor de 50 muestras.

## 9.3. Preprocesamiento de señales

### 9.3.1. Filtro de diferencia (C1)

Se configuró un ajuste diferencial para resaltar cambios rápidos en las señales y eliminar componentes lentas. Esto es funcionalmente equivalente a un filtro pasa-alto.

### 9.3.2. Filtro de suavizado (C2)

Se utilizó un promedio móvil con un factor de 50 muestras para reducir ruido de alta frecuencia, mejorando la claridad de las señales.

### 9.3.3. Umbrales de control (C3 y C4)

Los umbrales se configuraron de la siguiente manera:

- **C3 (Ojo derecho):** Nivel bajo ( $L1 = 0.10$ ), nivel alto ( $L2 = 0.16$ ).
- **C4 (Ojo izquierdo):** Nivel bajo ( $L1 = -0.18$ ), nivel alto ( $L2 = -0.12$ ).

## 9.4. Proceso de conversión de las señales

El proceso de conversión de señales bioeléctricas capturadas por el sistema biopac fue clave para transformar las señales analógicas en valores digitales, asegurando su utilidad en aplicaciones prácticas como el control de robots. Este procedimiento involucró tres etapas principales que trabajan de manera integrada: adquisición, conversión y salida.

Primero, las señales analógicas fueron capturadas desde los canales configurados (C1, C2, C3 y C4) y monitoreadas en tiempo real utilizando el software biopac. Esto permitió verificar

que las señales cumplieran con los criterios establecidos para su análisis, como la claridad y la ausencia de ruido significativo. Durante esta etapa, se emplearon configuraciones específicas, como la medición de la tasa de cambio en el canal C1 y el suavizado de señales en el canal C2, para garantizar la calidad de las señales adquiridas.

Posteriormente, las señales analógicas fueron procesadas mediante la aplicación de umbrales definidos previamente en los canales C3 y C4. Estos umbrales, establecidos en niveles bajos y altos, permitieron identificar eventos específicos como movimientos oculares hacia la derecha o la izquierda. Por ejemplo, si la señal en el canal C3 superaba el nivel alto ( $L2 = 0.16$ ), se interpretaba como un movimiento ocular hacia la derecha, mientras que niveles inferiores a  $L1 = 0.10$  indicaban un estado inactivo. Este procedimiento de comparación transformó las señales analógicas en valores binarios (0 o 1), lo que facilitó su interpretación como comandos digitales.

Finalmente, los datos digitalizados fueron enviados a los canales de salida (D3 y D4) para su integración con sistemas externos. Estos valores binarios sirvieron como señales de control, permitiendo que un robot móvil ejecutara movimientos específicos basados en los eventos detectados. Por ejemplo, una señal binaria activa en el canal D3 indicaba un movimiento ocular hacia la derecha, lo cual desencadenaba una acción en el robot, como girar en esa dirección. Este flujo continuo entre la adquisición, conversión y salida asegura que las señales capturadas puedan ser utilizadas de manera eficiente y en tiempo real para aplicaciones de control.

---

## Validación de algoritmos por medio de simulación

---

En este capítulo, se presentan las señales digitales que fueron recibidas y procesadas por el ESP32. Estas señales son generadas por los pines de entrada, que están conectados a las salidas digitales provenientes de los dispositivos de adquisición de datos como bitalino o biopac.

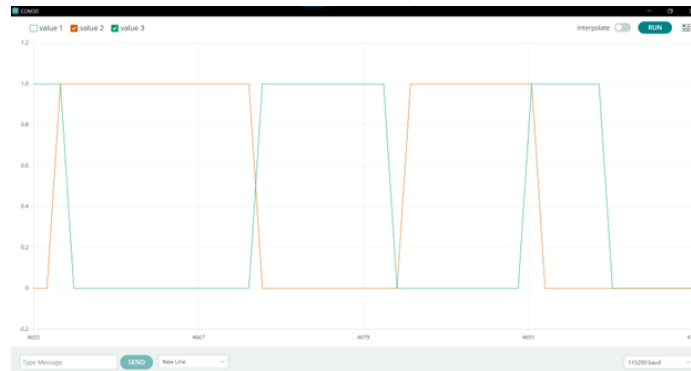
### 10.1. Verificación de recepción de señales

#### 10.1.1. Señales digitales del *ESP32*

En esta etapa, se implementó un sistema para recibir señales digitales provenientes de los pines de entrada del ESP32 configurados con resistencias pull-down (GPIO 27 y GPIO 13). Estas señales fueron procesadas para generar comandos que incrementaban o reducían el ángulo del servomotor en función de la dirección detectada (izquierda o derecha).

Las señales de los pines de entrada son leídas y procesadas en tiempo real. La posición del servo (en grados) se ajusta dependiendo de la señal detectada. Los resultados son enviados al monitor serial para visualización y análisis como se observa en la Figura 85.

Figura 85: Visualización del monitor serial del servomotor con las señales digitales



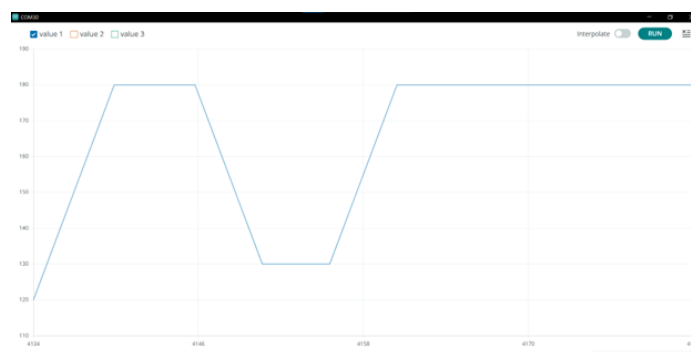
### 10.1.2. Servomotor

La segunda etapa consistió en utilizar las señales digitales procesadas para mover un servomotor. Dependiendo de las señales detectadas, el ángulo del servo se ajustaba entre  $0^\circ$  y  $180^\circ$ .

Comportamiento:

Incrementos de  $10^\circ$  en la posición del servo al detectar señales en el pin izquierdo. Decrementos de  $10^\circ$  al detectar señales en el pin derecho. Limitación del ángulo para evitar movimientos fuera del rango permitido. La Figura 86 muestra cómo el servomotor responde en tiempo real a las señales recibidas, representando su posición angular durante las pruebas realizadas.

Figura 86: Movimiento del servomotor en función de las señales digitales procesadas.



## 10.2. Captura de datos con biopac

El sistema biopac fue configurado para la captura de señales electrooculográficas (EOG), representando los movimientos oculares hacia la izquierda y la derecha. Estas señales, adquiridas mediante el software de biopac, se procesaron utilizando umbrales específicos para determinar la dirección del movimiento ocular. Por ejemplo, como se observa en la Figura 87, el software detectó movimientos oculares a la izquierda y derecha en función de las señales analógicas capturadas.

Cuando la señal superaba el umbral definido, se interpretaba como un movimiento ocular hacia la derecha; por el contrario, valores por debajo del umbral indicaban un movimiento hacia la izquierda. Estas configuraciones permitieron transformar las señales analógicas en valores digitales binarios mediante un microcontrolador ESP32, como se detalla en el diagrama de flujo de la Figura 87.

El siguiente paso consistió en establecer una conexión TCP/IP entre el ESP32 y el entorno de simulación Webots. Como se muestra en la Figura 88, las señales digitales convertidas (1 para derecha y 0 para izquierda) se transmitieron correctamente al entorno de simulación, replicando en tiempo real los movimientos detectados por el sistema biopac. Este flujo de datos continuo permitió el control eficiente de un robot simulado, que ejecutaba movimientos basados en las señales recibidas.

El diagrama de flujo en la Figura 87 describe el proceso completo, desde la configuración inicial del sistema biopac y el ESP32 hasta la transmisión de datos al entorno Webots. En caso de fallos en la conexión TCP/IP, el sistema está diseñado para reiniciar el proceso o finalizarlo si es necesario, garantizando la robustez del sistema.

Finalmente, las imágenes capturadas en el entorno de simulación, como la Figura 88, muestran los resultados del procesamiento de las señales en tiempo real, evidenciando el éxito en la integración entre el biopac, el ESP32 y Webots.

Figura 87: Diagrama de flujo de la configuración entre el biopac y el ESP32

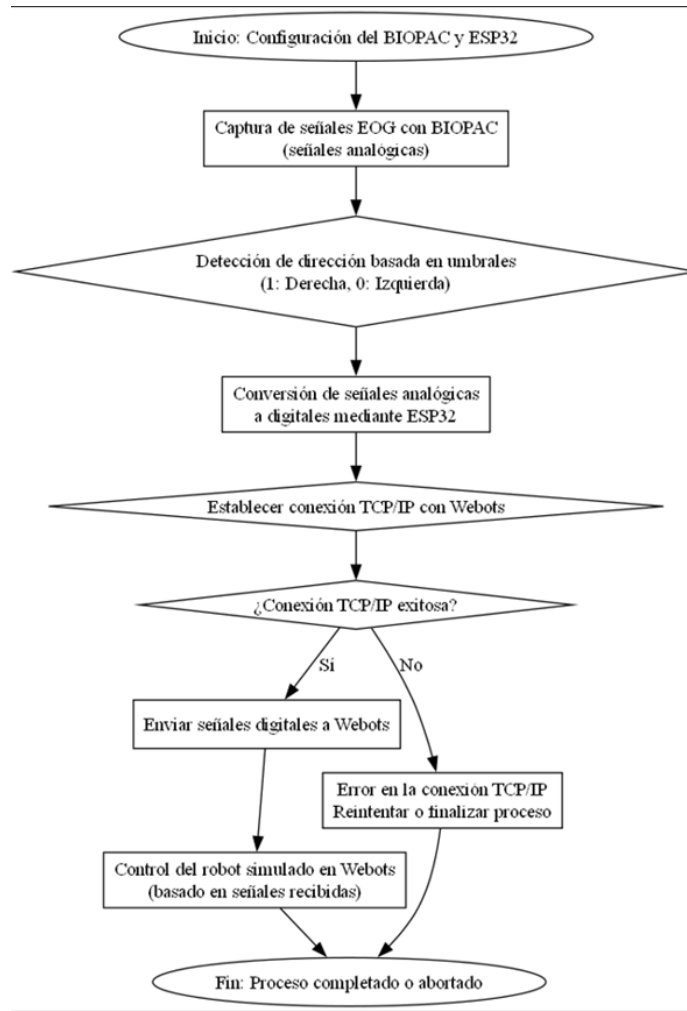
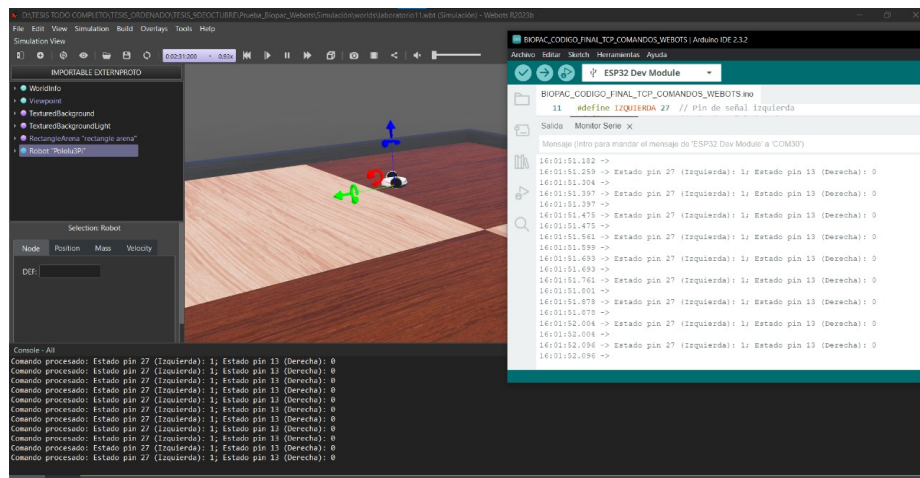


Figura 88: Visualización del funcionamiento de la simulación en Webots.

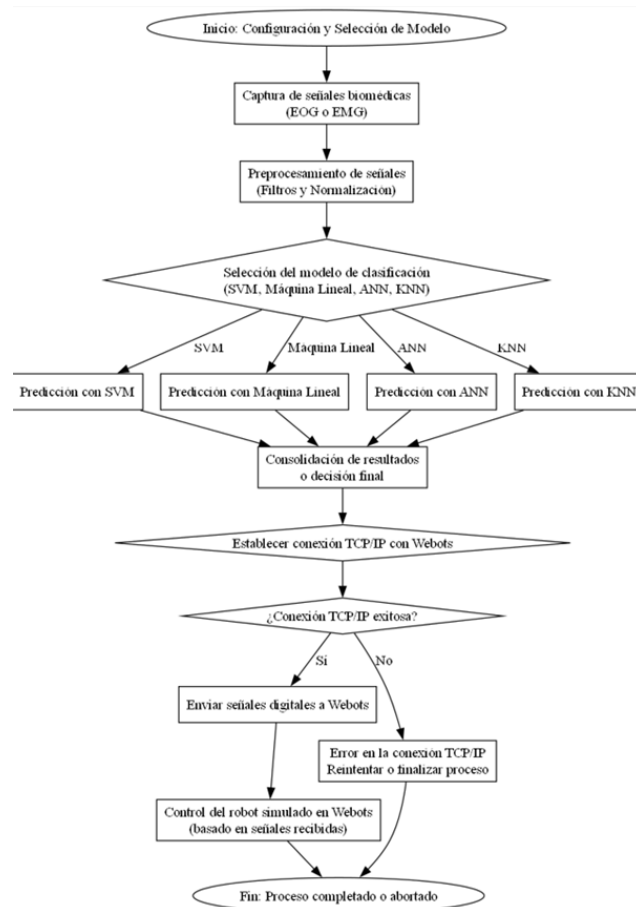


### 10.3. Captura de datos con bitalino

El sistema bitalino fue empleado para predecir gestos específicos basados en datos fisiológicos capturados. Estas señales se procesaron en Matlab utilizando un modelo de predicción previamente entrenado, almacenado en un archivo .mat. El modelo, implementado con un clasificador SVM, permitió identificar gestos como movimientos hacia la izquierda o la derecha en función de las características extraídas de las señales.

Tras realizar la predicción en Matlab, el sistema transmitió la información al ESP32. Este microcontrolador recibió los datos clasificados y los convirtió en señales digitales compatibles con Webots. A través del protocolo TCP/IP, las predicciones se enviaron al entorno de simulación, donde controlaron un robot que ejecutó las acciones asociadas a los gestos detectados. En la figura 89 es el flujo de trabajo integrado entre el bitalino, Matlab, el ESP32 y Webots demostró ser eficaz para traducir datos fisiológicos en movimientos controlados en un entorno simulado.

Figura 89: Diagrama de flujo del bitalino.



---

## Validación de algoritmos por medio de agente físico

---

### 11.1. Modo de operación manual

El modo de operación manual ofrece una herramienta fundamental para controlar directamente el robot físico, permitiendo realizar pruebas específicas de hardware y garantizar su correcto funcionamiento antes de implementar algoritmos automatizados. Este modo es clave para verificar la conectividad entre el sistema y el robot, así como para realizar ajustes en tiempo real en un entorno controlado.

La interfaz gráfica, representada en la Figura 90, está diseñada para ser intuitiva y accesible, facilitando el control del robot por parte del usuario. En la parte superior, se encuentra la sección de conexión, donde el operador puede seleccionar el dispositivo de control y establecer la comunicación con el robot. Este paso inicial asegura que el sistema esté preparado para recibir y ejecutar comandos. Además, se incluyen opciones para reconectar o reiniciar el sistema en caso de fallas en la comunicación, ofreciendo una mayor robustez operativa.

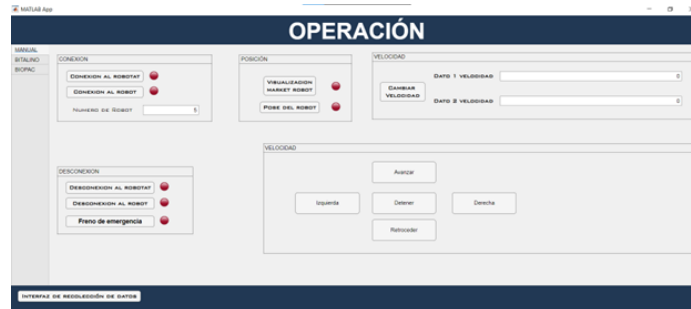
El control del movimiento del robot se lleva a cabo a través de dos secciones principales: la configuración de velocidades y la dirección. La configuración de velocidades permite ajustar individualmente los motores izquierdo y derecho, brindando al usuario un control preciso sobre la propulsión del robot. Esto es especialmente útil para evaluar el rendimiento de cada motor y garantizar que respondan de manera uniforme. Por otro lado, la sección de dirección incluye comandos básicos que permiten al robot avanzar, retroceder, girar a la izquierda o a la derecha, así como detenerse por completo. Estas funcionalidades ofrecen al operador la capacidad de evaluar cómo responde el robot a diferentes movimientos y ajustes.

Como medida de seguridad, la interfaz incluye un botón de paro de emergencia, diseñado para detener de inmediato todas las operaciones del robot en caso de movimientos

inesperados o errores. Este botón garantiza la protección tanto del robot como del entorno de trabajo, previniendo posibles daños durante las pruebas.

Permite al usuario evaluar el desempeño del robot físico, garantizar la conectividad del hardware y realizar ajustes necesarios para optimizar su funcionamiento. Este modo de prueba es un paso previo fundamental antes de pasar a operaciones más complejas o automatizadas. La Figura 90 ilustra la disposición de la interfaz, destacando la simplicidad y eficacia en su diseño, lo que facilita el control y la supervisión del sistema.

Figura 90: Interfaz de operación con el agente robótico.



## 11.2. Recepción de datos del *ESP32*

Se describe el proceso de recepción y procesamiento de señales digitales provenientes del sistema biopac utilizando la placa ESP32. El biopac envía señales digitales que representan movimientos oculares hacia la izquierda y la derecha, las cuales se capturan mediante los pines GPIO 27 (izquierda) y GPIO 13 (derecha) configurados como entradas digitales con resistencias pull-down. Esta configuración asegura que los pines permanezcan en un estado bajo por defecto, evitando fluctuaciones o ruido en ausencia de señales.

El ESP32 se encargó de procesar estas señales en tiempo real, asignando el valor digital 1 para indicar la detección de una señal activa, mientras que el valor digital 0 representa la ausencia de actividad. Estas señales fueron monitoreadas y visualizadas mediante el puerto serial, lo que permitió analizar el estado de las entradas y depurar el funcionamiento del sistema. El código implementado imprime constantemente el estado de los pines en el monitor serial, proporcionando una referencia clara de las señales capturadas.

Adicionalmente, el LED interno del ESP32 se utilizó como indicador visual del estado del sistema. Cuando se detectaba una señal activa en cualquiera de los pines, el LED se encendía, proporcionando una confirmación inmediata de la recepción de la señal. En caso de no detectarse señales en los pines, el LED permanecía apagado, indicando que el sistema estaba en reposo. Este enfoque simplifica la supervisión del sistema durante las pruebas y garantiza una respuesta confiable a las señales digitales del biopac.

El uso de señales digitales para representar movimientos oculares permitió una integración eficiente con el ESP32, sirviendo como puente entre el sistema biomédico biopac y otras plataformas, como Matlab o sistemas de control de robots.

## 11.3. Traducción de las señales a comandos de control

### 11.3.1. Biopac

El sistema biopac, previamente descrito en la Sección 11.2, se utilizó para capturar señales electrooculográficas (EOG) y transformarlas en señales digitales representativas de movimientos oculares. Estas señales, procesadas mediante umbrales predefinidos en el ESP32, se tradujeron en valores binarios: 1 para movimientos hacia la derecha y 0 para movimientos hacia la izquierda. Este procesamiento fue fundamental para simplificar la interacción entre el sistema biomédico y los comandos de control que se enviaron al robot.

Una vez que las señales digitales fueron obtenidas y validadas en el ESP32, estas se transmitieron a Matlab mediante comunicación serial. El código de Matlab se encargó de interpretar estas señales y traducirlas en comandos de control para el robot móvil. El flujo básico del procesamiento incluye los siguientes pasos:

Recepción de señales digitales en Matlab: las señales binarias enviadas desde el ESP32 fueron capturadas en Matlab mediante funciones de comunicación serial. Se procesaron en tiempo real, asegurando que los comandos se ejecutaran de forma inmediata al detectar un cambio en las señales.

Generación de comandos de control: si la señal recibida era 1 (derecha), el robot móvil ejecutaba un movimiento previamente configurado; si la señal era 0 (izquierda), el robot permanecía en reposo.

Ejecución en el robot móvil: los comandos de control generados en Matlab se transmitieron al robot físico, que reaccionó en tiempo real de acuerdo con las señales recibidas.

El flujo completo del proceso se presenta en el diagrama de flujo a continuación, el cual detalla la interacción entre el biopac, el ESP32, Matlab y el robot físico, en la figura 91 y en la figura 92 se visualiza el biopac físico conectado a un sujeto.

Figura 91: Diagrama de flujo de la configuración entre el biopac y el ESP32.

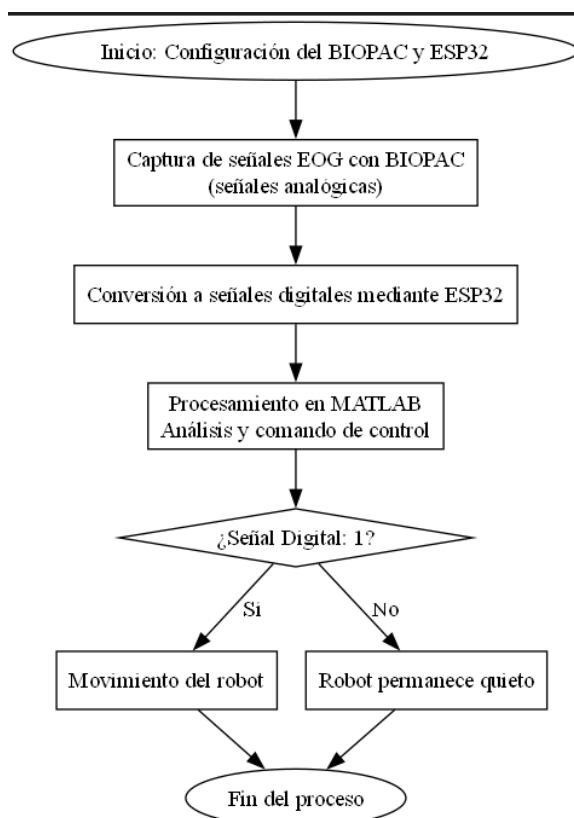
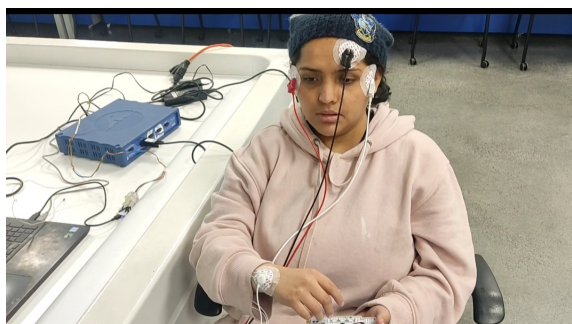


Figura 92: Visualización de la conexión del biopac.



### 11.3.2. Bitalino

El sistema implementado con bitalino traduce las señales electromiográficas (EMG) capturadas durante la ejecución de gestos en comandos específicos para el control de un robot móvil en tiempo real. Este flujo de trabajo abarca desde la captura inicial de señales hasta la ejecución de movimientos del robot en función de predicciones realizadas por modelos previamente entrenados y almacenados.

La adquisición de señales EMG se realiza a través del dispositivo bitalino, que captura la actividad eléctrica muscular del usuario. Estas señales se procesan mediante filtros pasa-

banda y notch, eliminando ruido e interferencias no deseadas, con el objetivo de obtener datos más limpios y fiables. Posteriormente, las señales se almacenan en un buffer circular, lo que permite un análisis continuo en tiempo real.

Las características extraídas de las señales procesadas incluyen métricas como la Media Absoluta de Valores (MAV) y el número de cruces por cero (ZC). Estas características se utilizan como entradas para modelos de predicción previamente entrenados y almacenados en un archivo .mat. Cabe destacar que no solo se empleó la Máquina de Vectores de Soporte (SVM), sino también otros modelos como redes neuronales artificiales (ANN), el algoritmo de los  $k$  vecinos más cercanos (*k-Nearest Neighbors*, *KNN*) y modelos lineales. Estos modelos fueron entrenados con un conjunto de datos representativo de los gestos deseados, y sus resultados se guardaron para ser reutilizados durante el control del sistema.

El proceso de predicción comienza cargando el modelo seleccionado desde el archivo .mat. Una vez cargado, el sistema analiza las características extraídas y realiza la predicción correspondiente. Cada modelo genera una predicción basada en el gesto detectado, la cual se traduce en un comando específico para el robot móvil:

Clase 1 (Avanzar): El robot se mueve hacia adelante a una velocidad constante.

Clase 2 (Girar izquierda): El robot ajusta sus ruedas para girar a la izquierda.

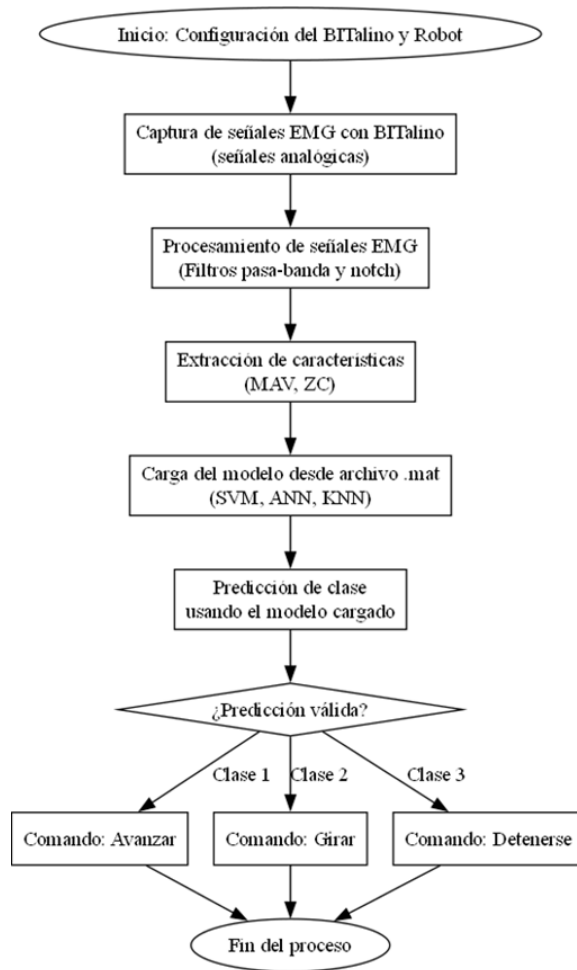
Clase 3 (Girar derecha): El robot realiza un giro hacia la derecha.

Clase 4 (Detenerse): El robot se detiene por completo.

Estos comandos son enviados al robot móvil a través de una conexión con el sistema Robotat, que permite controlar directamente las velocidades de las ruedas o detener el robot según sea necesario. Este enfoque asegura que el robot reaccione de manera precisa y eficiente a las señales del usuario.

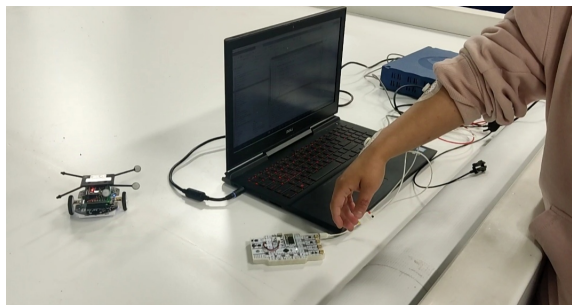
El diagrama de flujo en la Figura 93 ilustra este proceso completo, destacando cada etapa desde la adquisición de señales hasta el control final del robot móvil, incluyendo el almacenamiento y uso de modelos predictivos entrenados.

Figura 93: Diagrama de flujo de la configuración entre el bitalino y el robot móvil.



En la figura 94, se tiene como resultado la conexión del bitalino con el robot móvil.

Figura 94: Visualización de la conexión del bitalino.



- Se desarrolló un flujo robusto de adquisición de señales EMG, que incluye el filtrado y la segmentación en tiempo real, que permite detectar gestos de forma eficiente y almacenar características en una base de datos para análisis posteriores.
- La integración del sistema biopac con el ESP32 permitió una conversión precisa de señales EOG analógicas a digitales, logrando un control en tiempo real de sistemas físicos, como robots móviles.
- Los experimentos mostraron que el Kernel gaussiano tuvo un mejor desempeño en la clasificación de gestos multclasificados en comparación con el Kernel polinomial, alcanzando niveles satisfactorios de precisión.
- La implementación del sistema en Webots evidenció la funcionalidad de los algoritmos de control y clasificación.
- La división de las señales en segmentos permitió extraer características específicas, como *MAV* y *Zero Crossing*, mejorando la capacidad de clasificación y el entrenamiento de los modelos.
- La validación por medio de comunicación TCP/IP con Webots permitió determinar que el envío de datos fuera correcto utilizando una placa ESP32 que se encarga de manejar los comandos a utilizar en Webots

- Realizar experimentos con una mayor cantidad de sujetos para evaluar la robustez y generalización del sistema frente a variaciones individuales en las señales EMG y EOG.
- Implementar un mecanismo que permita mantener las acciones del robot en ejecución mientras se detectó una señal activa (EMG o EOG). Esto solucionará el comportamiento actual en el que el carrito solo realiza la acción una vez y luego se detiene, optimizando el control en tiempo real.
- Considerar Python como una alternativa para el desarrollo de los algoritmos y las interfaces debido a su mayor flexibilidad y eficiencia en comparación con Matlab, especialmente en lo relacionado con App Designer. Python también ofrece bibliotecas avanzadas para procesamiento de señales y desarrollo de interfaces gráficas.
- Ampliar el sistema para incluir más clases de gestos o acciones, lo que permitirá diversificar las aplicaciones del sistema y aumentar su funcionalidad.
- Realizar pruebas en entornos más complejos con escenarios realistas para validar la funcionalidad del sistema fuera de condiciones controladas.

- [1] L. F. Morales, “Modelo virtual de prótesis de mano controlada mediante señales EEG,” Tesis de licenciatura, Centro de investigación y de estudios avanzados del instituto politécnico nacional, 2022.
- [2] D. G. Calvo, “Teleoperación de un robot terrestre mediante biopotenciales.,” Tesis de licenciatura, Universidad Carlos III de Madrid, 2008.
- [3] M. F. Girón, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EMG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [4] R. Ralda, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EEG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [5] R. J. Cáceres, “Interfaz para el Control de Sistemas Robóticos Utilizando Combinaciones de Señales Biomédicas,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [6] L. Sornmo y P. Laguna, *Bioelectrical signal processing in caridac and neurological applications*. Elsevier Academic Press, 2005.
- [7] J. M. Avila y F. F. Fajardo, “Análisis entre señales electromiográficas en los músculos cervicales y el ángulo de inclinación de la cabeza en niños con PCI entre 5 y 10 años aplicada a terapia cervical,” Tesis de licenciatura, Universidad Politécnica Salesiana Sede Cuenca, 2019.
- [8] J. C. M. Contreras, “Análisis, implementación y evaluación de prácticas con el equipo Biopac BSL MP45 para el laboratorio de ingeniería biomédica de la UPS,” Tesis de licenciatura, Universidad Politécnica Salesiana, Cuenca, Ecuador, 2014.
- [9] B. Mokhlesabadifarahani y V. J. Gunjan, *EMG Signals Characterization in Three States of Contraction by Fuzzy Network and Feature Extraction*. India: Springer, 2015.
- [10] F. J. Carrasco, “Manual de conexión y uso de BITalino: Una plataforma para la adquisición de señales biomédicas,” Universidad Adolfo Ibáñez, Facultad de Ingeniería y Ciencias.
- [11] R. Merletti y D. Farina, *Surface electromyography: Physiology, engineering, and applications*. 2016.

- [12] I. BIOPAC Systems, “Procedimientos para la colocación de electrodos EOG con el sistema Biopac.”
- [13] I. BIOPAC Systems, “Application Note 292: Interfacing with E-Prime®,” 2022.
- [14] A. Moreno, E. Armengol, J. Béjar et al., *Aprendizaje Automático*. Barcelona, España: Edicions UPC, 1994, Edicions de la Universitat Politècnica de Catalunya, SL, ISBN: 84-7653-460-4.
- [15] I. C. Education. “¿Qué es el aprendizaje supervisado?” (2023), dirección: <https://www.ibm.com/es-es/topics/supervised-learning>.
- [16] R. O. Duda, P. E. Hart y D. G. Stork, *Pattern Classification*, 2nd. New York: Wiley, 2001.
- [17] S. Guevara, E. Robalino, A. Bouchet, M. Brun y V. Ballarin, “Diseño automático de un clasificador para filtrado de ruido en imágenes binarias utilizando análisis discriminante lineal,” es, *Revista de Investigación en Ingeniería Electrónica*, vol. 4, n.º 1, págs. 1-10, 2021, Disponible en: <http://reddi.unlam.edu.ar>. dirección: <http://reddi.unlam.edu.ar>.
- [18] A. Unknown, *Bayesian Learning*, Lecture notes on Bayesian Learning in Machine Learning, 2024. dirección: <https://example.com/bayesian-learning>.
- [19] E. D. Monroy Jordan, J. E. Pérez Neira y E. Gómez Vásquez, *Máquinas de Soporte Vectorial (SVM)*. Universidad Tecnológica de Bolívar, 2005.
- [20] E. Campo León, “Introducción a las Máquinas de Soporte Vectorial (SVM),” TFG-2057, Universidad de Castilla-La Mancha, 2016.
- [21] A. Zubiaga, “Análisis y Aplicación de Estrategias Multiclase en Máquinas de Vectores de Soporte,” Master’s Thesis, Universidad del País Vasco, Bilbao, España, 2008. dirección: <https://ruta-del-documento.com>.
- [22] A. Ng y T. Ma, “CS229 Lecture Notes: Kernel Methods,” dirección: <https://cs229.stanford.edu/notes/cs229-notes3.pdf>.
- [23] X. Basogain, *Redes Neuronales Artificiales y sus Aplicaciones*. Escuela Superior de Ingeniería de Bilbao, EHU, 2015, Página 20, Referencia para redes neuronales artificiales y elementos procesadores.
- [24] P. J. García-Laencina, R. Verdu-Monedero, J. Larrey-Ruiz, J. Morales-Sánchez y J.-L. Sancho-Gómez, “Algoritmo KNN basado en Información Mutua para Clasificación de Patrones con Valores Perdidos,” en *Proceedings of the Congreso Nacional de Tecnologías de la Información y las Comunicaciones*, Financiado parcialmente por el Ministerio de Educación y Ciencia, proyecto TEC2006-13338/TCM., Universidad Politécnica de Cartagena, Cartagena, Murcia, España, 2010. dirección: <https://upct.es>.
- [25] P. Corporation, *Pololu Maestro Servo Controller User’s Guide*, Guía oficial para el controlador de servos Pololu Maestro., 2024. dirección: <https://www.pololu.com/docs/0J83>.
- [26] IMEPI, *¿Qué es una interfaz hombre-máquina (HMI)?* Última consulta: 19 de noviembre de 2024, 2024. dirección: <https://imepi.com.mx/que-es-una-interfaz-hombre-maquina-hmi/>.
- [27] L. Jaulin, *Mobile Robotic*. Hisham Abou-Kandil, 2013.