

MAURICIO GUERRERO DORIA  
JOSE LUIS CORTES CARVAJAL

SISTEMA DETECTOR DE INTENCIÓN DE MOVIMIENTO A PARTIR DE  
SEÑALES ELECTROMIOGRÁFICAS SUPERFICIALES

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Trabajo de grado  
Área: Electrónica y Telecomunicaciones  
Popayán  
2012

**SISTEMA DETECTOR DE INTENCIÓN DE MOVIMIENTO A PARTIR DE  
SEÑALES ELECTROMIOGRÁFICAS SUPERFICIALES**



**Trabajo de Grado**

**Mauricio Guerrero Doria**

**Jose Luis Cortes Carvajal**

Director: Mg. Judy Cristina Realpe Chamorro

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Electrónica, Instrumentación y Control**

**Línea de I+D en Robótica y Control**

**Ingeniería en Automática Industrial**

Popayán, Octubre de 2012

## AGRADECIMIENTOS

*Al término de esta etapa de mi vida, quiero expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión me alentaron a lograr esta hermosa realidad.*

*Atte. Mauricio Guerrero Doria.*

*Agradezco de todo corazón a Dios y a mis padres porque a través de ellos me concedió la vida en este mundo, así como a mis abuelos, tíos, hermanos y a todas las personas que directa o indirectamente han tenido a bien ayudarme en forma moral y académica para mi formación como ser humano y profesional, en respuesta a esto, cuentan con mi mayor gratitud.*

*Atte. Jose Luis Cortes.*

## RESUMEN

Este documento presenta la construcción de un sistema detector de intención de movimiento a partir de señales electromiográficas superficiales, diseñado en la Universidad del Cauca, Colombia. El sistema consta de un software que fue adaptado a código C para la clasificación eficientemente de diferentes tipos de intención de movimiento en mano. El sistema trabaja con señales electromiográficas captadas del antebrazo de un paciente sano, que se encuentran representadas en la base de datos suministrada por el Ph.D. Kevin Englehart del Instituto de Ingeniería Biomédica de la Universidad de New Brunswick. El hardware del proyecto consta de un procesador digital de señal (*DSPIC33fj256gp710a*) de Microchip, programado para llevar a cabo la clasificación de señales electromiográficas en un sistema externo al computador.

## ABSTRACT

This document presents the construction of a intention motion detector system from surface electromyographic signals, designed at the University of Cauca, Colombia. The system consists of software that was adapted to C code for the classification and sufficiently intention of different types of hand movement. The system works with electromyographic signals captured from the forearm of a healthy patient, which are represented in the database provided by the Ph.D. Kevin Englehart Institute of Biomedical Engineering, University of New Brunswick. The hardware of the project consists of a digital signal processor (*DSPIC33fj256gp710a*) from Microchip, programmed to carry out the classification of electromyographic signals in an external system to the computer.

## TABLA DE CONTENIDO

<b>RESUMEN .....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>LISTA DE TABLAS .....</b>	<b>vi</b>
<b>LISTA DE FIGURAS.....</b>	<b>vii</b>
<b>LISTA DE ABREVIATURAS .....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>1. MARCO CONCEPTUAL.....</b>	<b>5</b>
1.1. ESTUDIO DESCRIPTIVO DE LAS SEÑALES EMG.....	5
1.2. RECONOCIMIENTO DE PATRONES.....	8
1.2.1. Extracción de Características de Señales EMGS.....	9
1.2.1.1. <i>Análisis Temporal</i> .....	10
1.2.1.2. <i>Análisis Tiempo – Frecuencia</i> .....	13
1.2.2. Reducción de Dimensionalidad .....	16
1.2.2.1. <i>Reducción por Selección de Características</i> .....	16
1.2.2.2. <i>Reducción por Proyección de Características</i> .....	17
1.2.3. Clasificación de Patrones.....	17
1.2.3.1. <i>Fundamentos Teóricos</i> .....	19
1.2.3.2. <i>Consideraciones para evaluar un clasificador</i> .....	26
1.3. DESCRIPCIÓN BASE DE DATOS.....	27
1.4. DESCRIPCIÓN DE UN SISTEMA EMBEBIDO .....	29
1.4.1. Circuitos Integrados.....	30
1.4.1.1. <i>Principales Características de los CI</i> .....	31
1.5. PROTOCOLO DE COMUNICACIÓN RS-232 .....	38

1.5.1.	Características Eléctricas .....	38
1.5.2.	Características de conexión.....	39
1.5.3.	Métodos de Transmisión en Serie .....	40
1.6.	TRABAJOS PREVIOS EN LA CLASIFICACIÓN DE SEÑALES EMGS .....	41
<b>2.</b>	<b>SELECCIÓN E IMPLEMENTACIÓN DEL ALGORITMO DE CLASIFICACIÓN .....</b>	<b>45</b>
2.1.	ELECCIÓN DEL ALGORITMO DE CLASIFICACIÓN .....	45
2.1.1.	Algoritmos de Clasificación Disponibles.....	46
2.1.2.	Criterios de Selección .....	47
2.1.3.	Evaluación de los Criterios de Selección .....	51
2.1.4.	Selección del Algoritmo de Clasificación .....	54
2.1.5.	Descripción del Algoritmo Seleccionado .....	55
2.2.	ADECUACIÓN DEL ALGORITMO DE CLASIFICACIÓN .....	58
2.2.1.	Preparación del Algoritmo Seleccionado .....	59
2.2.2.	Creación de la Nueva Base de Datos de Vectores de Características .....	59
2.2.3.	Creación y Entreno de las Redes Neuronales (MLP) .....	60
2.2.4.	Traducción del Algoritmo de Clasificación al Lenguaje C .....	63
2.2.5.	Algoritmo de Clasificación en Lenguaje C .....	63
2.2.6.	Desempeño del Algoritmo de Clasificación en Lenguaje C .....	65
<b>3.</b>	<b>SELECCIÓN DEL CIRCUITO INTEGRADO.....</b>	<b>69</b>
3.1.	TOMA DE DECISIONES .....	69
3.1.1.	Problema .....	70
3.1.2.	Búsqueda de Alternativas.....	70
3.1.2.1.	<i>Microchip</i> .....	70
3.1.2.2.	<i>Texas Instruments</i> .....	72
3.1.2.3.	<i>Freescale</i> .....	73
3.1.3.	Selección de Alternativas .....	73
3.1.4.	Criterios de Evaluación .....	77
3.1.5.	Asignación de Pesos .....	79

3.1.6.	Análisis de la Matriz de Decisión .....	80
3.2.	Análisis del Dispositivo Seleccionado .....	81
<b>4.</b>	<b>DISEÑO Y CONSTRUCCIÓN DEL SISTEMA .....</b>	<b>83</b>
4.1.	DOCUMENTACIÓN .....	83
4.1.1.	Circuito integrado.....	83
4.1.2.	Regulador de Voltaje .....	84
4.1.3.	Adaptador de Nivel TTL a RS-232 .....	85
4.1.4.	Componentes adicionales .....	86
4.2.	DISEÑO .....	87
4.2.1.	Herramientas de Diseño.....	87
4.2.2.	Procedimientos de Diseño .....	88
4.2.3.	Diseño Interfaz de Usuario .....	92
4.3.	CONSTRUCCIÓN .....	94
<b>5.</b>	<b>PRUEBAS Y VALIDACIÓN .....</b>	<b>97</b>
5.1.	PRUEBAS DE INTEGRACIÓN .....	97
5.2.	PRUEBA DE FUNCIONALIDAD .....	99
5.3.	ANÁLISIS DE RESULTADOS .....	103
<b>6.</b>	<b>CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>107</b>
6.1.	CONCLUSIONES.....	107
6.2.	TRABAJOS FUTUROS .....	108
<b>7.</b>	<b>BIBLIOGRAFÍA.....</b>	<b>110</b>
<b>8.</b>	<b>ANEXOS.....</b>	<b>114</b>



## LISTA DE TABLAS

TABLA 1. ESPECIFICACIONES DE CONEXIÓN PARA UN CONECTOR DB-9.....	39
TABLA 2. USO DE LA FUNCIÓN CPUTIME.....	48
TABLA 3. USO DE LA FUNCIÓN TIC-TOC.....	48
TABLA 4. ERROR DE CLASIFICACIÓN PARA LAS TÉCNICAS DT-DWT Y DT-WPT .....	49
TABLA 5. USO DE LOS COMANDOS WHO Y WHOS DE MATLAB.....	50
TABLA 6. EVALUACIÓN DE LOS CRITERIOS DE SELECCIÓN .....	52
TABLA 7. PUNTUACIÓN DE LOS CRITERIOS DE SELECCIÓN .....	54
TABLA 8. ERROR DE CLASIFICACIÓN DEL ALGORITMO DE CLASIFICACIÓN EN C .....	67
TABLA 9. PRINCIPALES CARACTERÍSTICAS DEL ALGORITMO DE CLASIFICACIÓN EN C.....	68
TABLA 10. CARACTERÍSTICAS PRINCIPALES DE LOS CIRCUITOS INTEGRADOS DE MICROCHIP. ....	74
TABLA 11. CARACTERÍSTICAS PRINCIPALES DE LOS CIRCUITOS INTEGRADOS DE TEXAS INSTRUMENTS.....	75
TABLA 12. CARACTERÍSTICAS PRINCIPALES DE LOS CIRCUITOS INTEGRADOS DE FREESCALE. ....	75
TABLA 13. RESUMEN DE LAS ALTERNATIVAS SELECCIONADAS.....	77
TABLA 14. PONDERACIÓN DE CRITERIOS. ....	79
TABLA 15. MATRIZ DE DECISIÓN .....	81
TABLA 16. ESPECIFICACIONES TÉCNICAS DEL DISPOSITIVO MAX-232 DE MAXIM ELECTRONICS.....	86
TABLA 17. MEJORES CANALES POR PACIENTE.....	100
TABLA 18. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 1. ....	101
TABLA 19. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 2. ....	101
TABLA 20. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 3. ....	102
TABLA 21. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 4. ....	102
TABLA 22. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 5. ....	102
TABLA 23. REGISTROS DE LAS PRUEBAS DE CLASIFICACIÓN EN EL PACIENTE 6. ....	103
TABLA 24. RESULTADOS DE CLASIFICACIÓN, ACIERTOS Y DESACIERTOS.....	104
TABLA 25. PORCENTAJES DE ACIERTO Y DESACIERTO PARA CADA MOVIMIENTO. ....	105

## LISTA DE FIGURAS

FIGURA 1. DETECCIÓN DE SEÑALES EMG .....	8
FIGURA 2. FASES EN EL RECONOCIMIENTO DE PATRONES .....	9
FIGURA 3. ESTRUCTURA PERCEPTRÓN MULTICAPA .....	20
FIGURA 4. ARQUITECTURA GENERAL DE LA RED PNN .....	23
FIGURA 5A. DIVISIÓN DE LOS DATOS EN DOS CLASES.....	24
FIGURA 6. TIPOS DE MOVIMIENTO .....	27
FIGURA 7. UBICACIÓN DE LOS ELECTRODOS EN VISTA TRANSVERSAL DEL ANTEBRAZO [3].....	29
FIGURA 8. ETAPAS PARA EL PROCESAMIENTO DE UNA SEÑAL [17].....	31
FIGURA 9. REPRESENTACIONES NUMÉRICAS COMUNES EN LOS CI .....	33
FIGURA 10. FORMATO NUMÉRICO CON COMA FIJA Y COMA FLOTANTE.....	33
FIGURA 11. ARQUITECTURA VON NEUMANN .....	36
FIGURA 12. ARQUITECTURA HARVARD .....	37
FIGURA 13. CARACTERÍSTICAS ELÉCTRICAS DEL PROTOCOLO DE COMUNICACIÓN RS-232 .....	38
FIGURA 14. FUNCIÓN DE ACTIVACIÓN TANGENTE HIPERBÓLICA SIGMOIDAL .....	57
FIGURA 15. FUNCIÓN DE ACTIVACIÓN LINEAL.....	57
FIGURA 16. ARQUITECTURA DEL PERCEPTRÓN MULTICAPAS DEL ALGORITMO SELECCIONADO.....	58
FIGURA 17. BASE DE DATOS DE LOS VECTORES DE CARACTERÍSTICAS PARA UN PACIENTE. ....	60
FIGURA 18. ARQUITECTURA DEL CLASIFICADOR A IMPLEMENTAR (PERCEPTRÓN MULTICAPAS).....	61
FIGURA 19. MODULO MA330012 .....	84
FIGURA 20. DISPOSITIVO LM1117T DE NATIONAL SEMICONDUCTOR .....	84
FIGURA 21. DISPOSITIVO MAX-232 DE MAXIM ELECTRONICS.....	85
FIGURA 22. DISEÑO PCB EAGLE 6.1 .....	89
FIGURA 23. DIAGRAMA DE CONEXIONES EAGLE 6.1.....	90
FIGURA 24. DIAGRAMA DE FLUJO DE LA INTERFAZ DE USUARIO.....	93
FIGURA 25. INTERFAZ DE USUARIO .....	94
FIGURA 26. DISPOSITIVO CLASIFICADOR DE INTENCIÓN DE MOVIMIENTOS.....	96

## LISTA DE ABREVIATURAS

(ADC): (*Analog to Ddigital Converter*), Conversor análogo - Digital

(ASIC): (*Application Specific Integrated Circuit*), Circuito Integrado Para Aplicación Especifica

(CSP): Cambio de Signo de Pendiente

(CC): Cruce por Cero

(CWT ): Transformada Wavelet Continúa

(CSI ): (*Class Separability Index*), Indice de Separabilidad de Clases

(DWT ): (*Discrete wavelet Transform*), Transformadas Wavelets Discreta

(FT): (*Fourier Transform*), Transformada de Fourier

(FPGA): (*Field Programmable Gate Array*), Dispositivo semiconductor programable

(ISIS): (*Intelligent Schematic Input System*)

(LO): Longitud de la Forma de Onda

(LDA): Analizador Discriminante Lineal

(LVQ): Aprendizaje de Cuantización Vectorial

(MAC): (*Multiply And Acumulate*), Multiplicador Acumulador

(MIPS): Millones de Instrucciones por Segundo

(MLP): (*Multi Layer Perceptron*) Perceptrón Multicapas

(PCB): (*Printed Circuit Board*)

(PCA): (*Principal Component Analysis*),

(PNN): (*Probabilistic Neuronal Network*) una red neuronal probabilística

(RBFN): (*Radial Basis Function Network*),

(RAM): (*Random Access Memory*), Memoria de Acceso Aleatorio

(RISC): (*Reduced Instruction Set Computer*)

(ROM): (*Read Only Memory*), Memoria de Solo Lectura

(STFT): (*Short Time Fourier Transform*), Transformada de Fourier en Tiempo Corto

(SVM): (*Support Vector Machine*), Maquina Soporte Vectorial

(SE): Sistema Embebido

(VMA): Valor medio Absoluto

(VMAP): Valor Medio Absoluto de Pendiente

WPT (*Wavelet Package transform*), Transformada de Paquetes Wavelet

## INTRODUCCIÓN

Las señales electromiográficas de superficie (EMGS) se han convertido en un material de investigación muy importante en el reconocimiento de patrones, control de dispositivos, rehabilitación, detección de patologías, entre otros campos debido a la facilidad con la que se pueden adquirir de forma no invasiva y sin peligro para el individuo [1]. Actualmente, una de las aplicaciones más importantes para este tipo de señales es el control de dispositivos activos como las prótesis mioeléctricas, las cuales consisten en un sistema accionado por servomotores que se gobierna a partir de señales electromiográficas, bien sean intramusculares, capturadas mediante agujas o superficiales, recogidas directamente en el muñón del paciente mediante electrodos [2].

Uno de los principales componentes en el desarrollo de una prótesis mioeléctrica, es el sistema clasificador de las señales electromiográficas, el cual interpreta la intención de movimiento del paciente a partir de dicha señal. No obstante, existen distintos tipos de sistemas de clasificación de señales EMGS, los cuales difieren en algunas de sus especificaciones tales como: la técnica de extracción de características, el tipo de características que se tienen en cuenta para la clasificación o el tipo de clasificador. La Universidad del Cauca ha generado un gran aporte con el trabajo realizado por el Mg H. Romo en [3], el cual mediante el análisis de señales EMG con técnicas basadas en *wavelets*, extrae características que ayudan a mejorar el desempeño en fase de clasificación. Este tipo de trabajos han creado nuevos enfoques en la instauración de proyectos macro para el desarrollo de prótesis de mano.

Uno de los proyectos en el que la Universidad del Cauca a través del Departamento de Electrónica, Instrumentación y Control ha enfocado su trabajo, es el *Prototipo Funcional de Prótesis de Mano Robótica* en donde el objetivo es desarrollar una prótesis de mano que integre las fases de extracción de

características, clasificación de las señales y sistema actuador. Una de las fases fundamentales y sobre la cual se desarrolla el proyecto *Sistema Detector de Intención de Movimiento a Partir de Señales Electromiográficas Superficiales*, es la clasificación de señales EMG realizada mediante un dispositivo externo al computador.

La monografía del proyecto en mención describe los procedimientos que se llevaron a cabo para obtener los objetivos propuestos; en el capítulo 1 se realiza una descripción de las tres fases que componen el proceso de reconocimiento de patrones; en el capítulo 2, se presenta el proceso de selección del algoritmo de identificación de movimiento y su respectiva implementación; en el capítulo 3, se evidencia el proceso de selección del componente hardware que cumple con los requerimientos para la implementación del algoritmo de identificación de movimiento; en el capítulo 4, se presenta el diseño y la construcción de una arquitectura hardware capaz de identificar las señales electromiográficas almacenadas en la base de datos especificada; en el capítulo 5 se exponen las pruebas efectuadas, los resultados obtenidos y se hace la evaluación de la implementación de los métodos y finalmente, en el capítulo 6 se exponen las conclusiones y recomendaciones obtenidas en el proyecto, que puedan ayudar al desarrollo de futuros trabajos e investigaciones en esta área.

## 1. MARCO CONCEPTUAL.

### 1.1. ESTUDIO DESCRIPTIVO DE LAS SEÑALES EMG

Las señales bioeléctricas constituyen una parte fundamental de las diferentes ciencias encargadas del estudio y extracción de información de las señales fisiológicas presentes en algunos organismo biológicos. Una de las técnicas bioeléctricas más utilizadas en la actualidad es la electromiografía, la cual es la encargada del registro de la actividad eléctrica muscular que se genera en respuesta a la estimulación nerviosa. Para comprender más a fondo las señales electromiográficas se darán a conocer algunos términos que serán útiles para la comprensión de proceso.

- **Neurona**

La neurona es la célula principal del sistema nervioso. Tiene la capacidad de responder a los estímulos generando un impulso nervioso que se transmite a otra neurona, a un músculo o a una glándula. Existen diferentes tipos de neuronas, que se diferencian por su forma o tamaño [4]. Funcionalmente las neuronas se pueden clasificar en tres tipos:

- **Neuronas sensitivas:** Aisladas o localizadas en órganos sensoriales o en zonas del sistema nervioso relacionadas con la integración de las sensaciones.
- **Neuronas motoras:** Localizadas en áreas del sistema nervioso responsables de la respuesta motora.
- **Interneuronas o neuronas de asociación:** Relacionan distintos tipos de neuronas entre sí.

- **Unidad Motora (UM)**

Una neurona motora y el conjunto de todas las fibras musculares a las que inerva se constituyen como una unidad motora. Las unidades motoras pueden variar de tamaño, en promedio una neurona motora tiene relación con 150 fibras musculares, pero existen neuronas que inerva 10 fibras musculares y hasta neuronas que inervan 200 o más fibras musculares. En el primer caso los movimientos son finos y de poca potencia, en el segundo caso son burdos y de gran potencia [5].

- **Electromiografía**

La electromiografía básicamente consiste en el registro y estudio de las señales eléctricas involucradas en la activación y contracción muscular. Estas señales son producidas por el intercambio de partículas cargadas eléctricamente (iones) a través de las membranas de las fibras musculares. La estimulación de una neurona motora produce la contracción de todas las fibras musculares de una determinada unidad motora. Así cuando todas las fibras musculares asociadas a una unidad motora UM son excitadas repetidamente se crea un tren de impulsos conocidos como Tren de MUAP (Motor Unit Action Potencial). Y a la combinación aditiva de la actividad eléctrica creada por cada unidad motora excitada al momento de realizar un movimiento o contracción muscular se conoce como señal electromiográfica EMG [3]. La amplitud de la señal EMG depende del número de UM reclutadas, su profundidad y el número de fibras que componen cada una de ellas. Adicionalmente, también depende de la frecuencia con la que son activadas.

La amplitud de la señal es un parámetro que se puede analizar como indicador de actividad muscular utilizando diversos parámetros, entre ellos el valor cuadrático medio (RMS). Adicionalmente, dentro del sistema



representado por el modelo de generación de impulsos, el principio de Henneman (1957) [3] propone que a bajos niveles de esfuerzo, sólo se activan unidades motoras de bajo umbral que permiten obtener mayor precisión en los movimientos y bajas tasas de fatiga, permitiendo realizar tareas de habilidad. Dicha amplitud se mide entre picos de la señal y refleja la actividad de un número relativamente pequeño de fibras musculares, normalmente mide entre 200  $\mu\text{V}$  y 3 mV.

Las señales electromiográficas pueden ser registradas mediante diferentes tipos de mecanismos, siendo uno de los más utilizados el registro de señales electromiográficas superficiales (EMGS). Este mecanismo de registro superficial se realiza mediante elementos conductivos (electrodos) de plata o cloruro de plata (Ag - AgCl) que son situados en posiciones correctamente identificadas en donde se localizan las fibras musculares comprometidas en la ejecución de los movimientos, como se observa en la figura 1. El espectro de frecuencias esperado de la señal EMG superficial, comprende la banda entre 10 y 400Hz, dicho espectro se ve afectado principalmente por la forma de onda de los MUAP de las UM activas, aunque también depende de la frecuencia de generación de los MUAP [5].

Esencialmente las señales EMGS se consideran como un patrón unidimensional, por lo que cualquier técnica de procesamiento de señales para extracción de características y reconocimiento de patrones se puede aplicar a este tipo de señales. La información extraída de las señales EMGS, se considera de buena calidad cuando se maximiza la amplitud y se minimiza el ruido, aportando una señal óptima que reduzca el error en el control del sistema. Actualmente las señales EMGS son utilizadas en gran medida en el control de prótesis robóticas y exoesqueletos, ya que su adquisición constituye una de las técnicas más sencillas de implementar por su facilidad en la recolección sin intromisión directa sobre el organismo, además

presenta gran facilidad al momento de retirar los electrodos, simplificando su mantenimiento y calibración [6].

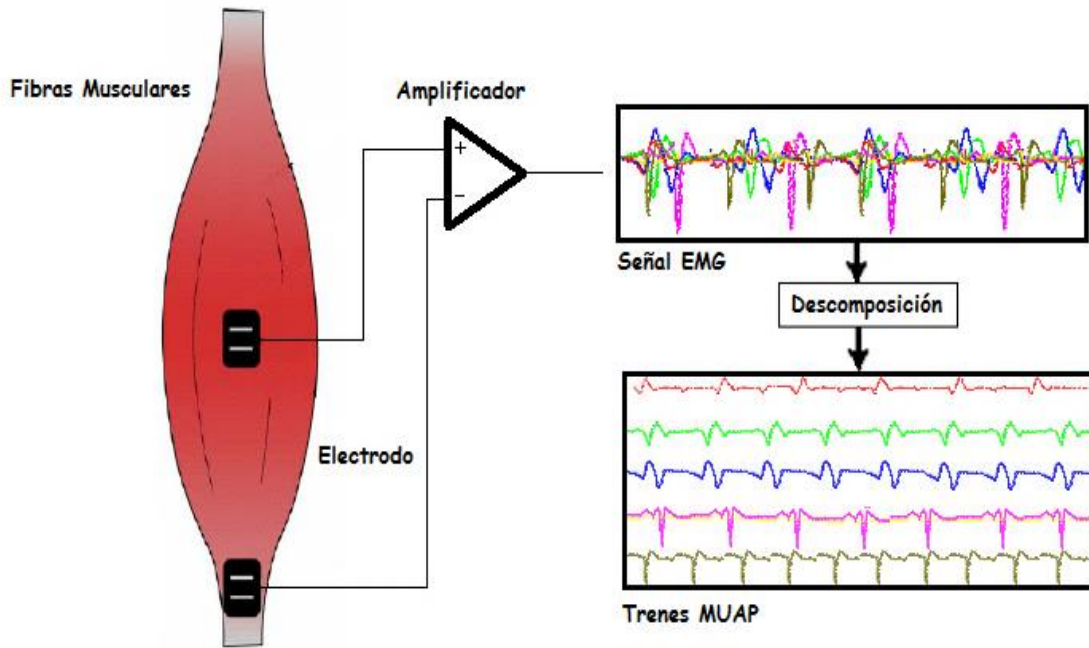


Figura 1. Detección de Señales EMG

## 1.2. RECONOCIMIENTO DE PATRONES

Es el proceso que tiene como intención la extracción de la mayor cantidad de información valiosa, que permita establecer propiedades de o entre conjuntos de diferentes objetos. El reconocimiento de patrones es descrito por algunos autores como un proceso de tres fases: *extracción de características*, *reducción de dimensionalidad* y *clasificación*, como se muestra en la figura 2.

La primera fase consiste en obtener una apropiada caracterización de las señales atenuando el ruido. Las características pueden extraerse del dominio temporal de la señal a partir de sus parámetros estadísticos o del dominio espectral de la señal mediante transformaciones como la transformada de Fourier (*FT*) o las transformadas wavelets *DWT* (*Discrete wavelet Transform*) y *WPT* (*Wavelet Package transform*). La segunda fase consiste en minimizar el número de características, eliminando datos que no representen información significativa o aquellos fuertemente correlacionados, conservando los más relevantes. Esto puede hacerse de dos maneras: por selección de características y por proyección. La tercera y última fase consiste en discriminar a partir de vectores conformados por las características extraídas, la clase a la que pertenece cada patrón. Existen clasificadores estadísticos, clasificadores estructurales y los más modernos denominados de aprendizaje [3].

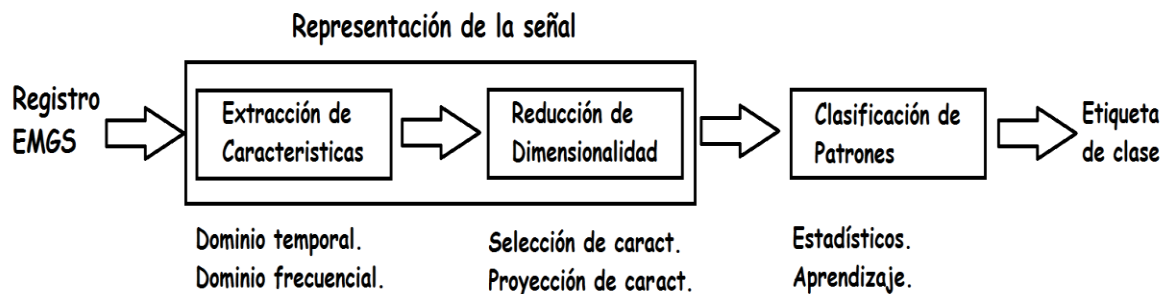


Figura 2. Fases en el Reconocimiento de Patrones

### 1.2.1. Extracción de Características de Señales EMGS

La extracción de características es un proceso mediante el cual los atributos de las señales son recolectados y asociados como componentes de un vector compacto.

El objetivo de este paso es crear una representación manejable y significativa de la señal EMG original, con el objetivo de maximizar el éxito potencial en la fase de clasificación y a su vez el rendimiento global del sistema. Un segundo objetivo de la fase de extracción de características es extraer la información útil comprimiendo los datos sin pérdida de información relevante, con el objetivo de reducir el número de variables de entrada que generalmente son de gran dimensión, por lo que conviene procesarla en secciones cortas (comúnmente de 256 muestras), y de esta forma incrementar el desempeño en la fase de clasificación [7].

En el registro mioeléctrico, los vectores de características que representan la información extraída de las señales EMG, son elegidos minimizando el error. Para lograr esto, el conjunto de características debe ser elegido de tal manera que se maximice la separación entre clases y minimice la distancia entre patrones de una misma clase. Así, la extracción de características es el punto principal del sistema de clasificación y es esencial para la caracterización de comandos de movimiento. [7]

Al interior de la fase de extracción de características los modelos admitidos para el modelamiento de señales EMG resultan ser estocásticos, por lo cual solo es posible obtener información de estas señales mediante las técnicas de extracción de características, dentro de las cuales se destacan las basadas en el análisis temporal por su facilidad de evaluación debido a que no requieren ningún tipo de transformación sobre la señal y las técnicas basadas en el análisis espectral, que involucran una transformación del dominio original de la señal [3].

A continuación se describen algunas de las técnicas más utilizadas en la fase de extracción de características para conformación de vectores descriptores.

#### **1.2.1.1. Análisis Temporal**

Cuando se habla de una secuencia de valores medidos en un determinado tiempo cronológico, que se encuentran espaciados uniformemente, se habla de una serie

temporal. Las características extraídas en una serie temporal se evalúan sobre segmentos muy breves de tiempo, típicamente 256ms. Este conjunto de características se evalúa sobre cada segmento y representa el patrón de la señal EMGS.

Las características en el dominio del tiempo no necesitan ninguna transformación por lo que generalmente se realizan más rápidamente, entre las más utilizadas tenemos:

- **Valor medio Absoluto (VMA):** Estima el valor medio absoluto de la señal, sobre cada segmento  $i$ , con longitud de  $N$  muestras. Se expresa en la ecuación 1.

$$\bar{X}_i = \frac{1}{N} \sum_{K=1}^N |x_k|, \quad i = 1, 2, \dots, I, \quad (1)$$

Donde  $x_k$  es la  $k^{ésima}$  muestra en el segmento  $i$ , siendo  $I$  el número total de segmentos.

- **Valor Medio Absoluto de Pendientes (VMAP):** Consiste en la diferencia entre las sumas de segmentos adyacentes,  $i$  e  $i+1$ , definido en la ecuación 2.

$$\Delta \bar{X}_i = \bar{X}_{i+1} - \bar{X}_i \quad i = 1, \dots, I - 1. \quad (2)$$

- **Cruce por Cero (CC):** Una forma simple para conocer la frecuencia de una señal es a partir del número de veces que la señal pasa por cero. Para este fin se debe definir un umbral que reduzca los cruces por cero por consecuencia del ruido. Ese se selecciona con respecto al voltaje máximo de la señal (su valor típico es 0.01 del valor pico de la señal). El contador de

cruces por cero se incrementa según la comparación entre dos muestras consecutivas  $x_k$  y  $x_{k+1}$  así:

$$x_k > 0 \text{ y } x_{k+1} < 0 \text{ o } x_k < 0 \text{ y } x_{k+1} > 0 \text{ y } |x_k - x_{k+1}| \geq 0.01V \quad (3)$$

- **Cambio de Signo de Pendiente (CSP):** Provee otra forma de medir el contenido de frecuencias en la señal mediante el conteo del número de veces que la pendiente de la forma de onda cambia de signo. Equivalentemente requiere de un umbral para reducir los cambios de signo de pendiente inducidos por el ruido. El contador de cambios de signo en la pendiente se incrementa según las relaciones:

$$x_k > x_{k-1} \text{ y } x_k > x_{k+1} \text{ o } x_k < x_{k-1} \text{ y } x_k < x_{k+1} \text{ y } |x_k - x_{k+1}| \geq 0.01V \text{ o } |x_k - x_{k-1}| \geq 0.01V \quad (4)$$

- **Longitud de la Forma de Onda (LO):** Suministra información de la complejidad de la forma de onda en cada fracción y está dada por la siguiente expresión:

$$lo = \sum_{k=1}^N |\Delta x_k|, \quad (5)$$

Donde  $\Delta x_k = x_k - x_{k-1}$  (diferencia entre valores de muestras consecutivas). Su valor da una idea de la amplitud, la frecuencia y la duración de la onda de la señal en un solo parámetro.

### 1.2.1.2. Análisis Tiempo – Frecuencia

El análisis tiempo-frecuencia permite la localización conjunta de sucesos en tiempo y frecuencia, siendo adecuado para el análisis de las señales no estacionarias, señales que no poseen una frecuencia constante durante el periodo de tiempo analizado

Una de las transformadas utilizadas para el tratamiento de señales, es la transformada de Fourier, este tipo de transformada es ideal en señales estacionarias, pero presenta bajos rendimiento en señales no estacionarias como las señales EMGS. En su lugar, se considera la STFT (*Short Time Fourier Transform*) que logra una representación simultánea de la señal en los dominios tiempo-frecuencia con un cierto grado de resolución, pero presenta la desventaja de poseer una resolución limitada para el dominio del tiempo y de la frecuencia. Por estas razones se decide el estudio de la transformada wavelet, la cual genera resultados sobresalientes en el análisis de señales EMGS.

La Transformada Wavelet (WT), a diferencia de la STFT hace posible una buena representación de una señal tanto en tiempo como en frecuencia de forma simultánea, con lo que se puede determinar el intervalo de tiempo en el cual aparecen determinadas componentes espectrales [7]. Este tipo de transformada pertenece a una serie de técnicas de análisis de señal denominadas comúnmente análisis multiresolución, técnica que estudia la señal de tal forma que cada componente de frecuencia es analizado con una resolución diferente y descompone una serie de tiempo en diferentes escalas temporales. Sus principales características son:

- Ofrece buena resolución temporal y baja resolución frecuencial en eventos de altas frecuencias.
- Ofrece buena resolución frecuencial y baja resolución temporal en eventos de bajas frecuencias.

Generalmente el uso de esta técnica tiene sentido cuando la señal a analizar tiene componentes de alta frecuencia con corta duración, y componentes de baja frecuencia con larga duración, como es el caso de la mayoría de las señales EMGS [8].

Existen diferentes tipos de transformadas *wavelet*, como la transformada *wavelet* *continúa* (CWT), la transformada *wavelet* *discreta* (DWT) y la transformada *wavelet* *packet* (WPT). Su principal diferencia radica en la forma en que los parámetros de translación y escala son discretizados. Estos métodos se describen a continuación [7]

- **Transformada Wavelet Continua (CWT):** La transformada *wavelet* constituye una técnica relativamente nueva que ha sido propuesta como una poderosa herramienta en el análisis sobre el comportamiento local de una señal. Al igual que la STFT, esta transformada utiliza una función ventana que encuadra una señal dentro de un intervalo y focaliza el análisis sólo en ese segmento de la señal. La transformada *continua wavelet* intenta expresar una señal  $x(t)$  continua en el tiempo, mediante una expansión de términos o coeficientes proporcionales al producto interno entre la señal y diferentes versiones escaladas y trasladadas de una función prototipo  $y(t)$  más conocida como función *wavelet* madre. La implementación digital de la transformada CWT puede hacerse mediante la integral de convolución, sin embargo, esta produce demasiados coeficientes con alta tasa de información redundante debido a la no ortogonalidad de sus funciones base. Por lo que resulta más práctico a nivel computacional implementar su versión discretizada [9].
- **Transformada Wavelet Discreta (DWT):** La WT tiene como fin tratar gran variedad de señales para su transmisión, estudio, etc. La mayor parte de las señales que se tendrán en cuenta para su manejo, se almacenan en formato digital, más eficiente y manejable. En caso de ser señales directamente



captadas del entorno, lo más frecuente es que, antes de pasar a la etapa de procesamiento, las señales sean preprocesadas, muestreadas y digitalizadas. Para aplicar la transformación Wavelet a señales discretas nace la Transformada Wavelet Discreta (DWT), que, además, resuelve los problemas de redundancia y de definición analítica de las funciones wavelet que sufre la CWT. Por tanto, es más frecuente el empleo de la DWT; concretamente, en el tratamiento de señales electrocardiográficas y electromiográficas. Básicamente la transformada wavelet discreta consiste en asociar valores discretos a las variables de escala  $a$  y traslación  $b$  según una escala diádica, esto es, haciendo  $a = 2^j$  y  $b = n2^j$ , de tal manera que el conjunto de funciones wavelets adquieren la forma.

$$\varphi_{j,n}(t) = 2^{-\frac{j}{2}}\varphi(2^{-j}t - n); \quad j, n \in \mathbf{Z}. \quad (6)$$

Donde el factor  $2^{-j/2}$  se introduce para garantizar la condición de ortonormalidad de las bases wavelets [3]. Con esto se logra mayor eficiencia computacional y una representación de la señal más compacta, pues la ortogonalidad de las bases evita la redundancia entre sus coeficientes.

- **Transformada de paquetes Wavelet (WPT):** La transformada de paquetes wavelets WPT es una generalización de la transformada DWT, donde la descomposición en cada nivel no se limita solamente a las bandas de aproximación, sino también sobre las bandas de detalles. De esta manera, una señal  $x[n]$  con  $N$  muestras de longitud genera en el primer nivel de descomposición, mediante filtraje pasa-bajas y pasa-altas, dos sub-bandas de longitud  $N/2$  conteniendo la mitad del espectro de la señal cada una (igual que la descomposición DWT). En el segundo nivel de descomposición el filtraje se realiza sobre estas dos sub-bandas, generando cada una dos sub-bandas mas, obteniéndose cuatro sub-bandas de longitud  $N/4$  cada una,

conteniendo cada una  $1/4$  del espectro de frecuencias. Así, del tercer nivel de descomposición resultan ocho (8) sub-bandas con los coeficientes wavelets identificados como  $\{W_{3,0}, W_{3,1}, \dots, W_{3,7}\}$  [3]. Este proceso de descomposición puede realizarse hasta un nivel máximo  $J \leq \log_2 N$ , resultando en  $JN$  coeficientes wavelets en total, y a pesar de que este procedimiento de descomposición es más completo que el de la descomposición DWT, su costo computacional aún resulta ser menor que el requerido por la transformada STFT, esto es  $O(JN) \leq O(N \log_2 N)$ .

### **1.2.2. Reducción de Dimensionalidad**

Teóricamente se pensaría que el tener un vector de características de mayor dimensión, daría más poder discriminatorio. Sin embargo, la experiencia de algunos autores en algoritmos de aprendizaje ha demostrado que no es siempre así, detectándose algunos problemas como tiempos de ejecución muy elevados, aparición de muchos atributos redundantes y/o irrelevantes. El papel fundamental en reducción de dimensionalidad es ofrecerle solo la información que es valiosa y necesaria al sistema de clasificación, reduciendo la dimensión de los vectores descriptores, minimizando la complejidad del proceso de clasificación e incremento su desempeño. Al interior de este apartado se tendrán en cuenta dos métodos de reducción de dimensionalidad, la reducción por selección de características y la reducción por proyección de características.

#### **1.2.2.1. Reducción por Selección de Características**

Este método trata de determinar el mejor subconjunto de características contenidas al interior del conjunto de características originales, mediante la selección y/o eliminación de características que presenten o carezcan de información relevante. Esta selección se realiza de acuerdo a criterios con los cuales se pueda juzgar cuando un subconjunto de características es mejor a otro.

El criterio ideal para la selección del mejor subconjunto debe ser la mínima probabilidad de error de clasificación, pero en la práctica este criterio resulta demasiado complejo y generalmente son aplicados criterios más simples basados en separabilidad de clases [3] [7].

El criterio de índice de separabilidad de clases (CSI - Class Separability Index) es un tipo de criterio que se fundamenta en la separabilidad de clases por distancia euclidiana, y se considera como un método de supervisión, ya que las características se clasifican utilizando información perteneciente a la clase [6].

#### **1.2.2.2. Reducción por Proyección de Características**

El método de proyección de características intenta determinar el mejor conjunto de características, a partir de la transformación del conjunto original de características, de esta forma se crea un nuevo conjunto de características de menor dimensionalidad. Esta técnica minimiza la complejidad de las características, resultando más útil y eficaz en el reconocimiento de patrones. Una de las técnicas más conocidas para la reducción de dimensionalidad y la cual es utilizada para que la representación de señales sea más precisa es el PCA (Principal Component Analysis), esta técnica busca conocer qué tan exacta se puede representar una señal con un conjunto de características, es decir, si un pequeño conjunto de características puede ser usado para reconstruir la señal exactamente, entonces se puede decir que las características son efectivas y son útiles para representar esta señal. Esta técnica es un procedimiento no supervisado en razón de que ningún criterio de pertenencia de clase califica los datos cuando se especifican los vectores propios [3] [7].

#### **1.2.3. Clasificación de Patrones**

La clasificación de patrones es fundamental en el sistema detector de intención de movimiento, su desempeño está fuertemente ligado al proceso de extracción de

características. La clasificación de patrones se considera como la última fase a implementar, ya que esta define la pertenencia de cada una de las señales a una clase en particular.

La clasificación de patrones según los métodos utilizados se pueden agrupar en diferentes categorías, dentro de las cuales encontramos técnicas tradicionales como modelos ocultos de Markov, algoritmos de k-medidas, arboles de decisión, métodos basados en distancias Euclidianas y Mahalanobis, pero generalmente en métodos estadísticos (o de decisión teórica), estructurales (o sintácticos) o de aprendizaje (o neuronales).

Los clasificadores estadísticos se basan en el análisis de las características estadísticas de los datos. Los datos se asignan a una clase particular de acuerdo a un modelo probabilístico de su función de densidad de probabilidad en un espacio N-dimensional, el cual se divide en regiones correspondientes a cada clase. Los clasificadores más difundidos en esta clase, por su precisión en la clasificación, se encuentra el Bayesiano y el analizador discriminante lineal (LDA), sin embargo, sus algoritmos tiene un costo computacional muy elevado.

Los clasificadores estructurales o sintácticos se basan en la utilización de la estructura de patrones y la interrelación entre componentes de un patrón. El reconocimiento de patrones estructurales o sintácticos involucra la identificación de componentes principales o “primitivas” de los patrones y el desarrollo de una sintaxis formal o “gramática” que describa la síntesis de los patrones a partir de sus primitivas, lo que los hace ver más como una forma de representación de señales que como una tarea de clasificación.

Y los clasificadores de aprendizaje o neuronales, que se basan en redes neuronales artificiales, son considerados como determinísticos debido a que un algoritmo de aprendizaje no asume nada de las características estadísticas de los patrones de una clase (como lo hacen los estadísticos), a pesar de ser los dos

muy similares en su forma y objetivos. Para efectos de evaluar el proceso de representación de las señales EMGS mediante el error de clasificación, se implementaron como clasificadores: el perceptrón multicapas *MLP (Multi Layer Perceptron)*, una red neuronal probabilística *PNN (Probabilistic Neuronal Network)* y una maquina soporte vectorial *SVM (Support Vector Machine)* [3].

### 1.2.3.1. Fundamentos Teóricos

- **Perceptrón Multicapa (MLP):** El modelo neuronal denominado perceptrón y su algoritmo de convergencia fue publicado en (1958) por Rosenblatt. Este modelo está formado por tres elementos fundamentales, las unidades sensoriales, unidades asociativas y unidades de respuesta. Cada perceptrón con una única neurona está limitado a la clasificación de patrones de dos clases o categorías, pero si expandimos la capa de salida incluyendo más de una neurona en ella, podremos clasificar correctamente más de dos clases con algunas limitaciones [10].

El desarrollo en el perceptrón de una única neurona genero un nuevo tipo de red cuya modificación principal respecto a la estructura del perceptrón se basa en el uso de varias capas de neuronas artificiales. El perceptrón multicapa *MLP (Multi Layer Perceptrón)* es una red neuronal artificial *ANN (Artificial Neural Network)*, supervisada, no lineal y no paramétrica, formada por múltiples capas que están dispuestas de forma secuencial de modo que las salidas de las neuronas de una capa son las entradas de las neuronas de la capa siguiente [11], como me muestra en la figura 3.

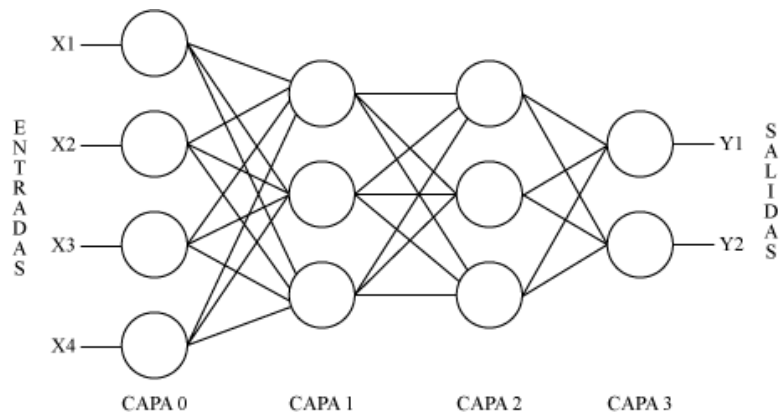


Figura 3. Estructura Perceptrón Multicapa

Cada neurona utiliza una función de base lineal, que consiste en la suma “pesada” de acuerdo a los pesos  $w_i$ , de las  $N$  componentes del vector de entrada  $X = [x_1, x_2, \dots, x_N]^T$ , a la cual se le agrega un valor inicial o “sesgo” de peso  $w_0 = \theta$ , según la ecuación 7.

$$s(x) = \sum_{i=1}^N w_i x_i + \theta \quad (7)$$

Esta función de base *lineal* define un hiperplano en el espacio de entrada. El resultado de la suma  $s(x)$ , se pasa por una función de activación *no lineal*  $f(*)$ , cuyo valor es proporcional a la distancia del vector de entrada  $x$  al hiperplano. La salida  $u$ , puede ser una etiqueta de clase o un nuevo valor de entrada para la siguiente capa de la red, de acuerdo a la ecuación 8.

$$u = f(s(x)) \quad (8)$$

Dentro de las funciones de activación no lineales frecuentemente empleadas como las sigmoides, se encuentra la función tangente hiperbólica y logarítmica que están definidas en la ecuación 9 y la ecuación 10.

$$f_{\tanh}(s) = \frac{e^{\beta s} - e^{-\beta s}}{e^{\beta s} + e^{-\beta s}} \quad (9)$$

$$f_{\log}(s) = \frac{1}{1 - e^{-\beta s}} \quad (10)$$

Esta configuración permite la división del espacio con fronteras de decisión arbitrariamente complejas, en este caso del clasificador MLP con algoritmo de aprendizaje en modo propagación hacia atrás “*BackPropagation*” (BP) [3], este algoritmo de aprendizaje consiste básicamente en transitar bidireccionalmente, hacia adelante y hacia atrás sobre las diferentes capas de la red. En el tránsito hacia adelante, se aplica en la capa de entrada un vector de entrada, este propaga su efecto a través de las diferentes capas y como consecuencia produce un vector de salida. Durante este proceso, los pesos sinápticos de la red son fijos y no se modifican. Durante el tránsito hacia atrás los pesos si se modifican de acuerdo con la regla de corrección del error. La señal de salida real se compara con la señal deseada y como resultado se obtiene una señal de error, que se propaga en dirección contraria a través de la red modificando los pesos, de forma que, al volver a pasar el vector de entrada hacia adelante, la respuesta obtenida se asemeje más a la salida deseada. [8]

Así pues, para un vector  $x \in \mathbf{R}^N$ , una red de  $N$  entradas,  $V$  capas con  $W_v$  neuronas en la capa  $v^{esima}$  y por tanto  $M = W_v$  salidas, representa una función  $F(x) : \mathbf{R}^N \rightarrow \mathbf{R}^M$ . El vector de salida  $y_1$  de la primera capa puede expresarse como en la ecuación 11.

$$y_1 = f_1(\mu_1 x + \theta_1) \quad (11)$$

Donde  $\mu_1$  es una matriz con  $N$  columnas y  $w_1$  filas, que contiene los pesos de las neuronas de la primera capa,  $\theta_1$  es un vector columna con los valores

iniciales o sesgos de las neuronas de la primera capa y  $f_1(*)$  es una función sin memoria que comprende las  $W_1$  funciones de activación de las neuronas de la primera capa [3].

- **Red Neuronal Probabilística (PNN):** Este tipo de red neuronal fue introducida por Specht en 1990, con el objetivo de ser utilizada en aplicaciones de tiempo real gracias a su simplicidad, generalidad y rapidez de aprendizaje. Este tipo de red constituye una variante dentro de la familia de las redes neuronales de base radial *RBFN (Radial Basis Function Network)*, en donde se define una relación no lineal entre las variables de entrada y las variables de salida de la red. Propagando hacia la salida toda la información recibida en la entrada y siendo su arquitectura muy similar a la de las redes MLP [12].

La arquitectura de la red PNN consta de cuatro capas, la primera capa se denomina de entrada y contiene  $N$  cantidad de nodos, correspondiente a los  $N$  rasgos del vector de entrada  $X = [x_1, x_2, \dots, x_N]^T$ , esta capa se encarga de distribuir los rasgos a una segunda capa de neuronas que es conocida como capa de patrones, la cual posee un número de neuronas igual al número de patrones de entrenamiento y se dividen en  $K$  diferentes grupos, en esta capa cada uno de los rasgos del vector de entrada es sometido a cada neurona con función de base radial. Así la  $i$ -ésima neurona del  $k$ -ésimo grupo evalúa su salida mediante un kernel gaussiano ecuación 12.

$$F_{k,i}(X) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\|X - X_{k,i}\|^2}{2\sigma^2}\right) \quad (12)$$

Una tercera capa o capa de sumatoria cuyo número de neuronas es igual al número de clases y una cuarta capa o de salida cuyos nodos son neuronas binarias que producen la decisión de clasificación. Al final se genera un 1 para la neurona con la mayor probabilidad y un 0 para todas las demás.



Comúnmente se dice que esta neurona se activa y las demás no lo hacen. En la figura 4 se muestra un esquema general de este tipo de red.

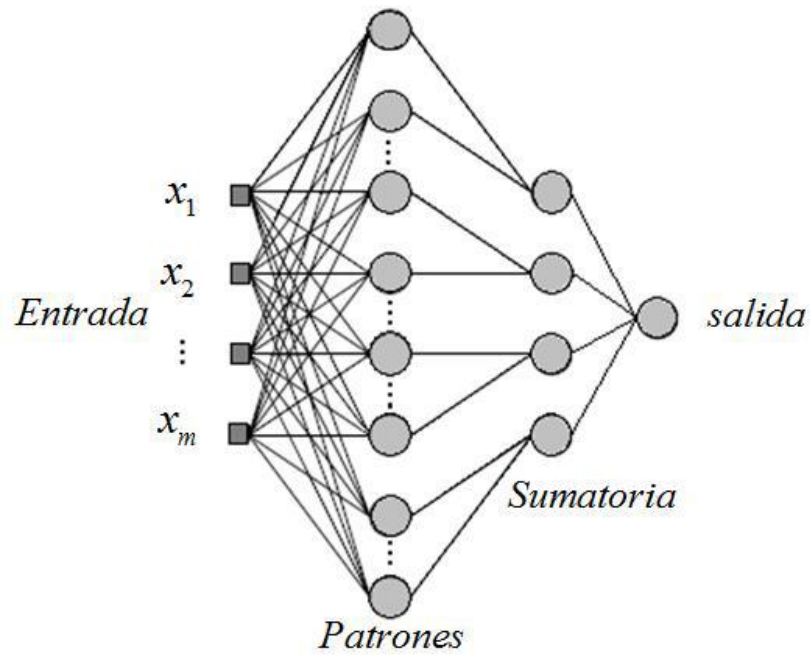


Figura 4. Arquitectura General de la Red PNN

Ya que una red PNN solo puede generar una configuración única, su entrenamiento requiere de una sola pasada sobre los patrones de entrenamiento, esto hace que su entrenamiento sea simple y rápido respecto a los tipos de redes neuronales como el MLP.

- **Maquina de Soporte Vectorial (SVM):** Las máquinas de soporte vectorial SVM (Support Vector Machine) se conocen como un conjunto de algoritmos de aprendizaje supervisado, desarrollado en 1992 por Vladimir Vapnik. Estos métodos se basan en la teoría de aprendizaje estadístico, básicamente herramientas estadísticas de optimización con características fundamentales como la ausencia de mínimos locales, alta capacidad de generación y de

control, utilizando el concepto general de la minimización del error, en donde se minimiza la probabilidad de una clasificación errónea sobre nuevos ejemplos.

Para solucionar un problema de clasificación, la SVM debe aprender una superficie de decisión adecuada, basándose en el conjunto de datos de entrenamiento [13]. La superficie de decisión es un hiperplano que separa los patrones de entrenamiento en dos clases o regiones como se muestra en figura (5a), según se encuentran a un lado u a otro.

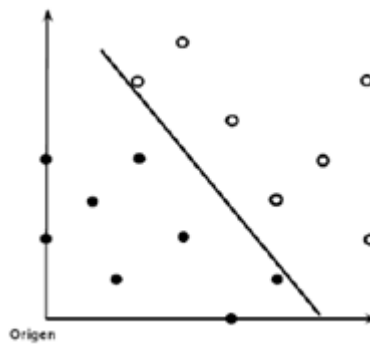


Figura 5a. División de los Datos en Dos Clases

Se dispone entonces de un conjunto de  $N$  datos de entrenamiento llamados patrones, de la forma  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  donde  $x \in \mathbb{R}^N$ ; cada escalar  $y_i$  (llamado etiqueta) corresponde a una de dos clases que se identificaran como +1 y -1. Se llamara vector de etiquetas al vector  $y = (y_1, y_2, \dots, y_N)$ .

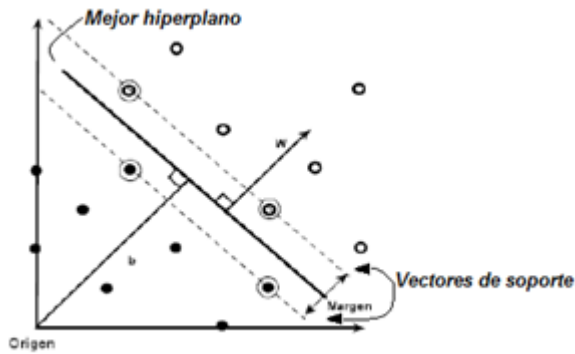


Figura 5b. División de los en Dos Clases Por el Mejor Hiperplano

En un problema linealmente separable existen muchos hiperplanos que pueden clasificar los datos. Pero la SVM no halla uno cualquiera de estos hiperplano, sino el único que maximiza la distancia entre él y el dato más cercano de cada clase, definido de manera general por la ecuación 13. Esta distancia es llamada margen, y el hiperplano que la maximiza, se llama hiperplano de máximo margen o vector de soporte. Figura (5b) [14].

$$\mathbf{0} = (\mathbf{W} \cdot \mathbf{x}) + \mathbf{b} \quad \text{En donde } \mathbf{W}, \mathbf{x} \in \mathbf{R}^N, \mathbf{b} \in \mathbf{R} \quad (13)$$

$$\mathbf{p}_{+1} = (\mathbf{w} \cdot \mathbf{x}) + \mathbf{b} = +1 \quad (14)$$

$$\mathbf{p}_{-1} = (\mathbf{w} \cdot \mathbf{x}) + \mathbf{b} = -1 \quad (15)$$

Asumiendo como  $1/\|\mathbf{W}\|$ , la distancia entre el punto más cercano y el mejor hiperplano, las ecuaciones de los dos planos paralelos estarán dadas por las ecuaciones (14), (15) y ya que teóricamente al interior de los vectores no se encuentra ninguna dato estos deben cumplir con las condiciones dadas por las ecuaciones (16), (17).

$$(\mathbf{w} \cdot \mathbf{x}) + b \geq +1 \text{ para } y_i = +1 \quad (16)$$

$$(\mathbf{w} \cdot \mathbf{x}) + b \leq -1 \text{ para } y_i = -1 \quad (17)$$

Por tanto puede observarse que la función de decisión  $f_{w,b}(x) = y_i$ , corresponde al signo que resulta de evaluar un dato en la ecuación 18 del hiperplano óptimo, por lo que puede escribirse como

$$f_{w,b}(x) = \text{sign}[g(x_i)] \quad (18)$$

Es así como el algoritmo de aprendizaje de la red SVM tiene que resolver un problema de optimización cuadrático para encontrar la solución óptima que separa con un margen máximo el conjunto de muestras.

### 1.2.3.2. Consideraciones para evaluar un clasificador

Dentro de las condiciones generales a tener en cuenta para la obtención de resultados óptimos de clasificación están [7]:

- **Exactitud y Precisión:** Representa el número o proporción de clasificaciones que se realizan correctamente. La precisión consiste en la capacidad de agrupar las diferentes respuestas en una zona determinada, de esta forma menor será la dispersión, mientras que la exactitud determina lo cerca que el resultado está del valor esperado o verdadero. En el caso de las señales EMG, por ejemplo, las variaciones de precisión y exactitud generalmente son el resultado de la mala ubicación de los electrodos sobre la piel o por interferencia (ruido).
- **Velocidad o Rapidez:** Es el tiempo calculado en que se toma una decisión, dependiendo de la aplicación, la velocidad puede resultar igual o más

importante que la precisión, por ejemplo, un clasificador con un 90% de precisión, sería preferible a uno con 95% de precisión, si el primero es 100 veces más rápido que el segundo.

- **Potencia de Clasificación:** Es la razón entre la tasa de éxito y el tiempo empleado en el proceso de clasificación, es decir, a mayor tasa de éxito y menor tiempo, se obtiene mayor potencia del clasificador.
- **Flexibilidad:** Adaptabilidad del clasificador a un entorno cambiante. Es decir, este debe comprender correctamente los vectores de características, incluso si esas características son considerablemente diferentes para cualquier característica que ha sido vista antes. Debe ser capaz de adaptarse a señales provenientes de diferentes fuentes, sin que esto afecte su buen desempeño.

### 1.3. DESCRIPCIÓN BASE DE DATOS

Las señales EMGS utilizadas para el desarrollo del presente trabajo están representadas en la base de datos “LeviData.zip” de Levi Hargrov, suministrada y autorizada por Kevin Englehart Ph.D, del Instituto de Ingeniería Biomédica de la Universidad de New Brunswick.



Figura 6. Tipos de Movimiento

Para la construcción de la base de datos los autores realizaron la adquisición de los datos en 6 pacientes para cuatro movimientos de mano (Figura 6), para este fin se utilizaron 16 electrodos superficiales, los cuales fueron ubicados linealmente alrededor del antebrazo de cada paciente, y espaciados 2cm uno del otro como se observa en la figura 7 (corte transversal del antebrazo). Este arreglo fue conectado a un sistema de grabación con preamplificación y configurado para mediciones diferenciales entre electrodos adyacentes. Cada uno de los 16 canales fue filtrado entre 10 y 500 Hz usando un filtro pasa banda tipo Besel de 4<sup>to</sup> orden con ganancia de 33. La salida analógica de la etapa de amplificación fue enviada a una tarjeta *DAQCard-6024E* de la *National Instrument* para su conversión digital. La base de datos obtenida se segmenta de la siguiente manera:

- Los registros de las señales EMGS están contenidos en una carpeta llamada “*Registros\_E\_EMG*”, esta carpeta contiene 6 archivos con extensión “.*mat*”, cada uno de los archivos contiene los datos de los cuatro movimientos para cada uno de los 6 pacientes.
- Para cada uno de los pacientes se utilizaron 16 canales, con los cuales se obtiene la información, cada canal representa la ubicación de un electrodo alrededor del antebrazo del paciente.
- Los registros de cada canal se fraccionan en un arreglo matricial de dimensiones 256x160 para cada paciente. Esto es 160 columnas con las muestras EMGS en donde cada columna contiene 256 datos.
- El arreglo matricial se subdivide en 40 columnas para cada uno de los 4 movimientos, así se tiene desde la columna 1 hasta la columna 20 el primer movimiento (movimiento 1, mano relajada), de la columna 21 hasta la columna 40 el segundo movimiento (mano abierta), de la columna 41 hasta la columna 60 el tercer movimiento (agarre fino) y de la columna 61 hasta la columna 80 el cuarto movimiento (agarre grueso) , a continuación se repite cada 20 columnas la secuencia de los movimientos hasta culminar en la columna 160.

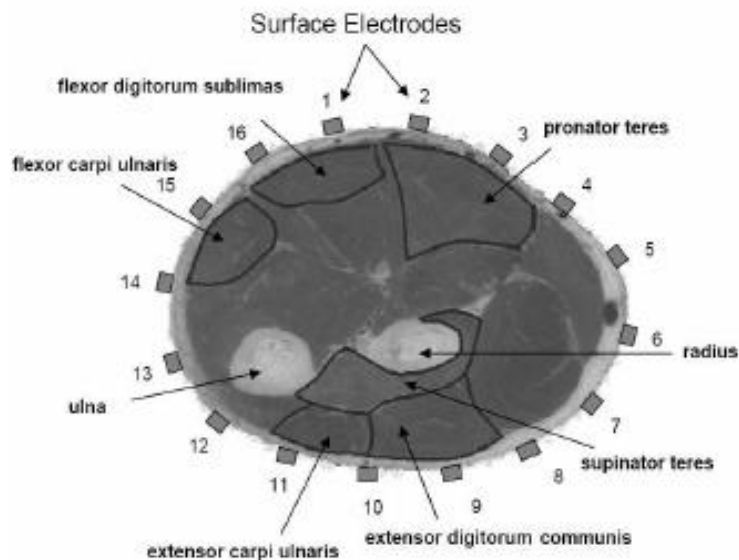


Figura 7. Ubicación de los Electrodoes en Vista Transversal del Antebrazo [3]

#### 1.4. DESCRIPCIÓN DE UN SISTEMA EMBEBIDO

Se conoce como Sistema Embebido (SE) a una combinación de hardware y software, diseñada para realizar una función específica, por lo cual difieren de los sistemas multifunción o de propósito general.

Los SE presentan una combinación que en muchos casos puede ser reemplazada por un circuito integrado que realice la misma tarea. Pero una de las ventajas principales de los SE, se encuentra en su flexibilidad, ya que a la hora de realizar alguna modificación resulta mucho más sencillo modificar algunas líneas de código que reemplazar todo un circuito integrado.

El cerebro de un SE es típicamente un circuito integrado como un microcontrolador, aunque los datos también pueden ser procesados por un DSP, una *FPGA (Field Programmable Gate Array)* un *ASIC (Application Specific*

*Integrated Circuit*), y su diseño está optimizado para reducir su tamaño, su costo y/o su consumo de energía, aumentar su confiabilidad y mejorar su desempeño [15].

Los SE constan de un programa ejecutándose al interior de un microprocesador, se debe tener claro que un SE necesita como obligación una cierta cantidad de memoria en donde el programa pueda ser almacenado, este espacio de memoria se encuentra dentro del mismo chip. Además de esto, los sistemas embebidos cuentan con una serie de entradas y salidas necesarias para la comunicación con periféricos externos. Sus líneas de entrada/salida soportan la conexión de sensores y actuadores indispensables para determinar variables físicas del medio en el cual está situado.

Generalmente, los sistemas embebidos emplean procesadores muy básicos, relativamente lentos y memorias pequeñas para minimizar los costos, puesto que una de las principales características de un sistema embebido son el bajo costo y bajo consumo de potencia.

#### **1.4.1. Circuitos Integrados**

Se le da el nombre de circuito integrado a una superficie pequeña de material semiconductor en el que se interconectan diversos dispositivos electrónicos como diodos, transistores, resistencias, etc. Al interior de esta clasificación encontramos diversos tipos de dispositivos como microcontroladores, DSP, FPGA, etc.

Los microcontroladores clásicos MCU, son circuitos integrados que contienen un procesador digital completo junto a diversos periféricos auxiliares que facilitan el desarrollo de las aplicaciones a las que se dedican. Su parecido con los DSP es muy grande, pero presentan algunas diferencias fundamentales ya que incorporan arquitectura y recursos especiales para poder controlar requerimientos específicos. De forma resumida puede decirse que los DSP son



microcontroladores dotados de los recursos físicos y lógicos necesarios para poder soportar las aplicaciones específicas del procesamiento digital de señales como bloques de memoria separados e independientes, capaces de lograr un menor costo al ser implementados en tareas que involucran una combinación de control y procesamiento de señal en donde resultarían engorroso utilizar sistemas analógicos [16].

#### 1.4.1.1. Principales Características de los CI

Un sistema para procesamiento digital de señales presenta una estructura general como la mostrada en la figura 8. En donde una señal analógica es obtenida a través de un sensor, esa señal eléctrica es transformada por medio de un conversor analógico – digital A/D en una secuencia numérica. Esta secuencia numérica ingresa a un procesador en donde se ejecuta una serie de líneas de código (procesamiento digital), a la salida de este proceso se obtiene una nueva y modificada secuencia numérica que es transformada en analógica por medio de un conversor digital – analógico D/A, al final esta señal analógica se transforma en una magnitud física por medio de un actuador [17].

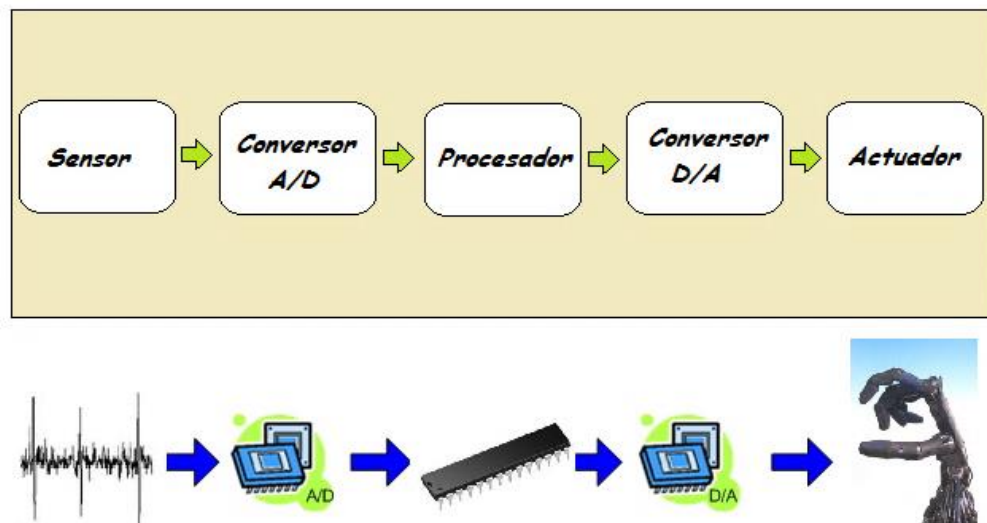


Figura 8. Etapas Para el Procesamiento de una Señal [17]

Un aspecto sumamente importante, es que las características del elemento procesador estarán impuestas por los requerimientos de las aplicaciones en qué se han de utilizar, y en las operaciones que se le realizaran a la señal. La operación básica es un producto acumulativo de dos secuencias, a las cuales se les denominan operaciones MAC (Multiply And Accumulate), dentro de la cual se encuentran aplicaciones como: filtrado, análisis espectral, etc. [18].

A continuación se describen las características más relevantes de los circuitos integrados:

- **Velocidad:** Para los sistemas electrónicos este parámetro expresa la velocidad de ejecución de las unidades básicas de trabajo. Este parámetro es medido generalmente tomando el tiempo de ciclo de instrucción que expresa el tiempo necesario para ejecutar una instrucción, su inverso dividido por un millón da lugar a la velocidad del procesador en millones de instrucciones por segundo o MIPS. En los CI la velocidad es sumamente importante ya que determina la aplicabilidad del dispositivo [17].
- **Formato Aritmético:** Una de las características fundamentales de los CI es el tipo de formato aritmético utilizado por el procesador. Este tipo de formato puede variar de un dispositivo a otro, pero en general se opera con números en punto fijo o en punto flotante (Figura 9). La representación de coma fija es una anotación en la cual se destina una cierta cantidad de dígitos para la parte entera y otra cierta cantidad a la parte fraccionaria, así se define la posición de la coma dentro del número. Al contrario, la representación en coma flotante expresa el número mediante un exponente y una mantisa.

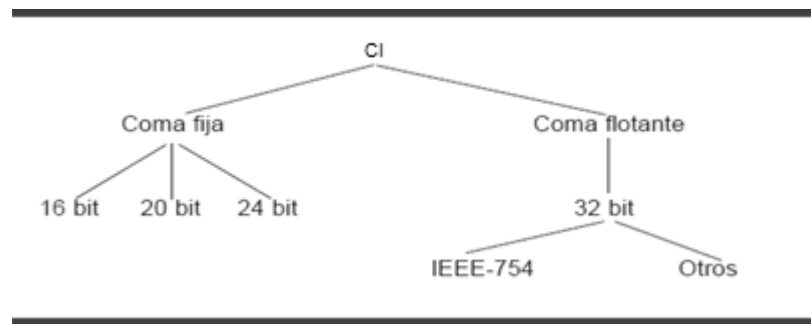


Figura 9. Representaciones numéricas Comunes en los CI

En el estándar IEEE-754 para la representación de coma flotante, el bit más significativo “s” representa el signo, donde un 1 indica que se trata de un número negativo y 0 positivo. La “e” indica exponente, este exponente está constituido por 8 bits y por último la “m”, de 23 bits que representa la mantisa del número (Figura 10 parte superior). El formato en coma fija al carecer de exponente, puede representar números con más bits significativos que el formato en coma flotante con un mismo tamaño en bits. Así por ejemplo en la figura 10 parte inferior, la representación en coma fija, “f” contiene 31 bits significativos, que es superior a los 23 del formato IEEE-754. [17]

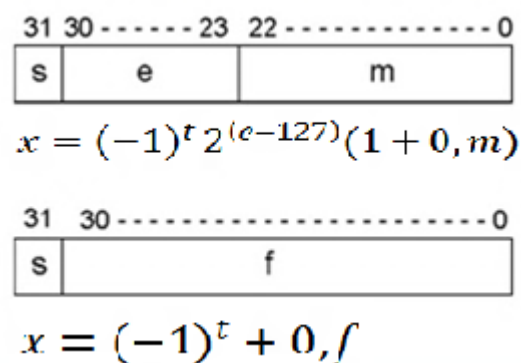


Figura 10. Formato Numérico con Coma Fija y Coma Flotante

Para un mismo tamaño en número de bits, el formato en coma fija proporciona una mejor resolución que el formato en coma flotante. Sin embargo, es este último quien permite un mayor rango en los números que se pueden representar. En consecuencia, los CI de coma flotante son generalmente más fáciles de programar que los de coma fija, pero son usualmente más caros. El mayor costo es resultado del requisito de una mayor complejidad circuital que se traduce en un mayor tamaño de chip. Sin embargo, el mayor margen dinámico facilita su programación pues el programador no debe preocuparse por el margen dinámico ni por la precisión. Por el contrario, en los CI de coma fija el programador a menudo debe escalar las señales en sus programas para asegurar una adecuada precisión numérica con el limitado margen dinámico de un CI de coma fija [18].

Por lo general, las aplicaciones con un gran volumen de unidades y/o bajo consumo utilizan los CI de coma fija al ser la prioridad en este tipo de aplicaciones el bajo costo. Los programadores determinan el margen dinámico y la precisión necesarias de la aplicación, ya sea analíticamente o a través de simulaciones, y entonces aplican operaciones de escalado dentro del código de la aplicación en los puntos en donde sea necesario.

En aplicaciones donde el costo no sea un requisito crítico o que demanden un margen dinámico y precisión elevadas, o donde la facilidad de desarrollo sea vital, los CI de coma flotante poseen ventaja.

- **Velocidad de Reloj:** Los sistemas electrónicos digitales se caracterizan por su velocidad de reloj. La velocidad de reloj se refiere a la velocidad a la cual el sistema ejecuta sus unidades de trabajo más básicas. Para los CI, la relación de la velocidad del reloj y la velocidad de muestreo es una de las características más importantes usadas para determinar cómo será implementado el sistema. Es decir, esta relación determina parcialmente el

monto de hardware necesario para implementar un algoritmo con una complejidad dada [17].

- **Velocidad de Muestreo:** Una característica fundamental de los CI es su tasa de muestreo (sample rate). El muestreo es el proceso mediante el cual se transforma una señal en tiempo continuo a una señal en tiempo discreto, en intervalos de muestreo también discretos. Las amplitudes de las señales en tiempo discreto se cuantifican en valores digitales con un ancho de palabra  $N = 2^n$  (donde  $n$  es el número de bits). Un ADC (Analog to Digital Converter) realiza los procesos de muestreo y cuantificación de una señal, de esta forma la tasa de muestreo  $f_M$  se puede replicar con exactitud con los valores tomados, si la tasa de muestreo  $f_s$  es superior al doble de la frecuencia máxima de la señal de entrada. Es decir,  $f_s \geq 2f_M$ .

Es un error frecuente y extendido creer que una misma señal muestreada con una tasa elevada se reconstruye mejor que una que se muestrea con una tasa inferior. Esto es falso (siempre que las tasas empleadas cumplan el criterio de Nyquist, naturalmente). El proceso de muestreo (que no debe ser confundido con el de cuantificación) es, desde el punto de vista matemático perfectamente reversible, esto es, su reconstrucción es exacta, no aproximada. Dicho de otro modo, desde el punto de vista matemático al que se refiere el teorema de muestreo de Nyquist-Shannon, la reconstrucción de una señal de 10 kHz es idéntica tanto si se obtiene de una tasa de muestreo de 25000 muestras por segundo como de una de 50000 muestras por segundo. No aporta nada incrementar la tasa de muestreo una vez que esta cumple el criterio de Nyquist. Cuando la cantidad de señales muestreadas es excesivamente mayor se produce el efecto de sobremuestreo, donde no se aporta nueva información y por el contrario, hay mayor demanda de memoria RAM y mayor tiempo de procesamiento. Esto puede ser un fuerte inconveniente en aplicaciones de procesamiento en tiempo real [17].

Si se utiliza una frecuencia menor a la establecida por el teorema de Nyquist, se produce una distorsión conocida como traslape espectral (aliasing). El traslape impide recuperar correctamente la señal cuando las muestras de ésta se obtienen a intervalos de tiempo demasiado largos. La forma de la onda recuperada presenta pendientes muy abruptas. Para ampliar los aspectos teóricos sobre el teorema del muestreo se pueden consultar las referencias [18].

- **Arquitectura de Memoria:** La arquitectura de Von Neumann es identificada por los procesadores que poseen el mismo dispositivo de almacenamiento en donde coexisten tanto datos como instrucciones. Las instrucciones y los datos son almacenados en el mismo formato dentro de la misma memoria, lo que quiere decir que las instrucciones son datos y comparten el mismo bus de datos en su transmisión figura 11. En principio un programa en ejecución se puede modificar a sí mismo y en la práctica usualmente esto significa que un error ocurrió. La ventaja de este tipo de arquitectura es que es eficiente en la utilización de la memoria pero su desventaja es que los tipos de datos no pueden ser reconocidos sin ambigüedad. La arquitectura se compone de las siguientes unidades o bloques: Unidad de memoria principal, Unidad aritmético-lógica. Unidad de control. Unidad de entrada y salida [18].

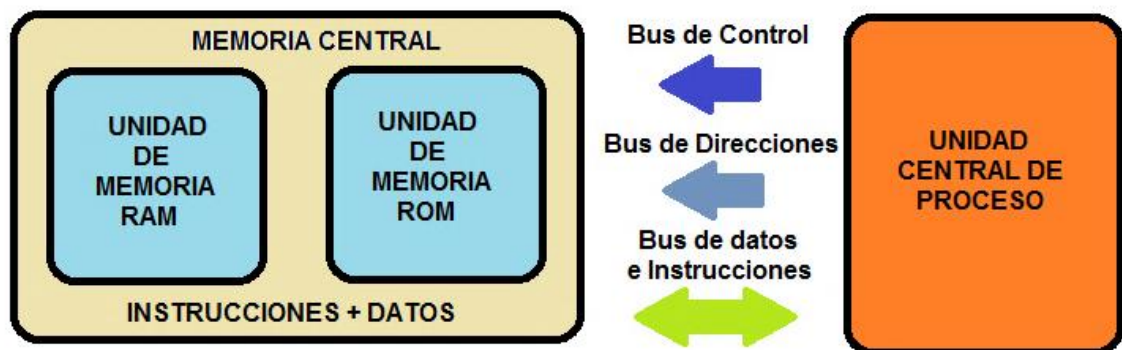


Figura 11. Arquitectura Von Neumann

La arquitectura Harvard básicamente posee una discrepancia muy marcada con respecto a la arquitectura Von Neumann, la cual es que los procesadores manipulan dispositivos de almacenamiento físicamente distanciados para el manejo de instrucciones y datos como se muestra en la figura 12 El acceso a cada espacio de memoria puede llegar a ser mediante buses distintos, es decir, existe un bus de datos y un bus de instrucciones; pero lo tradicional es que exista un único bus de direcciones, en cuyo caso debe existir alguna señal de control que permita diferenciar a que espacio de memoria hace referencia.



Figura 12. Arquitectura Harvard

Esta arquitectura asimismo se compone de las unidades o bloques de la arquitectura Von Neumann a diferencia del manejo de memoria con instrucciones y datos. La ventaja de la estructuración de este tipo de arquitectura es que los tipos de datos pueden ser reconocidos sin ambigüedad pero es ineficiente en la utilización de la memoria [18].

## 1.5. PROTOCOLO DE COMUNICACIÓN RS-232

El protocolo RS-232 es una norma o estándar mundial que gobierna los parámetros de uno de los modos de comunicación serial. Mediante este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, características físicas de los conectores, etc [20].

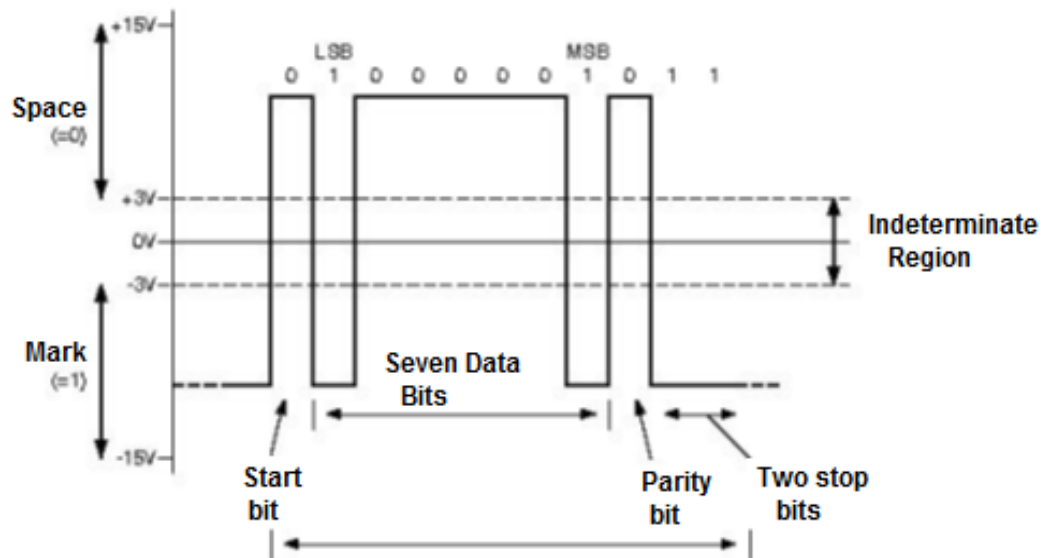


Figura 13. Características Eléctricas del Protocolo de Comunicación RS-232

### 1.5.1. Características Eléctricas

Este estándar especifica como lógica a la salida de una trasmisor un nivel alto con un voltaje entre +5 y +15 volts, mientras que un nivel bajo está definido con un voltaje entre -5 y -15 volts. Para el receptor fue diseñada permitiendo un nivel de ruido de 2 volts. Así, un nivel alto para el receptor está definido en el rango +3 a



+15 volts, mientras que un nivel bajo va desde los -3 a los -15 volts (Figura 13). Es importante notar que un nivel alto está representado por un valor lógico '0', mientras que un nivel bajo representa un valor lógico '1'.

Inicialmente, el estándar restringía la velocidad de transferencia de datos a 20[kbps]. Actualmente los circuitos integrados soportan velocidades mucho mayores, de hasta 350[kbps]. En un principio se constituyó un largo máximo del cable de 15 metros, pero luego fue modificado por la revisión D del estándar. Ésta especifica una máxima capacitancia de 2500[pF], en vez de establecer un largo máximo. Así, el largo máximo depende de la capacitancia característica del cable utilizado [20].

### 1.5.2. Características de conexión

El protocolo RS-232 incluye un conector tipo *DB-25* de 25 pines, aunque es normal encontrar la versión de 9 pines *DB-9*, más económico e incluso más extendido para cierto tipo de periféricos. En cualquier caso, los *PC* (Personal Computer) no suelen utilizar más de 9 pines en el conector *DB-25*. Las señales *TxD*, *DTR* y *RTS* son de salida, mientras que *RxD*, *DSR*, *CTS* y *DCD* son de entrada y presentan las siguientes funciones (Tabla 1) [20] [21].

Tabla 1. Especificaciones de Conexión Para un Conector DB-9

PIN	Señal	Descripción	E/S
1	-	Chasis	-
3	TxD	Transmisión de datos	S
2	RxD	Recepción de datos	E
7	RTS	Solicitud para envío	S

8	CTS	Listo para enviar	E
6	DSR	Grupo de datos listos	E
5	SG	Señal de tierra	-
1	CD/DCD	(Datos) Detección del portador	E
-	TxC(*)	Transmisión de reloj	S
-	RxC(*)	Recepción de reloj	E
4	DTR	Terminal de datos listo	S
9	RI	Indicador de llamada	E
-	RTxC(*)	Transmisión/recepción de reloj	S

### 1.5.3. Métodos de Transmisión en Serie

Existen dos métodos de transmisión en serie, la síncrona y la asíncrona. El primero es la comunicación síncrona en donde el emisor y el receptor son sincronizados utilizando una señal de reloj que indica el tiempo entre cada bit. Controlando esta señal, el receptor puede determinar si se ha perdido o se ha añadido un bit. . Antes de iniciar la comunicación ambos dispositivos deben de establecer una sincronización entre ellos. Para esto, antes de enviar los datos se envían un grupo de caracteres especiales de sincronía. Un aspecto a tener en cuenta en este tipo de comunicación es que si alguno de los extremos de la comunicación pierde la señal de reloj, la comunicación finaliza [21].

El segundo método es la comunicación asíncrona, este método es posiblemente la forma de conexión más extendida, en donde cada carácter (letra, número o símbolo) se introduce en una cadena de bits que está espaciado de las demás

cadenas mediante un bit de inicio y un bit de parada. El equipo destino manipula los marcadores de bit de inicio y parada para planificar sus funciones relativas al ritmo de recepción, de forma que esté preparado para recibir el siguiente byte de datos. En este método no existe un dispositivo reloj o método que permita coordinar la transmisión entre el emisor y el receptor por lo cual equipo emisor sólo envía datos y el equipo receptor simplemente los recibe. A continuación, el equipo receptor los comprueba para asegurarse de que los datos recibidos coinciden con los enviados. Las transmisiones asíncronas en líneas telefónicas pueden alcanzar hasta 28.800 bps. No obstante, los métodos de compresión de datos más recientes admiten pasar de 28.800 bps a 115.200 bps en sistemas conectados directamente no son necesarios los pulsos de reloj y la duración de cada bit está determinada por la velocidad con la cual se realiza la transferencia de datos.

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado alto. Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante un determinado tiempo, a lo cual se le conoce como bit de arranque (Start Bit) y a continuación empieza a transmitir en un intervalo de tiempo fijo, los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT más significativo. Si el receptor no está sincronizado con el transmisor, este desconoce datos. Por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de BIT de parada [21].

## **1.6. TRABAJOS PREVIOS EN LA CLASIFICACIÓN DE SEÑALES EMGS**

Las señales EMG se han convertido en una material fundamental en el desarrollo de múltiples trabajos de investigación y desarrollo. La clasificación de parámetros temporales o espectrales extraídos de dicha señal, se han convertido en un método de gran facilidad y buenos resultados en el momento de realizar su

implementación computacional. Después de realizar la clasificación de dicha señal, se puede considerar relativamente sencillo traducir estos resultados a instrucciones o comandos que permitan el control de algún sistema. Debido a esto, el procesamiento de señales EMG, ha tomado bastante fuerza en áreas como la medicina, en procesos de rehabilitación y detección de patologías, o en la industria en el control de distintos dispositivos. A continuación se realiza una recopilación de algunos de los trabajos más relevantes relacionados con el procesamiento de señales EMG utilizando técnicas de extracción de características basándose en la transformada Wavelet, así como también, en los cuales se utilizaron dispositivos procesadores de señal distintos al PC.

En el trabajo realizado por J. Salvador y H.de Bruin [22] se comparan los resultados obtenidos en la clasificación a partir de la transformada de Fourier con los resultados obtenidos en la clasificación hecha mediante la transformada wavelet, los datos que se pusieron a prueba fueron datos obtenidos de los músculos tenares en diez pacientes normales. Los resultados arrojados por este estudio, mostraron que la transformada wavelet es superior a la de Fourier en la clasificación de EMG M-wave con mejoras significativas en las variaciones inter e intra clases.

En [6], H.Romo, C. Realpe y E. Jojoa presentan una revisión de las técnicas más utilizadas en análisis y extracción de características discriminantes de las señales motoras EMGS con fines de aplicación en el control de prótesis virtuales de mano; haciendo énfasis en la evolución de las técnicas temporales clásicas a las técnicas espectrales más recientes basadas en transformadas wavelets, y su aceptación para procesar este tipo de señales según su desempeño en la clasificación y ejecución de movimientos sobre una prótesis.

En [23], Z. Xizhi utiliza la transformada wavelet para el análisis y la extracción de los coeficientes wavelet a partir de los cuales se encuentran los valores propios para dos canales de señales EMGS. La clasificación se hace por medio de una

red neuronal de base radial (RBF). Los resultados muestran que la precisión de la red neuronal RBF es mayor que la de una red neuronal con BP. Un trabajo similar es el realizado por L. Zhihong y L. Zhizeng en [24], en el que se utiliza la transformada de paquetes wavelet (WPT) para la extracción de patrones y, al igual que en el trabajo anteriormente mencionado, se utiliza una red neuronal con la diferencia de que en este caso se utiliza una de aprendizaje de cuantización vectorial (LVQ). Este clasificador puede identificar exitosamente cuatro movimientos tales como: extensión de la muñeca, flexión de la muñeca, extensión de la mano y la mano en agarre, midiendo las señales EMGS a través de dos electrodos montados en el antebrazo. Los resultados obtenidos muestran que el método propuesto alcanza una precisión de reconocimiento del 98%. Además, a través de una comparación cuantitativa con otros clasificadores de redes neuronales, el método LVQ tiene un mejor desempeño.

P. Parimal, S. Allen y G. Rapach expuesto en [25], en el que se utiliza una máquina de soporte vectorial como clasificador y la técnica de wavelet para la extracción de características, con la diferencia de que en este caso, se tienen ocho clases para la clasificación. Los resultados obtenidos muestran que la tasa de reconocimiento promedio es de más del 90%. La precisión del clasificador SVM es significativamente mejor que el clasificador RBF red neuronal.

En el área de programación de algoritmos de control en dispositivos digitales se han desarrollado diversos trabajos que consideran la implementación de sistemas de tiempo real de control a través de señales mioeléctricas (EMG). Uno de los primeros trabajos es expuesto en [25], donde se realiza un sistema microcontrolado, basado en un microcontrolador 68HC11. El sistema amplifica las señales mioeléctricas, las filtra, las digitaliza y el algoritmo decide el movimiento de un robot de dos grados de libertad.

Otro trabajo similar, es el de Fermo, Vincenzo y Dynnikov realizado en el año 2000 [26], en el cual, se captan las señales mioeléctricas, con el fin de controlar

una prótesis mioeléctrica de la extremidad superior. El análisis de la señal se lleva a cabo a través de software que se ejecuta en un microcontrolador que decide cómo abrir o cerrar la mano artificial. Las facilidades para hacer cambios en la forma de actuar por el software controlador, hace atractivo su uso para el paciente ya que se pueden elegir distintas funciones (por ejemplo, si la mano es abierta, cerrada, semicerrada, etc.) las cuales son fácilmente añadidas a la prótesis con una simple modificación en el programa del microcontrolador, sin ningún tipo de alteración de tipo hardware

En sistemas orientados a la asistencia en ambientes de trabajo industrial se encuentra el ejecutado por N. López, F. di Sciasco, E. Orosco, N. Ledesma, A. Echenique, and M. Valentinuzzi [27], en donde se expone el acondicionamiento de cuatro señales mioeléctricas para procesarlas en un PC y comandar un robot industrial. Posteriormente, el algoritmo implementado en el PC es modificado para ejecutarlo desde el dispositivo autónomo de DSP.

Entre los trabajos recientes se tiene como el ejecutado en [28], se implementa un algoritmo de procesamiento en un dispositivo digital de altas prestaciones, como lo es un procesador digital de señales (DSP). Se busca la integración del sistema, disminuyendo su tamaño, haciendo el diseño más anatómico y físicamente menos rígido, mejorando la aceptabilidad de los usuarios amputados al controlar manipuladores. Además, se busca una portabilidad del producto en caso de usar un manipulador móvil, como en el caso de una prótesis. Por último se tiene [29], en el cual presentan un sistema de adquisición de señales EMG para el control de un interfaz hombre-máquina. La señal EMG se somete a una etapa de extracción de características para luego generar la señal de control. El sistema está basado en un DSP *TMS320F2812* de la Texas Instruments.

## 2. SELECCIÓN E IMPLEMENTACIÓN DEL ALGORITMO DE CLASIFICACIÓN

En [3] H. Romo implementó una serie de algoritmos de clasificación los cuales difieren, principalmente, en las técnicas de extracción de características (*Temporales, Transformada de Paquetes Wavelet o Transformada Wavelet Discreta*) y los métodos de clasificación ahí utilizados (*Perceptrón Multicapas, Red Neuronal Probabilística o Máquina de Soporte Vectorial*). Estos algoritmos fueron realizados con ayuda del paquete de simulación de Matlab 7.0, con la utilización de las librerías del “*toolbox*” de *wavelet* versión 3.0.3 y las librerías del paquete *wavelab850*. La parte central de este capítulo se basa en la selección del algoritmo más adecuado entre los encontrados en [3], teniendo en cuenta criterios de selección como: tiempo de ejecución, error de clasificación, consumo de memoria, sencillez en la implementación. Además, en esta sección, se realiza una breve descripción de las principales características tanto del algoritmo seleccionado como de los demás algoritmos analizados. Por último, se muestra la metodología utilizada en la adecuación de dicho algoritmo al lenguaje de programación C, haciendo énfasis en la descripción de las principales características que tiene el algoritmo obtenido.

### 2.1. ELECCIÓN DEL ALGORITMO DE CLASIFICACIÓN

El proyecto macro *Prototipo Funcional de Prótesis de Mano Robótica* del Grupo de I+D en Automática Industrial del Departamento de Electrónica, Instrumentación y Control, del cual hace parte el presente trabajo, tiene como uno de sus principales objetivos el de construir una prótesis real accionada por las señales electromiográficas captadas directamente del brazo del paciente. Para alcanzar este objetivo, es necesario que cada uno de los componentes que hagan parte de dicho prototipo tenga características especiales que permitan que el sistema total tenga un funcionamiento adecuado y coherente con las necesidades del paciente.

Teniendo en cuenta estas características, se plantean a continuación los criterios considerados en la selección del algoritmo de clasificación más indicado.

### 2.1.1. Algoritmos de Clasificación Disponibles

En [3] se encuentran seis procesos de clasificación, los cuales difieren en algunas de sus especificaciones tales como: la técnica de extracción de características, el tipo de patrones que se tienen en cuenta para la clasificación y el tipo de clasificador.

A continuación se hace una breve descripción de cada uno de los seis procesos de clasificación de los cuales se dispuso, así como también la valoración dada a cada uno de los criterios de selección anteriormente mencionados para cada uno de los distintos procesos de clasificación.

- **Algoritmo 1 (DT – DWT – PNN):** Este algoritmo de clasificación extrae características de tiempo (DT) y espectrales al aplicar la transformada *Wavelet discreta* (DWT) sobre la señal EMG. Además, en el proceso de clasificación utiliza una red neuronal probabilística (PNN).
- **Algoritmo 2 (DT - WPT- PNN):** Al igual que el algoritmo anterior, éste extrae características de tiempo (DT) y utiliza una Red Neuronal Probabilística (PNN) en el proceso de clasificación. Sin embargo, en la extracción de características espectrales aplica la transformada de paquetes *Wavelet* (WPT).
- **Algoritmo 3 (DT – DWT – MLP):** Este algoritmo de clasificación extrae características de tiempo (DT) y espectrales al aplicar la transformada *Wavelet* discreta (DWT) sobre la señal EMG. En el proceso de clasificación utiliza el Perceptrón Multicapas (MLP).



- **Algoritmo 4 (DT- WPT – MLP):** Al igual que el algoritmo número 3, éste extrae características de tiempo (DT) y utiliza el Perceptrón Multicapas (MLP) en el proceso de clasificación. Sin embargo, en la extracción de características espectrales aplica la transformada de paquetes *Wavelet* (WPT).
- **Algoritmo 5 (DT – DWT – SVM):** Este algoritmo de clasificación extrae características de tiempo (DT) y espectrales al aplicar la transformada *Wavelet* discreta (DWT) sobre la señal EMG. En el proceso de clasificación utiliza una Máquina de Soporte Vectorial (SVM).
- **Algoritmo 6 (DT - WPT –SVM):** Al igual que el algoritmo anterior, éste extrae características de tiempo (DT) y utiliza una Máquina de Soporte Vectorial (SVM) en el proceso de clasificación. Sin embargo, en la extracción de características espectrales aplica la transformada de paquetes *Wavelet* (WPT).

### 2.1.2. Criterios de Selección

- **Tiempo de Ejecución:** En el desarrollo de una prótesis de mano funcional robótica es fundamental que el sistema tenga una respuesta relativamente rápida ante los estímulos dados por el paciente, por esta razón se considera que el tiempo de ejecución del algoritmo de clasificación es uno de los principales criterios de selección a tener en cuenta en esta etapa.

El tiempo que se tiene en consideración consiste en el requerido en extraer las características de los canales evaluados y realizar la clasificación de los patrones obtenidos en el paso anterior con el respectivo clasificador.

Para hacer la medición de estos tiempos, se utilizaron las funciones de Matlab 7.0 *TIC TOC* y *CPUTIME*, las cuales miden el tiempo que tarda en

ejecutarse una función o instrucción específica. Estas funciones entregan los tiempos en segundos, siendo estos bastante similares. La forma de utilizar cada una de estas funciones se muestra a continuación en las tablas 2 y 3.

Tabla 2. Uso de la función CPUTIME

t=cputime	Se guarda en la variable t, el tiempo que la unidad central de proceso usó para procesar las instrucciones anteriores a la función que se quiere evaluar.
Función o instrucción a evaluar	En esta parte se ubica la función o instrucción a la que se le quiere medir el tiempo de ejecución
e = cputime-t	Al tiempo que la unidad central de proceso ha usado hasta el momento, se le resta el utilizado por la misma hasta antes de iniciar la función que se quiere evaluar

Tabla 3. Uso de la función TIC-TOC

Tic	Se inicia un cronómetro
Función o instrucción a evaluar	En esta parte se ubica la función o instrucción a la que se le quiere medir el tiempo de ejecución
Toc	Se utiliza para leer el tiempo del cronómetro que fue iniciado con la instrucción tic.

A cada una de las funciones se les evaluó el tiempo de ejecución en repetidas ocasiones para así realizar un promedio de dichas mediciones. El tiempo que se determinó a partir de estas funciones es el tiempo total requerido para definir el tipo de movimiento referente a la señal procesada.

- **Error de Clasificación:** Para evaluar este criterio se utilizaron los resultados obtenidos por H. Romo en [3] (tabla 4).

Tabla 4. Error de clasificación para las técnicas DT-DWT y DT-WPT

PACIENTE	DT-DWT PNN	DT-WPT PNN	DT-DWT SVM	DT-WPT SVM	DT-DWT MLP	DT-WPT MLP
PACIENTE 1	5.000 %	0.000 %	0.000 %	4.375 %	1.250 %	0.625 %
PACIENTE 2	0.625 %	2.500 %	1.875 %	2.500 %	0.000 %	0.625 %
PACIENTE 3	0.000 %	2.500 %	0.000 %	1.250 %	1.250 %	1.875 %
PACIENTE 4	8.125 %	11.250 %	6.875 %	6.875 %	3.125 %	3.125 %
PACIENTE 5	4.375 %	6.875 %	3.750 %	6.875 %	3.125 %	3.750 %
PACIENTE 6	24.375 %	10.625 %	26.250 %	10.000 %	6.875 %	5.000 %
PROMEDIO	7.083 %	5.625 %	6.485 %	5.312 %	2.604 %	2.500 %

En la tabla anterior se muestran los errores de clasificación para cada uno de los seis pacientes de acuerdo a la técnica de extracción de características (DT-DWT o DT-WPT) y al tipo de clasificador (MLP, PNN o SVM) utilizado. De los datos mostrados en la tabla 4 se tomó, para la evaluación de este criterio, el promedio de los seis errores de clasificación de cada algoritmo. A modo de ejemplo, se toma el primer algoritmo de clasificación (PNN-DT-DWT), para evaluar el criterio de error de clasificación, se toma el valor de

7.083% que corresponde al promedio de los errores de clasificación de los seis pacientes.

- Sencillez en la Implementación:** Para evaluar este criterio se hizo una revisión de la estructura de los algoritmos haciendo énfasis en las funciones más complejas y extensas. Se buscó, en cada uno de los algoritmos, las funciones de Matlab que tuvieran un remplazo directo en C, por lo tanto, se determinó que un algoritmo sería menos complejo que otro, siempre y cuando tuviera más funciones con sus respectivas funciones análogas en C.
- Consumo de Memoria:** Al igual que el tiempo de ejecución de cada uno de los programas, el consumo de memoria es un factor de suma importancia en los sistemas embebidos. Este criterio se evaluó a partir de los comandos de Matlab *WHO* y *WHOS*, los cuales permiten ver información como: Su dimensión (en caso de ser un arreglo), valor que tiene asignado, número de bytes que ocupa en memoria etc. Esta información es tomada a partir del tamaño de todas las variables creadas y utilizadas para el óptimo rendimiento del algoritmo y que se guardan en el Workspace de Matlab. A continuación se muestra, a modo de ejemplo, la forma de utilizar estas funciones (Tabla 5).

Tabla 5. Uso de los comandos WHO y WHOS de Matlab

INSTRUCCIÓN	RETORNA
<code>x=2.2</code>	2.2000
<code>arreglo = [1:1:10]</code>	arreglo =   1   2   3   4   5   6   7   8   9 10
<code>Letra='a'</code>	Letra = a

3+5	ans=8																				
who	Your variables are: Letra a ans arreglo x																				
whos	<table border="1"> <thead> <tr> <th>Name</th> <th>Size</th> <th>Bytes</th> <th>Class</th> </tr> </thead> <tbody> <tr> <td>ans</td> <td>1x1</td> <td>8</td> <td>double array</td> </tr> <tr> <td>arreglo</td> <td>1x10</td> <td>80</td> <td>double array</td> </tr> <tr> <td>letra</td> <td>1x1</td> <td>2</td> <td>char array</td> </tr> <tr> <td>x</td> <td>1x1</td> <td>8</td> <td>double array</td> </tr> </tbody> </table>	Name	Size	Bytes	Class	ans	1x1	8	double array	arreglo	1x10	80	double array	letra	1x1	2	char array	x	1x1	8	double array
Name	Size	Bytes	Class																		
ans	1x1	8	double array																		
arreglo	1x10	80	double array																		
letra	1x1	2	char array																		
x	1x1	8	double array																		

En la tabla 5 se muestra la forma en que se deben utilizar los comandos WHO y WHOS de Matlab, así como también los parámetros que retornaría en distintas ocasiones. De estos parámetros se tomó el consumo de memoria dado en bytes para la evaluación de este criterio.

### 2.1.3. Evaluación de los Criterios de Selección

En la tabla 6 se muestran los resultados de tres de los cuatro criterios de evaluación; *Tiempo de Ejecución*, *Error de Clasificación* y *Consumo de Memoria*. En cuanto al criterio *Sencillez en la Implementación*, no es mencionado aquí ya que la valoración de éste no corresponde a una calificación cuantitativa como los demás.

Tanto el *Tiempo de Ejecución* como el *Consumo de Memoria* están divididos en dos columnas, una para el proceso de extracción de características y la otra para el proceso de clasificación de patrones, mostrando el total que corresponde a la suma de los valores de cada columna. Por otra parte, en cuanto al *Error de Clasificación*, se muestra el promedio de los errores de clasificación para cada uno

de los pacientes, tomado directamente de los resultados obtenidos en [3] mostrados en la tabla 4. Por ejemplo, el algoritmo de clasificación número 1 (DT – DWT - PNN) tardó 0.0266 segundos en extraer las características de las dos señales EMG y conformar el vector de patrones, además, tardó 0.411 segundos en realizar la clasificación de dicho vector, consumiendo así un tiempo total de 0.0677 segundos en todo el proceso. Por otra parte, tuvo un error de clasificación del 7.083 % al realizar la clasificación de 160 señales EMG. En cuanto al consumo de memoria, gastó un total de 99787 bytes de los cuales 33256 corresponden a la etapa de extracción de características y 66531 a la etapa de clasificación.

Tabla 6. Evaluación de los criterios de selección

Algoritmo de clasificación	Tiempo de ejecución (s)		Error de clasificación (%)	Consumo de memoria (Bytes)	
	<i>Extracción de características</i>	<i>Clasificación</i>		<i>Extracción de características</i>	<i>Clasificación</i>
DT– DWT	0.0266	0.0411	7.083	33256	66531
PNN	Total=0.0677			Total=99787	
DT-WPT	0.0375	0.0411	5.625	23320	66531
PNN					

	Total=0.0786			Total=89851	
DT-DWT MLP	0.0266	0.0569	2.604	33256	36413
	Total=0.0835			Total=69669	
DT-WPT MLP	0.0375	0.0569	2.500	23320	36413
	Total=0.0944			Total=59733	
DT-DWT SVM	0.0266	0.0061	6.458	33256	30096
	Total=0.0327			Total=63352	
DT-WPT SVM	0.0375	0.0061	5.312	23320	30096
	Total=0.0436			Total=53416	

#### 2.1.4. Selección del Algoritmo de Clasificación

A continuación se realiza la selección del algoritmo de clasificación de acuerdo a los criterios mencionados y evaluados en la sección anterior. Este proceso de selección se realizó proporcionando una puntuación, entre 1 y 6, a cada uno de los criterios para cada algoritmo de clasificación. La puntuación de 1 significa poca viabilidad del algoritmo y la de 6 mucha viabilidad en cuanto al criterio que se esté calificando. Después de obtenidos todos los puntajes se obtuvo el promedio de los criterios para cada algoritmo, y a partir de este se eligió el más indicado.

A modo de ejemplo se toma el criterio denominado *tiempo de ejecución*. Se puede notar (Tabla 6) que el algoritmo que consumió menos tiempo en su ejecución fue el número 5 (DT– DWT - SVM), por lo tanto a este le corresponde la calificación más alta que es de 6 puntos. Del mismo modo, el algoritmo que consumió menos tiempo, después del número 5, fue el algoritmo número 6 (DT– DWT - SVM) al cual se le asignó la segunda puntuación más alta, 5 puntos y así sucesivamente para el resto de los algoritmos.

En la siguiente tabla (Tabla 7) se pueden observar las puntuaciones de cada criterio de selección, así como el promedio de dichas puntuaciones para cada uno de los seis algoritmos de clasificación.

Tabla 7. Puntuación de los criterios de selección

Algoritmo de clasificación	Tiempo de ejecución	Error de clasificación	Consumo de memoria	Sencillez en la implementación	Promedio
DT– DWT PNN	4	1	1	5	2.75



DT-WPT PNN	3	4	2	4	3.25
DT-DWT MLP	2	5	3	6	4
DT-WPT MLP	1	6	5	2	3.5
DT-DWT SVM	6	2	4	3	3.75
DT-WPT SVM	5	3	6	1	3.75

De acuerdo a la tabla 7, se puede concluir que el proceso de clasificación seleccionado, según el método de selección realizado, es el número 3 (DT– DWT - MLP).

### 2.1.5. Descripción del Algoritmo Seleccionado

A continuación se describen las principales características (Proceso de extracción de características temporales – Proceso de extracción de características de frecuencia - Tipo de clasificador) del algoritmo seleccionado:

- I. En el proceso de extracción de características temporales se tuvieron en cuenta las siguientes características:
  - El valor absoluto medio de cada una de las 2 señales EMG representadas en un vector de 256 posiciones de tipo double.

- La longitud de onda de cada uno de las 2 señales EMG representadas en un vector de 256 posiciones de tipo double.
- II.** En el proceso de extracción de características de frecuencia se tuvieron en cuenta las siguientes cuatro características, obtenidas al aplicar la transformada WAVELET discreta (DWT) a cada señal:
- El promedio absoluto de coeficientes extraídos a partir de la WDT.
  - La varianza de coeficientes extraídos a partir de la WDT.
  - La energía de escala de coeficientes extraídos a partir de la WDT.
  - Los valores singulares de los coeficientes no nulos.

Es importante decir que esta última característica (valores singulares de los coeficientes no nulos) no fue tomada en cuenta al momento de realizar la implementación del algoritmo, debido a la gran complejidad y carga computacional que esta implicaba. De igual manera, se consiguieron resultados igual de satisfactorios que los obtenidos incluyéndola

- III.** En cuanto a la etapa de clasificación se utilizó un Perceptrón Multicapa con las siguientes características:
- 12 entradas para los 12 patrones que conforman el vector híbrido de tiempo y frecuencia.
  - Dos capas ocultas, ambas con 6 neuronas (Se pueden realizar cambios en esta configuración dependiendo del paciente para obtener mejores resultados).
  - Una capa de salida con 1 neurona que define la intención de movimiento del paciente (primer movimiento 1; segundo movimiento 2; tercer movimiento 3; cuarto movimiento 4).

- La función de activación para las neuronas de la primera y segunda capa es una tangente hiperbólica sigmoïdal (Tansig) figura 14, mientras que para la capa de salida es una función lineal (Purelin) figura 15.

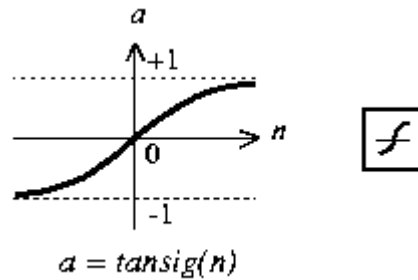


Figura 14. Función de activación tangente hiperbólica sigmoïdal.

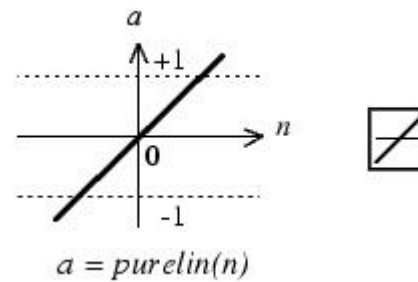


Figura 15. Función de activación lineal.

El algoritmo de entrenamiento de la red utilizado fue propagación hacia atrás (*BackPropagation*) Figura 16.

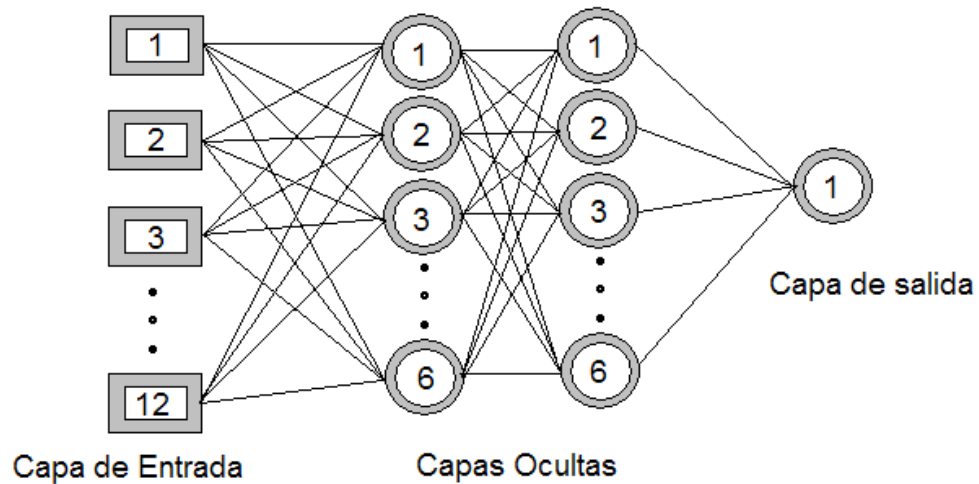


Figura 16. Arquitectura del Perceptrón Multicapas del algoritmo seleccionado

**Nota:** Al no incluir la característica “valores singulares de los coeficientes no nulos”, la arquitectura del perceptrón cambia en el número de entradas, siendo ahora de 10 posiciones ya que se pierde por cada señal una posición en el vector de entrada.

## 2.2. ADECUACIÓN DEL ALGORITMO DE CLASIFICACIÓN

El proceso de adecuar el algoritmo de clasificación, elegido en la sección anterior, desde Matlab 7.0 a C, se puede considerar como la parte central de este trabajo, teniendo en cuenta que la mayoría de procesadores digitales de señal son compatibles con este último lenguaje. El objetivo principal de esta parte del proyecto es la obtención de un algoritmo de clasificación en C con resultados similares o iguales a los obtenidos en Matlab y teniendo en cuenta algunas características específicas en cuanto peso y desempeño. Vale la pena destacar que este proceso se realizó sin la ayuda de ninguna herramienta o programa que específicamente hiciera esta labor.

A continuación se explica paso a paso como se realizó el proceso de adecuación del algoritmo elegido en la sección anterior:

### **2.2.1. Preparación del Algoritmo Seleccionado**

Debido a que el archivo implementado por H. Romo en [3] incluía todos los algoritmos de clasificación de forma integral, además de ser muy flexible para su uso, fue necesario extraer y organizar en un solo archivo las funciones referentes al algoritmo seleccionado anteriormente (DT- DWT - MLP), con el cual se trabajaría a partir de ese momento. Además, fue necesario realizar algunas modificaciones en la estructura del mismo, estas modificaciones consistieron en omitir una de las características espectrales que se obtenían a partir de los coeficientes que se generaban al aplicar la transformada Wavelet discreta a la señal EMG. La característica eliminada fue los valores singulares de los coeficientes no nulos. Debido a este cambio, fue necesario modificar la etapa de clasificación (Perceptrón Multicapa) ya que su arquitectura debería ser diferente a la que originalmente tenía, principalmente en el número de entradas a la red, que cambió de 12 a 10. Es importante decir que estos cambios fueron realizados en el mismo entorno de Matlab. El código de este algoritmo se puede observar en el anexo A.

### **2.2.2. Creación de la Nueva Base de Datos de Vectores de Características**

Después de tener el archivo en Matlab del algoritmo de clasificación con las modificaciones pertinentes, se pasó a generar la nueva base de datos de patrones, tanto de entreno como de pruebas, la cual contiene los vectores de características con los cuales se hará la posterior clasificación. Ésta se realizó extrayendo las características de cada una de las señales EMG (Representada en la base de datos descrita en apartado 1.3), para cada uno de los seis pacientes. Como resultado de esta etapa se obtienen seis archivos con extensión (.txt), uno para cada paciente, los cuales contienen una matriz de 10 filas (Vectores de características) por 160 columnas (corresponden a las características tanto

temporales como espectrales previamente extraídas). Dicha matriz se muestra en la figura 17.

Longitud de Onda Señal 1	LO_ch1	LO_ch1	• • •	LO_ch1
Longitud de Onda Señal 2	LO_ch2	LO_ch2	• • •	LO_ch2
Valor Medio Absoluto Señal 1	VMA_ch1	VMA_ch1	• • •	VMA_ch1
Valor Medio Absoluto Señal 2	VMA_ch2	VMA_ch2	• • •	VMA_ch2
Promedio Absoluto de los Coeficientes Señal 1	AVG_ch1	AVG_ch1	• • •	AVG_ch1
Promedio Absoluto de los Coeficientes Señal 2	AVG_ch2	AVG_ch2	• • •	AVG_ch2
Varianza de los Coeficientes Señal 1	VAR_ch1	VAR_ch1	• • •	VAR_ch1
Varianza de los Coeficientes Señal 2	VAR_ch2	VAR_ch2	• • •	VAR_ch2
Energía de Escalas Señal 1	ENER_ch1	ENER_ch1	• • •	ENER_ch1
Energía de Escalas Señal 2	ENER_ch2	ENER_ch2	• • •	ENER_ch2
	1	2		160

Figura 17. Base de datos de los vectores de características para un paciente.

### 2.2.3. Creación y Entreno de las Redes Neuronales (MLP)

Después de haber generado las bases de datos tanto de entreno como de prueba para cada uno de los seis pacientes en el paso anterior, se procedió a crear y entrenar las redes neuronales que realizan la clasificación de los vectores de características.

El clasificador que se implementó fue un Perceptrón Multicapas, al cual se le realizaron algunos cambios en su arquitectura respecto al original encontrado en [3]. El primero de estos cambios es el ya mencionado anteriormente que consistió en reducir el número de entradas de 12 a 10, debido a la exclusión de una de las características (valores singulares de los coeficientes no nulos) en el vector de patrones por cada señal EMG procesada. El otro cambio consistió en el número

de neuronas de las 2 capas ocultas, ubicando 10 y 5 neuronas en la primera y segunda capa respectivamente, a diferencia de las 6 que tenía en cada una de las dos capas del algoritmo original. En la figura 18 se observa la arquitectura del Perceptrón Multicapas implementado.

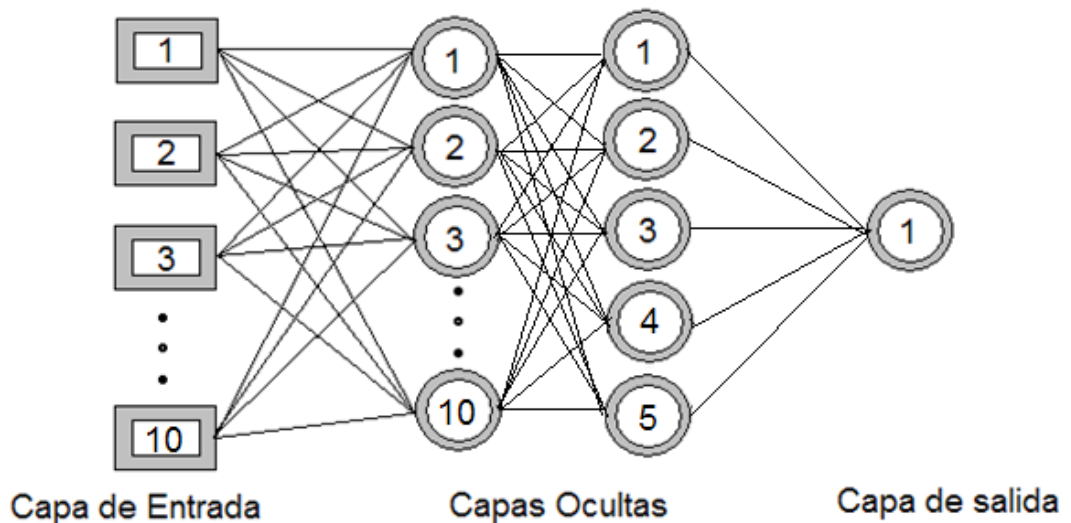


Figura 18. Arquitectura del clasificador a implementar (Perceptrón Multicapas)

La creación y el entreno de la red, se realizó en el entorno de Matlab 7.0 utilizando el toolbox de redes neuronales Neural Network Toolbox. A continuación se mencionan las principales funciones utilizadas de esta herramienta:

```
net = newff(pr,[10 5 1],{'tansig' 'tansig' 'purelin'});
```

*newff*: Esta función crea y devuelve, en *net*, una red MLP (Perceptrón Multicapas) con algoritmo de aprendizaje *BackPropagation*. A esta función se le deben pasar los argumentos: porque es una matriz con los valores máximos y mínimos entre todos los datos de entreno que se utilizarán en la entrada de la red; los valores 10, 5 y 1 que se encuentran entre llaves, son el número de neuronas que tendrá la primera, segunda y tercera (salida) capa de la red respectivamente. Las palabras *tansig* y *purelin* hacen referencia a las funciones de activación de cada una de las

capas de la red, en este caso las capas uno y dos tendrán una función de activación de tipo tangente hiperbólica sigmoïdal y la tercera capa (capa de salida) tendrá una función de activación lineal.

Es importante decir que esta función crea una estructura de datos a los cuales se puede tener fácil acceso. Esta característica es de suma importancia ya que permite conocer y utilizar variables de la red como por ejemplo las matrices de pesos, que son imprescindibles en el momento de implementar el clasificador en otro lenguaje de programación.

```
net = train(net,P_e,Tc);
```

Esta función es la encargada de entrenar la red previamente creada, en este caso *net*. Los parámetros que se le deben pasar son: *net*, que es la red que fue creada con la función *newff*; *P\_e* es una matriz que contiene los datos de entreno de la red, en este caso en particular es una matriz de 10x160 que contiene los vectores de características extraídas de las señales EMG. El otro parámetro que hace parte de esta función es *Tc* que es un vector de 160 posiciones que contiene las salidas ideales que debería tener la red, 1 en las primeras 40 posiciones, 2 en las siguientes 40 posiciones etc.

```
Y = sim(net,P_t);
```

Esta función es utilizada para simular una red neuronal de cualquier tipo. El parámetro *net* corresponde a la red que anteriormente fue creada y entrenada con las funciones *newff* y *train*. El otro parámetro es *P\_t*, este contiene la base de datos de los vectores de características que serán clasificados con la red. El resultado de la clasificación se guarda en el vector 'Y' de 160 posiciones que contiene valores de 1, 2, 3 ó 4 dependiendo de la intención de movimiento detectada por el sistema.



#### **2.2.4. Traducción del Algoritmo de Clasificación al Lenguaje C**

Después de tener, en Matlab 7.0, las etapas de extracción de características y clasificación del algoritmo seleccionado completamente adecuado, se pasó a realizar la traducción al lenguaje de programación en C. Como compilador se utilizó la herramienta *Dev C++* en la versión 4.9.9.2. Este proceso se realizó utilizando como principal ayuda el *Debugger* de Matlab, el cual sirve principalmente para localizar y corregir errores del programa que no han sido detectados por el compilador. Además, permite mantener el control del programa en tiempo de ejecución, siendo unas de sus principales características la de poder correr el programa paso a paso, deteniéndose en las sentencias más críticas, y la de permitir conocer el valor de todas las variables en cualquier instante de la ejecución. Esta última característica fue la más utilizada en esta parte del trabajo, debido a que la metodología implementada consistió en correr paso a paso el programa en Matlab, viendo cada función como una caja negra y de acuerdo a las señales tanto de entrada como de salida, definir una función existente o rutina en C que realizara exactamente la misma labor.

#### **2.2.5. Algoritmo de Clasificación en Lenguaje C**

Después de haber realizado todo el proceso de adecuación del algoritmo de clasificación, se obtuvo un programa en C que presentó un desempeño similar al conseguido en Matlab. A continuación se explica de forma resumida el programa obtenido en esta sección del proyecto.

La implementación de este algoritmo se encuentra en el archivo (C++Source File) llamado *CLASIFICACION\_EMG* de C. Éste se compone, principalmente, de 5 funciones las cuales son las encargadas de extraer las características de la señal EMG, clasificar el vector de patrones y determinar el tiempo de ejecución del

algoritmo. A continuación se hace una breve explicación de cada una de estas funciones.

- I. *long\_onda ()*: Esta función es una de las más sencillas de las cuatro que componen este archivo. En esta parte se recibe la señal EMG representada en un vector double de 256 posiciones, con el fin de extraer la longitud de onda de dicha señal. Esta función solo recibe una señal por vez, por lo tanto se debe recurrir a ésta en dos ocasiones.
- II. *valor\_medio\_abs ()*: Al igual que la función anterior, ésta recibe la señal EMG y extrae una característica de tiempo como lo es el valor medio absoluto de una señal por vez, haciéndose necesario el llamado de ésta una vez por cada canal a procesar.
- III. *caract\_wdtt ()*: Ésta se puede considerar como la función más importante y compleja de todas las implementadas en este archivo. En esta parte se aplica la transformada Wavelet discreta a la señal EMG con el fin de obtener una matriz con los coeficientes Wavelet, a partir los cuales se realiza la extracción de tres características espectrales como lo son: La energía de escalas, el valor absoluto y la varianza de dichos coeficientes.
- IV. *performancecounter\_diff ()*: Esta función no realiza ninguna labor relacionada con la extracción de características. Sin embargo, es la encargada de medir el tiempo de ejecución, en segundos, de las tres funciones anteriores. Éste tiempo es de suma importancia, ya que se debe tendrá en consideración, al igual que otras características de este algoritmo, en el momento de seleccionar un dispositivo (Circuito integrado) con características que lo hagan capaz de soportar dicho código.
- V. *Clasifica ()*: Este archivo realiza la clasificación del vector de características extraídas con las funciones anteriores. Esta clasificación es realizada con una red neuronal de tipo perceptrón multicapas. Ésta red fue previamente

entrenada con el toolbox de Matlab, el cual permitió copiar algunas variables de interés como lo son las matrices y vectores de pesos ( $w_1$  de  $10 \times 10$ ,  $w_2$  de  $5 \times 10$  y  $w_3$  de 5 posiciones) y los vectores de bias como se puede observar en el archivo. Esta red tiene como entrada el vector de patrones contiene 10 posiciones de tipo `double` y como salida una variable de tipo `float` ( $Y_3$ ) que corresponde al tipo del movimiento referente a la señal EMG procesada (1- movimiento 1, 2- movimiento 2, 3- movimiento 3, 4- movimiento 4,).

Este algoritmo tiene como salida la intención de movimiento del paciente, extraída desde la señal EMG procesada. Además, en la salida muestra la forma en que está compuesto el vector de patrones, imprimiendo las 10 posiciones que lo conforman.

#### **2.2.6. Desempeño del Algoritmo de Clasificación en Lenguaje C**

En esta sección era de suma importancia la obtención de un código en C que tuviera resultados relativamente iguales a los obtenidos en Matlab, así como también que dicho código tuviese unas características de desempeño que permitieran la utilización del algoritmo en un procesador de señal externo al *PC*. Las características principales tenidas en cuenta en esta parte del trabajo fueron el tiempo de ejecución, el espacio de memoria necesario para almacenar el código, el consumo de memoria RAM y el error de clasificación. A continuación se muestra la manera en que fueron evaluadas estas características en dicho algoritmo:

- **Tiempo de ejecución:** Esta característica se midió a partir la función de C++ `QueryPerformanceCounter()`. Esta función perteneciente a la librería *Windows.h* permite obtener una alta precisión al momento de querer medir el tiempo de ejecución de una función o rutina en particular.

Debido a que este tiempo depende, en gran medida, del equipo en el que se esté trabajando y de los procesos que se estén ejecutando en el mismo, se debe aclarar que este tiempo es bastante variable por lo que fue necesario realizar distintas medidas y con estas obtener un promedio.

Es substancial decir que el tiempo aquí medido hace referencia, específicamente, al utilizado para realizar la extracción de características y la clasificación del vector de patrones para una sola señal EMG.

- **Memoria para albergar el código:** Esta característica se pudo medir directamente desde el programa Dev C, ya que este permite, entre las propiedades del archivo realizado, observar la cantidad de memoria necesaria para almacenar las instrucciones que conforman el algoritmo.
- **Memoria RAM:** Ésta característica se puede considerar como una de las más críticas, ya que es la propiedad en la que la mayoría de dispositivos procesadores de señal presentan mayor déficit. Debido a esto, es de gran importancia el conocer una buena aproximación del consumo de memoria del algoritmo de clasificación, para así realizar una correcta selección del circuito integrado en el cual se embeberá dicho algoritmo.

Para la evaluación de este criterio, se recurrió al programa PROCESS EXPLORER versión 15.22. Ésta herramienta permite realizar un análisis detallado a cada uno de los procesos que se ejecutan en el PC. Con este programa se puede visualizar el consumo máximo de memoria RAM tanto virtual como física además de otros aspectos que no fueron tenidos en cuenta en el presente trabajo.

La cantidad de memoria RAM que aquí se muestra es aproximadamente la máxima utilizada por el programa para realizar tanto la extracción de características como la clasificación de patrones para una sola señal EMG.

- Error de clasificación:** Para la evaluación de este criterio se realizó la clasificación, para cada paciente, de los 160 vectores de características de la base de datos de prueba realizada previamente y se tomó el promedio de estos seis errores. Los resultados obtenidos se muestran a continuación en la tabla 8.

Tabla 8. Error de clasificación del algoritmo de clasificación en C

Paciente	Error de clasificación (%)
1	0
2	3.75
3	1.25
4	7.5
5	8.75
6	8.12
PROMEDIO	4.895

En la anterior tabla (Tabla 8) se pueden ver los errores de clasificación de los seis pacientes con el algoritmo de clasificación en C.

En la siguiente tabla (Tabla 9) se muestran los resultados obtenidos al realizar la medición de cada una de las 4 características del algoritmo de clasificación en C mencionadas arriba.

Tabla 9. Principales características del algoritmo de clasificación en C.

Tiempo de Ejecución	Memoria de Programa	Memoria RAM	Error de Clasificación
0.1654 milisegundos	21990 Bytes	24 Kilobytes	4.895 %

### **3. SELECCIÓN DEL CIRCUITO INTEGRADO**

En muchas ocasiones se observa, en diversos campos de la ingeniería, el requerimiento de dispositivos de procesamiento digital como DSP, FPGA, o microcontroladores para la construcción de sistemas embebidos con múltiples propósitos. Al interior del proceso de selección de un circuito integrado, generalmente se crea un margen de incertidumbre, ya que no se conoce con certeza la mejor alternativa. Como resultado los estudiantes que emprenden proyectos de esta índole deben dedicar un largo periodo de tiempo a familiarizarse con el dispositivo y con las herramientas necesarias para su manejo, además de buscar estrategias para superar dificultades como la poca información.

El objetivo principal de este capítulo consiste en la selección de un dispositivo integrado cuyas características lo hagan compatible con el algoritmo de clasificación seleccionado y descrito en el capítulo anterior. Para lograr este objetivo, se contó con 6 alternativas y un modelo de toma de decisiones basado en criterios de selección tales como: Espacio de memoria, facilidad de uso y programación, precio y eficiencia. A partir de esta metodología se realizó la selección del dispositivo más adecuado para la implementación del sistema.

#### **3.1. TOMA DE DECISIONES**

Al interior de un proceso de selección se ha de definir un modelo básico para la toma de decisiones, el cual consiste en una serie de pasos que traza una ruta para la consecución de la mejor elección. Este modelo radica en un proceso de cinco pasos, los cuales se desarrollan a continuación.

### **3.1.1. Problema**

El problema general radica en seleccionar un circuito integrado (CI) que posea las mejores características técnicas de funcionalidad, para obtener el mejor desempeño, la mayor eficiencia al clasificar y el menor tiempo de ejecución, al implementar el algoritmo de clasificación previamente elegido.

### **3.1.2. Búsqueda de Alternativas**

La búsqueda de alternativas comprende la recolección de información de los diferentes circuitos integrados que cumplan las características básicas de velocidad, espacio de memoria y comunicación, que permitan el correcto funcionamiento del sistema. Como primer paso se procedió a realizar una búsqueda detallada de los principales fabricantes en el segmento de circuitos integrados. Los fabricantes seleccionados deben poseer gran respaldo, excelentes redes de distribución en nuestro país y basta literatura sobre información concerniente a sus dispositivos. Dentro de las marcas líderes analizadas se encuentran.

#### **3.1.2.1. Microchip**

Microchip es una empresa Norteamericana dedicada a la fabricación de diferentes componentes electrónicos como microcontroladores, memorias, etc. Dentro de la familia de los microcontroladores encontramos gran variedad de dispositivos, los cuales difieren principalmente en su arquitectura en bits y sus especificaciones (la cual los ubica al interior de una respectiva gama).

#### **I. Arquitectura de 8-bits**

- **Gama Base:** Está compuesta por microcontroladores que disponen de 33 instrucciones de 12 bits de longitud cada una. Esta gama contiene 14



modelos diferentes que se destacan por su bajo costo y pocos pines, generalmente 6.

- **Gama Media:** En esta familia se ubican los dispositivos con prestaciones intermedias. Disponen de 35 instrucciones de 14 bits de longitud cada una, mayor cantidad de pines, de entre 8 hasta 64 y pilas de 8 niveles. En esta gama se exponen 71 modelos diferentes.
- **Gama mejorada:** Es la familia de más altas prestaciones entre los dispositivos de 8 bits, micros con una cantidad de pines de entre 64 y 80, con repertorio de 77 instrucciones de 16 bits de longitud cada una, pilas de 16 niveles y periféricos considerables. Esta gama es ideal para proyectos de con dificultad media – alta.

## II. **Arquitectura de 16-bits.**

- **Gama MCU:** Son microcontroladores de alto rendimiento pero de bajo costo. Esta gama ofrece dispositivos con velocidades de procesamiento de 16 Millones de Instrucciones Por Segundo (MIPS) hasta 40 MIPS, se pueden encontrar con memorias flash de programa de hasta 256kBytes, en encapsulados de entre 64 hasta 100 pines, ideales para proyectos de alto rendimiento.
- **Gama DSC:** Son dispositivos especializados para el procesamiento digital de señales, se subdividen en dos familias los DSPIC30f y DSPIC33f. Los DSPIC30 cuya aplicación es necesaria para proyectos de sistemas embebidos en tiempo real, con una alimentación de 5V y velocidad de procesamiento de 30MIPS. Los DSPIC33F que se distinguen de la anterior subfamilia por manejar una tensión de alimentación más baja de 3.3v y una velocidad de procesado de 40MIPS además disponen de memoria de programa flash de 256Kb.

Microchip deja a disposición del usuario un entorno de desarrollo integrado *IDE (Integrated Development Environment)* conocido con el nombre de MPLAB, el

cual es totalmente gratuito. Este *IDE* permite la selección de los distintos CI, realización de código además de su grabación física y depuración.

Para programación de los diferentes CI, microchip ofrece de forma gratuita el compilador de lenguaje ensamblador (ASM) el cuál se encuentra incluido en MPLAB, pero también cuenta con compiladores de lenguaje C como C18 para los PIC18, C30 para DSPIC30 - 33 y C32 para el PIC32 los cuales se ofrecen con versiones estudiantil gratuita.

### **3.1.2.2. Texas Instruments**

Empresa norteamericana conocida popularmente por sus siglas TI, es una empresa líder en fabricación de circuitos electrónicos, entre sus productos destacados se encuentran los DSP y microcontroladores 16-bits y 32-bits.

#### **I. Arquitectura de 16-bits:**

Estos microcontroladores destacan por ser de bajo consumo, con arquitectura de juego de instrucciones reducidas *RISC* (*Reduced Instruction Set Computer*), esta familia se denomina con *MSP430* dentro de la cual se destacan varias subcategorías.

- **MSP430Fxxx:** Son microcontroladores de propósito general, dispositivos que se destacan por el bajo consumo, basados en memoria flash, con velocidades de procesamiento bastante alta , cantidad de periféricos considerables para proyectos con algún nivel de dificultad y puertos de comunicación UART, segmentos LCD, etc.

#### **II. Arquitectura de 32-bits.**

Dentro de la arquitectura de 32-bits se encuentran los dispositivos de procesamiento de señal DSP. TI es líder en este segmento con los siguientes dispositivos:

- **TMS320F283xx:** Controladores en punto flotante.
- **TMS320F281x:** Controladores capaces de trabajar a 150MIPS de procesado.
- **TMS320F280xx:** Controladores capaces de trabajar a 100MIPS.

Texas Instruments, dispone para el usuario un entorno de desarrollo integrado conocido como Code Composer Studio, una herramienta que incluye los compiladores de cada una de las familias de dispositivos de TI, editor de código fuente, depurador, simuladores en tiempo real del sistema operativo entre otras características.

### **3.1.2.3. Freescale**

Empresa norteamericana dedicada a la fabricación de semiconductores, en una empresa relativamente nueva, que genera CI como microcontroladores, DSP, memorias entre otros.

Freescale distingue entre los microcontroladores de propósito general y de propósito específico, dentro de los cuales podremos encontrar las familias: HCS08, RS08 y HC08. Dispositivos que incorporan una variedad importante de periféricos indispensables para cubrir cualquier tipo de necesidad.

Freescale pone a disposición del usuario algunas alternativas, entre ellas se destaca Code Warrior el compilador por excelencia de Freescale y SDCC, un desarrollo Open Source muy interesante y funcional que permite compilar código escrito en lenguaje C para múltiples plataformas de microcontroladores.

### **3.1.3. Selección de Alternativas**

Tras el análisis se procede a una segunda instancia, en donde se definen los diferentes modelos de CI que posean las características mínimas, para que su adaptación a los algoritmos de clasificación sea apropiada. Las características que

se tuvieron en cuenta para la selección consistieron básicamente en la cantidad de memoria de programa que fue precisada con mínimo de 21990 Bytes y memoria RAM definida en 24Kilobytes. Los dispositivos en cada una de las marcas analizadas anteriormente, que cumplieron con los requerimientos básicos, pero que al mismo tiempo no sobrepasaron abultadamente las características, se especifican en las tablas 10, 11, 12.

Tabla 10. Características principales de los circuitos integrados de Microchip.

Dispositivos de Microchip	Pic24ep256gu810	DSPIC33fj128mc802	DSPIC33fj256gp710a
Arquitectura (Bits)	16	16	16
Velocidad CPU (MIPS)	50	40	40
Memoria de programa (KB)	256	128	256
RAM (B)	28672	16384	30720
Rango de temperatura	-40 to 125	-40 to 150	-40 to 150
Rango de voltaje	3V – 3.6V	3V – 3.6V	3V – 3.6V
I/O pins	83	21	85
Oscilador interno	7.37 MHz, 32 kHz	7.37 MHz, 512 kHz	7.37 MHz, 512 kHz
Comunicación digital	4 -UART 4 -SPI 2 -I2C	2 -UART 2 -SPI 1 -I2C	2 -UART 2 -SPI 2 -I2C
Timers	0 x 8-bit 9 x 16-bit 4 x 32-bit	0 x 8-bit 5 x 16-bit 2 x 32-bit	0 x 8-bit 9 x 16-bit 4 x 32-bit

Tabla 11. Características principales de los circuitos integrados de Texas Instruments

Dispositivos de TI	Msp430f5333	Msp430f5632
Arquitectura (Bits)	16	16
Velocidad CPU (MIPS)	50	60
Frecuencia Mhz	20	20
Memoria de programa (KB)	128	256
RAM (B)	10240	16638
I/O	74	74
Voltaje de operación (V)	1.8 - 3.6	1.8 - 3.6
Temperatura de operación (C)	-40°C to 85°C	-40°C to 85°C
Canales UART/IrDA/SPI	2	2
Timers-16 bit	4	4
Watchdog	SI	SI
SVS	SI	SI
DMA	2	3
Comparadores	SI	SI
ADC	12-bit	12-bit
ADC Channels	16	16

Tabla 12. Características principales de los circuitos integrados de Freescale.

Dispositivos de Freescale	Mc56f84462vlhl	Dsp56f827

Velocidad CPU (MIPS)	60	40
Frecuencia de operación (MHz)	60	80
Flash interna (KB)	128	128
RAM (KB)	24	21
Módulos ADC independientes	2	2
Tiempo de conversión ADC (ms)	30	25
Módulos independientes PWM	1	1
Resolución ADC	12 Bits	12 Bits
Comparador análogo	3	
Interfaz serial	QSCI QSPI I2C CAN	QSPI QSSI QSCI
I/O pines	39	
Voltaje de funcionamiento (V)	3.0V - 3.6V	2.5 V - 3.3V
Temperatura de operación (C)	-40 - 105	-40 -110

En la tabla 13 se observan los diferentes dispositivos analizados para cada una de las marcas seleccionadas, además en una tercera columna se realiza una notación o etiqueta la cual se utiliza para reconocer los dispositivos seleccionados.

Tabla 13. Resumen de las alternativas seleccionadas

Fabricante	Dispositivos	Etiqueta
Microchip	PIC24ep256gu810	Op1
	DSPIC30fj128mc802	Op2
	DSPIC33fj256gp710a	Op3
Texas Instruments	Msp430f5333	Op4
	Msp430f5632	Op5
Freescale	Mc56f84462vlh	Op6
	DSP56f827	Op7

### 3.1.4. Criterios de Evaluación

Todos los sistemas de computación poseen limitaciones en sus métricas de diseño en determinadas aplicaciones, éstas a su vez juegan un papel fundamental en los sistemas embebidos. Una métrica de diseño es la medida de algunas características de implementación, como: tamaño, costo, espacio de memoria, desempeño, confiabilidad y consumo de energía. Los sistemas embebidos generalmente deben ser poco costosos, poseer un tamaño reducido, tener un buen desempeño para procesar datos en tiempo real, ser sumamente confiables y además consumir un mínimo de energía para extender el tiempo de uso en caso de utilizar baterías.

Al interior de este proyecto se plantean 4 características fundamentales a tener en cuenta en el proceso de selección de un dispositivo que se adecue lo mejor posible a los requerimientos necesarios para el eficiente funcionamiento del sistema. Estas características son descritas a continuación.

- **Espacio de Memoria:** El espacio de memoria es fundamental para alojar las instrucciones de funcionamiento y los datos que el sistema utiliza y genera durante el proceso de ejecución. De acuerdo a la versión preliminar del pseudo-código clasificador, se calcula la cantidad necesaria de memoria que el CI debe poseer como mínimo para que su funcionamiento sea óptimo. Conociendo este parámetro se califican los dispositivos en las diferentes marcas que cumplan apropiadamente con este parámetro.
- **Facilidad de Uso y Programación:** La facilidad de uso y programación consiste en la calidad y la claridad de información que se pueda llegar a recolectar respecto de los dispositivos, herramientas y lenguajes de programación. Esta información es recolectada por medio de foros y videos en sitios web así como también de libros y revistas. Es importante tener en cuenta que las herramientas software de programación, necesarias para poder llevar a cabo el desarrollo del código y herramientas hardware para realizar la programación física CI deben ser de fácil acceso.
- **Precio:** El precio del CI es un factor importante en la toma de decisión final, lo que se busca del CI es que cumpla con los parámetros de espacio de memoria, eficiencia, comunicación y que además presente un costo reducido. Además se debe tener en cuenta en lo posible no generar ningún costo secundario, como programadoras, dispositivos electrónicos extras, soquets especiales para su manipulación y software que estén limitados por licencias de difícil o costosa adquisición.
- **Eficiencia:** Este trabajo se enmarca en la línea de I+D de control y robótica, formando parte del proyecto macro *Prototipo Funcional de Prótesis de Mano Robótica*, en donde el objetivo principal es la construcción de una prótesis robótica que pueda ser manipulada por señales EMGS obtenidas directamente del brazo del paciente, por lo cual la eficiencia del sistema embebido debe cumplir acorde con los requerimientos para poder ser



utilizado en conjunto a las fases próximas y anteriores de la etapa de clasificación en el proyecto macro. Considerando estos parámetros el CI debe tener la eficiencia para poder ser utilizado en el proyecto macro.

### 3.1.5. Asignación de Pesos

Tras definir los criterios de evaluación, es necesario ponderarlos de tal forma que se les asigne un peso apropiado para que la selección sea objetiva, por lo cual se realiza una división de los criterios en dos grupos: indispensables y deseables. Los indispensables tendrán un mayor peso al momento de la asignación, ya que representan criterios no negociables, al contrario los deseables son criterios que no son necesariamente un requerimiento obligatorio para el sistema embebido pero que puede brindar facilidades en etapas de desarrollo. De esta forma se definen “*espacio de memoria*” y la “*eficiencia*” como criterios indispensables, y “*facilidad de uso y programación*”, y “*costos*” como deseables. Los criterios serán ponderados en una escala del 1 a 4, en donde 4 representa una calificación de gran peso en la selección del CI y 1 representa una ponderación poco significativa (tabla 14).

Tabla 14. Ponderación de criterios.

Criterio	Ponderación
Espacio de memoria	4
Facilidad de uso y programación	2
Costos	1
Eficiencia	3

### 3.1.6. Análisis de la Matriz de Decisión

Para seleccionar una alternativa de la manera más objetiva, es preciso evaluar cada una de las elecciones según criterios que expresen juicios favorables o desfavorables en cada caso, de esta manera se evalúan las posibilidades y se califican cada uno de los criterios ponderados para llegar a una solución que tenga un alto grado de *asertividad*. La calificación se realizara de 1 a 10, en donde 10 representa que el CI cumple con los requerimientos mínimos exigidos por el algoritmo o que representa la mejor opción entre las alternativas analizadas, y 1 como una calificación baja, que expresa el poco o nulo cumplimiento de los requerimientos. En la tabla 15 se realiza la matriz de decisión, esta tabla se articula de la siguiente manera. En la primera columna se definen los criterios de evaluación anteriormente mencionados, en la segunda columna se expresa la ponderación correspondiente a cada criterio, en las siguientes siete columnas se expresan las alternativas seleccionadas y se realiza la calificación. Esta calificación se realiza en el extremo superior derecho (numero en rojo) y se multiplica por la ponderación, como resultado se obtiene un valor o una calificación ponderada (Numero negro). La ponderación final será la sumatoria de los criterios ponderados para cada una de las alternativas. El total con la mayor puntuación en su clasificación será la alternativa seleccionada. En la Tabla 15 se evidencia la matriz de decisión con la calificación asignada a cada alternativa.

Tabla 15. Matriz de decisión

		Alternativas						
Criterio	Pond.	Op1	Op2	Op3	Op4	Op5	Op6	Op7
Espacio de memoria		9	4	10	5	7	8	6
	4	36	16	40	20	28	32	24
Eficiencia		9	8	8	9	10	10	8
	3	27	24	24	27	30	30	24
Facilidad de uso y programación		9	9	9	7	7	5	5
	2	18	18	18	14	14	10	10
Costos		4	7	6	9	7	9	6
	1	4	7	6	9	7	9	6
<b>Total</b>		85	65	88	70	79	81	64

### 3.2. ANÁLISIS DEL DISPOSITIVO SELECCIONADO

A partir de la Matriz de decisión se logra evidenciar que la alternativa etiquetada con Op3, la cual corresponde al CI (*DSPIC33fj256gp710a*), es la mejor opción, ya que cumple con los requerimientos básicos de espacio de memoria, eficiencia,

facilidad de uso y programación, y costos acordes para el desarrollo del sistema clasificador. Además es importante destacar que en su mayoría las herramientas disponibles en la Universidad de Cauca son afines con el fabricante MICROCHIP, por lo cual existe más experiencia en el manejo del software para desarrollo de código y de herramientas de programación física.

## **4. DISEÑO Y CONSTRUCCIÓN DEL SISTEMA**

El objetivo principal de esta parte del proyecto consistió en realizar la construcción de un sistema embebido que permitiera la implementación del algoritmo de clasificación previamente elegido y adecuado, para obtener el mejor desempeño en cuanto a tamaño, eficiencia al clasificar y tiempo de ejecución.

Para la elaboración del dispositivo se determinó dividirlo en tres etapas con el fin de facilitar el proceso de elaboración del mismo. La primera etapa consistió en realizar una documentación de los diferentes dispositivos electrónicos utilizados en la construcción del sistema clasificador. En la segunda etapa se efectuó un diseño electrónico previo del sistema clasificador en su totalidad, este diseño fue realizado con ayuda de herramientas software especializadas. Posteriormente se diseñó la interfaz de usuario, la cual permite una fácil manipulación de la base de datos y una comunicación sencilla con el SE. Por último, se ejecuta la construcción del sistema, integrando todos los dispositivos especificados, obteniendo así un primer prototipo del sistema clasificador de señales electromiográficas. A continuación se describen cada una de las etapas mencionadas.

### **4.1. DOCUMENTACIÓN**

#### **4.1.1. Circuito integrado**

El circuito integrado escogido para realizar las diferentes operaciones lógicas, es el procesado de señal (*DSPIC33fj256gp710a*) en cual se encuentra integrado sobre el módulo *MA330012*. Este módulo integra tanto el circuito integrado de microchip como también 5 capacitores en su configuración adecuada y un socket para el acceso a cada uno de los 84 pines, además 17 pines que se encuentran

en la superficie del módulo, lo que permite una sencilla y rápida conexión. Este módulo se observa en la figura 19.



Figura 19. Módulo MA330012

#### 4.1.2. Regulador de Voltaje

Para normalizar el voltaje de la entrada del *DSPIC33fj256gp710a*, se eligió el dispositivo *LM1117T* de *National Semiconductor* (Figura 20). Este es un regulador de voltaje de baja potencia, capaz de proporcionar hasta 800 mA de corriente de salida, con un voltaje de referencia ajustable de 1,25 V. Para el desarrollo del sistema embebido es necesario escoger como voltaje de salida 3.3 voltios con el cual se alimentara el *DSPIC33fj256gp710a*, los dispositivos restantes trabajan a 5 voltios.

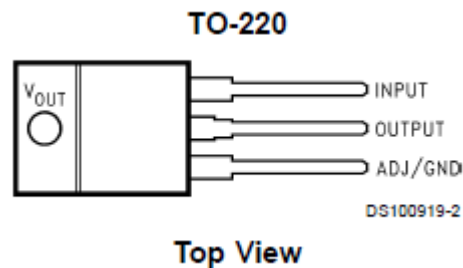


Figura 20. Dispositivo LM1117T de National Semiconductor

### 4.1.3. Adaptador de Nivel TTL a RS-232

Muchos microcontroladores poseen una interfaz UART para comunicación serial asincrónica, tipo RS-232, que en un PC se denomina puerto “COM”. Si bien los microcontroladores poseen hardware para generar la secuencia de bits en los tiempos correctos, no son capaces de generar el voltaje especificado por el estándar RS-232, por lo cual requieren de un chip externo que haga esta conversión de voltajes.

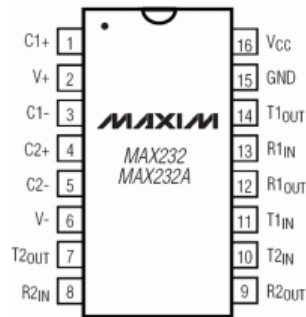


Figura 21. Dispositivo Max-232 de Maxim electronics

Si bien el circuito de conversión puede armarse fácilmente en un protoboard, disponer del circuito en un módulo tiene las siguientes ventajas:

- Ahorro de espacio en el protoboard.
- Ahorro de tiempo en el armado y testeo del circuito.
- Evita conexiones erróneas del chip conversor.
- Pueden incluirse LEDs monitores.

Para realizar la adaptación de niveles de voltaje TTL al estándar RS-232 se eligió el dispositivo Max-232 de *Maxim electronics*, el cual dispone internamente de 4 conversores de niveles TTL al estándar RS-232 y viceversa, para ser usados en la

comunicación serie con ordenadores en los puertos COM. Figura 21. Las especificaciones técnicas de este dispositivo se observan en la tabla 16.

Tabla 16. Especificaciones técnicas del dispositivo Max-232 de Maxim electronics.

<b>Nombre del parámetro</b>	<b>Valor</b>
Vcc	4,5v a 5,5v.
Consumo	4 mA (15 mA con carga a la salida de 3 Kohm).
Entradas compatibles	TTL y CMOS.
Tensión de entrada máxima RS-232	+/- 30v
Tensión de Salida RS-232	+/- 15v.
Resistencia entrada RS-232	5 Kohm (a masa).
Resistencia entrada TTL/CMOS	400 Kohm (a positivo).
Salida RS-232	+/- 22 mA. Salida
TTL/CMOS	A masa -10 mA, a positivo +30 mA.
Data Rate	200 Kbps (mín 116 Kbps).

#### 4.1.4. Componentes adicionales

Para el desarrollo final de la tarjeta se empleó, db9 hembra, 6 led, switch, resistores y 4 capacitores de 10  $\mu$ F. De igual manera se usaron de salida y entrada que facilitan el acople con otros dispositivos.



## 4.2. DISEÑO

En este apartado se realiza una descripción de las herramientas usadas para el diseño tanto de las pistas de interconexión eléctrica del sistema clasificador como de la interfaz de usuario, seguidamente se pautan los procedimientos de diseño realizados para la creación del PCB (*Printed Circuit Board*) y de la interfaz del usuario.

### 4.2.1. Herramientas de Diseño

- **Eagle V 6.1:** EAGLE (Easily Zpplicable Graphical Layout Editor) es un programa utilizado para el diseño de diagramas electrónicos de fácil uso, con componentes que pueden ser fácilmente etiquetados, además posee un editor de PCB, con funciones bastante eficientes como las de autoenrutador. Eagle incluye una librería gratuita al momento de la instalación, la cual contiene componentes pasivos, activos, conectores comúnmente usados y otras librerías en la red que contienen sin fin de dispositivos. La versión Light del programa está disponible en forma gratuita pero presenta limitaciones como el tamaño de los PCB y la cantidad de capas del circuito.
- **Proteus:** Este software es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics. Este software se encuentra compuesto por diferentes subprogramas tales como: Isis, Ares, y módulos VSM y Electra.

El Programa ISIS, Intelligent Schematic Input System (Sistema de Enrutado de Esquemas Inteligente) permite el diseño de multitud de planos eléctricos, ya que incorpora bases de datos con variados componentes de diseño electrónico, que varían desde simples resistencias hasta complejos microcontroladores y procesadores de señal. Las ventajas de añadir este

software son numerosas, ya que además permite la simulación los circuitos electrónicos en tiempo real, incorporando características electrónicas de las diferentes familias de microcontroladores.

- **Matlab:** MATrix LABoratory es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible en múltiples plataformas.

Entre sus prestaciones básicas se halla la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes) y las de Simulink con los paquetes de bloques (blocksets).

#### 4.2.2. Procedimientos de Diseño

- **Simulación:** Este procedimiento fue realizado con ayuda de la herramienta software de ISIS de Proteus, la cual permitió simular el funcionamiento del sistema de comunicación, extracción de características y lógica de clasificación en tiempo real. Debido a que este software no disponía de librerías para la simulación del CI (DSPic33fj256gp710a) se eligió realizar la simulación con el CI (DSPIC30f) que aunque no presenta las mismas características técnicas, si permitió darnos a conocer el funcionamiento del código.
- **Diseño de Diagrama y PCB:** Para realizar las pruebas es necesaria la construcción de un sistema físico sobre el cual corroborar el funcionamiento

del código y de las características básicas que agrupa la comunicación serial, la programación del dispositivo y manejo de entradas y salidas. Para el cumplimiento del objetivo se efectúa el diseño de una PCB del sistema clasificador de movimiento en el software Eagle V6.1, el diseño está constituido por las siguientes instancias: la primera instancia consta de la selección de los diversos dispositivos electrónicos usados en la creación del sistema, para esta selección es importante verificar que dichos dispositivos se encuentre dentro de las librerías instaladas. En la segunda instancia se realiza un diagrama en donde se especifican las conexiones entre los dispositivos seleccionados, etiquetando los terminales hacia nodos afines, figura 23. Tras una correcta conexión se ejecuta la función “board”, la cual permite el diseño del PCB, en esta instancia se seleccionan las medidas reales con las cuales va a contar el sistema y se ubican los dispositivos. Para realizar el enrutado de las pistas el programa tiene dos opciones: la primera es dejar que el software realice el enrutador automáticamente y la segunda realizar el enrutado manualmente. En este proyecto se realizó un enrutado manual, ya que presenta mejores terminados y de esta forma la construcción es más sencilla. En la figura 22 se puede apreciar el diseño de la PCB del sistema clasificador.

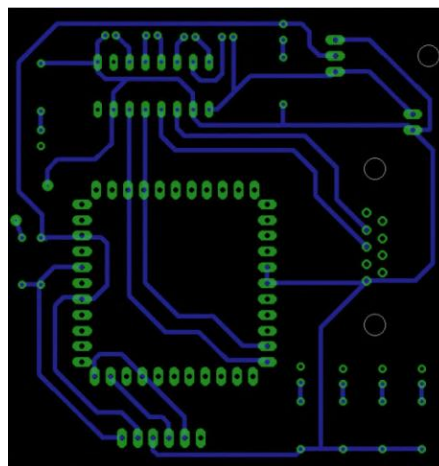


Figura 22. Diseño PCB Eagle 6.1

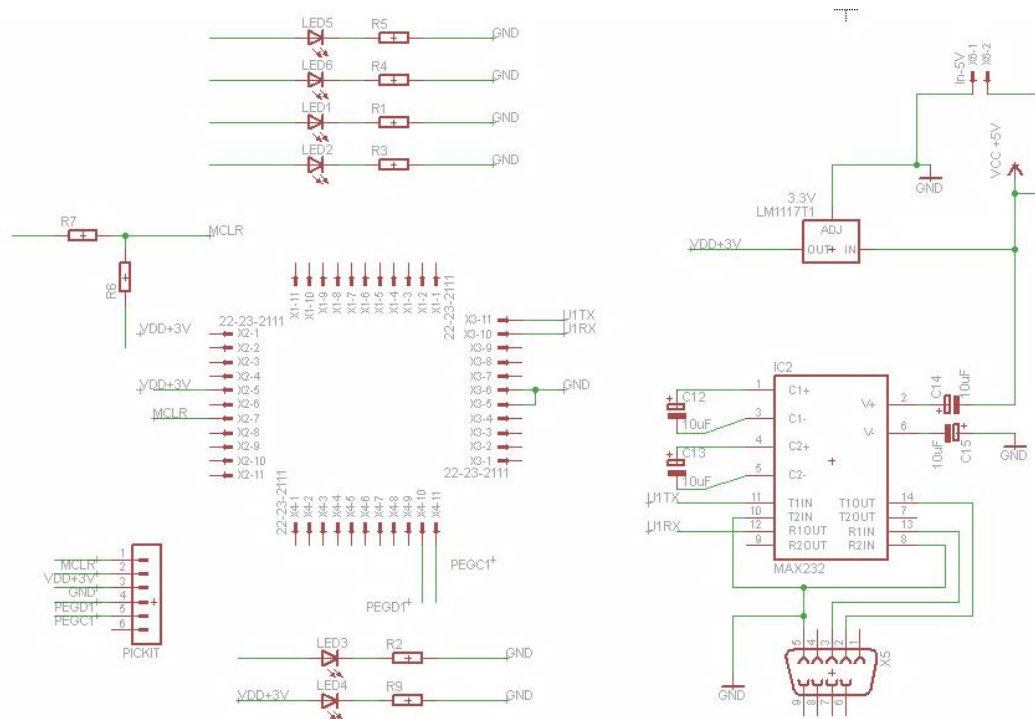


Figura 23. Diagrama de conexiones Eagle 6.1.

- Desarrollo del Código para el Envío de Datos:** Ya conociendo la base de datos y el acceso a cada uno de sus subniveles de información (la base de datos se especifica en el capítulo 1, sección 1.3), se inicia el desarrollo de un código que permita la comunicación entre la base de datos contenida en un PC y el sistema clasificador. Para este fin es fundamental el desarrollo de un programa que facilite la comunicación. El desarrollo del código con las instrucciones de comunicación se realizó en el software de Matlab, eligiendo como protocolo de comunicación el RS-232. Inicialmente se programan las instrucciones para realizar la carga de los datos al software de Matlab, de esta forma se establecen comandos para la carga de los datos de cada uno de los 6 pacientes, en esta instrucción se registra la ubicación en disco de los datos (línea 1). A continuación se realiza la selección de los canales. Ya que

el sistema trabaja con dos canales, se seleccionan los dos canales con el mayor contenido de información para cada uno de los pacientes (línea 2), y como cada canal aporta un vector de información de 256 datos es necesario unificarlo o concatenarlo, por esta razón se conforma el vector “*senal*”, vector constituido por 512 datos (línea 3).

```
1 load (['DB 4 MOV entreno\Registros_E_EMG\' 'pac6_4mov.mat']);
2 ch1 = ch8_4mov; ch2 = ch12_4mov;
3 senal=[ch1(:,61)' ch2(:,61)'];
```

Posteriormente se realiza la configuración de los diferentes aspectos en una comunicación serial, primero se designa el puerto a utilizar, en este caso se asocia “*s*” al puerto “COM1” (línea 4). Con la función “*set*” se modifican los parámetros de la comunicación, asignando una velocidad de trasmisión de 9600 baudios (línea 5), y por último se inicializa el puerto con la función “*fopen*”. Como cada uno de los 512 datos del vector “*senal*” se cambian de un valor double de 64 bits a string con formato exponencial *x.xxxxxe-xxx* (*una base decimal exacta con formato de 5 decimas y un exponente entero negativo*), se realiza el envío digito a digito, anteponiendo un signo “+” a los datos que sean mayores a cero “*dato > 0*”. Para finalizar se cierra el puerto utilizado con la función “*fclose*” y se realiza un borrado de la memoria con la función “*delete*”.

```
4 s=serial('COM1');
5 set(s,'BaudRate',9600);
6 fopen(s);
```

### **4.2.3. Diseño Interfaz de Usuario**

El dispositivo clasificador debe estar en constante comunicación con la base de datos contenida en el PC, por lo cual el uso de código de envío será reiterado, este código además presenta secciones que deben ser constantemente modificadas para su uso, generando dificultades de fluidez al momento de realizar las pruebas de validación del dispositivo. Por estas razones es fundamental el desarrollo de una interfaz de usuario que realice el envío de los vectores contenidos en la base de datos de una forma ordenada, eliminando problemas de reiteración en la modificación del código.

La interfaz fue desarrollada con la ayuda del GUIDE (Editor de Interfaces de Usuario) de Matlab y presenta las siguientes etapas: Inicialmente la interfaz de usuario carga los datos de un determinado paciente, seguidamente se seleccionan los dos diferentes canales y el movimiento, como último paso se envían los datos. La interfaz de usuario se comunica con el dispositivo a través del puerto serial con protocolo de comunicación RS-232, por medio de este puerto, se realiza el envío de un vector de 512 datos, cada dato es recibido, organizado y utilizado para la clasificación de la señal. El diagrama de flujo de esta interfaz se expone en la figura 24. La interfaz de usuario se encuentra como un archivo ejecutable (.exe) desde computadores que no tengan instalado previamente MATLAB. La interfaz de usuario se observa en la figura 25

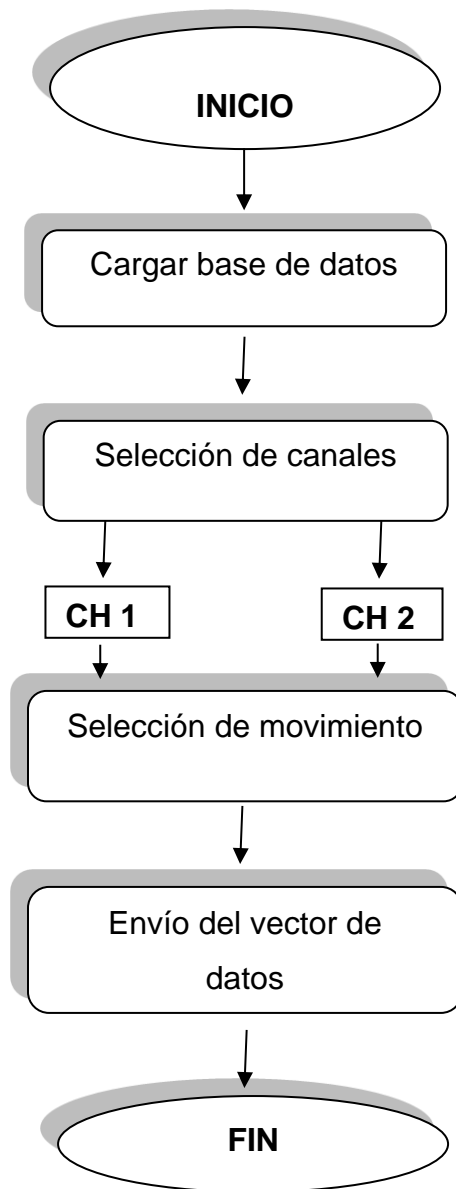


Figura 24. Diagrama de flujo de la interfaz de usuario.

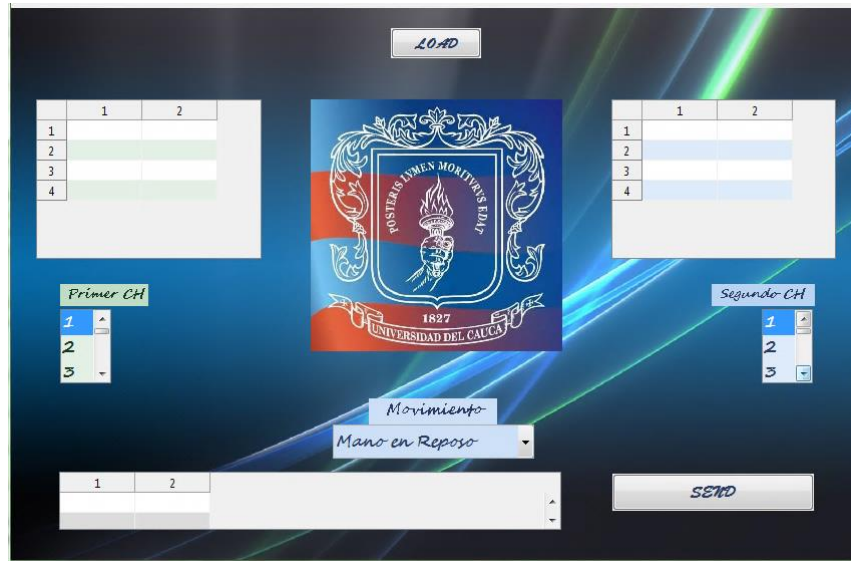


Figura 25. Interfaz de usuario

### 4.3. CONSTRUCCIÓN

Para la construcción del dispositivo clasificador de señales EMGS fueron utilizados diferentes materiales, entre los cuales se encuentran: una baquelita con medidas aproximadas de 9cm x 10cm y peso aproximado de 40gramos, un socket para la conexión del módulo MA330012 con *DSPIC33fj256gp710a* de microchip, un adaptador para niveles de voltaje de comunicación Max-232, un conector serial db9 hembra, un regulador de voltaje LM1117T, 6 diodos emisores de luz led, resistores y capacitores. (Figura 27). La construcción del dispositivo se realizó de forma casera, con herramientas de electrónica disponibles y fáciles de conseguir. A continuación se describen los pasos realizados para su construcción.

- Se imprima el PCB sobre papel propalcote, se debe tener en cuenta que la imprimación debe ser realizada en impresora laser o cualquier dispositivo de impresión que utilice tóner.



- Se ejecuta una limpieza sobre la baquelita de cobre. Primero se pule la superficie con una esponja de brillo con movimientos circulares. Segundo se eliminan residuos grasos de la superficie con enjuagues de alcohol.
- Se alinean los bordes y se asegura el papel impreso sobre la baquelita, de tal manera que el impreso del PCB quede de cara a la superficie de cobre de la baquelita,
- Con un dispositivo que genere calor generalmente una plancha, se procede a fundir el tóner y traspasarlo de la hoja de papel a la cara de cobre de la baquelita.
- A continuación se sumergió la baquelita en un recipiente de plástico o de vidrio, y se agregó la dosis suficiente de cloruro férrico concentrado como para cubrirla.
- Ya con la baquelita terminada se procedió a realizar las perforaciones en donde se incrustara la electrónica que componen el SE. Para esta tarea se emplean brocas con diámetro de 1mm.
- El paso siguiente fue ubicar los componentes. Por regla general y para tener un poco más de orden, conviene comenzar por los más pequeños y de menor altura. Como último paso del armado se realizan las soldaduras de los dispositivos electrónicos.
- Ya con SE completamente armado se realizan pruebas para verificar su correcto funcionamiento, las pruebas realizadas consistieron en verificar niveles de voltaje en algunos terminales y pruebas de continuidad en diferentes nodos.
- Concluida la fase de construcción se logra un dispositivo embebido capaz de recibir y clasificar una señal EMG, este sistema embebido cuenta con un led indicador de voltaje "Power ON", un led indicador que permite conocer cuando el SE está listo para recibir una nueva señal, un switch interruptor y 4 leds para verificar la clasificación de movimiento. El funcionamiento del sistema embebido consiste en recibir la señal por medio del puerto serie

(protocolo RS-232), esta señal es convertida a señal compatible con niveles TTL del circuito integrado, por medio del circuito Max-232. La señal recibida contiene dos vectores de datos de 256 posiciones cada uno, con datos en formato exponencial conteniendo la información de un movimiento para un paciente. Esta información es recibida y procesada por el módulo MA330012, el cual regido por el algoritmo, ordena y clasifica la señal. Para confirmar la clasificación de movimientos se situaron 4 leds, uno por cada movimiento, de esta forma el led que encienda expresará el resultado final de la clasificación.



Figura 26. Dispositivo clasificador de intención de movimientos.

## 5. PRUEBAS Y VALIDACIÓN

En este capítulo se explican y desarrollan las diferentes pruebas que comprenden la verificación dinámica del comportamiento del clasificador de intención de movimiento a partir de señales EMGS. En primer lugar se realiza una serie de pruebas de integración que comprenden el funcionamiento de los diferentes componentes del sistema por separado y luego en integración uno a uno. Ya con el sistema completo y en perfecto funcionamiento se efectúan pruebas de funcionalidad con las cuales se registra la precisión del SE al clasificar las señales de cada uno de los pacientes, al final se validan los resultados obtenidos.

### 5.1. PRUEBAS DE INTEGRACIÓN

La prueba de integración busca probar la combinación de los distintos componentes del sistema para determinar su correcto funcionamiento en conjunto. En esta fase se determinaron 3 componentes, el sistema hardware, la interfaz gráfica y el código clasificador. Primero se combinó la interfaz gráfica con el sistema hardware y se puso a prueba la comunicación serial, en la segunda prueba se combinó el sistema hardware con el código de clasificación y se verificó su funcionamiento y tiempo de ejecución.

- **Sistema hardware - Interfaz grafica:** Esta prueba consiste en verificar el funcionamiento de la comunicación entre el computador en donde se encuentra la base de datos y el sistema hardware clasificador de movimiento. La prueba inicia con la conexión física entre el sistema hardware y el computador por medio del cable RS-232. En el sistema de cómputo se ejecuta la interfaz gráfica haciendo uso del software de Matlab y desde el software MPLAB se ejecuta el código en c con el cual será programado el circuito integrado (DSP) y que contiene las instrucciones para la recepción de

datos. Para la verificación de las líneas de código se usa la herramienta *Debugger* en el software de MPLAB que se inicializa de la siguiente manera: se selecciona *Debugger* en la barra de menú, se elige la herramienta de programación en select tool que para este caso fue la *Pickit 3*, al final se programa el dispositivo y se ejecuta el código.

Los datos que se envían están en formato exponencial de la forma x.xxxxexxx con una posición inicial que identifica el signo. Los datos recibidos son comparados con los enviados con la ayuda de la opción watch de MPLAB donde se puede verificar la posición de llegada de cada dato y la información contenida.

Como resultado se obtuvo un error del 0% en la transmisión de los datos y un tiempo estimado de envío 6 segundos que depende altamente de la velocidad de procesamiento del computador.

- **Código clasificador - Sistema hardware:** Para realizar la prueba de integración del sistema hardware y del código clasificador, se realizó un seguimiento específico del código clasificador para detectar posibles fallas en su funcionamiento, la prueba consistió en depurar el código paso a paso, verificando las variables del sistema. Para realizar esta prueba se inicia seleccionando la opción *Debugger* en el software de MPLAB, a continuación se introducen 3 breakpoint los cuales fueron ubicados de la siguiente manera: el primero fue ubicado después de las líneas de código que contiene las instrucciones de recepción del vector de datos, un segundo breakpoint fue ubicado en la parte media del algoritmo, y el tercero al finalizar la función de clasificación. Se corroboró el buen funcionamiento al controlar el cambio de estado de las variables en cada tramo de código y se contabilizó el tiempo requerido de ejecución. Como resultado se obtuvo un tiempo de ejecución

aproximado de 57.868 ms que fueron calculados desde el momento en que la señal es completamente recibida hasta su clasificación final. Se debe tener claridad que el tiempo calculado se realizó con el Debugger de MPLAB por lo cual los tiempos pueden variar al realizar la clasificación directamente en el SE.

## **5.2. PRUEBA DE FUNCIONALIDAD**

Este tipo de prueba examina si el sistema cubre las necesidades de funcionamiento, acordes a las especificaciones de diseño. En ella se verificó si el sistema lleva a cabo correctamente las funciones requeridas, la validación de los datos y si el comportamiento es adecuado. Para estas pruebas se usó el esquema de prueba de caja negra, ya que interesa saber si funciona o no, independientemente de la forma en que lo haga.

El objetivo de las pruebas consiste en comparar los datos entregados por el sistema clasificador con deducciones obtenidas al ejecutar el código en Matlab y con los resultados obtenidos en la tesis [4], de esta manera se pretende conocer el máximo error que presenta el sistema. Antes de proceder a realizar las pruebas es fundamental corroborar el buen funcionamiento del sistema y la interfaz, de acuerdo al manual de usuario (Anexo B).

De acuerdo a los resultados obtenidos en [4], en donde por medio de la técnica basada en índice de separabilidad de clases CSI, se seleccionaron individualmente los canales para cada paciente correspondientes al menor error de clasificación. En la tabla 17 se observan los canales seleccionados para cada paciente.

Tabla 17. Mejores canales por paciente

Paciente	Mejores canales	
	Ch 1	Ch 2
Paciente 1	6	15
Paciente 2	4	15
Paciente 3	1	4
Paciente 4	3	13
Paciente 5	6	8
Paciente 6	8	12

Para validar el sistema clasificador de intención de movimiento a partir de señales electromiográficas superficiales se examinaron 160 vectores en los canales designados en la tabla 17, para cada uno de los 6 pacientes. Los vectores muestreados se catalogaron como acertados y desacertados. Acertado cuando el movimiento resultado del sistema clasificador de movimiento corresponda al movimiento seleccionado en la base de datos y desacertados cuando el movimiento resultado del sistema clasificador de movimiento no corresponda con el movimiento seleccionado. Con base en estos resultados se obtuvieron los porcentajes de acierto y de error para cada paciente. Los resultados fueron registrados para el paciente 1 en la tabla 18, para el paciente 2 en la tabla 19, para el paciente 3 en la tabla 20, para el paciente 4 en la tabla 21, para el paciente 5 en la tabla 22 y para el paciente 6 en la tabla 23.

Las tablas se organizan de la siguiente manera, en la primera columna se tiene 4 tipos de movimiento en donde el movimiento 1 corresponde a la mano en reposo, el movimiento 2 concierne a la mano abierta, el movimiento 3 concierne al

movimiento de agarre fino, y el movimiento 4 corresponde al movimiento de agarre grueso. En la segunda y la tercera columna se presentan los aciertos y desaciertos respectivamente, estas columnas exponen los resultados de las pruebas realizadas en cada uno de los movimientos. En la cuarta columna se expresa el vector de desacierto en donde se expone la notación del vector o los vectores de la base de datos en los cuales se generó el desacierto.

**Paciente 1:**

Tabla 18. Registros de las pruebas de clasificación en el paciente 1.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	40		
Movimiento 2	40		
Movimiento. 3	40		
Movimiento 4	40		

**Paciente 2:**

Tabla 19. Registros de las pruebas de clasificación en el paciente 2.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	40		
Movimiento 2	40		
Movimiento 3	35	5	43, 47, 50, 53,60.
Movimiento 4	40		

**Paciente 3:**

Tabla 20. Registros de las pruebas de clasificación en el paciente 3.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	40		
Movimiento 2	39	1	101
Movimiento 3	39	1	136
Movimiento 4	37	3	144, 145, 154

**Paciente 4:**

Tabla 21. Registros de las pruebas de clasificación en el paciente 4.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	40		
Movimiento 2	38	2	40, 115
Movimiento 3	39	1	59
Movimiento 4	34	6	62,152, 153, 154,156, 160

**Paciente 5:**

Tabla 22. Registros de las pruebas de clasificación en el paciente 5.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	39	1	19



Movimiento 2	40		
Movimiento 3	39	1	133
Movimiento 4	40		

**Paciente 6:**

Tabla 23. Registros de las pruebas de clasificación en el paciente 6.

<b>Movimientos</b>	<b>Aciertos</b>	<b>Desaciertos</b>	<b>Vector de Desacierto</b>
Movimiento 1	40		
Movimiento 2	35	5	23, 27, 104, 119, 120.
Movimiento 3	40		
Movimiento 4	40		

**5.3. ANÁLISIS DE RESULTADOS**

A partir de los resultados obtenidos en los procedimientos anteriores (tablas 23, 22, 21, 20, 19 y 18) en los cuales se obtuvieron los errores y los aciertos correspondientes a cada paciente y para cada movimiento. De esta forma se definió el porcentaje de acierto y error del sistema en la clasificación de las señales EMGS. Teniendo presente que como patrón las pruebas fueron realizadas por la misma persona en igualdad de condiciones.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + \dots + a_n}{n} \quad (19)$$

En la tabla 24 se indica el porcentaje de acierto y desacierto que se obtuvo en la clasificación de señales EMGS para cada uno de los 6 pacientes. Este porcentaje consiste en obtener la media aritmética (ecuación 19) de los aciertos y desaciertos para todos los registros de clasificación de cada paciente (tablas 23, 22, 21,20, 19,18). Los porcentajes fueron calculados con los obtenidos en la pruebas de clasificación realizadas por el Magister Harold Romo en [3].

El error total generado por el clasificador (porcentaje de desacierto) fue de 2.711%, un error superior al obtenido en [3] de 2.601%, con una diferencia de 0,11%. En la tabla 24 se puede apreciar y comparar los errores de clasificación obtenidos en este trabajo y los obtenidos en [3]. Se aprecia que para algunos pacientes el presente trabajo realizo mejor la clasificación de las señales EMGS, pero en otros como los pacientes 2,3 y 4 el error de clasificación en el dispositivo fue mayor.

Tabla 24. Resultados de clasificación, aciertos y desaciertos.

<b>Paciente</b>	<b>Porcentaje de acierto (%)</b>	<b>Errores de clasificación obtenidos (%)</b>	<b>Errores de clasificación obtenidos en [3] (%)</b>
Paciente 1	100	0.00	1.25
Paciente 2	96.87	3.13	0.00

Paciente 3	96.87	3.13	1.25
Paciente 4	94.37	5.63	3.12
Paciente 5	98.75	1.25	3.12
Paciente 6	96.87	3.13	6.87
TOTAL	97,288	2,711	2.604

En la tabla 25 se evidencia el porcentaje de acierto y desacierto discriminado para cada movimiento, en donde se especifica la frecuencia con la que el movimiento fue clasificado acertadamente para todos los pacientes (sumatoria de los aciertos de un determinado movimiento en todos los pacientes) de esta forma y utilizando la ecuación 19 se obtienen los porcentajes de acierto y desacierto para cada movimiento.

Tabla 25. Porcentajes de acierto y desacierto para cada movimiento.

<b>Movimiento</b>	<b>Frecuencia acierto</b>	<b>Porcentaje de acierto (%)</b>	<b>Porcentaje de desacierto (%)</b>
Movimiento 1	239	99.58	0.42
Movimiento 2	232	96.66	3.34
Movimiento 3	231	96.25	3.75

Movimiento 4	232	96.66	3.34
--------------	-----	-------	------

El mayor error se presenta en el paciente 4 correspondiente a un desacierto del 5.63%, en los pacientes 2,3, y 6 el porcentaje de desacierto fue del 3.13%, mientras en los pacientes 1 y 5 el porcentaje de desacierto es de 0% y 1.25% respectivamente. Esta tabla nos ilustra un porcentaje de error inferior para algunos pacientes y en determinados movimientos, pero que se ajusta aceptablemente a los obtenidos en el trabajo de Mg. H Romo.

## 6. CONCLUSIONES Y TRABAJOS FUTUROS

### 6.1. CONCLUSIONES

El principal propósito de este trabajo consistió en construir un sistema, diferente al PC, que fuera capaz de detectar la intensión de movimiento de un paciente a partir de señales EMG. Para alcanzar este propósito, primero, se realizó un estudio de las señales EMG, haciendo énfasis en las señales de superficie. Este estudio también incluyó un análisis de los principales componentes del proceso de reconocimiento de patrones como lo son: la extracción de características, reducción de dimensionalidad y la clasificación de dichos patrones. Por otra parte, teniendo en cuenta que el sistema que se propuso desarrollar en el actual trabajo corresponde a un sistema embebido, se realizó un estudio descriptivo de estos sistemas y de las partes principales que lo componen. En este mismo orden de ideas, se hizo una recopilación de algunos de los trabajos de investigación y/o desarrollo más representativos en el área de clasificación de patrones relacionados con señales EMG y dispositivos procesadores de señal distintos al PC, esto con el fin de observar los aportes realizados.

El dispositivo integrado compatible con el algoritmo de clasificación seleccionado y descrito en el capítulo dos del presente trabajo, fue el DsPIC33fj256 de Microchip, ya que cuenta con todas las características necesarias para el desarrollo de la tarjeta.

Para la adecuación del algoritmo de clasificación implementado en Matlab a C, no se utilizó el Toolbox de Matlab "MATLAB/SIMULINK Device Blocksets for dsPIC ", ya que se contaba con una versión demo, la cual presentaba numerables limitaciones.

A diferencia del algoritmo de clasificación original en Matlab implementado en [3], en el desarrollado en el presente trabajo no se tuvo en cuenta, en el proceso de extracción de características, la característica denominada Valores Singulares de los Coeficientes no Nulos, debido a que esta ameritaba una gran carga computacional para su implementación y además los resultados obtenidos fueron muy similares a los del algoritmo original.

Se implementó un interfaz de usuario en Matlab para facilitar la utilización del dispositivo en cuanto a la organización y envío de las señales, ya que el dispositivo clasificador debe estar en constante comunicación con la base de datos contenida en el PC.

El error de clasificación obtenido con el sistema detector de intención de movimiento descrito en el presente trabajo fue de 2,711 por ciento, logrando así un porcentaje de 0.107 por ciento menos que el obtenido por el Ingeniero Mg. Harold Romo en [3]. El paciente que presentó un menor error de clasificación fue el número 1 con un error del cero por ciento. De la misma forma, el paciente con el cual se obtuvo el mayor error fue el número cuatro con un 5,63 por ciento de error.

## **6.2. TRABAJOS FUTUROS**

A continuación se mencionan algunos posibles trabajos que podrían aportar como complemento al presente proyecto:

- I. La implementación de un sistema de adquisición de señales EMG, el cual capte dichas señales directamente desde la extremidad del paciente y las adecúe al formato indicado para su posterior procesamiento en el sistema detector de intención de movimiento que se plantea en el presente trabajo. Además, este sistema permitiría la creación de una base de datos con

señales del paciente, con la cual se pueda entrenar la red neuronal que realizará la posterior clasificación.

- II. El desarrollo de un sistema virtual que permita realizar el proceso de entrenamiento del paciente. Este proceso permitiría una mejor adaptación al sistema de clasificación, mostrándole la forma correcta de interactuar con el dispositivo.
- III. La integración del sistema detector de intención de movimiento que se plantea en el presente trabajo con el *Prototipo Funcional de Prótesis de Mano Robótica* del grupo de Automática Industrial del Departamento de Electrónica, Instrumentación y Control.

## 7. BIBLIOGRAFÍA

- [1]. P. Morales, D. Rubén, B Morales, D. Andrés, P. Grisales, V. HUGO, “Caracterización de Señales Electromiográficas para la Discriminación de Seis Movimientos de la Mano”, Universidad Tecnológica de Pereira, Colombia 2009.
- [2]. R. Merletti y P. Parker, “Electromyography: Physiology, Engineering and Non-Invasive Applications”. IEEE Press and John Wiley Publishers, 2001.
- [3]. H. Romo, “Aplicación de Wavelets en el análisis y la extracción de características para la clasificación de señales EMGS”, Universidad del Cauca, 2008.
- [4]. F. Córdoba, “Fundamentos Biológicos del Aprendizaje y la Memoria”, Universidad de Huelva, 2005.
- [5]. M. García, “Valoración de la electromiografía de superficie multicanal para el análisis de la mialgia de esfuerzo en músculos del antebrazo” Universidad Politécnica de Cataluña, Barcelona 2010.
- [6]. H. Romo, C. Realpe y E. Jojoa, “Análisis de Señales EMG Superficiales y su aplicación en Control de Prótesis de Mano” Universidad del Cauca, 27 de mayo 2007.
- [7]. C. Belalcázar, Z. Rengifo, “Clasificación de señales electromiográficas (EMG) para una prótesis de mano”, Universidad del Cauca, 2007.
- [8]. K. Englehart, B. Hudgins, P.A. Parker, and M. Stevenson, “Classification of the Myoelectric Signal using Time-Frequency Based Representations”



Institute of Biomedical Engineering, and Department of Electrical and Computer Engineering, Canada, 1999.

- [9]. P. Faundez, A. Fuentes, "Procesamiento Digital de Señales Acústicas utilizando Wavelets". Instituto de Matemáticas UACH.
  
- [10]. M. Sánchez, "Sistema de detección de intrusos en redes de comunicaciones utilizando redes neuronales". Departamento de Ingeniería Electrónica, Escuela de Ingeniería, Universidad de las Américas Puebla. Diciembre. 2004.
  
- [11]. E. Aldabas, "Introducción al reconocimiento de patrones mediante redes neuronales", Universidad Politécnica de Cataluña, Campus de Terrassa, Terrassa, Barcelona, 2002.
  
- [12]. J. castro, "Nuevas metodologías no invasivas de diagnosis de defectos incipientes en rodamientos de bola", Universidad Carlos III de Madrid, Mayo 2007.
  
- [13]. S. Valero Cubas, "Arquitectura de búsqueda basada en técnicas Soft Computing para la resolución de Problemas Combinatorios en Diferentes Dominios de aplicación", Universidad politécnica de Valencia, departamento de sistemas informáticos y computación. Valencia febrero 2010.
  
- [14]. L. Jiménez Moscovitz, P. Rengifo Rengifo, "Al interior de una máquina de soporte vectorial" Facultad de Ciencias Naturales y Exactas Universidad del Valle, Marzo 16 de 2010.
  
- [15]. F. Sandoval "Diseño y construcción de un módulo de Monitoreo y control del suministro de energía eléctrica a un data center a través de la internet." Escuela politécnica Nacional de Ecuador, 2008

- [16]. C. Pavón Castillo, O. Cruz Arguello, "Historia, manejo y aplicaciones de los controladores digitales de señales DSPIC". Facultad de Electrotecnia y Computación, Universidad Nacional de Ingeniería Managua, Nicaragua.
- [17]. González Ortega, "Estudio de Sistemas Digitales de Altas Prestaciones para Aplicaciones en Electrónica de Potencia", Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México. Abril 30 de 2008.
- [18]. G. Galeano "Programación de sistemas embebidos en C, teoría y práctica aplicadas a cualquier microcontrolador" universidad pontificia bolivariana de Medellín, 2009.
- [19]. Scribd, "Tutorial. "Arquitecturas Harvard y Von Neumann", [En línea]. <http://es.scribd.com/doc/50884379/Arquitecturas-Harvard-y-Von-Neumann>
- [20]. Punto flotante S.A, "Tutorial. Estándares de comunicaciones RS-232, RS-422/485" [En línea]. <http://www.puntoflotante.net/RS485.htm>
- [21]. M. Godínez García "Desarrollo de un instrumento virtual para control de un equipo de Potenciales Evocados" Universidad autónoma metropolitana. Departamento de ingeniería eléctrica. Noviembre 2003.
- [22]. J. Salvador, H.de Bruin, "The Use of the Wavelet Transform in EMG M-Wave Pattern Classification", Engineering in Medicine and Biology Society, 2006. EMBS '06.28th Annual International Conference of the IEEE. 2006.
- [23]. Z. Xizhi, "Study of Surface Electromyography Signal Based on Wavelet Transform and Radial Basis Function Neural Network", Future BioMedical Information Engineering, 2008. FBIE '08. International Seminar, 2008].

- [24]. L. Zhihong, L. Zhizeng, "Hand motion pattern classifier based on EMG using wavelet packet transform and LVQ neural networks", IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium, 2008.
- [25]. P. Parimal, S. Allen y G. Rapach (1988). "Microcontroller based prosthetic device using myoelectric signal". Engineering in Medicine and Biology Society, Proceedings of the Annual International Conference of the IEEE , 1427 – 1428.
- [26]. C. Fermo, C. Vincenzo, y V. Dynnikov (2000). "Development of an adaptative framework for the control of upper limb myoelectric prosthesis". Engineering in Medicine and Biology Society, 2402-2405.
- [27]. N. López, F. di Sciasco, E. Orosco, N. Ledesma, A. Echenique, and M. Valentinuzzi (2006). "Control Mioeléctrico de un brazo robótico". IV Congreso Iberoamericano sobre Tecnologías de Apoyo a la Discapacidad. Vittoria-Espiritu Santo-Brasil.
- [28]. E. Orosco, N. López, C. Soria y M. Guzzo, "Procesamiento De Señales Mioeléctricas Implementado En Procesador Digital De Señales", Universidad Nacional de San Juan, Laboratorio de Electrónica Digital, 2008.
- [29]. L. Zhao, Y. Pengxian, X. Longteng, y M. Qingguo, "Application of digital signal processor in EMG-based human machine interface", Control Conference (CCC), 2010 29th Chinese, 2010.

## 8. ANEXOS

### ANEXO A. Programa para la clasificación de señales EMGS

En este anexo se especifica el código de clasificación de señales EMG realizado en el software MPLAB y en el lenguaje de programación C para el *DSPIC33fj256gp710a* de Microchip. Este se compone, principalmente, de tres partes: la primera consiste en una etapa de adquisición de las señales EMG a través del puerto serial del dispositivo; la segunda etapa realiza la extracción de características temporales y espectrales a partir de la transformada Wavelet discreta, conformando a su vez, el vector de patrones que posteriormente es clasificado en la en la última parte del código por un clasificador de tipo perceptrón multicapas.

Este algoritmo corresponde al paciente número uno de los seis que conforman la base de datos de señales EMG con la cual se trabajó en el presente proyecto.

```
#include <p33FJ256GP710.h>
#include <delay.h>
#include <stdlib.h>
#include <math.h>
#define size 256
_FOSCSEL(1); //Oscilador FRC interno con PLL
_FOSC(195); //Osc primario desabilitado,
Cambio reloj desabilitado

#define FCY 4000000 //Instruction Cycle Frequency
#include <libpic30.h>
#define BAUDRATE 9600 //Baudios
#define BRGVAL ((FCY/BAUDRATE)/16)-1

void nivel_anterior(double mat_descomp[][4], int i, int *p2, double *s );
void DownDyadHi(double *s, int p2, double *hs );
void DownDyadLo(double *s, int *p2, double *ls );
void valor_medio_abs(double *signal, double *patron, int index);
void long_onda(double *signal, double *patron, int index);
void caract_wdtt(double *signal, double *patron, int index);
int clasifica(double *patron);
```

```

/*****VarGlob*****/
double signal[256], signal2[256], Y=0;
int x1=0,x=0;
char dato[15];
int i=0,i1=0,i2=0;
double patron[10];
int index1=1;
float VMA,lo;

/*****Inicializar_UART*****/
void InitUART1(){
    TRISBbits.TRISB1=1;
    TRISBbits.TRISB0=0;
    //RPNR18bits.U1RXR = 7;           //pin RB7 como RX
    //RPOR3bits.RP6R = 0b00011;     //Pin RB6 como TX
    U1MODEbits.STSEL = 0;           // 1-stop bit
    U1MODEbits.PDSEL = 0;           // No Parity, 8-data bits
    U1MODEbits.ABAUD = 0;           // Auto-Baud Disabled
    U1MODEbits.BRGH = 0;            // Low Speed mode
    U1BRG = BRGVAL;                 // BAUD Rate Setting
    U1STAbits.UTXISEL0 = 0;         // Interrupt after one Tx character is transmitted
    U1STAbits.UTXISEL1 = 0;
    IEC0bits.U1TXIE = 0;            // Disenable UART Tx interrupt
    IFS0bits.U1RXIF = 0;           //Clear the Recieve Interrupt Flag
    IEC0bits.U1RXIE = 1;           //Enable Recieve Interrupts
    U1MODEbits.UARTEN = 1;         // Enable UART
    U1STAbits.UTXEN = 1;           // Enable UART Tx
    U1STAbits.URXISEL1 = 1;        // Enable UART Rx
}

```

```

void valor_medio_abs(double *senal, double *patron, int index)
{
    float suma;
    double val_medio;
    int i=0;
    for( i=0; i<size; i++)
    {
        suma = suma + fabs(senal[i]);
    }
    val_medio=suma/size;

    if (index ==1) patron[2]=val_medio;
}

```

```

    if (index ==2) patron[3]=val_medio;
}

```

```

void long_onda(double *senal, double *patron, int index)
{
    float lo_ch = senal[0];
    double delta_xi_ch;
    int i=0;
    for ( i =1; i<size;i++)

        {   delta_xi_ch = senal[i] - senal[i-1];

            lo_ch = lo_ch + fabs(delta_xi_ch);
        }
    if (index ==1) patron[0]=lo_ch;
    if (index ==2) patron[1]=lo_ch;
}

```

```

int clasifica(double *patron)

{

/** MATRICES DE PESOS**

float w1[10][10]={{-2.3786, 3.3123, 3.4364, -13.4183, -5.1783, 28.6256, -24.6496, 23.4213, 1.7795,
2.4282},
    {0.5450, 0.2130, -0.4890, -0.2006, -9.7648, 19.0423, 72.6313, 227.1784, 1.1162, 3.0428},
    {-0.0267, 0.8754, 3.2177, -14.0038, 0.4298, -1.1302, -80.1166, 29.9725, 1.3095, 0.9813},
    {-0.1909, 1.1399, -6.0655, 15.0630, 14.8903, 25.7686, 41.2168, -86.3204, 0.7453, 1.5653},
    {0.2925, 0.6500, -5.9656, -12.7103, 3.9292, -19.6626, 21.7534, 50.5248, 1.4181, -3.8091},
    {-1.1812, 1.2781, 6.4858, -15.3351, -12.9639, -6.3783, 52.6269, -62.6946, -2.0702, -1.2851},
    {1.4313, -1.6044, 7.8701, 2.8745, -5.3693, 37.1601, -24.2288, 173.9898, -0.5906, -0.9485},
    {-0.7277, -0.3996, 9.2188, 9.8204, -15.3209, 25.5621, 41.8422, -41.0584, 0.5026, -2.4233},
    {0.0335, 0.2562, -9.9086, -8.2554, 0.0939, 16.1136, 69.0381, -4.4853, -0.9983, 2.8283},
    {-0.0036, 0.5005, -8.3079, -14.2951, -2.9363, 9.0210, -37.0267, 115.6257, -1.5340, 4.5217}};

float w2[5][10] = {{-1.4411, -2.2813, 0.8115, 0.1875, 1.7279, 1.6050, 0.0056, -1.7123, -0.5120, 0.9957},
    {1.7553, 0.8729, 0.5151, -3.1853, 0.2321, 0.6489, 0.2612, 0.5730, 0.2776, 2.2702},
    {-1.7513, -0.2513, 0.5647, 0.8236, 1.5786, -0.4322, -0.4217, -1.8606, 0.8105, -0.5077},
    {1.0777, 2.3178, 0.2668, 1.0937, -0.7029, 1.0129, 0.1994, 0.8076, -0.3083, -0.3428},
    {-1.2784, 1.9790, -0.7843, 0.0128, -0.7578, -0.4585, -0.2899, -0.1194, 0.0912, -0.7706}};

float w3[5]={-0.0826, -0.5009, 0.8404, 1.5015, -1.4927};

/** VECTORES DE VIAS**

```

```

float bias1[]={1.8182, -13.3056, 0.5422, -4.3184, 3.2676, 5.2546, -4.4344, -0.7881, 1.0900, 3.2248};

float bias2[]={2.0561, 0.1606, 0.4278, 0.7825, -1.6481};

float bias3=0.7514;

float i1=0;
float i2=0;
float i3=0;
float Y1[10];
float Y2[5];
float Y3=0;
int salida;
int j,j2,j3,j4,j5;

/**/ primer capa oculta capa oculta***/
for( j=0;j<=9;j++)
{
    i1=0;
    for (j2=0;j2<=9;j2++)
    {
        i1+=(patron[j2])*(w1[j][j2]);
    }
    i1+=bias1[j];
    Y1[j]=2/(1+exp(-2*i1))-1;

    // printf("%f \n",Y1[j]);
/**/ segunda capa oculta***/
}

for( j3=0;j3<=4;j3++)
{
    i2=0;
    for (j4=0;j4<=9;j4++)
    {
        i2+=(Y1[j4])*(w2[j3][j4]);
    }
    i2+=bias2[j3];
    Y2[j3]=2/(1+exp(-2*i2))-1;

    // printf("%f \n",Y2[j3]);
}

```

```
/** salida de la red**/  
//i3=0;  
for (j5=0;j5<=4;j5++)  
    {  
        i3+=(Y2[j5])*(w3[j5]);  
  
    }  
    i3=i3+ bias3;  
    Y3= floor(i3+0.5);  
if(Y3==1){LATA=0b000010; __delay_ms(10000);}  
else if(Y3==2){LATA=0b000100; __delay_ms(10000);}  
else if(Y3==3){LATA=0b001000; __delay_ms(10000);}  
else if(Y3==4){LATA=0b010000; __delay_ms(10000);}  
  
}
```



## **ANEXO B. Manual de Operación de la Tarjeta clasificadora de intensidad de movimiento a partir de señales EMGS v1.0.**

La tarjeta clasificadora de señales EMGS v1.0, es un dispositivo que clasifica las señales contenidas en una base de datos suministrada por *Ph.D. Kevin Englehart del Instituto de Ingeniería Biomédica de la Universidad de New Brunswick*. Esta versión fue diseñada para propósitos investigativos en aplicaciones de prótesis robóticas de mano, en particular para el *Prótipo Funcional de Prótesis de Mano Robótica* que se adelanta en la Universidad de Cauca - Colombia.

Este manual de usuario está dirigido a personas encargadas de la configuración y puesta en funcionamiento de la tarjeta, además de la actualización del software. En la sección se describe el programa que implementa el algoritmo de procesamiento embebido. Las conexiones básicas de esta tarjeta se observan en la figura 27.

Los aspectos que se incluyen en este manual son:

- Componentes de la Tarjeta.
- Funcionalidades y características de la Tarjeta.
- Instalación y manejo del Pickit 3
- Interfaz de usuario.

## COMPONENTES DE LA TARJETA

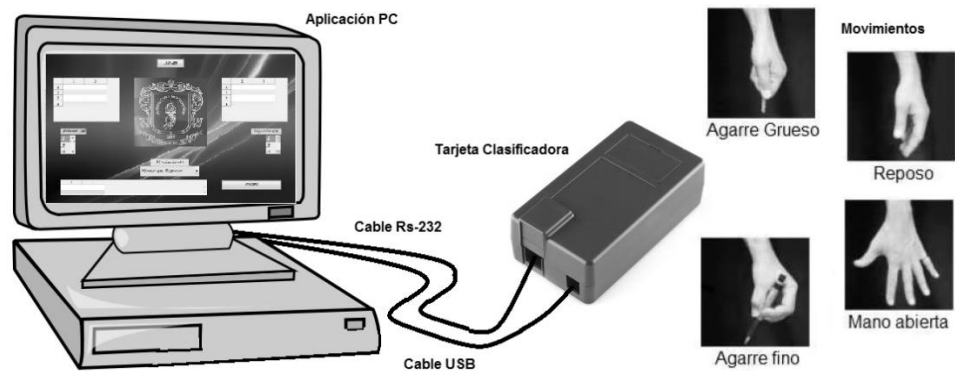


Figura 27. Conexión tarjeta clasificadora

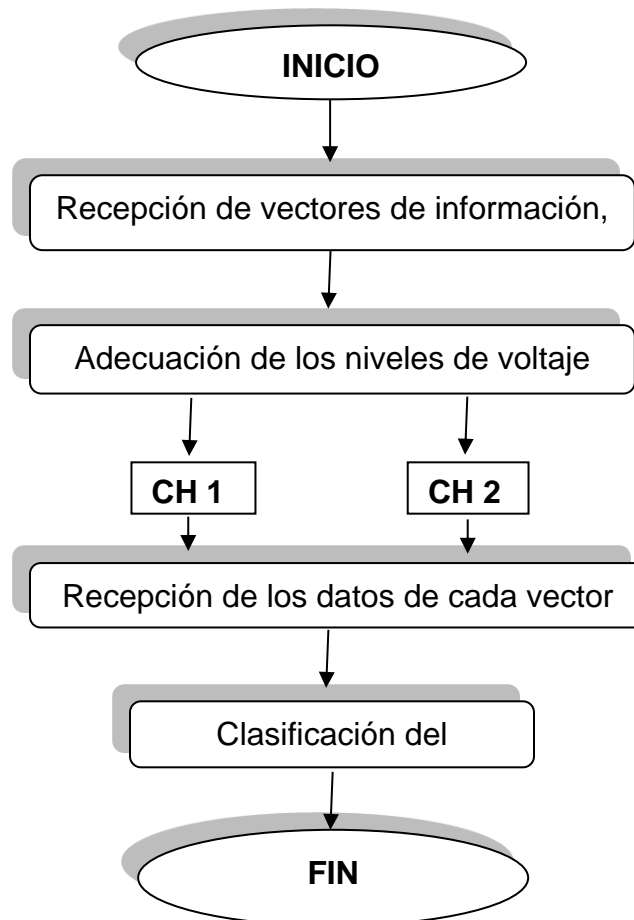


Figura 28. Diagrama de flujo tarjeta clasificadora de intención de movimiento

## FUNCIONALIDADES Y CARACTERÍSTICAS DE LA TARJETA

La tarjeta de procesamiento se encarga de recibir y clasificar las señales EMGS almacenadas en la base de datos especificada. Esta versión ha sido diseñada para procesar dos canales de información. Las funciones principales que se implementan en la tarjeta pueden ser resumidas para en el diagrama de bloques de la Figura. 28.

La tarjeta está provista con un canal de comunicación serial RS-232. Esta interfaz se encarga de transmitir muestras procesadas en el DSP desde el PC. La interfaz se implementa con terminales de comunicación UART (U1ARX) y con el integrado MAX232.

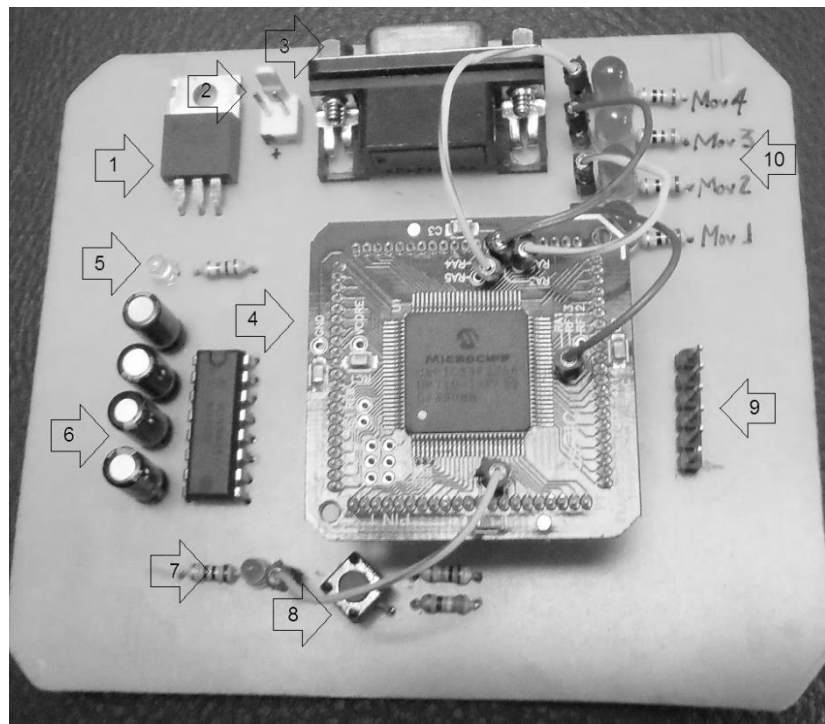


Figura 29. Tarjeta clasificadora y sus partes

- I. **Regulador de voltaje:** Para normalizar el voltaje de la entrada del *DSPIC33fj256gp710a*, se eligió el dispositivo LM1117T de *National Semiconductor*. Este es un regulador de voltaje de baja potencia, capaz de proporcionar hasta 800 mA de corriente de salida, con un voltaje de referencia ajustable de 1,25 V. Para el desarrollo del sistema embebido es necesario escoger como voltaje de salida 3.3 voltios con el cual se alimentara el dsPIC33fj256gp710a, los dispositivos restantes trabajan a 5 voltios (Figura 29).
- II. **Conector de poder:** Conector Molex de dos pines. Destinado a proveer a la placa base de la tensión de alimentación, proveniente de la fuente. Este conector garantiza fiabilidad y capacidad para el consumo que se requiera. Los conductores eléctricos que salen de la fuente de alimentación hacia conectores Molex tienen colores para distinguirlos (blanco - negro), además el conector está etiquetado para su correcta conexión.
- III. **Conector serial DB9:** La tarjeta está provista con un canal de comunicación serial RS232. Esta interfaz se encarga de recibir los vectores de información desde el PC. La interfaz se implementa con terminales de comunicación UART y con el integrado MAX232.

Para la función de conexión física de la tarjeta se adiciono un conector db9 hembra, e cual consta de 9 pines, cada pin cumple una función especial la información asociada en los pines es la misma de la siguiente manera:

**Pin 5** GND – Ground

**Pin 4** DTR – Data Terminal Ready

**Pin 3** TX – Transmission

**Pin 2** RX – receives

**Pin 1** DCD – Data Carrier Detect

**Pin 9** RI – Ring indicator

**Pin 8** CTS – Clear to send

**Pin 7** RTS – Request to send

**Pin 6** DSR – Data Sheet Ready

- IV. Modulo MA330012:** El circuito integrado escogido para realizar las diferentes operaciones lógicas, es el procesado de señal (*DSPIC33fj256gp710a*) en cual se encuentra integrado sobre el módulo MA330012. Este modulo integra tanto el circuito integrado de microchip como también 5 capacitores en su configuración adecuada y un socket para el acceso a cada uno de los 84 pines, además 17 pines que se encuentran en la superficie del modulo, lo que permite una sencilla y rápida conexión.
- V. Led indicador de poder:** Este LED indica la usuario la habilitación de potencia para la tarjeta clasificadora.
- VI. Max-232:** Muchos microcontroladores poseen una interfaz UART o USART para comunicación serial asincrónica, tipo RS-232, que en un PC se denomina puerto “COM”. Si bien los microcontroladores poseen hardware para generar la secuencia de bits en los tiempos correctos, no son capaces de generar el voltaje especificado por el estándar RS-232, por lo cual requieren de un chip externo que haga esta conversión de voltajes.

Si bien el circuito de conversión puede armarse fácilmente en un protoboard, disponer del circuito en un módulo tiene las siguientes ventajas:

Ahorro de espacio en el protoboard.

Ahorro de tiempo en el armado y testeo del circuito.

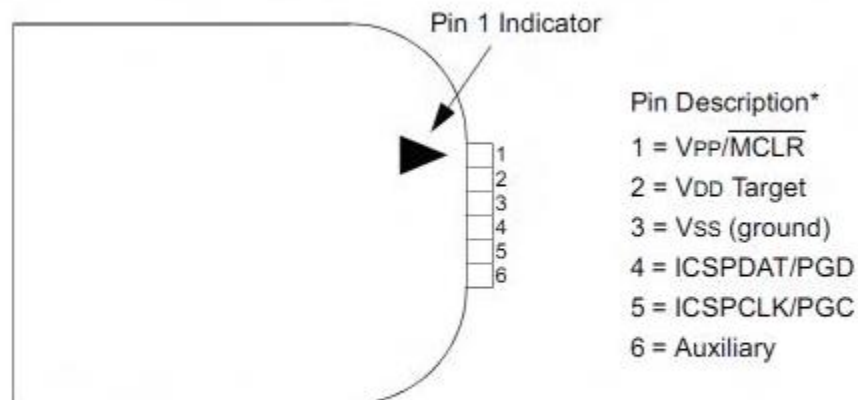
Evita conexiones erróneas del chip conversor.

Pueden incluirse LED monitores.

Para realizar la adaptación de niveles de voltaje TTL al estándar RS-232 se eligió el dispositivo Max-232 de *Maxim electronics*, el cual dispone internamente de 4 conversores de niveles TTL al estándar RS-232 y

viceversa, para ser usados en la comunicación serie con ordenadores en los puertos COM1 y el COM2.

- VII. **Led indicador de usuario:** Este LED está conectado al DSP. EL LED de usuario esta dispuesto para informara al usuario cundo la tarjeta esta lista para recibir una nueva trama de vectores, si el led no esta encendido indica que el DSP esta realizando el proceso de clasificación por lo cual no eta listo para recibir.
- VIII. **Botón reset:** Es un push button dispuesto para re-iniciar el algoritmo sobre el DSP
- IX. **Conector para PicKit 3:** A través de este conector es posible conectar el DSP al MPLAB PicKit3 para programación y/o depuración. En la figura 30 se ilustran los terminales utilizados



\* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

Figura 30. Conectores de la programadora pickit3

- X. **Leds indicadores de clasificación:** Estos leds indican la clasificación realizada por el sistema, de esta manera cada led indica un movimiento específico.

Led rojo (Parte inferior de la figura): Mano relajada

Led verde: Mano abierta

Led amarillo: Agarre fino

Led rojo (parte superior de la figura): Agarre grueso

## INTERFAZ DEL USUARIO

La interfaz de usuario fue creada con el guide de Matlab y es fundamental para ordenar y enviar la señal a la tarjeta clasificadora. En este espacio se realiza una descripción del entorno GUIDE de Matlab y se explica el funcionamiento de la interfaz grafica.

### Manejo GUIDE de Matlab.

Matlab Guide\* es un entorno de programación visual que ofrece Matlab para poder realizar y ejecutar programas de Simulación a medida de forma simple , tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

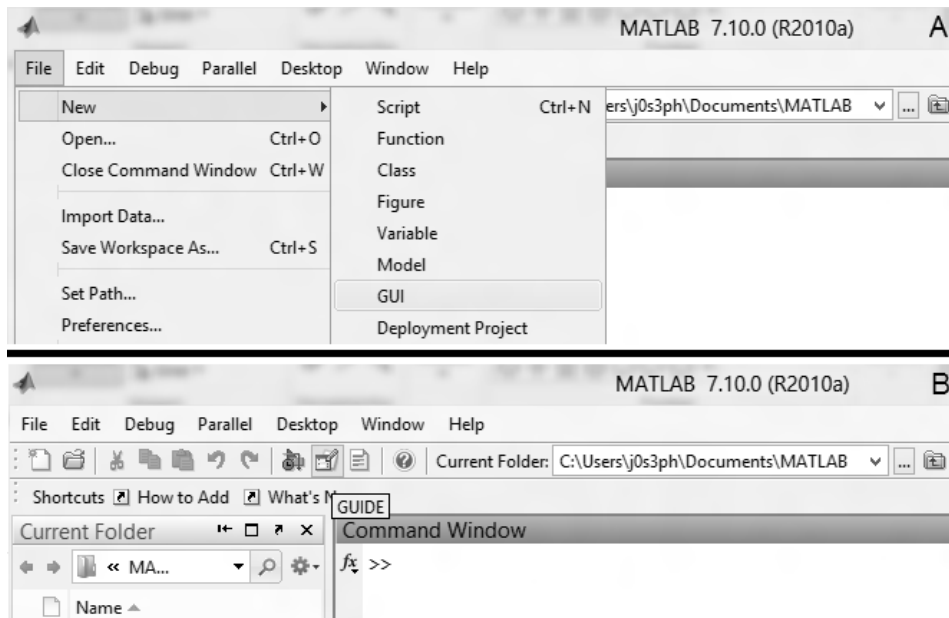


Figura 31a y 31b. Interfaz inicio del GUIDE

Para ejecutar este entorno se efectúa desde la ventana de comando del Matlab (figura 31a y 31b) o ejecutando desde el menú principal -> File ->New -> GUI. Este comando ejecuta la consola de edición de la parte grafica, un área de diseño desde compuesta por distintos botones de comando y menús. Desde esta nueva interfaz es sencillo adicionar botones, cuadros de dialogo, graficas, texto, que permiten la construcción de un entorno grafico de fácil entendimiento para el usuario figura 32. En la parte superior se encuentran los menús y opciones de GUIDE, en la parte izquierda se aprecian los diferentes controles y en la parte central el área de diseño donde pondremos los controles a usar.

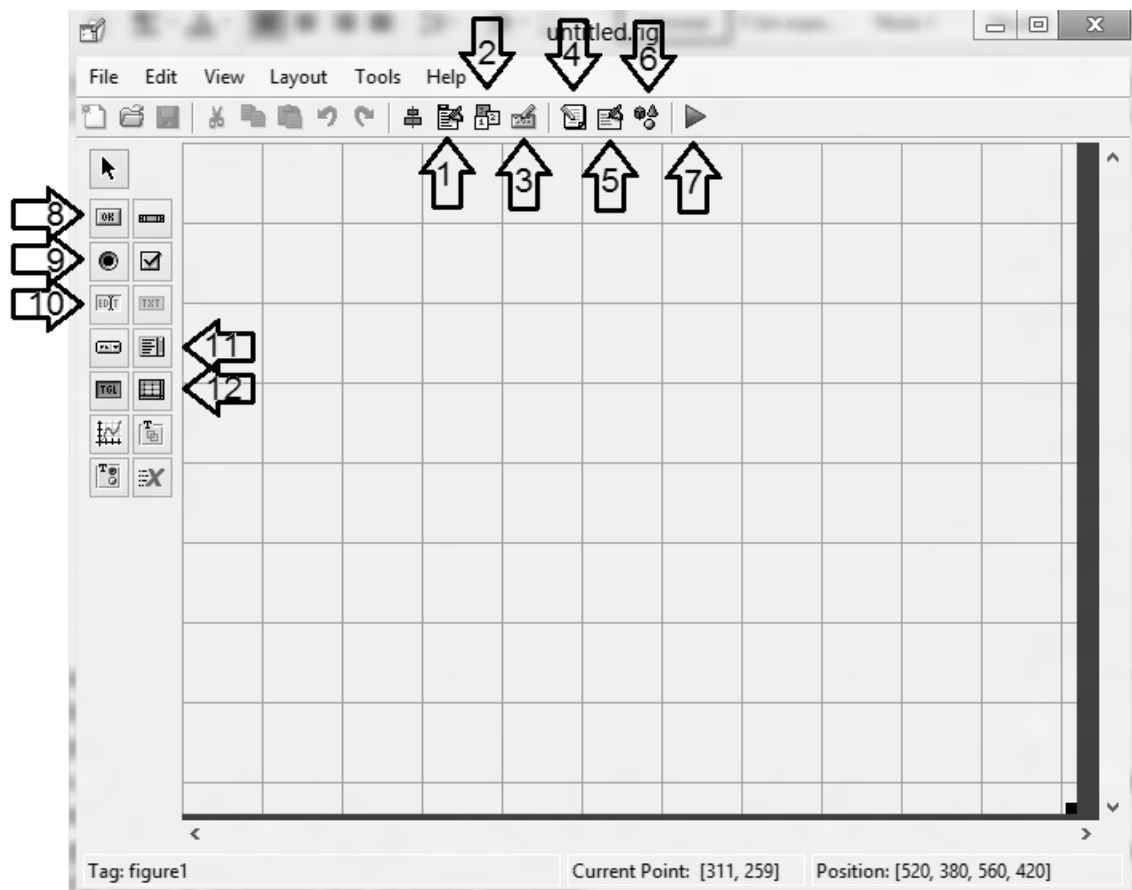


Figura 32. Área de diseño del GUIDE



## Las partes más importantes de GUIDE son:

- I. **Editor de menú:** Cada control cuenta con diferentes propiedades y es aquí donde podremos cambiar: el color, el nombre, el tag, el valor, el callback entre otros.
- II. **Editor para el orden de tab:** Asigna un determinado orden a la estructura de la interfaz
- III. **Editor de barra de herramientas:** Edición de la barra de hermanitas, podemos cambiar, eliminar o añadir botones.
- IV. **Editor de archivo .M:** Accede al archivo .M generado por entorno.
- V. **Inspector de propiedades:** Inspección de las propiedades de los diferentes objetos añadidos.
- VI. **Buscador de objetos:**
- VII. **Ejecutar:** Una vez que hayamos terminado de diseñar presionamos este botón para activar la figura y poder probar nuestra GUI.
- VIII. **Crear botón:** Crea botones
- IX. **Crear botón circular:** Crea botones circulares
- X. **Crear campo de texto:** Crear campos de texto
- XI. **Crear lista de texto:** Crea listas de texto
- XII. **Crear tabla:** Crea tabla

Cada uno de estos elementos poseen un conjunto de propiedades a las cuales se puede acceder con el botón derecho del mouse, una vez clickeado este surge una ventana figura 33.

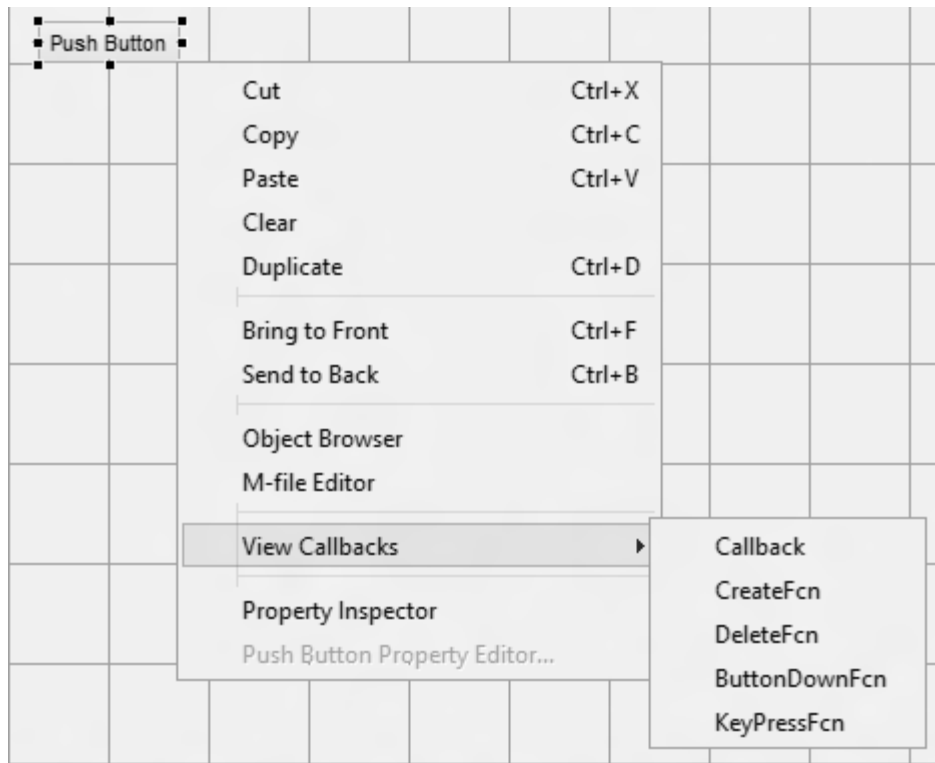


Figura 33. Propiedades de los elementos

Para editar las propiedades de cada elemento seleccionamos la opción *Properties Inspector* y se abre una consola con todas las propiedades que podemos editar, color, posición, tamaño, etc. Una de las opciones de mayor interés es *Edit Callback*. Esta función abre el archivo .m asociado a un código ejecutable por Matlab y nos posiciona en la sección del programa que corresponde a la subrutina que se ejecutara cuando se realice una determinada acción sobre el elemento que estamos editando.

El guide consta de dos archivos uno .m (ejecutable) y otro .fig la parte en donde se encuentran los gráficos. Las dos partes están unidas a través de las subrutinas callback. Una vez que se graba los archivos desde la ventana de programa, podemos ejecutar el programa en la ventana de comando de Matlab escribiendo el nombre del archivo.

El archivo .m que se crea tiene una estructura predeterminada. Consta de un encabezado y a continuación viene el código correspondiente a las siguientes subrutinas de los comandos añadidos, al interior de cada subrutina se encuentra la configuración de código que el programador determine, de esta forma se le asigna una función.

### Partes y manejo de la interfaz de usuario

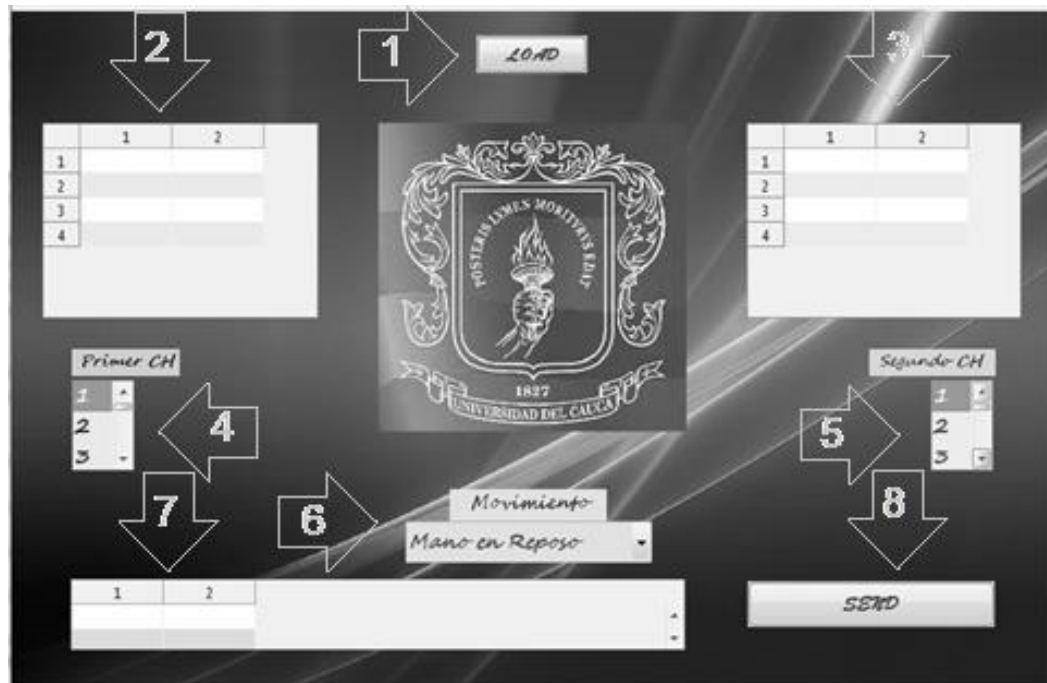


Figura 34. Interfaz grafica de usuario

- I. **Botón de cargar:** Este botón sirve para cargar los datos de un determinado paciente. Cuando presionamos el botón se abre una ventana que no solicitara la dirección en donde tengamos ubicada la base de datos, ahí seleccionamos el paciente y damos aceptar.
- II. **Tabla indicadora CH 1:** Indica la base de datos seleccionada en el canal 1 (tabla de 256x160)
- III. **Tabla indicadora CH 2:** Indica la base de datos seleccionada en el canal 2 (tabla de 256x160)

- IV. **Selector de CH 1:** Selector para el canal uno (Figura 34).
- V. **Selector de CH 2:** Selector para el canal dos.
- VI. **Selector de movimiento:** Con este selector se puede seleccionar uno de los cuatro movimientos a clasificar.
- VII. **Tabla indicador de vector de envío:** Esta tabla indica el vector que será enviado a SI.
- VIII. **Botón de envío:** Este botón realiza el envío.