Universidad del Valle de Guatemala
Inteligencia Artificial – CC3085
Sección 10
26 de Mayo de 2025

Clasificación de Movimientos con Señal EMG

Elda Daniela Navas Cinto 211000,
nav211000@uvg.edu.gt

DESCRIPCIÓN DEL PROBLEMA

La amputación transradial de un miembro superior, es decir por debajo del codo, representa una pérdida significativa en la funcionalidad del brazo, afectando directamente la autonomía y calidad de vida de la persona. Entre las principales consecuencias se encuentran la pérdida de habilidades motoras finas, disminución en la capacidad de agarre y manipulación de objetos, y la necesidad de readaptación para realizar tareas cotidianas [1].

Ante este escenario, el desarrollo de prótesis de miembro superior se ha convertido en una solución fundamental para restaurar parcialmente la funcionalidad perdida. Sin embargo, muchas prótesis actuales no ofrecen movimientos precisos o personalizados, lo que limita su utilidad práctica. Una de las principales necesidades en este campo es diseñar sistemas de control que permitan a las prótesis responder de manera natural e intuitiva a las intenciones del usuario, por ejemplo, mediante el uso de señales musculares y modelos de aprendizaje automático [1].

ANÁLISIS DEL PROBLEMA

La pérdida de una extremidad superior, particularmente por debajo del codo, representa una de las discapacidades físicas más desafiantes, ya que limita significativamente la autonomía de la persona en tareas cotidianas. Si bien las prótesis mioeléctricas han avanzado considerablemente, su integración funcional todavía presenta múltiples retos. Uno de los principales es el diseño de un sistema de control eficiente y personalizado que permita interpretar de forma precisa las intenciones de movimiento del usuario a partir de señales electromiográficas (EMG) [2].

Las señales EMG reflejan la actividad eléctrica generada por la activación de las unidades motoras en los músculos esqueléticos. Estas señales son altamente variables entre individuos debido a factores como la anatomía muscular, la distribución de fibras, el nivel de entrenamiento o tono muscular, y la ubicación de los electrodos. Esta variabilidad hace que los sistemas de control basados en modelos genéricos o preentrenados presenten limitaciones al ser implementados en usuarios reales, ya que no logran captar con precisión los patrones específicos de activación muscular [2].

En la mayoría de prótesis actuales, el control aún se basa en estrategias limitadas como el control binario (abrir/cerrar), con bajos niveles de personalización. Esto no solo

reduce la capacidad funcional de la prótesis, sino que también genera frustración en el usuario y un índice elevado de rechazo. Para superar estas limitaciones, es necesario desarrollar sistemas capaces de identificar múltiples patrones de movimiento, con alta precisión, y adaptados al perfil electromiográfico único de cada persona [3].

En este contexto, el uso de modelos de machine learning aplicados al análisis de señales EMG representa una alternativa prometedora [4]. Estos modelos pueden ser entrenados para reconocer patrones complejos y sutiles en la señal, permitiendo clasificar distintos movimientos de la mano como cerrar el puño, extender la palma o hacer una pinza. No obstante, para que estos modelos funcionen adecuadamente, es esencial contar con datos de calidad, una correcta extracción de características, y un entrenamiento específico para cada usuario [5].

PROPUESTA DE SOLUCIÓN

Se propone el desarrollo de un sistema de **clasificación** de movimientos de mano basado en señales de electromiografía (EMG) adquiridas mediante electrodos de superficie ubicados en el antebrazo. El objetivo principal es identificar, de forma precisa patrones musculares asociados a movimientos funcionales como cerrar el puño, extender la palma y realizar una pinza entre el pulgar y el índice. Esto con la finalidad de que esta clasificación permita su aplicación en el control de prótesis mioeléctrica del miembro superior, específicamente para personas con amputación transradial [1].

A través de este sistema, se busca facilitar una interfaz biológica que traduzca la actividad eléctrica muscular en comandos específicos que puedan ser utilizados por una prótesis. De esta manera, se contribuye a mejorar la funcionalidad y adaptabilidad del dispositivo protésico, ofreciendo una solución personalizada y potencialmente más accesible para personas con pérdida parcial del brazo [1].

DESCRIPCIÓN DE SOLUCIÓN

Para lograr el objetivo, se utilizarán datos propios obtenidos directamente mediante el sistema BIOPAC, empleando tres electrodos superficiales colocados en el antebrazo. La adquisición de señales personales cumple dos propósitos fundamentales: simular el entrenamiento de una prótesis mioeléctrica adaptada a un usuario específico. Así como, resaltar la importancia de la personalización, dado que los patrones de activación muscular son únicos para cada individuo, lo que refuerza la necesidad de desarrollar modelos individualizados en aplicaciones reales de control protésico [6].

En el contexto biomédico, cada paciente presenta variaciones en la estructura anatómica, fuerza y características de activación muscular. Por ello, resulta poco eficiente diseñar un sistema de control único que funcione universalmente. La

personalización a través del entrenamiento con datos propios permite mejorar la precisión, el control y la utilidad funcional de las prótesis, haciendo que los movimientos resultantes sean más naturales y adaptados a las necesidades del usuario [7].

Se seleccionan **tres movimientos específicos**: cerrar el puño, extender la palma y pinza con el dedo índice y pulgar, lo que responde a cierta importancia funcional en actividades diarias. El movimiento de cerrar el puño permite sujetar objetos de distintos tamaños con fuerza y estabilidad, mientras que abrir la palma facilita acciones como soltar o recibir elementos. Por otro lado, el gesto de pinza es particularmente significativo desde el punto de vista evolutivo y funcional, ya que representa una capacidad distintiva del ser humano en comparación con la mayoría de los animales. Esta pinza implica coordinación fina entre el pulgar y el índice, lo que permite manipular objetos pequeños, escribir, abotonar una camisa o utilizar ciertos tipos de herramientas. Incluir estos tres movimientos permite cubrir tanto funciones de fuerza como de precisión, representando así una base esencial para el control eficaz y versátil de una prótesis de mano y un buen primer acercamiento que puede luego evolucionar.

Finalmente, se utilizará un modelo de clasificación de *Machine Learning* que será entrenado con las características extraídas de las señales EMG, tanto en dominio de tiempo como en dominio de frecuencia. Este modelo tendrá como finalidad identificar los diferentes tipos de movimiento realizados por el usuario. Se evaluará su desempeño a través de métricas como la precisión, sensibilidad y especificidad, permitiendo validar la efectividad del sistema para su posible futura implementación en una prótesis funcional [8].

HERRAMIENTAS APLICADAS

PRINCIPIO FISIOLÓGICO

El músculo esquelético humano está formado por cientos de fibras musculares con forma cilíndrica, las cuales se agrupan mediante tejido conectivo. La contracción de estas fibras se genera cuando el sistema nervioso central, a través de las neuronas motoras somáticas, envía impulsos eléctricos desde el encéfalo o la médula espinal hacia los músculos esqueléticos. Cada axón de una neurona motora puede ramificarse para inervar múltiples fibras musculares, pero cada fibra muscular está controlada exclusivamente por una sola neurona motora. La unidad funcional básica del sistema neuromuscular es la **unidad motora**, compuesta por una neurona motora y todas las fibras musculares que esta controla (Figura 1) [1].

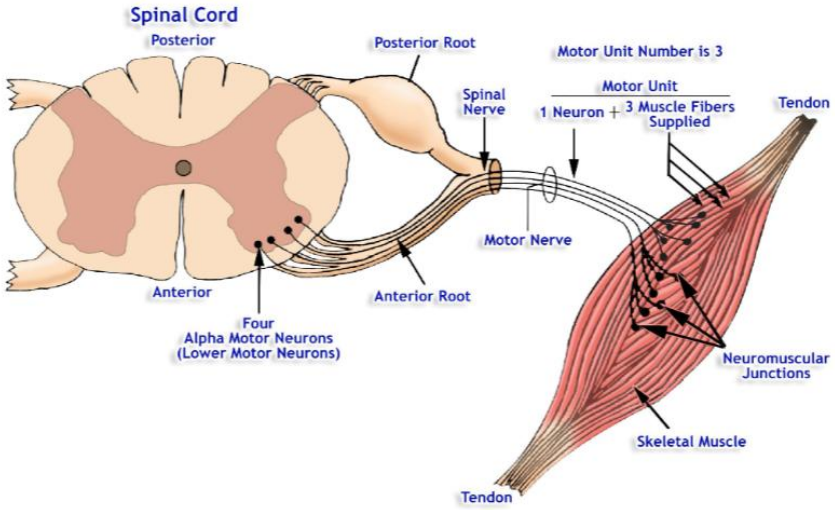

Figura 1. Unidad Motora

El grado de contracción muscular se regula mediante dos mecanismos principales: la activación de un número determinado de unidades motoras (reclutamiento) y el control de la frecuencia con la que estas unidades son estimuladas. Este proceso permite generar movimientos suaves y controlados, así como ajustar la fuerza muscular en función de la tarea a realizar. Incluso en reposo, los músculos presentan un leve estado de contracción denominado **tono muscular**, que mantiene la musculatura en un estado de preparación para la acción [6].

La señal de EMG representa el resultado de dos actividades bioeléctricas principales: la propagación de impulsos por los nervios motores y su transmisión en la unión neuromuscular, y la posterior propagación del potencial de acción a través del sarcolema y el sistema tubular T, que desencadena el proceso de contracción. A pesar de que las señales individuales generadas por cada fibra son de muy baja amplitud (menores a 100 μV), la activación simultánea de múltiples fibras genera una señal suficientemente grande como para ser detectada por electrodos de superficie [6].

La señal EMG, por tanto, es el reflejo de una suma compleja de múltiples potenciales de acción, cada uno con distinta magnitud y fase, y su morfología dependerá de la posición de los electrodos, del número de fibras activas y del nivel de fatiga muscular. Por ende, representa la **superposición de múltiples sinusoides**, esta complejidad genera una señal caótica a simple vista, pero que encierra patrones que pueden ser interpretados mediante técnicas de procesamiento de señales y aprendizaje automático. Las **frecuencias típicas** de la señal EMG se encuentran principalmente entre **20 Hz y 500 Hz**, siendo este el rango donde se concentra la mayor parte de la información muscular útil [3].

ADQUISIÓN DE SEÑAL DE ELECTROMIOGRAMA

Para la adquisición de la señal se utiliza el sistema **BIOPAC Student Lab**, un dispositivo especializado en el registro y análisis de bioseñales fisiológicas. Este sistema cuenta con una estructura modular que permite la conexión de múltiples tipos de sensores a través de cuatro canales analógicos, facilitando su uso en contextos experimentales controlados.

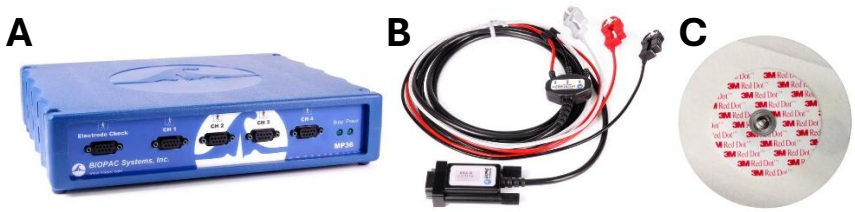

Figura 2. Sistema BIOPAC (A), Set de Electrodos de 3 Vías (B) y Electrodo Superficial (C)

En este experimento, se utilizó un set de electrodos superficiales de tres vías: tierra (GND), positivo (V+) y negativo (V-), junto con tres electrodos adhesivos desechables (Figura 2). El set fue conectado al canal 1 (CH1) del BIOPAC (Figura 3) y los electrodos se colocaron sobre el antebrazo del sujeto, siguiendo la orientación de las fibras musculares y según la configuración ilustrada en la Figura 4. El sistema mide primero el voltaje entre el electrodo positivo y tierra, y luego entre el electrodo negativo y tierra. Posteriormente, se calcula la diferencia entre ambas señales, permitiendo obtener el potencial diferencial real entre los electrodos positivo y negativo. La presencia del electrodo de referencia es esencial, ya que proporciona un punto de comparación estable y reduce la interferencia común a ambos canales, mejorando así la relación señal/ruido.

Figura 3. Montaje BIOPAC y Set de Electrodos de 3 Vías en Canal 1 (CH1)

Dado el alcance experimental del proyecto, se optó por el uso de electrodos superficiales ubicados en el antebrazo. Sin embargo, en aplicaciones clínicas o de investigación avanzada, la señal EMG también puede obtenerse desde puntos más cercanos al origen del impulso nervioso, como el encéfalo mediante electrocorticografía (ECoG) o incluso a nivel espinal con técnicas invasivas. A pesar de estas alternativas, para el objetivo de simular el control de una prótesis funcional, la señal obtenida superficialmente en el antebrazo es suficiente, pues refleja de manera directa la actividad de los músculos responsables del movimiento de la mano.

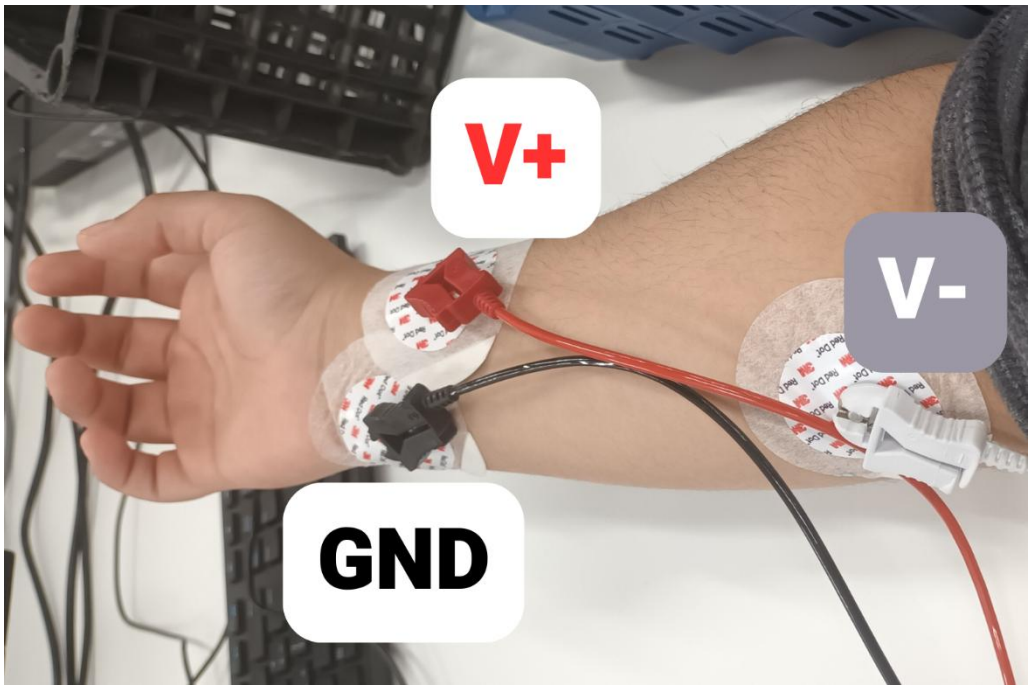

Figura 4. Colocación de los Electrodos en el antebrazo

El software de BIOPAC permite configurar diversos parámetros de adquisición. En este caso, se utilizó una **frecuencia de muestreo de 2000 Hz (2 kHz).** Se selecciona esta frecuencia basado en el contenido espectral típico de la señal EMG, que se encuentra principalmente entre 20 Hz y 500 Hz. Además de que cumple con el teorema de muestreo de Nyquist, el cual establece que la frecuencia de muestreo debe ser al menos el doble de la máxima frecuencia contenida en la señal, garantizando así una reconstrucción adecuada sin aliasing (Figura 5) [9].

Adicionalmente, se configuraron filtros para mejorar la calidad de la señal: **un filtro pasa altas de 20 Hz** para eliminar componentes de muy baja frecuencia, eliminar el desplazamiento de DC (causado por la impedancia de la piel, que actúa como una barrera eléctrica) y posibles ruidos por movimiento. También se utilizó un **filtro pasa bajas de 500 Hz** para descartar frecuencias superiores al rango útil, y finalmente un **filtro Notch de 60 Hz** que atenúa la interferencia proveniente de fuentes eléctricas residenciales. Todos estos filtros fueron implementados con un **factor de calidad (Q)**

de 0.707, lo cual ofrece un buen compromiso entre atenuación y estabilidad, evitando distorsiones bruscas. Estos filtros aseguran que se conserven únicamente las **frecuencias de interés fisiológico**, facilitando el análisis posterior y reduciendo el ruido artefactual (Figura 6) [10].

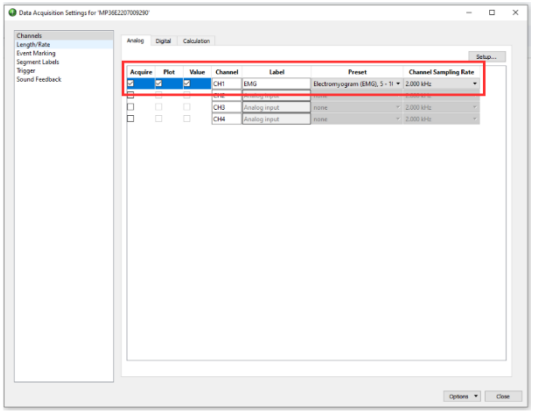

Figura 5. Frecuencia de Muestreo en BIOPAC

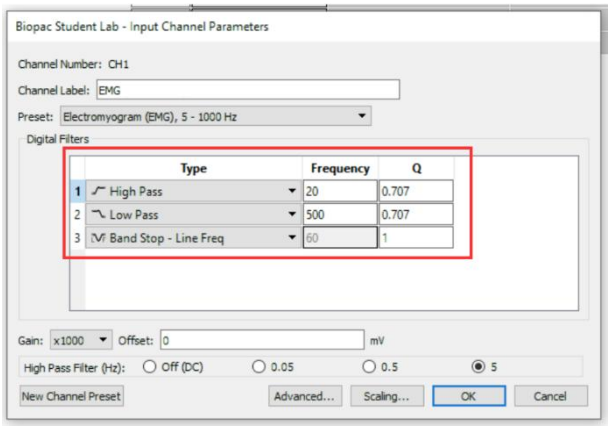

Figura 6. Configuración de Filtros en Sistema en BIOPAC

Se realizaron cuatro conjuntos de datos distintos: uno correspondiente a cada uno de los tres movimientos seleccionados (puño, palma abierta y pinza), y uno adicional en condición de reposo, utilizado como referencia comparativa [11].

Para la elaboración de cada conjunto, se diseñó un protocolo de adquisición basado en **ventanas temporales de 5 segundos**, en las cuales el sujeto realizaba el movimiento correspondiente. Estas ventanas se intercalaban con periodos de reposo de 5 segundos, dando lugar a ciclos de 10 segundos (5s de reposo + 5s de movimiento). Este ciclo fue repetido un total de 150 veces, lo que resulta en 150 ventanas de movimiento, distribuidas a lo largo de 1500 segundos de grabación continua por clase de movimiento (Figura 7). Durante la adquisición del movimiento de puño, se incluyeron pausas completas y tiempos de descanso prolongados entre repeticiones con el fin de minimizar el efecto de la fatiga muscular, que podría introducir sesgos (bias) no deseados en las características extraídas de la señal EMG [8].

En el caso del conjunto de datos de reposo, no se intercalaron fases activas, por lo que se registró una señal continua de 750 segundos, compuesta únicamente por 150 ventanas de reposo de 5 segundos cada una [8].

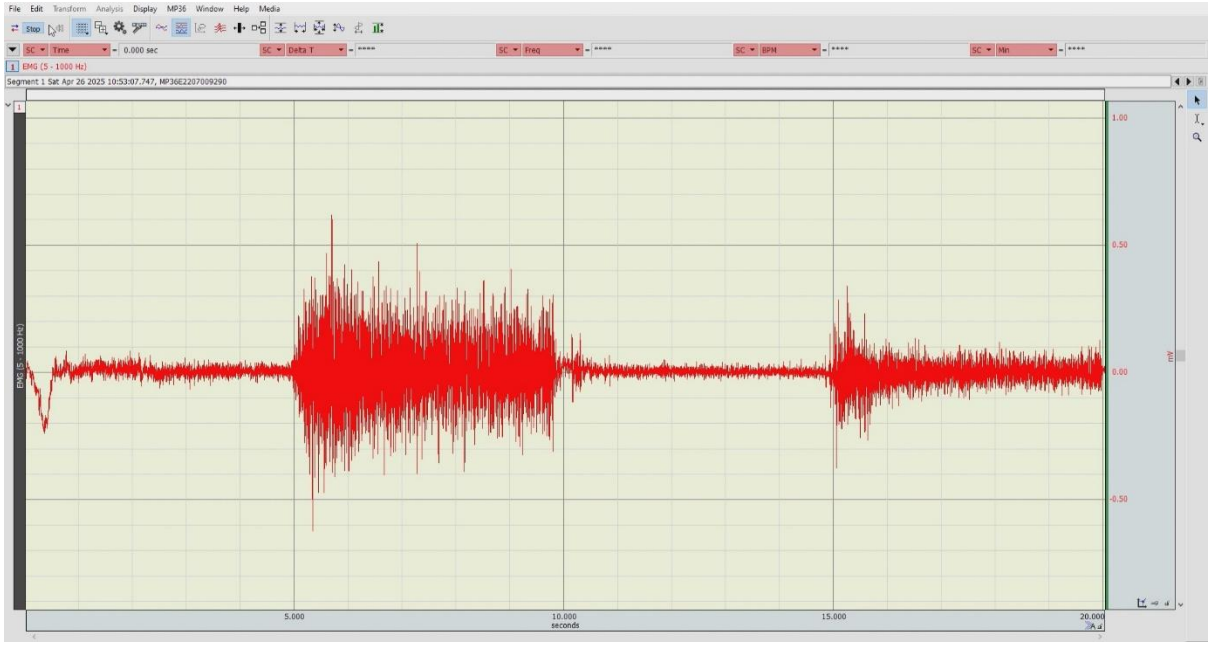

Figura 7. Adquisición de datos en entorno BIOPAC

PROCESAMIENTO DE LA SEÑAL

El sistema BIOPAC permite exportar los datos adquiridos en archivos con formato .mat, los cuales contienen tanto el vector de la señal EMG como metadatos relevantes del experimento. Este formato es nativo del entorno MATLAB, lo que facilita su procesamiento gracias a las capacidades especializadas que ofrece el software para análisis de señales biológicas, incluyendo funciones matemáticas avanzadas, visualización de datos y tratamiento de ruido.

En MATLAB, se inicia limpiando el entorno, luego se definen los parámetros clave: una frecuencia de muestreo de 2000 Hz y una ventana temporal de 5 segundos, lo que da como resultado 10,000 muestras por ventana. Se diseña un filtro Butterworth pasabanda de cuarto orden, con frecuencias de corte en 20 y 500 Hz, para conservar únicamente las componentes fisiológicamente relevantes de la señal.

```
% Limpiar espacio
close all; clear; clc;

% Definir parámetros
Fs = 2000;                % Frecuencia de muestreo (Hz)
tamano_ventana = 5;       % s
N = Fs * tamano_ventana;  % Número de Muetras

% Filtrado Señal
f_low = 20;  % Frecuencia de corte inferior (20 Hz)
f_high = 500;  % Frecuencia de corte superior (500 Hz)
Wn = [f_low f_high] / (Fs / 2);  % Frecuencias normalizadas
[b, a] = butter(4, Wn, 'bandpass');  % Diseñar el filtro Butterworth pasabanda Orden 4
```

Posteriormente, se cargan los archivos .mat correspondientes a cada uno de los movimientos (reposo, palma, pinza y puño), y se extrae el vector de datos de cada estructura. A partir de la cuarta ventana de movimiento, se define un segmento de interés de 5 segundos, el cual se filtra usando el filtro previamente diseñado.

```
% Cargar datos (variable 'data')
reposo = load('Reposo.mat', 'data');
palma = load('Palma.mat', 'data');
pinza = load('Pinza.mat', 'data');
puno = load('Puño.mat', 'data');

% Extraer la variable 'data' de cada estructura
reposo = reposo.data;
palma = palma.data;
pinza = pinza.data;
puno = puno.data;

% Definir inicio y fin del primer movimiento
inicio_movimiento = ( Fs * tamano_ventana * 3) + 1;
fin_movimiento = inicio_movimiento + N - 1;

% Cortar y filtrar segmentos de interés
reposo_segmento = filter(b, a, reposo(inicio_movimiento:fin_movimiento));
palma_segmento = filter(b, a, palma(inicio_movimiento:fin_movimiento));
pinza_segmento = filter(b, a, pinza(inicio_movimiento:fin_movimiento));
puno_segmento = filter(b, a, puno(inicio_movimiento:fin_movimiento));
```

Se genera un vector de tiempo que va de 0 a 5 segundos con 10,000 puntos, útil para graficar la señal en el dominio temporal. Finalmente, se usa la función de Plot para generar una figura con los 4 tipos de movimientos. Este procedimiento permite observar una ventana representativa y asegurarse de que el filtrado se haya aplicado correctamente antes de avanzar en el análisis o la extracción de características.

```
% Crear vector de tiempo de 0 a 5 segundos
t = linspace(0,5,N);

% Graficar Señal
figure(1);

subplot(2,2,1);
plot(t, reposo_segmento, 'Color', [0 0.4470 0.7410]);
title('Reposo', 'FontSize', 14);
xlabel('Tiempo (s)');
ylabel('Amplitud (mV)');
ylim([-1 1]);

subplot(2,2,2);
plot(t, palma_segmento, 'Color', [0.8500 0.3250 0.0980]);
title('Palma Extendida', 'FontSize', 14);
xlabel('Tiempo (s)');
ylabel('Amplitud (mV)');
ylim([-1 1]);

subplot(2,2,3);
plot(t, pinza_segmento, 'Color', [0.4660 0.6740 0.1880]);
title('Pinza', 'FontSize', 14);
```

```
xlabel('Tiempo (s)');
ylabel('Amplitud (mV)');
ylim([-1 1]);

subplot(2,2,4);
plot(t, puno_segmento, 'Color', [0.4940 0.1840 0.5560]);
title('Puño Cerrado', 'FontSize', 14);
xlabel('Tiempo (s)');
ylabel('Amplitud (mV)');
ylim([-1 1]);

sgtitle('Patrones Musculares', 'FontSize', 28, 'FontWeight', 'bold');
```

Dado que la señal fue adquirida en ventanas de 5 segundos, se procede a graficar una ventana de actividad como ejemplo ilustrativo. Para ello, se definen los puntos de inicio y fin de una ventana y se extrae únicamente ese segmento de señal (Figura 8).

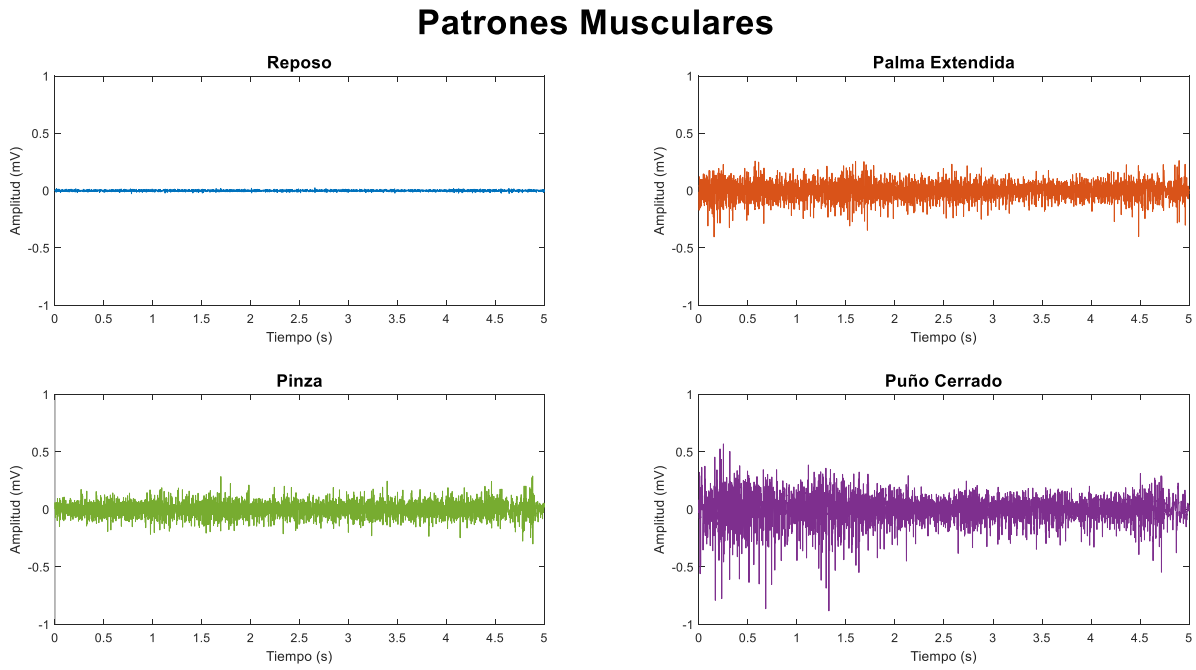

Figura 8. Patrones Musculares | Ejemplo una ventana

Posteriormente, se realiza el análisis en el dominio de la frecuencia mediante la aplicación de la Transformada Rápida de Fourier (FFT). Esta herramienta permite descomponer la señal en sus componentes sinusoidales fundamentales, revelando las frecuencias presentes en la señal y su amplitud relativa. Como resultado de la transformada, se obtiene un espectro simétrico respecto al eje central debido a la naturaleza compleja de la FFT (parte real e imaginaria). Sin embargo, en aplicaciones de señales reales como la EMG, solo el espectro unilateral (la mitad positiva del espectro) es de interés, ya que contiene toda la información no redundante. Este se obtiene extrayendo únicamente los valores reales del espectro y conservando la mitad del vector de salida.

Para ello en MATLAB se aplica la Transformada Rápida de Fourier (FFT) a cada uno de los segmentos filtrados de señal EMG, con el fin de analizar su contenido en el dominio de la frecuencia. Primero, se construye un vector de frecuencias que corresponde a los puntos del espectro unilateral, es decir, desde 0 Hz hasta la mitad de la frecuencia de muestreo (1000 Hz en este caso), ya que la FFT de señales reales es simétrica. Luego, se calcula la FFT para cada señal (reposo, palma, pinza y puño) y se normaliza dividiendo por el número total de muestras para obtener la magnitud real. Posteriormente, se extrae únicamente la mitad del espectro (espectro unilateral) y se duplican todos los valores excepto el primero y el último, para conservar la energía total original de la señal.

```
%% TRANSFORMADA DE FOURIER
f = Fs*(0:(N/2))/N; % Vector de frecuencias

% FFT
Y_reposo = fft(reposo_segmento);
Y_reposo = abs(Y_reposo/N);
Y_reposo = Y_reposo(1:N/2+1);
Y_reposo(2:end-1) = 2*Y_reposo(2:end-1);

Y_palma = fft(palma_segmento);
Y_palma = abs(Y_palma/N);
Y_palma = Y_palma(1:N/2+1);
Y_palma(2:end-1) = 2*Y_palma(2:end-1);

Y_pinza = fft(pinza_segmento);
Y_pinza = abs(Y_pinza/N);
Y_pinza = Y_pinza(1:N/2+1);
Y_pinza(2:end-1) = 2*Y_pinza(2:end-1);

Y_puno = fft(puno_segmento);
Y_puno = abs(Y_puno/N);
Y_puno = Y_puno(1:N/2+1);
Y_puno(2:end-1) = 2*Y_puno(2:end-1);
```

Igualmente, se usa Plot para graficar y en este espectro, el eje X representa las frecuencias en Hz, mientras que la altura de los picos indica la magnitud (amplitud) de la señal en esa frecuencia específica. Gracias a este análisis espectral, es posible observar características distintivas en cada tipo de movimiento muscular.

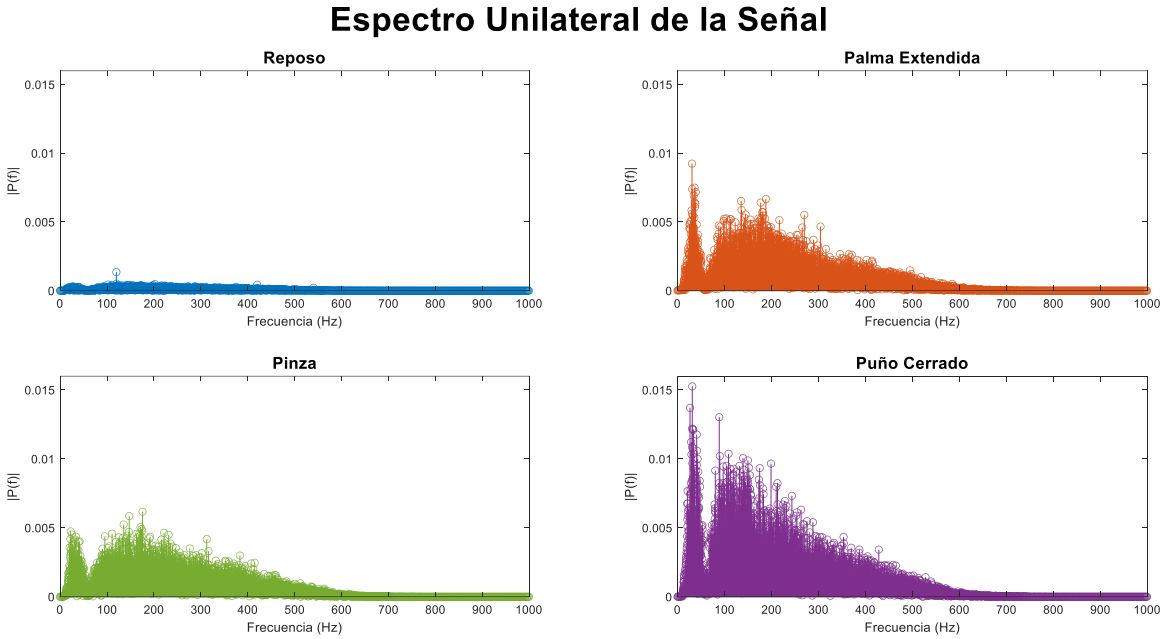

Figura 9. Espectro Unilateral de la Señal | Ejemplo una ventana

Las figuras de Fourier (Figura 9) generadas para cada señal evidencian que los filtros configurados durante la adquisición fueron aplicados correctamente, ya que no se detectan componentes por fuera del rango de interés (20–500 Hz). Además, se aprecian diferencias significativas en las amplitudes y frecuencias dominantes de cada movimiento: el puño cerrado presenta las mayores amplitudes, seguido de la palma extendida, y por último la pinza, que muestra amplitudes más bajas. En contraste, la señal de reposo se muestra prácticamente silenciosa, con excepción de un pico menor alrededor de 120 Hz, probablemente atribuible a ruido residual o a alguna tensión basal mínima.

Este análisis espectral no solo valida el correcto funcionamiento del sistema de adquisición y filtrado, sino que también permite anticipar la separabilidad de clases en un futuro sistema de clasificación, dado que cada tipo de movimiento exhibe un patrón de frecuencia característico.

EXTRACCIÓN DE CARACTERÍSTICAS Y CONSTRUCCIÓN DE DATASET

Para construir el dataset que servirá de base para el modelo de clasificación, se implementó un proceso automatizado de **extracción de características** en MATLAB. El procedimiento se aplicó sobre las señales previamente adquiridas para los tres tipos de movimiento (puño cerrado, palma extendida y pinza), así como para el estado de reposo.

El algoritmo de procesamiento se diseñó para recorrer las señales en ventanas de 5 segundos, pero solo se procesan las ventanas correspondientes a actividad (evitando las de reposo intercaladas). Cada dos ventanas, se extraen las características de la

señal en una de ellas, generando así 150 segmentos de análisis por clase. Para el caso de la clase de reposo, el procedimiento es similar, pero dado que no hay alternancia con actividad, se procesan todas las ventanas (también 150 en total).

```
%% DATASET PALMA EXTENDIDA
% Inicializar dataset para Palma
dataset_palma = [];

num_ventanas = floor(length(palma) / N); % Número de ventanas

% Recorrer cada 2 ventanas para extraer características
for i = 1:2:num_ventanas
    ventana = palma((i-1)*N+1:i*N); % Extraer la ventana de 5 segundos

    % Extraer características en dominio de tiempo
    MAV = mean(abs(ventana));      % Mean Absolute Value
    RMS = sqrt(mean(ventana.^2));  % Root Mean Square
    V = var(ventana);              % Varianza
    WL = sum(abs(diff(ventana)));  % Waveform Length
    SSC = sum(diff(ventana) > 0);  % Slope Sign Changes
    IEMG = sum(abs(ventana));      % Integral of EMG
    LOGVAR = log(var(ventana));  % Log-variance

    % Transformada de Fourier FFT
    Y = fft(ventana);           % Transformada de Fourier
    Y = abs(Y/N);               % Magnitud normalizada
    Y = Y(1:N/2+1);             % Solo el espectro positivo
    Y(2:end-1) = 2*Y(2:end-1);  % Escala adecuada

    % Extraer características en dominio de frecuencia usando FFT
    PF = f(find(Y == max(Y), 1));  % Peak Frequency
    TP = sum(Y.^2);                % Total Power
    SE = -sum((Y ./ sum(Y)) .* log(Y ./ sum(Y)));        % Spectral Entropy
    FR = sum(Y(f >= 50 & f <= 150)) / sum(Y(f >= 0 & f <= 500));  % Frequency Ratio
    BW = f(find(Y >= 0.5 * max(Y), 1, 'first')) - f(find(Y >= 0.5 * max(Y), 1, 'last'));
% Bandwidth
    P2P = max(Y) - min(Y);         % Peak-to-Peak
    PSD = sum(Y.^2);               % Power Spectral Density (potencia total)
    MOV = "palma";

    % Guardar caracterísitcas en el dataset
    dataset_palma = [dataset_palma; MAV, RMS, V, WL, SSC, IEMG, LOGVAR, PF, TP, SE, FR,
BW, PSD, P2P, MOV];
end

% Guardar el dataset en un archivo .mat
save('dataset_palma.mat', 'dataset_palma');
```

Cada ventana procesada genera un vector de características que se almacena en una matriz creciente, a la cual también se le asigna un etiquetado correspondiente a su clase (puño, palma, pinza o reposo). Al finalizar el procesamiento por clase, los datasets individuales se concatenan en un único dataset general que contiene todas las observaciones y sus respectivas etiquetas.

```
%% CONCATENAR TODAS LAS BASES
close all; clear; clc;

% Cargar los datasets
load('dataset_reposo.mat'); % Cargar dataset_reposo
load('dataset_palma.mat');  % Cargar dataset_palma
load('dataset_pinza.mat');  % Cargar dataset_pinza
load('dataset_puno.mat');   % Cargar dataset_puno

% Concatenar los datos en un solo dataset
dataset_completo = [dataset_reposo; dataset_palma; dataset_pinza; dataset_puno];

% Crear el encabezado
encabezado = {'MAV', 'RMS', 'V', 'WL', 'SSC', 'IEMG', 'LOGVAR', 'PF', 'TP', 'SE', 'FR',
'BW', 'PSD', 'P2P', 'MOV'};

% Escribir el dataset combinado en un archivo CSV
csvwrite_with_headers('dataset_completo.csv', dataset_completo, encabezado);

% Función para escribir CSV con encabezado
function csvwrite_with_headers(filename, data, headers)
    % Abrir el archivo para escribir
    fid = fopen(filename, 'w');

    % Escribir los encabezados
    fprintf(fid, '%s,', headers{1:end-1});
    fprintf(fid, '%s\n', headers{end});

    % Escribir los datos
    for i = 1:size(data, 1)
        fprintf(fid, '%f,', data(i, 1:end-1));
        fprintf(fid, '%s\n', data{i, end});
    end

    % Cerrar el archivo
    fclose(fid);
end
```

A continuación, se describen las **características extraídas**:

Características en el dominio del tiempo

1. **MAV (Mean Absolute Value):** Representa el valor promedio absoluto de la señal; útil para estimar la intensidad global del esfuerzo muscular en una ventana de tiempo.
2. **RMS (Root Mean Square):** Mide la energía o potencia media de la señal; es sensible a las variaciones de amplitud y útil para diferenciar contracciones fuertes y débiles.
3. **Varianza (V):** Evalúa la dispersión de la señal respecto a su media. Indica qué tan variable es la señal y ayuda a distinguir contracciones estables de otras más erráticas.
4. **WL (Waveform Length):** Suma total de las diferencias absolutas entre muestras sucesivas; refleja la complejidad y la frecuencia del cambio en la señal.

5. **SSC (Slope Sign Changes):** Número de veces que cambia la dirección de la pendiente entre muestras; asociado a la actividad motora fina y la complejidad del movimiento.
6. **IEMG (Integrada de EMG):** Suma de los valores absolutos de la señal; proporciona una medida acumulativa de la actividad muscular durante la ventana.
7. **LOGVAR (Log-Varianza):** Aplicación del logaritmo a la varianza; reduce el impacto de valores extremos, haciendo más estables las comparaciones entre ventanas.

Características en el dominio de la frecuencia (a partir de Fourier)

8. **Peak Frequency (PF):** Frecuencia dominante con mayor amplitud; permite distinguir patrones de contracción típicos de ciertos movimientos.
9. **Total Power (TP):** Suma de la energía contenida en todo el espectro de frecuencias; útil para estimar el esfuerzo total del músculo.
10. **Spectral Entropy (SE):** Mide el desorden o aleatoriedad del espectro; valores más altos indican mayor complejidad o ruido en la señal muscular.
11. **Frequency Ratio (FR):** Relación entre distintas bandas de frecuencia; puede discriminar movimientos según cómo se distribuye la energía en el espectro.
12. **Bandwidth (BW):** Ancho de banda efectivo del espectro; representa la dispersión de la energía en frecuencia, útil para analizar tipos de contracción.
13. **Peak-to-Peak :** Diferencia entre el valor máximo y mínimo de la señal (en tiempo o frecuencia); refleja la intensidad máxima de contracción.
14. **Power Spectral Density (PSD):** Distribución de la potencia por unidad de frecuencia; permite identificar rangos de frecuencia con mayor aporte energético.

Este conjunto inicial de características permite representar adecuadamente tanto la intensidad, complejidad y distribución espectral de la señal EMG, como los cambios sutiles entre diferentes tipos de movimiento. No obstante, posteriormente se aplicará un proceso de selección de características (feature selection) con el objetivo de reducir la dimensionalidad y mejorar la eficiencia y precisión del modelo de clasificación [8].

FEATURE SELECTION

Una técnica adecuada para la selección de características en este tipo de problema es Recursive Feature Elimination (RFE), ya que permite identificar cuáles características aportan mayor relevancia al desempeño del modelo. RFE funciona entrenando un modelo iterativamente, eliminando en cada ciclo las características menos importantes según el peso asignado por el modelo subyacente. Esta técnica es especialmente útil en

contextos donde se tienen múltiples descriptores en distintos dominios, como tiempo y frecuencia, en el caso de señales EMG, ya que ayuda a reducir la dimensionalidad sin perder información relevante para la clasificación.

Se implementa el algoritmo Recursive Feature Elimination (RFE) y el criterio de impureza Gini, similar al utilizado por los modelos de Random Forest. Para ello, en primer lugar, se definen las características y las clases presentes en el conjunto de datos. Luego, se utiliza una función para leer un archivo CSV que contiene el dataset completo, convirtiendo cada fila en un diccionario para facilitar su manipulación. Posteriormente, se divide el dataset en conjuntos de entrenamiento y prueba, asegurando reproducibilidad mediante el uso de una semilla aleatoria.

A partir del conjunto de entrenamiento, se extraen los vectores de características y etiquetas. Para calcular la importancia de cada característica, se evalúa cómo contribuye cada una a separar las clases utilizando distintos umbrales de decisión, y se mide la pureza de las divisiones resultantes mediante el índice de Gini: cuanto menor es la impureza obtenida con una característica, mayor es su importancia. Con base en estas puntuaciones, el algoritmo elimina iterativamente la característica menos importante, hasta conservar únicamente el número deseado de variables.

Para cada uno se evalúa qué tan bien separa las clases (reposo, palma, etc.) al dividir los datos por debajo y por encima del umbral. Luego se calcula la impureza Gini de esa división, y se selecciona el umbral que produce el menor Gini (mayor pureza). También, por ello se grafica el inverso como se muestra en la figura 10; y se seleccionan las mejores 10.

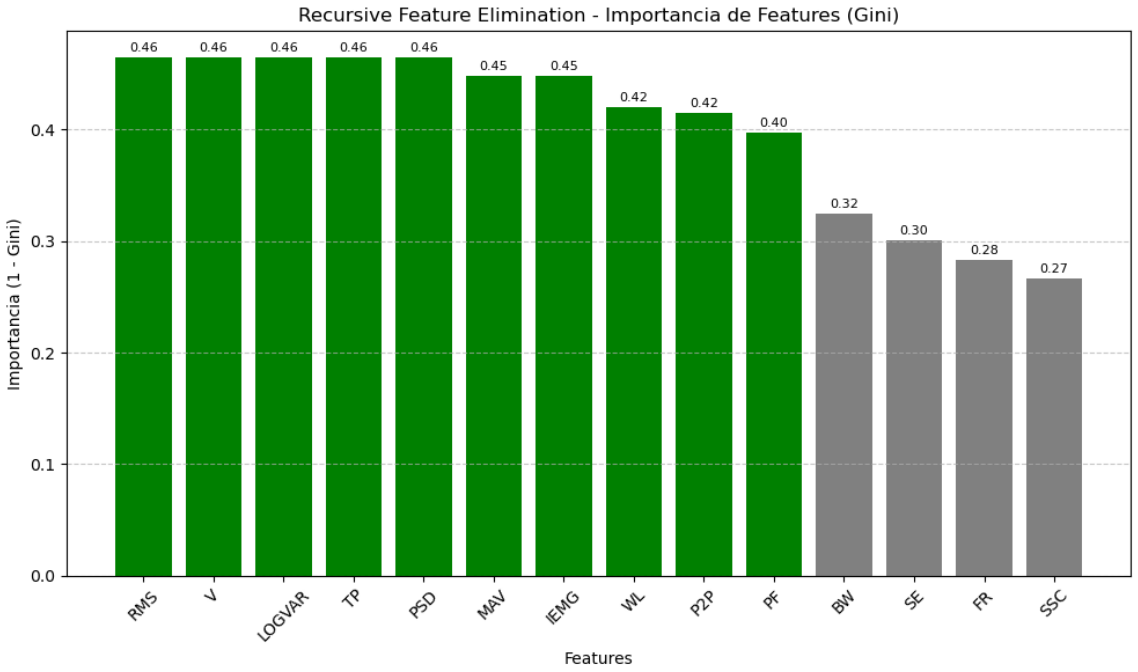

Figura 10. Recursive Feature Elimination – Importancia de Features (Gini)

SELECCIÓN DE MODELOS

Para la etapa de clasificación se seleccionaron cuatro modelos de machine learning supervisado que son ampliamente utilizados en tareas de reconocimiento de patrones y clasificación multiclase: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naive Bayes y Random Forest. Se seleccionan estos modelos por su desempeño comprobado en problemas con características tanto en el dominio del tiempo como de la frecuencia, como es el caso de las señales EMG [11].

KNN es un modelo no paramétrico sencillo pero eficaz, especialmente útil cuando las fronteras entre clases son complejas o no lineales, ya que clasifica con base en la proximidad en el espacio de características. **SVM**, por otro lado, es un modelo robusto que funciona bien en espacios de alta dimensión y es ideal cuando se requiere maximizar la separación entre clases, lo cual puede ser útil dada la variabilidad entre señales musculares [8].

Naive Bayes se considera por su simplicidad computacional y rapidez, y aunque asume independencia entre características, puede funcionar sorprendentemente bien en problemas biomédicos con grandes volúmenes de datos y ruido moderado. Finalmente, **Random Forest** se incluye por su capacidad de manejar datos no lineales y de seleccionar automáticamente características relevantes, lo cual es ventajoso al trabajar con múltiples descriptores extraídos de señales biológicas. El uso conjunto de estos modelos permite comparar distintos enfoques y seleccionar el que brinde el mejor desempeño en función de métricas como la precisión, sensibilidad y especificidad [5].

ENTRENAMIENTO DE MODELO

K-NEAREST NEIGHBORS (KNN)

Se implementa un clasificador **K-Nearest Neighbors (KNN)** desde cero y se evalúa sobre el set de datos con los features definidos previamente. Para calcular las distancias entre ejemplos, se utiliza la función `euclidean_distance`, que computa la distancia euclidiana entre dos instancias en el espacio definido por las características seleccionadas. El núcleo del clasificador está en la función knn, que calcula las distancias de una instancia a todos los ejemplos del conjunto de entrenamiento, ordena dichas distancias y toma los k vecinos más cercanos para determinar la clase más común entre ellos como predicción [6].

Posteriormente, el código divide el conjunto de datos en entrenamiento y prueba mediante una función, utilizando una proporción del 80% para entrenamiento y el resto para prueba, con una semilla fija para asegurar reproducibilidad. Luego, la función `ejecutar_knn` aplica el clasificador a cada elemento del conjunto de prueba y compara las etiquetas predichas con las reales, generando así una matriz de confusión que resume el desempeño del modelo [4].

SUPPORT VECTOR MACHINES (SVM)

Acá se usa un clasificador **SVM multiclase** usando el enfoque *One-vs-Rest*. Primero, se leen los datos, se mezclan aleatoriamente y se dividen en conjuntos de entrenamiento y prueba (80-20). Cada fila se transforma en un vector numérico con las características seleccionadas, mientras que la clase objetivo (reposo, palma, pinza, puño) se almacena por separado [4].

El entrenamiento del modelo se realiza en la función `entrenar_svm_binario`, que aplica una versión simplificada del algoritmo de SVM con descenso de gradiente para clasificar entre una clase positiva (+1) y el resto (-1), ajustando un vector de pesos y un sesgo (bias). Para manejar múltiples clases, la función `entrenar_svm_multiclase` entrena un modelo binario por cada clase (One-vs-Rest), generando así un conjunto de clasificadores independientes. La predicción se realiza evaluando cada vector de entrada contra todos los modelos y eligiendo la clase que produce el mayor puntaje (score) de decisión. Una vez obtenidas las predicciones para el conjunto de prueba, se genera una matriz de confusión que resume los aciertos y errores del modelo por clase.

NAIVE BAYES

Seguido, se usa un clasificador **Naive Bayes multiclase**, inicialmente, el archivo es leído y procesado para obtener las instancias de datos en forma de vectores numéricos, donde cada fila contiene características extraídas de señales EMG. Estas instancias se dividen en un conjunto de entrenamiento y otro de prueba de manera aleatoria (80-20) [4].

El algoritmo de Naive Bayes parte del supuesto de independencia entre características y modela la distribución de cada una como una distribución normal (gaussiana) por clase. En la función `calcular_parametros`, se calcula la media, varianza y probabilidad a priori de cada clase para cada característica, lo que constituye el modelo probabilístico del clasificador. Luego, en la función `predecir`, se calcula la probabilidad de que una instancia pertenezca a cada clase utilizando la función de densidad gaussiana. La predicción final para cada muestra es la clase con mayor probabilidad logarítmica conjunta. Finalmente, se genera una matriz de confusión comparando las etiquetas reales con las predichas, y se calcula el desempeño del clasificador a través de métricas de desempeño.

RANDOM FOREST

Por último, con un modelo de **Random Forest**, se usa una técnica de ensamble basada en la construcción de múltiples árboles de decisión para realizar una clasificación más robusta. El proceso comienza leyendo un archivo CSV con muestras caracterizadas por

un conjunto de features seleccionadas extraídas de señales EMG. El entrenamiento del modelo se realiza mediante la función `entrenar_random_forest`, la cual genera múltiples árboles sobre subconjuntos aleatorios del conjunto de entrenamiento y con una selección aleatoria de características en cada división interna de los árboles [4].

Cada árbol se construye con la función `construir_arbol`, utilizando el criterio de impureza de Gini para determinar la mejor división de los datos en cada nodo. El objetivo es maximizar la homogeneidad de las clases en cada rama del árbol. La predicción de una muestra nueva se realiza evaluando cada árbol del bosque y decidiendo por votación mayoritaria la clase final. Finalmente, el desempeño del modelo puede evaluarse usando métricas estándar de clasificación.

HIPERPARÁMETROS DE MODELOS

Modelo	Hiperparámetros	Descripción
KNN	k = 3	Número de vecinos más cercanos para votar la clase
SVM	epochs = 200	Iteraciones del entrenamiento
	lr = 0.005	Tasa de aprendizaje
	lam = 0.001	Regularización L2 (lambda)
Naive Bayes	Var mínima: $1x10^{-6}$	Evita división por cero en varianzas
	Probabilidad mínima: $1x10^{6}$	Evita log(0) en la probabilidad
Random Forest	n_arboles = 10	Número de árboles
	profundidad_max = 10	Profundidad por árbol
	min_muestras = 5	Mínimas muestras para dividir

PARÁMETROS COMUNES DE MODELOS

Modelo	Parámetros	Descripción
features	10	Usadas en todos los algoritmos, seleccionadas con RFE.
clases	reposo	Variable de salida en todos los casos.
	palma	
	pinza	
	puño	
semilla	42	Control de aleatoriedad para reproducibilidad
Proporción Training - Testing	Training: 80%	División porcentual de la base de datos para entrenamiento y testeo.
	Testing: 20%	

RESULTADOS

K-NEAREST NEIGHBORS (KNN)

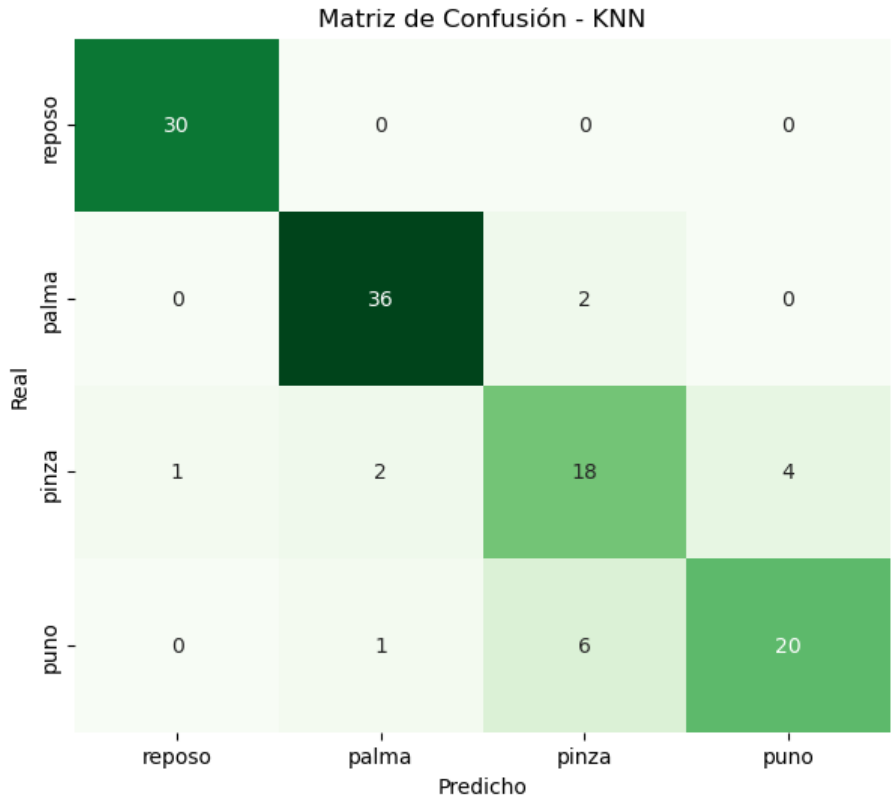

Figura 11. Matriz de Confusión | KNN

El modelo KNN obtuvo un rendimiento general adecuado, alcanzando una exactitud global de 0.87, lo que indica que el 87% de las predicciones fueron correctas en comparación con los valores reales. Al analizar el desempeño por clase, se observa que el modelo clasificó correctamente todos los casos de reposo, logrando una sensibilidad de 1.00 y una especificidad de 0.99, lo cual indica que no se presentaron falsos negativos y que los falsos positivos fueron mínimos.

Para la clase palma, la sensibilidad fue de 0.95, mostrando que el modelo fue muy eficaz en detectar correctamente esta clase. Su especificidad, de 0.96, sugiere que también fue capaz de distinguirla correctamente de las demás clases en la mayoría de los casos. Sin embargo, el desempeño decrece en las clases pinza y puño, que suelen presentar características más solapadas entre sí. En el caso de pinza, la sensibilidad fue de 0.72, indicando una mayor tasa de falsos negativos, mientras que su especificidad se mantuvo razonablemente alta en 0.92. Para puño, la sensibilidad fue de 0.74 y la especificidad de 0.96, lo que evidencia que el modelo fue mejor evitando falsos positivos que identificando correctamente todos los casos reales de esta clase.

Estos resultados sugieren que, aunque KNN funciona bien para movimientos claramente diferenciables como reposo y palma, su rendimiento disminuye ante clases más complejas o similares entre sí, como pinza y puño. Esta limitación puede deberse a su naturaleza basada en la vecindad, que lo hace vulnerable al solapamiento de patrones entre clases.

SUPPORT VECTOR MACHINES (SVM)

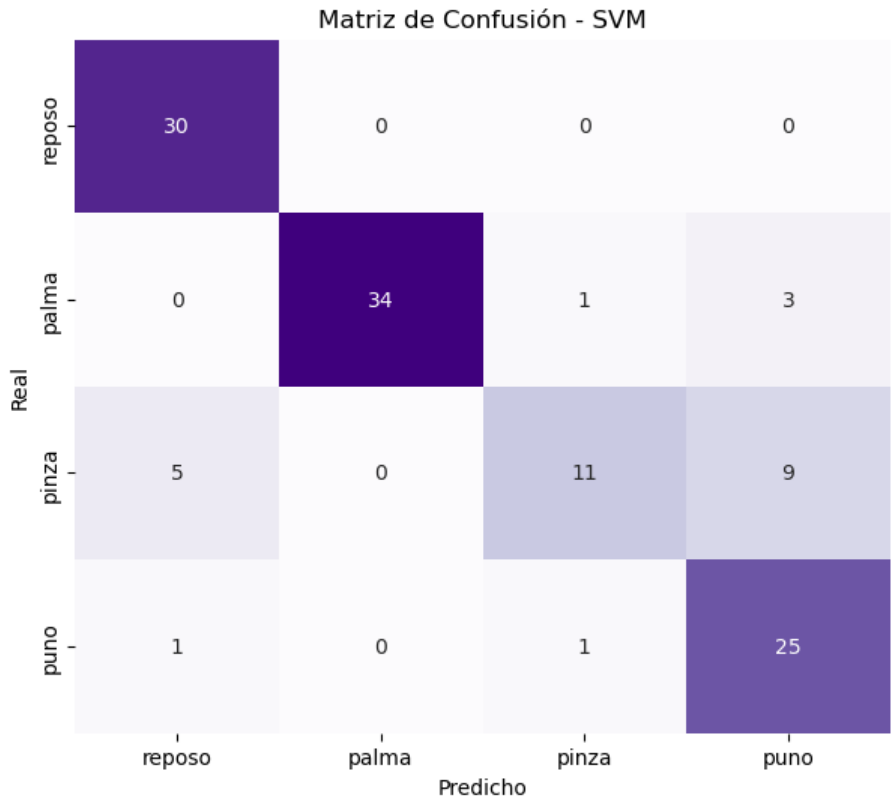

Figura 12. Matriz de Confusión | SVM

El modelo SVM mostró un comportamiento mixto en la clasificación. Comenzando con la clase reposo, el modelo logró una sensibilidad perfecta (1.00), lo que indica que todos los ejemplos reales de reposo fueron correctamente identificados. No obstante, la especificidad fue de 0.93, lo que sugiere que hubo algunos falsos positivos: es decir, otras clases fueron clasificadas erróneamente como reposo.

Para la clase palma, la sensibilidad fue de 0.89, un valor alto que indica un buen desempeño al identificar correctamente esta clase. Más destacable aún, su especificidad fue de 1.00, lo que significa que ningún caso de otra clase fue clasificado como palma, lo que refleja una excelente capacidad discriminativa respecto a las demás clases.

En contraste, la clase pinza presentó resultados significativamente más bajos. Su sensibilidad fue de apenas 0.44, lo que evidencia una alta tasa de falsos negativos: la mayoría de los casos reales de pinza fueron mal clasificados como puño. Sin embargo, su especificidad fue muy alta (0.98), lo cual indica que rara vez se clasificaron otras clases como pinza por error. Esta disparidad sugiere que el modelo tiene dificultad para reconocer adecuadamente los patrones que definen a esta clase, aunque es eficaz en evitar falsas asignaciones hacia ella. Por último, la clase puño obtuvo una sensibilidad de 0.93, evidenciando una muy buena detección de los casos reales. Sin embargo, su especificidad bajó a 0.87, lo que indica que otras clases fueron confundidas con puño en varias ocasiones, especialmente pinza, como se refleja en la matriz de confusión.

Se alcanzó una exactitud global del 0.83, este resultado refleja un desempeño general adecuado, aunque limitado por la baja sensibilidad obtenida en la clase pinza (0.44), que afectó de manera considerable la capacidad global del modelo para clasificar correctamente todos los tipos de movimientos. A pesar de este punto débil, el modelo mostró fortalezas claras en la clasificación de reposo, palma y puño, con sensibilidades cercanas o iguales a 0.9 y especificidades altas, destacando especialmente el comportamiento preciso al identificar la clase palma.

NAIVE BAYES

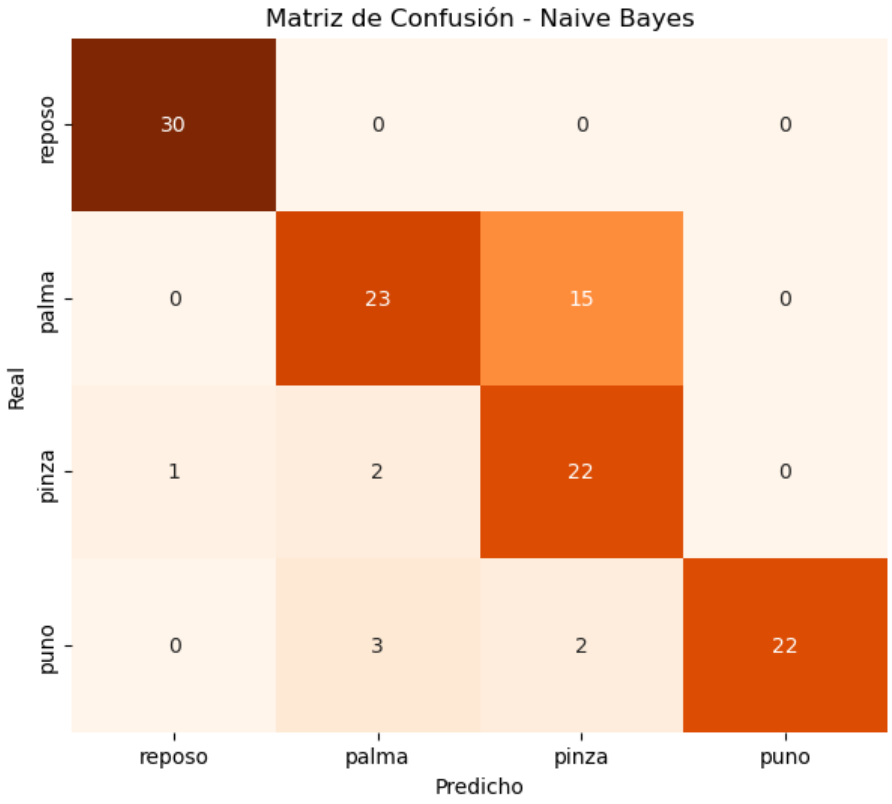

Figura 13. Matriz de Confusión | Naive Bayes

El modelo Naive Bayes obtuvo una exactitud global del 81 %, lo que indica un rendimiento general aceptable, aunque inferior en comparación con otros modelos evaluados. Al analizar la matriz de confusión, se observa que la clase reposo fue perfectamente clasificada, sin errores, lo cual se refleja en su sensibilidad de 1.00 y una especificidad de 0.99, indicando una alta capacidad del modelo para reconocer correctamente esta clase y diferenciarla de las demás. Sin embargo, el rendimiento en otras clases fue más desigual. La clase palma presentó una sensibilidad de 0.61, lo que revela una importante dificultad del modelo para identificar correctamente este movimiento, con 15 instancias clasificadas erróneamente como pinza. Esta confusión sugiere que, para Naive Bayes, las características extraídas de ambas clases tienen distribuciones similares, lo que genera ambigüedad en la predicción. A pesar de esto, la especificidad de palma fue alta (0.94), lo que indica que la mayoría de las muestras que no eran palma fueron correctamente identificadas como tal.

La clase pinza, por su parte, alcanzó una sensibilidad de 0.88, mostrando que el modelo identificó correctamente la mayoría de las instancias de esta clase. No obstante, su especificidad fue de 0.82, lo que indica cierta tendencia del modelo a clasificar erróneamente instancias de otras clases como pinza, en especial aquellas correspondientes a palma. En el caso de puño, la sensibilidad fue de 0.81, reflejando una capacidad adecuada del modelo para identificar correctamente este movimiento, aunque con algunos errores, como la clasificación incorrecta de instancias como palma o pinza. Sin embargo, su especificidad fue perfecta (1.00), lo que significa que el modelo no confundió instancias de otras clases como puño.

RANDOM FOREST

El modelo Random Forest obtuvo una exactitud global del 87 %, situándose entre los algoritmos con mejor desempeño general en esta evaluación. Su capacidad para combinar múltiples árboles de decisión permitió capturar patrones complejos en los datos y mejorar la robustez frente a errores de clasificación. La clase reposo fue clasificada con una sensibilidad perfecta de 1.00, lo que indica que todas las instancias reales de esta clase fueron correctamente identificadas. Además, su especificidad de 0.98 muestra que casi ninguna instancia de otra clase fue erróneamente clasificada como reposo, reflejando un rendimiento sobresaliente.

En el caso de la clase palma, la sensibilidad fue de 0.97, evidenciando que el modelo fue altamente eficaz en detectar este tipo de movimiento, con apenas una instancia mal clasificada como pinza. Su especificidad de 0.94 también fue elevada, lo que refuerza la solidez del modelo en esta categoría. La clase pinza presentó un desempeño más moderado, con una sensibilidad de 0.64, lo que indica que varias instancias fueron clasificadas incorrectamente como palma o puño. Este resultado sugiere cierta

dificultad del modelo para distinguir adecuadamente las características específicas de la pinza, a pesar de contar con una especificidad alta (0.95) que indica que el modelo rara vez clasifica incorrectamente otras clases como pinza.

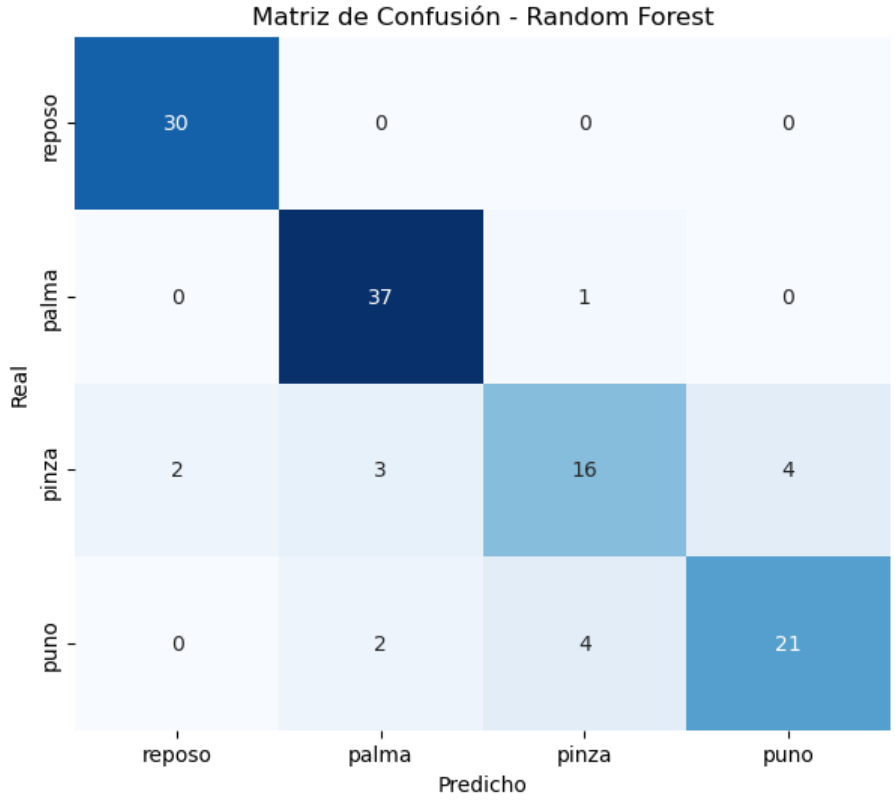

Figura 14. Matriz de Confusión | Random Forest

Por último, la clase puño logró una sensibilidad de 0.78, con algunas instancias mal clasificadas, principalmente como pinza y palma. Su especificidad de 0.96, sin embargo, indica que el modelo fue bastante confiable para identificar correctamente los movimientos que no eran puño. Random Forest mostró un rendimiento equilibrado y robusto, destacando especialmente en las clases reposo y palma, mientras que aún presentó desafíos en la discriminación de clases con patrones más similares como pinza y puño. Esto sugiere que, aunque el modelo es eficaz en general, podría beneficiarse de un ajuste fino en la selección de características o en la optimización de parámetros para mejorar su capacidad discriminativa en esas clases.

COMPARACIÓN DE MODELOS

		Sensibilidad	Especificidad	Exactitud
KNN	**Reposo**	100%	99%	87%
	Palma	95%	96%	
	Pinza	72%	92%	
	Puño	74%	96%	
SVM	**Reposo**	100%	93%	83%
	Palma	89%	100%	
	Pinza	44%	98%	
	Puño	93%	87%	
Naive Bayes	**Reposo**	100%	99%	81%
	Palma	61%	94%	
	Pinza	88%	82%	
	Puño	81%	100%	
Random Forest	**Reposo**	100%	98%	87%
	Palma	97%	94%	
	Pinza	64%	96%	
	Puño	78%	96%	

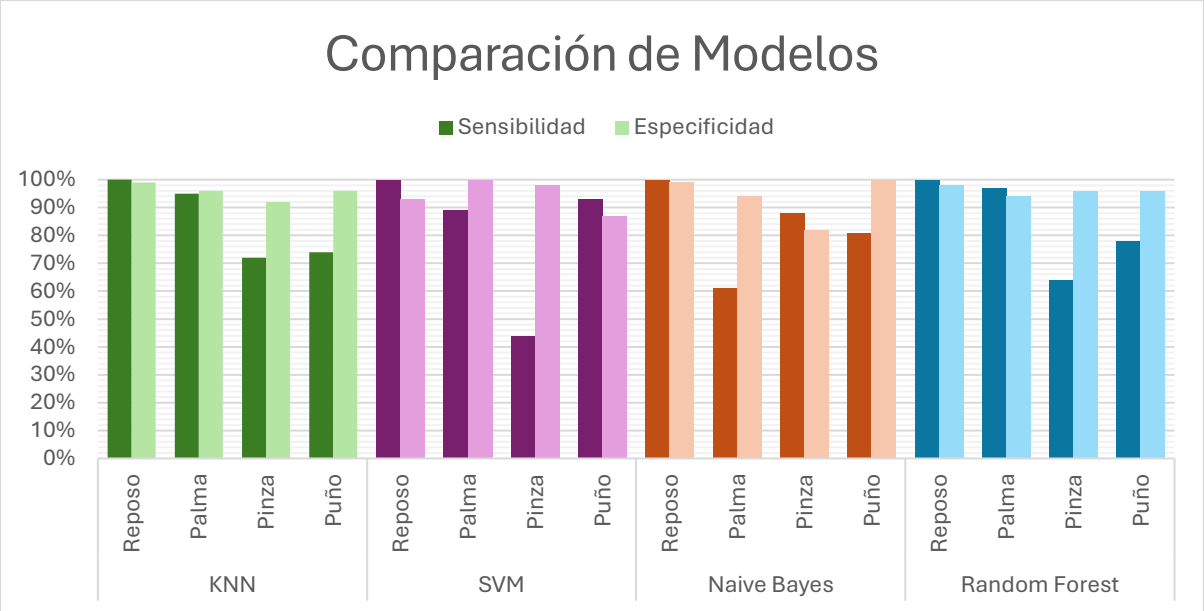

Figura 15. Comparación de Métricas de Desempeño entre Modelos

Entre los cuatro modelos evaluados, KNN y Random Forest son los que lograron la mayor exactitud global, con un 87%. Sin embargo, KNN destaca por ofrecer un balance más homogéneo entre sensibilidad y especificidad a lo largo de todas las clases. Esto indica

que KNN no solo clasifica correctamente una gran proporción de instancias, sino que también es consistente al detectar tanto las verdaderas positivas como evitar falsos positivos en cada categoría de movimiento. Por otro lado, aunque Random Forest iguala a KNN en exactitud global, presenta una mayor variabilidad en la sensibilidad, especialmente en la clase “pinza”, donde su rendimiento es menor. Esto sugiere que el modelo puede ser menos robusto para distinguir ciertos movimientos, lo cual es crucial en aplicaciones biomédicas donde la detección precisa es necesaria.

KNN probablemente sea el mejor modelo para esta tarea debido a su simplicidad y su capacidad para adaptarse a la estructura local de los datos. Al basar sus predicciones en la proximidad de los ejemplos de entrenamiento más cercanos, KNN puede capturar patrones específicos y variaciones sutiles en las señales asociadas a cada movimiento, lo que resulta en un mejor equilibrio entre sensibilidad y especificidad. Además, KNN no asume una forma paramétrica del modelo, lo que le permite manejar datos con distribuciones complejas o no lineales, característica común en señales biomédicas. Esta flexibilidad hace que KNN sea particularmente adecuado para clasificar correctamente diferentes tipos de movimientos, minimizando tanto falsos positivos como falsos negativos.

El modelo SVM obtuvo una exactitud global del 83%, mostrando buen desempeño en las clases reposo y puño, con sensibilidades de 1.00 y 0.93 respectivamente. Sin embargo, presentó una baja sensibilidad en la clase pinza, apenas del 0.44, lo que limitó su eficacia general. Aunque su especificidad fue alta en la mayoría de las clases, la falta de balance en sus métricas hace que no supere a Random Forest ni a KNN en rendimiento global. El desempeño más bajo de SVM en este contexto puede deberse a que el modelo lineal utilizado no logró capturar la complejidad y no linealidad inherente a los datos de señales biomédicas. Las características extraídas pueden no estar separables mediante un hiperplano lineal simple, lo que limita la capacidad de SVM para discriminar correctamente entre clases, especialmente en categorías con mayor superposición o ruido.

Finalmente, Naive Bayes fue el modelo con menor exactitud global, con un 81%. Mostró sensibilidad variable, con buenos resultados en reposo y pinza, pero un desempeño notablemente bajo en palma, con solo un 0.61 de sensibilidad. Esto indica que Naive Bayes tiene dificultades para distinguir ciertas clases, afectando su capacidad general de clasificación en este contexto.

CONCLUSIONES

- KNN y Random Forest lograron la mayor exactitud global (87%), siendo ambos modelos adecuados para la clasificación de movimientos.
- KNN destaca por su mejor balance entre sensibilidad y especificidad en todas las clases, lo que garantiza una clasificación más consistente y confiable.
- Random Forest, aunque también preciso, mostró mayor variabilidad en la sensibilidad de ciertas clases, lo que podría afectar la robustez en la detección de movimientos específicos.
- SVM presentó una exactitud ligeramente menor (83%) y mostró dificultades especialmente en la clase “pinza”, con baja sensibilidad, indicando que le costó detectar correctamente ese movimiento.
- Naive Bayes obtuvo la menor exactitud global (81%) y aunque tuvo buena sensibilidad en algunas clases, mostró problemas para diferenciar la clase “palma”, reflejado en una baja sensibilidad.
- Los modelos basados en árboles y neighbors (Random Forest y KNN) demostraron ser más adecuados para este problema, probablemente por su capacidad para manejar relaciones no lineales y variabilidad en los datos.
- El equilibrio entre sensibilidad y especificidad es fundamental en problemas biomédicos para minimizar falsos negativos y falsos positivos, razón por la cual KNN es la mejor opción en este estudio.
- La elección del modelo debe basarse no solo en la exactitud global, sino en la consistencia del rendimiento para cada clase específica, como evidencia KNN en este caso.

REFERENCIAS

[1] D. Segura, E. Romero, V. E. Abarca, y D. A. Elias, “Upper Limb Prostheses by the Level of Amputation: A Systematic Review”, *Prosthesis*, vol. 6, núm. 2, pp. 277–300, mar. 2024, doi: 10.3390/prosthesis6020022.

[2] T. R. Makin, N. Filippini, E. P. Duff, D. Henderson Slater, I. Tracey, y H. Johansen-Berg, “Network-level reorganisation of functional connectivity following arm amputation”, *Neuroimage*, vol. 114, pp. 217–225, jul. 2015, doi: 10.1016/j.neuroimage.2015.02.067.

[3] J. Duchene y J.-Y. Hogrel, “A model of EMG generation”, *IEEE Trans Biomed Eng*, vol. 47, núm. 2, pp. 192–201, 2000, doi: 10.1109/10.821754.

[4] H. A. Romo Esp., J. C. Realpe, y P. E. Jojoa, “Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano”, *Revista Avances en Sistemas e Informática*, vol. 4, jun. 2007.

[5] P. K. Artemiadis y K. J. Kyriakopoulos, “EMG-Based Control of a Robot Arm Using Low-Dimensional Embeddings”, *IEEE Transactions on Robotics*, vol. 26, núm. 2, pp. 393–398, abr. 2010, doi: 10.1109/TRO.2009.2039378.

[6] A. L. Hof, “EMG and muscle force: An introduction”, *Hum Mov Sci*, vol. 3, núm. 1–2, pp. 119–153, mar. 1984, doi: 10.1016/0167-9457(84)90008-3.

[7] P. Grosse, M. J. Cassidy, y P. Brown, “EEG–EMG, MEG–EMG and EMG–EMG frequency analysis: physiological principles and clinical applications”, *Clinical Neurophysiology*, vol. 113, núm. 10, pp. 1523–1531, oct. 2002, doi: 10.1016/S1388-2457(02)00223-7.

[8] Sukhan Lee y G. Saridis, “The control of a prosthetic arm by EMG pattern recognition”, *IEEE Trans Automat Contr*, vol. 29, núm. 4, pp. 290–302, abr. 1984, doi: 10.1109/TAC.1984.1103521.

[9] C. L. Kok, C. K. Ho, F. K. Tan, y Y. Y. Koh, “Machine Learning-Based Feature Extraction and Classification of EMG Signals for Intuitive Prosthetic Control”, *Applied Sciences*, vol. 14, núm. 13, p. 5784, jul. 2024, doi: 10.3390/app14135784.

[10] J. Yousefi y A. Hamilton-Wright, “Characterizing EMG data using machine-learning tools”, *Comput Biol Med*, vol. 51, pp. 1–13, ago. 2014, doi: 10.1016/j.compbiomed.2014.04.018.

[11] Z. Karapinar Senturk y M. Sevgul Bakay, “ Machine Learning-Based Hand Gesture Recognition via EMG Data”, *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 10, núm. 2, abr. 2021.

ANEXOS

Link Github:
https://github.com/danielanavas2002/InteligenciaArtificial/tree/main/Proyecto%20Final

Link Video en Youtube:
https://youtu.be/kzs8qsl-iXQ