

	<b>GESTIÓN DE RECURSOS Y SERVICIOS BIBLIOTECARIOS</b>	<b>Código</b>	FO-SB- 12/v0
	<b>ESQUEMA HOJA DE RESUMEN</b>	<b>Página</b>	1/1

**RESUMEN TRABAJO DE GRADO**

AUTOR(ES):

NOMBRE(S): EDGAR ALEXANDER APELLIDOS: MENDIVELSO PEÑARANDA

NOMBRE(S): JONATHAN JAVIER APELLIDOS: RINCON MEZA

FACULTAD: INGENIERÍA

PLAN DE ESTUDIOS: INGENIERÍA ELECTROMECAÁNICA

DIRECTOR:

NOMBRE(S): NORBEY APELLIDOS: CHINCHILLA HERRERA

CODIRECTOR:

NOMBRE(S): \_\_\_\_\_ APELLIDOS: \_\_\_\_\_

TÍTULO DEL TRABAJO (TESIS): IDENTIFICACION DE MOVIMIENTO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS PRESENTES EN EL ANTEBRAZO, UTILIZANDO REDES NEURONALES ARTIFICIALES EN MATLAB

RESUMEN

Este proyecto realizó una identificación de movimiento a partir de señales electromiograficas presentes en el antebrazo, utilizando redes neuronales artificiales en Matlab. Para ello, se realizó una investigación tipo investigativo y desarrollo (proyecto I+D) ya que se complementa de manera dinámica el incremento del conocimiento y engloba tres actividades: investigación básica, investigación aplicada y desarrollo experimental. La recolección de información se basó en libros especializados de acceso público que se encuentran en la Red. Se logró, identificar el movimiento a partir de las señales electromiografías que se presentan en los músculos del antebrazo humano, mediante la implementación de redes neuronales artificiales (RNA) utilizando el software MATLAB. Se elaboro el estado del arte. Se llevó a cabo la selección de los circuitos eléctricos, electrónicos o dispositivos correspondientes para realizar la toma de señales electromiografías. Seguidamente, se implementó la interfaz que permita visualizar de manera didáctica los movimientos parciales producidos en la mano y los dedos, dependiendo de las señales electromiografías características. Posteriormente, se validó el correcto funcionamiento de la interfaz. Finalmente, se socializaron los resultados del proyecto con la comunidad académica.

PALABRAS CLAVE: redes neuronales, señales electromiograficas, Matlab.

CARACTERÍSTICAS:

PÁGINAS: 142 PLANOS: ILUSTRACIONES: CD ROOM: 1

Elaboró		Revisó		Aprobó	
Equipo Operativo del Proceso		Comité de Calidad		Comité de Calidad	
<b>Fecha</b>	24/10/2014	<b>Fecha</b>	05/12/2014	<b>Fecha</b>	05/12/2014

COPIA NO CONTROLADA

IDENTIFICACION DE MOVIMIENTO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS  
PRESENTES EN EL ANTEBRAZO, UTILIZANDO REDES NEURONALES ARTIFICIALES  
EN MATLAB.

EDGAR ALEXANDER MENDIVELSO PEÑARANDA

JONATHAN JAVIER RINCON MEZA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTROMECAÁNICA

SAN JOSÉ DE CÚCUTA

2020

IDENTIFICACION DE MOVIMIENTO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS  
PRESENTES EN EL ANTEBRAZO, UTILIZANDO REDES NEURONALES ARTIFICIALES  
EN MATLAB.

EDGAR ALEXANDER MENDIVELSO PEÑARANDA

JONATHAN JAVIER RINCON MEZA

Trabajo de grado presentado como requisito para optar al título de:

Ingeniero Electromecánico

Director:

NORBEY CHINCHILLA HERRERA

Ingeniero Electromecánico

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTROMECAÁNICA

SAN JOSÉ DE CÚCUTA

2020

**ACTA DE SUSTENTACIÓN PROYECTO DE GRADO  
MODALIDAD TRABAJO INVESTIGATIVO**

**FECHA:** 26 de febrero de 2020

**HORA:** 10:00 A.M

**LUGAR:** Sala 3 del Cread Ufps

**PLAN DE ESTUDIOS:** INGENIERÍA ELECTROMECAÁNICA

**TITULO DEL TRABAJO DE GRADO:** "IDENTIFICACIÓN DE MOVIMIENTO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS PRESENTES EN EL ANTEBRAZO UTILIZANDO REDES NEURONALES ARTIFICIALES EN MATLAB".

**JURADOS:** Ph.D: JOHNNY OMAR MEDINA DURAN

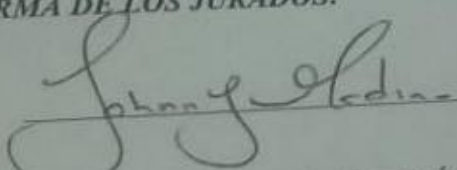
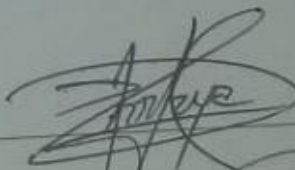
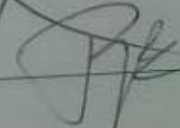
Esp: RONI MAURICO JAYA CAMACHO

**DIRECTOR:** MSc. NORBEY CHINCHILLA HERRERA

**APROBADA**

<b>NOMBRE DEL ESTUDIANTE:</b>	<b>CÓDIGO</b>	<b>CALIFICACION</b>
JONATHAN JAVIER RINCON MEZA	1090437	4.4
EDGAR ALEXANDER MENDIVELSO PEÑARANDA	1090338	4.4

**FIRMA DE LOS JURADOS:**

  
  
  
VOBO. COORDINADOR COMITÉ CURRICULAR

Mayerlin Ch.



**CARTA DE AUTORIZACIÓN DE LOS AUTORES PARA  
LA CONSULTA, LA REPRODUCCIÓN PARCIAL O TOTAL Y LA PUBLICACIÓN  
ELECTRÓNICA DEL TEXTO COMPLETO**

Cúcuta,

Señores  
BIBLIOTECA EDUARDO COTE LAMUS  
Ciudad

Cordial saludo:

Yo Jonathan Javier Rincón Meza identificado con la C.C. N° 1.090.407.160 de Cúcuta y Edgar Alexander Mendivelso Peñaranda identificado con la C.C. N° 1.090.428.159 de Cúcuta, autores de la tesis y/o trabajo de grado titulado **IDENTIFICACION DE MOVIMIENTO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS PRESENTES EN EL ANTEBRAZO, UTILIZANDO REDES NEURONALES ARTIFICIALES EN MATLAB.** presentado y aprobado en el año 2020 como requisito para optar al título de **Ingeniero Electromecánico**; autorizamos a la biblioteca de la Universidad Francisco de Paula Santander, Eduardo Cote Lamus, para que con fines académicos, muestre a la comunidad en general a la producción intelectual de esta institución educativa, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios pueden consultar el contenido de este trabajo de grado en la página web de la Biblioteca Eduardo Cote Lamus y en las redes de información del país y el exterior, con las cuales tenga convenio la Universidad Francisco de Paula Santander.
- Permita la consulta, la reproducción, a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato CD-ROM o digital desde Internet, Intranet etc.; y en general para cualquier formato conocido o por conocer.

Lo anterior, de conformidad con lo establecido en el artículo 30 de la ley 1982 y el artículo 11 de la decisión andina 351 de 1993, que establece que **“los derechos morales del trabajo son propiedad de los autores”**, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

Jonathan Rincón  
Jonathan Javier Rincón Meza  
C.C 1.090.407.160

EDGAR MENDIVELSO  
Edgar Alexander Mendivelso Peñaranda  
C.C 1.090.428.159

## Contenido

	<b>pág.</b>
Introducción	17
1. Problema	18
1.1 Título	18
1.2 Planteamiento del Problema	18
1.3 Formulación	18
1.4 Objetivos	19
1.4.1 Objetivo general	19
1.4.2 Objetivos específicos	19
1.5 Justificación	19
1.6 Limitaciones	20
1.7 Delimitaciones	21
1.7.1 Delimitación temporal	21
1.7.2 Delimitación de espacio tiempo	21
2. Marco Referencial	22
2.1 Antecedentes	22
2.2 Estado del Arte	26
2.2.1 Anatomía de la mano y el antebrazo humano	26
2.2.2 La Mano	27
2.2.3 Los dedos y su movimiento	29
2.2.4 Estructura ósea desde el antebrazo a los dedos	32
2.2.5 Movimientos de la muñeca	34
2.2.6 Tipos de movimiento	35

2.2.7 Músculos del antebrazo y movimientos en los que intervienen	37
2.2.8 Señales Electromiográficas	39
2.2.9 Dificultades para la toma de señales electromiográficas (EMG)	41
2.2.10 Sensor muscular	43
2.2.11 Convertidor analógico/digital	51
2.2.12 Microcontrolador	51
2.2.13 Arduino	52
2.2.14 Redes Neuronales artificiales (RNA)	56
2.2.15 Interfaz grafica	62
2.2.16 MathWorks versión 2017b de las familias de productos de MATLAB y Simulink	65
2.3 Marco Contextual	67
3. Diseño Metodológico	69
3.1 Tipo de Proyecto	69
3.2 Técnicas de Recolección de Datos	69
4. Desarrollo de Proyecto	70
4.1 Selección de Microcontrolador	70
4.2 Selección del Sensor Muscular	71
4.3 Conexión del Sensor MyoWare y la Interfaz IDE de Arduino	76
4.4 Selección de la Posición final del Sensor Muscular	80
4.5 Conexión de Arduino y Matlab	81
4.6 Diagrama de Bloques general del Proyecto	82
4.7 Lectura y Almacenamiento de Las señales Musculares en MATLAB	83
4.7.1 Creación de los vectores de datos pertenecientes a los diferentes movimientos	87

4.8 Definición de Características Presentes en Las señales	92
4.8.1 Medición de parámetros que miden la forma y la oscilación de las señales	99
4.9 Integración del código, y Revisión de la Respuesta de los Algoritmos	124
4.10 Interfaz Gráfica (GUI)	127
5. Conclusiones	137
6. Recomendaciones	138
Referencias Bibliográficas	139



## Lista de Figuras

	<b>pág.</b>
Figura 1. Abscisas de la mano	27
Figura 2. Dedo	29
Figura 3. Nombres comunes de los dedos	29
Figura 4. Movilidad de la mano	31
Figura 5. Movimientos del dedo pulgar	32
Figura 6. Huesos de la mano y antebrazo derecho (vista anterior)	33
Figura 7. Articulaciones de muñeca: radio carpiana (muñeca), intercarpianas, carpometacarpianas e intermetacarpianas vista coronal	33
Figura 8. Ejes de la muñeca en el movimiento	34
Figura 9. Tipos de movimiento en la muñeca y grados de libertad	35
Figura 10. Músculos superficiales del brazo	37
Figura 11. Señales electromiografías para distintos movimientos	40
Figura 12. Tipos de EMG	45
Figura 13. Sensores AgCl modificados	46
Figura 14. Distancia de los sensores sEMG	47
Figura 15. Esquema de puntos de conexión del Sensor muscular Myoware AT 04 001.	48
Figura 16. Sensor Myo	49
Figura 17. Amplificador de instrumentación INA106	50
Figura 18. Amplificador TL 074	50
Figura 19. Rectificación de la señal	51
Figura 20. Esquema de la neurona artificial	56
Figura 21. Perceptrón de dos capas.	58

Figura 22. Red neuronal artificial feed-forward de una sola capa oculta	59
Figura 23. RNA Hopfield con cuatro nodos	60
Figura 24. Red neuronal Convolutacional de dos capas ocultas.	61
Figura 25. Aplicación personalizada con una GUI en MATLAB	64
Figura 26. RAW EMG vs EMG	74
Figura 27. Dimensiones del sensor	75
Figura 28. Esquema de conexión de salida y polarización Sensor muscular	77
Figura 29. Diagrama de bloques de conexión de los equipos físicos desde el sensor al PC.	77
Figura 30. Programación básica del Arduino Micro para lectura de la señal del Sensor	78
Figura 31. Señales graficadas (movimiento del puño) en el IDE Arduino V1.8.10	79
Figura 32. Ubicación de los músculos presentes en la región del antebrazo	80
Figura 33. Posición final del sensor	81
Figura 34. Captura de pantalla de Add-On Explorers de Matlab para la instalación de paquetes de soporte de Hardware	82
Figura 35. Código 1	82
Figura 36. Diagrama de bloques, etapas de desarrollo del proyecto	83
Figura 37. Captura de pantalla sesión inicial del Script para lectura y almacenamiento de la señal Electromiográfica. Software MATLAB	84
Figura 38. Captura de pantalla de segunda sesión del Script (LecturaMyoWare.m) de la señal Electromiográfica. Software MATLAB	85
Figura 39. Captura de pantalla Representación gráfica realizada gracias al bucle For. (LecturaMyoWare.m) señal Electromiográfica. Software MATLAB	86
Figura 40. Captura de pantalla Representación gráfica realizada gracias al bucle while (LecturaMyoWare.m) señal Electromiográfica. Software MATLAB	87

Figura 41. Variables creadas a partir del script (LecturaMyoWare.m) señal Electromiográfica Software MATLAB	89
Figura 42. Graficas de 22 señales EMG acondicionadas. Software MATLAB	90
Figura 43. Histogramas de las señales EMG acondicionada. Software MATLAB	91
Figura 44. Gráfica Vector Smovitotales con etiquetas de identificación. Software MATLAB	91
Figura 45. Código 2	93
Figura 46. Integración trapezoidal	94
Figura 47. Código 3	95
Figura 48. Código 4	96
Figura 49. Diseño de filtro para la señal EMG acondicionada mediante la App Filter Designer. Software MATLAB	97
Figura 50. Función de Filtrado generada a partir App Filter Designer. Software MATLAB	98
Figura 51. Gráfica de vector “apertura índice” sin filtrar (naranjado) y filtrada (Color azul). Software MATLAB	98
Figura 52. Código comando pwelch aplicado a toda la señal filtrada Software MATLAB	100
Figura 53. Estimación espectral de potencia de Welch aplicada a toda la señal filtrada Software MATLAB	101
Figura 54. PSD (Power Spectral Density) de Welch comparación entre dos movimientos Software MATLAB	102
Figura 55. Código comando pwelch aplicado a toda la señal filtrada Software MATLAB	103
Figura 56. Código 5	105
Figura 57. Código 6	106
Figura 58. Tabla de características extraídas de la señal Electromiográfica acondicionada	

movimientos. Software MATLAB	108
Figura 59. Tabla de características cargada y elección de validación en App Classification Learner. Software MATLAB	109
Figura 60. Tabla de características cargada y elección de validación en App Classification Learner Software MATLAB	110
Figura 61. Ventada Data browser con los diferentes algoritmos funcionales para identificación de App Classification Learner. Software MATLAB	111
Figura 62. Ilustra estas definiciones, con + indicando puntos de datos de tipo 1 y – indicando puntos de datos de tipo – 1.”. Software MATLAB	112
Figura 63. Matriz de confusión del desempeño del algoritmo, Linear SMV (modelo 1.7) con los 22 movimientos iniciales. Software MATLAB App Classification Learner	113
Figura 64. Toolbox nntool con los datos cargados Matriz Crtx(Input Data), Vector Identfx(Target Data). Software MATLAB App	114
Figura 65. Ventana de creación de la Red neuronal de datos Crtx (Input Data), Vector Identfx (Target Data) Software MATLAB App	116
Figura 69. Variables de entrada tomadas por el comando learnngd Software MATLAB App	118
Figura 70. Gráfico, comando y símbolo de la función de transferencia sigmoidea tangente hiperbólica.	119
Figura 71. Red neuronal personalizada con 26 entradas y 10 neuronas en la capa oculta. Software	120
Figura 72. Entrenamiento de la Red neuronal artificial con las características definidas anteriormente. Software MATLAB, nntool	121
Figura 73. Grafica error cuadrático medio vs número épocas. Software MATLAB, nntool	122
Figura 74. Grafica parámetros de progreso Vs número de épocas Software MATLAB,	

nntool	122
Figura 75. Grafica de $1 \cdot \text{vector objetivo} + \text{error}$ vs. Vector objetivo. Software MATLAB, nntool	123
Figura 76. Datos del error según la respuesta obtenida de la validación. Software MATLAB, nntool	123
Figura 77. Código 7	126
Figura 78. Inicio guide	128
Figura 79. Inicio de la GUI	128
Figura 80. Ventana de trabajo	130
Figura 81. Paleta de componentes	131
Figura 82. Conjunto de opciones	132
Figura 83. Entorno de la interfaz	133
Figura 84. Interfaz en funcionamiento	134
Figura 85. Software Blender	135

## Lista de Tablas

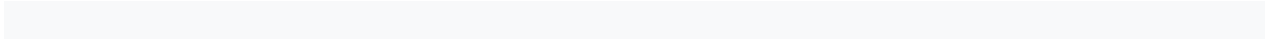
	<b>pág.</b>
Tabla 1. Músculos ventrales, dorsales y sus movimientos	38
Tabla 2. Músculos laterales y sus movimientos	38
Tabla 3. Ventajas y desventajas Arduino	52
Tabla 4. Tipos de tarjetas	53
Tabla 5. Convertidores para selección	70
Tabla 6. Características del sensor	73
Tabla 7. Especificaciones eléctricas	75
Tabla 8. Equipos y materiales	76
Tabla 9. Planos musculares de la región del antebrazo	80
Tabla 10. Métodos de evaluación de señales electromiográficas	88
Tabla 11. Orden de operaciones	107

## **Resumen**

Este proyecto realizó una identificación de movimiento a partir de señales electromiográficas presentes en el antebrazo, utilizando redes neuronales artificiales en Matlab. Para ello, se realizó una investigación tipo investigativo y desarrollo (proyecto I+D) ya que se complementa de manera dinámica el incremento del conocimiento y engloba tres actividades: investigación básica, investigación aplicada y desarrollo experimental. La recolección de información se basó en libros especializados de acceso público que se encuentran en la Red. Se logró, identificar el movimiento a partir de las señales electromiográficas que se presentan en los músculos del antebrazo humano, mediante la implementación de redes neuronales artificiales (RNA) utilizando el software MATLAB. Se elaboro el estado del arte. Se llevó a cabo la selección de los circuitos eléctricos, electrónicos o dispositivos correspondientes para realizar la toma de señales electromiográficas. Seguidamente, se implementó la interfaz que permita visualizar de manera didáctica los movimientos parciales producidos en la mano y los dedos, dependiendo de las señales electromiográficas características. Posteriormente, se validó el correcto funcionamiento de la interfaz. Finalmente, se socializaron los resultados del proyecto con la comunidad académica.

## **Abstrac**

This project carried out an identification of movement from electromyographic signals present in the forearm, using artificial neural networks in Matlab. For this, a research-type research and development (R&D project) was carried out, since it dynamically complements the increase in knowledge and encompasses three activities: basic research, applied research and experimental development. The collection of information was based on specialized public access books found on the Internet. It was possible to identify movement from the electromyography signals that are presented in the muscles of the human forearm, through the implementation of artificial neural networks (RNA ) using MATLAB software. The state of the art was elaborated. The selection of the corresponding electrical, electronic circuits or devices to carry out the electromyography signal taking was carried out. Subsequently, the interface was implemented to didactically visualize the partial movements produced in the hand and fingers, depending on the characteristic electromyographic signals. Subsequently, the correct operation of the interface was validated. Finally, the results of the project were socialized with the academic community.





## Introducción

La humanidad a través del tiempo ha implementado los avances tecnológicos con el fin de obtener una mejor calidad de vida, en cuanto al campo de la medicina es completamente común la utilización de equipos médicos que permitan diagnosticar, curar o tratar una enfermedad, de igual manera con el fin de suplir todo tipo de necesidades, logra incorporar la aplicación de nuevas tecnologías para la construcción de prótesis de miembros amputados de manera congénita, traumática o quirúrgica; En este campo se presentan distintas variantes, las que se usan solo con el fin de mejorar estéticamente la apariencia del paciente, las que permiten una movilidad parcial y limitada mediante el uso de técnicas mecánicas, también se encuentran las prótesis mioeléctricas y las híbridas que tratan de unir todas estas características.

Las mioeléctricas están basadas en la toma de señales electromiografías (EMG), producidas por los músculos cuando se contraen o relajan, estas señales permiten operar el entorno para el cual se adecuaron ya sea virtual o físico; antes de que el paciente pueda utilizar satisfactoriamente dicha prótesis es necesario someterlo a un entrenamiento para su manejo, tomando en cuenta el alto costo de estas, el aprendizaje debe llevarse a cabo de una manera segura para ambas partes evitando daños en los equipos y brindando la mayor comodidad.

Debido a ello este trabajo está dirigido a la investigación e identificación de señales encargadas del movimiento de la mano, a partir de las señales EMG producidas en los músculos del antebrazo.

## **1. Problema**

### **1.1 Título**

IDENTIFICACION DE MOVIMIENTO APARTIR DE SEÑALES ELECTROMIOGRAFICAS PRESENTES EN EL ANTEBRAZO, UTILIZANDO REDES NEURONALES ARTIFICIALES EN MATLAB.

### **1.2 Planteamiento del Problema**

Las personas que presentan amputaciones ya sea de tipo congénitas, traumáticas o quirúrgicas de la mano o parte de ella y que decidan la utilización de una prótesis mioeléctrica, antes de poder operarla, deben someterse a un entrenamiento exhaustivo para lograr un manejo satisfactorio de la misma, debido a los altos costos que presenta dicha tecnología y dependiendo de su funcionalidad, se evidencia que el aprendizaje debe realizarse de manera segura, didáctica y efectiva garantizando que las señales electromiografías producidas por los músculos de un individuo sean identificadas de manera coherente al movimiento que produce.

### **1.3 Formulación**

¿La aplicación de redes neuronales artificiales (RNA) utilizando el software de MATLAB, permiten identificar de manera adecuada y coherente las señales electromiografías producidas por los músculos presentes en el antebrazo, que intervienen en el movimiento de la mano y los dedos?

## 1.4 Objetivos

**1.4.1 Objetivo general.** Identificar el movimiento a partir de las señales electromiografías que se presentan en los músculos del antebrazo humano, mediante la implementación de redes neuronales artificiales (RNA) utilizando el software MATLAB.

**1.4.2 Objetivos específicos.** Elaborar el estado del arte.

Seleccionar los circuitos eléctricos, electrónicos o dispositivos correspondientes para realizar la toma de señales electromiografías.

Implementar la interfaz que permita visualizar de manera didáctica los movimientos parciales producidos en la mano y los dedos, dependiendo de las señales electromiografías características.

Validar el correcto funcionamiento de la interfaz.

Socializar resultados del proyecto con la comunidad académica.

## 1.5 Justificación

Para la elaboración de innumerables equipos médicos que permitan el diagnóstico de pacientes y para la implementación de prótesis mioeléctricas las señales electromiografías constituyen una parte característica y fundamental en dichos avances, debido a que son señales eléctricas producidas por la contracción y relajación de los músculos. Las prótesis que son implementadas con esta tecnología buscan brindar al usuario una mayor cantidad de movimientos, lo que incrementa su costo de manera exponencial dependiendo también de la funcionalidad; quienes acceden a estos dispositivos deben tener un aprendizaje que permita al usuario o paciente tener un control completo y operarla de manera correcta.

El presente proyecto busco contribuir con el entrenamiento muscular previo de manera segura didáctica y efectiva, de personas que presentan amputaciones ya sea de tipo congénitas, traumáticas o quirúrgicas de las manos, basándose en la identificación de las señales que se producen en los músculos del antebrazo al realizar movimientos naturales como la apertura y cierre, movimientos de los dedos entre otros. Implementando el modelo computacional de redes neuronales artificiales, puesto que estos son más viables en las áreas donde la detección de las respuestas con respecto a las características iniciales es difícil de expresar con los diferentes tipos de programación convencional. Se realizó una interfaz gráfica que permita observar de manera análoga los movimientos en los dedos y la mano, la variación de las posiciones dependerá de las señales medidas.

En cuanto a la medición de las señales, acondicionamiento y comunicación con la computadora, se realizó una relación costo-beneficio de los dispositivos eléctricos o electrónicos utilizados para este propósito.

El desarrollo del proyecto fue un instrumento de investigación donde se aplicaron conocimientos y habilidades teórico-prácticas adquiridas en el transcurso de la carrera contribuyendo así, al mejoramiento de la calidad educativa del programa de ingeniería electromecánica.

## **1.6 Limitaciones**

La realización de este proyecto se limitó a los recursos existentes como son equipos electrónicos de medición y pruebas presentes en los laboratorios del departamento de electricidad y electrónica asignados a los alumnos en desarrollo de proyecto de grado de la Universidad Francisco De Paula Santander.

La utilización de software estuvo condicionada a los programas con licencia ya sea de tipo comercial o educativa con los que contaba en el departamento de E&E, como es el caso de MATLAB 2017 b y otros programas de licencias GNU (software libre).

La adquisición de dispositivos eléctricos y electrónicos esenciales para la realización del proyecto como son los electrodos, sensores, el convertidor Análogo/digital entre otros, se restringió a los recursos aportados por los integrantes del proyecto y su director.

Además, este proyecto quedo limitado a la medición e identificación de algunas señales electromiografías de buena apreciación, generadas por los músculos que están presentes en el antebrazo como son: El extensor de los dedos, el Flexor y el extensor del carpo, entre otros, permitiendo realizar una analogía entre las señales y algunos de los movimientos realizados por la mano y los dedos.

## **1.7 Delimitaciones**

**1.7.1 Delimitación temporal.** Este proyecto se realizó durante el segundo semestre del 2018 a partir de la aprobación del anteproyecto y continuó durante el año 2019.

**1.7.2 Delimitación de espacio tiempo.** Como se muestra a continuación:

**Espacio geográfico:** Se desarrolló en las instalaciones de la universidad Francisco de Paula Santander, en los laboratorios correspondientes del departamento de electricidad y electrónica que fueron asignados para los alumnos que se encontraron realizando proyecto de grado.

Tiempo: 12 meses.

## 2. Marco Referencial

### 2.1 Antecedentes

Se presenta a continuación el resultado de la búsqueda de estudios previos que son de gran utilidad y contribución para el tema de identificación de movimiento a partir de señales electromiografías presentes en los músculos del antebrazo, mediante la implementación de redes neuronales artificiales con MATLAB y su visualización en una interfaz gráfica didáctica.

Ayoroa & Kang (2016). *“Propuesta de implementación de prótesis mioeléctrica de bajo costo para miembro superior bajo codo con entorno virtual de entrenamiento”*. Universidad Nacional del Este, ciudad del este, alto Paraná, Paraguay. Ante la realidad de amputados de antebrazo a nivel de bajo codo en Paraguay, se presenta la problemática de necesidad de prótesis. Al adquirir la prótesis, el paciente pasa por un proceso de adaptación tras el cual puede desistir del uso de esta por motivos tales como excesivas expectativas con respecto a la funcionalidad. En el caso de prótesis robóticas que utilizan sensores mioeléctricos, un entrenamiento basado en un entorno virtual posibilita al paciente entrenar sus músculos para interactuar con los sensores. Además, el entorno virtual posibilita experimentar con anticipación las funciones que obtendrá antes de poseer la prótesis física. Este artículo presenta un entorno virtual para entrenamiento de prótesis robótica mioeléctrica de mutilación de antebrazo con muñón de bajo codo, que provee las funciones manuales: empuñamiento, pinza lateral y pinza distal.

Este artículo proporciona información sobre algunos dispositivos electrónicos para acondicionar señales electromiografías y software libre como es el motor de videojuego Unity para desarrollo de interfaz gráfica.

Haro & Román (2017). “*Diseño y construcción de un prototipo de prótesis mioeléctrica para desarticulación de muñeca*”. Universidad de las Fuerzas Armadas. Sangolquí, Ecuador. Este proyecto se basa en la necesidad de realizar la investigación en el campo de las prótesis mioeléctricas y cuyo principal objetivo es construir un prototipo de prótesis mioeléctrica de mano que posea un equilibrio entre factores primordiales como funcionalidad, estética y asequibilidad. Entre las ventajas de este tipo de prótesis se tiene que proveen mayor versatilidad de movimiento y permiten una rehabilitación muscular; ventajas a las que se le sumaran asequibilidad y personalización que serán características propias del prototipo en desarrollo. Además, el prototipo fue diseñado para satisfacer las necesidades de un usuario en específico. Esto con el fin de alcanzar el mayor equilibrio posible entre los factores mencionados anteriormente, finalmente el prototipo es probado por el usuario con el fin de evaluar su desempeño y encontrar mejoras para el desarrollo de futuros prototipos.

El anterior proyecto logro proporcionar gran información sobre prótesis mioeléctricas, historia, funcionamiento, aplicabilidad, también desempeño y construcción. De igual manera información característica los movimientos realizados por la muñeca y los dedos.

Milán, Vásquez, Choquehuanca & Achanccaray (2015), “*Adquisición, parametrización y clasificación de patrones de señales mioeléctricas para el control de movimiento de un exoesqueleto que asiste la bipedestación*”. En el proyecto se busca desarrollar una interface para la adquisición, parametrización y clasificación de patrones de las señales mioeléctricas de las extremidades inferiores con el objetivo de controlar los actuadores de un exoesqueleto que le permita al usuario/paciente, levantarse de una silla. El proyecto fue dividido en cuatro etapas: Registro y digitalización, que consiste en la adquisición de las señales mioeléctricas que intervienen durante el movimiento de bipedestación mediante electrodos para, posteriormente,

almacenarlos en una base de datos Parametrización, que consiste en la extracción de características, en el dominio de tiempo y la frecuencia, de las señales mioeléctricas previamente adquiridas. Reconocimiento y clasificación de patrones, que consiste en el modelamiento y entrenamiento de una red neuronal adecuada para la correlación entre la señal adquirida y el correspondiente movimiento del proceso de bipedestación. Control basado en detección de la intención de movimiento; dado que algunos pacientes tienen muy reducida movilidad, incluso para iniciar movimientos diarios comunes, el control del exoesqueleto tiene que estar basado en la intención de la acción, por tanto, las señales mioeléctricas adquiridas y procesadas serán las que generen las órdenes para el movimiento de los actuadores del exoesqueleto.

Cabrera & Montes (2012) *“Obtención y análisis de señales electromiografías de las articulaciones tibio femoral y femororotuliana aplicada a la detección y rehabilitación de problemas musculares en el movimiento de la rodilla”*. Este proyecto se enfoca principalmente en la superación de lesiones de rodilla investigando la anatomía muscular de las articulaciones de la misma, obteniendo las señales electromiografías que intervienen en los movimientos de flexión y extensión procesando estas señales para el diseño de una órtesis de rodilla siglas KO (Knee Orthoosis) activa, Una ortesis es un apoyo u otro dispositivo externo (aparato) aplicado al cuerpo para modificar los aspectos funcionales o estructurales del sistema músculo esquelético. Esta permitirá realizar un control automatizado de los movimientos y un numero de repeticiones, garantizando de este modo una mayor precisión en las técnicas de rehabilitación, permitiendo además al fisioterapeuta, avalar su diagnóstico con el estudio electromiográfico a nivel de la rodilla.

Mendoza, Peña, Muños & Belandia (2013). *“Procesamiento de señales provenientes del habla sub vocal usando Wavelet Packet y Redes Neuronales”*. Este artículo presenta los



resultados obtenidos de la grabación, procesamiento y clasificación de palabras en el idioma español mediante el análisis de señales de habla subvocales. La base de datos procesada tiene seis palabras (adelante, atrás, derecha, izquierda, iniciar y detener). En este trabajo, las señales (EMG) se detectaron con electrodos de superficie colocados en la superficie de la garganta y se adquirieron con una frecuencia de muestreo de 50 kHz. El acondicionamiento de la señal consistió en: la ubicación del área de interés mediante el análisis de energía y el filtrado mediante la transformada de wavelet discreta. Finalmente, la extracción de características se realizó en el dominio de la frecuencia de tiempo utilizando paquetes de Wavelet y técnicas estadísticas para la creación de ventanas. La clasificación se realizó con una red neuronal de propagación inversa cuya capacitación se realizó con el 70% de la base de datos obtenida. La tasa de clasificación correcta fue del  $75\% \pm 2$ .

Quinayás & Gaviria (2015). “*Sistema de identificación de intención de movimiento para el control mioeléctrico de una prótesis de mano robótica*”, *Ing. Univ.*, vol.19, no.1, pp.27-50, Ene. Jun, 2015. <http://dx.doi.org/10.11144/javeriana.iyu19-1.siiim>. Este artículo presenta un sistema embebido que detecta en tiempo real la intención de movimiento para el control de un prototipo de prótesis de mano robótica. El trabajo muestra que usando características temporales de cálculo simple es posible obtener un agrupamiento de vectores lo suficientemente discriminante como para que se puedan usar clasificadores de patrones muy simples. Así, en este trabajo se propone un clasificador basado en la mínima distancia al centroide de los grupos que caracterizan los movimientos que se van a identificar, modificando el conocido algoritmo k vecinos más próximos para sacar mayor provecho de la fase de entrenamiento del clasificador en la fase de clasificación y obtener respuesta en tiempo real. Se presentan resultados de clasificación de intención de movimiento, obtenidos usando el porcentaje de éxito como medida de efectividad, al

realizar pruebas sobre tres sujetos con músculos sanos. Los resultados experimentales muestran que el sistema puede ser efectivo para el control de ejecución de cuatro primitivas motoras de un prototipo de prótesis de mano robótica.

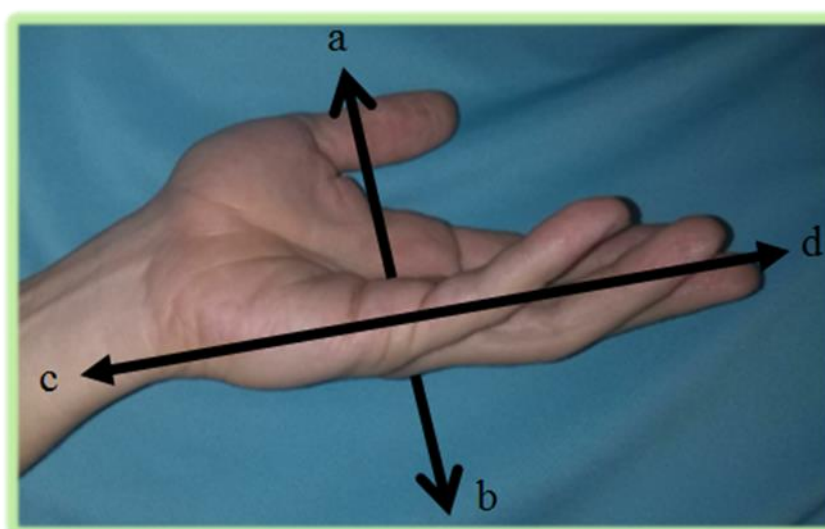
Reyes, Arias, Duarte & Loaiza (2015) *“implementación de FPGA de un clasificador de movimientos de la mano utilizando señales EMG”*. *Redes de ingeniería*, 6(1), 85-94. Este trabajo presenta el diseño e implementación de un clasificador de señales electromiográficas (EMG) para tres movimientos de la mano: flexión, extensión y cierre, usando dos músculos del antebrazo, palmar largo y extensor común de los dedos. El desarrollo comprende dos bloques principales, el hardware para la adquisición y adecuación de las señales EMG analógicas y el sistema de procesamiento para la identificación y clasificación del movimiento realizado; el sistema completo fue implementado en hardware usando un kit de desarrollo DE2-70 que cuenta con un FPGA Cyclone II de Altera. Para la extracción de características se implementó la Transformada Rápida de Fourier (FFT), para cada canal, a la cual se le calcularon técnicas de procesamiento como la varianza y el promedio. Finalmente, se establece un umbral de decisión para identificar el movimiento realizado. El tiempo de respuesta del sistema total fue de 17,7  $\mu$ s, obteniendo una tasa de identificación mayor al 87%.

## **2.2 Estado del Arte**

**2.2.1 Anatomía de la mano y el antebrazo humano.** El miembro superior es una de las extremidades más utilizada por las personas, con la cual se desempeña un rol social importante al momento de relacionarse con su entorno; además, es un miembro con el cual se desarrollan actividades necesarias en lo habitual -pensar, agarrar, sostener y trasladar objetos-, razones por las que se constituye en un órgano imprescindible en el cuerpo humano (España, 2016).

Centrándose en la estructura, disposición y forma se realiza una pequeña reseña de algunos músculos, huesos y movimientos desde la parte del codo hasta la yema de los dedos.

**2.2.2 La Mano.** Forma parte de las extremidades superiores del ser humano, localizada en los extremos del antebrazo, son prensiles, en estado completo o sano posee cinco dedos, cada una abarca desde la muñeca hasta la yema de los mismos. Son consideradas el principal órgano para la manipulación física del medio. (Kapandji, 2012).



a) Palmar, b) Dorsal, c) proximal, d) distal.

### Figura 1. Abscisas de la mano

**El esqueleto de la mano:** está formado por 27 huesos diferenciados en tres zonas ocho huesos cortos, cuyo conjunto constituye el esqueleto de la muñeca o carpo, cinco huesos largos que forman el metacarpo, por último, catorce huesos también largos llamados falanges que constituye el esqueleto de los dedos. (Jorge Washington Amangandi Aguilar. 2018).

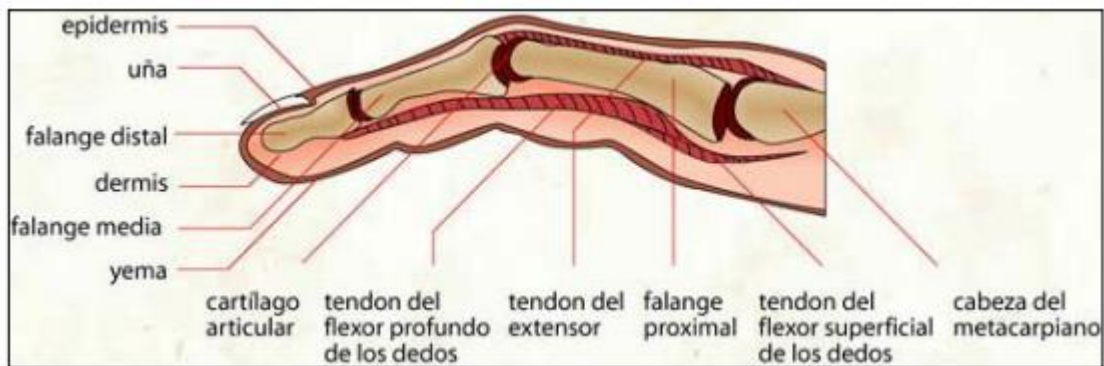
**Carpo:** Los ocho huesos que lo integran se hallan colocados en dos hileras superpuestas que, en conjunto, forman una masa, ósea convexa por su cara posterior y cóncava, transversalmente,

por la anterior; está última constituye una especie de canal por donde pasan los tendones de los músculos flexores de los dedos. La hilera superior se halla formada, procediendo de afuera adentro, por los siguientes huesos: escafoides, semilunar, piramidal y pisiformes, en la hilera inferior, procediendo en el mismo orden, se encuentra los siguientes: trapecio, trapezoide, hueso grande y hueso ganchudo. (Pró, 2012)

**Metacarpo:** Se halla formado por cinco huesos largos o huesos metacarpianos, que se designa como primero, segundo, etc., de afuera adentro y forman el esqueleto de la palma de la mano. Se articular por arriba con la segunda hilera de huesos del carpo y, por abajo, con las primeras falanges de los dedos. Presentan para su estudio caracteres comunes entre sí, y otros particulares, que dan personalidad propia a cada una. (Pró, 2012)

**Falanges:** Forman el esqueleto de los dedos, siendo en un numero de dos para el pulgar (falange y falangeta) y de tres para los otros cuatro (falange, falangina y falangeta). Son, como los anteriores, huesos largos, provistos, por tanto, de un cuerpo y dos extremidades. (Pró, 2012)

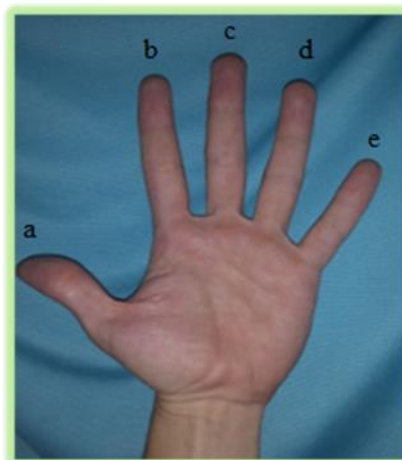
**Tendones de la mano:** Los tendones en las manos permiten el movimiento de la mano y dedos que se encuentran conectados directamente con los músculos de los brazos y se clasifican en tendones extensores y los flexores. El tendón extensor permite a las articulaciones de los dedos enderezarse o extenderse como de la zona de la muñeca. Los tendones flexores permiten realizar los movimientos de flexión como hacer un puño, agarrar los objetos y doblar los dedos y los pulgares (Amangandi, 2018).



**Figura 2. Dedo**

Fuente: Amangandi, 2018.

**2.2.3 Los dedos y su movimiento.** Son las extremidades que poseen tanto las manos como los pies, se presentan en los humanos y algunos animales, para el proyecto la importancia radica en el movimiento de los dedos de la mano que realiza un ser humano. Cada dedo es conocido por un nombre propio: Como muestra la **figura 3**.



a) Pulgar, b) índice, c) corazón, d) anular, e) meñique.

**Figura 3. Nombres comunes de los dedos**

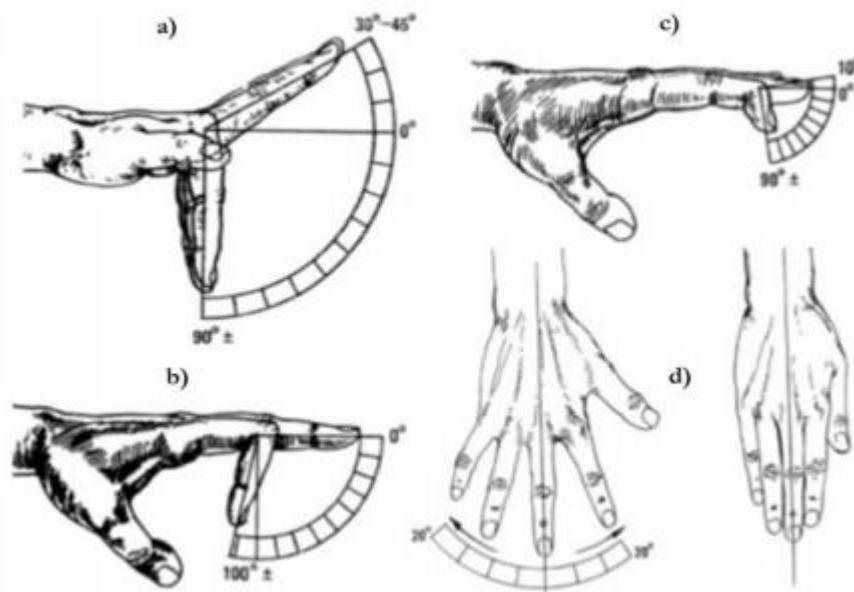
El pulgar (conectado al trapecio) está en el lado externo de la mano hallándose en posición anatómica, paralelo al brazo. El pulgar puede rotar fácilmente  $90^\circ$ , perpendicularmente a la palma, no como el resto de los dedos que solamente pueden rotar cerca de  $45^\circ$ . Una forma fiable de reconocer manos verdaderas en el resto de los animales (no humanos) es observar si poseen pulgares oponibles. Los pulgares oponibles se diferencian por poder oponerse al resto de los dedos en una acción muscular conocida como oposición. Los otros cuatro dedos de la mano se localizan en el borde exterior de la palma. Estos cuatro dedos pueden ser plegados hacia la palma, esto permite sostener objetos y además agarrar otros más pequeños (Kapandji, 2012).

Por tener los dedos y en especial del pulgar, la mano tiene numerosos movimientos en los cuales intervienen las articulaciones falángicas entre ellas y con los metacarpos.

Flexión-Extensión metacarpofalángica: el arco total de movimiento es de  $120^\circ$  a  $135^\circ$  y está dividido en  $90^\circ$  para la flexión y  $30^\circ$  a  $45^\circ$  para la extensión.

Flexión-Extensión Interfalángica proximal: un arco de movimiento total de  $100^\circ$  asociados únicamente para la flexión, en la normalidad no existe extensión de esta articulación.

Flexión-Extensión Interfalángica distal: un arco de movimiento total de  $90^\circ$  a  $100^\circ$  dividido en  $90^\circ$  para extensión y  $0^\circ$  a  $10^\circ$  para la extensión. (Jorge Washington Amangandi Aguilar. 2018)



**Figura 4. Movilidad de la mano**

Fuente: Barrera & Merchán, 2017.

a) Movilidad de la articulación metacarpofalángica, flexión y extensión, b) Movilidad de la articulación proximal interfalángica, flexión y extensión, c) Movilidad de la articulación distal interfalángica, flexión y extensión, d) abducción y aducción de los dedos.

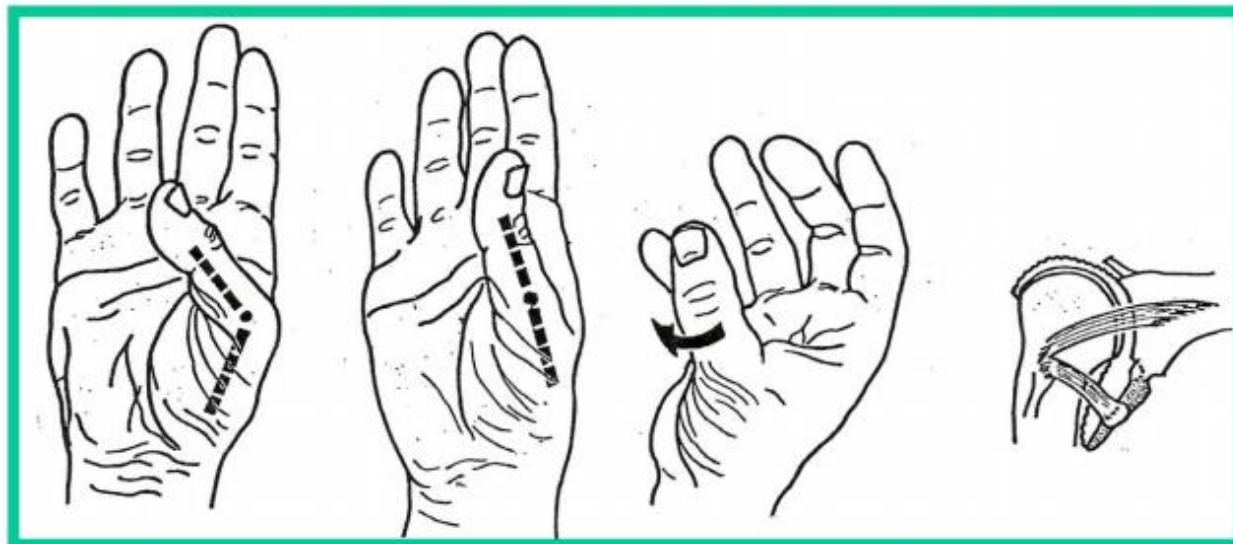
Flexión-extensión total del pulgar: En la normalidad, este movimiento será capaz de que toquemos la punta del meñique con la de nuestro pulgar si ambos están flexionados.

Flexión-extensión metacarpofalángica del pulgar: Describe un arco de  $50^\circ$  en flexión sin haber ángulo de extensión.

Flexión-extensión interfalángica de pulgar: Tiene un arco total de movimiento de  $90^\circ$  a  $110^\circ$  de movimiento dividiéndose en  $90^\circ$  para flexión y  $0^\circ$  a  $20^\circ$  para la extensión.

Abducción-aducción palmar del pulgar: Rango total de movimiento de 70° solo en flexión.

(Jorge Washington Amangandi Aguilar. 2018).



**Figura 5. Movimientos del dedo pulgar**

Fuente: Universidad de Zaragoza, 2009.

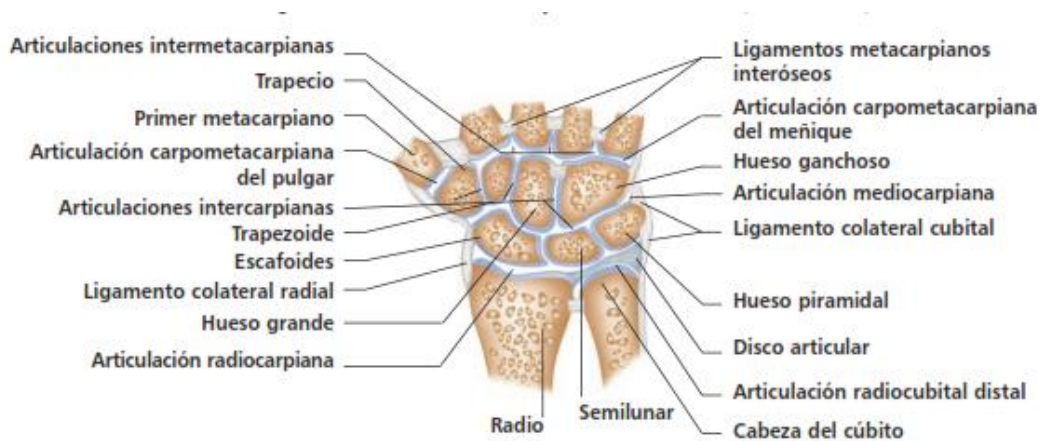
**2.2.4 Estructura ósea desde el antebrazo a los dedos.** La muñeca y la mano se componen de veintisiete huesos, numerosos ligamentos y muchos músculos y tendones, todo lo cual dota a los dedos de capacidades motoras finas. La muñeca y la mano alojan los ocho huesos del carpo, cuya fila proximal se compone del escafoides, el semilunar, el piramidal y el pisiforme, y se articula con el radio y el cúbito para crear la articulación radiocarpiana. Aquí es donde se producen las acciones principales de la muñeca; al ser una articulación condiloidea (elipsoidea), permite flexión, extensión, abducción y aducción. La combinación de estas cuatro acciones se llama circunducción (Amicivirtual, 2018).





**Figura 6. Huesos de la mano y antebrazo derecho (vista anterior)**

Fuente: Staugaard, 2014.



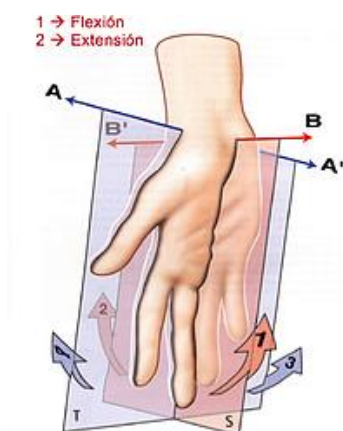
**Figura 7. Articulaciones de muñeca: radio carpiana (muñeca), intercarpianas, carpometacarpianas e intermetacarpianas vista coronal**

Fuente: Staugaard, 2014.

**2.2.5 Movimientos de la muñeca.** Los movimientos fundamentales de la mano son flexión y extensión y desviación radial y ulnar. Presenta movimientos que están mediados por las articulaciones que forman radio y cubito con los huesos de la mano.

Flexión-Extensión: el arco total es de  $170^\circ$  dividiéndose  $80^\circ$  para flexión y  $90^\circ$  para extensión.

Desviación: está puede ser radial o cubital y describen un arco total de  $50^\circ$  dividiéndose  $30^\circ$  para la desviación cubital y  $20^\circ$  para la desviación radial. (Jorge Washington Amangandi Aguilar, 2018).



**Figura 8. Ejes de la muñeca en el movimiento**

Fuente: Kapandji, 2012.

Eje transversal AA', Plano frontal. Flexión (1) la cara anterior o palmar de la mano se aproxima a la cara anterior del antebrazo; Extensión (2): la cara posterior o dorsal de la mano se aproxima a la cara posterior del antebrazo. Eje anteroposterior BB', Plano Sagital. (Kapandji A.I. 2012).

### 2.2.6 Tipos de movimiento. Como se muestra a continuación:



**Figura 9. Tipos de movimiento en la muñeca y grados de libertad**

Fuente: Kapandji, 2012.

Aducción o inclinación cubital (3): la mano se aproxima al eje del cuerpo y su borde interno [o borde cubital (el del meñique)], forma, con el borde interno del antebrazo.

Abducción o inclinación radial (4): la mano se aleja del eje del cuerpo y su borde externo [o borde radial (el del pulgar)] forma, con el borde externo del antebrazo.

En realidad, son movimientos naturales de la muñeca son movimientos combinados en torno a ejes oblicuos: Flexión / Aducción y Extensión / Abducción. (Kapandji A.I. 2012)

**Abducción-Aducción:**

La amplitud de movimientos se puede medir a partir de la posición anatómica, el eje de la mano, representado por el tercer metacarpiano, que es una prolongación del eje del antebrazo.

La amplitud de aducción o inclinación cubital es de 45°. Sin embargo, esta amplitud difiere según se considere el eje de la mano: en cuyo caso es de 30°; o el eje del dedo corazón: en cuyo caso es de 55°. Esto se debe a que la aducción de la mano se asocia con la aducción de los dedos. Es mayor en la supinación que en pronación no sobrepasa los 10°.

La amplitud de la abducción o inclinación radial no sobrepasa los 15°. (Kapandji A.I. 2012)

**Flexo-extensión:**

La amplitud de los movimientos se mide a partir de la posición anatómica: muñeca alineada, cara dorsal de la mano en la prolongación de la cara posterior del antebrazo.

La amplitud de la flexión activa es de 85°

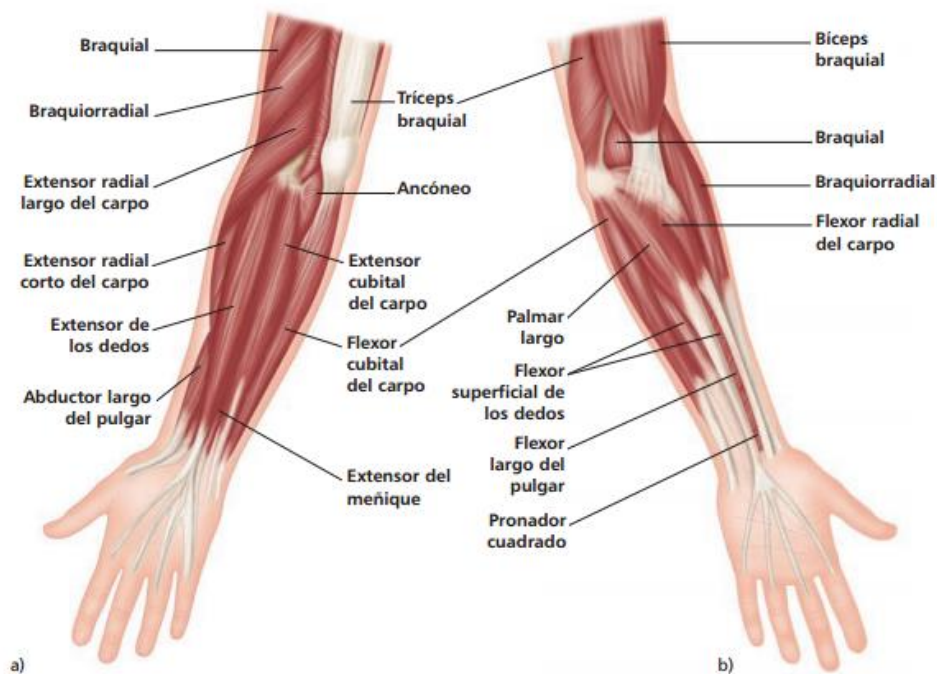
La amplitud de la extensión también es de 85°

La flexo-extensión es máxima cuando la mano no está ni en abducción ni en aducción

La amplitud de la flexión pasiva es mayor de 90° en pronación (100°).

La amplitud de la extensión pasiva es mayor de 90° tanto en pronación como en supinación (95°). (Kapandji A.I. 2012)

**2.2.7 Músculos del antebrazo y movimientos en los que intervienen.** En la región del antebrazo se presenta una gran cantidad de músculos, la figura 10 muestra los respectivos nombres que estos reciben, seguido se presentan los movimientos a los cuales se encuentran asociados.



a) vista superior, b) vista anterior.

### Figura 10. Músculos superficiales del brazo

Fuente Staugaard, 2014.

Para una mayor comprensión en la descripción del movimiento que producen se agrupan los músculos por ventrales (anteriores), dorsales (posteriores) y laterales dependiendo de la distinción del plano que ocupan.

**Tabla 1. Músculos ventrales, dorsales y sus movimientos**

VENTRALES		DORSALES	
MUSCULO	MOVIMIENTO	MUSCULO	MOVIMIENTO
Pronador Cuadrado	pronador del antebrazo	<b>supinador corto</b>	Supinador
Flexor Profundo De Los Dedos	Flexor de los dedos, flexor de la muñeca (indirectamente)	<b>Abductor largo pulgar</b>	Abductor pulgar, abductor radial de mano, flexor volar mano
Flexor Largo Del Pulgar	Flexor dedo gordo, flexor indirecto de la mano.	<b>Extensor corto pulgar</b>	Extensor pulgar, abductor radial mano, abductor pulgar.
Flexor Superficial De Los Dedos	Flexor dedos y mano indirectamente	<b>Extensor largo pulgar</b>	Extensor del pulgar, abductor radial de mano, flexor dorsal mano
Pronador Redondo:	Pronador de antebrazo y flexor de codo (aunque muy poco)	<b>Extensor índice</b>	Extensor de dedo índice, flexor dorsal mano.
Flexor Radial Del Carpo	Flexor volar de la mano, flexor radial y un poco pronador	<b>Extensor de dedos</b>	Extensor de dedos, flexor dorsal mano
Palmar Largo (O Menor)	Flexor volar mano, tensa aponeurosis palmar	<b>Extensor 5° dedo</b>	Flexor dorsal mano, Extensor 5° dedo, aductor de mano.
Flexor Cubital Del Carpo	Flexor volar mano, abductor cubital.	<b>Extensor cubital (posterior)</b>	Flexor dorsal, y abductor mano

Fuente: Amicivirtual, 2018.

Al observar la tabla es preciso denotar cuatro conceptos para un mayor entendimiento, La pronación se denomina a la rotación del antebrazo que permite situar la mano con el dorso hacia arriba. El movimiento contrario se denomina supinación.

Los aductores son unos tipos de músculos que mueven una parte del cuerpo para acercarla a su eje. Por consiguiente, los Abductores son los que las alejan.

**Tabla 2. Músculos laterales y sus movimientos**

LATERALES	
MUSCULOS	MOVIMIENTO
<b>Extensor corto radial del carpo (o 2° radial externo)</b>	Flexor dorsal mano, abductor radial mano.
<b>Extensor largo radial del carpo, (o 1r radial externo)</b>	Flexor dorsal mano, y abductor radial mano.
<b>Musculo braquiorradial (o supinador largo)</b>	Flexor del codo y supinador

Fuente: Amicivirtual, 2018.

**2.2.8 Señales Electromiográficas.** Para que exista una contracción muscular voluntaria, un potencial de acción nervioso tiene que viajar a lo largo de un nervio motor hasta sus terminaciones en las fibras musculares, donde induce un cambio en las concentraciones existentes de los iones sodio y potasio en el interior y exterior de la célula. Este fenómeno, conocido como despolarización de la membrana, genera un cambio en las cargas netas existentes en la célula conocido como potencial de acción muscular. (A. E. Armas Álvarez, A. K. López Castañeda.2015).

Las señales electromiográficas (EMG) son el registro muscular de la actividad eléctrica que genera tejido muscular, producidas durante el proceso de contracción y relajación de los músculos. Es decir, implícito que el electromiograma consiste en un tren aleatorio de potenciales de acciones que se registra extracelularmente y que son generados por las fibras musculares acompañadas de un trabajo mecánico. Dichas señales dan una idea de la actividad neuromuscular asociada a una contracción. Pero no indica la fuerza desarrollada por el músculo, sino cual musculo se activa más por el sistema nervioso central e indirectamente qué músculo presta una función más protagónica en una función o movimiento particular.

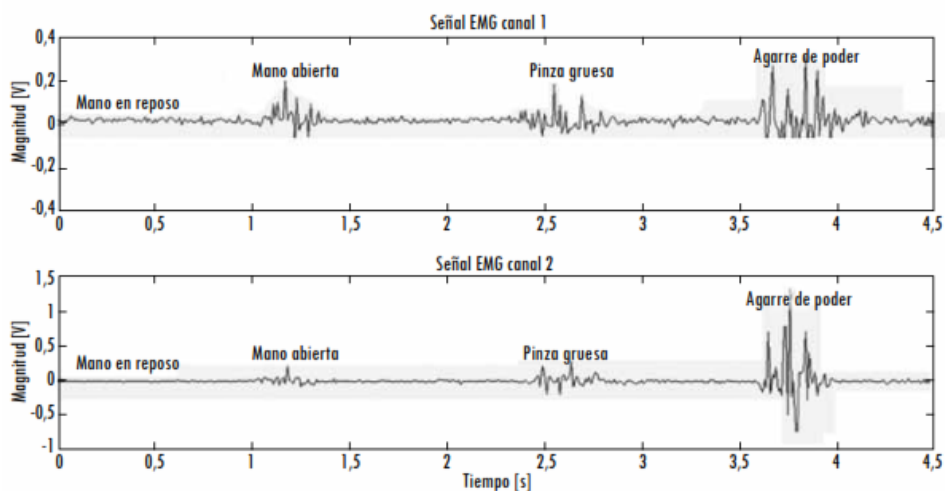
El electromiograma (EMG) es la sumatoria de los potenciales de acción de las unidades motoras de cierta área durante la contracción muscular. El EMG es la principal herramienta para el estudio de la actividad eléctrica muscular tanto en el área clínica para la detección de neuropatologías musculares, como en el área de investigación para el control de dispositivos activos como lo son las prótesis mioeléctricas. La señal electromiográfica posee parámetros físicos característicos que permiten que pueda ser registrada por un sistema físico, estos parámetros son:

El tiempo de duración de los potenciales de unidad motora, que van de 3 a 8 milisegundos, viéndose afectados por la fatiga y la edad.

Los valores normales de la amplitud generada por las fibras musculares son de entre 100 a 500 microvolts.

Los potenciales de acción de unidad motora poseen una frecuencia característica de 0 a 500 Hz.

Existen dos métodos de adquisición de la señal del EMG fundamentados en las propiedades físicas de la señal, estos son; la detección invasiva, que requiere de la inserción de una fina aguja directamente al músculo y técnicas no invasivas o también conocidas como electromiografía de superficie, que utiliza como medio transductor un electrodo adherido a la piel (Armas & López, .2015).



**Figura 11. Señales electromiografías para distintos movimientos**

Fuente: Quinayás & Gaviria, 2015.



**2.2.9 Dificultades para la toma de señales electromiográficas (EMG).** La señal EMG se puede cuantificar de varias formas. El método más simple es medir solo la amplitud. En este caso, se registra la amplitud máxima lograda con un determinado tipo de actividad muscular. Lamentablemente la amplitud es solo una indicación general de la magnitud de la actividad muscular y depende de la situación del electrodo de medida respecto al músculo.

Existen diversas dificultades que intervienen en una toma de señales electromiográficas satisfactorias para una correcta extracción e identificación de características que deben tenerse en cuenta como lo son:

**Variabilidad de los datos:**

Pocas de las variables que se pueden medir en el organismo humano son variables exactamente determinísticas de hecho, tales variables se deberían considerar como procesos estocásticos. Un proceso estocástico es una función temporal relacionada con otras variables de una forma no determinista. Las variables fisiológicas no se pueden considerar como valores estrictamente determinísticos, sino que se deben representar por algún tipo de distribución estadística o probabilística. En otras palabras, las medidas realizadas en un instante bajo un conjunto de condiciones determinado no serán necesariamente las mismas que las medidas similares realizadas bajo las mismas condiciones en otro instante. La variabilidad de un individuo a otro es todavía mayor. Aquí se deben emplear de nuevo métodos estadísticos para estimar las relaciones entre variables (Salas & Becerra, 2007).

**Amplificación:**

Toda la instrumentación EGM va dirigida a obtener una representación inteligible de los PAs musculares. Para ello los sistemas deben ser suficientemente sensibles porque las magnitudes bioeléctricas son muy pequeñas, siendo necesario amplificar la señal entre 50 y 250.000 veces. Las señales amplificadas son de entre 1 y 10 V, que es el rango en el que funcionan los circuitos electrónicos de los microprocesadores (Gila & Malanda, 2009).

**Ruido de componentes electrónicos:**

Son ruidos pequeños casi imperceptibles, los dispositivos que mayor propensión al ruido presentan son los que mayor cantidad de capacitancias e inductancias poseen, estos elementos forman parte de la circuitería de alimentación de los ordenadores, por esta razón es uno de los aspectos a considerar en la conexión del computador al cargador de batería. La frecuencia de la red se sobrepone a la señal EMG y dificulta la adquisición de la señal (Cabrera & Montes, 2012).

**Ruido ambiental:**

Todo dispositivo electro magnético genera ruido. La superficie del cuerpo humano generalmente se encuentra con bastante radiación electromagnética y resulta prácticamente imposible evitar la exposición a este tipo de ruido (Cabrera & Montes, 2012).

**Artefactos en movimiento:**

Principalmente este tipo de interferencia se presenta en el contacto existente entre la piel y el electrodo y aumenta debido a movimientos excesivos. Además, se presenta entre el cable del electrodo y el equipo que procesara la señal. Estos tipos de ruidos se presentan con mayor énfasis en las frecuencias existentes entre 0 Hz y 20 Hz. (Cabrera & Montes, 2012).

### **Inestabilidad de la señal:**

Debido a que se considera a la señal EMG de tipo aleatorio, las señales entre 0 Hz y 20 Hz se consideran inestables ya que son afectados por la naturaleza de las unidades motoras (Cabrera & Montes, 2012).

### **Tejido subcutáneo:**

Se considera al cuerpo humano como un buen conductor eléctrico, la conductibilidad de la piel varía dependiendo del tipo de tejido subcutáneo. Existe una variación de la amplitud en la señal EMG en condiciones normales y cuando el tejido adiposo subcutáneo es de grandes dimensiones lo que dificulta la toma de la amplitud normal de la señal (Cabrera & Montes, 2012).

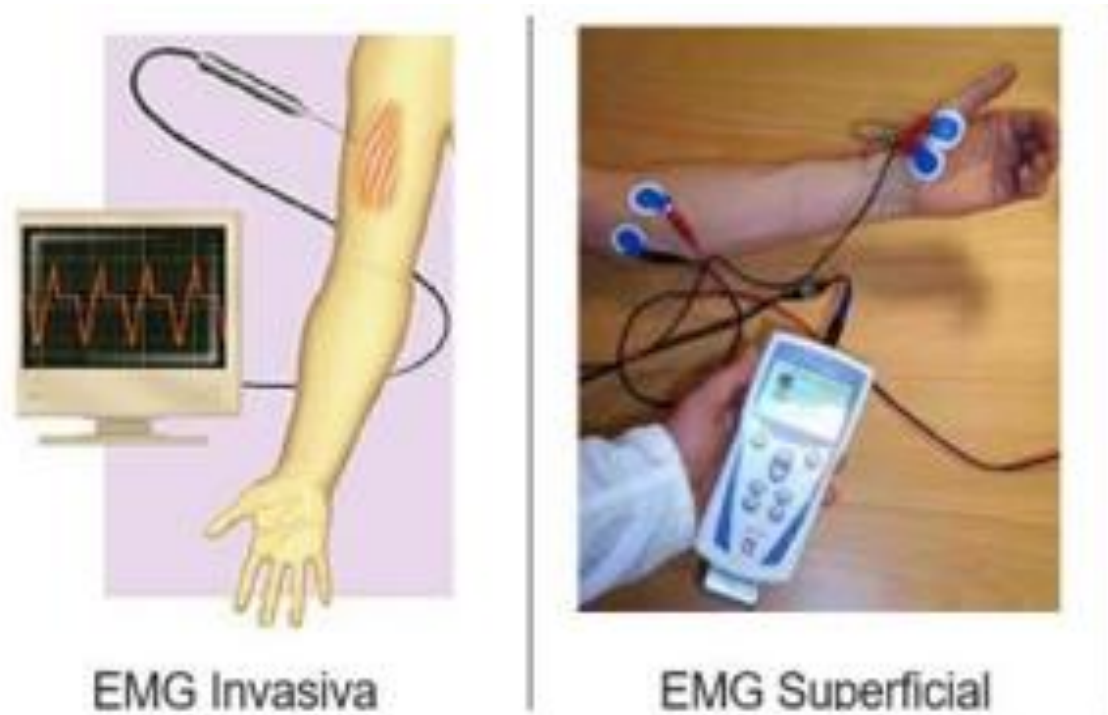
**2.2.10 Sensor muscular.** La medición de la actividad muscular a través del potencial eléctrico, tradicionalmente se ha utilizado para la investigación médica en el campo de trastornos neuromusculares. Debido a la incorporación de los circuitos integrados que aportan más potencia y tamaño reducido, se han empleado en prótesis, en robótica y otros sistemas de control (Advancer Technologies, 2015).

Como se desarrolla la historia de la Electromiografía, es el tipo de sensor quien clasifica el estudio de dicha señal.

**Sensores invasivos:** Los sensores invasivos se refiere a los electrodos que necesariamente deben penetrar la piel y estar adjunto al músculo en estudio, a pesar que este trabajo de investigación no está involucrado a este tipo de electrodos, es importante remarcar las diferencias para tener en claro el uso recomendado para este tipo de sensor. Entre los que destacan es el electrodo aguja, este tiene la capacidad de tener una amplia calidad para la visualizar PUM,

control de las propiedades de los UM (tasa de disparo, reclutamiento, etc.) y electromiografía clínica exploratoria. Por último, el otro de tipo de sensor es el electrodo tipo alambre fino recomendado para el estudio profundo de kinesiología, estudio profundo neurofisiológico, estudio limitado de las propiedades de las unidades motoras y un profundo estudio con grabación de músculos profundos.

**Sensores no invasivos:** El sensor no invasivo hace referencia de los sensores EMG que se ubica por encima de la piel, más no ingresado a través de este órgano; permitiendo un estudio sin agujas que implica un estricto uso. Eso no quita responsabilidad para el uso de los electrodos adhesivos, se debe tener en cuenta aun el riesgo eléctrico que puede tener el paciente en caso de no contar con un sistema de aislamiento o seguridad del sistema electrónico. La electromiografía superficial se recomienda para el estudio de la relación de tiempo de fuerza y la señal sEMG, estudios kinesiológicos superficial, estudio neurofisiológico superficial del músculo, estudios fisiológicos y hacer una interface entre un individuo y un aparato electromecánico (Alva, 2012).



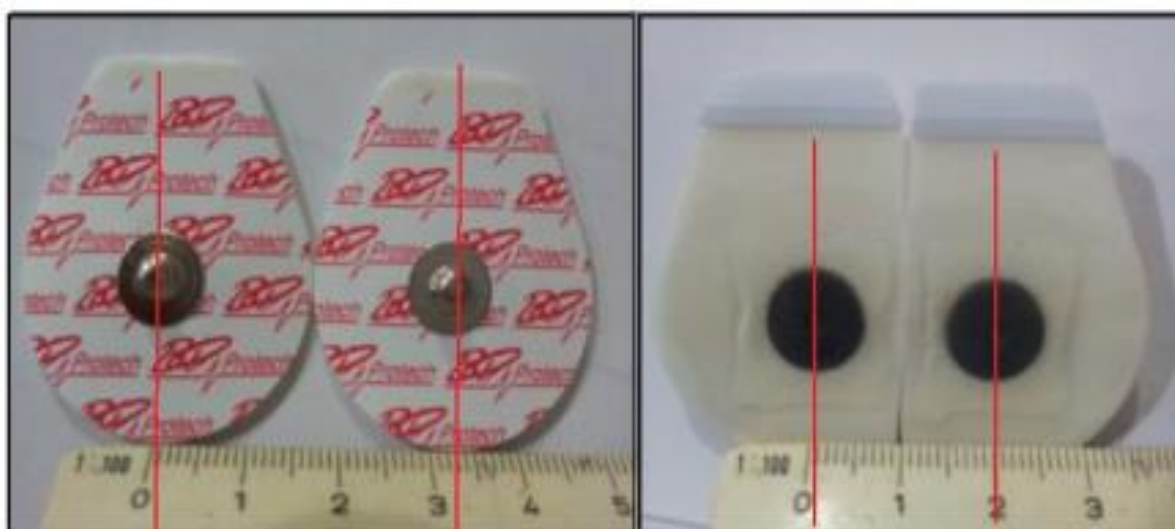
**Figura 12. Tipos de EMG**

Fuente: Huitzil, Pajaro & Ramírez, 2017.

**Electrodo:** aquel dispositivo que entra en contacto con la capa superficial de la piel, para lo cual se necesita el mejor contacto y este a la vez debe tener una baja impedancia. La composición del gel conductor es por lo más común es Ag/AgCl. Sin embargo, existen patentes de unas marcas de proveedores los cuales tiene una configuración o composición únicas las cuales tienen sus ventajas sobre otros o una aplicación específica. Para este caso, en su mayoría se ha utilizado los sensores ECG los cuales tienen la misma composición y tamaño, claro está que la distancia de los sensores tiene que estar lo más cerca posible para evitar ruido por diafonía, la cual comprende de la superposición de señales ajenas al estudio. Según las recomendaciones producidas por la European Concerted Action Surface EMG for noninvasive assessment of muscle, indica que las medidas de un electrodo en su diámetro deben ser menor o igual a 10mm y la separación entre

ambos electrodos debe ser menor o igual que 20 mm (Alva. 2012).

**Distancia entre sensores:** La distancia entre los sensores está comprendida en el tipo de topología de sistema electrónico de amplificación. En este caso de estudio, se da uso a los amplificadores diferenciales simples, el cual es característico de amplificadores instrumentales. Este tipo de configuración para pre-amplificación establece la ubicación y separación de los electrodos, veamos con más detalle la explicación teórica del amplificador diferencial simple y como acortando la distancia entre sensores permite disminuir notablemente la calidad de la señal.



A la izquierda distancia entre ambos electrodos aproximadamente a 3.3mm. A la derecha distancia aproximada a 2 mm siendo modificado los electrodos descartables

### Figura 13. Sensores AgCl modificados

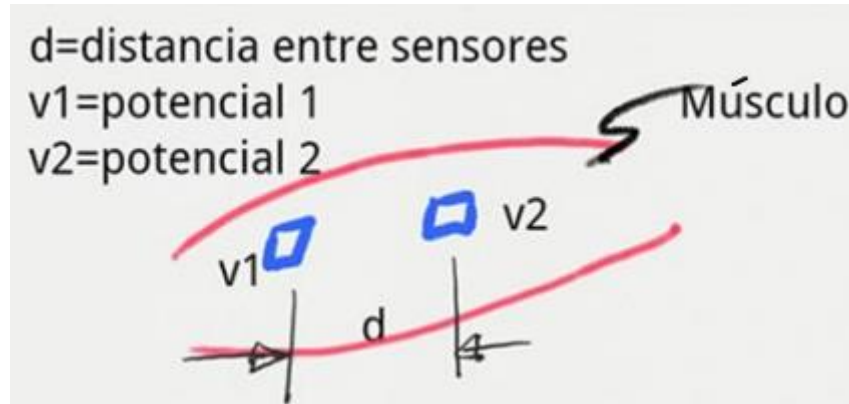
Fuente: Alva, 2012.

La distancia definida por  $d$  en el espacio entre los dos electrodos los cuales tiene dos potenciales eléctricos  $v_1$  y  $v_2$ , siendo captados por una actividad muscular. Al derivar la emanación de señal a través de la piel,  $v_1$  y  $v_2$  no son iguales dado que la velocidad en la que viaja la señal sEMG es un aproximado de 2 a 5 m/s en la cual fluye en un sentido y dirección. Sin

embargo, el ruido es común en ambos puntos, definiendo el ruido  $n$  la magnitud registrada en cada uno de los electrodos.

$$V_{out} = (v1 + n) - (v2 + n)$$

$$V_{out} = v1 - v2$$



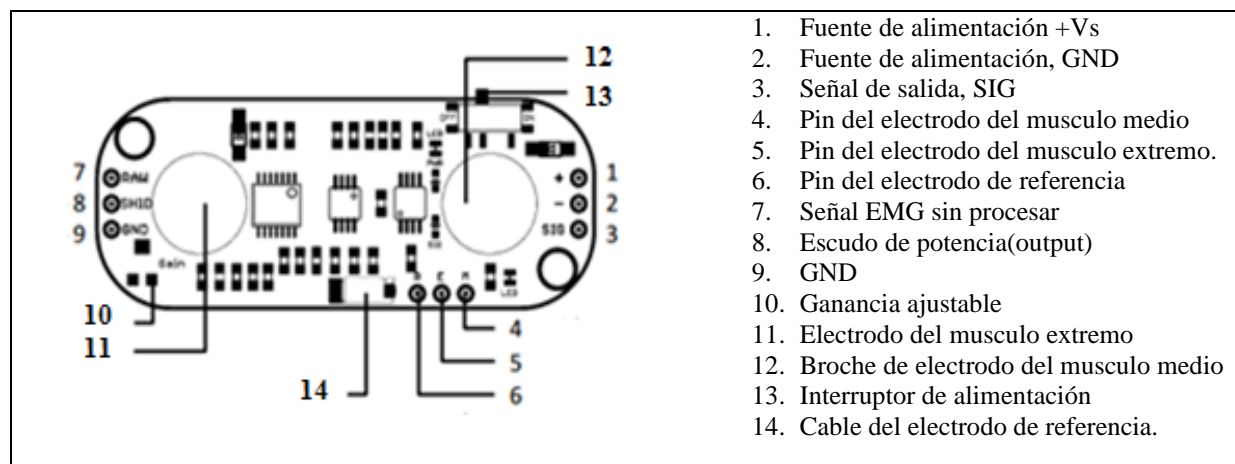
**Figura 14. Distancia de los sensores sEMG**

Fuente: Alva, 2012.

Como se ve siempre la salida de un amplificador diferencial hace una diferencia según ecuación 2.1, las magnitudes de las señales captadas y como  $n$  es común en ambos lados, este teóricamente eliminado según la ecuación, a pesar que esta configuración ayuda en mucho la calidad de la señal, existe un ruido extrínseco e intrínseco permanente los cuales, para el primer caso, puede ser eliminado con el uso de tecnología como, por ejemplo, valores altos de la relación CMRR. Este tipo de ubicación tiene como fin principal evitar la interferencia característica de la diafonía con señales provenientes de otras fuentes bioeléctricas.

Un claro ejemplo de sensor muscular es el sensor MyoWare, este presenta unas valiosas características, debido a que posee tanto los electrodos para medir la señal electromiografía, como un circuito integrado donde se realiza la etapa de acondicionamiento de dicha señal, contando con una parte de amplificación de la señal y el filtrado de la misma, permitiendo

conectarse directamente a un micro controlador o un convertidor analógico/digital, como ejemplos se puede presentar el Arduino o el RaspberryPi en sus distintas versiones dependiendo de la aplicación que desee realizarse.

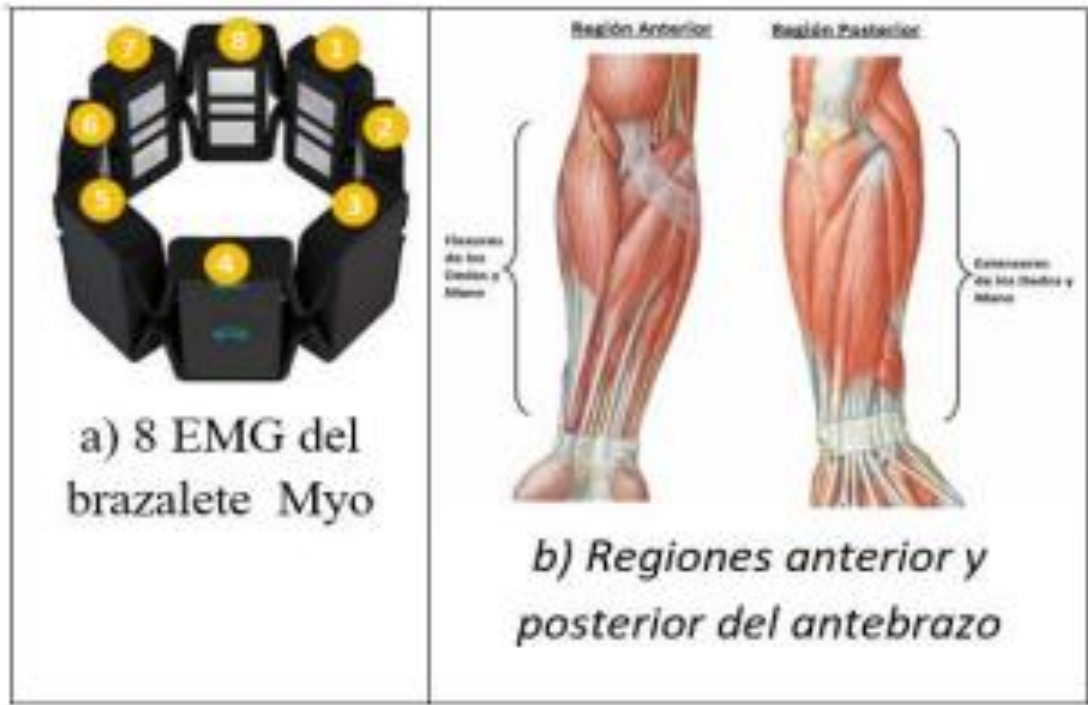


**Figura 15. Esquema de puntos de conexión del Sensor muscular Myoware AT 04 001.**

Fuente: Advancers Technologies, 2015.

Otra alternativa es el sensor Myo, es un brazalete de 8 sensores EMG diseñado para ser colocado en el antebrazo y obtener la activación eléctrica. Myo tiene un tamaño de 13 a 34 cm y un peso de 93 gramos. Es compatible con las plataformas de computadoras de escritorio Windows 7, 8, 10 y Mac OS X 10.8. Para dispositivos móviles en iPads con IOS 7.0 y nuevas versiones, dispositivos Android con versión 4.3 y actuales que soporten Bluetooth 4.0. El fabricante de Myo ofrece el programa Myo Connect para configurar aspectos de hardware, un software administrador de aplicaciones. Un SDK que cuenta con ejecutables, librerías, documentación, controladores y ejemplos dirigido a desarrolladores.

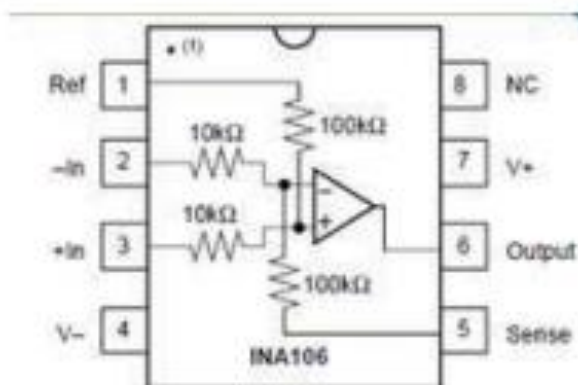




**Figura 16. Sensor Myo**

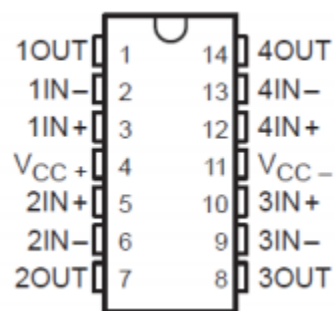
Fuente: Huitzil, Pajaro & Ramírez, 2017.

Además, también existe la posibilidad de obtener señales electromiográficas a través de la construcción por partes del proceso de recolección de señal, un ejemplo es el INA106, es un amplificador diferencial que mide y ligeramente amplifica las diferencias de voltaje muy pequeñas entre los dos electrodos que se colocan en el músculo.



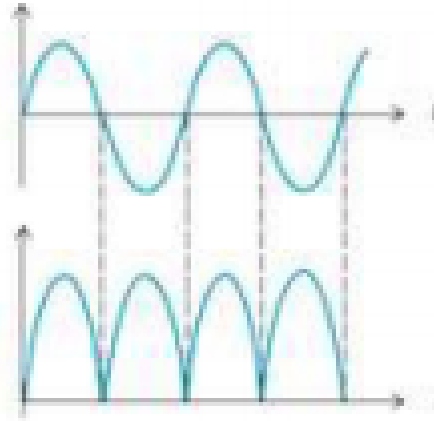
**Figura 17. Amplificador de instrumentación INA106**

Etapa de amplificación; En esta fase, se toman potenciales muy pequeños medidos en la fase de adquisición de señal y se amplifican con un TL074. Se añade un condensador para acoplar la señal de CA. El acoplamiento de Vca es útil en la eliminación de error de desplazamiento del Vdc en una señal.



**Figura 18. Amplificador TL 074**

Se rectifica la señal mediante un rectificador activo de onda completa. Se rectifica la parte negativa de la también. Suavizado-Amplificación de la señal; en esta última fase, se utiliza un filtro paso bajo para filtrar los picos de la señal para producir una señal sin ruido y luego conectarla con el sistema Arduino.



**Figura 19. Rectificación de la señal**

Fuente: Universidad Tecnológica de Pereira, 2012.

**2.2.11 Convertidor analógico/digital.** Los convertidores A/D son dispositivos electrónicos que establecen una relación o correspondencia uno a uno entre el valor de la señal de su entrada y la palabra digital obtenida en la salida. Esta relación se establece con una señal de referencia.

Para realizar la conversión de datos analógicos a digitales es posible emplear hardware de bajo costo y con acceso a Software libre, como lo son los RapsberryPi, el Arduino estos presentan distintas características dependiendo su versión y con fin de satisfacer las necesidades del usuario.

**2.2.12 Microcontrolador.** Los microcontroladores son circuitos compactos o integrados que pueden ser programados para la ejecución de las tareas asignadas. El circuito integra las tres funciones principales de las computadoras que son el procesador, la memoria y los puertos de conexiones entradas y salidas. Dependiendo de la necesidad de utilización estos varían en costo, velocidad de procesamiento, cantidad de puertos periféricos de entradas y salidas entre otros aspectos. Un ejemplo claro en estos tiempos de microcontrolador, con mucha aceptación por su

precio en el mercado para realización de proyectos de todo tipo es el Arduino.

**2.2.13 Arduino.** Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra, los que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (principalmente con cables DuPont). la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna, de esta forma el usuario final no se debe preocupar por las conexiones eléctricas que necesita el microcontrolador para funcionar, y puede empezar directamente a desarrollar las diferentes aplicaciones electrónicas que necesite.

**Tabla 3. Ventajas y desventajas Arduino**


<b>GENERALIDADES DE ARDUINO</b>	
<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
<p>Presenta al usuario una simplicidad al trabajar con microcontroladores.</p> <p>Son las placas que poseen más bajo costo comparadas con otras plataformas microcontroladoras.</p> <p>Su software es funcional en diversas plataformas como Windows, Macintosh OSX y Linux, sobre otros entornos de microcontroladores que están solo para Windows.</p> <p>Posee un entorno de programación basado en processing, es fácil para usuarios nuevos, bastante flexible para usuarios avanzados ya que su entorno se considera directo y sencillo.</p> <p>Software ampliable y de código abierto, es de distribución de licencia libre y puede ser adaptado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++.</p> <p>Hardware ampliable y de Código abierto. Los planos de los módulos están publicados bajo licencia <b>Creative Commons</b>, por lo que diseñadores de circuitos con experiencia y sin ella pueden hacer su propia versión del módulo.</p>	<p>No es un buen estándar para aprender a programar y beneficiarte del poder de los sistemas electrónicos.</p> <p>Muchos usuarios de nivel avanzado catalogan su lenguaje de programación como muy limitado, debido a que al hacerlo de manera sencilla no permite al usuario un aprendizaje óptimo en cuanto a procesos de programación.</p> <p>Los archivos generados que involucran muchos elementos quedan con un peso considerable lo que influye en la rapidez de la respuesta del programa. Esto es debido a que la programación no se realiza en lenguaje ensamblador y que se usan librerías produciendo retrasos en la ejecución en el orden de los microsegundos, para la adquisición de datos puede ser significativo.</p>

Fuente: Arduino. 2018, Iescamp. 2016.

Cuando hablamos de “Arduino” deberíamos especificar el modelo concreto, ya que se han fabricado diferentes modelos de placas Arduino oficiales, cada una pensada con un propósito diferente y características variadas (como el tamaño físico, número de pines E/S, modelo del microcontrolador, etc). A pesar de las varias placas que existen todas pertenecen a la misma familia (microcontroladores AVR marca Atmel), esto significa que comparten la mayoría de sus características de software, como arquitectura, librerías y documentación.

En la siguiente tabla se presentan diferentes tipos de tarjetas Arduino.

**Tabla 4. Tipos de tarjetas**

<b>TIPOS DE TARJETAS</b>		
 <p><b>Arduino Uno Rev3</b> Resolución: 10 bits</p> <p>Precio: 71.214 COP</p>	Microcontrolador	ATmega328P
	Tensión de funcionamiento	5V
	Voltaje de entrada (recomendado)	7-12V
	Pernos digitales de E / S	6-20V
	PWM Digital I / O Pins	14 (of which 6 provide PWM output)
	Clavijas de entrada analógica	6
	Corriente DC Pin de E / S	6
	Corriente DC para 3.3V Pin	20 mA
	Memoria flash	50 mA
	SRAM	32 KB (ATmega328P)
	EEPROM	2 KB (ATmega328P)
	Velocidad de reloj	1 KB (ATmega328P)
	LED_BUILTIN	16 MHz
	Longitud	68.6 mm
	Anchura	53.4 mm
Peso	13	
<p><b>Arduino Mega 2560 Rev3 10 bits</b> Resolución: 10 bits</p>	Microcontrolador	ATmega2560
	Tensión de funcionamiento	5V
	Voltaje de entrada	7-12V
	Pernos digitales de E / S	6-20V
	PWM Digital I / O Pins	54 (of which 15 provide PWM output)
	Clavijas de entrada analógica	16
Corriente DC Pin de E / S	20 mA	

---

**TIPOS DE TARJETAS**


---



Precio: 124.624 COP

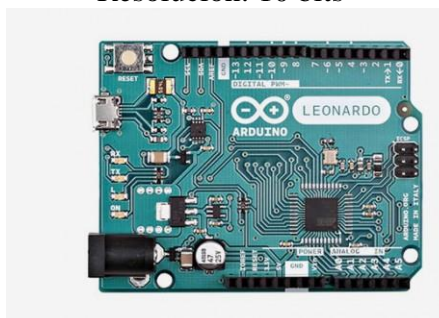
**Arduino Nano**  
Resolución: 10 bits



Precio: 71.214 COP

**Arduino Leonardo with Headers**

Resolución: 10 bits



Precio: 64.092 COP

Corriente DC para 3.3V Pin	50 mA
Memoria flash	256 KB of which 8 KB
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52 mm
Anchura	53.3 mm
Peso	37 g
Microcontrolador	ATmega328
Arquitectura	AVR
Tensión de funcionamiento	5 V
Memoria flash	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Velocidad de reloj	16 MHz
Analog IN Pins	8
EEPROM	1 KB
Corriente DC por los pernos de E / S	40 mA (I/O Pins)
Voltaje de entrada	7-12 V
Pernos digitales de E / S	22 (6 of which are PWM)
Salida PWM	6
El consumo de energía	19 mA
Tamaño de PCB	18 x 45 mm
Peso	7 g
Código de producto	A000005
Microcontrolador	ATmega32u4
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Pernos digitales de E / S	20
Canales PWM	7
Canales de entrada analógica	12
Corriente DC por Pin de E / S	40 mA
Corriente DC para 3.3V	50 mA
Memoria flash	32 KB (ATmega32u4)

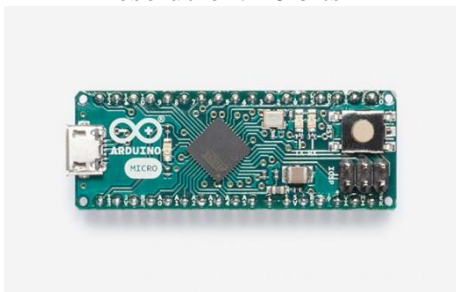
---

---

**TIPOS DE TARJETAS**


---

Arduino Micro  
Resolución: 10 bits



Precio: 64.092 COP

SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad de reloj	16 MHz
Longitud	68.6 mm
Anchura	53.3 mm
Peso	20 g
Microcontrolador	ATmega32U4
Tensión de funcionamiento	5V
Voltaje de entrada	7-12V
Voltaje de entrada (límite)	6-20V
Pernos digitales de E / S	20
Canales PWM	7
Canales de entrada análoga	12
Corriente DC Pin de E / S	20 mA
Corriente DC para 3.3V	50 mA
Memoria flash	32 KB (ATmega32U4)
SRAM	2.5 KB (ATmega32U4)
EEPROM	1 KB (ATmega32U4)
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	48 mm
Anchura	18 mm
Peso	13 g

---

También existe la opción de la tarjeta Raspberry Pi 3 Modelo B+



Precio: 147.154 COP

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)

---

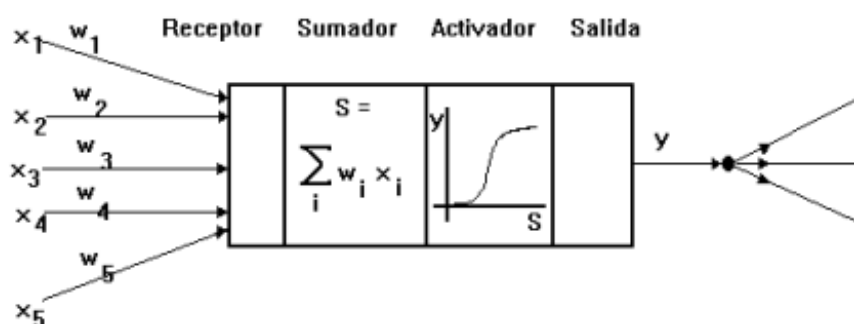
Fuente: Fabricantes

**2.2.14 Redes Neuronales artificiales (RNA).** Son sistemas conexionistas, su funcionalidad y estructura está fundamentada e inspirada en la biología del ser humano. Al igual que las redes neuronales biológicas y haciendo analogía de ellas consisten en un grupo o conjunto de elementos simples de procesamiento los cuales reciben el nombre de Neuronas o nodos las conexiones entre ellas poseen un valor numérico modificable denominado peso.

La labor que una neurona artificial ejecuta en un sistema por lo general es simple, En la mayoría de casos consiste en la suma de las entradas (Inputs), las cuales son obtenidas de otras unidades que se encuentran conectadas a esta, debe realizar una comparación entre el valor obtenido y el umbral y si llega a superarlo o igualarlo enviara activación o salida (output), a todas las unidades con las cuales se encuentre en conexión. Las entradas y las salidas también dependen del peso o la fuerza que posea las conexiones por las cuales se llevan a cabo dichas operaciones.

### La neurona artificial:

La neurona artificial es una unidad procesadora con 4 elementos funcionales:



**Figura 20. Esquema de la neurona artificial**

Fuente: Lara, s.f.



El elemento receptor: a donde llegan una o varias señales de entrada  $x_i$ , que generalmente provienen de otras neuronas y que son atenuadas o amplificadas cada una de ellas con arreglo a un factor de peso  $w_i$  que constituye la conectividad entre la neurona fuente de donde provienen y la neurona de destino en cuestión.

El elemento sumador: que efectúa la suma algebraica ponderada de las señales de entrada, ponderándolas de acuerdo con su peso, aplicando la siguiente expresión:

$$S = \sum_{i=0}^{i=n} W_i X_i$$

El elemento de función activadora: que aplica una función no lineal de umbral (que frecuentemente es una función escalón o una curva logística) a la salida del sumador para decidir si la neurona se activa, disparando una salida o no.

El elemento de salida que es el que produce la señal: de acuerdo con el elemento anterior, que constituye la salida de la neurona. Este modelo neuronal es el utilizado en casi todas las Redes Neuronales artificiales, variando únicamente el tipo de función activadora. A continuación, se presentarán los modelos más simples de Redes Neuronales artificiales.

### **El Perceptrón:**

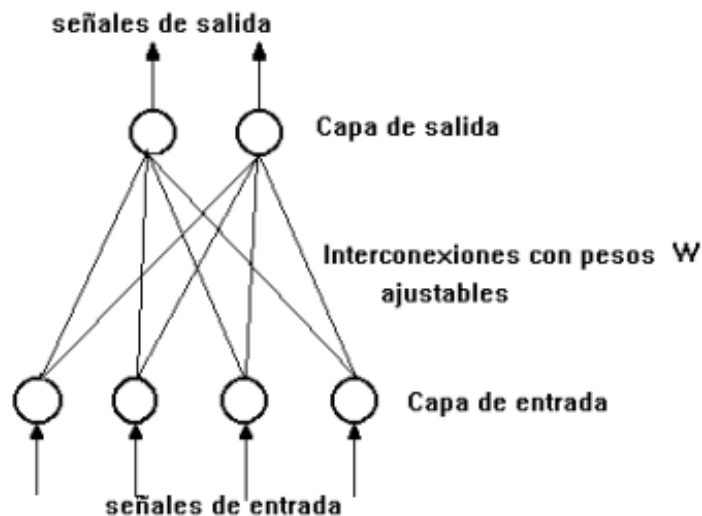
En se presenta de manera sencilla El Perceptrón, desarrollado por Rosenblatt (1958), consiste en una neurona procesadora, con sus elementos de entrada, sumador, activador y de salida. A la cual llegan señales de entrada  $X_i$ ,  $i = 1, 2n$  cada una con una a través de una línea con conductividad o peso asociado  $W_i$ , El elemento sumador efectúa entonces una suma ponderada de las entradas, en tanto que el activador emplea una función escalón de umbral: si la suma

ponderada es mayor o igual a un valor de umbral  $U$ , da una salida y de tal manera que:

$$Y = 1 \text{ si } S > U$$

$$Y = 0 \text{ si } S < U$$

La red neuronal más simple construida con perceptrones tiene dos capas: una capa receptora de entrada, en la que la salida de cada neurona reproduce simplemente su entrada (Fig. 10) y una capa de salida formada por perceptrones como los descritos, totalmente conectados con la capa de entrada, a través de líneas de comunicación con conductividades o pesos ajustables. Así, cada neurona de entrada está conectada con cada neurona de salida a través de una línea de comunicación con una conductividad o peso ajustable. La ley de aprendizaje del perceptrón ajusta estos pesos, de manera que se obtenga con mayor probabilidad la salida deseable correspondiente a un cierto conjunto de entradas. (Felipe Lara Rosano, 2018) .



**Figura 21. Perceptrón de dos capas.**

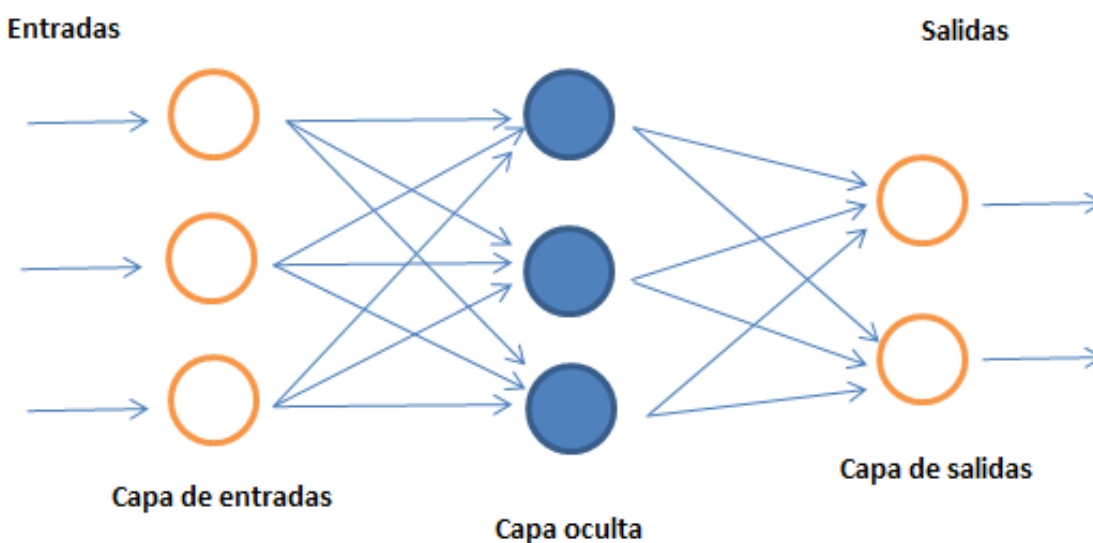
Fuente: Lara, s.f.

### Tipos de redes neuronales artificiales:

Los tipos de redes neuronales artificiales o RNAs se diferencian dependiendo de la forma en sus conexiones ya que esto afecta la manera en la cual son procesadas las entradas. A continuación, se presentan algunos de ellos:

#### Red neuronal Feedforward (FNN):

Es también llamada red neuronal pre alimentada en ella las conexiones entre las neuronas no forman un ciclo. Esta ha sido la más sencilla y primera en implementarse la información solo sigue un camino desde las entradas hasta las capas ocultas para finalizar en las salidas.

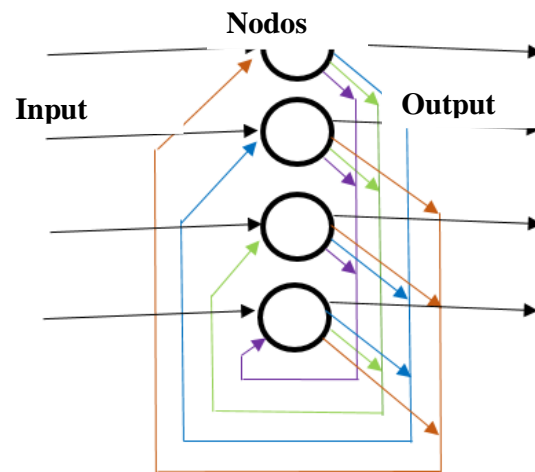


**Figura 22. Red neuronal artificial feed-forward de una sola capa oculta**

#### Red neuronal Artificial Hopfield:

Fue inventada por John Hopfield son usadas como sistemas de memoria asociativa con unidades binarias. Están diseñadas para converger a un mínimo local pero la convergencia a uno

de los patrones almacenados no está garantizada.

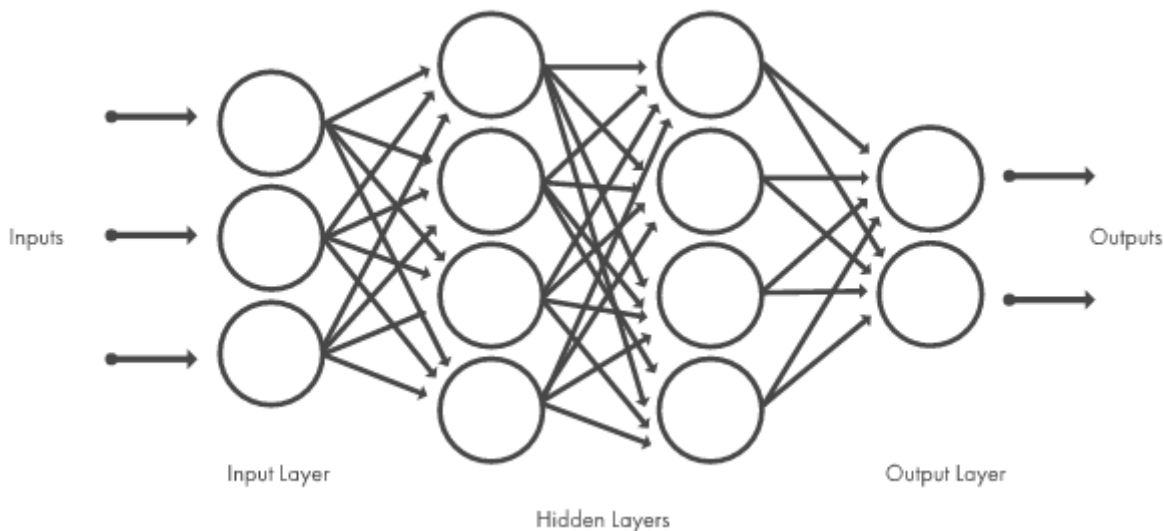


**Figura 23. RNA Hopfield con cuatro nodos**

#### **Red neuronal convolucional (CNNs):**

Una CNNs o ConvNet se encuentra entre los algoritmos de mayor aceptación para aprendizaje profundo; se define como un tipo de aprendizaje automático donde el modelo aprende la realización de tareas de clasificación en el cual las entradas son textos, sonidos, imágenes o videos.

Estas están compuestas por neuronas de entradas, capas o cultas y neuronas de salida; una RNA convolucional puede tener decenas o cientos de capas que aprenden a detectar diferentes características de una imagen mediante la aplicación de filtros en sus capas ocultas identificando atributos como brillo y bordes hasta definir completamente el objeto. (MathWorks, 2018)



**Figura 24. Red neuronal Convolutiva de dos capas ocultas.**

Fuente: MathWorks, 2018.

### **Backpropagation:**

La propagación hacia atrás de errores o retro propagación se emplea en los algoritmos de aprendizaje supervisado que se utilizan para entrenar redes neuronales artificiales.

Después de aplicar un patrón a la entrada de la red este se propaga a través de las capas de la misma y genera una salida, esta es comparada con la salida deseada y se calcula una señal de error para cada una de las señales de salida.

### **Técnicas de clasificación:**

En la actualidad las técnicas de clasificación, basadas en reconocimiento de patrones, más utilizadas son: las Redes Neuronales (NN), Lógica Difusa (FL) y el enfoque probabilístico.

**Redes Neuronales (Neural Networks)** Las redes neuronales son las más utilizadas como clasificadores. Estas se basan en el aprendizaje de nuevas tareas a partir de un conjunto de datos. Mediante el uso de redes neuronales de tipo Perceptrones Multicapa (MLP)

**Lógica Difusa (Fuzzy Logic)** Esta técnica de clasificación permite lidiar con datos contradictorios y descubrir patrones que no son fáciles de encontrar con otros métodos de clasificación. Permite además incorporar información proveniente de experiencia previa de expertos.

**Enfoque probabilístico** Dado que las señales EMG son estocásticas, un enfoque de clasificación basado en la probabilidad podría superar a los demás. Para esto se suelen utilizar modelos de tipo: Mixto Gaussiano (GMM) o de tipo Markov Oculto (HMM). En particular el modelo HMM provee buena exactitud y bajo costo computacional. Ambos implican trabajar con densidades de probabilidad cuyos parámetros se estiman a partir de los datos disponibles. Contrario al caso de las técnicas anteriores, donde previo a su uso se debe cuantificar la amplitud o potencia de la señal EMG, en el enfoque probabilístico es posible utilizar directamente los datos “crudos” de la señal EMG. (Jorge Brazeiro, Sabrina Petraccia. 2015).

**2.2.15 Interfaz gráfica.** La interfaz gráfica establecida para el operario o usuario también denominada (GUI), representa la información y acciones disponibles dentro de ella mediante objetos gráficos o mediante un conjunto de imágenes, su finalidad es proporcionar de manera sencilla un medio para comunicarse con el sistema operativo de una maquina o computadora.

**Tipos de interfaz gráfica.** Los tipos de interfaz varían dependiendo de los tipos de periféricos de entradas que poseen:

**GUI:** interfaz de grafica de usuario, representa por medio de objetos e imágenes las acciones disponibles.

**CLI:** Interfaz de línea de comandos, un ejemplo son las aplicaciones de cliente servidor.

**OUI:** interfaz de usuario orgánica, poseen una pantalla no plana o plegable.

**NUI:** Interfaz natural de usuario, es en la que el usuario por inercia o sentido común puede utilizar.

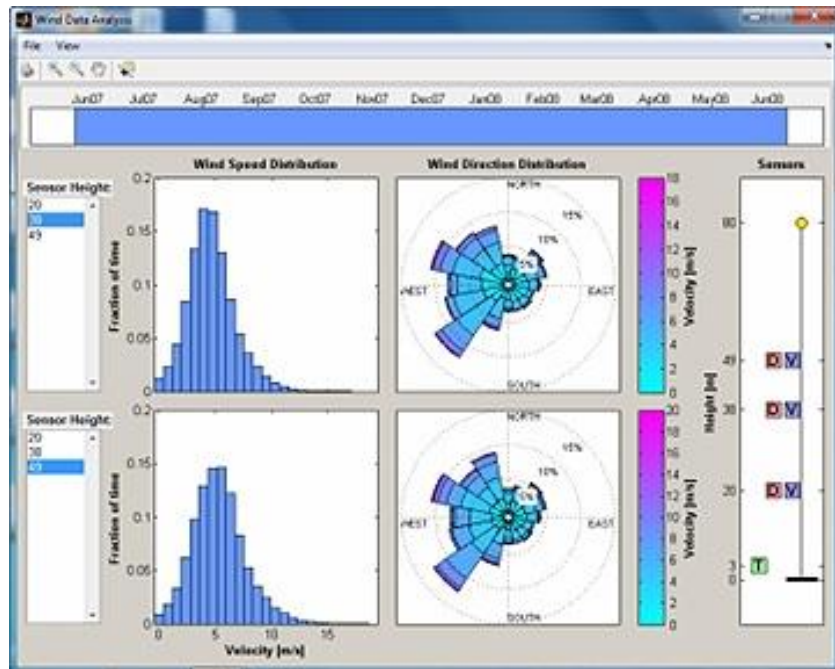
**PUI:** interfaz perceptual de usuario, su objetivo es lograr que la comunicación entre el usuario y la computadora se realice de una manera similar a como interactúan dos personas presentando la información de forma natural y significativa.

**Creación de apps con interfaces gráficas de usuario en MATLAB.** Las GUI (también conocidas como interfaces gráficas de usuario o interfaces de usuario) permiten un control sencillo (con uso de ratón) de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos a fin de ejecutar una aplicación.

Las apps de MATLAB son programas autónomos de MATLAB con un frontal gráfico de usuario GUI que automatizan una tarea o un cálculo. Por lo general, la GUI incluye controles tales como menús, barras de herramientas, botones y controles deslizantes. Muchos productos de MATLAB, como Curve Fitting Toolbox, Signal Processing Toolbox y Control System Toolbox, incluyen apps con interfaces de usuario personalizadas. También es posible crear apps personalizadas propias, incluidas las interfaces de usuario correspondientes, para que otras personas las utilicen.

**Creación de una GUI de MATLAB de forma interactiva.** GUIDE (entorno de desarrollo de GUI) proporciona herramientas para diseñar interfaces de usuario para Apps personalizadas. Mediante el editor de diseño de GUIDE, es posible diseñar gráficamente la interfaz de usuario. GUIDE genera entonces de manera automática el código de MATLAB para construir la interfaz, el cual se puede modificar para programar el comportamiento de la app.

**Creación de una GUI de MATLAB de forma programática.** A fin de ejercer un mayor control sobre el diseño y el desarrollo, también se puede crear código de MATLAB que defina las propiedades y los comportamientos de todos los componentes. MATLAB contiene funcionalidad integrada que le ayudará a crear la GUI para su app de forma programática. Cabe la posibilidad de agregar cuadros de diálogo, controles de interfaz de usuario (como botones y controles deslizantes) y contenedores (como paneles y grupos de botones).



**Figura 25. Aplicación personalizada con una GUI en MATLAB**

Fuente: Mathworks, 2019.



### **2.2.16 MathWorks versión 2017b de las familias de productos de MATLAB y Simulink.**

Refuerza las capacidades de aprendizaje profundo para simplificar el diseño, el entrenamiento y el despliegue de modelos Natick, MA, United States - (21 Sep 2017)

“MathWorks ha presentado hoy la versión 2017b (R2017b), que ofrece nuevas funciones en MATLAB y Simulink, así como seis nuevos productos y actualizaciones y correcciones para otros 86 productos. Esta versión también añade importantes capacidades nuevas de aprendizaje profundo que simplifican la forma en que los ingenieros, investigadores y otros expertos diseñan, entrenan y despliegan modelos.”

#### **Compatibilidad con el aprendizaje profundo:**

Entre las funciones, los productos y las capacidades específicas para el aprendizaje profundo de la versión R2017b se encuentran:

Neural Network Toolbox ofrece ahora compatibilidad con arquitecturas complejas, incluidas las redes DAG (grafo acíclico dirigido) y LSTM (memoria de corto-largo plazo), además de proporcionar acceso a modelos populares previamente entrenados, tales como GoogLeNet.

La aplicación Image Labeler de Computer Vision System Toolbox proporciona ahora una forma cómoda e interactiva para etiquetar datos de validación en terreno en una secuencia de imágenes. Además de los flujos de trabajo de detección de objetos, esta toolbox también admite ahora la segmentación semántica mediante aprendizaje profundo para clasificar regiones de píxeles en las imágenes, así como evaluar y visualizar los resultados de la segmentación.

Un nuevo producto, GPU Coder, convierte automáticamente los modelos de aprendizaje profundo en código de CUDA destinado a GPUs de NVIDIA. Las comparativas internas

muestran que el código generado para la inferencia mediante aprendizaje profundo obtiene un rendimiento hasta siete veces superior al de TensorFlow y cuatro veces y media superior al de Caffe2 en el caso de los modelos desplegados. \*

Además de las capacidades introducidas en la versión R2017a, se pueden emplear modelos previamente entrenados para el aprendizaje por transferencia, incluidos modelos de redes neuronales convolucionales (CNN) (AlexNet, VGG-16 y VGG-19), así como modelos de Caffe (incluido Caffe Model Zoo). Los modelos se pueden desarrollar desde cero, lo cual incluye el uso de redes CNN para la clasificación de imágenes, la detección de objetos, la regresión, etc.

“Con el desarrollo de los dispositivos inteligentes y de Internet de las cosas, los equipos de diseño se enfrentan al reto de crear productos y aplicaciones más inteligentes mediante el desarrollo por su cuenta de habilidades relacionadas con el aprendizaje profundo o confiando en otros equipos con experiencia en aprendizaje profundo que pueden no comprender el contexto de la aplicación”, según afirma David Rich, director de marketing de MATLAB en MathWorks.

“Con la versión R2017b, los equipos de ingeniería e integración de sistemas pueden ampliar el uso de MATLAB para el aprendizaje profundo a fin de aumentar el control sobre todo el proceso de diseño y lograr diseños de mayor calidad con más rapidez. Pueden emplear redes previamente entrenadas, colaborar en el código y los modelos, y realizar el despliegue en GPUs y dispositivos embebidos. El uso de MATLAB puede aumentar la calidad del resultado, a la vez que se reduce el tiempo de desarrollo del modelo gracias a la automatización del etiquetado de validación en terreno.”

### **Actualizaciones adicionales:**

Además del aprendizaje profundo, la versión R2017b también incluye una serie de actualizaciones en otras áreas clave, tales como:

#### **Análisis de datos con MATLAB**

El nuevo producto Text Analytics Toolbox, un almacén de datos ampliable, más gráficos y algoritmos de big data para aprendizaje automático y compatibilidad con el almacenamiento de blobs mediante Microsoft Azure.

#### **Modelado de software en tiempo real con Simulink**

Modele los efectos de la planificación e implemente componentes utilizables en los entornos de software.

#### **Verificación y validación con Simulink**

Nuevas herramientas para modelado de requisitos, análisis de cobertura de pruebas y comprobación de conformidad.

### **2.3 Marco Contextual**

El proyecto de investigación titulado Identificación de movimiento a partir de señales electromiografías presentes en el antebrazo, utilizando redes neuronales artificiales en Matlab. Se desarrolló en las Instalaciones de la Universidad Francisco De Paula Santander en las áreas que comprenden los laboratorios que son designados para los alumnos en desarrollo de su proyecto de grado.

En dicho proyecto se realizó el estado del arte permitiendo el estudio del conocimiento que esta acumulado en libros y publicaciones textuales, referentes a las áreas específicas concernientes.

La medición, interpretación y análisis de las señales electromiográficas ha sido una contribución de gran importancia para la medicina, permitiendo la creación de dispositivos electrónicos que monitorean el estado de los músculos de los pacientes, logrando determinar alguna afección; del mismo modo la elaboración de prótesis mioeléctricas avanzadas que generan movimiento dependiendo de dichas señales permitiéndole a sujetos que presenten amputaciones ya sea de tipo traumático, quirúrgico o de nacimiento mejorar su calidad de vida.

El proyecto busco identificar de manera eficiente y análoga el movimiento que es producido en la mano de un sujeto en condiciones normales, esto mediante la lectura y acondicionamiento de las señales electromiográficas que se producen en el antebrazo, para ello se basara en la selección, implementación y posterior entrenamiento de una Red neuronal artificial en el software Matlab. Generando una interfaz didáctica que permita al usuario la visualización de las acciones que se produce con la mano, permitiendo el entrenamiento muscular de personas que presenten amputaciones.

### 3. Diseño Metodológico

#### 3.1 Tipo de Proyecto

Este es un proyecto de tipo investigativo y desarrollo (proyecto I+D). La investigación y desarrollo complementan de manera dinámica el incremento del conocimiento para ello es importante el conocimiento creativo del hombre y otros factores elementales como la cultura y la sociedad, el uso de todos estos conocimientos es necesario para crear y desarrollar nuevas aplicaciones esenciales para el entendimiento de diferentes factores electromecánicos. El termino I+D del tipo de proyecto engloba tres actividades: investigación básica, investigación aplicada y desarrollo experimental.

#### 3.2 Técnicas de Recolección de Datos

**Fuentes:** las fuentes que se utilizaron para la realización de este proyecto de investigación se basaron en: libros especializados de acceso público que se encuentren en la Red, artículos de Investigación que correspondan al tema del proyecto o semejantes, libros pertenecientes a la universidad Francisco de Paula Santander; en estos se buscara toda la información concerniente al desarrollo de la toma de mediciones de señales Electromiográficas como acondicionarlas e interpretarlas.

También se contó con las páginas web principales de Matlab, Arduino, Advancer Technologies, entre otras las cuales permitirán adquirir información software y documentación para un desarrollo integral de la propuesta.

## 4. Desarrollo de Proyecto

### 4.1 Selección de Microcontrolador

La selección del convertidor a usar en este proyecto se realizó mediante el análisis de características específicas de diferentes microcontroladores para hacer una evaluación costo beneficio. Se tuvo en cuenta aspectos relevantes como lo son tamaño, peso, resolución y costo económico.

A continuación, se referencia los posibles microcontroladores que hicieron parte de la selección.

**Tabla 5. Convertidores para selección**

NOMBRE	COSTO (COP)	PESO	BENEFICIO		RESOLUCIÓN (bits)
			TAMAÑO (mm)		
<b>Arduino Uno Rev3</b>	71.214	13g	Longitud	68.6	10
			Anchura	53.4	
<b>Arduino MKR WAN 1300 (LoRa connectivity)</b>	129.156	32g	Longitud	67.64	32
			Anchura	25	
<b>ARDUINO UNO WiFi REV2</b>	145.341	25g	Longitud	68.6	10
			Anchura	53.4	
<b>Arduino Mega 2560 Rev3</b>	124.625	37g	Longitud	101.52	10
			Anchura	53.3	
<b>Arduino MKR WiFi 1010</b>	109.734	32g	Longitud	61.5	32
			Anchura	25	
<b>Arduino Nano</b>	71.214	7g	Tamaño de PCB	18 x 45	10
<b>Arduino Due</b>	124.625	36g	Longitud	101.52	32

NOMBRE	COSTO (COP)	PESO	BENEFICIO		RESOLUCIÓN (bits)
			TAMAÑO (mm)		
			Anchura	53.3	
<b>Arduino MKR1000 WIFI</b>	113.263	32g	Longitud	61.5	32
			Anchura	25	
<b>Arduino Micro</b>	64.093	13g	Longitud	48	10
			Anchura	18	
<b>Arduino Leonardo with Headers</b>	64.093	20g	Longitud	68.6	10
			Anchura	53.3	
<b>Arduino Uno Rev3 SMD</b>	67.653	25g	Longitud	68.6	10
			Anchura	53.4	
<b>Raspberry Pi 3 Modelo B+</b>	147.154	65g	Longitud	101.52	32
			Anchura	53.3	

Fuente: fabricantes

**Microcontrolador seleccionado.** Atendiendo a los criterios de selección anteriormente expuestos luego de análisis por parte del equipo de trabajo de este proyecto se determinó que la mejor opción es el microcontrolador **Arduino Micro**, el cual cumple con los aspectos requeridos con un precio de \$19.80, un peso de 13g, un tamaño de Longitud 48 mm y Anchura 18 mm, una resolución de 10 bits, siendo en conjunto los valores menores en la tabla de comparación costo beneficio, y a pesar de que otros microcontroladores tienen una resolución mayor a 10 bits, se concluyó que esta resolución es muy aceptable para la realización este proyecto y al cumplir con los otros parámetros de selección, como el tamaño, se optó por este microcontrolador.

#### 4.2 Selección del Sensor Muscular

Para la selección del sensor de señales electromiográficas se tuvieron en cuenta diversos aspectos a considerar, ya que la medición de señales incluye por si sola para su correcta

interpretación la inclusión de etapas de acondicionamiento, amplificación y filtrado para así obtener la señal suministrada lo más fiel posible.

Teniendo en cuenta esto se opta por la búsqueda de un sensor que incluya todas estas etapas integradas, aun conociendo que es posible la construcción de dichos implementos por parte del equipo de este proyecto. Sin embargo, al sopesar el tiempo adicional junto al trabajo investigativo y operativo que esto representaría, sumado a que la diferencia económica no es tan significativa en la obtención del sensor que incluya dichas características se determina la compra del sensor que incluya lo antes mencionado.

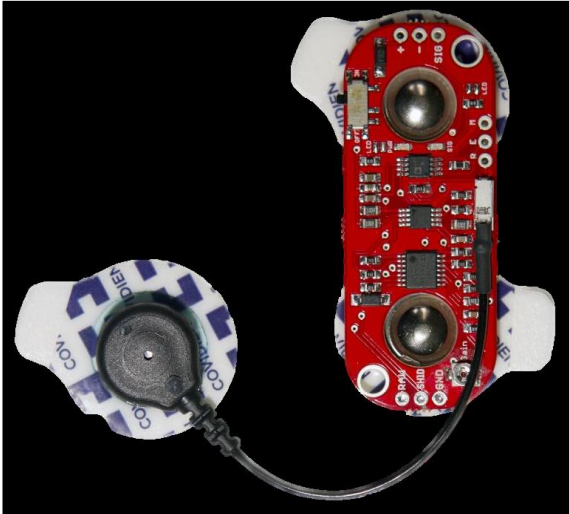

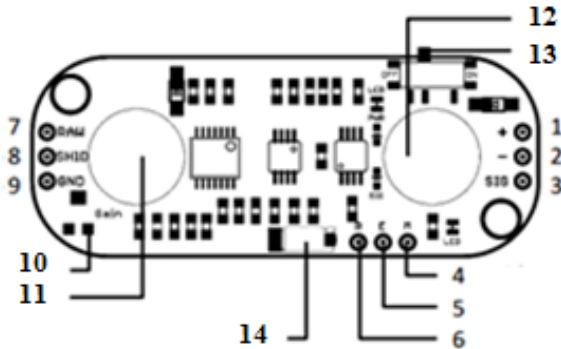
#### **Sensor seleccionado:**

Se eligió el sensor MyoWare™ Muscle Sensor (AT-04-001) que cumple con criterios como tamaño, peso, economía y como condición adicional nos permite una fácil conexión con microcontroladores.



## Características:

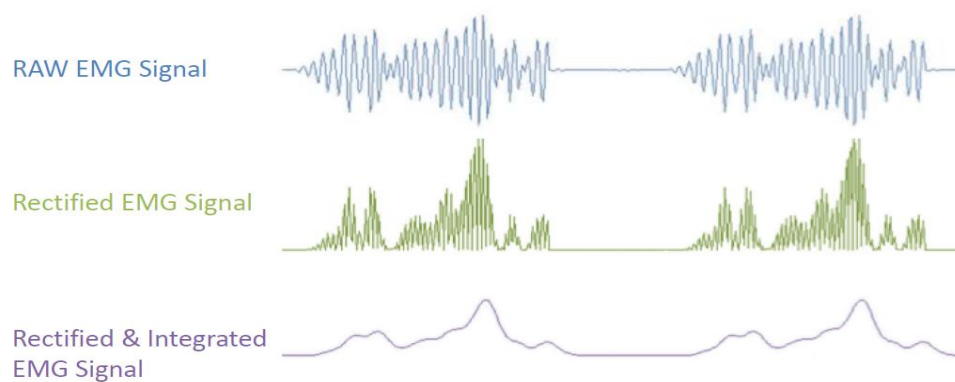
Tabla 6. Características del sensor

	<p>Diseñado para llevar (portable)          Suministro único</p> <ul style="list-style-type: none"> <li>+ 2.9V a + 5.7V</li> <li>Protección contra inversión de polaridad.</li> </ul> <p>Dos modos de salida</p> <ul style="list-style-type: none"> <li>Sobre EMG</li> <li>EMG sin procesar</li> </ul> <p>Expandible vía Shields          Indicadores LED          Especialmente diseñado para microcontroladores.          Ganancia ajustable</p>
	<p style="text-align: center;"><b>APLICACIONES</b></p> <ul style="list-style-type: none"> <li>Videojuegos.</li> <li>Robótica</li> <li>Dispositivos médicos</li> <li>Electrónica usable / móvil</li> <li>Prótesis / Ortesis</li> </ul>
	<ol style="list-style-type: none"> <li>Fuente de alimentación +Vs</li> <li>Fuente de alimentación, GND</li> <li>Señal de salida, SIG</li> <li>Pin del electrodo del musculo medio</li> <li>Pin del electrodo del musculo extremo</li> <li>Pin del electrodo de referencia</li> <li>Señal EMG sin procesar</li> <li>Escudo de potencia(output)</li> <li>GND</li> <li>Ganancia ajustable</li> <li>Electrodo del musculo extremo</li> <li>Broche de electrodo del musculo medio</li> <li>Interruptor de alimentación</li> <li>Cable del electrodo de referencia.</li> </ol>

Fuente: Fabricantes

### RAW EMG vs EMG:

Este sensor muscular está diseñado para ser utilizado directamente con un microcontrolador. Por lo tanto, la salida primaria del sensor no es una señal RAW EMG sino una señal amplificada, rectificadora y señal integrada (también conocida como la envolvente de EMG) que funcionará bien con un microcontrolador, Convertidor analógico a digital (ADC). Esta diferencia se ilustra a continuación utilizando una Señal EMG representativa.



**Figura 26. RAW EMG vs EMG**

Fuente: fabricantes

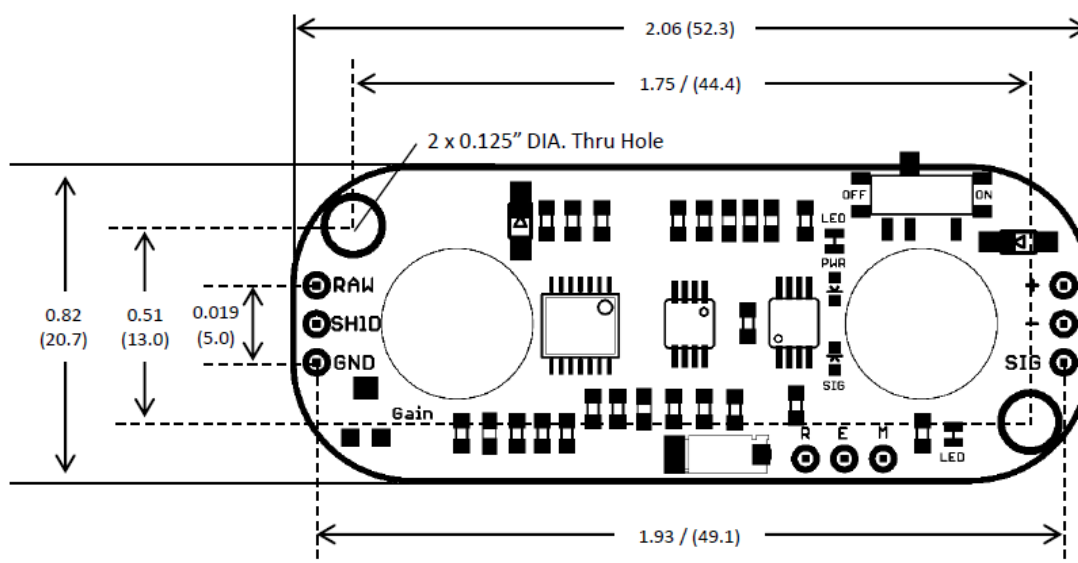
## Especificaciones eléctricas:

**Tabla 7. Especificaciones eléctricas**

Parámetro	Min	TYP	Max
Voltaje de alimentación	+2.9V	+3.3V or +5V	+5.7V
Potenciómetro de ganancia ajustable	0.01 $\Omega$	50 k $\Omega$	100 k $\Omega$
Voltaje de la señal de salida	0V	--	+Vs
Sobre EMG	0V	--	+Vs
EMG sin procesar (centrado alrededor de + Vs / 2)			
Impedancia de entrada	--	110 G $\Omega$	--
Corriente de suministro	--	9 mA	14 mA
Relación de rechazo de modo común (CMRR)	--	110	--
Sesgo de entrada	--	1 pA	--

Fuente: fabricantes

## Dimensiones del sensor (dadas en pulgadas – milímetros):



**Figura 27. Dimensiones del sensor**

Fuente: Fabricantes

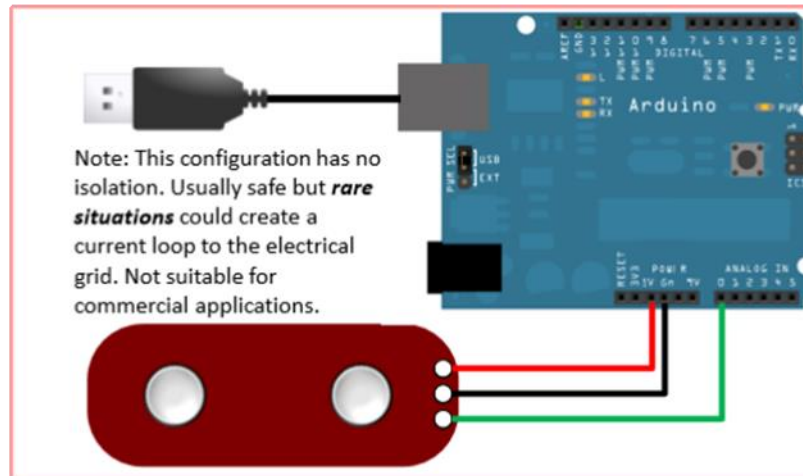
### Otros elementos utilizados.

**Tabla 8. Equipos y materiales**

ÍTEM	Equipo y materiales	Cantidad	Descripción
1	Papelería	1	Fotocopias, lapiceros, lápices, papel tamaño carta, CD, DVD, tinta, impresora, tóner, carpetas.
2	Computador Personal	2	Intel core i5; Sistema operativo Windows 10; RAM de 8 Gb; Disco duro 1 Tb.
3	Impresora	1	Multifuncional, tinta a color y negro.
4	Software	1	Matlab 2017b licenciado proporcionado por universidad Francisco De Paula Santander
5	Cables de conexión	1	Cables según requerimientos de los dispositivos electrónicos

### 4.3 Conexión del Sensor MyoWare y la Interfaz IDE de Arduino

Después de realizar con éxito la selección de materiales y adquirirlos del mercado internacional (En el caso del Sensor Muscular y electrodos para el mismo). Se procede a realizar la conexión entre el MyoWare y el Arduino micro, para posteriormente conectarlos al equipo de cómputo y visualizar las señales que esta entrega, con la interfaz de IDE de Arduino versión 1.8.10. En el datasheet del Sensor se muestra distintas configuraciones de conexión entre los dispositivos y métodos de polarización en la **Figura 28** se aprecia el esquema, y la advertencia de que este método no establece un asilamiento eficiente entre el usuario y el dispositivo por dicha razón para la utilización, el computador portátil debe estar desconectado de la red eléctrica comercial. La conexión entre el PC y el Arduino se realiza mediante un cable USB a Micro USB (Comúnmente utilizados para cargar la mayoría de Celulares de plataforma Android).

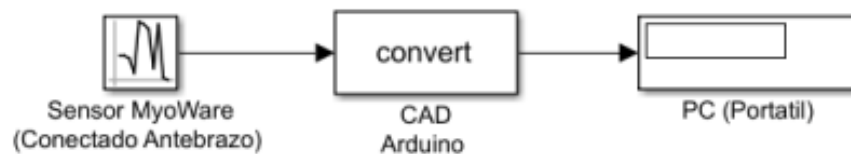


**Figura 28. Esquema de conexión de salida y polarización Sensor muscular**

MyoWare AT 04 001.

Fuente: Advancers Thecnologies, 2015.

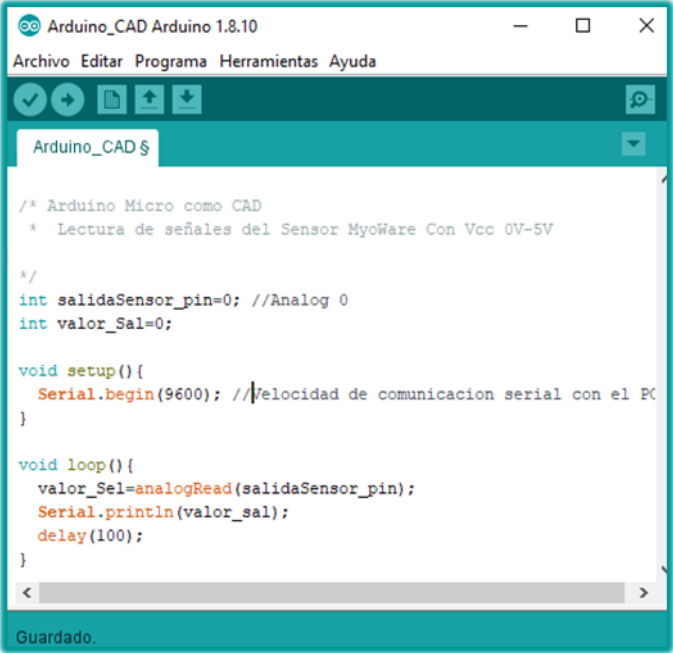
El dispositivo micro controlador se usa como convertidor A/D (análogo/Digital). A continuación, se presenta el primer diagrama de conexión básica de los elementos eléctricos y electrónicos para realizar una lectura de señal,



**Figura 29. Diagrama de bloques de conexión de los equipos físicos desde el sensor al PC.**

Se determinó con el equipo de trabajo que se desarrollaría toda la programación necesaria en el software de MATLAB, dejando el dispositivo Arduino solo para adquirir los datos Analógicos y entregarlos en digital. Fue necesario realizar una pequeña configuración del pin de entrada y de

salida y la velocidad de comunicación serial con el PC. La programación del Arduino queda de la siguiente manera:



```

Arduino_CAD Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

Arduino_CAD $

/* Arduino Micro como CAD
 * Lectura de señales del Sensor MyoWare Con Vcc 0V-5V

 */
int salidaSensor_pin=0; //Analog 0
int valor_Sal=0;

void setup() {
  Serial.begin(9600); //Velocidad de comunicacion serial con el PC
}

void loop() {
  valor_Sal=analogRead(salidaSensor_pin);
  Serial.println(valor_sal);
  delay(100);
}

Guardado.

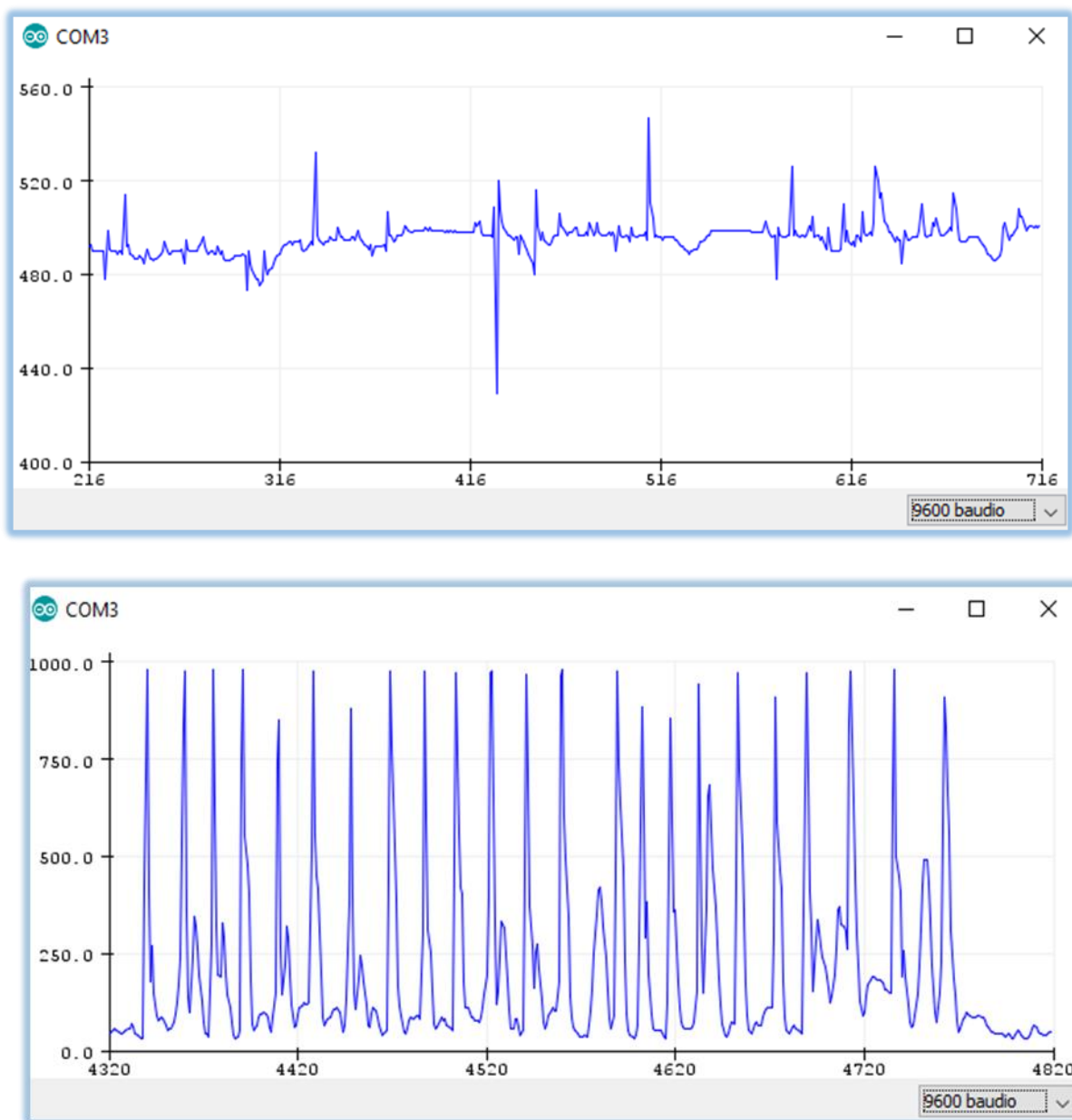
```

**Figura 30. Programación básica del Arduino Micro para lectura de la señal del Sensor**

Del dispositivo de medición se pueden obtener dos señales, la electromiografía original (RAW) y la misma señal, pero completamente acondicionada y con una ganancia máxima de 1000. Para la señal RAW es posible visualizarla con eje de referencia de  $V_{cc}/2$  ( $V_{cc}$  es voltaje de polarización) y un rango de voltaje aproximado entre 2-3 Voltios máximo, la señal acondicionada se presenta de 0-5V y se convierte de A/D en rango de 0-1023 bits.

En la señal RAW, que se presenta al realizar el movimiento del puño, es posible observar demasiadas variaciones y ruido a pesar de que solo realizó un desplazamiento vertical de la mano empuñada hacia arriba con el brazo completamente extendido. En la gráfica de la señal acondicionada se aprecia de una mejor manera la similitud de la señal, aunque también presenta

algunas variaciones indeseadas. Esto se puede observar a continuación;



(a) Electromiográfica original RAW. (b) Señal Acondicionada 0-5V

**Figura 31. Señales graficadas (movimiento del puño) en el IDE Arduino V1.8.10**

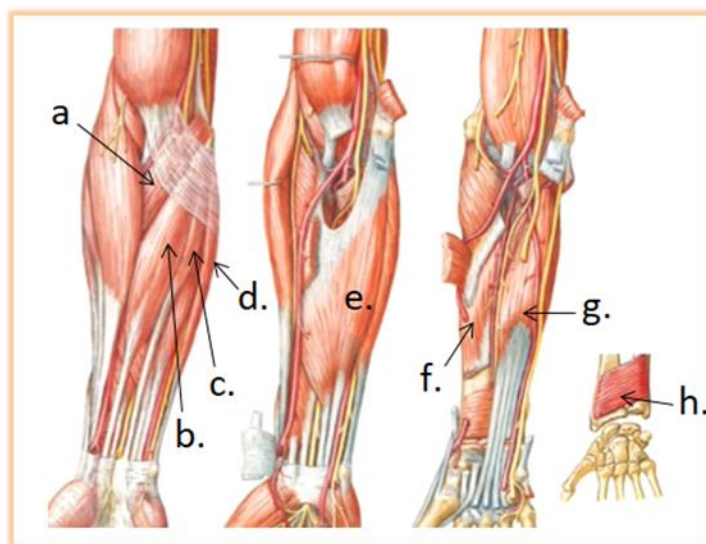
#### 4.4 Selección de la Posición final del Sensor Muscular

Según la investigación, las recomendaciones de hoja de datos del sensor y señales arrojadas por el mismo, se procede a la selección de la posición fija del mismo en el brazo con tal de identificar y utilizar el musculo o los músculos que permita leer la señal de movimiento de dedos.

**Tabla 9. Planos musculares de la región del antebrazo**

PLANOS MUSCULARES	
<b>Primer Plano</b>	a. Pronador redondo. b. Palmar mayor o Flexor Radial del Carpo. c. Palmar menor o palmar largo. d. Cubital anterior Flexor cubital del carpo.
<b>Segundo Plano</b>	e. Flexor Superficial de los dedos.
<b>Tercer Plano</b>	f. Flexor largo del pulgar. g. Flexor profundo de los dedos.
<b>Cuarto Plano</b>	h. Pronador Cuadrado.

Fuente: <https://www.youtube.com/watch?v=0jUq21h9ylc>



**Figura 32. Ubicación de los músculos presentes en la región del antebrazo**

Fuente: Garcia, 2017.



Gracias a la tabla anterior y la figura de ubicación de los músculos se elige la posición donde se encuentra el Flexor superficial de los dedos, también conocido como Flexor común, es importante reconocer que está en la segunda capa de músculos y también intervendrían los músculos que encuentran resaltados en la Tabla 9 por lo tanto, para la realización del presente proyecto el sensor se utilizó de la siguiente manera:

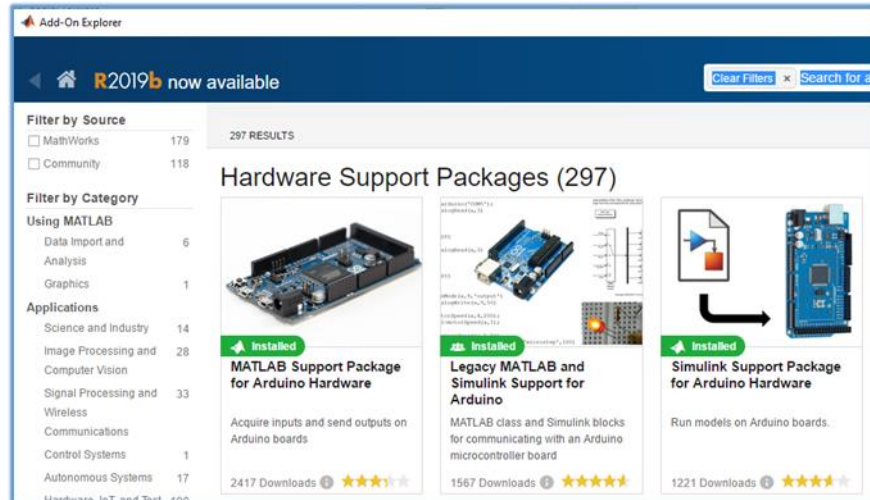


(a) Marcación y (b) Ubicación del sensor en el antebrazo para toma de muestras de las señales musculares

### Figura 33. Posición final del sensor

#### 4.5 Conexión de Arduino y Matlab

La conexión en Matlab del sensor y el Arduino se realizó de manera sencilla gracias a que Matlab permite descargar paquetes que conectan el hardware de Arduino, solo es necesario en la barra de **HOME** dirigirse al icono **Add-Ons** se despliega la ventana de opciones y se elige **Get Hardware support packages**. Esto despliega una ventana con las opciones para descarga. (Se utilizó **MATLAB Support package for Arduino Hardware**).



**Figura 34. Captura de pantalla de Add-On Explores de Matlab para la instalación de paquetes de soporte de Hardware**

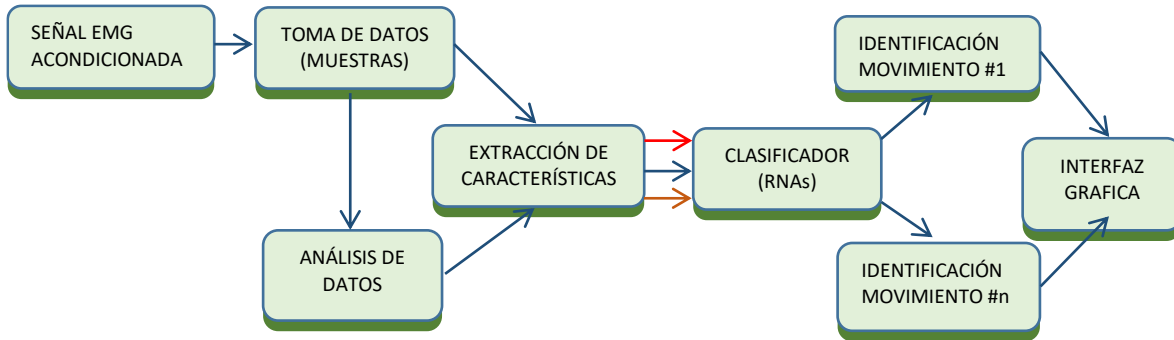
Después de tener estos paquetes la conexión es automática y Matlab identifica el dispositivo de hardware conectado, La única configuración que se detalló en todos los Script de conexión, es el siguiente Código básico que se puede encontrar utilizando el comando de ayuda (help Arduino)

```
a=arduino('COM3','MICRO');%crea una conexión con el hardware
Arduino en el puerto serie especificado.
configurePin(a,'A0');%se define el pin analógico donde se
realiza la lectura de la señal.
```

**Figura 35. Código 1**

#### 4.6 Diagrama de Bloques general del Proyecto

Para el desarrollo del proyecto fue esencial la construcción de un diagrama de bloques que permitiera visualizar de manera didáctica las distintas etapas a realizarse, con el fin de cumplir con los objetivos.

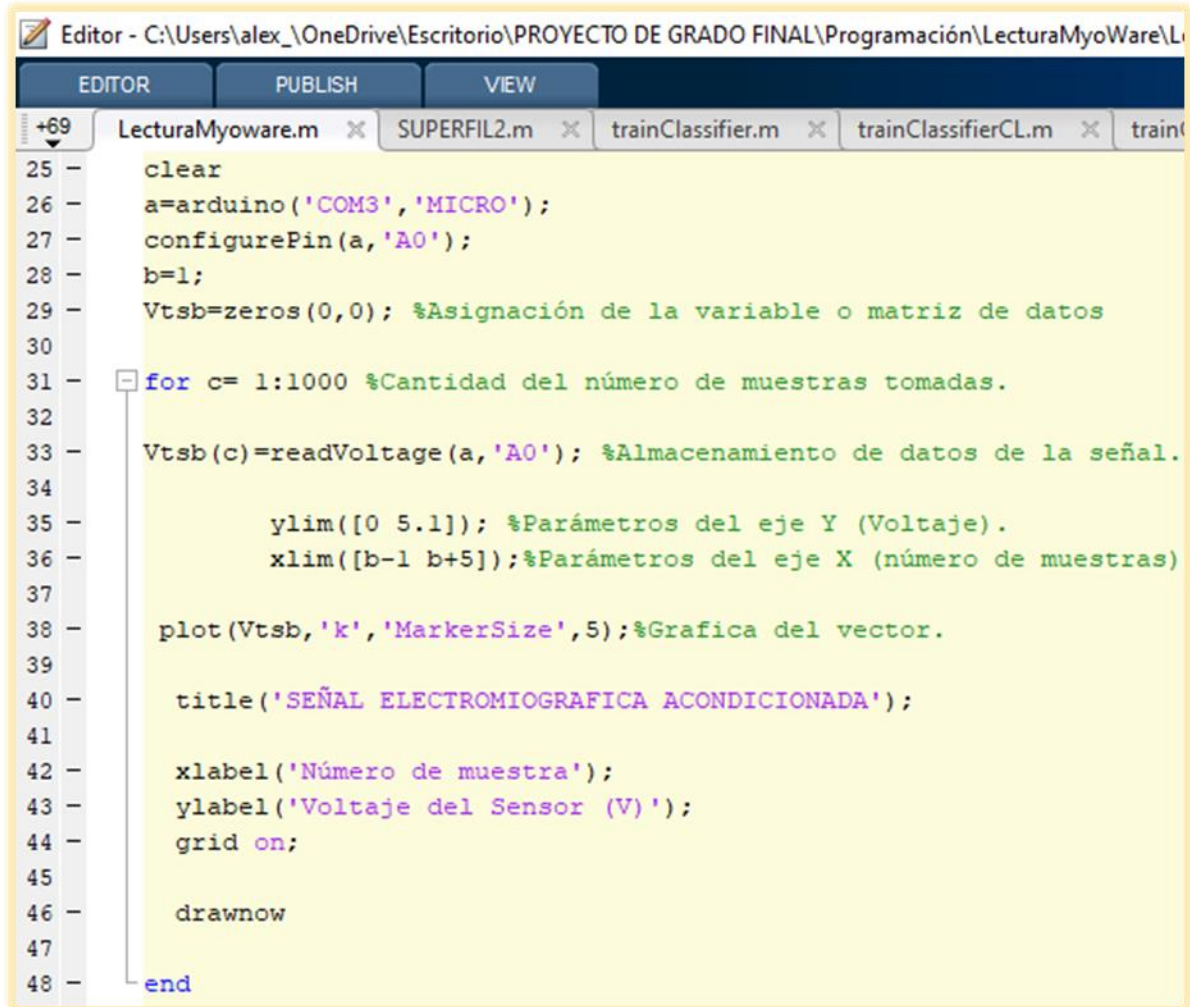


**Figura 36. Diagrama de bloques, etapas de desarrollo del proyecto**

#### **4.7 Lectura y Almacenamiento de Las señales Musculares en MATLAB**

Para la toma de la señal en el software de Matlab se elaboró un Script (**LecturaMyoWare.m**) que permitiera realizar la conexión teniendo en cuenta el puerto COM, leer, graficar y almacenar los datos como vectores(**\*.mat**), siendo posible variar la cantidad de muestras que se desea tomar.

Dicha programación cuenta con dos sesiones la primera es un bucle **for** que permite visualizar la señal desde el punto inicial 0 hasta el número de muestra puesto por el operador y almacenar la señal como un vector (**Vtsb.mat**).



```

Editor - C:\Users\alex_OneDrive\Escritorio\PROYECTO DE GRADO FINAL\Programación\LecturaMyoWare\L
EDITOR PUBLISH VIEW
+69 LecturaMyoware.m x SUPERFIL2.m x trainClassifier.m x trainClassifierCL.m x train
25 - clear
26 - a=arduino('COM3','MICRO');
27 - configurePin(a,'A0');
28 - b=1;
29 - Vtsb=zeros(0,0); %Asignación de la variable o matriz de datos
30
31 - for c= 1:1000 %Cantidad del número de muestras tomadas.
32
33 - Vtsb(c)=readVoltage(a,'A0'); %Almacenamiento de datos de la señal.
34
35 -     ylim([0 5.1]); %Parámetros del eje Y (Voltaje).
36 -     xlim([b-1 b+5]);%Parámetros del eje X (número de muestras)
37
38 -     plot(Vtsb,'k','MarkerSize',5);%Grafica del vector.
39
40 -     title('SEÑAL ELECTROMIOGRAFICA ACONDICIONADA');
41
42 -     xlabel('Número de muestra');
43 -     ylabel('Voltaje del Sensor (V)');
44 -     grid on;
45
46 -     drawnow
47
48 - end

```

**Figura 37. Captura de pantalla sesión inicial del Script para lectura y almacenamiento de la señal Electromiográfica. Software MATLAB**

La segunda parte es un bucle **while** que permite realizar una gráfica con eje x variable dependiendo del número de muestras tomadas, estas se muestran punto a punto, lo que permite al usuario observar que un movimiento brusco o acelerado provoca un salto grande entre los intervalos del eje Y (voltaje) de la gráfica, esto debido a la resolución de 10 bits del Arduino.

```

55
56 %Declara un contador del número de muestras ya tomadas
57 contador_muestras=1;
58
59 %Crea la ventana para la gráfica
60 figure('Name','Grafica punto a punto de la Señal del Sensor')
61 title('SEÑAL ELECTROMIOGRAFICA ACONDICIONADA');
62 xlabel('Número de muestra');
63 ylabel('Voltaje (V)');
64 grid on;
65 hold on
66
67 %Bucle while para que tome y dibuje las muestras que se requieran.
68 while contador_muestras %inicio del bucle
69     ylim([0 5.1]);
70     xlim([contador_muestras-20 contador_muestras+5]);
71     contador_muestras=contador_muestras+1; %permite avanzar la grafica en el eje x
72
73     Vs=readVoltage(a,'AO');
74     plot((contador_muestras),Vs,'-.');%Grafica de Vs con respecto al contador d emuestras.
75
76
77     drawnow
78
79     if xlim>1078 %La cantidad de muestras esta dado por (Valor deseado-22 = valor Limite);(Ejemplo 100-22=78)
80         break
81     end
82
83 end

```

**Figura 38. Captura de pantalla de segunda sesión del Script (LecturaMyoWare.m) de la señal Electromiográfica. Software MATLAB**

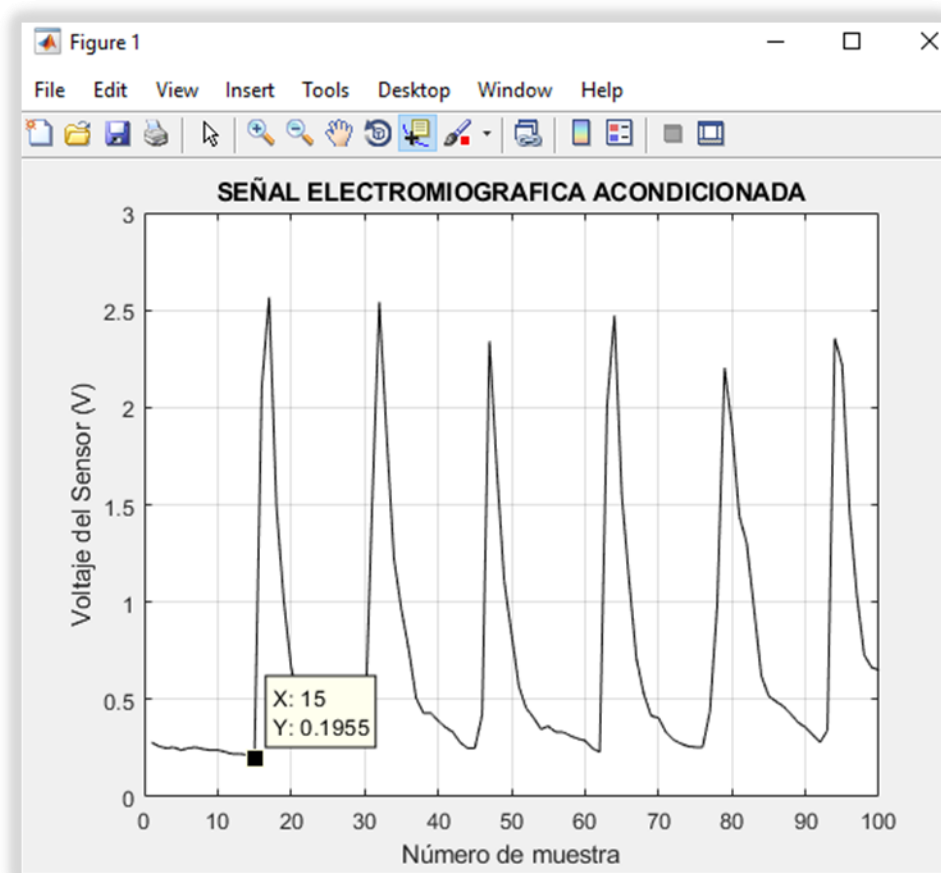
Las señales de respuesta que se presentaron son las siguientes:

Aproximadamente se tarda 2 segundos antes de que el programa empiece a realizar la lectura de la primera señal de voltaje o pulso enviado por el sensor. (A partir de la muestra 15).

$$\text{Frecuencia de muestreo}(F_s) = \frac{\text{NUMERO DE MUESTRAS}}{\text{TIEMPO}} = \frac{500}{60\text{Seg}} = 8.33333333 \frac{1}{\text{Seg}}$$

$$\begin{aligned} \text{Tiempo por muestra} &= \frac{1 \text{ MUESTRAS}}{F_s} = \frac{1}{8.33333333} = 0.12\text{seg} \\ &= 120\text{mseg} \end{aligned}$$

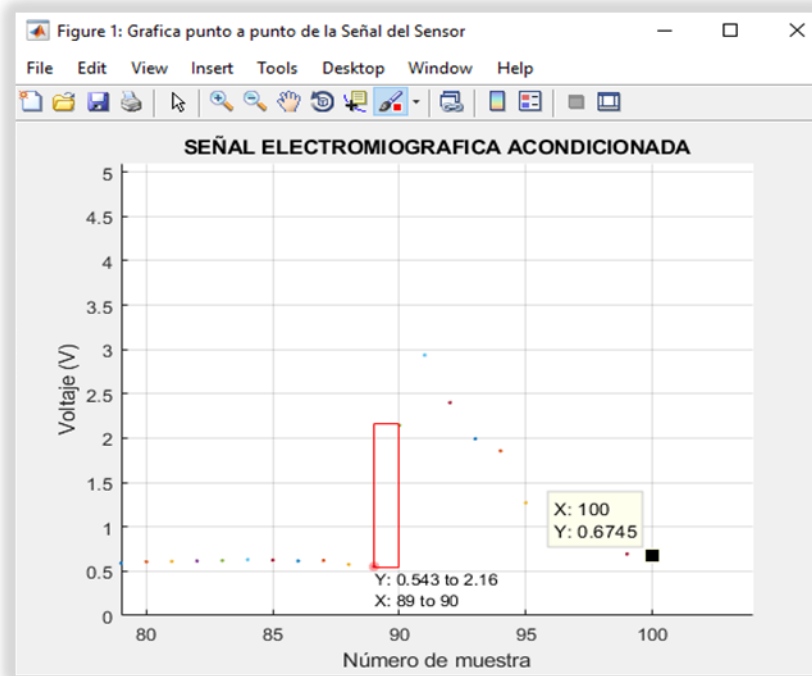
$$\text{Tiempo de inicio} = 15 * 0.12\text{seg} = 1.8\text{seg}$$



**Figura 39. Captura de pantalla Representación gráfica realizada gracias al bucle For. (LecturaMyoWare.m) señal Electromiográfica. Software MATLAB**

En la representación gráfica de la segunda sesión fue posible apreciar las separaciones entre los valores en el eje Y que representa el voltaje de acondicionamiento de la señal. Como ejemplo cuando se produce un movimiento repentino se aprecia un salto desde el dato 0,543V a 2.16V.

Considerando esto se tomó como prioridad la realización de la gráfica de una manera continua de tal manera que permita al usuario comprender de mejor manera la variación que se produce.



**Figura 40. Captura de pantalla Representación gráfica realizada gracias al bucle while (LecturaMyoWare.m) señal Electromiográfica. Software MATLAB**

**4.7.1 Creación de los vectores de datos pertenecientes a los diferentes movimientos.** Por medio del código anterior se dio inicio al almacenamiento de vectores de 500 muestras, cada uno de ellos representando un movimiento o tipo de movimiento experimental, posteriormente se realizó un análisis mediante la investigación de los diferentes tipos de patrones y características que las señales presentaron.

Mediante la observación en la toma de muestras se logra verificar que la Señal Electromiográfica Original (RAW) presentaba demasiado ruido y cambiaba constantemente, debido a que el valor máximo entregado por el sensor en el movimiento más intenso no sobrepasa los 700mV pico a pico ( $\pm 350\text{mV}$  Amplitud) y existen ruidos presentes de tamaño considerable. A pesar de que no es viable realizar la identificación del movimiento a partir una sola señal y menos

con las falencias que se presentan en la medición, se realizó una investigación sobre los patrones y características que presentan las señales Electromiográficas sin acondicionar y algunos métodos de evaluación de las mismas.

**Tabla 10. Métodos de evaluación de señales electromiográficas**

	<b>Característica</b>	<b>Comando en Matlab</b>
Valor Integrado de EMG (IEMG)	Hacer una sumatoria a lo largo de toda la ventana de segmentación.	integral
Valor Medio Absoluto (MVA)	Estima el promedio del valor absoluto de los datos	abs
Raíz media cuadrada (RMS)	Cuando la señal de EMG es modelada como un proceso aleatorio Gaussiano.	sqrt
Desviación estándar	Representa qué tan separados están los datos con respecto a la media y se expresa en las mismas unidades que esta última.	std
Cruces por cero (CZ)	Es el número de veces que la señal pasa por cero, es decir, el número de veces que la señal cambia de signo en un segmento	zeros
Cambios en la pendiente de la señal (SSC)	Es el cambio de signo de la pendiente de la señal y proporciona una medida de la frecuencia	findchangepts
Histograma de EMG	Divide los elementos de la señal de EMG en k segmentos igualmente espaciados y retorna el número de elementos en cada segmento	hist o histogram
Coefficientes Auto Regresivos	Esta característica modela la señal de EMG como una serie de tiempo lineal auto regresiva y provee información acerca del estado de contracción muscular	polyfit
Coefficientes Cepstrum	La señal cepstral se define como la transformada inversa de Fourier del logaritmo del espectro de potencia de la señal	pspectrum
Frecuencia mediana (FMD)	La frecuencia mediana divide el espectro de densidad de potencia (PSD) en dos.	medfreq
Frecuencia Media (FM)	Valor medio de frecuencia	meanfreq
Transformada de Fourier en tiempo corto (STFT)	Consiste en aplicar la transformada de Fourier en intervalos de tiempo cortos.	stft
Transformada Wavelet	Es una transformación donde una señal es integrada con una función o wavelet madre desplazada en tiempo y en escala.	wt continuo Dwt discrete
Transformada Hilbert-Huang	Esta técnica consiste en aplicar la transformada Hilbert a un grupo de señales obtenidas de la descomposición adaptativa de la señal original	hht

Fuente: Cifuentes, 2012.

Para el desarrollo de la medición y almacenamiento de las señales Electromiográficas iniciales el sujeto de prueba cuenta con las siguientes características:

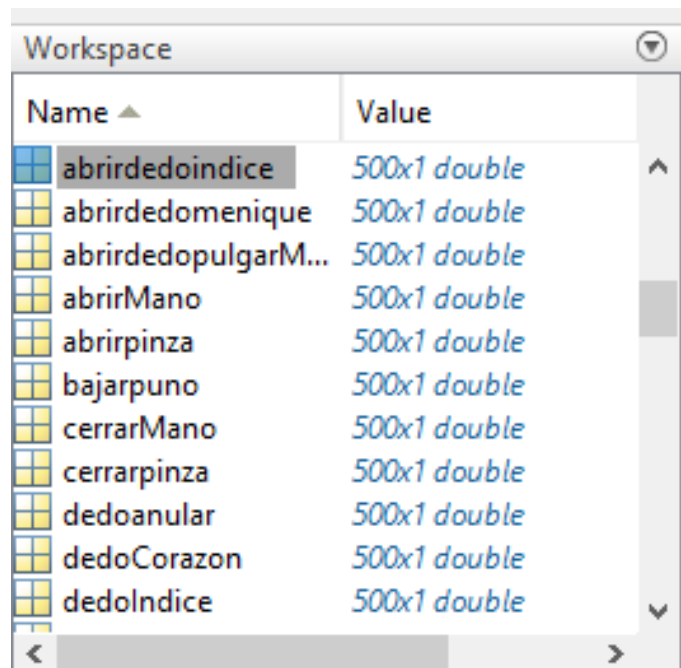


**Género:** Masculino **Altura:** 1.60m. **Compleción:** Delgada. **Peso:** 56kg.

**Estado de salud:** Sano **Amputaciones o traumatismos musculares:** No presenta.

Las variables que se almacenan en el Workspace son renombradas dependiendo del movimiento que se ejecutó en la toma de muestras, y guardadas en el equipo como archivos de extensión \*.m estas son del tipo **double**.

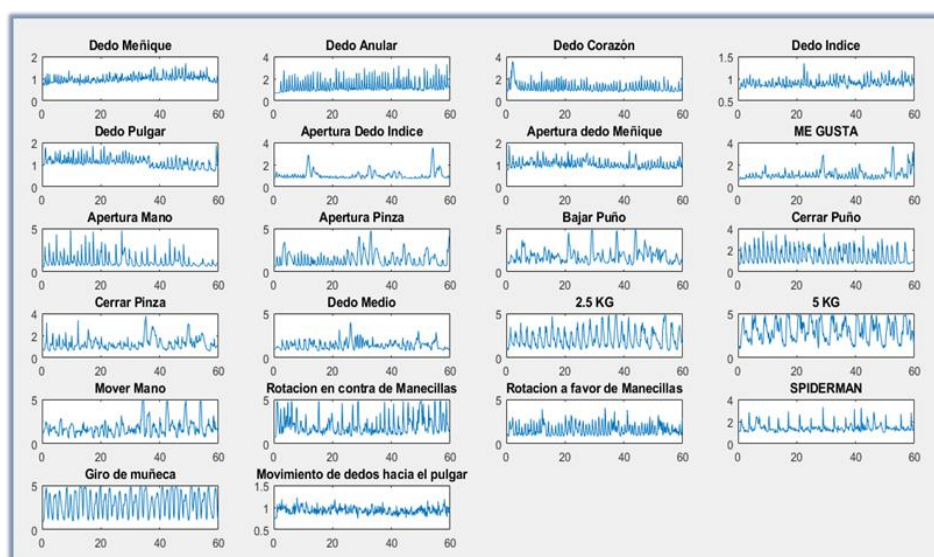
Es decir, datos numéricos predeterminado (clase) en MATLAB®, que proporciona precisión suficiente para la mayoría de las tareas computacionales. Las variables numéricas se almacenan automáticamente como valores de coma flotante de precisión doble de 64 bits (8 bytes). (Mathworks, 2019).



Name ▲	Value
abrirdedoindice	500x1 double
abrirdedomenique	500x1 double
abrirdedopulgarM...	500x1 double
abrirMano	500x1 double
abrirpinza	500x1 double
bajarpuno	500x1 double
cerrarMano	500x1 double
cerrarpinza	500x1 double
dedoanular	500x1 double
dedoCorazon	500x1 double
dedoIndice	500x1 double

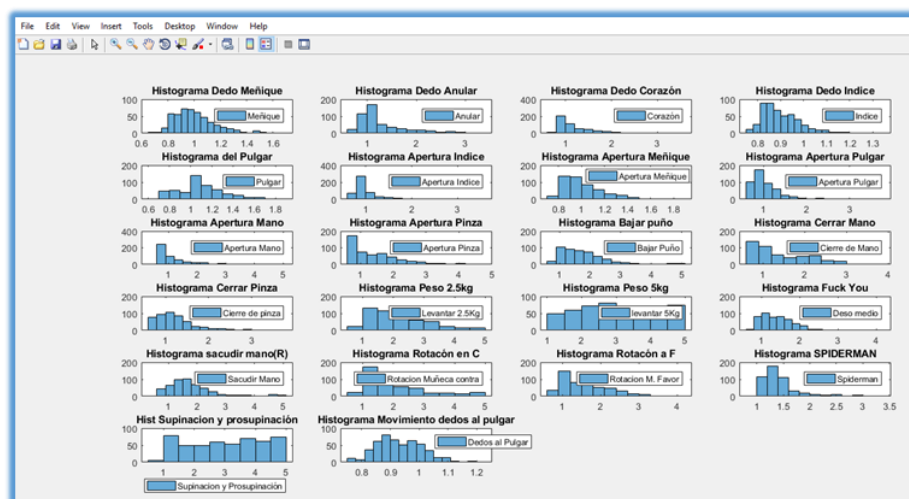
**Figura 41. Variables creadas a partir del script (LecturaMyoWare.m) señal Electromiográfica Software MATLAB**

Los vectores anteriores fueron graficados(**Graficademosmovimientos.m**) con respecto al tiempo, cada vector de 500 datos de voltaje dura un minuto (60seg) y contiene aproximadamente 50 movimientos repetitivos del nombre que lleva en su etiqueta, la figura que se presenta a continuación, es importante, porque al mostrar todas las gráficas a la vez se logran apreciar diferencias de amplitudes también es posible observar que se presenta una señal DC (offset) no deseada entre 0.5v-1v que disminuye la resolución e interfiere con la toma de efectiva. De igual manera, el equipo de trabajo mediante el ensayo y error determino que cuando el tamaño de muestras es muy alto, se empieza a producir retrasos en la toma de la señal y disminución del voltaje real de la misma.



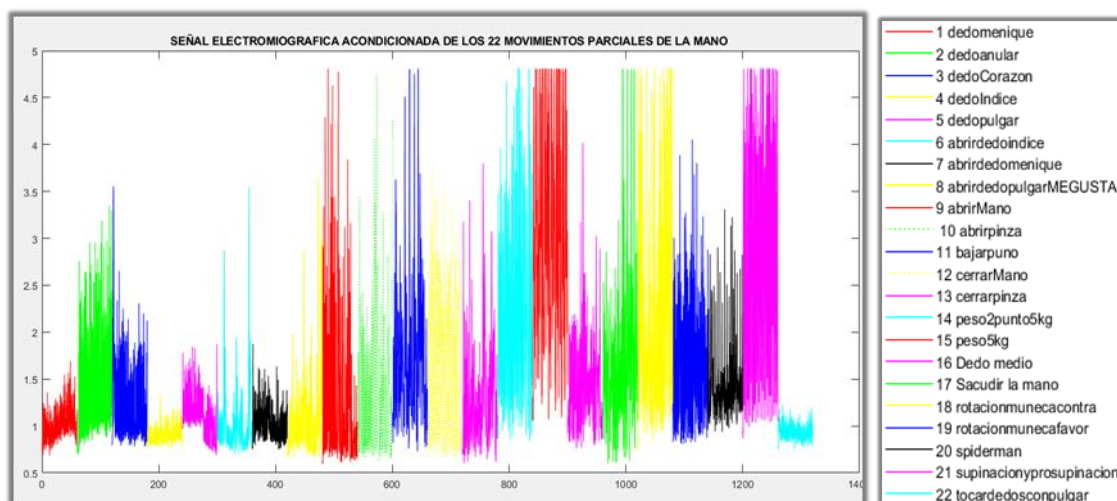
**Figura 42. Graficas de 22 señales EMG acondicionadas. Software MATLAB**

Mediante la aplicación del comando **Histogram** de MATLAB es posible obtener el Histograma (**Ejemplo:** Histogram (abrirMano)) de todos los movimientos que se realizaron, esto permite identificar la cantidad de muestras (eje y) según la Amplitud de voltaje (eje x) que posee cada señal.



**Figura 43. Histogramas de las señales EMG acondicionada. Software MATLAB**

Se realizó la unión de los vectores de movimientos individuales en un solo vector denominado **Smovitotales**, para aplicar sobre él la extracción de características implementando un nuevo Script de Matlab de nominado **PRUEBA50muestras.m** utilizando las funciones presentes en el Web-binario de Mathworks “Aprendizaje automático para análisis de datos de sensores” (MathWorks, 2014-2015).



**Figura 44. Gráfica Vector Smovitotales con etiquetas de identificación. Software MATLAB**

#### 4.8 Definición de Características Presentes en Las señales

Una vez que se obtuvo el vector de las señales se procedió a definir los patrones o características que se deseaba extraer de las mismas, mediante métodos comunes de procesados de señales (Conocidos por medio de la investigación del equipo de trabajo) para después realizar su identificación.

“En la práctica para cada segmento de vector debemos extraer un conjunto de medidas normalmente llamado features (características) a partir de las señales originales, estas características deben ser capaz de identificar las similitudes entre las señales que pertenecen a la misma clase y las diferencias.” (Paz Tárrega, MathWorks, 2018).

##### **Aprendizaje Automático para Análisis de Datos de Sensores:**

Para ello se usaron comandos propios de Matlab como (media(**mean**); integral trapezoidal (**trapz** ); raíz del valor medio (**rms**), pico mínimo(**min**) ,pico máximo(**max**), con el fin de crear variables de entrada para la Red Neuronal Artificial, todas estas características se seleccionan debido a que la **señal EMG ACONDICIONADA** trabaja de 0-5V como la mayoría de los sensores(Rango de voltaje).

##### **Media o valor promedio o media aritmética:**

En matemáticas y estadística se define como la suma de los valores que se tienen divididos en la cantidad de muestras. Dados los números  $n$   $\{x_1, x_2, \dots, x_n\}$  la media se define como:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$

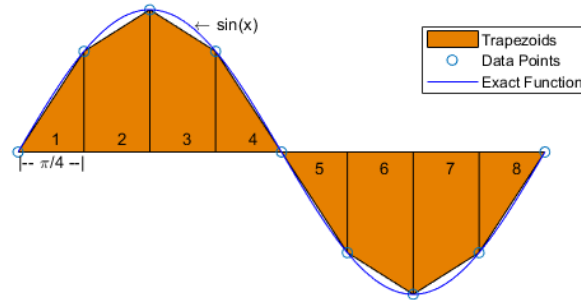
En Matlab para calcular la media de los elementos de un vector se utiliza el comando **mean**, para desarrollar este procedimiento automático se creó una pequeña función:

```
%calcular media
function rmedia = media(x)
if ~isvector(x)% si(if) no(~) es un vector
    error ('No es un vector')
end
rmedia =mean(x);
end
```

**Figura 45. Código 2**

### **Integración numérica trapezoidal:**

Comando trapz realiza la integración numérica mediante el método trapezoidal. Este método aproxima la integración sobre un intervalo rompiendo el área abajo en trapezoides con áreas más fácilmente comparables. Por ejemplo, aquí hay una integración trapezoidal de la función sinusoidal utilizando ocho trapezoides espaciados uniformemente:



**Figura 46. Integración trapezoidal**

Fuente: MathWorks

Para una integración con  $N+1$  puntos espaciados uniformemente la aproximación es:

$$\int_a^b f(x) dx \approx \frac{b-a}{2N} \sum_{n=1}^N (f(X_n) + f(X_{n+1}))$$

$$= \frac{b-a}{2N} (f(X_1) + 2f(X_2) + \dots + 2f(X_N) + f(X_{N+1}))$$

Donde el espacio entre cada punto es igual al valor escalar  $\frac{b-a}{N}$  de forma predeterminada,

Matlab utiliza un espaciado de 1. Si el espaciado entre los puntos  $N+1$  no es constante, la formula se generaliza a:

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{n=1}^N (X_{n+1} - X_n) (f(X_n) + f(X_{n+1}))$$

donde  $a = X_1 < X_2 < \dots < X_N < X_{N+1} = b$  y  $(X_{n+1} - X_n)$  es el espaciado entre cada par de puntos consecutivos. (MathWorks, 2018).

La siguiente función permite realizar el cálculo de la integración numérica trapezoidal en cualquier vector.

```
%calcular trapz (Integración numérica trapezoidal)
function rtrapz = area(x)
if ~isvector(x) % si (if) no (~) es un vector
    error ('No es un vector')
end
rtrapz = trapz(x);
end
```

**Figura 47. Código 3**

### **Raíz del valor medio (rms):**

También conocido como valor eficaz de una magnitud eléctrica. El concepto de valor eficaz se utiliza especialmente para estudiar las formas de onda periódicas, a pesar de ser aplicable a todas las formas de onda, constantes o no. En ocasiones se denomina con el extranjerismo RMS (del inglés, *root mean square*).

El nivel medio de la raíz media cuadrada de un vector es:

$$X_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N |X_n|^2}$$

$y = \text{rms}(x)$  Devuelve el nivel de la raíz-media-cuadrada (RMS) de la entrada X Si es un vector de fila o columna, es un valor escalar real, se aplicó de la siguiente manera:

```
%calcular raiz del valor medio (señal filtrada)
function rrms = valormediorms(x)
if ~isvector(x) % si(if) no(~) es un vector
    error ('No es in vector')
end
rrms = rms(x);
end
```

**Figura 48. Código 4**

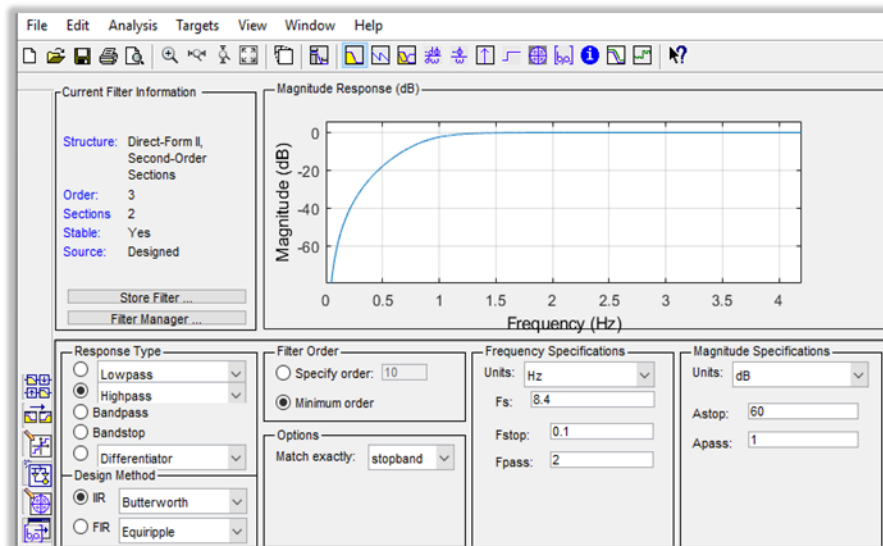
Contrario a este proyecto de ser posible trabajar con la señal Electromiográfica original (RAW) los procedimientos, métodos o ecuaciones para la extracción de las características que debían tener en cuenta son las planteadas ítem 4.7 en la tabla 10.

Con la revisión de los histogramas, se evidencio que las distribuciones de las señales eran muy similares por lo tanto se generó la necesidad de plantear la utilización de un análisis más avanzado. (En estadística, un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.).

Debido al ruido que se presentaba en la señal del sensor se llevó acabo la eliminación del voltaje en DC (corriente directa) no deseado, esto manera sencilla determinando a través de las pruebas de medición, las subidas de tensión (Magnitud del offset), después se procedió restar este valor a los distintos vectores mejorando el rango de medición de 0.9V-5V a 0.25V-5V.



Se vio la necesidad de aplicar un filtro digital que lograra suprimir los puntos donde la frecuencia es menor, cuando se presentaban variaciones en la toma de muestras el sujeto de prueba evitaba realizar movimiento con el fin de que el ruido y la señal no se sumaran, la amplitud del voltaje cambia considerablemente en ese punto, pero pudo observarse que en los picos altos el ancho en el eje X (muestras) era mayor, además se lleva a realizar un cruce por cero a toda la señal. Para la elaboración del mismo se usó la **APP de Matlab** denominada **Filter Designer**, que presenta una interfaz gráfica de fácil acceso y muy cómoda para trabajar, donde es posible verificar la respuesta del filtro diferentes entradas como el escalón o un impulso. Además, facilito la generación del código de la función que realiza el proceso de Filtrado de la señal EMG acondicionada.



**Figura 49. Diseño de filtro para la señal EMG acondicionada mediante la App Filter Designer. Software MATLAB**

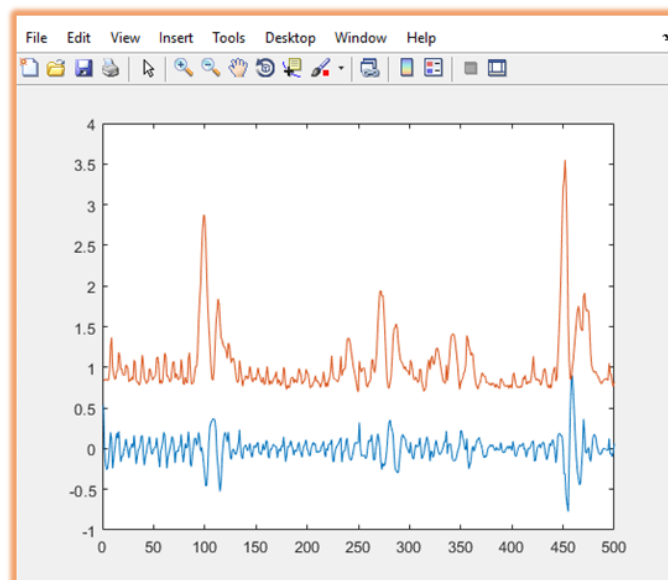
```

Editor - C:\Users\alex_\OneDrive\Escritorio\PROYECTO DE GRADO FINAL\Programación\Script MATLAB\Filtro de
EDITOR PUBLISH VIEW
+69 filtrado.m x filtrador2.m x forlmean.m x Graficadeidentificacion.m x Graficasdehistogr
1 function Hd = filtrado
2 %FILTRADO Returns a discrete-time filter object.
3 % Chebyshev Type II Highpass filter designed using FDESIGN.HIGHPASS.
4
5 % All frequency values are in Hz.
6 Fs = 8.4; % Sampling Frequency
7
8 Fstop = 0.1; % Stopband Frequency
9 Fpass = 2; % Passband Frequency
10 Astop = 60; % Stopband Attenuation (dB)
11 Apass = 1; % Passband Ripple (dB)
12 match = 'stopband'; % Band to match exactly
13
14 % Construct an FDESIGN object and call its CHEBY2 method.
15 h = fdesign.highpass(Fstop, Fpass, Astop, Apass, Fs);
16 Hd = design(h, 'cheby2', 'MatchExactly', match);
17

```

**Figura 50. Función de Filtrado generada a partir App Filter Designer. Software MATLAB**

El filtro se aplicó a toda la señal de datos Smovitotales, a continuación, se presenta la variación que se produce (Como ejemplo la señal movimiento del dedo índice). En la señal global se observa un comportamiento transitorio que era esperado.



**Figura 51. Gráfica de vector “apertura índice” sin filtrar (naranjado) y filtrada (Color azul). Software MATLAB**

**4.8.1 Medición de parámetros que miden la forma y la oscilación de las señales.** Debido a la manera en que el sujeto de prueba efectuó los movimientos las señales que tomamos realizan oscilaciones prácticamente de manera regular y periódica.

Con respecto a esto la página oficial de MATLAB (Mathworks.com), en sus diferentes seminarios web y documentación propone algunos métodos como la representación espectral de la señal y densidad espectral de potencia de la misma, también la utilización de Apps y toolbox como **Signal Processing** para como su nombre lo indica procesar o pre procesar las señales.

Con el fin de calcular la densidad espectral de potencia de la señal se utilizó el método Welch en Matlab se usa como comando pwelch. Denominado periodograma de Welch en el que la DEP es estimada aplicando la FFT directamente sobre los datos y realizando un posterior promediado.

#### **Periodograma de Welch:**

El periodograma de Welch utiliza una definición de la densidad espectral de potencia para procesos ergódicos basada en un promediado temporal. El periodograma o sample spectrum se calcula directamente a partir de N muestras del segmento de señal miográfica  $x(n)$  mediante la siguiente expresión [Marple, 1987]: (Mañanas, 1999).

$$\hat{P}_{ss}(f) = \frac{1}{NT} \left| T \sum_{n=0}^{N-1} x(n) e^{-j2\pi fnT} \right|^2 \quad -\frac{1}{2T} \leq f \leq \frac{1}{2T}$$

Mediante la utilización del comando pwelch es posible aplicar el método Welch para estimar la densidad espectral de potencia, el código empleado para realizar esta tarea es el siguiente.

```

5 %Densidad Espectral de potencia.
6 - figure
7 - plot(f,pxx)
8 - title('La estimación de densidad espectral de potencia de Welch')
9 - grid on
10 - xlabel('Frecuencia (Hz)') % tenemos la frecuencia y se obtiene Fs/2=4.2
11 - ylabel('PSD (dB/Hz)') % Se obtienen decibelios/Hz o densidad espectral de potencia.
12 - [pxx,f] = pwelch(Filsmovitotales,[],[],[],fs); %se realiza en de la señal completa

```

**Figura 52. Código comando pwelch aplicado a toda la señal filtrada Software MATLAB**

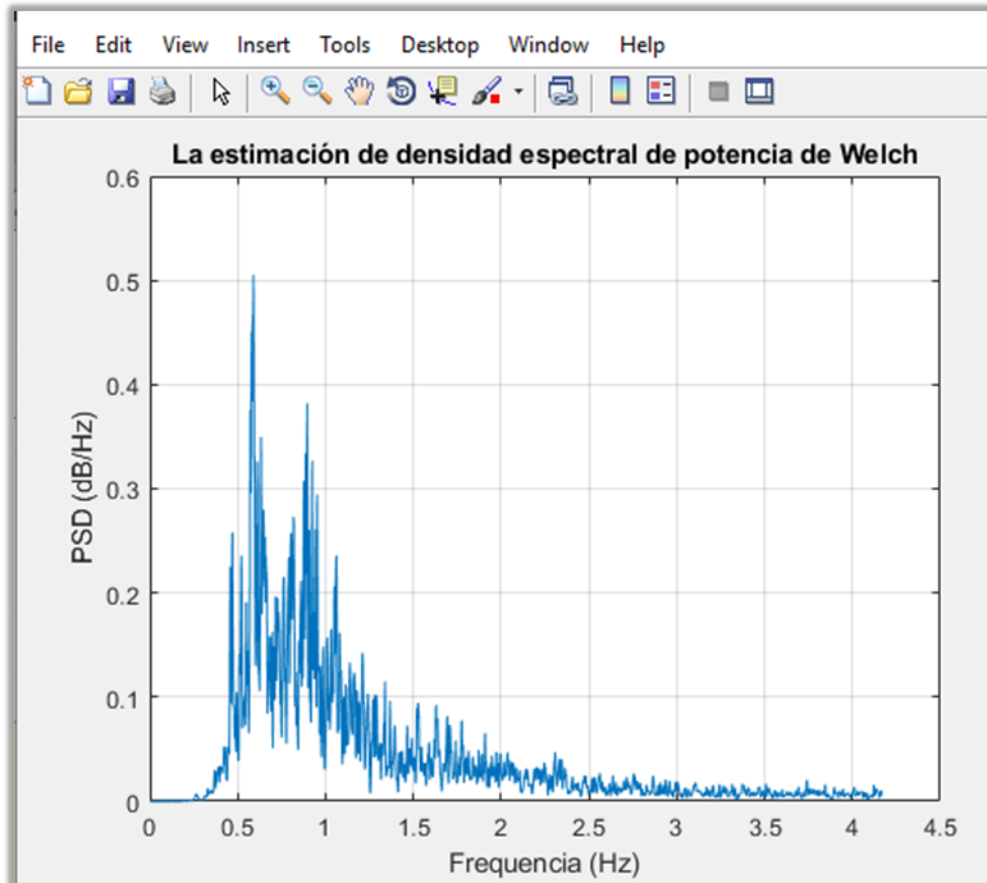
La región en la cual los picos que produce la gráfica son altos es muy probable que

Contengan la información necesaria para extraer características. Para la señal de los movimientos totales se midieron valores entre los puntos 0.38 y 0.5 Db/Hz. A partir de este, se logró apreciar en la gráfica que la frecuencia fundamental esta entre 0.6 y 1Hz la distancia en frecuencia entre estos picos, es la tasa de oscilación en el dominio del tiempo Y la amplitud relativa de los picos describe las formas de las oscilaciones.

En el eje de las x obtenemos la frecuencia:

$$F = \frac{Fs}{2} = \frac{8.4}{2} = 4.2hz$$

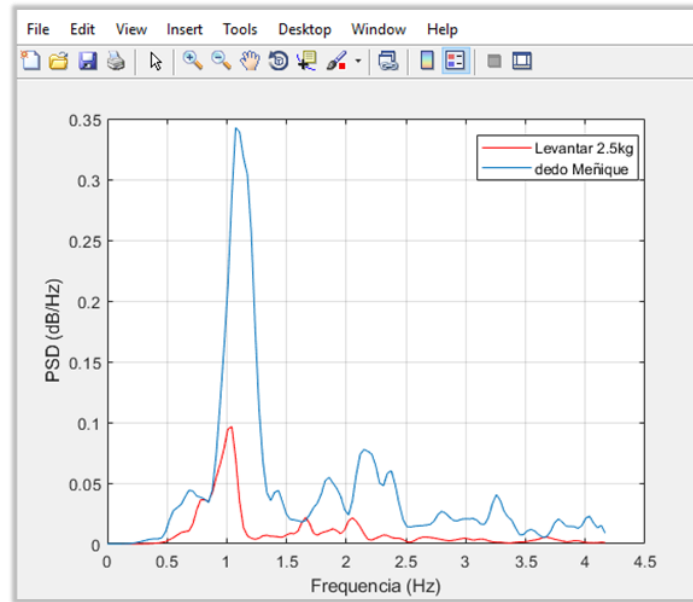
Y en el eje vertical la magnitud de la densidad de potencia espectral (Decibeles/Hz).



**Figura 53. Estimación espectral de potencia de Welch aplicada a toda la señal filtrada**

### **Software MATLAB**

También se hizo posible realizar una verificación de algunas señales individuales y observar su comportamiento, la mayoría mostraba los picos máximos de la densidad de potencia cuando la frecuencia es de 1HZ. La suavidad en el dominio del tiempo se nota si los picos decrecen muy rápido.



**Figura 54. PSD (Power Spectral Density) de Welch comparación entre dos movimientos Software MATLAB**

Una vez que se conocieron los valores más altos que alcanza de la densidad de potencia espectral, y se determina estos son los que posean la información necesaria, gracias a las herramientas que posee Matlab es posible extraer estos picos de manera efectiva en este caso se utilizara la función **findpeaks**, si se utiliza solo aplicando el comando muestra todos los picos que posee la señal del SPD.

Con el fin de implementar un código ya funcional en la extracción de característica de señales de sensores se optó por utilizar algunas de las funciones ya definida en el seminario web,

Entre las cuales se encuentran las siguientes: (se da todo el reconocimiento y crédito por la creación de las mismas según corresponda al seminario web de Matlab “Aprendizaje automático para análisis de datos de sensores” de Mathworks.com (Copyright 2014-2015 The MathWorks, Inc.).

### Función para extraer picos característicos del SPD:

La función es llamada `spectralPeaksFeatures` se almacenada en un Script. (\*.m) permite extraer Características del pico espectral (12 cada una): altura y posición de los primeros 6 picos.

*`feats = spectralPeaksFeatures(x,fs)`*

### Código:

```
function feats = spectralPeaksFeatures(x, fs)
n_channels = size(x,2);
mindist_xunits = 0.3;

feats = zeros(1,6*n_channels);

N = 4096;
minpkdist = floor(mindist_xunits/(fs/N));
opks = 0;

% Cycle through number of channels
for k = 1:n_channels
    [p, f] = periodogram(x(:,k),rectwin(length(x)),4096,fs);
    [pks,locs] =
findpeaks(p,'npeaks',20,'minpeakdistance',minpkdist);
    if (~isempty(pks))
        mx = min(3,length(pks));
        [spks, idx] = sort(pks,'descend');
        slocs = locs(idx);
        pks = spks(1:mx);
        locs = slocs(1:mx);
        [slocs, idx] = sort(locs,'ascend');
        spks = pks(idx);
        opks = spks;
        locs = slocs;
    end
    ofpk = f(locs);

    % Features 1-6 positions of highest 6 peaks
    feats(6*(k-1)+(1:length(opks))) = ofpk;

    % Features 7-12 power levels of highest 6 peaks
    feats(6*(k-1)+(4:4+length(opks)-1)) = opks;
end
end
```

**Figura 55. Código comando `pwelch` aplicado a toda la señal filtrada Software MATLAB**

Para la aplicación de la misma se modificaron algunos parámetros para que se pueda obtener solo 6 datos de los picos espectrales 3 de altura y 3 de posición, los comandos que se utilizaron principalmente para esta función son los siguientes, pero la función completa permite mejorar los parámetros en su aplicación.

**[p, f] = periodogram (\_\_, fs)**

Devuelve un vector de frecuencia, en ciclos por unidad de tiempo (frecuencia de muestreo Fs) Si la unidad de tiempo es de segundos, entonces está en ciclos/segundo (Hz).

**[ pks , locs ] = findpeaks( x)**

Devuelve un vector con los máximos picos del vector de señal de entrada, x. Un *pico local* es una muestra de datos que es más grande que sus dos muestras vecinas adicionalmente entrega los índices en los que ocurren los picos.

### **Función para extraer características de auto correlación:**

La función recibe el nombre de AutocorrFeatures se encuentra almacenada en un Script. (\*.m) esta permite obtener la altura del pico principal; Altura y posición del segundo pico.

***feats = autocorrFeatures(x,fs)***



**Código:**

```

function feats = autocorrFeatures(x,fs)
n_channels = size(x,3);
feats = zeros(1,3*n_channels);

[c,lags]
=arrayfun(@(i)xcorr(x(:,i)),1:n_channels,'UniformOutput',false);

minprom = 0.0005;
mindist_xunits = 0.3;
minpkdist = floor(mindist_xunits/(1/fs));
% Separate peak analysis for all channels
for k = 1:n_channels
    [pks,locs] = findpeaks(c{k},...
    'minpeakprominence',minprom,...
    'minpeakdistance',minpkdist);

    tc = (1/fs)*lags{k};
    tcl = tc(locs);

    % Feature 1 - peak height at 0
    if(~isempty(tcl)) % else f1 already 0
feats(n_channels*(k-1)+1) = pks((end+1)/2);
    end
    % Features 2 and 3 - position and height of first peak
    if(length(tcl) >= 2) % else f2,f3 already 0
feats(n_channels*(k-1)+2) = tcl(1);
feats(n_channels*(k-1)+3) = pks(1);
    end
end
end

```

**Figura 56. Código 5**

Esta función utiliza el comando **xcorr**

**r = xcorr(x, y)** o **[ r , lags ] = xcorr( \_\_\_ )**

“Este permite devolver la correlación cruzada de dos secuencias de tiempo discreto. La correlación cruzada mide la similitud entre un vector  $x$  y copias desplazadas (rezagadas) de un vector y en función del rezago. Si y tienen diferentes longitudes, la función agrega ceros al final del vector más corto para que tenga la misma longitud que el otro.” (mathworks.com/help, 2019)

### **Función para extraer características de potencia espectral:**

La función se denomina `spectralPowerFeatures` y se encuentra guardada en un Script (\*.m) permite extraer características de potencia espectral (5 cada una): potencia total en 5 bandas de frecuencias adyacentes y predefinidas.

***feats = spectralPowerFeatures(x,fs)***

### **Código:**

```
function feats = spectralPowerFeatures(x, fs) %
n_channels = size(x,2);

edges = [0, 0.5, 1, 1.5, 2, 2.5];
n_feats = length(edges)-1;
feats = zeros(1,n_feats,n_channels);

for k=1:n_channels
    [p, f] = periodogram(x(:,k),rectwin(length(x)),4096,fs);
    for kband = 1:n_feats
        feats(n_feats*(k-1)+kband) = sum(p( (f>=edges(kband)) &
(f<edges(kband+1)) ));
    end
end
end
```

**Figura 57. Código 6**

(Las funciones que se presentaron anteriormente fueron concatenadas con el Script final para que la extracción de características se realice de forma automática.)

Es esencial mencionar que la extracción de características por medio de los comandos y las funciones no en todos los casos se aplicó al vector original (Smovitotales), al vector filtrado, o al vector de estimación potencia espectral, para dejar claro el orden en que desarrollaron las operaciones y cálculos se presenta la siguiente tabla:

**Tabla 11. Orden de operaciones**

<b>CARACTERÍSTICA A EXTRAER</b>	<b>COMANDO O FUNCION</b>	<b>VECTOR ANALIZADO</b>
Media	Mean	Señal EMG acondicionada.
Integral trapezoidal	Trapz	
Altura del pico principal y altura y distancia del segundo pico.	Función AutocorrFeatures	
Raíz del valor medio (RMS)	Rms	Señal EMG filtrada
Picos espectrales 3 de altura y 3 de posición,	Función spectralPowerFeatures	Vector resultado de aplicar el método de Welch o periodogram a la señal EMG ya filtrada para obtener la densidad de potencia espectral.
Potencia total en 5 bandas de frecuencias adyacentes y predefinidas.	Función spectralPowerFeatures	
Vector del movimiento.	Bucle While (Código final)	Datos de la señal

Al obtener los distintos parámetros mencionados anteriormente se crearon una tabla y una matriz, con el objetivo de empezar a desarrollar los algoritmos de clasificación, para ello se utilizaron tanto la App de Matlab Classification Learner, y la caja de herramientas (toolbox) de nntool.

### **App Classification Learner:**

Con el fin de verificar si los distintos datos que se extrajeron de las señales permitían desarrollar un algoritmo funcional para la identificación del movimiento, se procede a usar esta

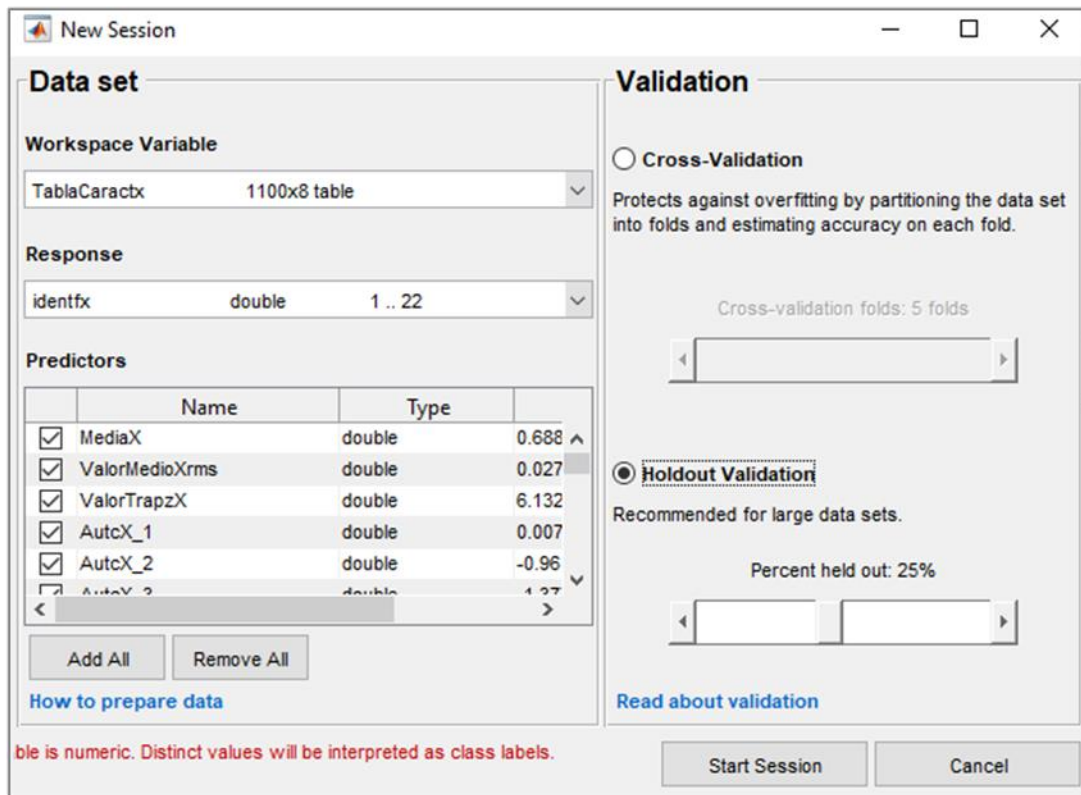
App debido a que presenta una manera sencilla y rápida de validación, permitiendo la aplicación de diversos tipos de clasificadores.

Fue necesario crear una tabla con todas las características (**TablaCaractx.mat**) obtenidas y agregar una última columna de identificación, en esta se otorga un número que identifica a que movimiento pertenecen esas características quedando así con una dimensión de 1100 filas por 8 columnas de etiquetas (cada columna de etiqueta contiene columnas adicionales según la cantidad de características extraídas. Ejemplo la etiqueta de **AutcX** contiene 3 columnas de datos obtenidos).

	1 MediaX	2 ValorMedioXrms	3 ValorTrapzX	4 AutcX			5 SpkX						
1	0.8174	0.2461	7.2095	0.6054	-0.9600	0.1367	0.9318	1.9918	2.9256	0.0636	0.0183	0.0092	
2	0.8418	0.1571	7.5854	0.2469	-0.7200	0.0325	0.0020	1.2878	4.1646	0.0012	0.0160	0.0040	
3	0.8643	0.1303	7.8491	0.1697	-0.7200	0.0196	1.2573	2.4963	4.1646	0.0086	0.0054	0.0033	
4	0.9189	0.1254	8.3130	0.1572	-0.6000	0.0289	1.5808	3.4912	0	0.0116	0.0011	0	
5	0.9312	0.1433	8.3911	0.2053	-0.7200	0.0930	1.3794	2.6326	4.1646	0.0126	0.0121	0.0084	
6	0.9048	0.1311	8.2324	0.1720	-0.8400	0.0377	0.0020	1.1882	3.6499	0.0017	0.0174	0.0021	
7	0.9287	0.1151	8.3716	0.1324	-0.8400	0.0377	0.0020	1.0681	3.3203	0.0017	0.0151	0.0016	
8	0.9634	0.1356	8.7573	0.1838	-0.6000	0.0161	0.0020	1.2268	3.6214	0.0016	0.0105	0.0060	
9	0.9785	0.0845	8.8818	0.0714	-0.9600	0.0032	0.9338	2.3926	4.1646	0.0065	7.0563e-04	0.0015	
10	0.8750	0.0867	7.9468	0.0752	-0.8400	0.0031	1.1597	3.5970	0	0.0059	1.9469e-04	0	
11	0.9390	0.1460	8.3960	0.2130	-0.9600	0.0641	1.0335	2.1729	3.1616	0.0085	0.0096	0.0108	
12	0.8955	0.0671	8.0469	0.0450	-0.9600	0.0082	0.8830	1.9552	4.1646	0.0033	0.0024	6.9189e-04	

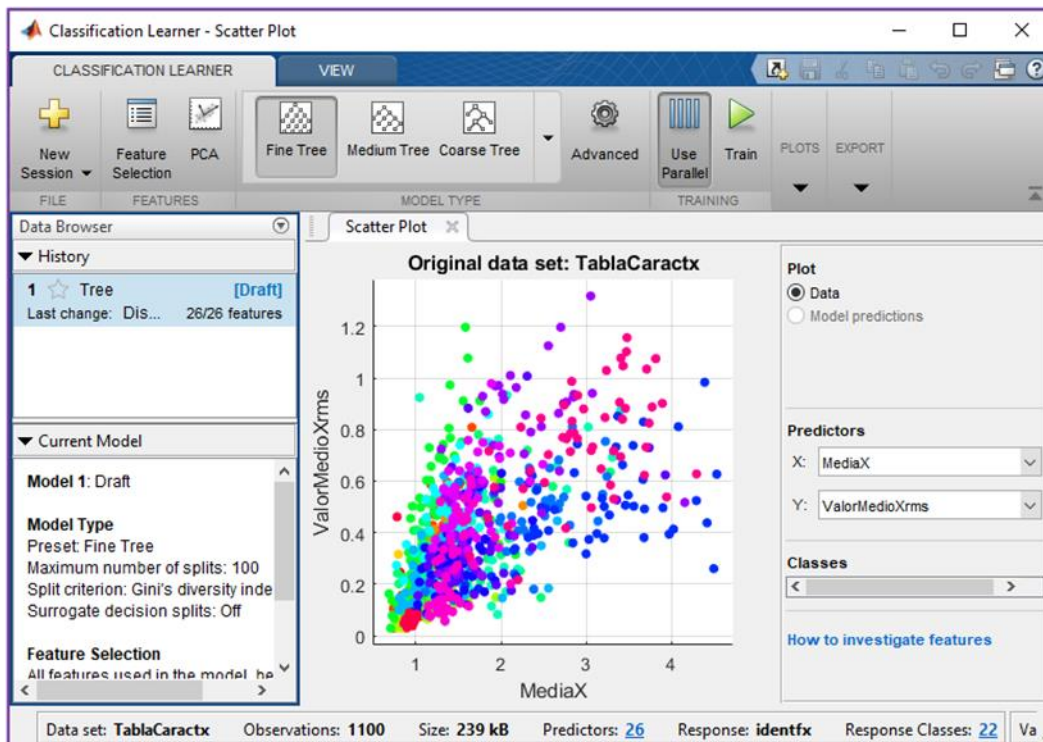
**Figura 58. Tabla de características extraídas de la señal Electromiográfica acondicionada movimientos. Software MATLAB**

Se importaron los datos del Workspace a la app y se selecciona el método de validación recomendado para grandes conjuntos de datos (**Holdout Validation**) con un porcentaje de retención de salida de 25%. Dando inicio a la sesión.



**Figura 59. Tabla de características cargada y elección de validación en App Classification Learner. Software MATLAB**

La interfaz permite la visualización grafica de las distintas variables originales que se tienen en la tabla.



**Figura 60. Tabla de características cargada y elección de validación en App Classification Learner Software MATLAB**

Se ejecutó la elaboración de todos los tipos de algoritmos posibles. Con el fin de observar el porcentaje de Exactitud (**Accuracy**) que estos entreguen. Esto tarda algunos minutos dependiendo de las especificaciones que posee la máquina de computo en la que se trabaja (especificada en la tabla elementos utilizados) mientras se abre y ejecuta de manera automática el **Parallel pool**.

“Un grupo paralelo es un conjunto de trabajadores de MATLAB<sup>®</sup> en un clúster de computación o escritorio. De forma predeterminada, un grupo paralelo se inicia automáticamente cuando lo necesitan, permite resolver problemas con un uso intensivo de cálculos y datos mediante procesadores multinúcleo.” (mathworks.com/help, 2019).

Se selecciona el clasificador que entrega mayor exactitud y se procede a generar código automático (guardarlo en una función) o encapsulado (Enviarlo a workspace).

Item	Algorithm	Accuracy	Last change	Features
1.1	Tree	99.3%	Fine Tree	26/26 features
1.2	Tree	94.5%	Medium Tree	26/26 features
1.3	Tree	21.8%	Coarse Tree	26/26 features
1.4	Linear Discriminant	Failed	Linear Discriminant	26/26 features
1.5	Quadratic Discriminant	Failed	Quadratic Discriminant	26/26 features
1.6	SVM	100.0%	Linear SVM	26/26 features

**Figura 61. Ventana Data browser con los diferentes algoritmos funcionales para identificación de App Classification Learner. Software MATLAB**

Como se aprecia en la imagen anterior después de la ejecución en búsqueda de algoritmos funcionales se obtienen algunas alternativas con exactitudes bastante altas, algunos de los métodos de clasificación que realizaron una validación correcta de los datos son:

### **SVM (Maquinas de soporte vectorial)**

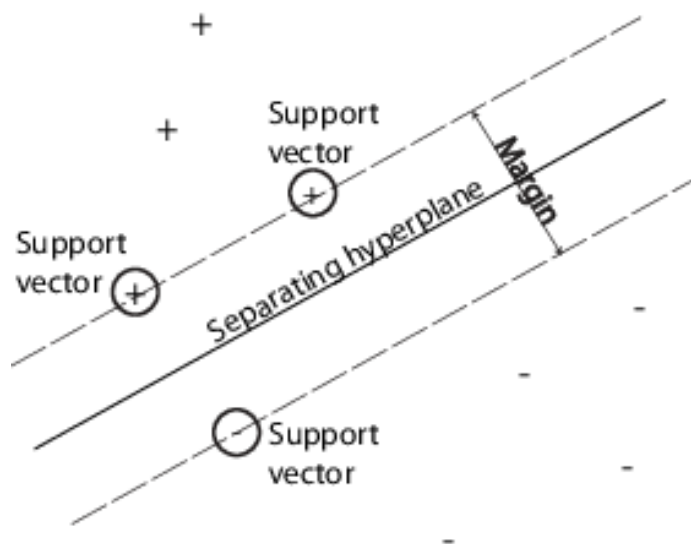
Las **máquinas de soporte vectorial**, son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T.

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y

entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra.

Un SVM clasifica los datos encontrando el mejor hiperplano que separa los puntos de datos de una clase de los de la otra clase. El mejor hiperplano para un SVM significa el que tiene el margen más grande entre las dos clases. Margen significa la anchura máxima de la losa paralela al hiperplano que no tiene puntos de datos interiores.

Son los puntos de datos más cercanos al hiperplano separador; estos puntos están en el límite de los vectores de soporte. (mathworks, 2019).



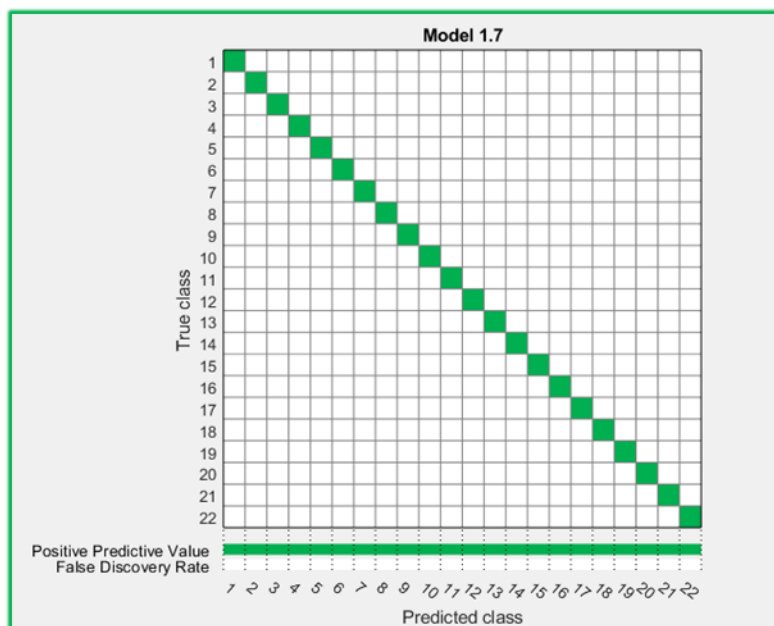
**Figura 62. Ilustra estas definiciones, con + indicando puntos de datos de tipo 1 y – indicando puntos de datos de tipo – 1.”. Software MATLAB**

Fuente: Web Mathworks.com, 2019.

La App también permite visualizar el desempeño del algoritmo generado por medio de diversas técnicas graficas una de estas es la matriz de confusión.



En el campo de la inteligencia artificial una **matriz de confusión** es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.



**Figura 63. Matriz de confusión del desempeño del algoritmo, Linear SMV (modelo 1.7) con los 22 movimientos iniciales. Software MATLAB App Classification Learner**

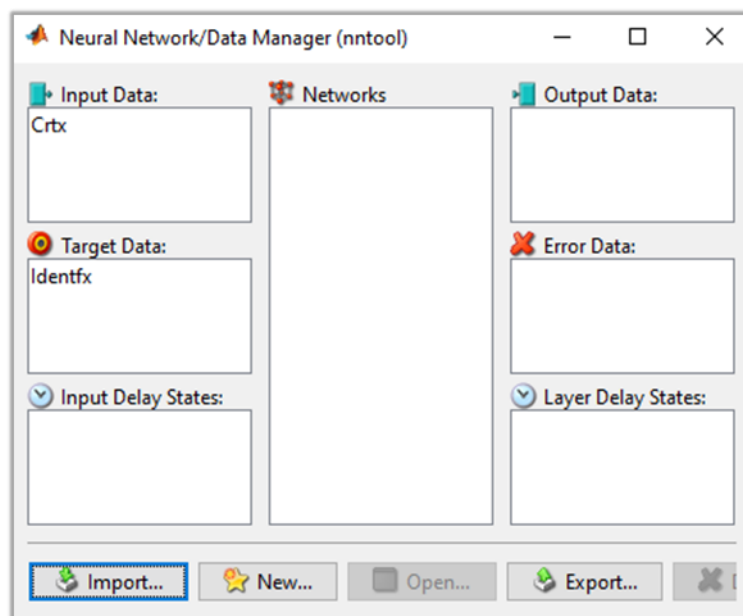
Con respecto a estos aparecen los siguientes **Linear, Quadratic y Cubic , Coarse gaussian SVM**, también fue posible obtener otros tipos algoritmos con exactitud alta pero solo se tomó uno (Linear SMV) pues el objetivo era realizar una validación de datos y ver qué tan factibles eran para su uso en un identificador.

La información necesaria para la implementación y uso del Classification Learner de manera superficial se obtuvo de la ayuda que brinda el propio software MATLAB y su página web:

<https://es.mathworks.com/help/stats/choose-a-classifier.html>

### **Caja de herramientas (toolbox) nntools:**

Con el fin de crear la red neuronal artificial que permita clasificar los movimientos dependiendo de las características que extraen de las señales Electromiográficas acondicionadas que son entregadas por el sensor, se utilizó la toolbox de matlab denominada nntool, esta permite cargar nuestros datos de entrada como una matriz de medidas (m,n) siendo m las filas y n las columnas, es necesario tener en cuenta que el número de filas de la matriz define la cantidad de parámetros de entrada de la red neuronal artificial (RNA).



**Figura 64. Toolbox nntool con los datos cargados Matriz Crtx(Input Data), Vector Identfx(Target Data). Software MATLAB App**

Se creó un vector de identificación de la misma longitud que poseen los vectores de movimientos en la matriz de características, en este se representan con un número cada

movimiento a diferencia del método anterior en el cual era necesario agregarlo directamente en la tabla de datos, para trabajar en la nntool se debe ingresar este vector como el objetivo o target, en pocas palabras es la respuesta que se espera obtener. (Vector **Identfx.mat** de dimensiones **(1100x1) tipo double**)

Después de cargar los datos, se crea la neuronalnetwork y se muestran las propiedades que deben tenerse en cuenta para el desarrollo de la misma, entre estas se encuentran las siguientes:

**Network type:** feed forward backprop

**Input data:** Datos de entrada.

**Target data:** Datos objetivos o respuesta esperada.

**Training function:** Función de entrenamiento.

**Adaptation learning function:** Función de aprendizaje de adaptación.

**Performance funtion:** Función de rendimiento.

**Number of layers:** Numero de capas de la RNA.

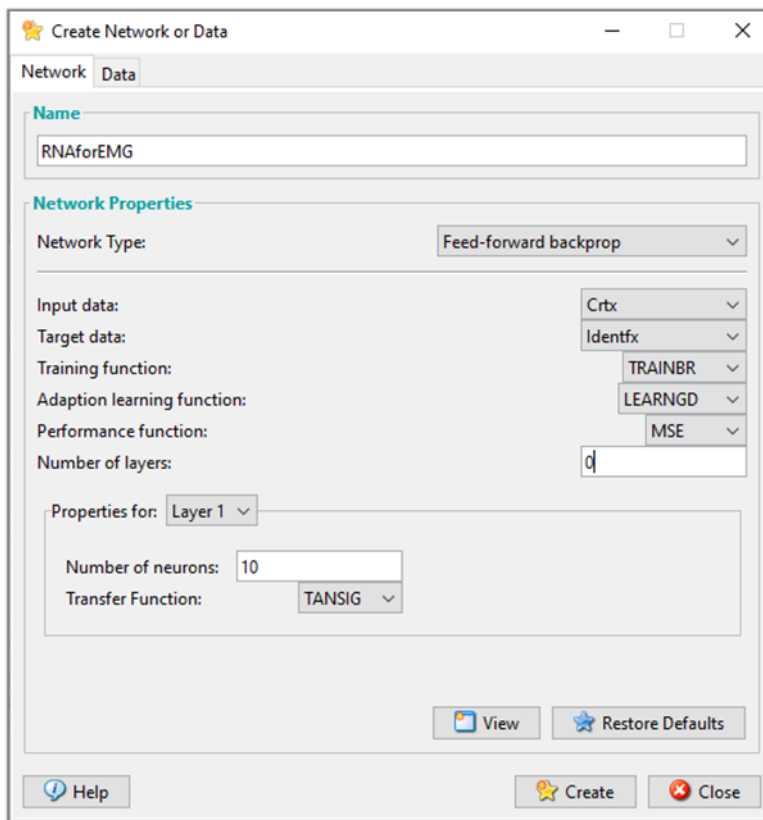
**Properties for layers:** Propiedades de las capas.

**Number of neurons:** Numero de neuronas.

**Transfer function:** Funcionen de transferencia.

El entrenamiento de la Red neuronal artificial se realizó mediante método de ensayo y error variando algunas de las propiedades anteriormente mencionadas y realizando comparaciones

entre el vector objetivo y los datos de salida y de error que eran entregados después de cada entrenamiento de la red.



**Figura 65. Ventana de creación de la Red neuronal de datos Crtx (Input Data), Vector Identfx (Target Data) Software MATLAB App**

Después de realizar una cantidad de pruebas considerable se encontró que la configuración de los parámetros que permitían generar una RNA funcional a partir de los datos de características que se extraen de las señales Electromiográficas acondicionadas son las siguientes:

**Input data:** Matriz de datos de entrada (Matriz\*.mat)

**Target data:** Vector de identificación (Vector\*.mat)

**Number of layers:** cantidad de capas 2

**Number of neurons:** 10 neuronas (capa oculta)

**Network type:** feed forward backprop:

El tipo de red neuronal de propagación hacia atrás de errores o retro propagación se encuentra explicada en el capítulo 2.2.11.

**Training function:** Bayesian regularización.

trainbr es una función de entrenamiento en red que actualiza los valores de peso y sesgo según la optimización de Levenberg-Marquardt. Minimiza una combinación de errores al cuadrado y pesos, y luego determina la combinación correcta para producir una red que generalice bien. El proceso se llama regularización bayesiana. (mathworks.com/help, 2019).

**Adaptation learning function:** LEARNGD

[dW,LS] = learngd(W,P,Z,N,A,T,E,gW,gA,D,LP,LS) toma varias entradas:

W	Matriz de peso S -by- R (o vector de sesgo S -by- 1 )
P	Vectores de entrada R -by- Q (o ones(1,Q) )
Z	Gradiente de salida S -by- Q con respecto al rendimiento x vectores de entrada ponderados Q
N	S -by- Q vectores de entrada neta
A	Vectores de salida S -by- Q
T	S -by- vectores objetivo de la capa Q
E	S -by- vectores de error de capa Q
gW	Gradiente S -by- R con respecto al rendimiento
gA	Gradiente de salida S -by- Q con respecto al rendimiento
D	S -by- S neurona distancias
LP	Parámetros de aprendizaje, ninguno, LP = []
LS	Estado de aprendizaje, inicialmente debe ser []

**Figura 69. Variables de entrada tomadas por el comando learngd Software MATLAB App**

Fuente: Web Mathworks.com, 2019.

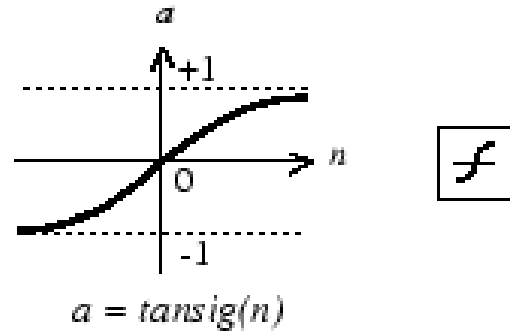
Learngd es la función de aprendizaje de inclinación y peso descendente en gradiente.

Learngd calcula el cambio de peso  $dW$  para una neurona dada a partir de la entrada  $P$  de la neurona y el error  $E$ , y la tasa de aprendizaje de peso (o sesgo)  $LR$ , de acuerdo con el descenso del gradiente  $dw = lr * gW$ . (mathworks.com/help, 2019).

**Performance funtion:** Mean Square error

Error Cuadrático Medio (MSE): expresa la diferencia entre las salidas correctas y las proporcionadas por red; la aproximación es mejor si MSE es más pequeño (más cercano a 0).

**Transfer function:** TANSING (Función de transferencia sigmoidea tangente hiperbólica).



Tan-Sigmoid Transfer Function

**Figura 70. Gráfico, comando y símbolo de la función de transferencia sigmoidea tangente hiperbólica.**

Fuente: Mathworks.com, 2019.

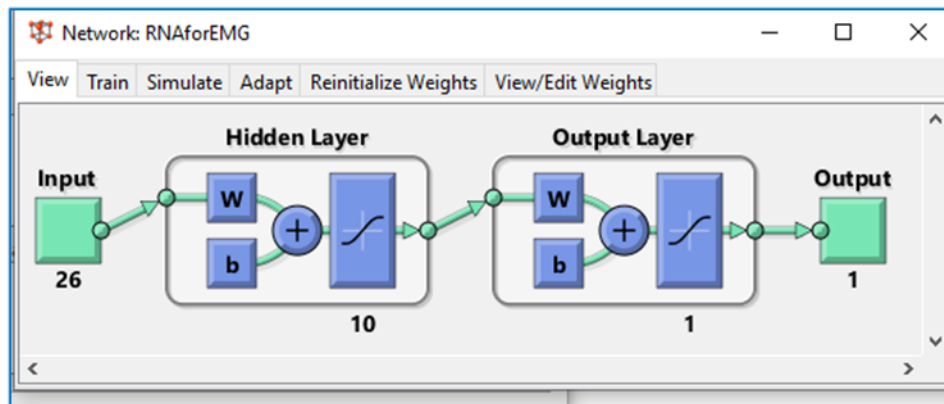
**tansig** es una función de transferencia neural. Las funciones de transferencia calculan la salida de una capa a partir de su entrada neta.

$$\mathbf{a} = \text{tansig}(\mathbf{n}) = 2 / (1 + \exp(-2 * \mathbf{n})) - 1$$

Esto es matemáticamente equivalente a  $\tanh(N)$ . Se diferencia en que se ejecuta más rápido que la implementación MATLAB de  $\tanh$ , pero los resultados pueden tener diferencias numéricas muy pequeñas. Esta función es una buena compensación para las redes neuronales, donde la velocidad es importante y la forma exacta de la función de transferencia no lo es.

(mathworks.com/help, 2019).

La imagen siguiente presenta el esquema de la red neuronal, se debe seleccionar Train, escoger las entradas y las salidas para posteriormente enviar la red a entrenamiento.

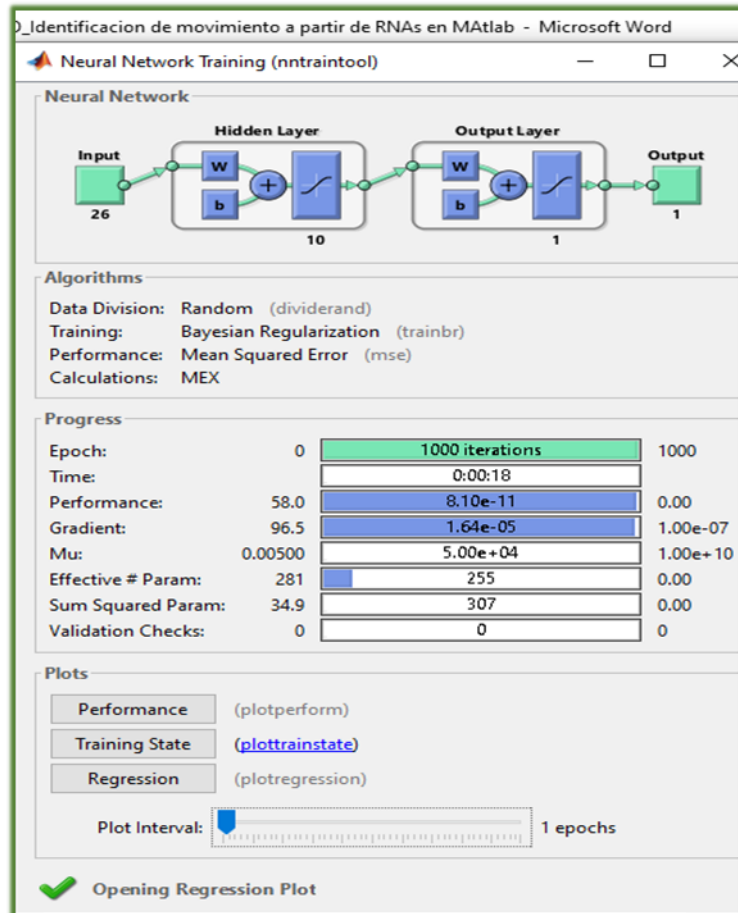


**Figura 71. Red neuronal personalizada con 26 entradas y 10 neuronas en la capa oculta.**

**Software MATLAB, nntool**

Una vez se inicie este, se despliega una nueva ventana donde se indican, el diagrama de la red las propiedades de los algoritmos, barras de carga en los procesos, como épocas, número de iteraciones que se realizan, tiempo, rendimiento, gradiente número de parámetros efectivos entre otros.

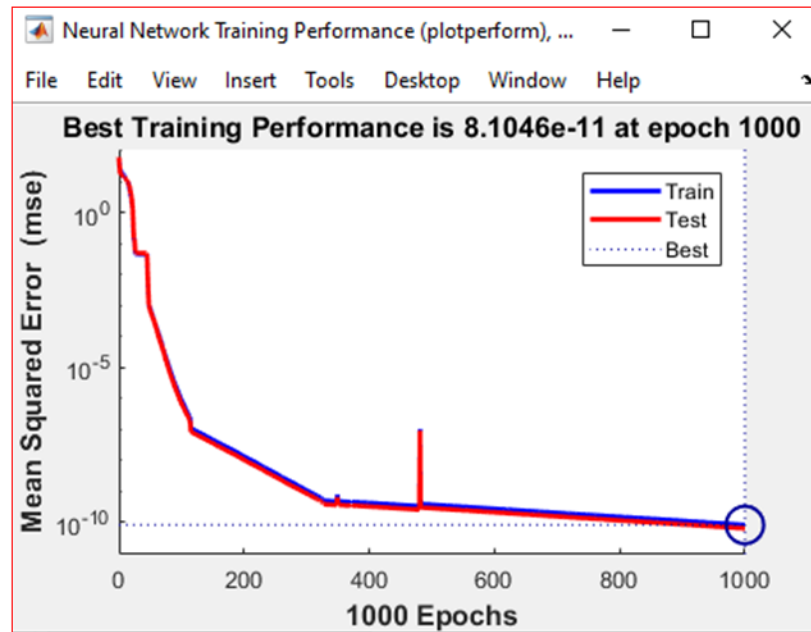




**Figura 72. Entrenamiento de la Red neuronal artificial con las características definidas anteriormente. Software MATLAB, nntool**

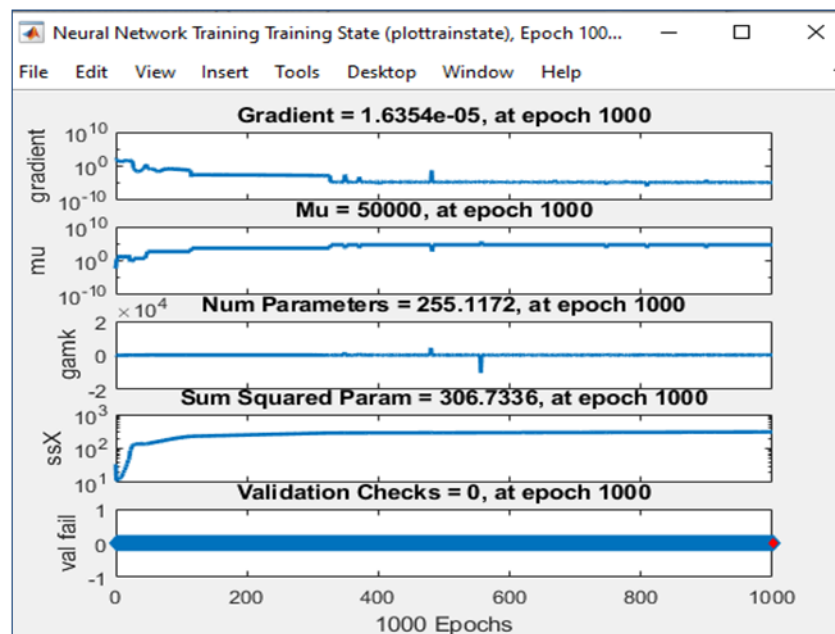
Después de realizado el entrenamiento permite la construcción automática de las gráficas performance (Rendimiento), Training State (Estado de entrenamiento) y Regresión.

La grafica siguiente representa el punto donde se encuentra el mejor rendimiento de entrenamiento y la época a la cual ocurre en este caso la 1000.



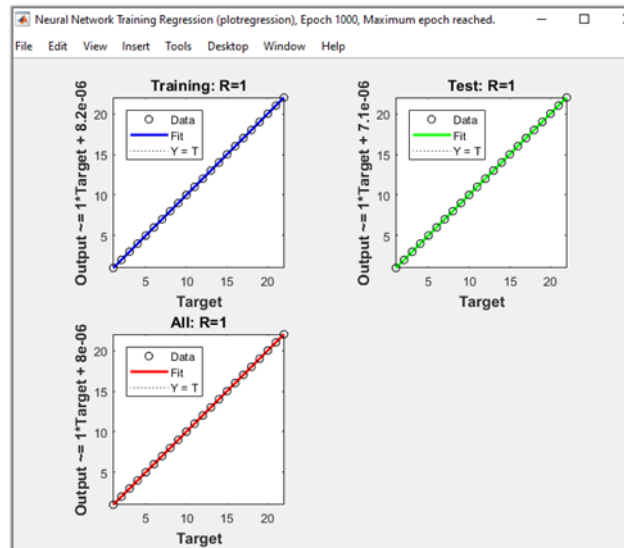
**Figura 73. Grafica error cuadrático medio vs número épocas. Software MATLAB, nntool**

Comportamiento de los diferentes parámetros de progreso durante el entrenamiento.



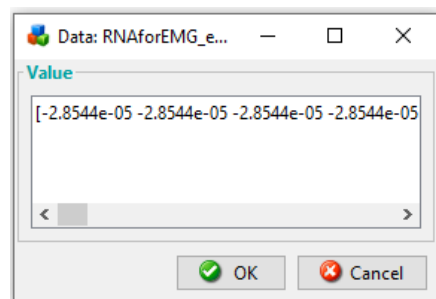
**Figura 74. Grafica parámetros de progreso Vs número de épocas Software MATLAB, nntool**

Muestra la regresión lineal entre el objetivo de la red neuronal y el objetivo+ el error



**Figura 75. Grafica de  $1 \times$  vector objetivo + error vs. Vector objetivo. Software MATLAB, nntool**

Una vez revisadas las gráficas se procedió a verificar que los datos de error fuesen pequeños, y que los datos de salida fueran similares a los datos de objetivo. La siguiente imagen demuestra que los errores que se presentan son bastante pequeños ya que poseen exponente de  $-05$ .



**Figura 76. Datos del error según la respuesta obtenida de la validación. Software MATLAB, nntool**

Una vez se concluye todo el proceso de entrenamiento y verificación, se procede a Exportar la variable **RNAforEMG** de clase Network (Red neuronal artificial) y almacenarla.

#### **4.9 Integración del código, y Revisión de la Respuesta de los Algoritmos**

Una vez se concluye con la elaboración de los dos algoritmos para clasificar las señales Electromiográficas acondicionadas, se llevó a cabo un encadenamiento de los scripts necesarios, con el fin de realizar todo el procedimiento de lectura, almacenamiento de la señal, extracción de características y posterior identificación del movimiento de una manera automática, y así validar las respuestas que entregan la red neuronal artificial y el clasificador lineal SVM.

Para todo ello el script de MATLAB denominado CDF.m, cuenta inicialmente con la configuración puerto COM y nombre del Arduino seleccionado ('MICRO'), seguido de la asignación de nombres de las variables que se presentan y con las cuales se trabajan durante todo el proceso, después de ello aparece un bucle While encargado de leer las señales, graficarlas, aplicar las funciones y extraer todas las características perteneciente a las mismas, para al final ejecutar la RNA o Linear SVM y entregar de manera numérica la respuesta de identificación según el movimiento realizado por el sujeto de prueba. El bucle central del código es el siguiente:

```

while(c < 1000)
    Vts1(i) = readVoltage(a, 'A0');%lectura para el vector
    if (Vts1(i)>=0.025)
        Vs(c) = readVoltage(a, 'A0');%lectura para la grafica
        Vsf = filter (filtrado, Vts1);%vector filtrado
        i = i + 1;
        if (i == 10)
i = 1;
    if (sum(Vts1)>3) && (trapz(Vts1)>1) && (max(Vts1)>1) &&
        (range(Vts1)>0.7)%
j=j+1;%Condicionales para extracción de
        características.

    figure(2)%
    ylim([0 5.1]);
    xlim([1 9]);
    plot(Vts1,'-rs',...
        'LineWidth',2,...
        'MarkerSize',10,...
        'MarkerEdgeColor','b',...
        'MarkerFaceColor',[0.5,0.5,0.5])

    Vsft = Vts1';%Traspuesta del vector de lectura
    %aplicación de funciones.
    MediaX = media(Vts1); %media al vector
    ValorTrapzX = area(Vts1);% Calculo de integral
    ValorMedioXrms = valormediorms(Vsf);% valor raíz del
        valor medio
    AutcX = autocorrFeatures(Vsft, 8.333333333);%/////
    SpkX = spectralPeaksFeatures(Vsft, 8.333333333);% //
Mathworks.
    SpwfX = spectralPowerFeatures (Vsft, 8.333333333);%
    VectMovi2 = Vts1;

    % vector de características(26 filas, 1 columna
    Vf=[MediaX, ValorMedioXrms, ValorTrapzX, AutcX, SpkX,
        SpwfX, VectMovi2]';
    %Creacion de Tabla de características.
    Tab = table(MediaX, ValorMedioXrms, ValorTrapzX,
        AutcX, SpkX, SpwfX, VectMovi2);

    %uso del clasificador Neuronal Network encapsulado.
if length(Vf)== 26
    RNA2=RNA_Final(Vf);
    RNA=VIERNES(Vf);
    TRNA=vertcat(TRNA, RNA);

```

Funciones de

```

    TRNA2=vertcat(TRNA2, RNA2);
end

if (1.0000<RNA2)&&(RNA2<1.9999)
    disp ("Movimiento Dedo Anular");
end

if (3.0000<RNA2)&&(RNA2<3.9999)
    disp ("Movimiento Dedo Corazón");
end

%reinicio de variables
Vts1 = 0;
Vsf = 0;
MediaX = 0;
ValorTrapzX = 0;
ValorMedioXrms = 0;
AutcX = 0;
SpkX = 0;
SpwfX = 0;
VectMovi2 = 0;
%RNA=0;
end
end

figure(1)
ylim([0 5.1]);
xlim([c-20 c+5]);
plot(Vs,'k','MarkerSize',5);
drawnow
c = c+1;

end
end

```

**Figura 77. Código 7**

Como se indicó anteriormente este código se encarga de llamar y ejecutar todas las funciones, así como procedimientos que fueron descritos en el ítem 4.8 (Definición de características presentes en las señales.) Debido a la extensión del código en conjunto durante todo el desarrollo

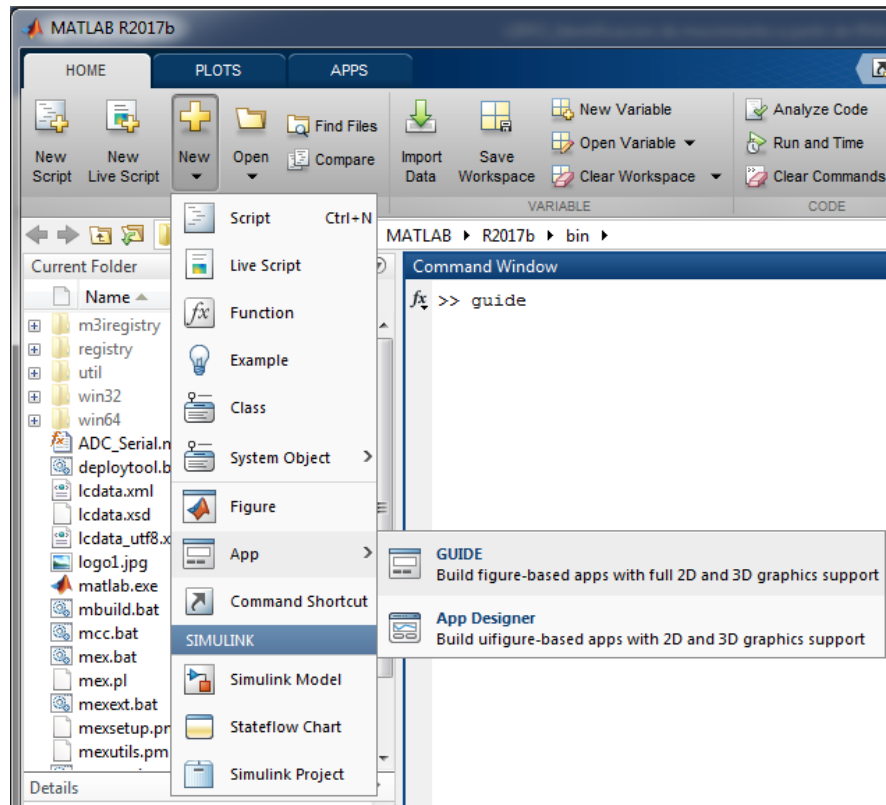
experimental se presenta por etapas, pero se anexa una copia digital en el CD.

Todos los procedimientos anteriormente mencionados contaron con una cantidad considerable de ensayos, realizando variaciones, desde cambio en el sujeto de prueba, los movimientos que se medían, funciones y características, pero siempre siguiendo la misma metodología consignada y en pro de cumplir los objetivos del proyecto.

#### **4.10 Interfaz Gráfica (GUI)**

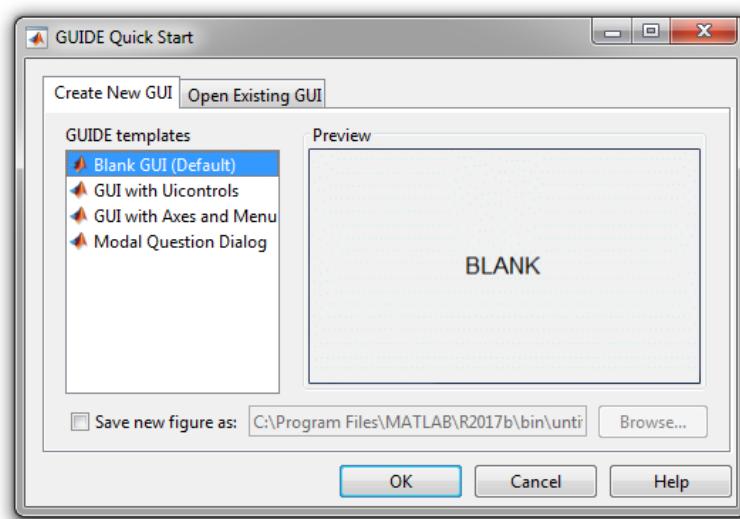
Una vez definido el código de extracción de características Matlab nos ofrece la opción para la elaboración de la interfaz gráfica denominado GUI. Una GUI debe ser consistente y fácilmente entendida por el usuario además proporcionar al usuario con la habilidad de usar un programa sin tener que preocuparse sobre los comandos para ejecutar un programa.

MATLAB dispone de una utilidad para el desarrollo interactivo de interfaces de usuario (GUI) llamado GUIDE. Para ello se escribe el comando `guide` en el command window y presionar la tecla enter, o se pulsa sobre su icono en el entorno de desarrollo Matlab (Barragán, 2017).



**Figura 78. Inicio guide**

Se presentan cuatro opciones en la ventana de inicio de la GUI:



**Figura 79. Inicio de la GUI**



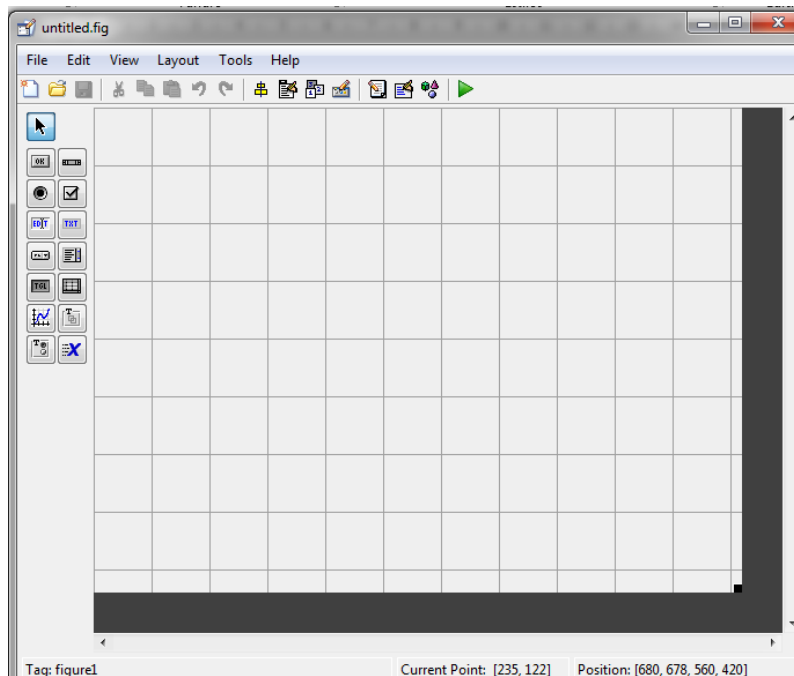
**Blank GUI (Default):** La opción de interfaz gráfica de usuario en blanco (viene predeterminada), nos presenta un formulario nuevo, en el cual podemos diseñar nuestro programa.

**GUI with Uicontrols:** Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades. Podemos ejecutar este ejemplo y obtener resultados.

**GUI with Axes and Menu:** Esta opción es otro ejemplo el cual contiene el menú File con las opciones Open, Print y Close. En el formulario tiene un Popup menu, un push button y un objeto Axes, podemos ejecutar el programa eligiendo alguna de las seis opciones que se encuentran en el menú despegable y haciendo click en el botón de comando.

**Modal Question Dialog:** Con esta opción se muestra en la pantalla un cuadro de diálogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No, dependiendo del botón que se presione, el GUI retorna el texto seleccionado (la cadena de caracteres 'Yes' o 'No').

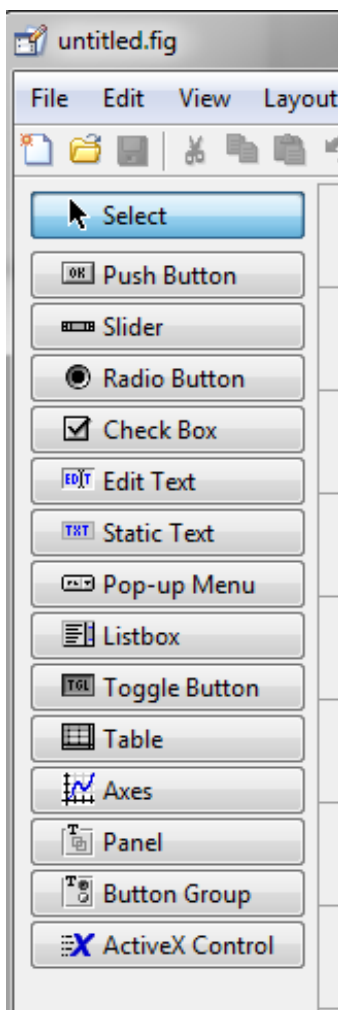
Se elige la primera opción (Blank GUI) y se tendrá la ventana de trabajo.



**Figura 80. Ventana de trabajo**

El editor permite construir interfaces arrastrando y soltando componentes en el área de diseño de la GUI. Todas las GUIs creadas con guide empiezan con una función inicial (callback) que se invoca cuando se invoca la interfaz, la operación automática de guardado (save) genera un fichero .m y un fichero .fig. Una GUI en Matlab consta de dos archivos con extensiones .m y .fig. El archivo .m contiene el código o funciones correspondientes a los componentes de la interfaz. El archivo .fig contiene los elementos gráficos. Cada vez que se adicione un nuevo componente en la interfaz gráfica, se genera de forma automática el código en el archivo .m. Para ejecutar una interfaz gráfica, si se la etiquetó, por ejemplo, con el nombre curso.fig, simplemente ejecutamos en la ventana de comandos» curso. O haciendo clic derecho en el archivo .m y seleccionando la opción run. (2)

Para obtener la etiqueta en cada elemento de la paleta de componentes se ejecuta:  
File»Preferentes y seleccionamos Show names in component palette.

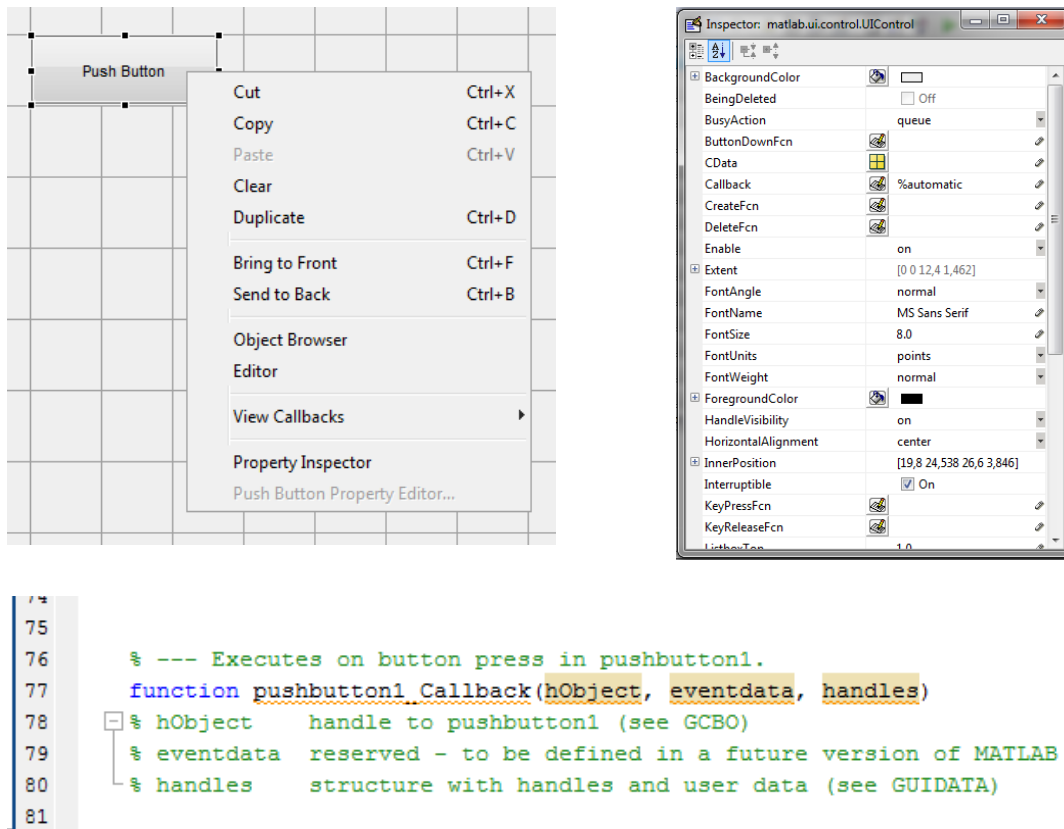


**Figura 81. Paleta de componentes**

Cada uno de los elementos de la paleta de componentes tiene un conjunto de opciones o propiedades que se acceden con el clic derecho, La opción Property Inspector permite personalizar cada componente, como por ejemplo cambiar el tipo y tamaño de letra, color, etiqueta, posición y reacción del componente a clics del usuario. En esta ventana podemos modificar el nombre (campo tag) con el cual este componente aparecerá en el archivo .m asociado. Al hacer clic derecho en el componente ubicado en el área de diseño, una de las opciones más importantes es View Callbacks, la cual, al ejecutarla, abre el archivo .m asociado a

nuestro diseño y nos posiciona en la parte del programa que corresponde a la función que se ejecutará cuando se realice una determinada acción sobre el componente que estamos editando.

(2)

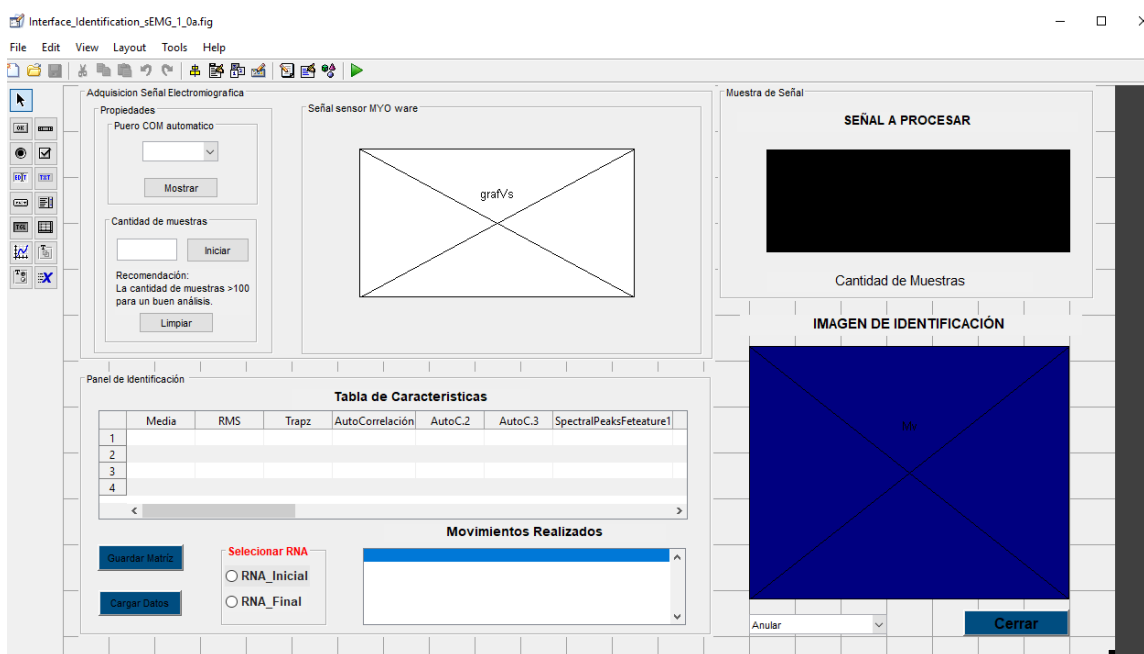


**Figura 82. Conjunto de opciones**

Todos los valores de las propiedades del componente (color, valor, posición, etiqueta, etc.) y los valores de las variables transitorias del programa se almacenan en una estructura, los cuales son accedidos mediante un único y mismo identificador para todos éstos. Tomando el programa listado anteriormente, el identificador se asigna en `handles.output = hObject`. La variable `handles` es nuestro identificador a los datos de la aplicación. Esta definición de identificador es actualizada con la siguiente instrucción: `guidata(hObject, handles);`. La variable `guidata` guarda

los datos de la función asociada al componente. Como regla general, en cada función se debe escribir en la última línea la siguiente sentencia: `guidata(hObject, handles);`. Esta sentencia nos garantiza que cualquier cambio o asignación de propiedades o variables quede almacenado. La asignación u obtención de valores de los componentes se realiza mediante las funciones `get` y `set`.

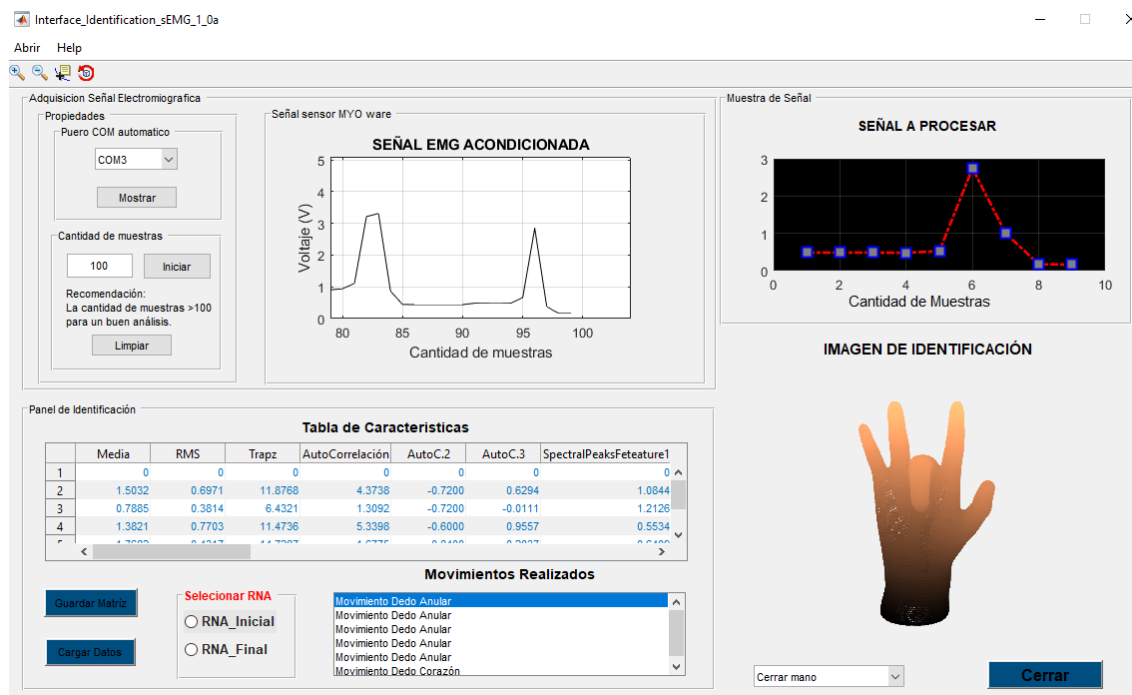
Con todas estas alternativas para creación del gui se estableció por parte de los autores de este proyecto el entorno de la interfaz de la siguiente manera:



**Figura 83. Entorno de la interfaz**

Donde cada elemento permite la interacción de forma rápida y fácil, solo se requiere la conexión del dispositivo (microcontrolador conectado con el sensor correctamente ubicado en la posición descrita anteriormente en este proyecto) en un puerto COM del computador y como dato adicional solicita la cantidad de muestra para dar inicio a la interacción con la interfaz; durante dicha interacción se muestra la señal procesada en tiempo real y una toma de señal característica

que es producida al realizarse el movimiento de la mano y cada uno de sus dedos, de a movimiento por vez; esta señal es identificada y nombrada por parte de la red neuronal toda vez que se cumplen con las condiciones para cada respuesta, dando como resultado un dato escrito y almacenando los datos para ser vistos en cualquier momento, adicionalmente mostrando dicha interpretación con imágenes en tercera dimensión para mejor comprensión de los movimientos identificados.

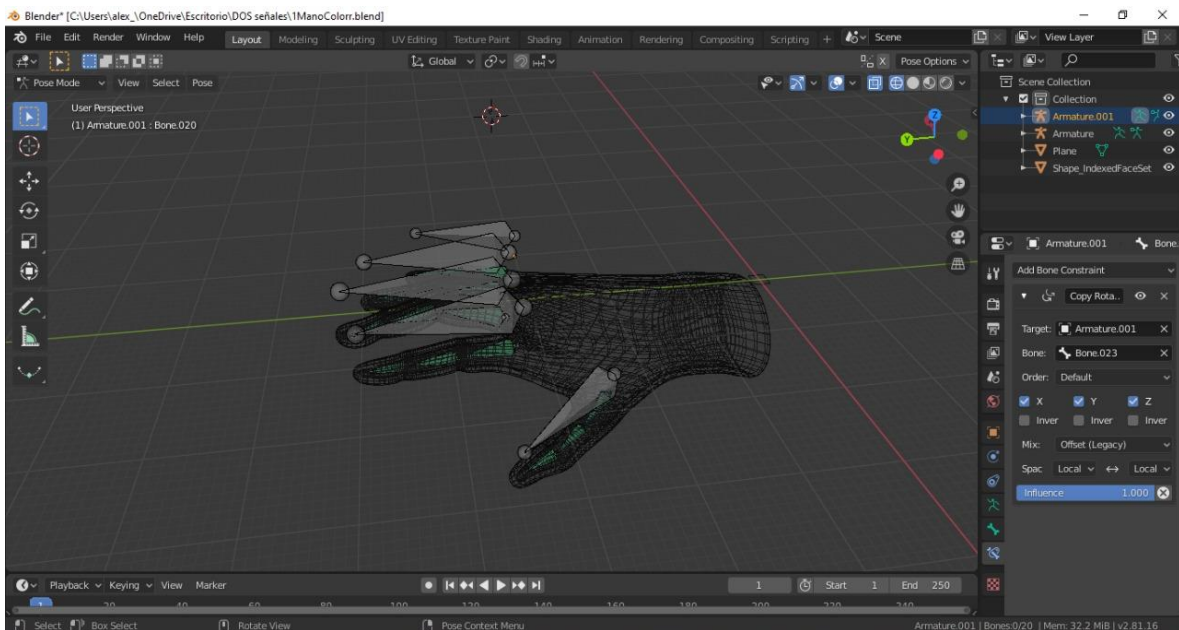


**Figura 84. Interfaz en funcionamiento**

Cabe aclarar que la programación antes descrita durante el proyecto se agregó a la funcionalidad de la interfaz y se presentara como anexo al proyecto.

Para la elaboración de las imágenes en tercera dimensión se usó un software de uso libre llamado Blender el cual se adquirió de forma directa desde su sitio web [www.blender.org](http://www.blender.org).

Blender es la suite de creación 3D gratuita y de código abierto. (Blender se presenta a través de un sistema de código fuente accesible al público bajo la Licencia Pública General de GNU.). Admite la totalidad de la canalización 3D: modelado, aparejo, animación, simulación, renderizado, composición y seguimiento de movimiento, edición de video y canalización de animación 2D. Permite la creación de imágenes 3D como archivos \*.ply que pueden ser leídos y mostrados por el software Matlab.



**Figura 85. Software Blender**

El modelo de mano mostrado en este proyecto se adquirió del sitio web [www.free3d.com](http://www.free3d.com) el cual nos proporcionó una amplia variedad de objetos 3D de forma gratuita y con licencia libre de derechos para trabajos con estos. También se pueden encontrar de forma paga, los cuales presentan mayor complejidad en su elaboración, pero de igual manera permiten el uso libre después de su adquisición.

Adicionalmente Matlab proporciona los comandos y herramientas necesarias para la generación de código ejecutable con el Matlab Compiler (sin requerir el entorno Matlab). En los ejecutables o librerías creadas se puede incluir el MATLAB Compiler Runtime (MCR), que permite la ejecución de estos en ordenadores que no tienen una versión instalada de Matlab, una aplicación o librería generada por Matlab tiene dos partes: un fichero binario dependiente de la plataforma y un fichero con el código y datos Matlab encriptado.



## 5. Conclusiones

El dispositivo Myo (sensor) es bastante aceptable para el desarrollo de aplicativos con electromiografía, sin embargo, el uso de un solo sensor es insuficiente para la toma de las señales producidas por el antebrazo; esto debido a la cantidad de movimientos posibles y la variedad de músculos que intervienen en dicha generación, por ende se produce que las señales sean muy variables y de difícil caracterización, puesto que un mismo movimiento genera señales con patrones diferente cada vez que es tomado.

Las afectaciones de la señal por agentes externos dificultan en gran medida la buena caracterización y toma de datos, dado que elementos como relojes, manillas, pulseras incluso la presencia de abundante bello afectan la señal electromiografica directamente. Se notó como el ruido de la red eléctrica (pc conectado para carga) afectaba la percepción del sensor y por ende la variación en los datos para la caracterización.

Se elaboró una interfaz gráfica de fácil manejo con un entorno virtual muy agradable que permitirá un buen entrenamiento a personas con dificultades de movilidad en el antebrazo y la mano, sin embargo, se hace necesario la implementación de mayores dispositivos para una mejor toma de datos característicos y que esto contribuya positivamente en el avance de la funcionalidad de la interfaz.

Se realizó gran cantidad de toma de datos por parte de los autores de este proyecto para la búsqueda mejores resultados pero la falta de recursos no permitió avanzar como se esperaba.

## 6. Recomendaciones

Realizar toma de datos electromiográficos no solo en personas sanas, sino tener en cuenta personas con discapacidad de movimiento en antebrazo y mano ya que este proyecto va dirigido a brindar una ayuda a esta población en particular.

Implementar más dispositivos (sensores) y de mejor calidad en la toma de señales electromiográficas para lograr una correcta interpretación de características.

Evaluar diferentes métodos matemáticos a los descritos en este proyecto para la caracterización y así lograr mejorar el rendimiento de la red neuronal y a su vez de la interfaz.

La búsqueda de información con entidades altamente calificadas en el tema y profesionales del saber facilitara un trabajo mancomunado para obtener mejores resultados.

Cuando se desee tomar datos de señales electromiográficas se recomienda hacerlo en un ambiente libre de artefactos electrónicos y además el aislamiento del individuo de muestra los mejor posible.

## Referencias Bibliográficas

- Advancer Technologies. (2015). 3-lead Muscle / Electromyography Sensor for Microcontroller Applications. Recuperado de:  
<https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf>.
- Alva, A. (2012). “Procesamiento de señales de electromiografía superficial para la detección de movimiento de dos dedos de la mano”. Universidad Ricardo Palma. Lima – Perú.
- Amangandí, J. (2018). “*Diseño e implementación de manos robóticas para la generación de movimientos del alfabeto dactilológico*”. Riobamba – Ecuador.
- Amicivirtual. (2018). Muñeca y mano, músculos del antebrazo y de la mano. Recuperado de:  
<http://www.amicivirtual.com.ar/Anatomia/11MucaMano.pdf>.
- Aprendiendo Arduino. (2020). Pagina. Recuperado de:  
<https://aprendiendoarduino.wordpress.com>
- ARDUINO. (2018). Descarga de software de programación. Recuperado de:  
<https://www.arduino.cc/en/Main/Software>
- ARDUINO. (2018). Homenaje. Recuperado de: <http://www.arduino.cc>
- Armas, A; López, A; Uriarte, I; Díaz, M. & Barboza, N. (2015). “*Control de modelo de prótesis de mano por señal mioeléctrica*”. Escuela de Ciencias de la Ingeniería y Tecnología, Universidad Autónoma de Baja California, Tijuana, México.  
[dx.doi.org/10.24254/CNIB.15.55](https://doi.org/10.24254/CNIB.15.55).

Ayoroa, R. & Kang, D. (2016). “*Propuesta de implementación de prótesis mioeléctrica de bajo costo para miembro superior bajo codo con entorno virtual de entrenamiento*”. Universidad Nacional del Este,

Barrera, A; Merchán, E; Rodríguez, R; Hernandez, D. & Hernandez, G. (2017). “*Obtención del rango de movilidad de los dedos índice, medio, anular y meñique*”. Cuernavaca, Morelos, México.

Brazeiro, J; Petraccia, S. & Valdés, M. (2015). “Mano controlada por señales musculares”. Universidad de la República. Uruguay.

Cabrera, E. & Montes, E. (2012) *Obtención y análisis de señales electromiográficas de las articulaciones tibiofemoral y femororotuliana aplicada a la detección y rehabilitación de problemas musculares en el movimiento de la rodilla*. Universidad politécnica salesiana, sede cuenca, Ecuador.

España, J. (2016). “*Diseño y construcción de un prototipo electromecánico de prótesis transhumeral comandado por movimientos del brazo funcional*”. Facultad de Ingeniería Mecánica, Corporación Universitaria Autónoma de Nariño, Pasto - Colombia.

Gila, J; Malanda, A; Rodríguez, I; Rodríguez, J. & Navallas, J. (2009). “*Métodos de procesamiento y análisis de señales electromiográficas*”. *An. Sist. Sanit. Navar*, 32(3).

Goconqr. (2018). Tipos de interfaz gráfica. Recuperado de:

[https://www.goconqr.com/p/3597306-tipos-de-interfaz-grafica-mind\\_maps](https://www.goconqr.com/p/3597306-tipos-de-interfaz-grafica-mind_maps).

IESCAMP. (2016). Placa Arduino Recuperado de:

<http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>. Recuperado septiembre

2018.

JUANCARLOSUSOMATLAB. (2015). Definición Matlab. Recuperado de:

<https://juancarlosusomatlab2015.weebly.com/definicionmatlab.html>

Kapandji, A. (2012). *Fisiología articular*. España: Ed. Médica Panamericana, Tomo I, 6<sup>a</sup>. ed.

pág. 146-151, recuperado octubre 2018 <https://www.mirandafisioterapia.com/single-post/2018/01/11/Muneca-Movimientos-y-amplitudes>

Lara, F. (2018). “*Fundamentos de redes neuronales artificiales*” laboratorio de cibernética aplicada. Universidad nacional autónoma de México. Recuperado de:

[http://conceptos.sociales.unam.mx/conceptos\\_final/598trabajo .pdf](http://conceptos.sociales.unam.mx/conceptos_final/598trabajo.pdf). Recuperado octubre 2018.

Lazarte, E. (2017). “Entrenamiento de una red neuronal Hardware desde Matlab (Hardware in the loop)”. Pg., 12-14 universidades Politécnica de Madrid, España.

MathWorks. (2018). Redes neuronales convolucionales tres cosas que es necesario saber en. Recuperado de: <https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>.

Mendoza, L; Peña, J; Muños, L. & Belandia, H. (2013). “*Procesamiento de señales provenientes del habla sub vocal usando Wavelet Packet y Redes Neuronales*”. Universidad de Pamplona, España.

Milían, L; Vásquez, D; Choquehuanca, E. & Achanccaray, D. (2015). *Adquisición, parametrización y clasificación de patrones de señales mioeléctricas para el control de movimiento de un exoesqueleto que asiste la bipedestación*. 12° congreso iberoamericano

de ingeniería mecánica Guayaquil-Ecuador, recuperado octubre 2018.

Montaño, J. (2002). *Redes neuronales artificiales aplicadas al análisis de Datos*. pg., 17,

Universitat de les illes Balears, Palma de Mallorca, España disponible en

[http://dspace.uib.es/xmlui/bitstream/handle/11201/2511/Montano\\_Moreno\\_](http://dspace.uib.es/xmlui/bitstream/handle/11201/2511/Montano_Moreno_)

[JuanJose.pdf?sequence=1&isAllowed=y](http://dspace.uib.es/xmlui/bitstream/handle/11201/2511/Montano_Moreno_JuanJose.pdf?sequence=1&isAllowed=y) Recuperado octubre 2018.

Puiceibalxo. (2009). Interfaz Perceptual de Usuario. Recuperado de:

<http://puiceibalxo.blogspot.com/2009/04/perceptual-user-interface-pui.html>

Quinayás, C. & Gaviria, C. (2015). “*sistema de identificación de intención de movimiento para el control mioeléctrico de una prótesis de mano robótica*”, *Ing. Univ.*, vol.19, no.1, pp.27-50, Ene. Jun, 2015. <http://dx.doi.org/10.11144/javeriana.iyu19-1.sim>.

Reyes, D., Arias, M., Duarte, J. & Loaiza H. (2015) “*implementación de FPGA de un clasificador de movimientos de la mano utilizando señales EMG*”. *Redes de ingeniería*, 6(1), 85-94.

Sangoquiza, E. & Román, C. (2017). “*Diseño y construcción de un prototipo de prótesis mioeléctrica para desarticulación de muñeca*”. Universidad de las Fuerzas Armadas.

Staugaard, J. (2014). *Anatomía del ejercicio y el movimiento*. México: Paidotribo.