

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **SISTEMA DE EMULACIÓN DE MOVIMIENTO DE LA EXTREMIDAD SUPERIOR PARA DETERMINAR POSICIÓN**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL**

**ANTONIO ARIEL TOBAR BERMEO**

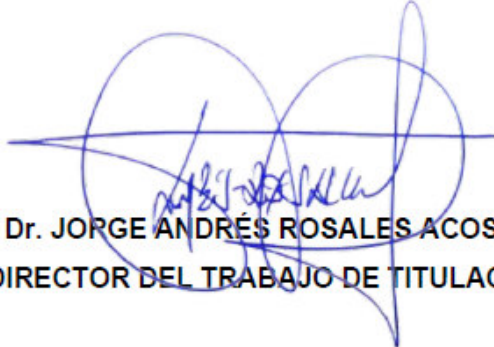
**BRYAN XAVIER LÓPEZ CAMACHO**

**DIRECTOR: Dr. JORGE ANDRÉS ROSALES ACOSTA**

**Quito, 10 de febrero 2022**

## **AVAL**

Certifico que el presente trabajo fue desarrollado por Antonio Ariel Tobar Bermeo y Bryan Xavier López Camacho, bajo mi supervisión.



**Dr. JORGE ANDRÉS ROSALES ACOSTA**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## DECLARACIÓN DE AUTORÍA

Nosotros, Antonio Ariel Tobar Bermeo y Bryan Xavier López Camacho, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejamos constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



Antonio Ariel Tobar Bermeo



Bryan Xavier López Camacho

## DEDICATORIA

A Dios, mis abuelitos Carlos, Ángel y Blanca, quienes son la fuerza y empuje que me ha inspirado a salir adelante cumpliendo el sueño de estudiar una ingeniería.

A mis padres Beatriz y Xavier, quienes en todo momento y bajo cualquier circunstancia me han brindado su apoyo incondicional, confianza y amor.

A mi hermano Kevin, quien cursa la mitad de su vida universitaria y siempre a sido un ejemplo de fuerza y dedicación para mí.

A mis hermanas de otra madre Verónica y Abigail quienes han estado a mi lado en los mejores y peores momentos y siempre han estado apoyándome.

A Elvira y Elena Vallejo, quienes siempre han estado pendiente de mi progreso académico y me han ayudado para cumplir mi objetivo, incondicionalmente.

A mi abuelita, familia, primos y amigos quienes siempre me han apoyado en todo momento.

Bryan Xavier López Camacho



## **DEDICATORIA**

A mi abuelita Blanca, a mi madre Grace y a mi hermano Mateo, que han estado ahí conmigo en cada paso que he dado en mi carrera y me han apoyado, porque todo lo que soy se lo debo a ellos.

A mis amigos más cercanos, a los cuales considero como mi familia, Diego, Gustavo y Milagros, que con sin sus consejos, experiencias, tiempos prestados y risas jamás podría haber llegado tan lejos.

A mis amigos que tuve la oportunidad de hacer en la universidad, que con su ayuda y amistad brindada me daban fuerzas para seguir adelante en cada escalón que daba en mi formación académica.

Antonio Ariel Tobar Bermeo

# AGRADECIMIENTO

Agradezco a mis padres Xavier López y Beatriz Camacho, por confiar siempre en mí, por estar siempre a mi lado a pesar de las más grandes dificultades y por apoyarme durante toda mi carrera universitaria, dándome siempre ese pequeño empujón tan necesario e importante para salir adelante y cumplir todos mis objetivos.

Gracias a mi querida Escuela Politécnica Nacional, la cual me permitió cumplir uno de mis proyectos de vida, el cual es ser un profesional en el área de la ingeniería. También agradezco a la Facultad de Ingeniería Eléctrica y Electrónica por brindarme educación de calidad de la mano de los mejores docentes.

A mi director PhD. Andrés Rosales, por su apoyo incondicional durante la elaboración de este proyecto de titulación, a quien agradezco por su interés y gran disposición de ayudar en este trabajo.

A la MSc. María Fernanda Trujillo, quien a sido mi docente desde el inicio de la carrera y tuve la fortuna de tener su guía y apoyo en este proyecto, agradezco por su interés en nuestro tema para el trabajo de titulación, por sus consejos que siempre fueron útiles y nos han permitido concluir el proyecto, también le agradezco por su paciencia y ayuda con los problemas que se presentaron durante la elaboración del trabajo de titulación.

Agradezco mi compañero del proyecto Antonio Tobar, por su apoyo incondicional durante estos años de carrera, quien siempre me apoyó para cumplir todos mis objetivos académicos, por su amistad y por ser un pilar fundamental para terminar el presente proyecto.

A mi hermano Kevin López quien siempre a estado a mi lado guiándome y siendo un gran amigo en quien puedo confiar, motivándome a ser mejor cada día.

Agradezco a mi familia que siempre a estado pendiente de mi progreso en la carrera y han confiado en mí en todo momento.

Finalmente agradezco a todos mis amigos que conocí durante toda la carrera universitaria en especial a Bryan, Gabriel, Francisco, David, Paul y Francesca quienes me motivaron a ser cada día mejor y me han apoyado en todo momento.

Bryan Xavier López Camacho

## **AGRADECIMIENTO**

Agradezco a mi abuelita Blanca Egas Fuller y a mi madre Grace Bermeo, por apoyarme, no solo en mi vida universitaria, sino en todo mi desarrollo; por todo el esfuerzo y dedicación que tuvieron conmigo y por fortalecerme haciendo la persona y profesional que soy ahora, por lo que siempre estaré agradecido toda mi vida.

Doy gracias a mi hermano Mateo y a mis amigos más cercanos Diego, Gustavo, Milagros y Mishell, por siempre estar ahí y ayudarme en cualquier inconveniente que haya tenido, por brindarme su compañía y tiempo en los momentos que más débil me sentí.

Agradezco a la Escuela Politécnica Nacional por brindarme la oportunidad de demostrar mi potencial en mi carrera y por brindarme la formación requerida para ser un gran profesional y mas que nada de darme la oportunidad de poder crear un cambio con mi conocimiento.

Agradezco a mi tutor Ph.D. Andrés Rosales por su disposición, consejos, conocimiento, tiempo y ayuda incondicional en la realización del trabajo.

Agradezco a la MSc. María Fernanda Trujillo que ha sido una magnifica docente, con quien he tenido la oportunidad de ser su estudiante desde el principio de mi carrera y se nos ha apoyado en el presente proyecto con su conocimiento, tiempo, paciencia y consejos con los cuales nos ha permitido finalizar este trabajo de la mejor manera.

Agradezco a mi compañero de tesis Bryan López por brindarme su tiempo, paciencia y conocimiento para lograr obtener los mejores resultados en el presente trabajo y cumplir con los objetivos de este.

Finalmente, agradezco a todos mis amigos que he hecho a lo largo de mi vida universitaria, especialmente a Paúl y Daya que me han ayudado con sus consejos, tiempo y risas las cuáles me han permitido salir adelante.

Antonio Ariel Tobar Bermeo

# ÍNDICE DE CONTENIDO

AVAL .....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
DEDICATORIA.....	IV
AGRADECIMIENTO.....	V
AGRADECIMIENTO.....	VI
ÍNDICE DE CONTENIDO.....	VII
RESUMEN .....	IX
ABSTRACT .....	X
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS .....	2
1.2 ALCANCE .....	3
1.3 MARCO TEÓRICO.....	4
1.3.1 FISIOLÓGÍA DE LA EXTREMIDAD SUPERIOR.....	4
1.3.2 SENSORES INERCIALES .....	10
1.3.3 ADQUISICIÓN DE SEÑALES.....	14
1.3.4 TÉCNICAS DE PREPROCESAMIENTO.....	18
1.3.5 APRENDIZAJE AUTOMÁTICO .....	21
2. METODOLOGÍA.....	27
2.1 ADQUISICIÓN DE DATOS .....	27
2.1.1 SENSORES DE MOVIMIENTO.....	27
2.1.2 LOCALIZACIÓN DE SENSORES.....	28
2.1.3 SELECCIÓN DEL MICROCONTROLADOR.....	30
2.1.4 FORMATO DE ADQUISICIÓN DE DATOS.....	32
2.1.5 HARDWARE IMPLEMENTADO .....	37
2.2 PREPROCESAMIENTO DE LA SEÑAL .....	39
2.2.1 TÉCNICAS DE PREPROCESAMIENTO DE DATOS.....	40
2.2.2 EXTRACCIÓN DE CARACTERÍSTICAS.....	43
2.3 PROCESAMIENTO DE LA SEÑAL.....	46
2.3.1 TÉCNICAS DE CLASIFICACIÓN .....	47
2.4 DISEÑO DE LA INTERFAZ GRÁFICA.....	50

3. RESULTADOS Y DISCUSIÓN .....	60
3.1. RESULTADO DE LOS CLASIFICADORES .....	60
3.1.1. RESULTADOS MÁQUINA DE SOPORTE VECTORIAL - LINEAL...	62
3.1.2. RESULTADOS K-VECINOS MAS CERCANOS- COSENO .....	69
3.1.3. RESULTADOS REDES NEURONALES - AMPLIA.....	75
3.1.4. COMPARACIÓN DE CLASIFICADORES .....	83
4. CONCLUSIONES Y RECOMENDACIONES.....	85
4.1. CONCLUSIONES.....	85
4.2. RECOMENDACIONES .....	86
5. REFERENCIAS BIBLIOGRÁFICAS .....	88
A4.1 INSTALACIÓN DE LA APLICACIÓN.....	102
A4.2 INICIO DE LA TOMA DE DATOS.....	104

## RESUMEN

El reconocimiento de actividades humanas es un ámbito que está ganando gran relevancia a lo largo de los últimos años, con el objetivo de facilitar el reconocimiento de la fisiología humana y lo que conlleva el comportamiento motriz humano para poder realizar aplicaciones como es la telemedicina [2]. Teniendo en consideración este antecedente, el presente proyecto plantea una herramienta de reconocimiento de movimientos específicos de la extremidad superior como extensión, flexión, pronación y supinación, específicamente de un maniquí, el cual tiene como objetivo emular los movimientos de una persona. Esta detección de movimientos se logra con el uso de sensores de medición inercial (IMU, por sus siglas en inglés), los cuales se encuentran ubicados a lo largo de la extremidad superior, ubicados de manera estratégica, para una lectura acertada de las variables que poseen estos sensores.

Para el estudio adecuado de los movimientos detallados anteriormente de la extremidad superior, se va a partir de la creación de una base de datos inicial que está formada a partir de la toma de 40 muestras de 15 minutos cada una que se dividen de la siguiente manera, 20 de estas identifican los movimientos de flexión y extensión, y cambios entre ellas, y las 20 muestras sobrantes para identificar los movimientos de pronación y supinación y cambio entre ellas.

La determinación de movimiento de la extremidad superior se hará por medio del uso del aprendizaje automático, y para determinar un modelo idóneo de aprendizaje automático se realizará una etapa de preprocesamiento y extracción de características de la base de datos detallada con anterioridad para la creación de una tabla, dividida de manera estructurada, que se ingresará como datos de entrada y de salida a los diferentes clasificadores con los que se va a trabajar, con una partición de datos de 50%-50%, donde la mitad de los datos son para entrenamiento y la otra mitad para validación.

Finalmente, realizará una comparativa con los principales clasificadores de aprendizaje automático con mejores resultados, que en este caso son KNN, SVM y redes neuronales; para la determinación de uno de estos modelos para la aplicación final que se expondrá mediante una interfaz realizada en el ambiente de Matlab.

**PALABRAS CLAVE:** extremidad superior, detección de movimientos, sensores de medición inercial, IMU, entrenamiento de datos, aprendizaje automático, redes neuronales, SVM, KNN.

## ABSTRACT

The recognition of human activities is a field that is gaining great relevance in recent years, with the aim of facilitating the recognition of human physiology and what human motor behavior entails to perform applications such as telemedicine [2]. Taking into consideration this precedent, the present project proposes a tool for the recognition of specific movements of the upper extremity such as extension, flexion, pronation, and supination, specifically of a mannequin, which aims to emulate the movements of a person. This movement detection is achieved with the use of inertial measurement sensors (IMU), which are located along the upper limb, strategically located, for an adequate reading of the variables that these sensors have.

For the proper study of the movements detailed above of the upper extremity, we will start from the creation of an initial database that is formed from the taking of 40 samples of 15 minutes each, which are divided in the following way 20 of these identify the movements of flexion and extension and changes between them, and the 20 remaining samples to identify the movements of pronation and supination and change between them.

The determination of the movement of the upper limb will be done through the use of machine learning. To determine a suitable machine learning model, a preprocessing and feature extraction stage will be performed on the previously detailed database for the creation of a table, divided in a structured way, which will be entered as input and output data to the different classifiers to be worked with, with a data partition of 50%-50%, where half of the data is for training and the other half for validation.

Finally, a comparison will be made with the main machine learning classifiers with better results, which in this case are KNN, SVM and neural networks, for the determination of one of these models for the final application that will be exposed through an interface made in the Matlab environment.

**KEYWORDS:** upper extremity, movement detection, inertial measurement sensors, IMU, data training, machine learning, neural networking, SVM, KNN.

# 1. INTRODUCCIÓN

Una de las principales características de la estructura biomecánica del miembro superior, es que representa uno de los sistemas más complejos de articulaciones del cuerpo humano, pero a pesar de ello logra realizar movimientos precisos sin realizar un gran esfuerzo y coordinación. Por lo tanto, es importante realizar el monitoreo de la extremidad superior, el mismo que puede ser utilizando el área de la telemedicina enfocada al manejo de la información de manera virtual y ser enfocada a personas que no tienen acceso a una consulta médica de carácter presencial, para varias atenciones como diagnóstico de enfermedades óseas y musculares, rehabilitación, entre otras [1].

Considerando los problemas mencionados anteriormente, en el presente proyecto técnico se plantea diseñar e implementar un sistema de emulación que permita obtener los movimientos característicos de la extremidad superior, utilizando un conjunto de sensores de unidad de movimiento inercial (IMU), que ubicados de manera estratégica permitan identificar el movimiento realizado por un maniquí, el mismo que permite emular los movimientos característicos como flexión, extensión, supinación y pronación. La emulación del movimiento se lo realizará utilizando un maniquí, considerando que es un modelo de estudio general que se asemeja a las características de movimiento de un ser humano y que posteriormente se permita adaptar a condiciones en las cuales se puedan determinar los movimientos de la extremidad superior de una persona.

El proyecto técnico desarrollado es de carácter investigativo, por lo que, para su implementación en personas deberá cumplir con normativa y sobre todo la supervisión de un profesional.

La extremidad o miembro superior del cuerpo humano es uno de los más importantes para la ejecución de actividades cotidianas de ahí la importancia de enfocar el presente trabajo a su estudio y generación de una herramienta de monitoreo del movimiento. La extremidad está constituida por tres segmentos que son el brazo, antebrazo y mano, estos tres segmentos cuentan con conjunto de articulaciones y segmentos musculares complejos que permiten el movimiento preciso y coordinado sin necesidad de realizar un gran esfuerzo [1], es por ello por lo que se emplea el estudio de la biomecánica que a definición general implica el estudio del movimiento de los seres vivos utilizando la ciencia y tecnología de la mecánica [2].

Para facilitar el estudio de la biomecánica de la extremidad superior, se utilizarán sensores de unidad de medición inercial (IMU), estos sensores cuentan con tres instrumentos internos que permiten determinar medidas de aceleración, velocidad angular y orientación



ya que cuentan con un acelerómetro, giroscopio y magnetómetro. Las señales que entregan estos sensores permiten determinar las características de movimiento del lugar en el que se realiza una medición, por lo tanto, se utilizarán un conjunto de sensores, se los colocarán en los segmentos que conforman el miembro superior y se realizará un análisis de las señales obtenidas para determinar un movimiento.

Para poder realizar el análisis de los movimientos principales de la extremidad superior que se mencionan, es importante determinar un modelo de aprendizaje automático que se acople a los datos que se manejan de cada uno de los sensores. Para ello, se realiza un preprocesamiento de los datos y posteriormente una extracción de características principales tanto en tiempo como en frecuencia, que a partir de estos se pueda entrenar y validar en varios métodos de aprendizaje automático. Para la metodología aplicada, en el presente proyecto se consideró realizar un entrenamiento y validación de 50%-50%, lo cual indica que la mitad de los datos finales se los entrena y con la mitad restante se realiza la validación de estos.

Además, en el presente proyecto se realizará una comparativa entre los principales métodos de aprendizaje automático que están conformado por la máquina de soporte vectorial (SVM), el K vecinos más cercanos (KNN), y redes neuronales para posteriormente aplicar el clasificador con mayor porcentaje de acierto dentro de una interfaz realizada en Matlab donde se evidenciará los cuatro movimientos principales según vaya realizando el maniquí.

## **1.1 OBJETIVOS**

El objetivo general de este Proyecto Técnico es:

- Diseñar e implementar un sistema de emulación de movimiento de la extremidad superior para determinar posición.

Los objetivos específicos del Proyecto Técnico son:

- Investigar información sobre la fisiología de la extremidad superior humana orientado al movimiento y características de posición de este, y la ubicación adecuada de los sensores a lo largo de la extremidad superior, para obtener información del estado del movimiento durante una actividad.
- Analizar técnicas de adquisición, preprocesamiento y procesamiento de las señales biológicas orientadas al movimiento y posicionamiento de la extremidad superior.

- Diseñar e implementar el hardware y software que permita adquirir señales de movimiento y posición a partir de los sensores ubicados en la extremidad superior por medio del uso de técnicas de adquisición, preprocesamiento y procesamiento de dichas señales, así también se diseñará una interfaz gráfica que permita la presentación de datos de forma concisa y clara para el usuario, con el uso de herramientas visuales de Matlab.
- Verificar el funcionamiento del sistema de monitoreo mediante el uso de videos que permitan corroborar el análisis de movimiento realizado por un maniquí.

## **1.2 ALCANCE**

En el proyecto técnico descrito en este documento, se realizará una investigación orientada en la fisiología de la extremidad superior humana donde se pueda determinar características específicas de posicionamiento y movimiento de la extremidad como la flexión, relajación, supinación y pronación; para esto se analizará la relación de las señales obtenidas de al menos tres sensores que poseen similares características con respecto a la detección de movimiento y posición, donde esta relación de señales dependerá de la ubicación de los sensores a lo largo de la extremidad superior para determinar un movimiento específico.

Se analizará y escogerá un microprocesador adecuado, que permita realizar la adquisición de datos de los sensores colocados a lo largo de la extremidad superior y poder enviar dichos datos a la computadora para poder procesar las señales. Se analizarán al menos dos técnicas de preprocesamiento y procesamiento de señales dentro del ambiente de trabajo de Matlab en estado dinámico en el dominio del tiempo y de la frecuencia.

Se diseñará e implementará el hardware que se encontrará conformado por los sensores que permiten adquirir las señales de movimiento y posición, un sistema micro procesado, una batería y un sistema de comunicación inalámbrica para el envío de datos a la computadora.

Se diseñará e implementará el software que realizará la adquisición de datos dentro del microcontrolador, el preprocesamiento y procesamiento de las señales utilizando el algoritmo seleccionado, que permita determinar de manera efectiva el movimiento y posición que se realice. Se implementará una interfaz gráfica que será visualizada por medio de la computadora y está contará con datos que permitan observar el movimiento realizado, así también como características esenciales para determinar la posición y tipo de movimiento que se realiza al mover la extremidad superior de un maniquí.

Se realizará la validación del sistema a través de videos donde se constatará el movimiento realizado en el maniquí y que este se muestre en la interfaz, así también como las características propias de dicho movimiento.

## 1.3 MARCO TEÓRICO

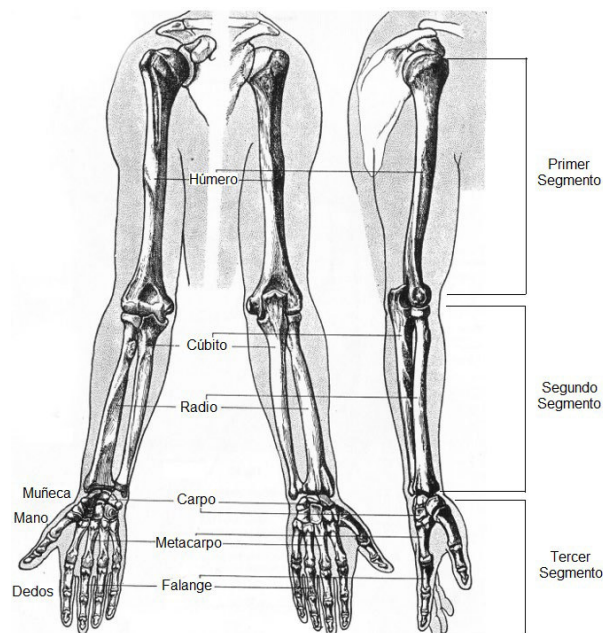
En este apartado se realizará una revisión general sobre las temáticas necesarias para el desarrollo del sistema de emulación de movimiento de la extremidad superior, considerando aspectos que van desde el análisis anatómico del miembro superior hasta el tratamiento de señales que se realizará para obtener los resultados adecuados mediante uso de algoritmos especializados.

### 1.3.1 FISIOLÓGÍA DE LA EXTREMIDAD SUPERIOR

En el presente apartado se realizará un análisis de los movimientos asociados a la anatomía de la extremidad superior del cuerpo humano y su relación directa con la estructura biomecánica del miembro superior [3].

#### 1.3.1.1 Anatomía de la extremidad superior

La extremidad o miembro superior está compuesta del brazo, antebrazo, mano y dedos, los cuales articulan sus movimientos desde la clavícula y un eje compuesto por la espina. El análisis que se realizará en este proyecto es el movimiento realizado por todo el conjunto de la extremidad superior, considerando la Figura 1.1 se pueden observar detalladamente la estructura a nivel óseo de la extremidad y sus principales elementos en los que se tiene el soporte de la estructura del movimiento [4].



**Figura 1.1** Estructura anatómica de la extremidad superior [4]

En base a la Figura 1.1 se delimitarán las partes de la extremidad superior, considerando su estructura anatómica ósea.

- **Primer segmento:** Este segmento es también conocido como brazo y se encuentra constituido por un solo hueso, el húmero.
- **Segundo segmento:** Este segmento es también conocido como antebrazo y se encuentra constituido por los huesos, radio y cúbito o ulna.
- **Tercer segmento:** Este segmento es también conocido como mano y se compone de la muñeca constituida por el carpo, la mano constituida por el metacarpo y los dedos constituidos por las falanges [4].

### 1.3.1.2 Articulaciones y músculos de la extremidad superior

Las articulaciones de la extremidad superior cumplen dos funciones principales, la primera es mantener unidos los huesos del esqueleto y la segunda es dar movilidad a cada segmento del miembro. En este caso se debe considerar de primordial estudio las articulaciones, ya que gracias a estas se puede determinar un movimiento específico de la extremidad. A continuación se describen las articulaciones que permiten obtener el movimiento de flexión, extensión, supinación y pronación del miembro superior [2].

- **Articulación del hombro:** Esta articulación permite realizar el movimiento de abducción y aducción del miembro superior completamente [5].
- **Articulación del codo:** Esta articulación cuenta con dos articulaciones internas las cuales son la articulación sinovial que es la que permite realizar los movimientos de flexión y extensión de todo el miembro, mientras que la otra articulación es proximal o radiocubital la cual permite realizar los movimientos de pronación y supinación [5].
- **Articulación de la mano:** Esta permite obtener mayor grado de libertad en el movimiento de pronación y supinación [5].

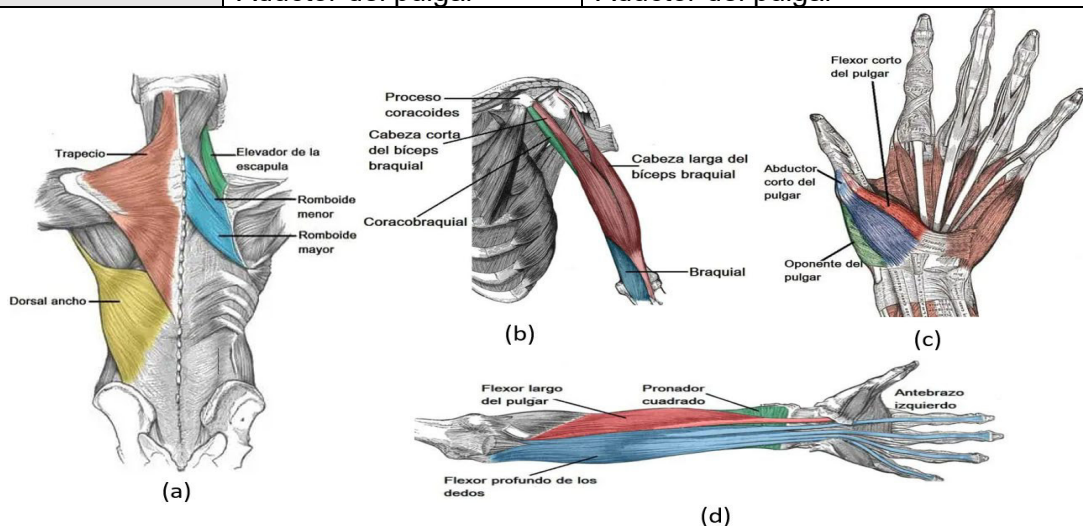
El análisis del grupo de músculos de la extremidad superior también es importante ya que las fuerzas musculares actúan como frenos y motores internos para el movimiento humano, es por ello que el estudio de este proyecto se centrará en las acciones del músculo esquelético que crean el movimiento y lo limitan para obtener distintas posiciones de la extremidad superior [2].

En la Tabla 1.1 y Figura 1.2 se muestran los principales segmentos musculares del miembro superior y los movimientos a los que se limita la extremidad por los ligamentos

musculares, ya que estos pueden limitar el movimiento de las articulaciones. Los músculos presentados en la Tabla 1.1 son los más importantes en el estudio de este proyecto [2].

**Tabla 1.1** Movimientos y descripción de los músculos de la extremidad superior. [5]

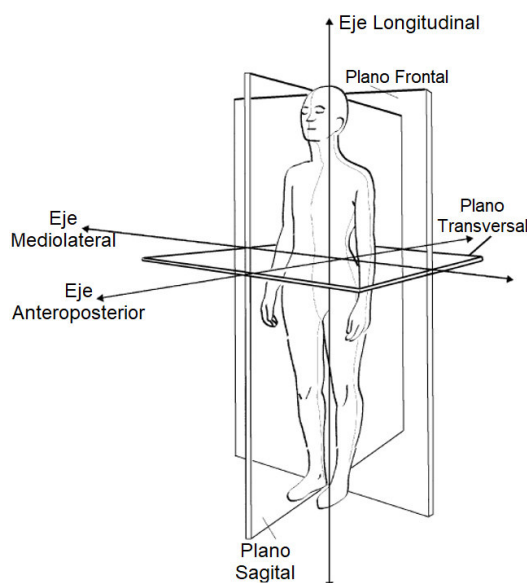
Segmento de la Extremidad Superior	Músculos	Movimientos
Hombro	Deltoides	Abductor del brazo, flexiona y rota
	Redondo Mayor	Aduce y extiende el hombro
	Trapezio	Giran durante la abducción del brazo
	Dorsal Ancho	Extiende, rota, abduce la extremidad superior
	Elevador de la escápula	Eleva la escápula
	Romboide menor	Rota y retrae la escápula
	Romboide mayor	Rota y retrae la escápula
Brazo	Coracobraquial	Flexión en el hombro, aducción débil.
	Braquial	Flexión del antebrazo en la articulación del codo.
	Bíceps braquial	Supinación del antebrazo, flexión del codo y hombro
Antebrazo	Flexor cubital	Flexión y abducción de la muñeca
	Palmar largo	Flexión de la muñeca
	Flexor radial	Flexión y abducción de la muñeca
	Pronador redondo	Pronación del antebrazo
	Pronador cuadrado	Pronuncia el antebrazo
Mano	Lumbricales	Flexión y extensión de las articulaciones interfalángicas
	Interóseos dorsales	Abducir los dedos
	Interóseos palmares	Abduce los dedos
	Aductor del pulgar	Aductor del pulgar



**Figura 1.2** (a) Músculos del hombro (b) Músculos del brazo (c) Músculos del antebrazo (d) Músculos de la mano [5].

### 1.3.1.3 Movimientos de la extremidad superior

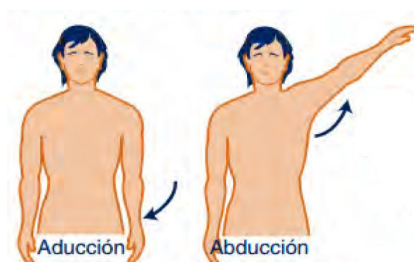
En la Tabla 1.1 se puede ver que existen distintos movimientos que se pueden realizar con la extremidad superior, para tener una mejor referencia de los movimientos se deben definir los planos del movimiento del cuerpo humano, donde se trabajan con tres espacios dimensionales los cuales son: frontal, transversal y sagital, también conocidos como planos anatómicos cuyos ejes son anterior-medio-posterior, longitudinal y lateral. En la Figura 1.3 se muestran los planos anatómicos del cuerpo humano [2].



**Figura 1.3** Plano anatómico del cuerpo humano [2]

Una vez descrito el espacio dimensional en el que se representan los movimientos del cuerpo se procede a describir los más importantes.

- **Abducción:** Se lo describe como un movimiento situado en el plano frontal hacia el eje longitudinal. Ver Figura 1.4.
- **Aducción:** Se lo describe como un movimiento situado en el plano frontal hacia el eje medio-lateral. Es conocido como el movimiento de regreso de la abducción [2]. Ver Figura 1.4.

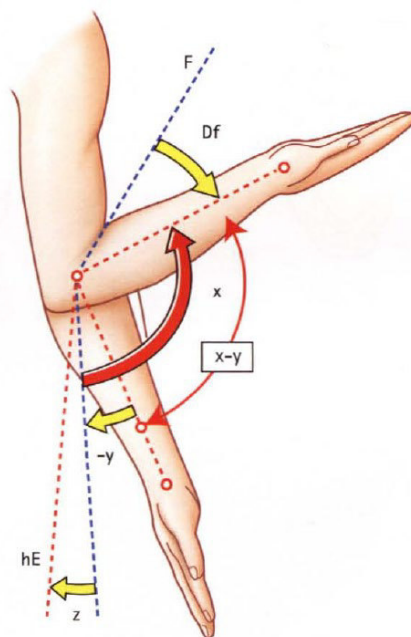


**Figura 1.4** Movimiento de Abducción y Aducción [1]

En el presente proyecto se monitorearán cuatro movimientos en específico, los cuales son los más relevantes al realizar una rehabilitación o detención de alguna patología.

- **Flexión:** Para este movimiento de toda la extremidad superior, se requiere hacer el monitoreo en los movimientos en la articulación del codo, que une el brazo y el antebrazo. Se lo describe como la disminución del ángulo que forma la articulación en el plano sagital [2]. Para describir de una manera más clara es el movimiento que permite que la cara anterior del antebrazo contacte con la cara anterior del brazo. La amplitud de la flexión pasiva es decir cuando se lleva la muñeca al hombro es de  $160^\circ$  [6].
- **Extensión:** Al igual que para la flexión se debe realizar un monitoreo en los movimientos en la articulación del codo. Se lo describe como el aumento del ángulo que forma la articulación en el plano sagital [2]. Para describir de una manera más clara, es el movimiento que dirige el antebrazo hacia atrás y por definición no existe amplitud para la extensión del codo, se lo trabaja con un déficit del de extensión respecto la flexión [6].

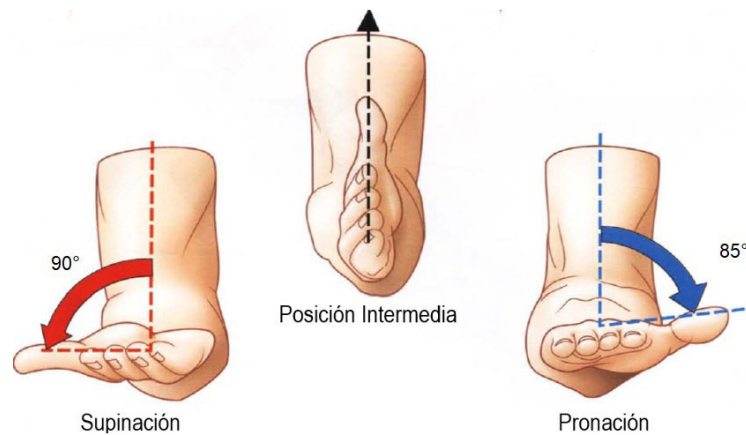
En la Figura 1.5 se puede observar que hay algunas personas que cuentan con una laxitud ligamentosa que pueden lograr una hiperextensión, en la Figura 1.5 se representa como hE, mientras que la flexión se representa como +x y la extensión como -y [6].



**Figura 1.5** Movimientos de Extensión y Flexión de la extremidad superior [6]

- **Pronación:** El monitoreo de este movimiento permite determinar algunas características sobre el plano transversal, se lo realiza cuando la palma de la mano se dirige o “mira” hacia arriba con el pulgar hacia afuera. Cuando la mano y el antebrazo se encuentran alineados la amplitud de movimiento es de 85° [6].
- **Supinación:** El monitoreo de este movimiento permite determinar algunas características sobre el plano transversal, se lo realiza cuando la palma de la mano se dirige o “mira” hacia arriba con el pulgar hacia dentro. Cuando la mano y el antebrazo se encuentran alineados la amplitud de movimiento es de 90° [6].

En la Figura 1.6 se pueden ver los tipos de movimiento y ángulos de limitación para realizar pronación y supinación.



**Figura 1.6** Movimientos de Supinación y Pronación de la extremidad superior [6]

#### 1.3.1.4 Patologías de la extremidad superior

Las patologías de la extremidad superior son de las más comunes, ya que el esfuerzo que se realiza cotidianamente provoca que un movimiento incorrecto con una fuerza errónea aplica a las articulaciones o sistemas musculares, tiendan a crear lesiones en el miembro superior, los tipos de lesiones más comunes son:

- Tendinitis y Bursitis
- Torceduras
- Fracturas
- Dislocaciones
- Osteoartritis

Generalmente cuando se producen este tipo de lesiones una de las principales causas es la afectación en el movimiento de la extremidad por lo que en su recuperación generalmente se recomienda como terapia la rehabilitación física y el establecer la correcta realización de los ejercicios involucrados. Es así como en el siguiente apartado se hablará



de los sensores usados para esta detección de los movimientos involucrados durante la rehabilitación [3].

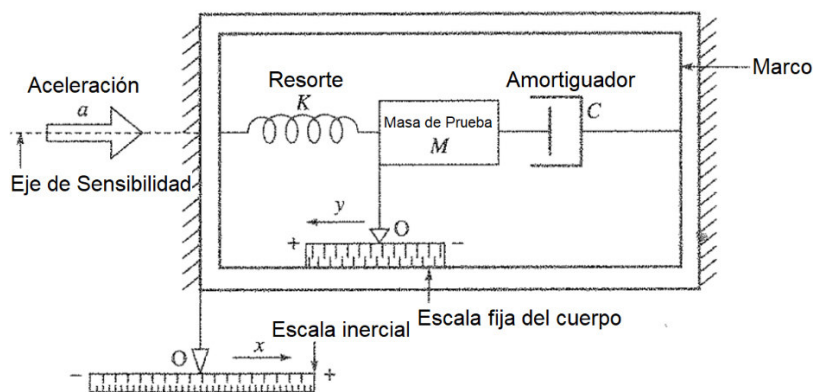
### 1.3.2 SENSORES INERCIALES

Los sensores inerciales también conocidos como unidad de medida inercial representado con sus siglas del inglés **IMU** (Inertial Measurement Units) son sensores que toman tres medidas fundamentales, entre ellas está la velocidad angular, la aceleración y el vector de campo magnético en un sistema de coordenadas propio a nivel tridimensional [7]. En el presente proyecto se trabajará con este tipo de sensores para determinar el movimiento de la extremidad superior utilizando un conjunto de sensores colocados en distintas partes de la extremidad.

Al trabajar con distintas unidades de medida, es sensor IMU cuenta con tres sensores específicos, los cuales en este apartado serán analizados según su principio de funcionamiento.

#### 1.3.2.1 Acelerómetro:

El acelerómetro es uno de los principales sensores que se incorpora en una IMU, este basa su principio de funcionamiento en la segunda ley de Newton que permite medir y estimar la fuerza específica de una masa que se encuentra en estado de vibración cuando el sistema se encuentra en movimiento, generalmente la representación del principio de funcionamiento de un acelerómetro se la realiza con una masa que normalmente se encuentra suspendida utilizando resortes y está se conecta a un sistema que permita indicar aceleración tomando como base el desplazamiento de la masa [8]. Su diagrama de funcionamiento se encuentra en la Figura 1.7.



**Figura 1.7** Principio de funcionamiento de un acelerómetro tradicional [8].

Actualmente, se pueden encontrar acelerómetros electrónicos utilizados en microchips los cuales son capaces de tomar medidas en 3 ejes fundamentales que son  $x$ ,  $y$ ,  $z$  y esto permite medir la aceleración en cualquier dirección. Las unidades de medición de los

acelerómetros que se encuentran en las IMUs son en  $m/s^2$  o en g, una de las características principales de estos acelerómetros es que al encontrarse en reposo estos miden la gravedad [8].

Cuando se trabaja con el acelerómetro de una IMU se deben considerar cuatro tipos de errores fundamentales, entre ellos están el error por escalamiento, desplazamiento, ortogonal y aleatorio. Por lo tanto, la expresión de la medición de aceleración se representa por la Ecuación 1.1, donde se considera el trabajo de los sensores en eje ortogonal y se desprecian los errores aleatorios [9].

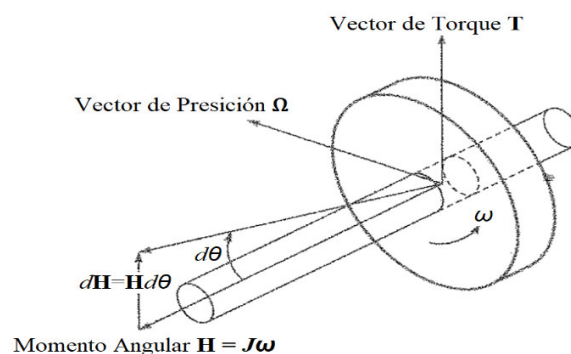
$$(k_r(r - o_r))^2 + (k_s(s - o_s))^2 + (k_t(t - o_t))^2 = g^2 \quad (1.1)$$

Donde:

- $g$  = Gravedad de la tierra ( $9,81m/s^2$ )
- $r, s, t$  = Son los datos del sensor
- $o_r, o_s, o_t$  = Errores de desplazamiento
- $k_r, k_s, k_t$  = Errores de escalamiento

### 1.3.2.2 Giroscopio:

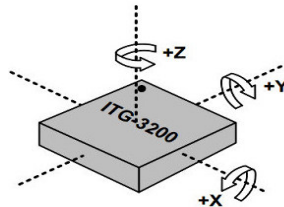
El giroscopio es el segundo sensor implementado en una IMU, permite medir la velocidad angular durante el movimiento de una estructura, generalmente se los ocupa como sensores para estabilización, navegación o pilotos automáticos, por lo tanto, en este proyecto permitirán obtener la característica de estabilización y velocidad angular de la extremidad superior. Este sensor basa su principio de funcionamiento en un rotor con inercia  $J$  que gira a una velocidad angular  $\omega$  y produce un momento angular en el vector  $H$  como se muestra en la Figura 1.8. Cuando esta gira a una alta velocidad hace que su eje de rotación permanezca estable lo que hace difícil que tienda a moverse en esa posición.



**Figura 1.8** Principio de funcionamiento básico de un giroscopio [8]

El giroscopio implantado en las IMU generalmente trabaja con las unidades de velocidad angular en grados/segundo, lo que permiten mantener una orientación de estabilización al momento de realizar las mediciones. Uno de los principales problemas que presentan este tipo de sensores es un fenómeno llamado drifting en el que el sensor respecto al tiempo cambia su error de desplazamiento siendo este incremental, este problema sucede cuando se trabaja con mayor tiempo en la toma de datos, para evitar este problema se debe calibrar al sensor dando una velocidad constante en las tres posiciones del eje para satisfacer las condiciones de medida [9].

En la Figura 1.9 se puede ver los ejes de orientación en los que generalmente trabajan los giroscopios de las IMU.

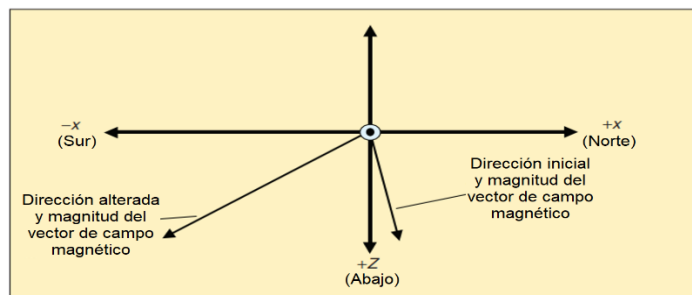


**Figura 1.9** Ejes de medición de un giroscopio [8].

### 1.3.2.3 Magnetómetro:

El magnetómetro permite mantener una referencia de orientación durante la medición de las variables del acelerómetro y el giroscopio, ya que este trabaja con el carácter vectorial del campo magnético lo que produce una medición de magnitud y dirección utilizando la intensidad de campo magnético (H) de la tierra [10].

El funcionamiento de este sensor se basa en el trabajo de la variación de intensidad del campo magnético sobre el sensor, respecto al campo magnético terrestre, es decir al momento de la variación del nivel de orientación del sensor, este cambiará su estimación de dirección y magnitud de orientación, en la Figura 1.10 se muestra el cambio de la dirección y magnitud del vector de campo magnético del sensor al cambiar los niveles de excitación del campo magnético interno del sensor [10].



**Figura 1.10** Principio de funcionamiento de un magnetómetro [10].

Los magnetómetros acoplados a las IMU miden la fuerza del campo magnético en los tres ejes fundamentales y su unidad de medición suele ser en Gauss que es la unidad de medida para el flujo magnético o en  $\mu\text{T}$  que es la unidad de medida de la intensidad de campo magnético, uno de los principales problemas de estos sensores es que sus señal se puede ver distorsionada con la presencia de metales o circuitos electrónicos cercanos, por lo que la unidad de medición debe estar aislada de estos ruidos y dependen del módulo y su capacidad de tratamiento del ruido [10].

La calibración de estos sensores depende de una referencia indirecta externa que se calcula midiendo la velocidad angular del movimiento realizado comparándola con una referencia inicial que generalmente se toma de una brújula lo que permite trabajar con la Ecuación 1.2. para obtener dicha referencia y el valor vectorial de campo magnético medido [9].

$$\varphi = \arctan (H_y, H_x) \quad (1.2)$$

Donde:

- $\varphi$  = Flujo vectorial del campo magnético.
- $H_y$  = Intensidad del campo magnético en el eje y.
- $H_x$  = Intensidad del campo magnético en el eje x

#### **1.3.2.4 Sensor Inercial MPU9250/650**

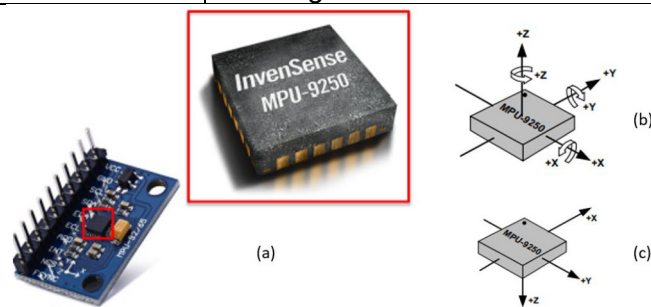
El instrumento principal para la detección de movimientos es la IMU MPU9250, en el apartado anterior se pudo realizar una breve introducción a los sensores inerciales y de igual manera se presentaron las principales características que posee, al ser un sistema integrado por un acelerómetro, giroscopio y magnetómetro, permite obtener detalladamente los datos de orientación, aceleración, velocidad angular, para determinar un movimiento en específico.

El sensor MPU9250 de la Figura 1.11 es un dispositivo MotionTracking de 9 ejes, estos ejes son la combinación de 3 ejes del acelerómetro, 3 ejes del giroscopio y 3 ejes del magnetómetro, cuenta con comunicación I2C para enviar directamente a la salida los datos obtenidos en los 9 ejes. Cuenta con convertidores analógicos digitales (ADC) de 16 bits que permiten digitalizar las salidas del giroscopio, acelerómetro y magnetómetro, cuenta con un ADC para cada eje medido, es decir cuenta con nueve convertidores analógicos digitales. Una característica fundamental es que el dispositivo puede soportar una fiabilidad de impacto de 10000 g [11].

En la Tabla 1.2, se presentarán las principales características de los sensores que incorporan la IMU.

**Tabla 1.2** Características de los sensores incorporados en la IMU MPU9250 [11]

<b>Características del Giroscopio</b>	
Rango de escala programable:	$\pm 250, \pm 500, \pm 1000, \text{ y } \pm 2000$ [°/sec]
Corriente de operación:	3.2 [mA]
Corriente en modo reposo:	8 [μA]
Características adicionales:	<ul style="list-style-type: none"> <li>— Salidas digitales para los ejes x, y, z.</li> <li>— Escala de sensibilidad calibrada de fábrica.</li> <li>— Cuenta con filtro digital pasa bajos.</li> </ul>
<b>Características del Acelerómetro</b>	
Rango de escala programable:	$\pm 2, \pm 4, \pm 8 \text{ y } \pm 16$ [g]
Corriente de operación:	450 [μA]
Corriente en baja potencia:	8.4[μA] a 0.98[Hz], 19.8[μA] a 31.25[Hz]
Corriente en modo reposo:	8 [μA]
Características adicionales:	<ul style="list-style-type: none"> <li>— Salidas digitales para los ejes x, y, z.</li> <li>— Interrupciones programables.</li> <li>— Cuenta con un interruptor para activar la baja potencia del acelerómetro.</li> </ul>
<b>Características del Magnetómetro</b>	
Tipo de sensor:	Sensor de efecto Hall
Resolución de datos a la salida:	14 bits (0.6μT / LSB)
Rango de escala:	$\pm 4800\mu\text{T}$
Corriente de operación:	280 [μA] a 8[Hz]
Corriente en modo reposo:	8 [μA]
Características adicionales:	<ul style="list-style-type: none"> <li>— Cuenta con una función de prueba para confirmar el funcionamiento del sensor magnético.</li> </ul>



**Figura 1.11** (a) Sensor MPU 9250 [14] (b) Ejes de orientación para el acelerómetro y el giroscopio (c) Ejes de orientación para el magnetómetro [11]

### 1.3.3 ADQUISICIÓN DE SEÑALES

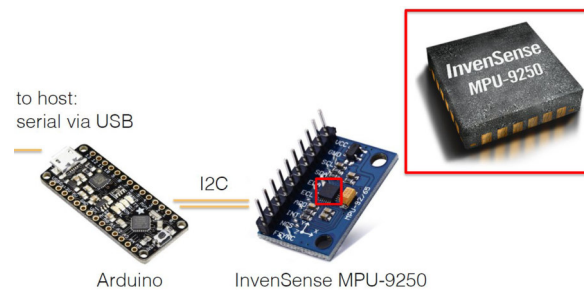
Una vez que se trataron los temas fundamentales como la fisiología de la extremidad superior y los sensores inerciales que serán los que permitan obtener características del sistema, se procede a realizar una descripción de la adquisición de señales donde los conceptos ya antes revisados son importantes ya que permiten tener una noción de la

integración humano – máquina, donde el sistema de adquisición es el microprocesador Arduino nano que es descrito a continuación:

### 1.3.3.1 Arduino Nano

Dentro de los sistemas microprocesados que permiten la comunicación de sensores de unidad de medida inercial se encuentra el microprocesador Arduino Nano que es una placa basada en el Atmega328, está placa destaca por su tamaño que es de 18.5mm x 43.2mm, esto es un beneficio para el presente proyecto ya al ser colocada en el antebrazo, esta debe ser pequeña y compacta, además su velocidad de comunicación en I2C es lo suficientemente rápida para poder trabajar con los otros sensores que se colocarán a lo largo de la extremidad superior [13].

A nivel físico, los puertos que tienen las dos entradas para las señales SDA y SLC son en las entradas analógicas (A4 y A5, respectivamente). En la Figura 1.12 se presenta el esquema de conexión para realizar la comunicación con la placa de Arduino.

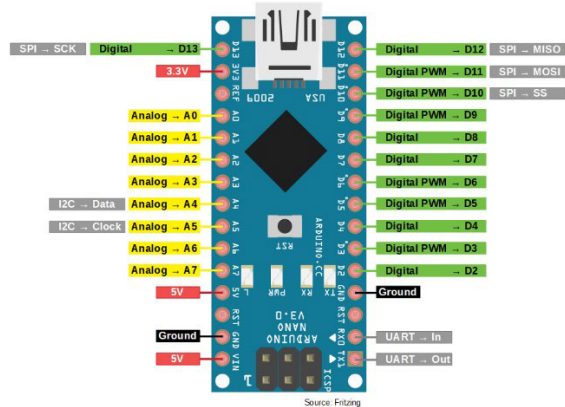


**Figura 1.12** Esquema de comunicación entre Arduino y MPU 9250 [14]

Finalmente, en la Tabla 1.3 se presentan algunas características principales del Arduino Nano, a utilizar en el presente proyecto, mientras que en la Figura 1.13 se muestra un diagrama con la distribución de pines del Arduino Nano.

**Tabla 1.3** Especificaciones Técnicas Arduino Nano V.3 [13].

Características Principales del Arduino Nano V.3	
<b>Microcontrolador</b>	Atmega328
<b>Tensión de entrada</b>	7 – 12 [V]
<b>Tensión de operación a nivel lógico</b>	5 [V]
<b>Puertos de E/S digitales</b>	14 (6, son de salida PWM)
<b>Entradas Analógicas</b>	8
<b>Corriente máxima de cada Pin</b>	40 [mA]
<b>Memoria Flash</b>	32 [KB]
<b>SRAM</b>	2 [KB]
<b>EEPROM</b>	1 [KB]
<b>Frecuencia</b>	16 [MHz]



**Figura 1.13** Distribución de Pines del Arduino Nano V.3 [14]

### 1.3.3.2 Bus de comunicación I2C

El bus de comunicación I2C o I<sup>2</sup>C son siglas del inglés IIC (Inter Integrated Circuit), es decir en general, es la comunicación entre dos circuitos integrados como las memorias EEPROM, RAM, convertidores A/D y D/A, puertos paralelos, etcétera [12].

El uso de este bus para la comunicación entre los sensores y el microprocesador se da por varias ventajas como que un dispositivo tiene la capacidad de actuar como emisor y receptor, y su sentido de enlace deberá ser bidireccional, es decir que para este caso se utilizará una sola línea de datos que sirve para el flujo de los dos sentidos, otra de las principales ventajas es que este bus admite topologías multipunto es decir que se pueden conectar múltiples dispositivos que pueden interactuar entre ellos como emisores o receptores aunque no lo pueden hacer simultáneamente. Estas ventajas se adaptan a las necesidades del proyecto, ya que se necesita tener varios dispositivos conectados en el bus de comunicación como los sensores y el microprocesador [12].

#### Características de señales del bus I2C.

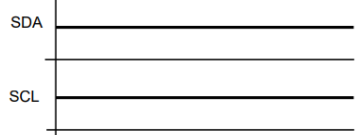
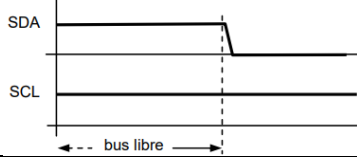
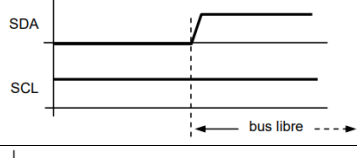
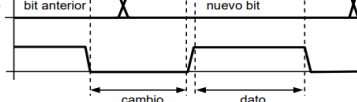
La comunicación del bus I2C depende específicamente de dos señales únicamente, estas son:

- **SDA:** Las siglas son del inglés (Serial Data), que indica que es la línea de datos serie, es una línea bidireccional y hablando a nivel electrónico implica la señal de un colector abierto.
- **SCL:** Las siglas son del inglés (Serial Clock), que indica la línea de señal de sincronía, es una línea bidireccional y hablando a nivel electrónico implica la señal de un colector abierto. Esta señal para un maestro significa una señal de salida, mientras que para un esclavo es la señal de entrada.

## Estados del bus I2C.

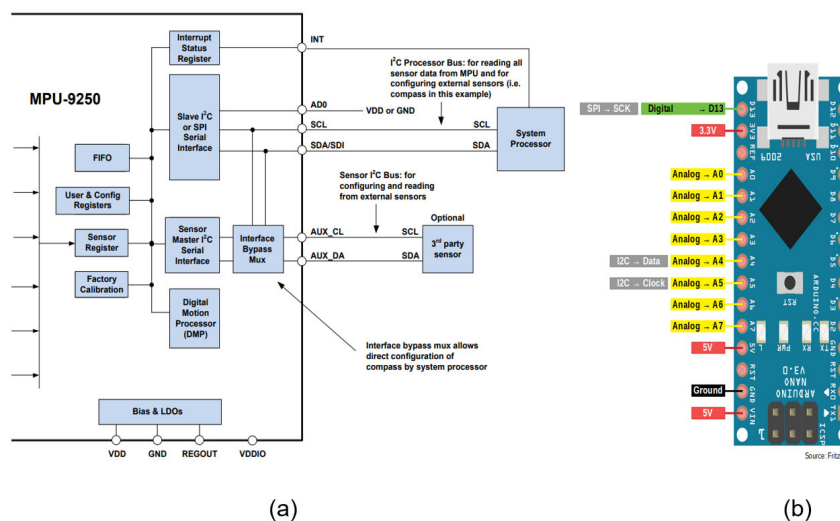
Para el funcionamiento adecuado de la comunicación se debe trabajar con la norma de la trama de datos que define:

**Tabla 1.4** Estados del bus I2C [12].

<b>Libre</b>		Este estado indica que las líneas SDA y SCL se encuentran en 1.
<b>Inicio</b>		Este estado se produce cuando un maestro inicia una transacción de información. La línea SDA cambia del estado alto a bajo.
<b>Parada</b>		Este estado se produce cuando se indica el fin o la parada de transmisión de información y ocurre cuando SDA hace un cambio de bajo a alta y SCL está en alta.
<b>Cambio</b>		Se produce cuando SCL está en bajo la línea SDA puede cambiar de estado, mientras al cambio de SCL se pueden escribir datos.

## Interfaz de comunicación.

En la Figura 1.14 se muestran las interfaces de comunicación para IIC entre la MPU y el Arduino Nano, esto permite obtener una noción del tipo de conexión que se debe realizar para no tener fallos durante la comunicación.



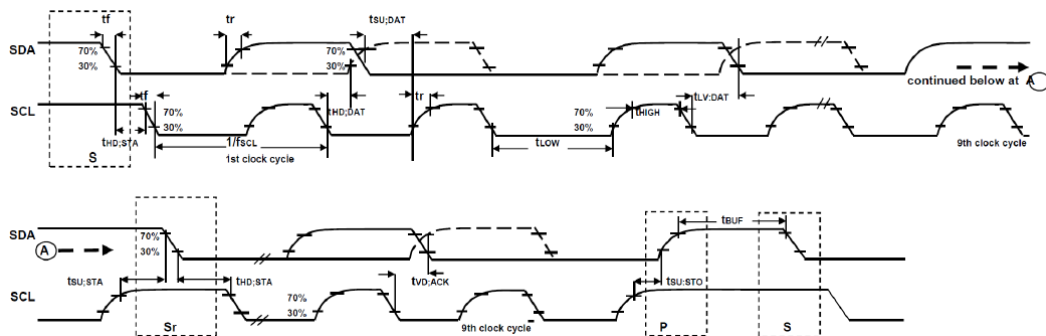
**Figura 1.14** (a) Interfaz de comunicación I2C para MPU9250 [12] (b) Interfaz de comunicación para Arduino Nano [13]



Considerando que en el presente proyecto se trabajará con los sensores IMU MPU9250, en la Tabla 1.5 se presentan las especificaciones técnicas de la comunicación I2C, así también como en el Figura 1.15 se muestran las tramas de comunicación IIC para la MPU9250.

**Tabla 1.5** Especificaciones técnicas de I2C de la IMU MPU9250/6500 [11].

Especificaciones del Magnetómetro					
VDD = 2.4V a 3.6V, VDDIO = 1.71 a VDD, TA=25°C					
Condiciones de I <sup>2</sup> C en Modo Rápido					
Parámetro	Condiciones	MIN	TYP	MAX	UNIDAD
t <sub>LOW</sub> , SCL Periodo bajo		1.3			µs
t <sub>HIGH</sub> , SCL Periodo alto		0.6			µs
t <sub>r</sub> , SDA y SCL Tiempo de subida		20+0.1Cb		300	µs
t <sub>f</sub> , SDA y SCL Tiempo de bajada		20+0.1Cb		300	µs
t <sub>VD.DAT</sub> , Tiempo de validación de datos				0.9	µs



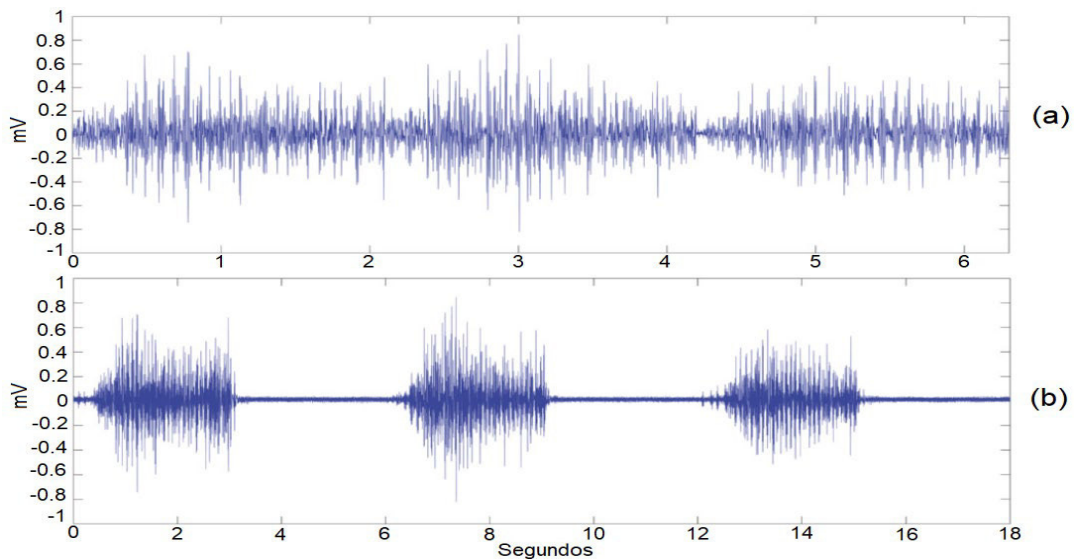
**Figura 1.15** Diagrama de tiempo de la trama de comunicación de I2C de la IMU MPU9250/6500 [11].

### 1.3.4 TÉCNICAS DE PREPROCESAMIENTO

El preprocesamiento de las señales es muy importante en la elaboración del presente proyecto ya que permite clasificar las señales obtenidas por los sensores y realizar el respectivo análisis y tratamiento de los datos antes de entregar un resultado de medición final, ya que existen varios factores que impiden que la señal enviada por el sensor sea un valor de medición final porque existen valores atípicos, señales con ruido que deben ser tratadas para poder obtener un resultado final que en este caso serían posiciones de movimiento de la extremidad superior.

Otra de las principales razones para realizar un preprocesamiento de la señal es entregar las señales medidas por el sensor en un formato más fácil de ser interpretado, eliminando

el ruido, u obteniendo características fundamentales de la señal que permitan interpretar los valores de medición correctamente como se muestra en la Figura 1.16.



**Figura 1.16** Resultado del preprocesamiento de señales de un sensor [16] (a) Señal en crudo (b) Señal pre procesada

Existen varias técnicas para realizar el preprocesamiento de las señales, entre ellas se encuentran la aplicación de filtros, inventado de datos y extracción de características.

#### 1.3.4.1 Extracción de características

Las características de una señal son también conocidos como indicadores, estos permiten crear patrones de datos específicos de la señal y a partir de ello se puede crear un sistema de aprendizaje de los patrones que pueden seguir próximas señales a ser analizadas, las características más comunes se encuentran analizadas en el dominio del tiempo y la frecuencia. Aunque no existe un patrón específico de características especiales que se deban analizar para obtener datos, mientras más información se obtenga de la señal es mejor, ya que se genera un patrón de detección más eficiente, en el presente proyecto se trabajaran con las características que se mencionan a continuación [17].

- **Valor máximo:** El valor máximo de una señal permite reconocer el rango máximo del valor que puede medir la señal, este valor permite delimitar los datos para no tener distorsión durante el entrenamiento de las señales.
- **Valor mínimo:** El valor mínimo de una señal permite reconocer el rango mínimo de un valor que puede medir la señal, este valor permite delimitar los datos para no tener distorsión durante el entrenamiento de las señales.

- **Promedio:** Es una medida de posición de tendencia central, también conocido como media aritmética, toma valores entre el valor máximo y el valor mínimo, una de las principales características es que se ve afectada por la dispersión de datos y permite determinar el punto de equilibrio de la muestra [18].

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad (1.3)$$

Dónde: n es el número de muestras, y  $X_i$  es la señal analizada.

- **Mediana:** Es una medida de posición de tendencia central, esta medida a comparación del promedio no se ve afectada por la presencia de datos atípicos y es más robusto ante las oscilaciones de valores, esta medida busca el punto medio del total de las observaciones que ya hayan sido ordenadas, con este valor se puede tomar el eje de referencia para el muestreo de la señal a ser analizada [18].
- **Moda:** Es una medida de posición de tendencia central, esta busca el elemento o la observación de mayor frecuencia en las muestras de la señal, se utiliza para describir los datos dominantes de un conjunto y gracias a esto se puede determinar una tendencia en la dispersión de datos de las señales [18].
- **Desviación estándar:** La desviación estándar de una señal es utilizada para determinar la potencia de desviación de la señal, esta permite obtener los valores específicos de la variación de la señal en valor absoluto [19].

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_{i+1} - X_i)^2} \quad (1.4)$$

Dónde: N es el número de muestras, y  $X_i$  es la señal analizada.

- **Varianza:** Este valor permite obtener la potencia de una señal, y medir la robustez de esta para determinar la fiabilidad de dispersión en una muestra [19].

$$VAR = \frac{1}{N-1} \sum_{i=1}^N (X_i)^2 \quad (1.5)$$

Dónde: N es el número de muestras, y  $X_i$  es la señal analizada.

- **Valor RMS:** Esta característica permite determinar la generación de retardos homogéneos en una señal para así considerar dichas pérdidas en el algoritmo de detección.

$$RMS = \frac{1}{N} \left( \sum_{i=1}^N (X_i)^2 \right) \quad (1.6)$$

Dónde: N es el número de muestras, y Xi es la señal analizada.

- **Desviación absoluta media o mediana (DMA):** Es la desviación respecto a la media, considerando la diferencia absoluta entre cada valor y permite determinar cuan dispersos se encuentran los datos de la señal [20].

$$DMA = \frac{1}{N} \left( \sum_{i=1}^N |X_{i+1} - \bar{X}_i| \right) \quad (1.7)$$

Dónde: N es el número de muestras, y Xi es la señal analizada.

- **Potencia de banda:** Esta característica, permite obtener el valor de a potencia promedio de la señal y se calcula utilizando la estimación de la densidad espectral de potencia [21].

$$P = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{n=N-1} [X_i]^2 \quad (1.8)$$

Dónde: N es el número de muestras, y Xi es la señal analizada.

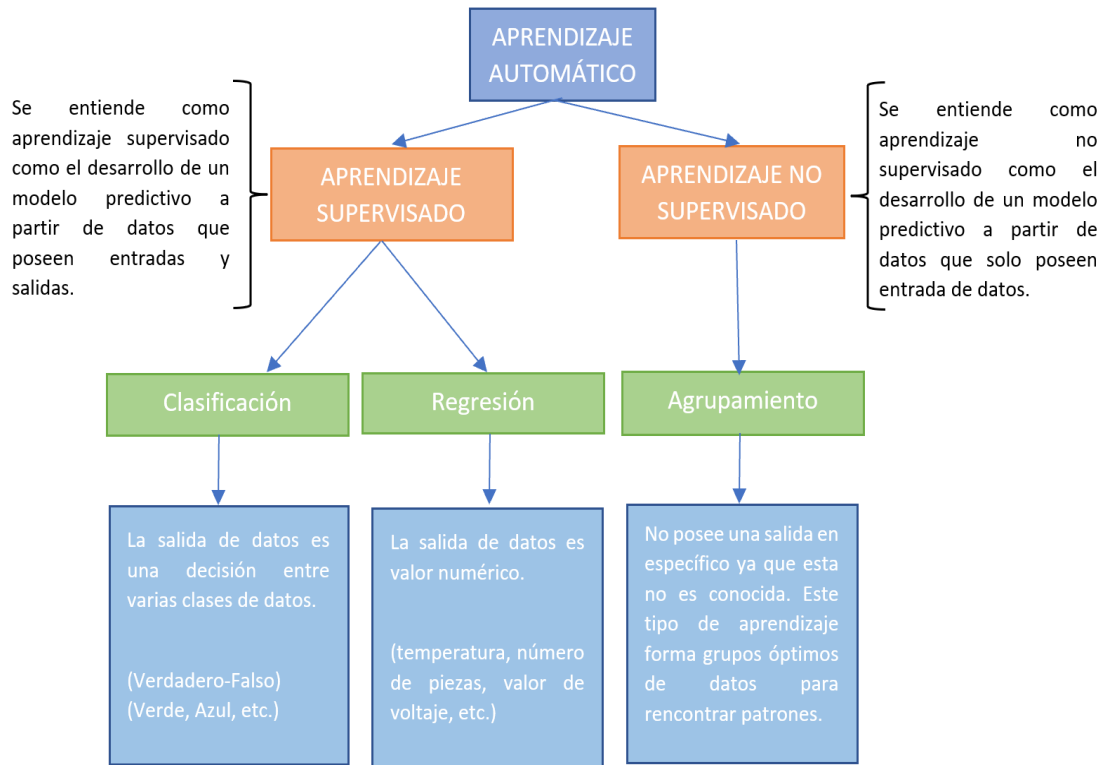
### 1.3.5 APRENDIZAJE AUTOMÁTICO

El aprendizaje automático se entiende como un tipo de inteligencia artificial cuyo objetivo es la creación de un modelo, el cual va a permitir resolver alguna tarea en específico por medio del uso de algoritmos que permitan la interpretación de datos relacionados con dicha tarea. Una característica importante de los datos que utiliza el aprendizaje automático, para la creación de un modelo válido, es que estos deben ser características importantes extraídas de la toma de datos en crudo, para que el algoritmo que se implemente pueda encontrar patrones que permitan un porcentaje alto de validación [22]. El aprendizaje automático, conocido normalmente como “Machine Learning”, se divide en dos grandes grupos como se puede evidenciar en la Figura 1.17.

#### 1.3.5.1 Aprendizaje automático supervisado

El aprendizaje supervisado, como lo indica la Figura 1.17, consiste en trabajar con una base de datos las cuales poseen datos de entrada y salida. Esto quiere decir que el aprendizaje supervisado trabaja con un conjunto de datos, los cuales poseen una etiqueta, donde la intención de esta etiqueta es dividir estos datos obtenidos según características propias de variables las cuales deseamos conocer. Para entender de mejor manera este concepto, se sobrentiende a los datos numéricos obtenidos por la extracción de

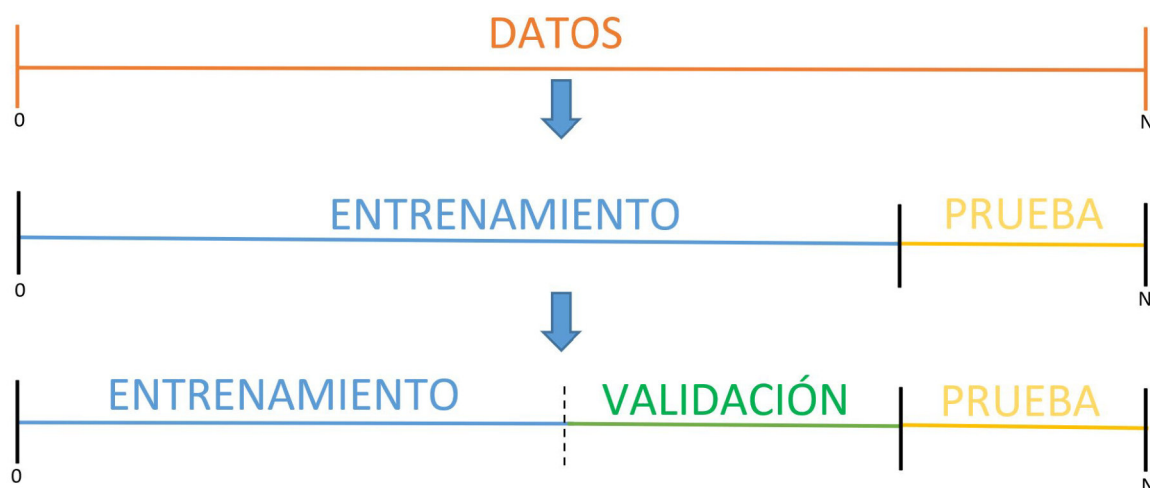
características como la entrada y a las “etiquetas” como las salidas que van a ser el objetivo, con el uso de algoritmos, de encontrar una similitud o patrón adecuado que relacione a estas con el conjunto de datos [23].



**Figura 1.17** Diagrama de división del aprendizaje automático

### 1.3.5.2 Métodos de partición y validación de datos

Dentro del ambiente del aprendizaje automático es importante definir y dividir los datos con los que se está trabajando de manera que se pueda obtener los resultados adecuados de los diferentes clasificadores con los que se puede trabajar, esto con el objetivo de conseguir resultados con alta calidad y fiabilidad de este sistema de aprendizaje [24]. Considerando esta premisa, la determinación la partición de datos es un punto importante para tomar en consideración ya que se debe verificar la manera más acertada y eficaz de dividir los datos con los que se está trabajando según la cantidad que se tenga de los mismos. En el esquema de la Figura 1.18 se puede apreciar de manera general las dos principales formas de división o partición de los datos que se hacen normalmente. El tamaño de cada partición de cada bloque de división dependerá del tamaño de los datos con los que se esté trabajando [25].



**Figura 1.18** Esquema básico de partición de datos

Dependiendo de la cantidad de datos que se tenga y la aplicación que se desee, esta metodología puede cambiar, dependiendo de cuál de ellas se aplique de manera más adecuada. Partiendo del esquema de la Figura 1.18, se entiende lo siguiente:

- **Grupo de datos para entrenamiento:** Este bloque de datos es aquel que va a servir meramente para el entrenamiento de los diferentes clasificadores, encargado de obtener algún patrón de coincidencia de los datos y así obtener un modelo predictivo.
- **Grupo de datos para validación:** La fase de validación, dependiendo cuál método de validación se utilice, permite que se pueda comparar el rendimiento del aprendizaje automático que se esté trabajando mientras los hiper parámetros se van ajustando mediante el entrenamiento de estos.
- **Grupo de datos para prueba:** Consiste en tomar un grupo de datos y reservarlos para una prueba final, para verificar la eficiencia del clasificador con datos que no estuvieron dentro del entrenamiento y validación para que se pueda tener una apreciación imparcial [24].

Para la validación de los datos existen dos técnicas principales, la cual dependerá de la cantidad de datos con los que se esté trabajando. La primera se conoce como *Holdout* que se utiliza cuando existe una gran cantidad de datos y consiste en tomar un porcentaje de la tabla de datos, según las entradas y las salidas que se tenga, e ir validando, determinando si consigue la salida que corresponde con el conjunto de datos que se va ingresando. La segunda validación se conoce como validación cruzada, la cual consiste en tomar de igual manera un porcentaje de los datos de entrada, pero estos datos los va

comparando en pequeños grupos tanto a lo largo de filas como en columnas. Para entender la separación y agrupamiento de los datos para la validación cruzada se puede evidenciar a manera de ejemplo y de forma gráfica la Figura 1.19.

Separación 1	Pliegue 1	Pliegue 2	Pliegue 3	Pliegue 4	Pliegue 5
Separación 2	Pliegue 1	Pliegue 2	Pliegue 3	Pliegue 4	Pliegue 5
Separación 3	Pliegue 1	Pliegue 2	Pliegue 3	Pliegue 4	Pliegue 5
Separación 4	Pliegue 1	Pliegue 2	Pliegue 3	Pliegue 4	Pliegue 5
Separación 5	Pliegue 1	Pliegue 2	Pliegue 3	Pliegue 4	Pliegue 5

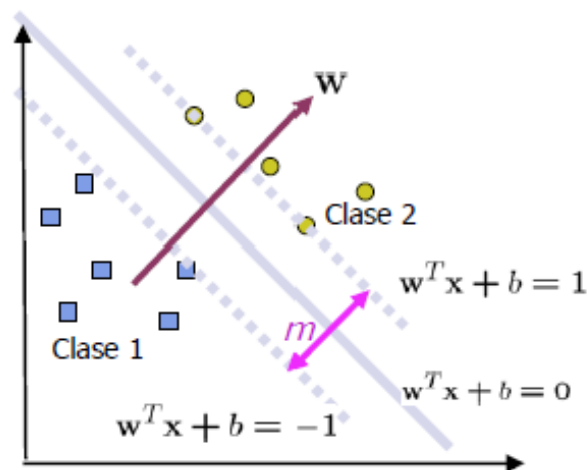
Datos Entrenados

Datos de Prueba

**Figura 1.19** Ejemplo de agrupamiento de datos para la validación cruzada [25]

### 1.3.5.3 Máquina de soporte vectorial – SVM

Una máquina de soporte vectorial o SVM (Support Vector Machine) es un método de clasificación que se basa en el mapeo de los datos de entrada y los pasa a una dimensión mayor a la que se encuentran los datos como tal, para posteriormente crear un hiperplano que separe estos datos en un margen que se lo denomina un margen “m” [26]. Esta denominación se puede entender de mejor manera en la Figura 1.20.



**Figura 1.20** Técnica de clasificación de una máquina de soporte vectorial [26]

De manera más simplificada, la idea principal de una máquina de soporte de vectores es ir comparando los datos, dentro de los planos que se crean al poner como ejes las características que se hayan ingresado en la base de datos, para poder crear superficies que permitan identificar una diferenciación de los datos y poder crear un espacio de

separación que se denomina como un margen tipo “m” para poder identificar a que espacio pertenece cada dato y así poder encontrar una relación entre estos datos y la salida que se desea obtener como resultado. Normalmente, la manera de obtener el margen “m” óptimo es trabajando por medio de los multiplicadores de Lagrange [26].

#### 1.3.5.4 K-vecinos más cercanos – KNN

K-vecinos más cercanos o KNN (K-Nearest Neighbors) es un clasificador sencillo que su idea principal se basa en tomar un dato y a partir de este ir comparando con los datos vecinos o más cercanos para ver si poseen propiedades semejantes y estos se los va agrupando por grupos que al final se lo asemejará con los datos de salida que se desea obtener [27]. Se puede observar en la Figura 1.21 un caso ejemplo de cómo funciona este clasificador.

Como se puede entender, este clasificador permite crear correlaciones de datos según la cercanía o el área que se determine, por lo que la viabilidad de este algoritmo está en la manera de encontrar los valores más cercanos de cada dato y como se compara las propiedades de estos entre sí, para encontrar los grupos más adecuados y poder obtener patrones adecuados para la determinación de las variables de salida deseadas.

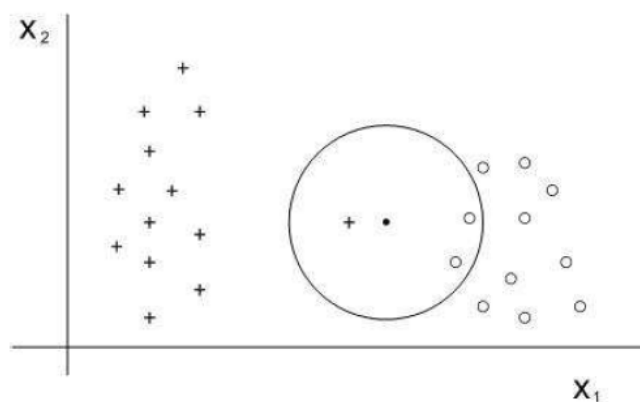


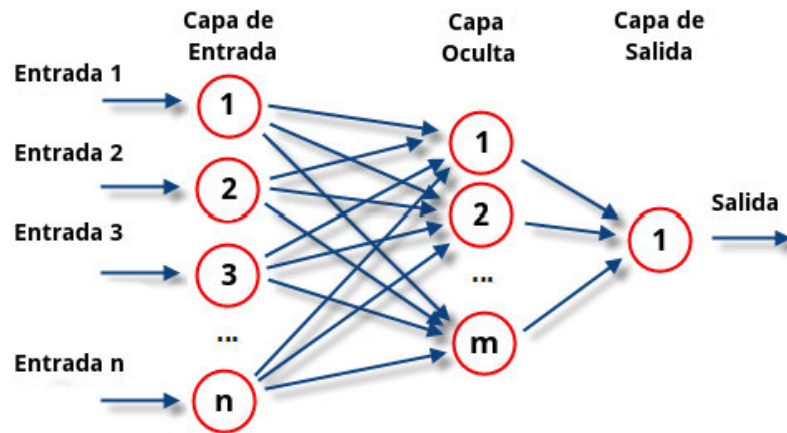
Figura 1.21 Ejemplo del clasificador K-vecinos más cercanos [28]

#### 1.3.5.5 Redes Neuronales

Las redes neuronales, como su nombre lo indica, es un modelo de algoritmo que trata de emular el procesamiento de información que logra el cerebro humano. La metodología de trabajo de este clasificador consiste en trabajar con tres capas que son: capa de entrada, capa oculta y capa de salida. Se toma una capa de entrada y según las propiedades y características que esta posea, va a tener un peso numérico en cada capa oculta. Una vez realizado este sistema en toda la capa de entrada, se verifica una relación entre todas las capas ocultas dentro de la capa de salida para poder determinar el peso numérico que se



obtuvo y determinar una salida como tal [29]. En la Figura 1.22 se puede apreciar de manera más significativa el clasificador de tipo red neuronal.



**Figura 1.22** Funcionamiento de una red neuronal y sus partes [29]

Este clasificador se lo considera uno de los más robustos por lo que tiende a obtener resultados eficientes, donde la clave de este es determinar la capa de entrada de manera eficiente y de esta depende la complejidad del algoritmo y el tiempo de determinación de una respuesta.

## **2. METODOLOGÍA**

En el presente apartado, se detallará la metodología a implementar en el proyecto. Utilizando como principio fundamental la información de la biomecánica de la extremidad superior, para que a partir de los movimientos y posiciones se pueda adquirir toda la información que brindan los sensores. Posterior a esto, se investigará las diferentes metodologías de preprocesamiento y procesamiento de las señales para la creación de una base de datos formada a partir de características relevantes de estas, para su debido entrenamiento y validación de resultados, basándose en información de clasificadores que trabajan con señales de unidades de medición inercial.

### **2.1 ADQUISICIÓN DE DATOS**

La adquisición de datos cumple un procedimiento específico hablando a nivel de hardware y software, el cual se describe a continuación en donde primero se realizará la adquisición de señales a partir de los sensores IMU, luego estos datos se conectarán a un sistema microprocesado utilizando comunicación I2C, y finalmente se tratarán las señales en el software Matlab en manera general. Es por ello que en este apartado se abarcará las ventajas desde el uso de los sensores, recursos de preprocesamiento hasta procesamiento de las señales.

#### **2.1.1 SENSORES DE MOVIMIENTO.**

En el presente proyecto se utilizará la IMU MPU 9250/6500 por las características que fueron descritas en el apartado 1.3.3.1 y también por su facilidad de comunicación el sistema microprocesado Arduino Nano, ya que este tipo de sensor cuenta con un soporte especial del grupo de Arduino, es decir, estos sensores fueron diseñados y probados con el sistema de adquisición de datos de Arduino.

Otra de las razones por las que se eligió este sensor son por las especificaciones del giroscopio, acelerómetro y magnetómetro, estas especificaciones garantizan trabajar con un buen rango de escala, con alta sensibilidad en las mediciones y buena velocidad de toma de datos, estas especificaciones se las presentan en la Tabla 2.1, donde se destacan las principales especificaciones técnicas de cada sensor implementado en la IMU MPU 9250/6500.

En base a la Tabla 2.1, se puede considerar que el sistema de adquisición de datos necesita una alimentación eléctrica DC para los sensores, pero está será entre 2.4V y 3.6V, para el presente proyecto ya que se mencionó que este sensor obtuvo sus parámetros con soporte de Arduino y se utilizara el voltaje de 3.3V que en la placa del Arduino Nano se

puede obtener lo que mantiene el voltaje de alimentación dentro de los parámetros establecidos y podrá trabajar en un rango de temperatura de -40°C a +85°C sin ningún problema [11].

**Tabla 2.1** Especificaciones Técnicas del Giroscopio, Acelerómetro y Magnetómetro de la IMU MPU9250/6500 [11].

<b>Especificaciones del Giroscopio</b>					
VDD = 2.5V, VDDIO = 2.5V, T <sub>A</sub> =25°C					
Parámetro	Condiciones	MIN	TYP	MAX	UNIDAD
Rango de escala	FS_SEL=3		±2000		°/s
Tamaño de palabra ADC			16		bits
Factor de escala de sensibilidad	FS_SEL=3		16.4		LSB/(°/s)
Tolerancia de la sensibilidad	25°C		±3		%
Ruido total RMS	(92 Hz)		0.1		°/s-rms
Tiempo de inicio del giroscopio	Del modo sleep		35		ms
Respuesta del filtro pasa bajos		5		250	Hz
Tasa de salida de datos		4		8000	Hz
<b>Especificaciones del Acelerómetro</b>					
VDD = 2.5V, VDDIO = 2.5V, T <sub>A</sub> =25°C					
Parámetro	Condiciones	MIN	TYP	MAX	UNIDAD
Rango de escala	AFS_SEL=3		±16		g
Tamaño de palabra ADC			16		bits
Factor de escala de sensibilidad	AFS_SEL=3		2.048		LSB/g
Tolerancia de la sensibilidad			±3		%
Ruido total RMS	(94 Hz)			8	mg-rms
Tiempo de inicio del giroscopio	Del modo sleep		20		ms
Respuesta del filtro pasa bajos		5		260	Hz
Tasa de salida de datos		4		4000	Hz
<b>Especificaciones del Magnetómetro</b>					
VDD = 2.5V, VDDIO = 2.5V, T <sub>A</sub> =25°C					
Parámetro	Condiciones	MIN	TYP	MAX	UNIDAD
Rango de escala			±4800		μT
Tamaño de palabra ADC			14		bits
Factor de escala de sensibilidad			0.6		μT/LSB
Calibración inicial			±500		LSB

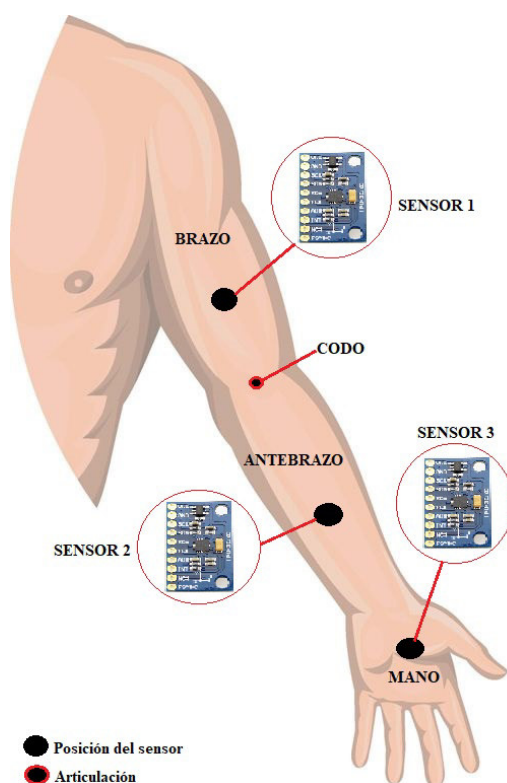
Otra de las características principales del este sensor es que permite realizar la comunicación entre el sistema microprocesado y el sensor utilizando el protocolo I2C, las ventajas de utilizar este tipo de comunicación se las presentarán más adelante. Mediante este sensor se podrá tomar datos de aceleración, velocidad angular del giroscopio y magnetometría con una frecuencia de muestreo de 12510 Hz.

### 2.1.2 LOCALIZACIÓN DE SENSORES.

Los sensores IMU MPU 9250/6500 se encuentran adaptados a una placa que permite obtener los datos y facilita su conexión con la placa de Arduino, sus medidas son 2.1cm x

1.5cm esto hace que esta placa a la cual está conectado el sensor sea lo suficientemente compacta para poder colocarla en el brazo del maniquí.

Considerando que el tamaño de la placa es adecuado para colocarlo en el miembro superior, se debe analizar cuantos sensores y en que los lugares deben colocarse, en el capítulo anterior se mostraron las articulaciones, el sistema muscular que compone la extremidad y los movimientos que se realiza en cada segmento de la mano, es por ello que se opta por colocar 3 sensores a lo largo de la extremidad superior, el primero se colocara en el primer segmento de la extremidad llamado brazo, este sensor servirá de referencia para detectar el giro durante la pronación y la supinación de igual manera permitirá determinar la referencia durante la flexión y la extensión, el segundo sensor se colocará en el segundo segmento llamado antebrazo, este sensor permitirá determinar la flexión y extensión considerando que el antebrazo se une al brazo por la articulación del codo, esta articulación permite el movimiento directo de flexión y extensión de la extremidad superior y se facilitará la detección de movimiento, finalmente se colocará en el último segmento llamado mano, este sensor que permitirá determinar la supinación y la pronación del miembro superior ya que al realizar este movimiento en la mano se muestra un cambio considerable y es más fácil su detección. En la Figura 2.1 se puede ver la ubicación de los sensores en la extremidad superior.



**Figura 2.1** Localización de los sensores en la extremidad superior.

### 2.1.3 SELECCIÓN DEL MICROCONTROLADOR.

Es muy importante determinar el microcontrolador adecuado, según las necesidades del sistema a implementar es por ello que a continuación se presentan los requerimientos que se deben cumplir para el presente proyecto.

#### Requerimientos a cumplir:

Para el presente proyecto se trabajará con sistema microprocesado que cuenta con los requerimientos de la Tabla 2.2.

**Tabla 2.2** Requerimientos para la selección del microcontrolador

Parámetro	IN	OUT
3 sensores IMU MPU 9250 / 6000	1	4
Comunicación inalámbrica bluetooth, módulo HC-06	1	1

Dentro de la Tabla 2.2 se muestran los requerimientos de selección entre ellos se debe aclarar que cada sensor MPU 9250 necesita un pin digital para la asignación de la dirección IIC del sensor, por lo tanto, se requieren 3 señales digitales de salida del microcontrolador. Además, se implementará la comunicación IIC por lo que se requiere un pin de salida para la señal SCL y otro pin para la señal SDA que es bidireccional y funciona como entrada o salida, pero solo requiere un pin del microprocesador.

Por lo tanto, se puede concluir que para el presente proyecto se debe trabajar con un microcontrolador que cuente con un puerto de al menos 7 bits, donde 3 pines serán para la asignación de la dirección IIC, 2 pines para la línea SCL y SDA y 2 pines para la comunicación serial del módulo bluetooth.

#### Asignación de recursos:

En la Tabla 2.3 se muestra la asignación de recursos mínimos para el microcontrolador a implementar en el presente proyecto.

**Tabla 2.3** Tabla de asignación de recursos.

<b>Puerto 1</b>	P1.0	A0 (Sensor 1)
	P1.1	A0 (Sensor 2)
	P1.2	A0 (Sensor 3)
	P1.3	SDA
	P1.4	ACL
	P1.5	RX
	P1.6	TX

### Comparación de microcontroladores:

En la Tabla 2.4 se muestra una comparación de tres microprocesadores que permiten realizar comunicación IIC y comunicación serial asincrónica que son unos de los requerimientos del presente proyecto.

**Tabla 2.4** Comparación de características entre microprocesadores [30] [31] [32]

	<b>Arduino Nano</b>	<b>Atmega 164p</b>	<b>P8X32A</b>
<b>Número de pines</b>	30	40	40
<b>Comunicación IIC</b>	Si	Si	Si
<b>Comunicación serial asincrónica</b>	Si	Si	Si
<b>Pines digitales</b>	14	32	32
<b>Pines analógicos</b>	6	8	6
<b>Oscilador interno</b>	8 - 16[MHz]	20 [MHz]	8 – 20[MHz]
<b>Precios</b>	7 dólares	8,5 dólares	12 dólares

### Selección del microcontrolador:

Después de haber realizado el análisis adecuado para la selección del microcontrolador se elige utilizar el microcontrolador Arduino Nano, considerando las siguientes características:

- Es uno de los microcontroladores que posee la menor cantidad de pines en comparación con los ya estudiados, considerando que estos modelos son de fácil acceso comercial dentro del país.
- Es un microprocesador de bajo costo y cumple con los requisitos mínimos de diseño, lo que hace que el presente proyecto sea económicamente viable para su implementación.
- El ambiente de programación cuenta con librerías específicas para distintos tipos de sensores y dispositivos, en el caso de este proyecto para la IMU MPU 9250 / 6000 y el módulo bluetooth HC-06, facilitando la adquisición y envío de datos permitiendo que estos sean fiables al momento de leer las señales de los sensores.
- El Arduino Nano es una placa robusta en comparación a los modelos analizados, que cuenta con protecciones a nivel de hardware. Estas protecciones son tanto a nivel de alimentación como protección contra el ruido externo.

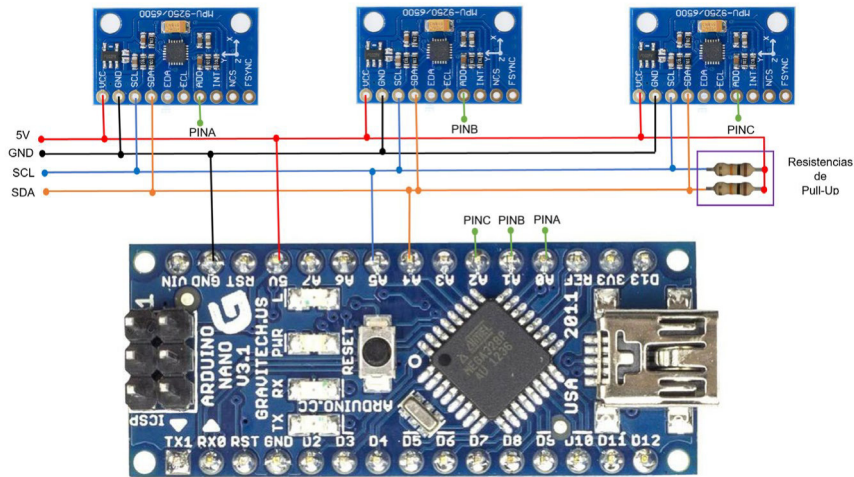
## **2.1.4 FORMATO DE ADQUISICIÓN DE DATOS.**

Una vez que se toman las señales, los datos de las mismas vienen con un formato específico y es una parte fundamental dentro del presente proyecto, ya que este determinará la manera de conexión entre los sensores y la interfaz que se va a tener como resultado final dentro de Matlab. Teniendo en consideración este precedente, dentro del proyecto se tiene dos divisiones principales con respecto a la adquisición de los datos. La primera parte conlleva la conexión de los sensores con el microprocesador por medio de la comunicación I2C y la segunda parte conlleva la conexión del microprocesador con el computador, específicamente con el ambiente de Matlab.

### **2.1.4.1 Comunicación Sensores - Microprocesador**

La primera parte de la adquisición de los datos consiste en la comunicación entre los sensores y el microcontrolador. Esta comunicación se lo hace por medio del protocolo I2C, la cual se hace la toma de los datos según la dirección que se fije según lo proporcionado por el fabricante. La razón por la cual se utiliza la comunicación I2C son por dos factores principales. El primer factor es por la cantidad de pines, ya que la comunicación I2C posee menor cantidad pines disminuyendo la necesidad de uso de más puertos del microcontrolador. El segundo factor es con respecto a la facilidad de obtención de los datos gracias a las librerías de Arduino, ya que una librería en particular trabaja directamente con la comunicación I2C y entrega los datos adecuados, parametrizados y filtrados, según los límites que se configuró inicialmente, además de poseer funciones de inicialización fáciles de manejar como la del filtro digital.

En primer lugar, el sensor MPU9250 posee dos direcciones I2C diferentes 0X68 (dirección principal) y 0x69 en hexadecimal. Para poder cambiar la dirección de la principal a la secundaria se debe alimentar el pin AD0 del MPU9250. Teniendo en consideración que se va a trabajar con 3 sensores que poseen las mismas direcciones, se tendrá una configuración de tal manera que inicialmente todos los sensores tendrán la dirección 0x68, pero en el programa solo se leerá la 0x69 con el objetivo de que, si se desea realizar la lectura de uno de los sensores, se alimenta el pin AD0 del MPU9250 correspondiente y se lee los valores deseados. Ya que son tres sensores, se utilizará tres salidas digitales del Arduino con el fin de activar y desactivar el pin AD0 de cada MPU9250 según se vaya realizando la lectura de los datos. Con este manejo propuesto, a nivel de hardware la conexión quedaría como se muestra en la Figura 2.2.



**Figura 2.2** Conexión de los 3 sensores MPU9250 con Arduino Nano.

Las resistencias de Pull-Up se utiliza regularmente en la comunicación I2C ya que las salidas de esta comunicación se encuentran a colector abierto, por lo que es necesario el uso de resistencias para mantener el estado lógico del pin en valores estándares. El valor de estas resistencias se encuentra determinado dentro de un rango de valores máximos y mínimos que se obtienen a partir de las siguientes fórmulas [33]:

$$R_{min} = \frac{V_{CC} - V_{OL[max]}}{I_{OL}} \quad (2.1)$$

$$R_{max} = \frac{t_r}{0.8473 * C_b} \quad (2.2)$$

Donde:

- $V_{CC}$ : Voltaje del bus de comunicación
- $V_{OL}$ : Voltaje de salida de bajo nivel. Este valor es 0.4 V según [34].
- $I_{OL}$ : Corriente de salida de bajo nivel. Este valor es 3 mA según [34].
- $t_r$ : Tiempo de subida de la rampa de comunicación.
- $C_b$ : Capacitancias del bus de comunicación.

Como se puede verificar en estas fórmulas, depende principalmente del tiempo de subida del puerto de comunicación, de la capacitancia del bus de comunicación, así como del voltaje con el que se está trabajando. Considerando que el voltaje  $V_{CC} = 5V$ , la velocidad de I2C es estándar, obteniendo  $t_r = 1\mu s$  [X1] y la capacitancia de los puertos de cada uno



de los sensores y del microcontrolador [35] es  $C = 10 \text{ pF}$  dando un total de  $40 \text{ pF}$  para la capacitancia del bus de comunicación, tenemos los siguientes resultados.

$$R_{min} = \frac{5 - 0.4V}{3 \text{ mA}} = 1533.33 \text{ } [\Omega] \quad (2.3)$$

$$R_{max} = \frac{1 * 10^{-3}}{0.8473 * 40 * 10^{-12}} = 29505.5 \text{ } [\Omega] \quad (2.4)$$

Considerando estos resultados y la existencia de posibles capacitancias parásitas correspondientes a los cables y la placa, se determinó que el valor de resistencia de Pull-Up sea de 10 Kohm.

Con el uso de librerías propias de Arduino, se obtiene los datos de manera directa y sin necesidad de realizar ningún tipo de acondicionamiento o escalamiento de valores. Estos valores se encuentran dentro de los rangos con los que se determinó en la inicialización a nivel de software, los cuales se pueden observar en la Tabla 2.5.

**Tabla 2.5** Escala de valores de las señales de los sensores inerciales con los que se trabaja en el presente proyecto

Escala Giroscopio	Escala Acelerómetro	Escala Magnetómetro
$\pm 250 \text{ DPS}$	$\pm 2 \text{ G}$	$\pm 4800 \text{ } \mu\text{T}$

#### 2.1.4.2 Comunicación Microprocesador - Computador

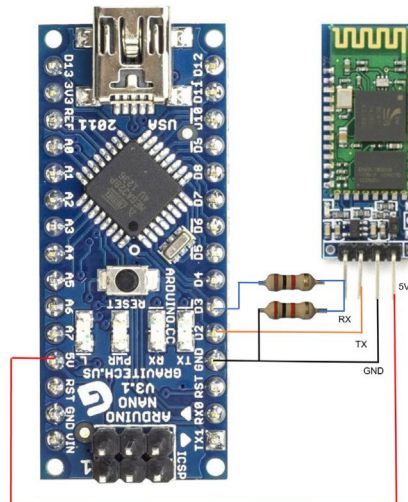
Una vez obtenido los valores de los sensores, es primordial determinar el tipo de comunicación que se va a utilizar para el envío de estos datos al computador. Teniendo en consideración que el presente proyecto se hace uso de un maniquí y que debe existir la menor cantidad de elementos que alteren físicamente la posición de este, por lo que, se determinó el uso de un elemento de comunicación inalámbrica que permita el movimiento del maniquí de manera cómoda. Para ello, se hace uso de un módulo bluetooth HC06 el cual permite crear un canal de comunicación serial vía inalámbrica en la computadora para poder comunicarse con cualquier aplicación dentro del computador. Las principales características de este módulo bluetooth se pueden observar en la Tabla 2.6.

**Tabla 2.6** Especificaciones del módulo bluetooth HC-06 [36]

ESPECIFICACIONES MÓDULO BLUETOOTH HC-06	
Parámetros	Valores y modos de operación
Voltaje de operación	$3.3V - 5V$
Corriente de operación	$< 40 \text{ mA}$

Corriente modo sleep	< 1 mA
Chip	BC417143
Tipo de bluetooth	V2.0+EDR
Frecuencia	Banda ISM de 2.4 GHz
Modulación	GFSK (Gaussian Frequency Shift Keying)
Potencia de Emisión	4dBm, clase 2
Sensibilidad	-84dBm a 0.1% VER
Alcance	Máximo 10 metros
Interfaz de Comunicación	Serial UART TTL
Velocidad de transmisión	1200bps hasta 1.3 Mbps
Baudrate por defecto	9600, 8, 1, n
Velocidad asíncrona	2.1 Mbps (máx)/ 160 kbps
Velocidad síncrona	1Mbps/ 1Mbps
Seguridad	Autenticación y encriptación
Dimensiones	37*16 mm
Peso	3.2 gramos

A nivel de hardware, la conexión del módulo bluetooth es sencillo ya que este posee simplemente 4 pines que se conectan directamente a los pines del Arduino como se muestra en la Figura 2.3.



**Figura 2.3** Conexión módulo bluetooth con Arduino Nano

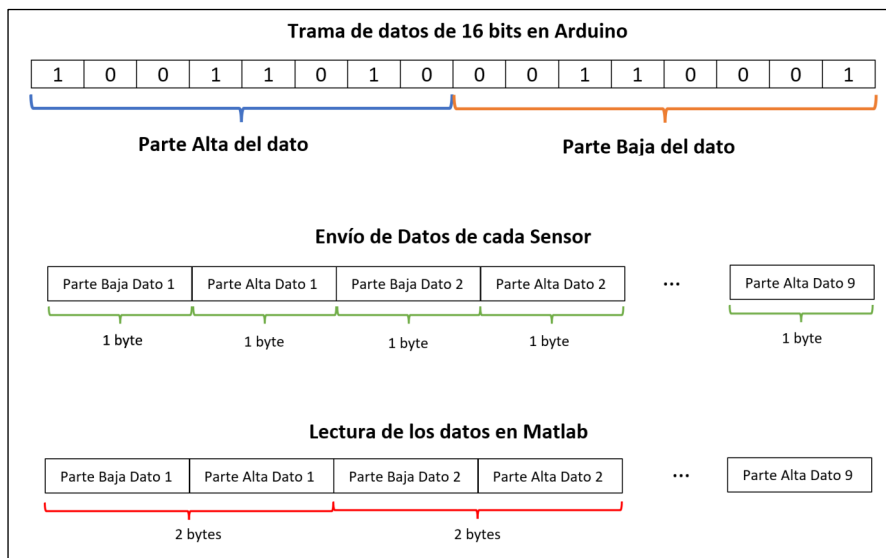
Como se puede observar en la Figura 2.3, se tiene un divisor de voltaje en el pin de RX ya que, en este módulo bluetooth, este pin no permite valores de voltaje superiores a 3.3V. Por esa razón, los valores de resistencia son de 1 Kohm al pin D3 y de 2 Kohm a GND.

Debido a que se tiene una comunicación de tipo serial, existen limitaciones con los datos que se tiene ya que este solo permite el envío de un máximo de 8 bits, es decir un valor máximo de 255. Al obtener los valores de los sensores en los diferentes rangos, se tiene valores mayores a 255 por lo que se realiza un acondicionamiento de los datos para tener valores entre  $\pm 32767$  y posteriormente dividir este valor en parte alta y parte baja para enviar por separado y en el software de Matlab construir de nuevo este valor. Para los cálculos de acondicionamiento, se hace uso de las escalas con las que se trabaja que se muestran en la Tabla 2.5 dando como resultado las siguientes ecuaciones:

$$valor\ acelerómetro = \frac{señal * 32767}{2 * 9.81} \quad (2.1)$$

$$valor\ giroscopio = \frac{señal * 32767}{250} \quad (2.2)$$

$$valor\ magnetómetro = \frac{señal * 32767}{4800} \quad (2.3)$$



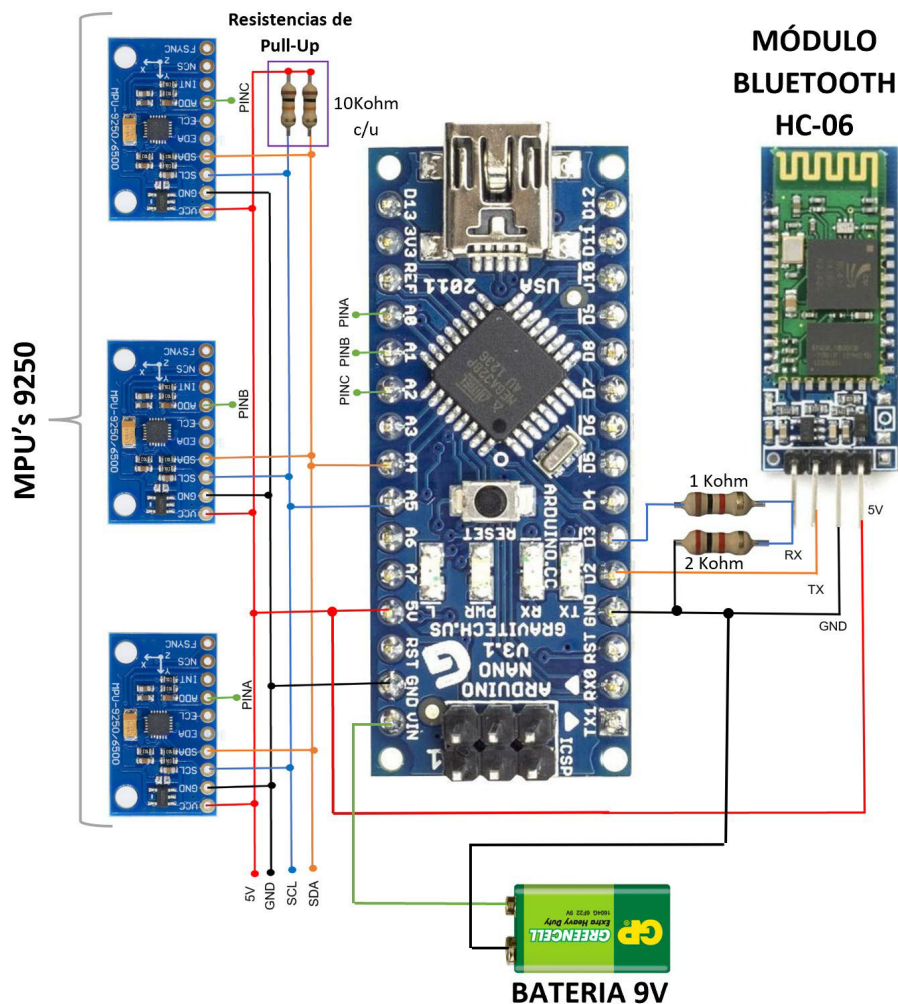
**Figura 2.4** Trama de envío de datos de Arduino y su lectura en el ambiente de Matlab

Para determinar la manera de envío tanto de la parte alta como de la parte baja de cada uno de los valores de los sensores, se lo hace a partir de la función de lectura del software de Matlab. Dentro del ambiente de Matlab se puede comunicar con dispositivos bluetooth por medio del uso de la función "bluetooth" la cual permite habilitar la lectura y escritura dentro del bus que crea el módulo HC06 en el computador. Una vez abierta la vía de

comunicación, se procede al acceso de los datos que con la función “read” puede leer valores de 16 bits con signo, si la información llega la parte baja primero y posteriormente la parte alta. Por ello, la manera de envío sería primero la parte baja de los datos de los sensores y posteriormente la parte alta de los mismos. Véase Figura 2.4.

### 2.1.5 HARDWARE IMPLEMENTADO

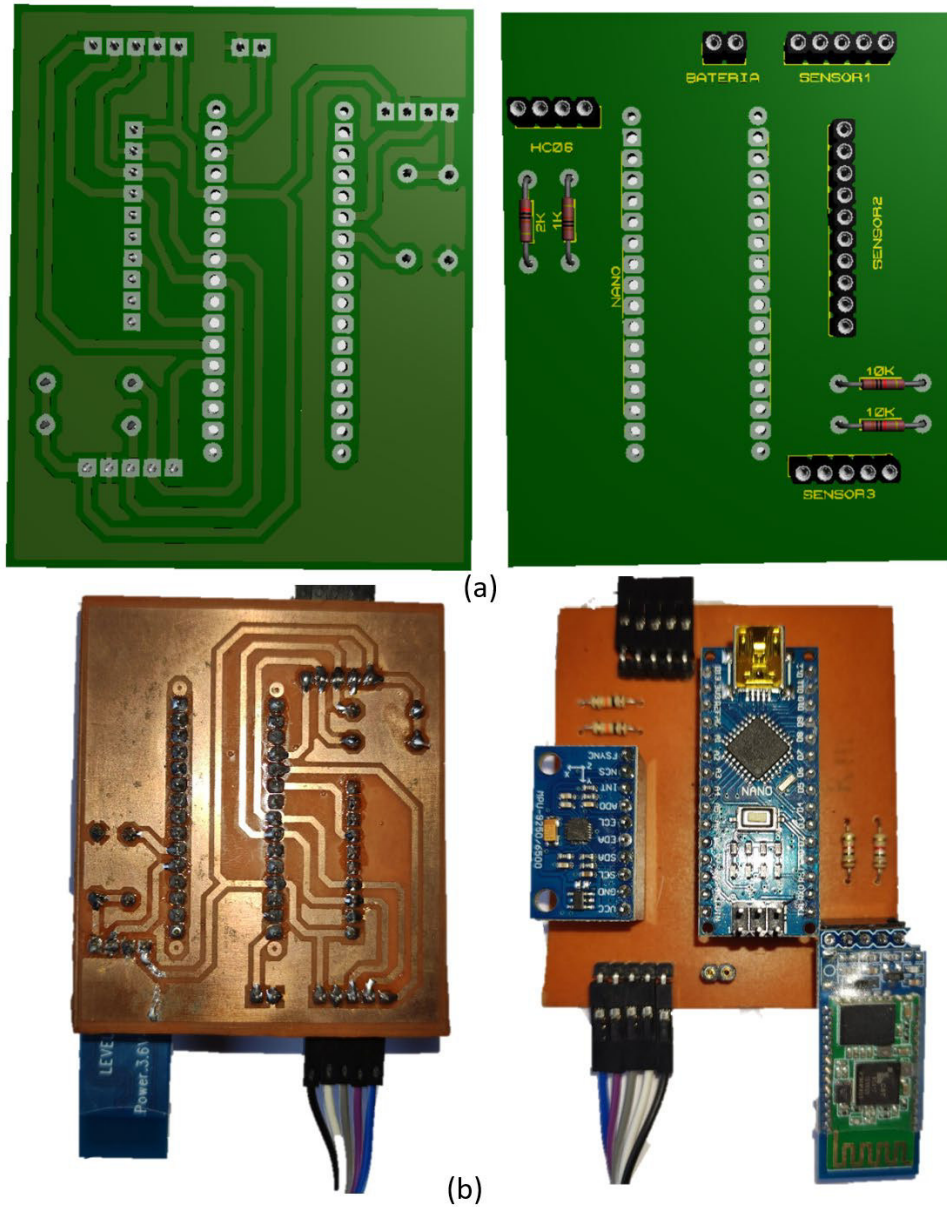
Como se indicó en el apartado 2.1.4, el circuito completo consta de tres sensores de medición inercial MPU-9250, un Arduino Nano, un módulo bluetooth HC-06 y un conjunto de resistencias que conforman tanto las resistencias de Pull-Up y las de control de voltaje del pin del receptor de módulo de comunicación bluetooth. Además, todo el sistema esta alimentado con una batería de 9V. Las conexiones del circuito final se las puede evidenciar en la Figura 2.5.



**Figura 2.5** Conexión del circuito realizado para el proyecto

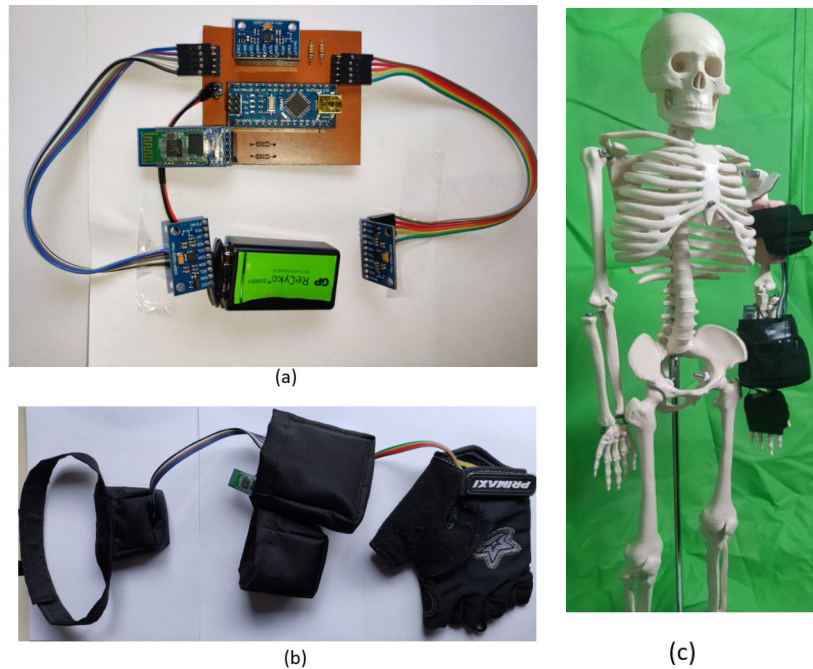
Finalmente, se realizó la implementación de este circuito en una placa que va a ir ubicada, al igual que los sensores, a lo largo de la extremidad superior que hace uso de dos correas

que van ubicadas en el brazo y antebrazo y un guante para el sensor ubicado a la altura de la mano. En la Figura 2.6 se puede evidenciar el circuito implementado en las placas de PCB tanto el modelado de la palca en el programa de Proteus y como estos van ubicados dentro del sistema de correas y el guante en la Figura 2.7.



**Figura 2.6** (a) Cara posterior y frontal de la placa modelada en Proteus, (b) Cara posterior y frontal implementada





**Figura 2.7** (a) Circuito completo implementado, (b) sistema implementado para la extremidad superior, (c) implementación de las bandas en el maniquí.

A modo de información adicional, se tiene algunas características adicionales del hardware final implementado:

- La batería posee una autonomía de 15 horas aproximadamente, debido al consumo de energía que se genera por todos los elementos que conforman el sistema final.
- Se realizó el modelo físico conformado por las diferentes correas que se ubican a lo largo de la extremidad superior, en consideración de diferentes pruebas de posicionamiento. Estas pruebas de posicionamiento se enfocaron en colocar los sensores en diferentes regiones de la extremidad superior ya sean que los sensores se encuentren los más juntos posibles, los más alejados posibles o en puntos medios de cada región. Con estos diferentes posicionamientos se determinó que la posición más acertada de cada uno de ellos sea en los puntos medios de cada sección de la extremidad superior conformada por el brazo, antebrazo y mano, ya que los datos poseían rasgos característicos diferenciados entre sí, favoreciendo el proceso de análisis de datos que se realizará posteriormente.

## 2.2 PREPROCESAMIENTO DE LA SEÑAL

La etapa de preprocesamiento cumple un rol importante para el tratamiento de los datos, ya que esta etapa se la considera como la etapa de limpieza de los datos debido a que es la encargada del mejoramiento de los datos para posteriormente, poder utilizarlos en las

diferentes aplicaciones que se desee. El preprocesamiento se puede llegar a dividir en dos principales fases: la primera que constituye la preparación de los datos y la segunda fase se conoce como la de extracción o reducción de datos. [37]

La fase de preparación de los datos conlleva realizar diferentes procesos o técnicas que permitan el mejoramiento previo de los datos, que estos no posean ruido, datos anómalos, vacío de datos, entre otros; con el objetivo de tener los datos iniciales con la mínima cantidad de perturbaciones [37]. Subsecuentemente, para la fase de extracción o reducción de datos se debe realizar un análisis de las características de los datos que se considere primordiales o que posean una alta relevancia según la forma que posean los datos.

Considerando las dos fases primordiales del preprocesamiento, dentro del presente proyecto se realizará dos diferentes técnicas para la fase de preparación de los datos que conlleva la prevención de datos anómalos y la eliminación de datos fuera de rango. Se realiza únicamente estas dos técnicas de preparación de los datos debido a la manera de lectura de movimiento obtenida por los sensores, dichas señales poseen la forma que se muestra en la Figura 2.4, donde no se requiere un tratamiento exhaustivo de los datos posterior a la toma de estos, ya que se puede identificar y dividir zonas específicas según el tipo de movimiento fácilmente sin recurrir a filtros enfocados a mejorar la señal; por lo que la técnica más importante a tomar en consideración, según los datos obtenidos, en esta etapa es la detección y eliminación de datos anómalos, ya que afecta directamente al momento de extracción de características. Si este valor anómalo se encuentra fuera de una zona normal, el valor característico puede variar drásticamente y el clasificador puede dar resultados erróneos. Además, la segunda técnica consiste en un filtrado inicial que se da al momento de la toma de estos, gracias a que Arduino proporciona una función enfocada en los sensores MPU9250 como se explicará a continuación, que permiten eliminación de valores que sobrepasen cierta frecuencia que se considera como ruido y permite que la cantidad de datos anómalos disminuya, por lo que estas dos técnicas se complementan entre sí.

Por otra parte, para la fase de extracción o reducción de datos se hará uso de diez diferentes metodologías, mencionadas en el capítulo anterior, que permitirán obtener una base de datos amplia para su posterior uso en los diferentes clasificadores.

### **2.2.1 TÉCNICAS DE PREPROCESAMIENTO DE DATOS**

En el actual proyecto se hará uso de dos técnicas de preparación de los datos. La primera técnica es por medio del uso de un filtro digital pasa bajo DLPF (Digital Low Pass Filter) que permite utilizar una función propia de Arduino con el MPU9250. La segunda técnica

consiste en la eliminación de datos anómalos, esto con el objetivo de que todos los datos posean la misma regularidad para que al aplicar las diferentes técnicas de extracción de datos no se obtenga rangos fuera de lugar.

### 2.2.1.1 Filtro Digital Pasa Bajos - DLPF

Un filtro digital pasa bajos tiene como objetivo permitir el paso o toma de señales que se encuentren por debajo de una frecuencia específica, que se conoce como frecuencia de corte, por medio de un algoritmo de programación. Considerando esta idea principal se sobreentiende que todo dato que se encuentre por encima de esta frecuencia de corte se lo considera como valor no deseado o ruido. [38]

Dentro del ambiente de Arduino se puede optar por varias librerías, las cuales cumplen con el objetivo de comunicarse de manera fácil que al realizarlo manualmente. Considerando la utilización del sensor de unidad inercial MPU9250, se hace uso de la librería correspondiente en Arduino, la cual da la opción de utilizar un filtro digital pasa bajos. Esta función permite trabajar a valores específicos de ancho de banda como se puede observar en la Tabla 2.7.

**Tabla 2.7** Selección de ancho de banda y el retardo que provoca en cada señal de giroscopio y acelerómetro [39]

Ancho de Banda - DLPF	Retardo Giroscopio	Retardo Acelerómetro
184 Hz	2.9 ms	5.8 ms
92 Hz	3.9 ms	7.8 ms
41 Hz	5.9 ms	11.8 ms
20 Hz	9.9 ms	19.8 ms
10 Hz	17.85 ms	35.7 ms
5 Hz	33.48 ms	66.96 ms

Cabe mencionar que este filtrado de datos se lo realiza solamente al giroscopio y al acelerómetro, ya que esta función se la puede utilizar gracias a que el sensor de medición inercial MPU9250 posee una MPU6050, de la cual permite esta opción de filtrado de datos.

Como se puede observar en la Tabla 2.7, se puede trabajar en un total de seis diferentes frecuencias. El objetivo de este filtrado es que, si se toma el menor valor de ancho de banda, se obtendrá una filtración de ruido muy alta pero el retardo en la obtención de los datos será mayor; de manera inversa, si se escoge la frecuencia más alta, se tendrá menor filtrado pero el retardo de obtención de los datos es menor. El creador de esta función recomienda trabajar a una frecuencia de 20 Hz ya que esta posee un filtrado adecuado de los datos para la mayoría de las aplicaciones y el retardo no es excesivo [39].

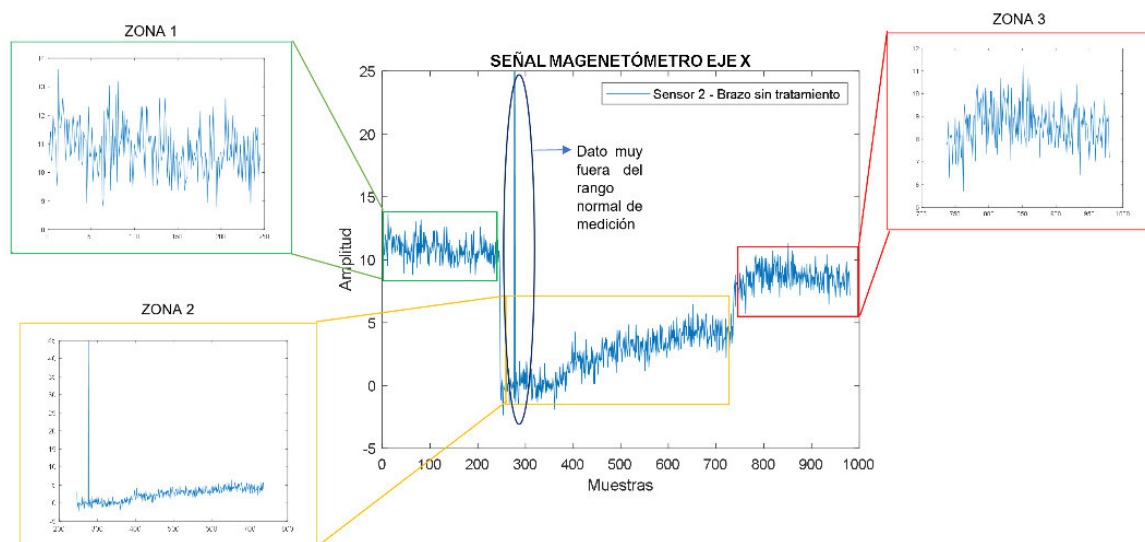


Tomando en consideración las recomendaciones del creador de la función y en adición con las pruebas realizadas, se decidió realizar el filtrado digital pasa bajos a una frecuencia de corte de 20 Hz ya que esta permitía que la mayoría de los datos anómalos, que la mayoría se presentaban en el giroscopio, sean eliminados y el retardo no influía de manera significativa al momento de la comunicación con Matlab.

### 2.2.1.2. Eliminación de datos anómalos

La eliminación de datos anómalos tiene como objetivo la limpieza de los posibles datos que se encuentren fuera de rango o sea muy diferente a los demás visualmente. Para determinar si un valor se encuentra fuera de rango se requiere de un análisis a nivel de software para la determinación de parámetros ideales a seguir, para que con esta información base se pueda determinar posibles picos considerados como ruido y la técnica que más se acople para la eliminación de estos. Un ejemplo de un dato anómalo se lo puede observar en la Figura 2.8.

En principio, se parte de la toma de datos dentro de Matlab creando una pequeña base de datos con todos los datos que se obtuvieron de los sensores, separándolos según sean las señales del giroscopio, acelerómetro y magnetómetro en los diferentes ejes x, y, z. Posteriormente, se grafica estos datos para verificar si existen datos que se encuentran fuera del rango normal con respecto a los demás.



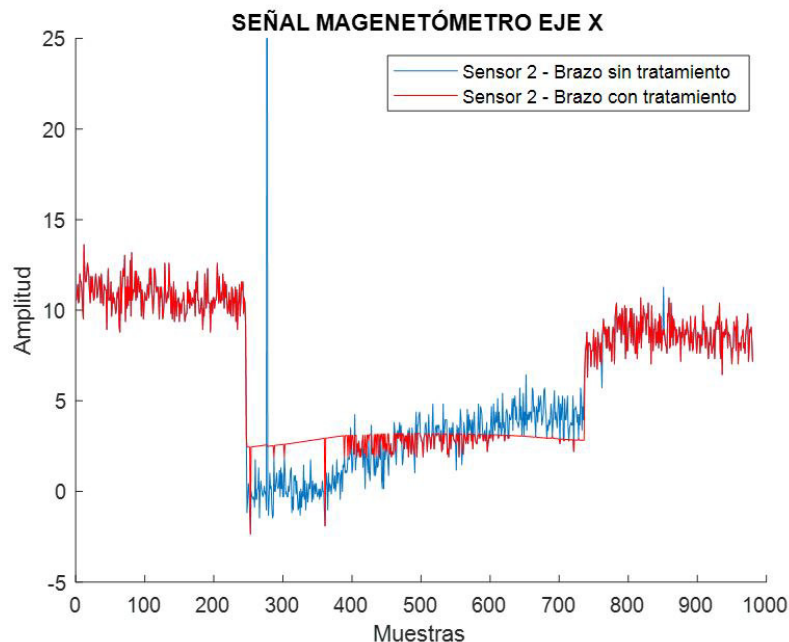
**Figura 2.8** Zonas de trabajo para la eliminación de datos anómalos

Posteriormente, se debe toma en cuenta la división de los datos. En este caso, los datos se clasifican de dos tipos, movimientos entre flexión-extensión y movimientos entre pronación-supinación, por lo que, para poder encontrar una técnica de eliminación de ruido que se acople a los datos que tenemos, se trabaja en 3 zonas diferentes. Ya que todos los

datos tomados se encuentran en el mismo rango de valores, las zonas se dividen de la siguiente manera: la zona 1 comprende en el rango de 1-245 muestras, la zona 2 comprende en el rango de 247-736 muestras y la zona 3 comprende en el rango de 738-980 muestras. En la Figura 2.8 se puede observar la diferenciación entre estas zonas.

Una vez dividido los datos en zonas, se procede a tomar el valor más frecuente o la moda del valor absoluto de estos. Se lo realiza con respecto al valor absoluto ya que el análisis de los datos se facilita si estos se encuentran siempre positivos. Una vez obtenido el valor más frecuente, se analiza cada valor que se encuentra dentro de cada zona y se lo compara a un  $\pm 30\%$  del valor de la moda. Esto quiere decir, que si el valor analizado está 1.3 veces o 0.7 veces la moda, este valor se identifica como anormal. Se realiza el análisis con respecto a un  $\pm 30\%$  de la moda debido a que además de eliminar los picos, también ayuda a disminuir de manera significativa el rizado que se tiene en cada zona y también ayuda a estabilizar los datos que se tiene.

En el caso de que se detecte un dato considerado anómalo, este será reemplazado con el valor promedio que se obtenga de la zona a la que corresponde, logrando que la eliminación de datos erróneos se haga efectivo y se tenga una señal como se muestra en la Figura 2.9.



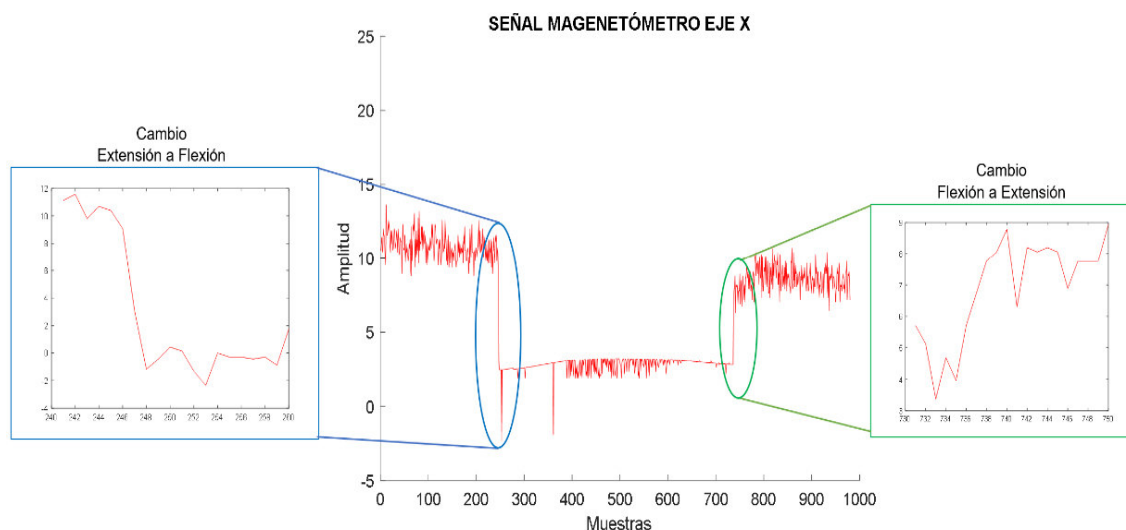
**Figura 2.9** Comparación entre la señal antes y después del tratamiento de datos

## 2.2.2 EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características es un paso fundamental para la creación de la base de datos final, que servirá como punto de partida para los clasificadores. Esta extracción de

características se divide en varios pasos para lograr obtener los datos con las características adecuadas.

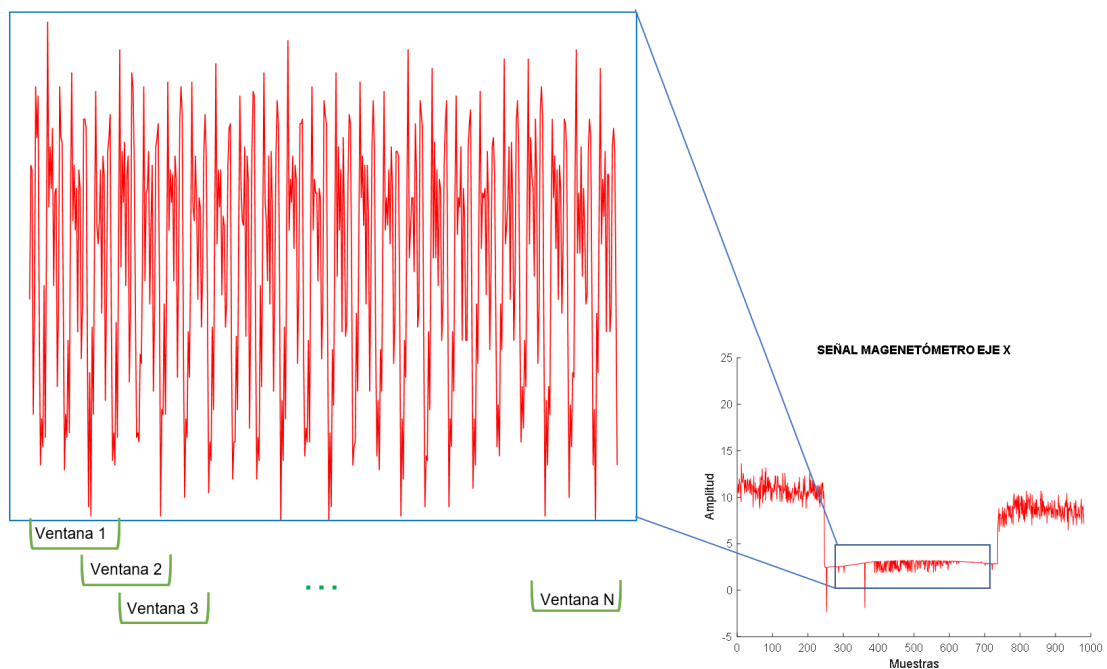
Primero, se debe dividir los datos ya preprocesados según el tipo de movimiento. Cada una de las muestras tomadas se asemejan a la señal de color rojo de la Figura 2.9, donde se tiene dos movimientos, ya sea extensión-flexión o pronación-supinación. Tomando en consideración la forma de partición de datos o creación de zonas de la Figura 2.8; la zona 1 y zona 3 corresponde a un tipo de movimiento (extensión o pronación), mientras que la zona 2 corresponde al otro movimiento (flexión o supinación). Analizando la cantidad de datos, se tiene 40 señales tomadas de cada variable que entregan los sensores, donde 20 de estas son movimientos entre extensión-flexión y los otros 20 movimientos entre pronación-supinación. Cada una de estas señales son de aproximadamente 15 minutos, que en número de muestras son 980 muestras. Realizando una toma de datos según las zonas de división de tal manera de no tomar algún dato entre los cambios de movimiento, se tiene 480 muestras de cada variable del sensor por cada movimiento principal, y si se tiene 20 muestras por cada movimiento, da como resultado un total de 9600 muestras de cada variable del sensor por cada movimiento principal, que son: extensión, flexión, pronación y supinación. De igual manera, se realiza la toma de 20 datos para el análisis entre los cambios de movimiento que sería de extensión a flexión, flexión a extensión, pronación a supinación y supinación a pronación dando como resultado 400 muestras de cada variable del sensor por cada tipo de cambio. Esta toma de datos con respecto al cambio de movimiento se puede visualizar en la Figura 2.10.



**Figura 2.10** Ejemplo de toma de valores entre cambios de movimientos

Una vez dividido los datos por movimiento se debe realizar un inventariado de las señales para posteriormente sacar las características de cada ventana que se consiga. Este

enventanado consiste en dividir y agrupar los datos en pequeños grupos, por ejemplo, que de una señal de 1000 datos se lo puede dividir en 100 ventanas de 10 datos cada una. Otro concepto importante al realizar el enventanado de las señales, es el porcentaje de traslape que se debe tener. Este traslape se refiere a que porcentaje de datos de la ventana anterior se va a tomar para formar mi nueva ventana, con el ejemplo anterior se puede entender que si se tiene un traslape de 50% se va a formar la primera ventana con los primeros 10 valores y la segunda ventana serán los últimos 5 valores de la primera ventana y los siguientes 5 valores que siguen después de la primera ventana teniendo en total 199 ventanas de 5 valores cada una. Esto se lo hace con el fin de obtener una relación óptima al momento de determinar las características de los datos y de aumentar la cantidad de datos que se tiene. Dentro del presente proyecto, se realizará un enventanado de 5 valores con un traslape del 80%, dando como resultado 9596 ventanas por cada movimiento principal y 396 ventanas por cada cambio de movimiento. La Figura 2.11 muestra un ejemplo visual de cómo se realiza el enventanado de un conjunto pequeño de datos del proyecto.



**Figura 2.11** Ejemplo de enventanado de un grupo de datos correspondiente a los datos de flexión

Una vez teniendo los datos separados según el tipo de movimiento y realizado el enventanado, se empieza con la extracción de características de los datos de manera que de cada ventana tenga un valor único de la característica deseada. Se utilizará 10 funciones matemáticas para valorar las características de cada una de las ventanas. En la

Tabla 2.8 se muestra las funciones que se van a utilizar y sus variables dentro del ambiente de Matlab determinando si es una característica en tiempo o en frecuencia.

**Tabla 2.8** Funciones características a utilizar

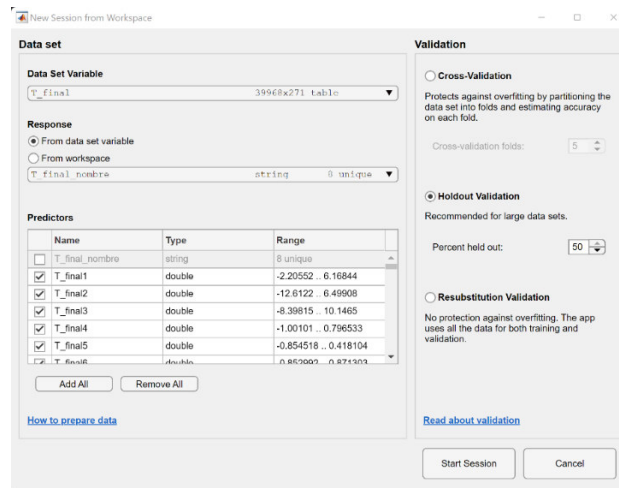
Función	Función en Matlab	Tipo de Característica
Valor máximo	max(grupo de datos)	tiempo
Valor Mínimo	min(grupo de datos)	tiempo
Valor Promedio	mean(grupo de datos)	tiempo
Valor frecuente	mode(grupo de datos)	tiempo
Valor medio	median(grupo de datos)	tiempo
Desviación estándar	std(grupo de datos)	tiempo
Varianza	var(grupo de datos)	tiempo
Valor RMS	rms(grupo de datos)	tiempo
Desviación absoluta media	mad(grupo de datos)	tiempo
Potencia de Banda	bandpower(grupo de datos)	frecuencia

Al realizar la extracción de características se posee una base de datos amplia ya que se tiene 9 señales de cada sensor del giroscopio, acelerómetro y magnetómetro en cada uno de sus ejes, que contando los 3 sensores son 27 señales de los sensores. De cada señal se tiene 9596 ventanas de movimientos principales y 396 de cambios de movimientos; sacando las 10 características principales de cada una de estas ventanas se tendría una base de datos compuesta de 2'590.920 datos por cada movimiento principal y 106.920 datos por cada cambio de movimiento dando como resultado una tabla de 10'791.360 datos en total. Finalmente, una vez realizado la extracción de los datos, se debe especificar en qué grupos de datos o valores pertenecen a tal movimiento, para que esta tabla final se pueda ingresar a los diferentes clasificadores.

## 2.3 PROCESAMIENTO DE LA SEÑAL

Una vez obtenida la base de datos que posee los datos de entrada (los valores característicos) y los valores de salida deseados (los 8 diferentes tipos de movimiento), se procede a realizar las diferentes pruebas de los diferentes clasificadores que proporciona Matlab. Para ello, se hará uso del aprendizaje automático para la determinación del tipo de movimiento, para lo cual, se debe realizar la partición de datos adecuados según la cantidad de datos que se posea.

Principalmente, dentro del ambiente de Matlab 2021b, se utiliza la aplicación "*Classification Learner*", la cual cuenta con varios clasificadores donde el usuario debe escoger como parámetros iniciales el tipo de validación que desea realizar y el porcentaje de partición de datos que se desea como se muestra en la Figura 2.12.



**Figura 2.12** Ventana de selección de datos, tipo de validación y partición de datos

Analizando la base de datos obtenida, se parte de una cantidad superior a 10'000.000 de datos, por lo que el software de Matlab aconseja que se utilice una validación de tipo de retención o de "Holdout" ya que la de validación cruzada requiere un tiempo considerable en lo que corresponde a la validación debido a la técnica que utiliza, mientras que la de retención solo toma el porcentaje de datos que se indica al inicio y los prueba, una vez realizado el entrenamiento correspondiente [25]. Considerando este antecedente de la cantidad de datos, se va a efectuar la partición de datos de 50%-50% donde la mitad de los datos sirve para el entrenamiento y la mitad sobrante para la validación. La partición de datos de prueba no se va a realizar como tal, ya que esta fase se lo va a implementar de manera que a medida que se van tomando nuevas medidas de los sensores, y después de un adecuado tratamiento de estos datos, se lo va a ingresar a estos modelos obtenidos y verificar los resultados según el movimiento que esté posicionado el maniquí.

### 2.3.1 TÉCNICAS DE CLASIFICACIÓN

La aplicación de Matlab permite elegir entre una alta cantidad de clasificadores, los cuales poseen sus características propias que la calidad de sus resultados dependerá de las características tomadas de los datos. Dentro de la Tabla 2.9 se tiene, a manera de resumen, todos los clasificadores que se puede utilizar en Matlab y todas las variantes que estos poseen.

**Tabla 2.9** Clasificadores que se pueden utilizar en el ambiente de Matlab [40]

Clasificador	Tipo de algoritmo	Descripción
Árboles de clasificación	Árbol de clasificación Fino	Un árbol de decisión con muchas hojas que realiza varias distinciones finas entre cada clase. Posee 100 divisiones máximo.
	Árbol de clasificación Medio	Un árbol de decisión con flexibilidad media con menos hojas. Posee 20 divisiones como máximo.

	Árbol de clasificación Robusto	Es un árbol de decisión simple con pocas hojas que hace distinciones robustas entre clases. Posee 4 divisiones como máximo.
Máquina de Soporte Vectorial	SVM Lineal	Una máquina de soporte vectorial que crea una simple separación lineal entre clases usando la linealización de Kernel.
	SVM cuadrático	Una máquina de soporte vectorial que realiza la separación utilizando la función cuadrática de Kernel.
	SVM cúbico	Una máquina de soporte vectorial que realiza la separación utilizando la función cúbica de Kernel.
	SVM Gaussiano Fino	Una máquina de soporte vectorial que realiza una distinción entre clases de manera finamente detallada, usando la función Gaussiana de Kernel con la escala de Kernel ajustada a $\sqrt{P}/4$ , donde P es el número de predictores.
	SVM Gaussiano Medio	Una máquina de soporte vectorial que realiza menores distinciones entre clases que la fina, usando la función Gaussiana de Kernel con la escala de Kernel ajustada a $\sqrt{P}$ , donde P es el número de predictores.
	SVM Gaussiano Robusto	Una máquina de soporte vectorial que realiza distinciones robustas entre clases, usando la función Gaussiana de Kernel con la escala de Kernel ajustada a $\sqrt{P} * 4$ , donde P es el número de predictores.
K-Vecinos más cercanos	KNN Fino	Es un clasificador que realiza distinciones finamente detalladas entre clases, con el número de vecinos ajustado a 1.
	KNN Medio	Es un clasificador que realiza menor cantidad distinciones entre clases, con el número de vecinos ajustado a 10.
	KNN Robusto	Es un clasificador que realiza distinciones robustas entre clases, con el número de vecinos ajustado a 100.
	KNN Coseno	Un clasificador KNN que hace uso de la distancia métrica cosenoidal.
	KNN Cúbico	Clasificador KNN que hace uso de la distancia métrica cúbica.
	KNN Ponderado	Clasificador KNN que hace uso de la distancia ponderada.
Clasificadores de Aproximación de Kernel	SVM Kernel	Un clasificador Gaussiano de Kernel para clasificación no lineal de datos con varias observaciones. El mapa predictor de Kernel dentro en un espacio de dimensión alta y se ajusta al modelo SVM lineal con los predictores transformados.
	Regresión Logística de Kernel	Un clasificador Gaussiano de Kernel para clasificación no lineal de datos con varias observaciones. El mapa predictor de Kernel dentro en un espacio de dimensión alta y se ajusta a los modelos de regresión logística lineal con los predictores transformados.
Clasificador de conjunto (Ensemble)	Árboles potenciados	El modelo crea un conjunto de árboles de decisiones medio usando el algoritmo AdaBoost. Comparado con el de árboles ensacados, este algoritmo usa relativamente menos tiempo o memoria, pero necesita mayor conjunto de miembros.
	Árboles ensacados	Es un conjunto potenciado-agregado de árboles de decisiones. Usualmente, es muy preciso, pero tiende

		a ser lento y bastante memoria para un amplio número de datos.
	Subespacio Discriminante	Es muy bueno para varios predictores, es relativamente rápido para ajuste y predicción y posee bajo uso de memoria, pero la precisión varía dependiendo el tipo de datos. El modelo crea un conjunto de clasificadores discriminantes usando el algoritmo de subespacios al azar.
	Subespacio KNN	Se usa este clasificador si se tiene varios predictores. El modelo crea un conjunto de clasificadores de vecinos cercanos usando el algoritmo de subespacios al azar.
	Árboles RUSBoosted	Se usa este clasificador si se tiene datos sesgados con muchas más observaciones que los de una clase.
Redes Neuronales	Red Neuronal Estrecha	Una red neuronal con una capa completamente conectada de tamaño 10.
	Red Neuronal Media	Una red neuronal con una capa completamente conectada de tamaño 25.
	Red Neuronal Amplia	Una red neuronal con una capa completamente conectada de tamaño 100.
	Red Neuronal Bicapa	Una red neuronal con dos capas completamente conectadas, excluyendo la capa final completamente conectada para clasificación.
	Red Neuronal de tres capas	Una red neuronal con tres capas completamente conectadas, excluyendo la capa final completamente conectada para clasificación.
Análisis Discriminante	Discriminante Lineal	Un clasificador discriminante rápido y fácil de interpretar, que crea límites lineales entre clases.
	Discriminantes Cuadrático	Un clasificador discriminante rápido y fácil de interpretar, que crea límites elípticos, parabólicos o hiperbólicos entre clases.
Clasificador Naive Bayes	Naive Bayes Gaussiano	Es un clasificador Naive Bayes que utiliza la distribución Gaussiana por número de predictores y distribuciones MVMN para predictores categóricos.
	Naive Bayes de Kernel	Es un clasificador Naive Bayes que utiliza la distribución de Kernel por número de predictores y distribuciones MVMN para predictores categóricos.

Teniendo en consideración la amplia cantidad de clasificadores que posee Matlab, se procede a realizar una selección de aquellos con los que se puede obtener resultados más favorables. En primer lugar, considerando la gran cantidad de datos y características con los que se está trabajando, utilizar clasificadores que requieran una toma de decisiones con un análisis amplio entre datos y que llegue a utilizar un tiempo de simulación y de memoria significativo, se deben descartar, como es el caso de árboles de clasificación, clasificadores de Kernel y clasificador de conjunto o ensemble debido a la cantidad de decisiones que llegan a tener, además de que estos clasificadores pueden demorar un tiempo considerable tanto al momento del entrenamiento y la validación como al momento del uso de estos a modo de función en el producto final. Tampoco se hará uso de los clasificadores análisis discriminante y Naive Bayes ya que estos son poco usados en

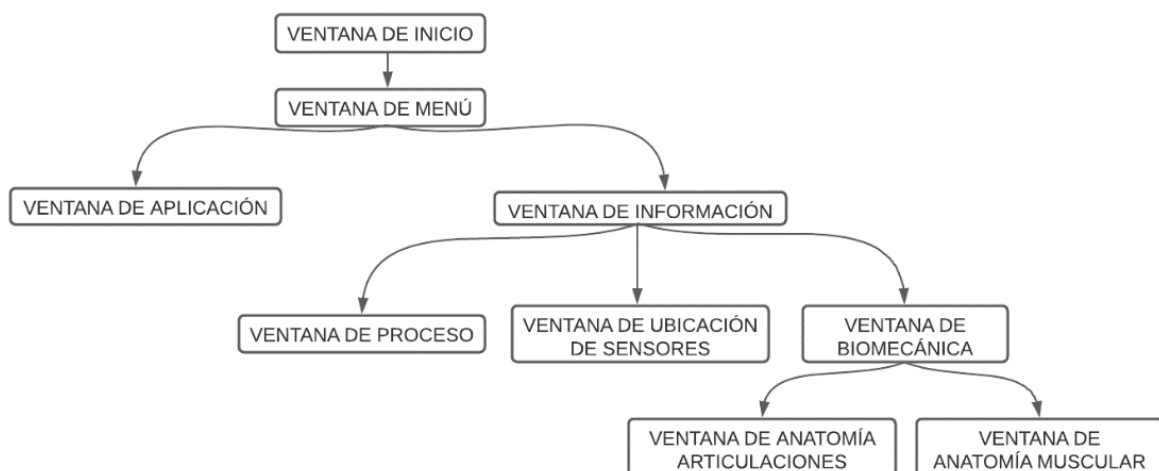


aplicaciones de detección de movimientos o tienden a tener un porcentaje bajo de efectividad [41]. Además, basándose en la investigación de [41] y [42], se considera utilizar los métodos de clasificación K-vecinos más cercanos (KNN) y máquina de soporte vectorial (SVM), ya que los resultados obtenidos con estos clasificadores, con el uso específico de acelerómetros y giroscopios, son altos en efectividad. Finalmente, también se analizará el uso de redes neuronales, ya que, por tipo de datos obtenidos, este clasificador podría tener resultados satisfactorios, debido a que no hace uso de diferencias por distanciamiento entre datos, como el KNN o SVM, sino por la relevancia y semejanza de prioridades que existe entre ellos.

## 2.4 DISEÑO DE LA INTERFAZ GRÁFICA

La interfaz gráfica de usuario implementada para el presente proyecto técnico será desarrollada en el software Matlab2021b de MathWorks®, con el objetivo de realizar una interfaz dinámica que permita navegar entre pantallas para obtener información relevante sobre el proyecto y sus utilidades, así como la presentación del resultado final del proyecto, se optó por utilizar la herramienta App Designer de Matlab, esta herramienta permite crear aplicaciones con un interfaz gráfico de alta calidad y dinamizar la interacción con el usuario. Finalmente se presentan dos secciones importantes en el interfaz, la primera es la información sobre el proyecto, como la ubicación de los sensores y el sistema anatómico que permite el movimiento de la extremidad superior, mientras que la otra sección presenta en una ventana los datos recibidos por los sensores y la detección de movimiento, permitiendo elegir el tipo de clasificador utilizado para trabajar con las señales de los sensores.

### Diseño de ventanas del interfaz.



**Figura 2.13** Diagrama jerárquico de las ventanas del HMI

Se procede a realizar una clasificación jerárquica de las ventanas que se mostraran, véase en Figura 2.13, este orden jerárquico permite mantener un diseño didáctico y de fácil acceso para el usuario, considerando que la información que se proporcionará en el mismo debe ser clara y de utilidad para la aplicación del proyecto.

### Implementación de las ventanas del interfaz.

La interfaz gráfica contará con 9 ventanas interactivas las cuales será descritas a continuación.

La primera ventana mostrada en la Figura 2.14 es la ventana de inicio, esta cuenta con la presentación del proyecto y solamente tiene un botón que permite dirigirse a la siguiente ventana que es el menú.



Figura 2.14 Ventana de inicio.

La siguiente ventana es la del menú, véase Figura 2.15 aquí se encuentran tres botonerías, una que permite regresar al inicio de la aplicación, otra que permite ir a la ventana de información y la última permite dirigirse a la ventana de aplicación, esta ventana es central y permite que se navegue por las 9 ventanas con mayor facilidad.



Figura 2.15 Ventana de menú.

La ventana mostrada en la Figura 2.16 muestra la información necesaria sobre el procedimiento, en ella se encuentra 6 botoneras, tres de ellas son pequeñas y se encuentran en la parte inferior, permiten movilizarse en las otras ventanas con facilidad son los botones de regreso, menú e inicio. Mientras que hay otros botones que permite dirigirse a la ventana de ubicación de los sensores, de información del proceso y de información sobre la biomecánica.



**Figura 2.16** Ventana de información general

En la Figura 2.17 se muestra la ventana de la ubicación de los sensores, esta muestra la información de donde se deben colocar los sensores para determinar el movimiento y posición de la extremidad superior, cuenta con tres botoneras que permiten navegar entre las otras ventanas del interfaz.



**Figura 2.17** Ventana de Ubicación de los Sensores.

En la Figura 2.18, se muestra la ventana de biomecánica, aquí se cuenta con una pequeña información sobre la biomecánica y la importancia de analizarla para este proyecto y cuenta con 5 botones, tres de ellos permiten navegar entre las otras ventanas del interfaz, y dos botones que permiten obtener una información más específica sobre la anatomía y el movimiento, estos son el botón de Anatomía de la Articulación y el botón de Anatomía Muscular.



**Figura 2.18** Ventana de Información de Biomecánica.

En la Figura 2.19, se muestra la ventana de información de la anatomía de la articulación, en esta ventana se encuentran datos de las articulaciones que se utilizan en el movimiento de la extremidad superior y que tipos de movimiento permiten realizar las mismas, esto permite mejorar la interpretación de los resultados obtenidos, también cuenta con dos botones que facilitan la navegación entre ventanas.



**Figura 2.19** Ventana de Información de Anatomía de las Articulaciones.

En la Figura 2.20 se muestra la ventana de información de la anatomía muscular, en esta se muestra una tabla que indica los músculos por segmentos de la extremidad superior y el movimiento que permiten realizar en el miembro superior facilitando la interpretación de los resultados entregados por el proyecto. También cuenta con dos botones que facilitan la navegación entre ventanas del interfaz.

Segmento de la Extremidad Superior	Músculos	Movimientos
Hombro	Deltoides	Abductor del brazo, flexiona y rota
	Redondo Mayor	Aduce y extiende el hombro
	Trapecio	Giran durante la abducción del brazo
	Dorsal Ancho	Extiende, rota, abduce la extremidad superior
	Elevador de la escápula	Eleva la escápula
Brazo	Romboide menor	Rota y retrae la escápula
	Romboide mayor	Rota y retrae la escápula
	Coracobraquial	Flexión en el hombro, aducción débil.
Antebrazo	Braquial	Flexión del antebrazo en la articulación del codo.
	Bíceps braquial	Supinación del antebrazo, flexión del codo y hombro
	Flexor cubital	Flexión y abducción de la muñeca
Mano	Palmar largo	Flexión de la muñeca
	Flexor radial	Flexión y abducción de la muñeca
	Pronador redondo	Pronación del antebrazo
	Pronador cuadrado	Pronuncia el antebrazo
	Lumbricales	Flexión y extensión de las articulaciones interfalángicas
	Interóseos dorsales	Abducir los dedos
	Interóseos palmares	Abduce los dedos
	Aductor del pulgar	Aductor del pulgar

Figura 2.20 Ventana de Información de Anatomía Muscular.

En la Figura 2.21 se muestra la ventana de información del proceso, esta se encuentra ubicada en la ventana de información, aquí se presentan los diagramas de flujo que se utilizan para la programación del microprocesador y el diseño del procesamiento y preprocesamiento de la señal. También cuenta con tres botones que facilitan la navegación entre ventanas del interfaz.

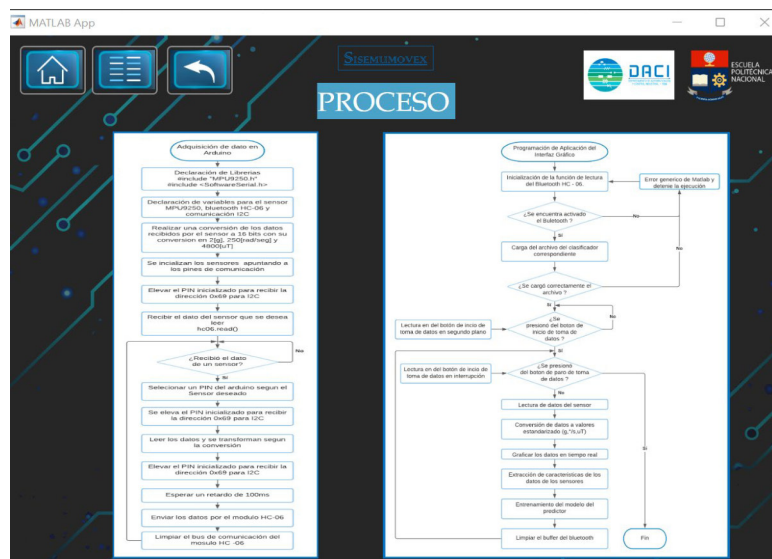


Figura 2.21 Ventana de Información del Proceso.

En la Figura 2.22 se muestra la ventana de aplicación, en esta ventana se inician la toma de datos de los sensores y entrega un resultado del movimiento y posición de la extremidad superior.

Cuenta con cuatro botones, dos de ellos permiten navegar entre varias ventanas de manera dinámica y dos botones uno para iniciar la toma de datos y otro para detener la toma de datos, también cuenta con una lista de botones donde una permite elegir el tipo de clasificador que se desea utilizar para el procesamiento de señal tomando en cuenta que solo puede elegir uno y que por defecto se elige el clasificador de RED NEURONAL, y otra lista de botones para elegir qué tipo de señal se desea observar del sensor IMU, donde solo se puede elegir una sola opción y que por defecto se muestran los datos del acelerómetro de los tres sensores.

Cuenta con tres Axes que permiten graficar las señales de los sensores y gracias a estas se pueden tomar los datos de movimiento en cada extremidad, hay tres gráficas ya que cada una representa un eje espacial (x, y, z) y en esta se grafican los datos de los tres sensores.

Y finalmente se muestra una animación con el tipo de movimiento realizado y detectado, tomando en cuenta que estas pueden ser supinación, pronación, flexión, y extensión.

















**Figura 2.22** Ventana de la Aplicación.

### **Botoneras y su Funcionalidad**

Para que el diseño de la interfaz gráfica sea amigable con el usuario se optó por utilizar una pequeña animación en las botoneras, esto permite diferenciar los botones de la información presentada en el interfaz, en la Tabla 2.10 se presentan todas botoneras implementadas en el interfaz y su funcionalidad.

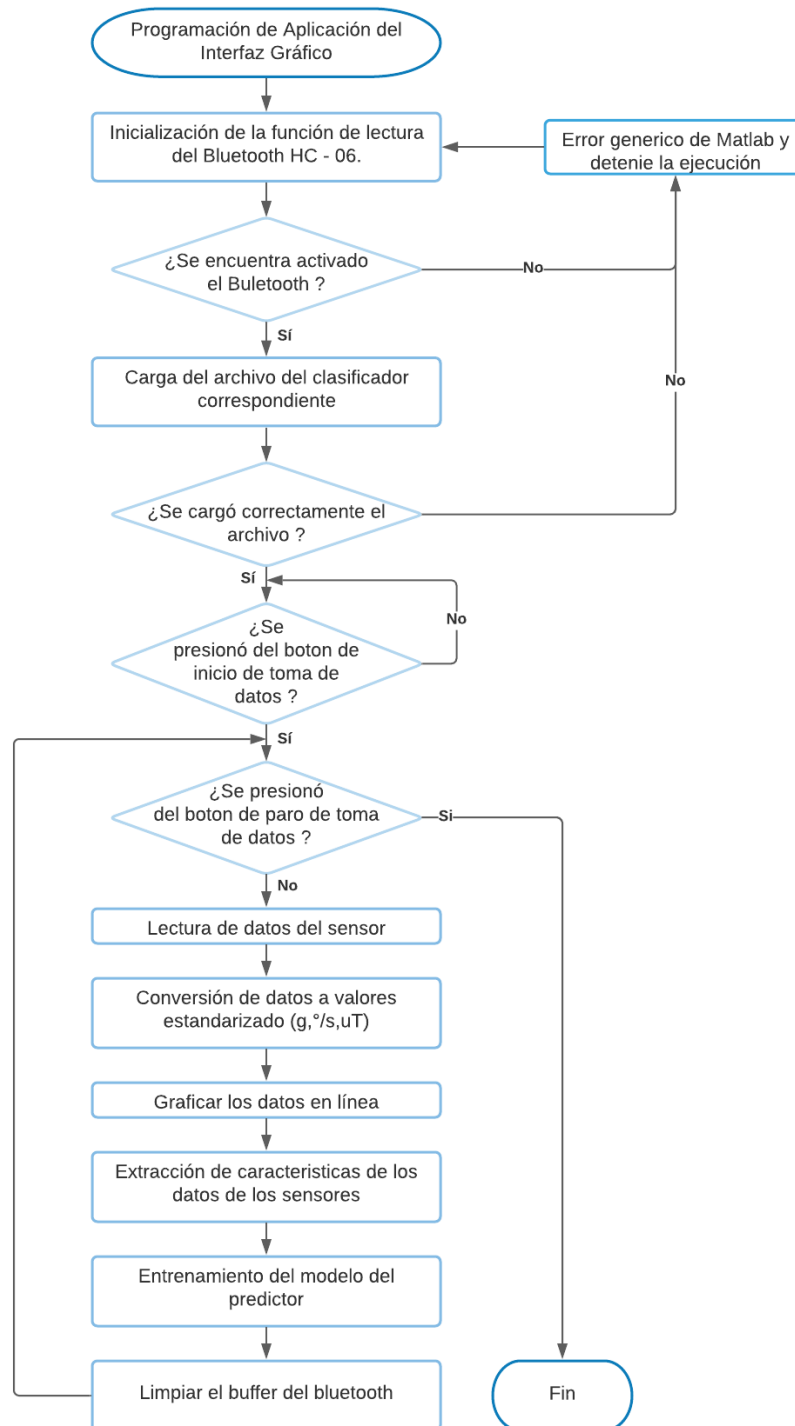


**Tabla 2.10** Botoneras y su funcionalidad en el interfaz gráfico implementado.

Botón	Función
	Se encuentra en la pantalla principal, y permite dirigirse a la ventana llamada menú donde se pueden elegir las dos secciones de la interfaz.
	Permite dirigirse a la ventana de inicio.
	Permite dirigirse a la ventana de información del proyecto.
	Permite dirigirse a la ventana donde se encuentra el resultado final del proyecto.
	Permite dirigirse a la ventana donde se encuentra la información de la ubicación de los sensores.
	Permite dirigirse a la ventana donde se presenta información sobre la biomecánica de la extremidad superior.
	Permite dirigirse a la ventana donde se presenta información del proceso del proyecto.
	Permite dirigirse a la ventana donde se presenta información sobre las articulaciones y movimientos de la extremidad superior.
	Permite dirigirse a la ventana donde se presenta información sobre los músculos y movimientos de la extremidad superior.
	Permite dirigirse a la ventana de inicio.
	Permite dirigirse a la ventana de menú.
	Permite dirigirse a la ventana anterior a la que se encuentra.
	Permite iniciar la toma de datos de los sensores.
	Permite detener la toma de datos de los sensores.

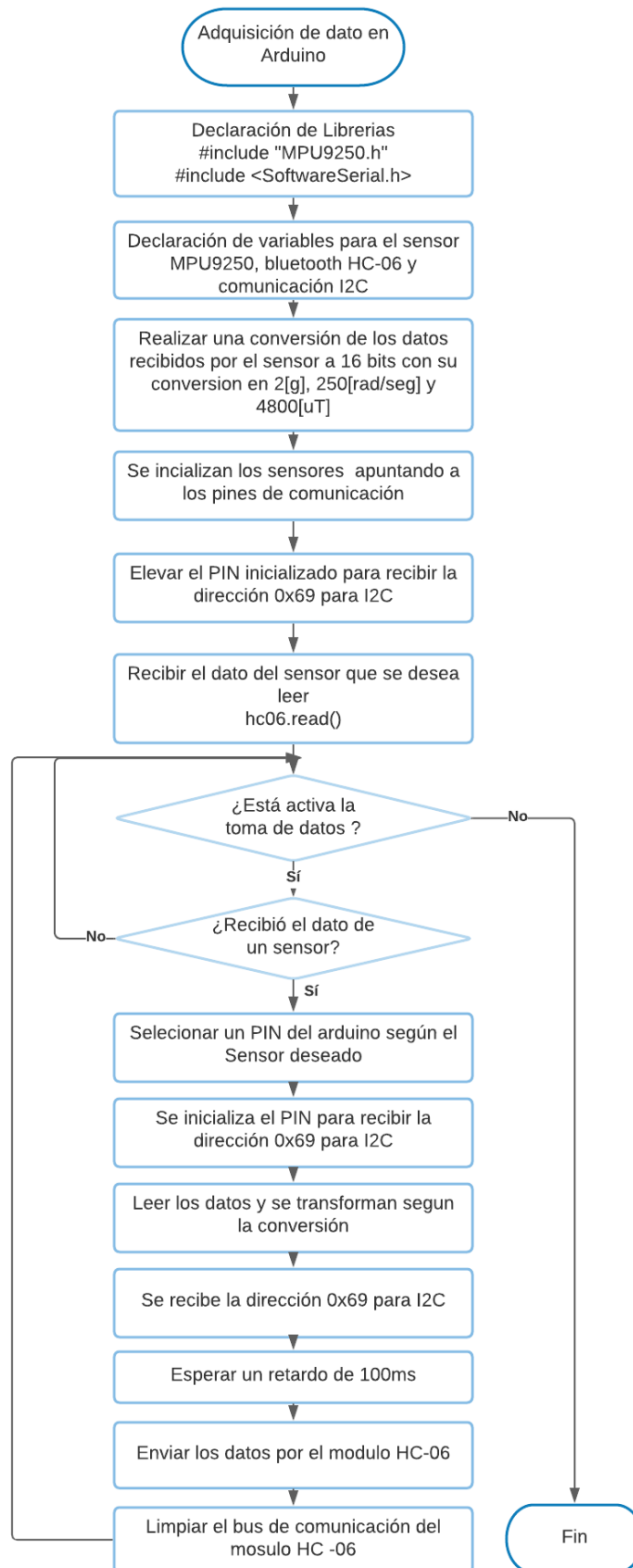
## Diagramas de Flujo de los Procesos.

Los diagramas de flujo correspondientes a la programación tanto de la interfaz de usuario como del Arduino Nano para la adquisición de las señales se encuentran en las Figuras 2.23 y 2.24 respectivamente.



**Figura 2.23** Diagrama de flujo de la programación implementada en la interfaz final





**Figura 2.24** Diagrama de flujo de la programación implementada en el Arduino Nano para la adquisición de las señales

La interfaz gráfica de la aplicación muestra los resultados de las señales emitidas por los sensores y una animación que muestra el tipo de movimiento realizado, una de las principales características de esta interfaz es que permite elegir tres clasificadores para el procesamiento de la señal, estos son redes neuronales, KNN y SVM, estos tres clasificadores se eligen antes de iniciar con la toma de datos a partir de los sensores.

La pantalla final de aplicación en la Figura 2.25 cuenta con cuatro áreas las cuales son:

- a) **Área de señales:** En esta área se encuentran las gráficas de las señales obtenidas por cualquier sensor de la IMU elegido, estos pueden ser el acelerómetro, giroscopio o magnetómetro.
- b) **Área de selección:** En esta área se encuentran dos listas de botoneras, la primera permite elegir el tipo de clasificador que se desea usar, pero este debe ser elegido antes de arrancar la toma de datos, la segunda lista de botoneras permite elegir el tipo de sensor de la IMU que se desea graficar, este se puede elegir durante la toma de datos y puede cambiar a conveniencia mientras se reciban datos de los sensores en la aplicación.
- c) **Área de ejecución:** En esta área se realiza la ejecución para la toma de datos, cuenta con dos botoneras, uno es el botón el inicio de toma de datos y el otro botón permite desactivar el bluetooth para ya no recibir los datos.
- d) **Área de resultados:** En esta área se presenta en una animación del tipo de movimiento o posición realizado, mostrando de manera textual el resultado que se obtiene del procesamiento de las señales.



**Figura 2.25** Ventana de Aplicación del Interfaz Gráfica

### 3. RESULTADOS Y DISCUSIÓN

En este apartado se expondrá el producto final del dispositivo implementado, así como las pruebas y los resultados obtenidos de los tres clasificadores utilizados dentro del ambiente de Matlab. Además, se indicará la interfaz creada con el fin de visualizar los resultados de manera clara y amigable con el usuario.

#### 3.1. RESULTADO DE LOS CLASIFICADORES

Una vez ingresada la base de datos preprocesada a la aplicación de Matlab “*Classification Learner*”, se procede a realizar un análisis de todos los resultados que la aplicación proporciona. Las respuestas obtenidas de todos los algoritmos de los tres clasificadores escogidos inicialmente, con el porcentaje de precisión de cada uno, se encuentran en la Tabla 3.1.

**Tabla 3.1** Respuesta según la eficiencia de cada algoritmo de los tres clasificadores de estudio obtenidas en la aplicación “Classification Learner” de Matlab

Clasificador	Tipo de Algoritmo	Porcentaje de precisión
Máquina de Soporte Vectorial	SVM Lineal	97.9%
	SVM cuadrático	97.8%
	SVM Gaussiano Medio	97.7%
	SVM cúbico	97.6%
	SVM Gaussiano Robusto	97.5%
	SVM Gaussiano Fino	95.5%
K-vecinos más cercanos	KNN Coseno	97.7%
	KNN Medio	97.5%
	KNN Cúbico	97.3%
	KNN Fino	97.2%
	KNN Robusto	97.2%
	KNN Ponderado	97.2%
Redes Neuronales	Redes Neuronales Amplia	97.3%
	Red Neuronal de tres capas	97.4%
	Red Neuronal Bicapa	97.3%
	Red Neuronal Media	97.2%
	Red Neuronal Estrecha	97.0%

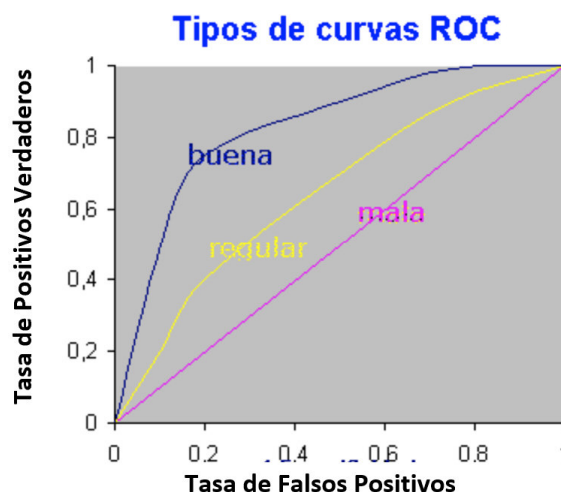
Como se evidencia en la Tabla 3.1, existen diferentes tipos de algoritmos de programación para cada clasificador donde cada uno de ellos presenta diferentes porcentajes de precisión debido a la manera de entrenar los datos basados en diferentes teoremas

matemáticos según el tipo de algoritmo utilizado. Por esta razón, se dispuso la Tabla 3.2 a manera de resumen de los clasificadores finales que se va a tomar para el análisis y aplicación final, que corresponden a los algoritmos o metodología que poseen el mayor porcentaje de precisión de cada clasificador de estudio que en el presente caso son: redes neuronales, k-vecinos más cercanos y máquina de soporte vectorial.

**Tabla 3.2** Clasificadores con mayor precisión obtenidas en la aplicación “*Classification Learner*” de Matlab

Clasificador	Tipo de Algoritmo	Porcentaje de precisión
Máquina de Soporte Vectorial	SVM Lineal	97.9%
K-vecinos más cercanos	KNN Coseno	97.7%
Redes Neuronales	Redes Neuronales Amplia	97.3%

Con este preámbulo, se realizará un análisis de cada clasificador y las características de los algoritmos que está utilizando la aplicación de Matlab, para posteriormente realizar una comparativa y determinar cuál de estos es el más eficaz para implementar en la aplicación final. Principalmente se estudiará dos respuestas inicialmente que proporciona el “*Classification Learner*” que es la matriz de confusión y la curva ROC (Receiver Operating Characteristic), ambas que se obtienen de la parte correspondiente a la validación de los datos. Posteriormente, se indicará los resultados obtenidos al momento de la implementación de estos clasificadores en la interfaz de simulación final mientras se realiza la toma de datos al mismo tiempo.



**Figura 3.1** Tipos de curvas ROC [43]

La matriz de confusión indica en qué proporción acertó o se equivocó el sistema con respecto a otra salida y la curva ROC indica la calidad o la eficiencia en la detección de falsos positivos. Esta detección de los falsos positivos se refiere a la cantidad de veces en

que el clasificador dio una respuesta o salida errónea y este lo consideró como la salida correcta. Este sistema se verifica gráficamente, donde los ejes de esta son: tasa de positivos verdaderos vs tasa de falsos positivos, por lo que la máxima eficiencia se obtendría si el punto máximo se encuentra en (0,1) y el peor de los casos se encontraría a lo largo de (0,0) a (1,1) [43]. En la Figura 3.1 se indica un ejemplo de diferentes curvas ROC y de la calidad de estas.

### 3.1.1. RESULTADOS MÁQUINA DE SOPORTE VECTORIAL - LINEAL

Una máquina de soporte vectorial con un algoritmo de programación lineal consiste en crear cierta cantidad de variables, dependiendo la cantidad de variables existentes, y reemplazarlas en estas, es decir, si una ecuación está determinada como  $f(x, y, z) = x^2 + y^2 + z^2 + x + y + z$ , se reemplazaría con  $g(p, q, r, s, t, u) = p + q + r + s + t + u$ , donde  $p \Leftrightarrow x^2$ ,  $q \Leftrightarrow y^2$ ,  $r \Leftrightarrow z^2$ ,  $s \Leftrightarrow x$ ,  $t \Leftrightarrow y$  y  $u \Leftrightarrow z$  y así crear ecuaciones lineales [44]. Para el análisis adecuado del sistema se debe verificar la Figura 3.2 correspondiente a la matriz de confusión, la Figura 3.3 correspondiente a las curvas ROC de los movimientos principales que son extensión, flexión, pronación y supinación, y la Figura 3.4 que corresponde a la curva ROC de los cambios de movimiento que son extensión-flexión, flexión-extensión, pronación-supinación y supinación-pronación.

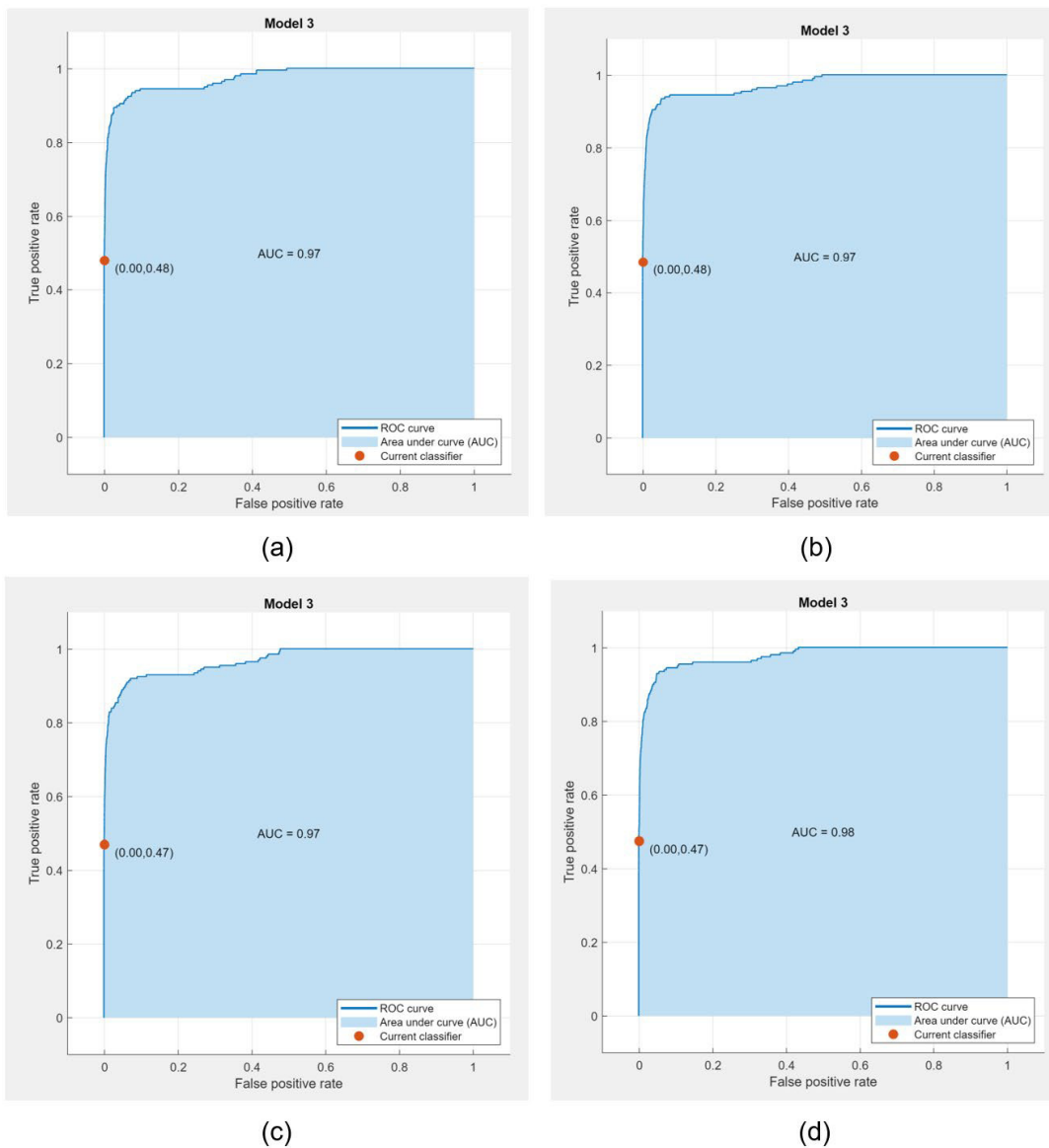
		Model 3								
		Extension	Extension-Flexion	Flexion	Flexion-Extension	Pronacion	Pronacion-Supinacion	Supinacion	Supinacion-Pronacion	
True Class	Extension	4798								
	Extension-Flexion	12	95	86	5					
	Flexion			4798						
	Flexion-Extension	86	4	12	96					
	Pronacion					4798				
	Pronacion-Supinacion					18	93	85	2	
	Supinacion							4798		
	Supinacion-Pronacion					93	3	8	94	
		Extension	Extension-Flexion	Flexion	Flexion-Extension	Pronacion	Pronacion-Supinacion	Supinacion	Supinacion-Pronacion	
		Predicted Class								

**Figura 3.2** Matriz de confusión del clasificador de máquina de soporte vectorial lineal

Se puede observar que de los movimientos principales la pronación es la que posee mayor confusión con 111 puntos que erró donde 18 fueron con el cambio los cambios entre pronación-supinación y 93 de estos fueron con el cambio entre supinación-pronación. Por



los puntos máximos tienden a ser (0,1). Con este análisis, se puede asumir en principio que no va a llegar a existir ningún inconveniente al utilizar este clasificador.



**Figura 3.4** Curva ROC de los cambios de movimiento que son: (a) extensión-flexión, (b) flexión-extensión, (c) pronación-supinación y (d) supinación-pronación, con el clasificador SVM lineal

Con las curvas ROC de la Figura 3.4, tienden a ser adecuadas, aunque el punto máximo de todas las gráficas con las que está trabajando este clasificador es el punto (0, 0.47) aproximadamente, lo que indica que el porcentaje de falsos positivos es nulo pero la eficiencia de detectar un movimiento adecuado es tendencialmente baja. Esto quiere decir que puede darse el caso que no detecte correctamente que se encuentra a lo largo del cambio y en vez de dar una respuesta, simplemente no da una respuesta como tal.

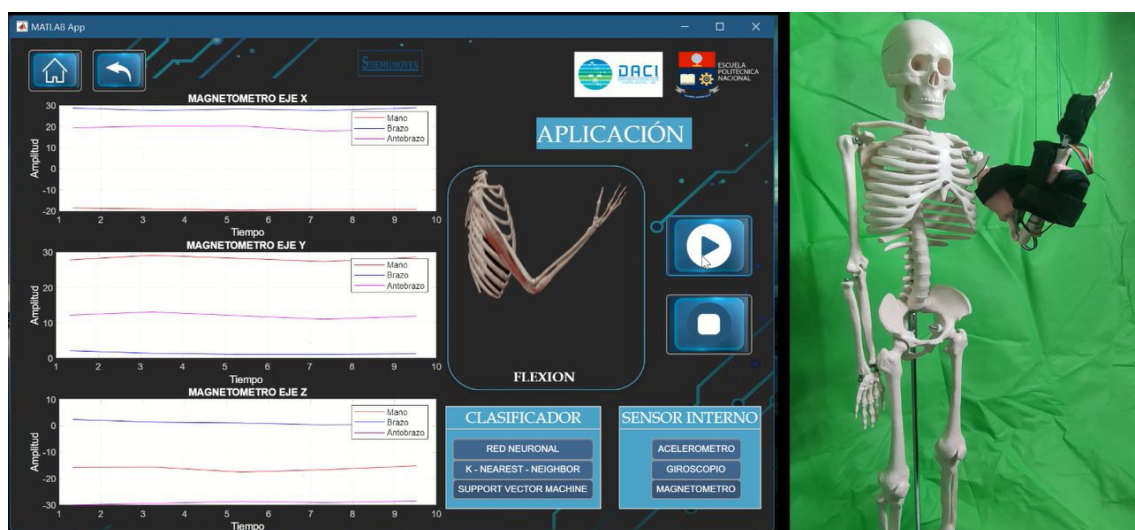
## Resultados de implementación:

En la Figura 3.5 y 3.6 se pueden observar los resultados obtenidos en la interfaz luego de que el maniquí realice dos movimientos para estabilizarse en dos posiciones específicas estas son flexión y supinación, considerando que el movimiento entre flexión y supinación se realiza en el eje Z y eje Y, los valores de las señales deben estar en el rango mostrado en la siguiente Tabla 3.3.

**Tabla 3.3** Rangos de valores de magnetómetro para detectar flexión y supinación.

		Flexión	Supinación	
Eje Y	Magnetómetro del brazo	1 a 3	-5 a -10	[ $\mu$ T]
	Magnetómetro del antebrazo	11 a 14	10 a 13	[ $\mu$ T]
	Magnetómetro de la mano	27 a 29	36 a 38	[ $\mu$ T]
Eje Z	Magnetómetro del brazo	0 a 3	-5 a -7	[ $\mu$ T]
	Magnetómetro del antebrazo	-25 a -29	-28 a -30	[ $\mu$ T]
	Magnetómetro de la mano	-30 a -28	-16 a -18	[ $\mu$ T]

Los rangos de los valores de las señales de la Tabla 3.3 son valores que se tomaron al realizar varias pruebas de movimiento, considerando que para la flexión se debe realizar un movimiento en el hombro de no más de  $30^\circ$ , para que el sensor del brazo detecte correctamente el movimiento y posición final. Mientras que para la detección de la supinación se debe asegurar el ángulo formado entre el brazo y el antebrazo deben tener un ángulo de  $90^\circ$  y la palma de la mano debe estar hacia arriba. Estas consideraciones se deben tomar en cuenta debido a que bajo estos conceptos se realizó la base de datos que se utiliza con los clasificadores. Las Figuras 3.5 y 3.6 muestran un ejemplo de detección de la posición.



**Figura 3.5** Detección de la posición de extensión utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.





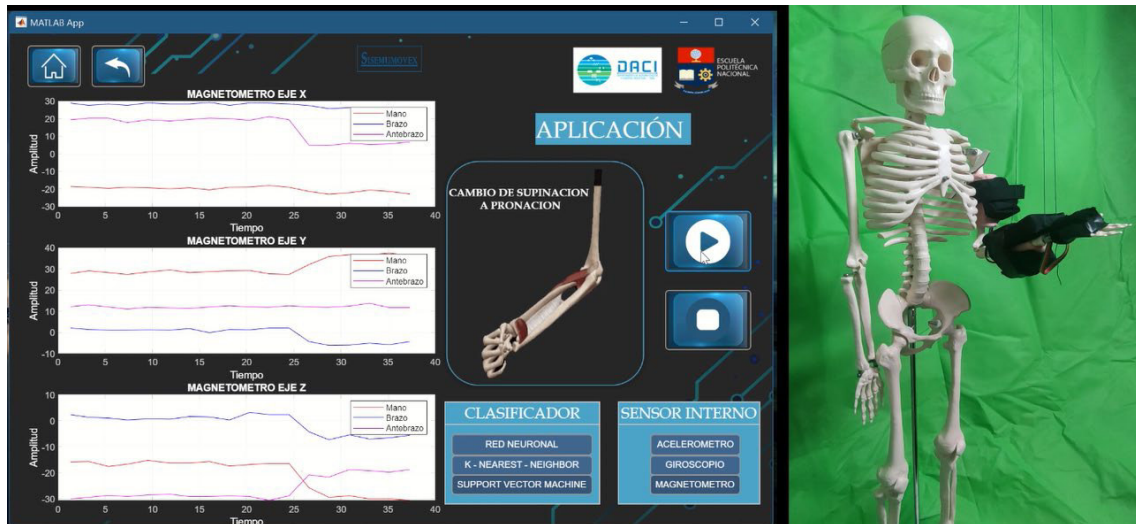
**Figura 3.6** Detección de la posición de supinación utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.

En la Figura 3.7 y 3.8 se muestran la detección de dos movimientos en específico, estos son el movimiento de cambio de supinación a pronación y el cambio de pronación a supinación, estos movimientos son detectados con facilidad en este clasificador, debido a que trabaja con los movimientos para las posiciones opuestas es decir para supinación y pronación, en este caso los movimientos más significativos que ayudan a la detección del movimiento se encuentran en el eje X, eje Y y eje Z.

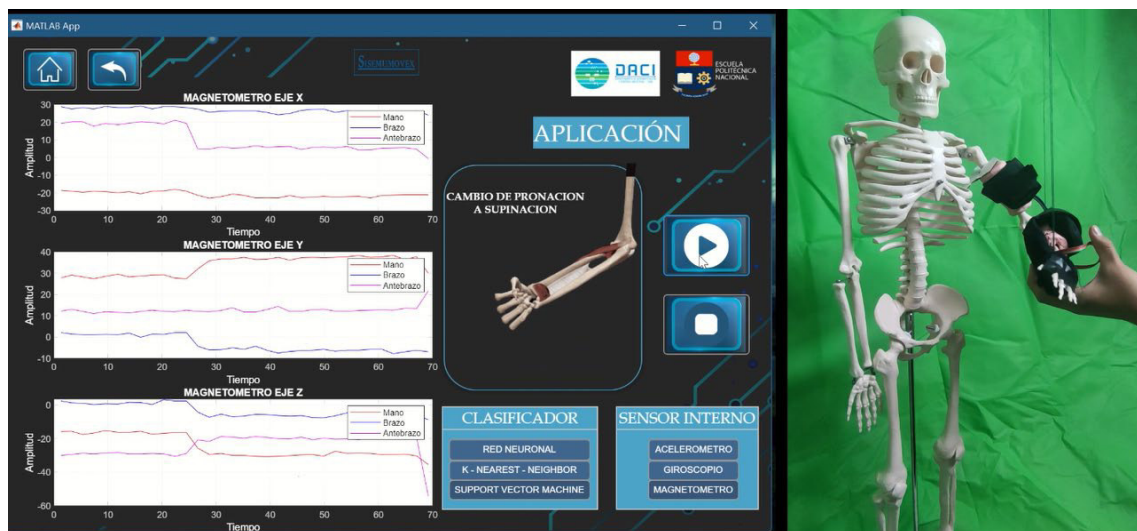
**Tabla 3.4** Rangos de valores de magnetómetro para los cambios de supinación a pronación y de pronación a supinación.

		Supinación a Pronación	Pronación a Supinación	
Eje X	Magnetómetro del brazo	30 a 3	3 a 31	[ $\mu$ T]
	Magnetómetro del antebrazo	6 a -19	5 a -20	[ $\mu$ T]
	Magnetómetro de la mano	-20 a 9	9 a -32	[ $\mu$ T]
Eje Y	Magnetómetro del brazo	-9 a 1	1 a -10	[ $\mu$ T]
	Magnetómetro del antebrazo	12 a 45	43 a 14	[ $\mu$ T]
	Magnetómetro de la mano	30 a 35	34 a 30	[ $\mu$ T]
Eje Z	Magnetómetro del brazo	-13 a 25	25 a -12	[ $\mu$ T]
	Magnetómetro del antebrazo	-19 a -50	-48 a -17	[ $\mu$ T]
	Magnetómetro de la mano	-30 a -60	-57 a -27	[ $\mu$ T]

Los rangos de los valores del movimiento de los cambios mostrados en la Tabla 3.4 se obtuvieron luego de realizar 10 pruebas donde una de estas pruebas son las que se muestran en las Figuras 3.7 y 3.8 que son tomadas del video de validación. Estos rangos permiten al clasificador determinar el tipo de movimiento realizado y este movimiento se muestra en las imágenes animadas colocadas en la interfaz.



**Figura 3.7** Detección del movimiento de cambio de supinación a pronación utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.



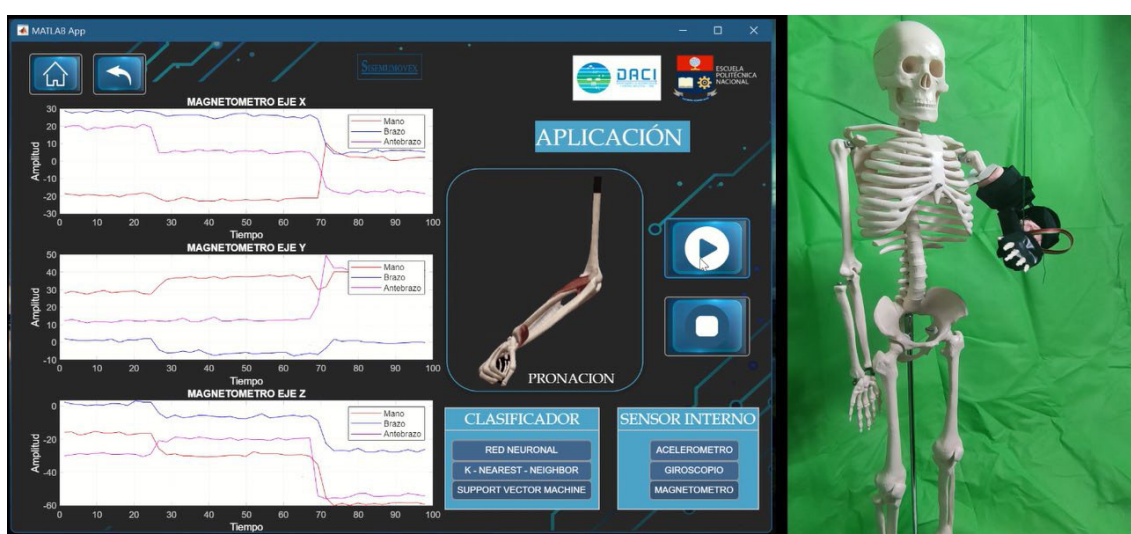
**Figura 3.8** Detección del movimiento de cambio de pronación a supinación utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.

En la Figura 3.9 y 3.10 se puede visualizar las detección de las posiciones de pronación y flexión, estos resultados mostrados en la interfaz permiten al usuario determinar la posición en la que se encuentra la extremidad superior del maniqui, mientras que en los axes que se presentan en el interfaz se muestran los valores de las señales en los que el clasificador toma como referencia para determinar el movimiento.

En la Tabla 3.5 se muestran los rangos de los valores que pueden tomar los sensores para determinar la posición considerando que estos se sacaron de 10 pruebas de movimiento donde dos de ellas se muestran en la Figura 3.9 y 3.10.

**Tabla 3.5** Rangos de valores de magnetómetro para detectar pronación y extensión

		Pronación	Extensión	
Eje X	Magnetómetro del brazo	8 a 10	25 a 27	[ $\mu$ T]
	Magnetómetro del antebrazo	10 a 3	-5 a -9	[ $\mu$ T]
	Magnetómetro de la mano	-19 a -17	-19 a -14	[ $\mu$ T]
Eje Y	Magnetómetro del brazo	3 a 0	-4 a -9	[ $\mu$ T]
	Magnetómetro del antebrazo	40 a 37	0 a -3	[ $\mu$ T]
	Magnetómetro de la mano	40 a 38	10 a 4	[ $\mu$ T]
Eje Z	Magnetómetro del brazo	-21 a -19	-9 a -12	[ $\mu$ T]
	Magnetómetro del antebrazo	-57 a -50	-23 a -25	[ $\mu$ T]
	Magnetómetro de la mano	-58 a -60	-53 a -50	[ $\mu$ T]



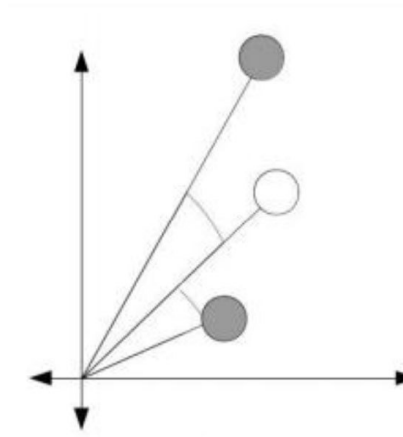
**Figura 3.9** Detección de la posición de pronación utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.



**Figura 3.10** Detección de la posición de extensión utilizando el clasificador de Máquina de Soporte Vectorial, con datos del magnetómetro.

### 3.1.2. RESULTADOS K-VECINOS MAS CERCANOS- COSENO

El algoritmo del KNN con distancia cosenoidal consiste en medir la distancia que existe entre los datos o puntos vectoriales, utilizando el coseno del ángulo, como se puede evidenciar en la Figura 3.11, y a partir de este determinar el más cercano [45].



**Figura 3.11** Distancia del coseno entre dos vectores [45]

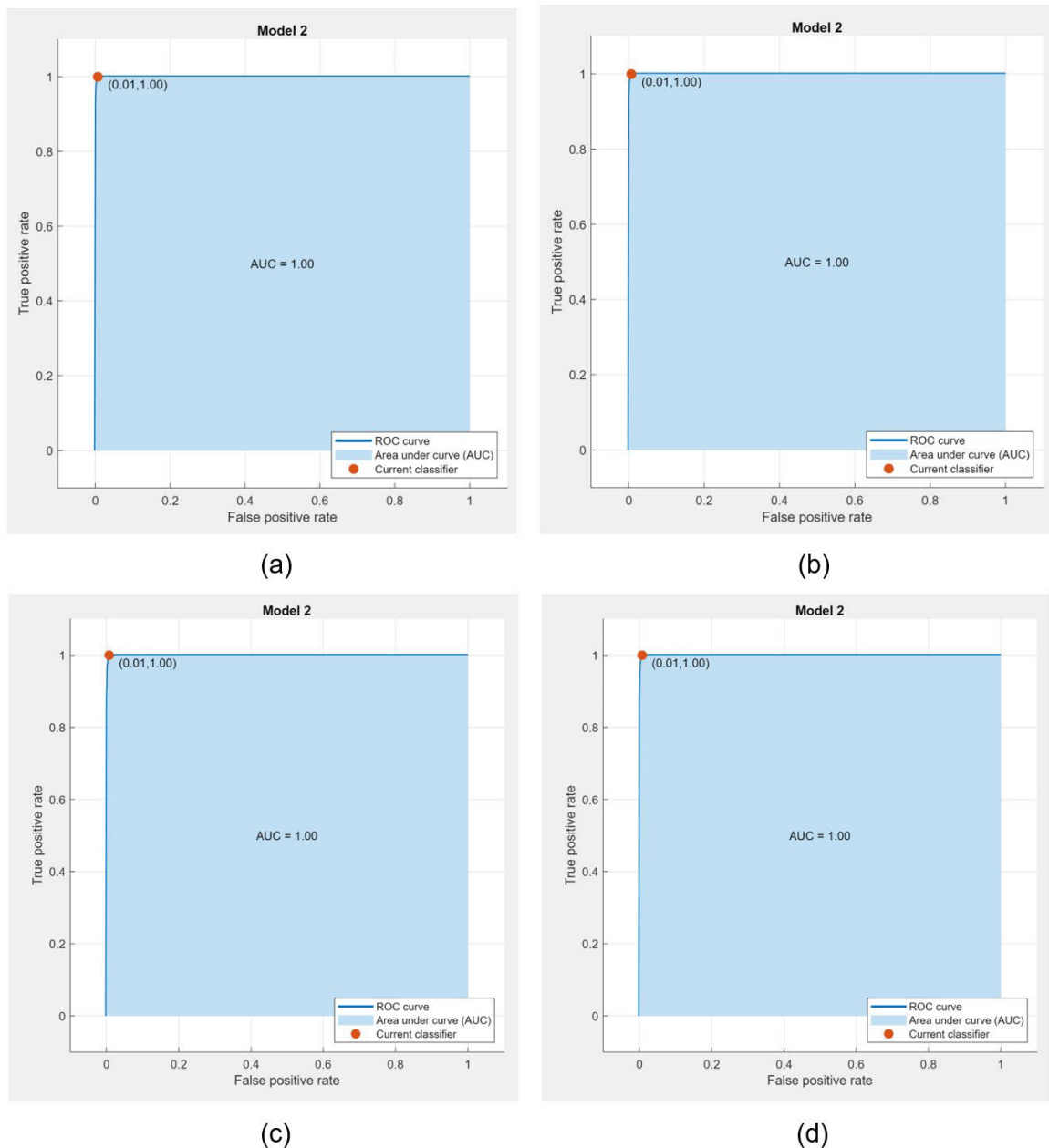
Al igual que en el punto anterior, para el análisis adecuado del sistema, se verifica la Figura 3.12 correspondiente a la matriz de confusión, la Figura 3.13 correspondiente a las curvas ROC de los movimientos principales que son extensión, flexión, pronación y supinación, y la Figura 3.14 que corresponde a la curva ROC de los cambios de movimiento que son extensión-flexión, flexión-extensión, pronación-supinación y supinación-pronación.

		Model 2							
True Class	Extension	4798							
	Extension-Flexion	13	81	85	19				
	Flexion	1	6	4787		4			
	Flexion-Extension	85	5	13	95				
	Pronacion					4794			4
	Pronacion-Supinacion					16	83	85	14
	Supinacion				1		2	4795	
	Supinacion-Pronacion					95	10	8	85
			Extension	Extension-Flexion	Flexion	Flexion-Extension	Pronacion	Pronacion-Supinacion	Supinacion
		Predicted Class							

**Figura 3.12** Matriz de confusión del clasificador de máquina de KNN cosenoidal

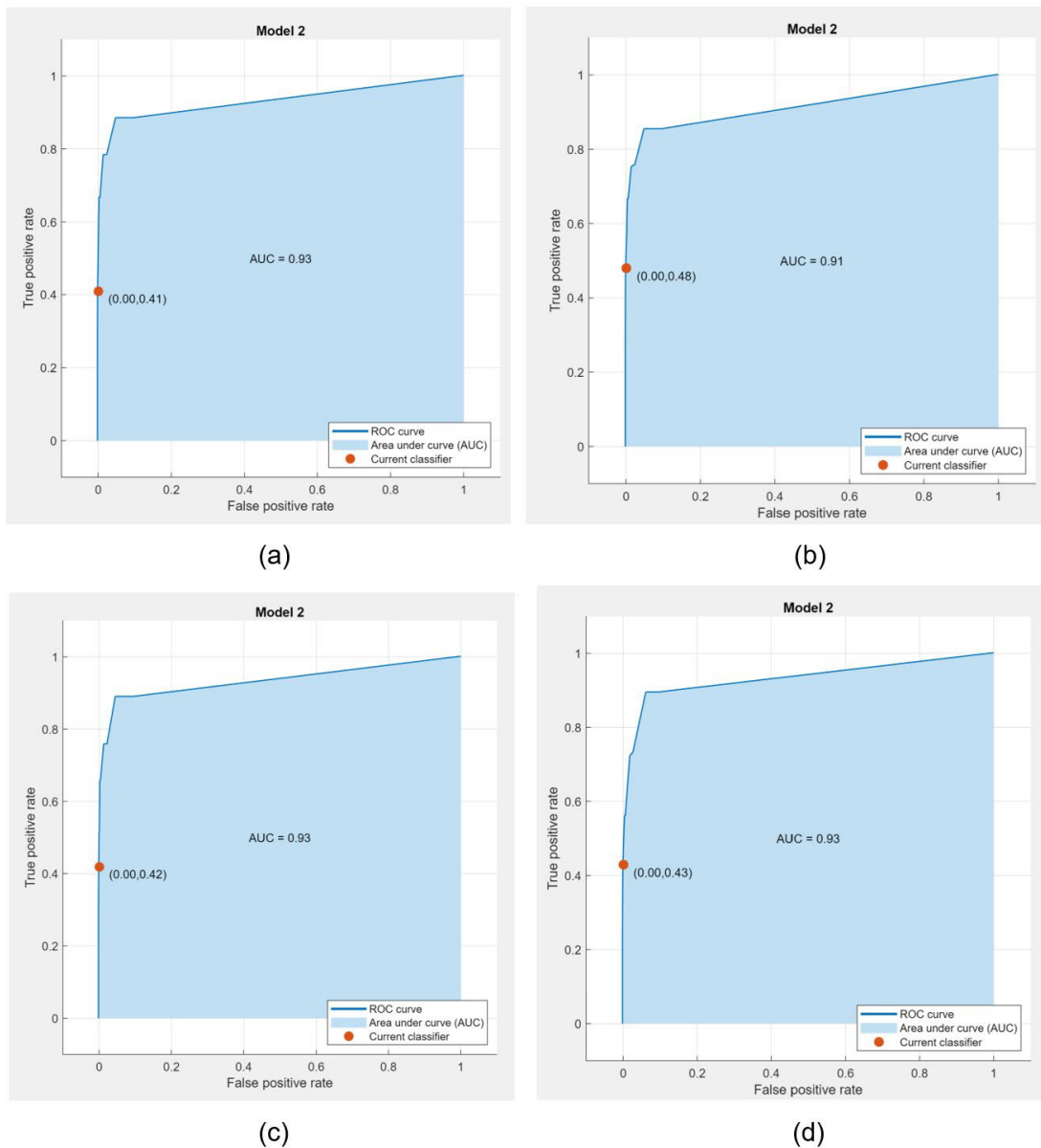


Se evidencia que por parte de los movimientos principales la pronación es la que posee mayor confusión con 115 puntos que erró donde 16 fueron con el cambio los cambios entre pronación-supinación y 95 de estos fueron con el cambio entre supinación-pronación. Por otro lado, de los cambios de movimiento, el cambio entre flexión-extensión contiene mayor cantidad de confusión con 20 datos de los cuales 19 son con extensión-flexión y 1 con supinación. Sin embargo, el porcentaje de confusión es muy adecuado ya que la mayor cantidad de datos los determina correctamente según las gráficas que nos proporciona Matlab.



**Figura 3.13** Curva ROC de movimientos principales que son: (a) extensión, (b) flexión, (c) pronación y (d) supinación, con el clasificador KNN cosenoidal

Las curvas ROC de los movimientos principales, al igual que en el caso anterior, se tiene que la eficiencia en la detección de los movimientos principales es alta, al punto de no existir ningún tipo de falencia como tal. Este análisis previo, permite tener una idea del comportamiento con respecto a los movimientos principales del clasificador.



**Figura 3.14** Figura 3.7 Curva ROC de los cambios de movimiento que son: (a) extensión-flexión, (b) flexión-extensión, (c) pronación-supinación y (d) supinación-pronación, con el clasificador KNN coseno

Al igual que en el punto anterior, se tiene que la fiabilidad del sistema con respecto a los cambios de movimiento se reduce al punto de tener casi nulos falsos positivos pero el

porcentaje de eficiencia de positivos verdaderos es bajo por lo que puede poseer el inconveniente de no detectar un cambio de movimiento como tal.

**Resultados de implementación:**

Para el clasificador K- vecinos más cercanos, se analizarán los datos del giroscopio, considerando dos variables, la primera es que al analizar las señales del sensor se encuentran señales con grandes picos, esto se debe a que las medidas tomadas por el giroscopio varían según su punto de referencia al momento de estabilizarse, por lo tanto, en la Tabla 3.6 se presentarán rangos de medida en los picos de las señales ya que en base a estos valores máximos el clasificador detecta los movimientos. La segunda variable es que al trabajar con los datos del giroscopio y detectar la posición son el clasificador K-vecinos más cercanos cosenoidal, se trabajan con puntos fijos y al trabajar con la distancia cosenoidal que consiste en medir la distancia que existe entre los datos o puntos vectoriales se comprueba la facilidad de detección de movimientos y posiciones y la razón del porque se utiliza este tipo de clasificadores para detectar el movimiento o posición.

**Tabla 3.6** Rangos de valores de magnetómetro para detectar extensión, flexión, pronación y supinación.

		Extensión, Flexión, Pronación y Supinación	
Eje X	Acelerómetro del brazo	-0.05 a -0.04	[rad/s]
	Acelerómetro del antebrazo	0 a -0.01	[rad/s]
	Acelerómetro de la mano	-0.01 a -0.02	[rad/s]
Eje Y	Acelerómetro del brazo	0.3 a 0.4	[rad/s]
	Acelerómetro del antebrazo	0 a 0.05	[rad/s]
	Acelerómetro de la mano	0 a 0.08	[rad/s]
Eje Z	Acelerómetro del brazo	0.02 a 0.025	[rad/s]
	Acelerómetro del antebrazo	0 a 0.005	[rad/s]
	Acelerómetro de la mano	0.015 a 0.02	[rad/s]

En las Figuras 3.15, 3.16, 3.17 y 3.18 se muestran una de las 10 pruebas que se realizaron para determinar un rango de valores que utiliza el clasificador según los datos del giroscopio para detectar la posición del maniquí, estos rangos se muestran en la Tabla 3.6. Estas figuras muestran el resultado de la posición detectada por el clasificador, su resultado se muestra con una imagen animada la cual cambia según los datos que el clasificador entrega, también se puede realizar el análisis de señales con el acelerómetro y magnetómetro, pero en este caso se optó el uso del giroscopio para demostrar los resultados del trabajo con valores específicos según la distancia entre los puntos de los datos.

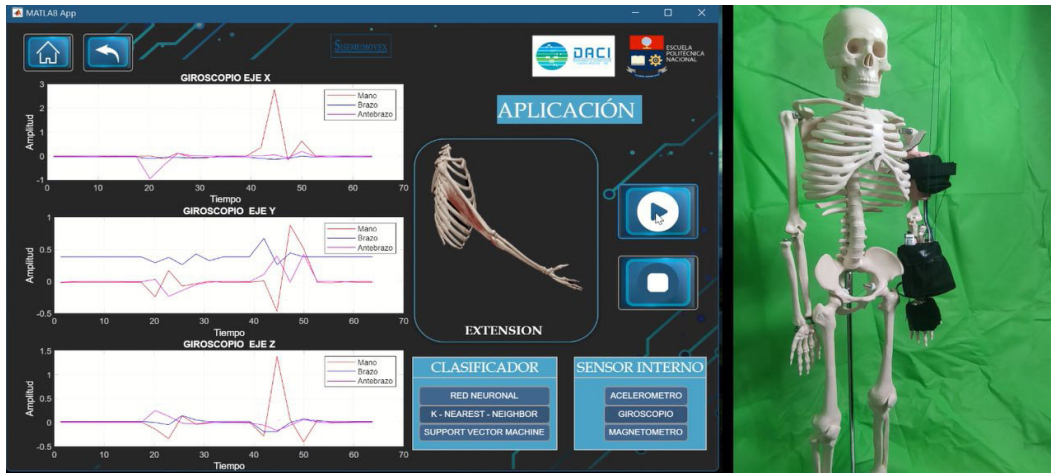


Figura 3.15 Detección de la posición de extensión utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.

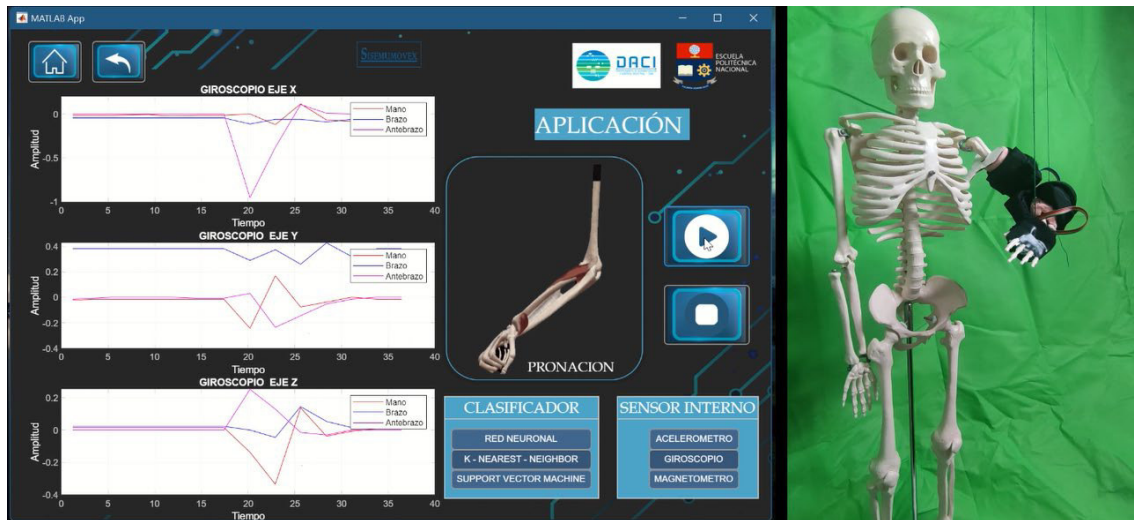


Figura 3.16 Detección de la posición de flexión utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.



Figura 3.17 Detección de la posición de supinación utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.





**Figura 3.18** Detección de la posición de pronación utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.

En las Figuras 3.19 y 3.20 se presentan una de las 10 pruebas realizadas para determinar los rangos sobre los cuales se puede detectar el cambio de flexión a extensión y de supinación a pronación, los rangos determinados en las pruebas realizadas con el maniquí se muestran en la Tabla 3.7.

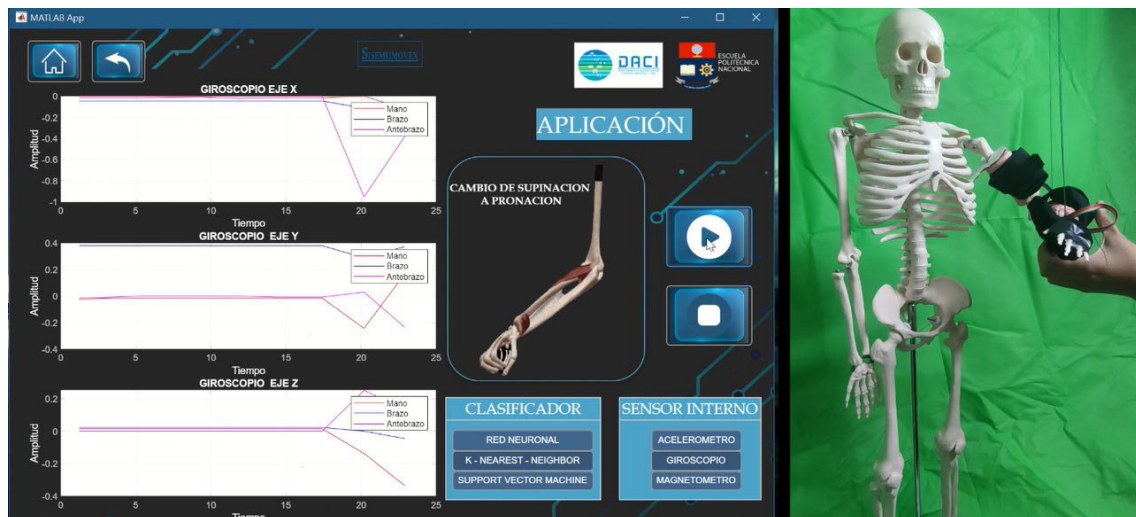
Cuando se realiza el análisis de los cambios de movimiento se trabaja con valores máximos en el giroscopio que se encuentran en los rangos de la Tabla 3.7, estos cambios son detectados con facilidad por las distancias que hay entre los datos, es decir hay una gran diferencia entre la distancia de un valor de posición a uno de movimiento, es por ello que su detección de movimiento facilita que el clasificador presente los resultados adecuados en al interfaz con la imagen animada presentada.

**Tabla 3.7** Rangos de valores de magnetómetro para los cambios de flexión a extensión y supinación a pronación.

		Flexión a Extensión	Supinación a Pronación	
Eje X	Acelerómetro del brazo	-0.01	-0.2	[rad/s]
	Acelerómetro del antebrazo	0.01	-0.95	[rad/s]
	Acelerómetro de la mano	3	-0.3	[rad/s]
Eje Y	Acelerómetro del brazo	0.8	0.2	[rad/s]
	Acelerómetro del antebrazo	0.4	0.05	[rad/s]
	Acelerómetro de la mano	-0.5	-0.25	[rad/s]
Eje Z	Acelerómetro del brazo	-0.4	-0.05	[rad/s]
	Acelerómetro del antebrazo	-0.3	0.25	[rad/s]
	Acelerómetro de la mano	1.4	-0.4	[rad/s]



**Figura 3.19** Detección del movimiento de cambio de flexión a extensión utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.



**Figura 3.20** Detección del movimiento de cambio de supinación a pronación utilizando el clasificador de K-vecinos más cercanos - coseno, con datos del giroscopio.

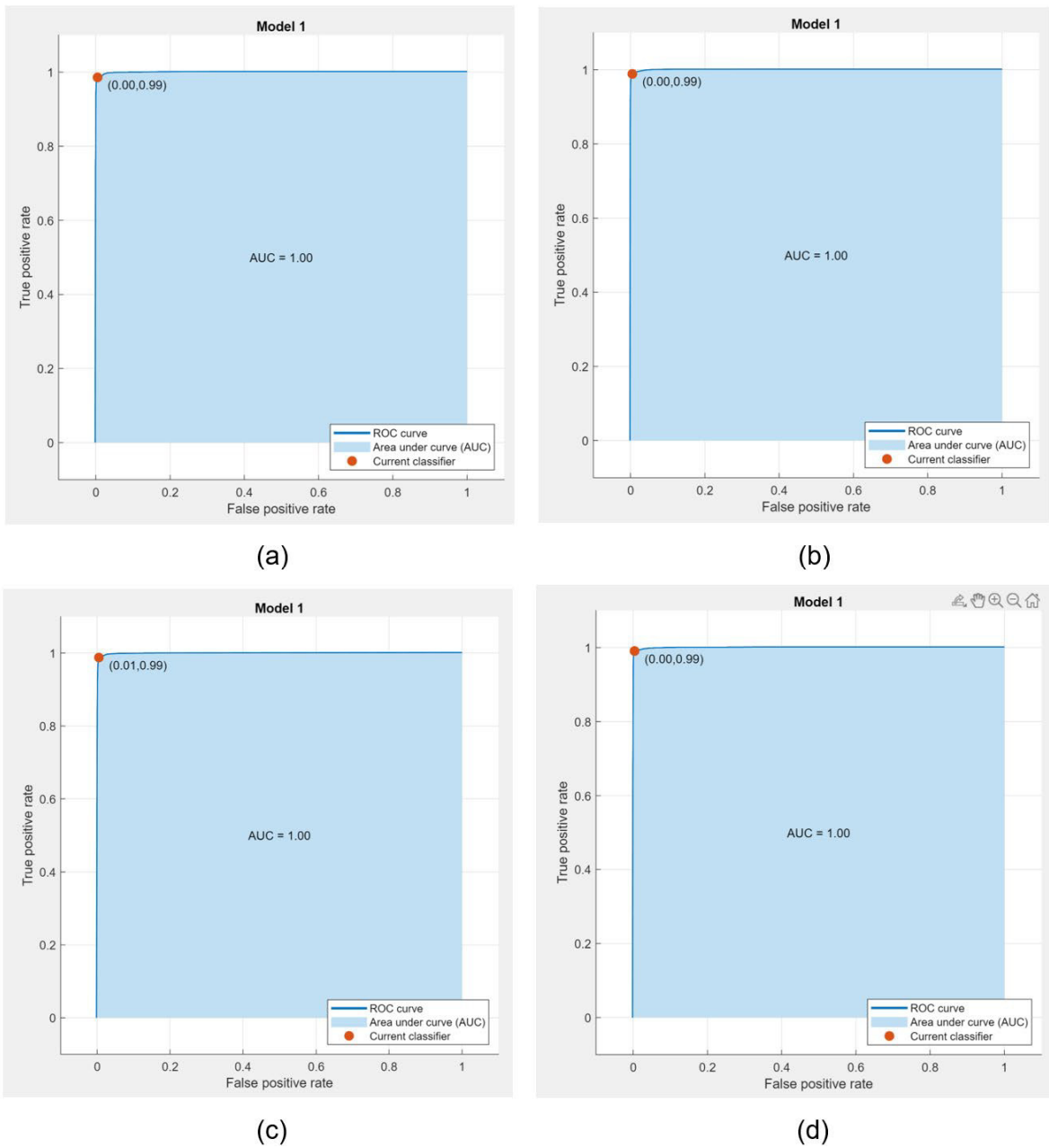
### 3.1.3. RESULTADOS REDES NEURONALES - AMPLIA

Una red neuronal con un algoritmo de programación amplia consiste en crear una capa oculta, que es aquella que permite la división de datos según las características que posee cada uno de estos, de hasta 100 divisiones, por lo que trata de crear la mayor cantidad de divisiones de datos de entrada para mejorar la capacidad de detección de la salida. Para el análisis adecuado del sistema se debe verificar la Figura 3.22 correspondiente a la matriz de confusión, la Figura 3.23 correspondiente a las curvas ROC de los movimientos principales que son extensión, flexión, pronación y supinación, y la Figura 3.24 que corresponde las curvas ROC de los cambios de movimiento que son extensión-flexión, flexión-extensión, pronación-supinación y supinación-pronación.

		Model 1							
True Class	Extension	4730	11		55	2			
	Extension-Flexion	13	119	60	6				
	Flexion		52	4744	2				
	Flexion-Extension	59	6	11	120		2		
	Pronacion					4739	6		53
	Pronacion-Supinacion					16	127	48	7
	Supinacion						44	4751	3
	Supinacion-Pronacion					64	2	11	121
		Extension	Extension-Flexion	Flexion	Flexion-Extension	Pronacion	Pronacion-Supinacion	Supinacion	Supinacion-Pronacion
		Predicted Class							

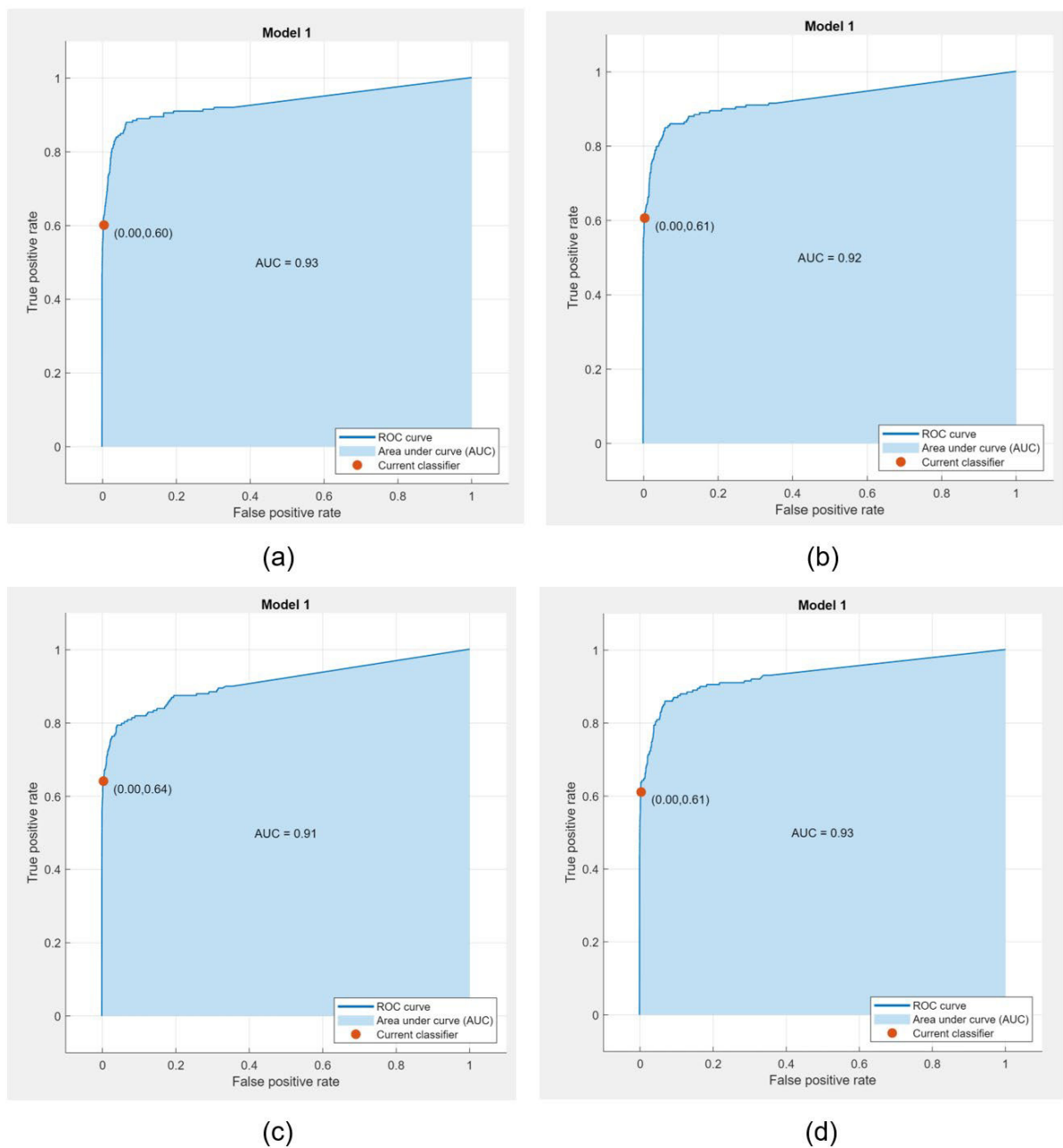
**Figura 3.21** Matriz de confusión del clasificador de redes neuronales amplia

Se puede observar que por parte de los movimientos principales la pronación es la que posee mayor confusión con 80 puntos que erró donde 16 fueron con el cambio los cambios entre pronación-supinación y 64 de estos fueron con el cambio entre supinación-pronación. Por otro lado, de los cambios de movimiento, el cambio entre extensión-flexión contiene mayor cantidad de confusión con 69 datos, donde 11 son con la extensión, 52 con la flexión y 6 con flexión y extensión. Algo a considerar en la respuesta de esta matriz de confusión es que este clasificador es el que menor cantidad de errores posee en comparación que los dos anteriores. Sin embargo, al momento de analizar lo cambios de movimiento posee un aumento de las fallas. De manera general se puede determinar que el porcentaje de confusión es muy adecuado ya que la mayor cantidad de datos los acierta de manera correcta.



**Figura 3.22** Curva ROC de movimientos principales que son: (a) extensión, (b) flexión, (c) pronación y (d) supinación, con el clasificador Red Neuronal amplia

Al igual que en los dos casos de los clasificadores anteriores, los falsos positivos son casi nulos con lo que respecta a los movimientos principales por lo que este clasificador también posee una eficiencia alta, con lo que respecta estos movimientos.



**Figura 3.23** Figura 3.7 Curva ROC de los cambios de movimiento que son: (a) extensión-flexión, (b) flexión-extensión, (c) pronación-supinación y (d) supinación-pronación, con el clasificador Red Neuronal amplia

Al igual que en el punto anterior, se tiene que la fiabilidad del sistema con respecto a los cambios de movimiento se reduce al punto (0,0.63) en promedio de los tres cambios de movimientos. Un punto en tomar a consideración es que a pesar de que el área bajo la curva es menor a la que da como resultado en los otros dos clasificadores, el punto más alto de trabajo es mayor en un 20%. Sin embargo, el porcentaje de eficiencia de positivos

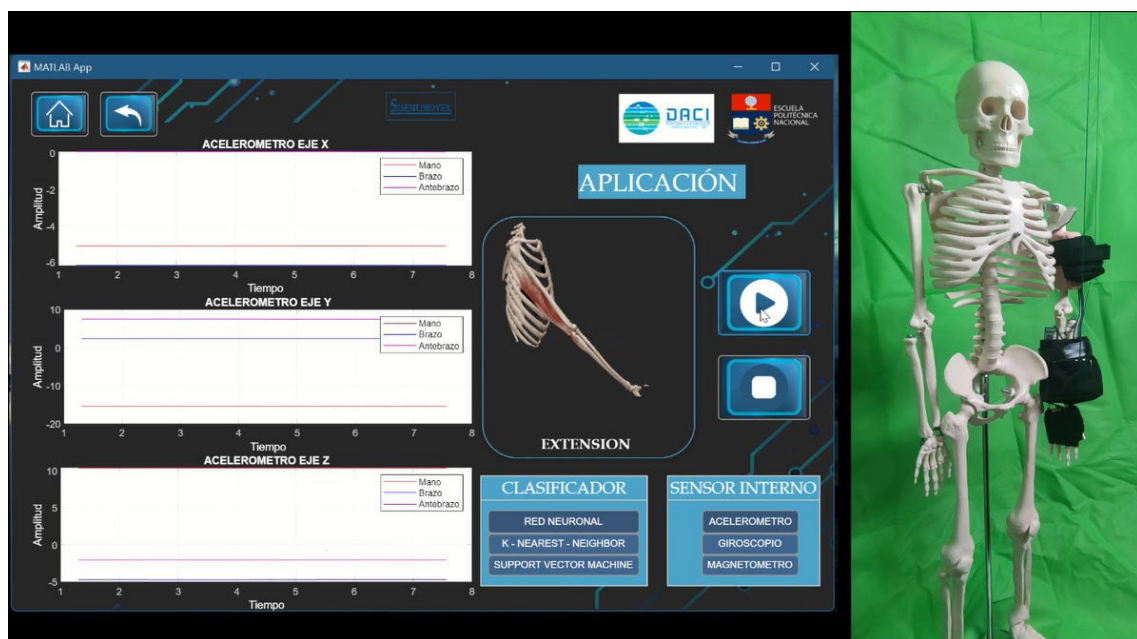
verdaderos es relativamente bajo, por lo que puede existir inconvenientes de detección de cambios de movimiento.

### Resultados de implementación:

En la Figura 3.25 y 3.26 se presentan los resultados de la posición de extensión y flexión, estos resultados son una prueba de diez que se realizaron considerando que estas pruebas permiten determinar un rango de valores sobre un movimiento específico, ver Tabla 3.8, en este caso se analizarán los datos del acelerómetro ya que estos datos permiten trabajar con directamente con las características presentadas y a su vez ayudan a determinar la eficiencia del clasificador según los resultados que se muestran con una imagen animada en la aplicación.

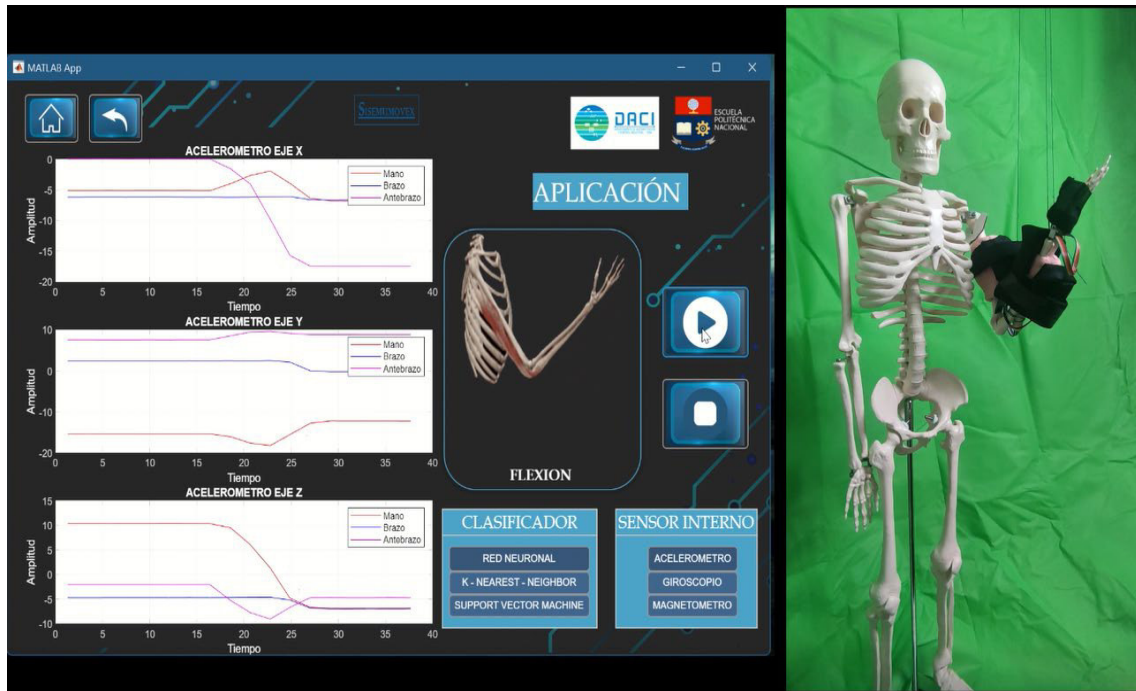
**Tabla 3.8** Rangos de valores de magnetómetro para detectar extensión, flexión.

		Extensión	Flexión	
Eje X	Acelerómetro del brazo	-6 a -7	-6 a -7	[g]
	Acelerómetro del antebrazo	1 a -1	-16 a -18	[g]
	Acelerómetro de la mano	-5 a -4	-6 a -7	[g]
Eje Y	Acelerómetro del brazo	0 a 3	-3 a -1	[g]
	Acelerómetro del antebrazo	7 a 8	9 a 10	[g]
	Acelerómetro de la mano	-15 a -18	-10 a -12	[g]
Eje Z	Acelerómetro del brazo	-4 a -5	-6 a -7	[g]
	Acelerómetro del antebrazo	-2 a -1	-4 a -5	[g]
	Acelerómetro de la mano	8 a 11	-6 a -7	[g]



**Figura 3.24** Detección de la posición de extensión el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.



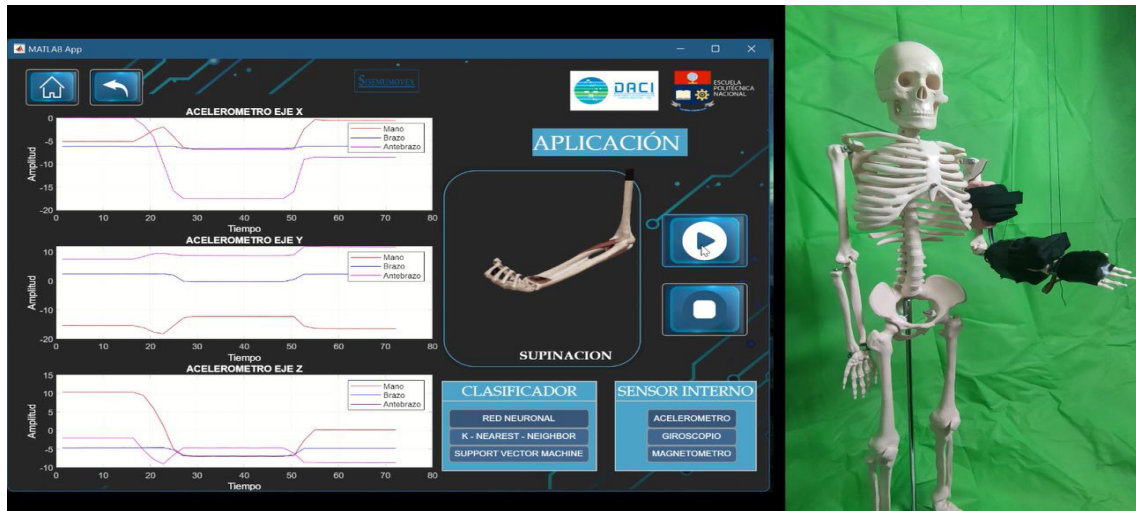


**Figura 3.25** Detección de la posición de flexión el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.

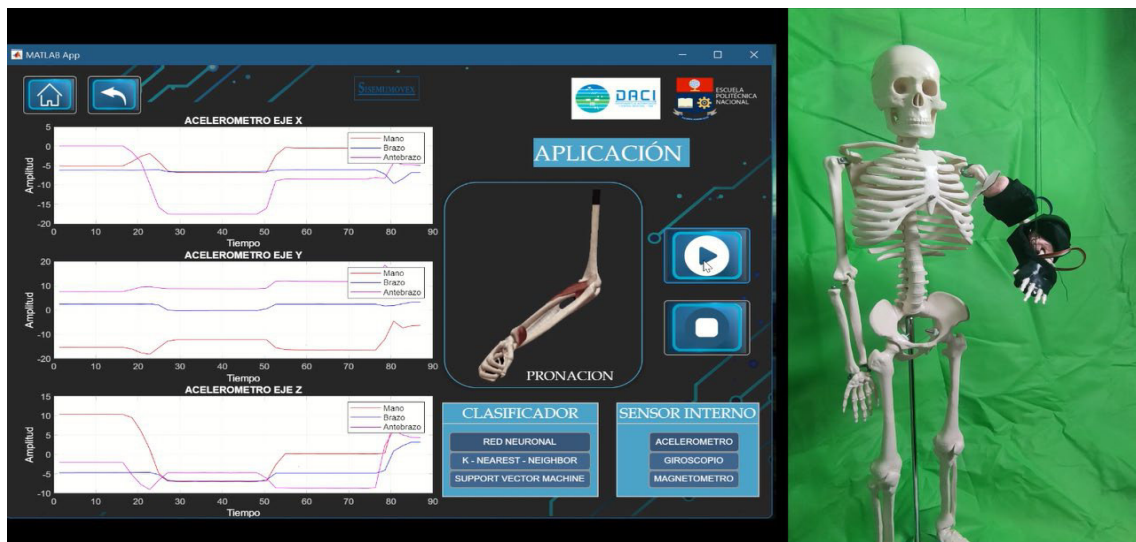
En la Figura 3.27 y 3.28 se muestran los resultados de implementación del sistema donde se puede observar que el maniquí realiza una posición de supinación y pronación y en el interfaz se muestra dicha posición con una imagen animada, además que se pueden notar los movimientos realizados en las gráficas correspondientes a cada eje, los valores de dichas graficas representan los valores tomados por los sensores, es por ello que se realizaron pruebas para determinar un rango de valores para el movimiento como se muestra en la Tabla 3.9.

**Tabla 3.9** Rangos de valores de magnetómetro para detectar supinación y pronación.

		Supinación	Pronación	
Eje X	Acelerómetro del brazo	-6 a -7	-6 a -7	[g]
	Acelerómetro del antebrazo	-8 a -10	-5 a -6	[g]
	Acelerómetro de la mano	0 a -1	0 a 1	[g]
Eje Y	Acelerómetro del brazo	0 a 3	1 a 4	[g]
	Acelerómetro del antebrazo	8 a 10	15 a 18	[g]
	Acelerómetro de la mano	-15 a -16	-5 a -6	[g]
Eje Z	Acelerómetro del brazo	-4 a -5	0 a -1	[g]
	Acelerómetro del antebrazo	-9 a -10	4 a 5	[g]
	Acelerómetro de la mano	-1 a -2	4 a 5	[g]



**Figura 3.26** Detección de la posición de supinación el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.



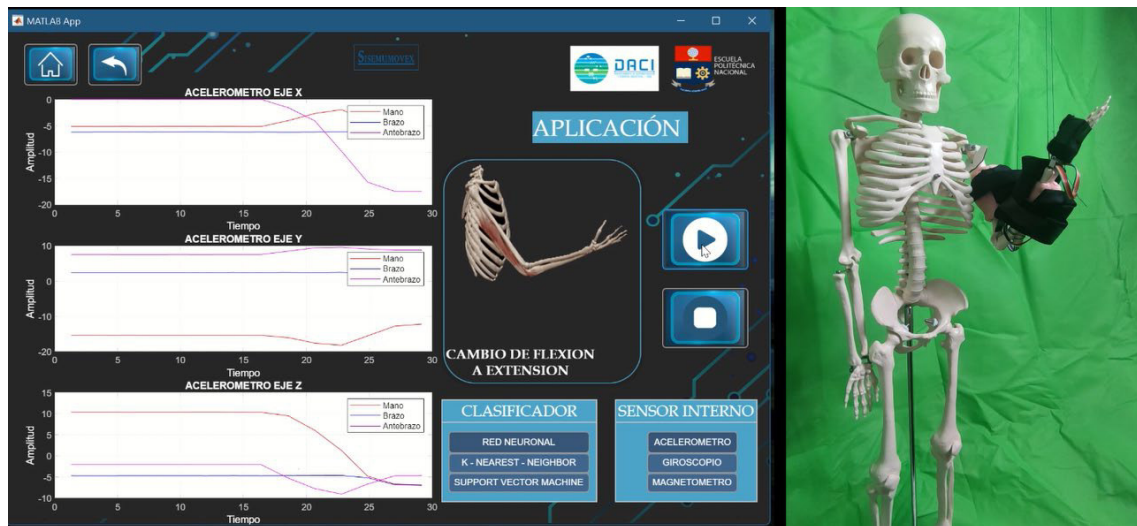
**Figura 3.27** Detección de la posición de pronación el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.

En la Figura 3.29 y 3.30 se muestra el resultado de los movimientos que se presenta durante el cambio de posición en este caso se muestran los cambios de flexión a extensión y de supinación a pronación, en estos cambios se debe realizar una acotación, la cual es que el al realizar un cambio de flexión a extensión o viceversa se puede dar el caso que en la animación nos muestre cambios de supinación o la posición de supinación como tal, esto se debe a que la pronación y supinación se encuentran en la mitad del rango de movimiento de la extensión y flexión por lo que se debe esperar un tiempo de estabilización del sensor para detectar el movimiento y posición adecuada. En la Tabla 3.9 se muestran los rangos de los valores del acelerómetro para detectar algunos movimientos.



**Tabla 3.10** Rangos de valores de magnetómetro para los cambios de flexión a extensión y supinación a pronación.

		Flexión a Extensión	Supinación a Pronación	
Eje X	Acelerómetro del brazo	-5 a -4	-6 a -4	[g]
	Acelerómetro del antebrazo	0 a -18	-18 a -10	[g]
	Acelerómetro de la mano	-5 a -2	-4 a 0	[g]
Eje Y	Acelerómetro del brazo	0 a 2	0 a 2	[g]
	Acelerómetro del antebrazo	8 a 10	10 a 12	[g]
	Acelerómetro de la mano	-18 a -15	-12 a -18	[g]
Eje Z	Acelerómetro del brazo	-5 a -8	-7 a -5	[g]
	Acelerómetro del antebrazo	-18 a -5	-8 a -1	[g]
	Acelerómetro de la mano	10 a -8	-8 a 0	[g]



**Figura 3.28** Detección del movimiento de cambio de flexión a extensión utilizando el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.



**Figura 3.29** Detección del movimiento de cambio de supinación a pronación utilizando el clasificador de Redes Neuronales Amplia, con datos del acelerómetro.

### 3.1.4. COMPARACIÓN DE CLASIFICADORES

Se puede observar que existen diferentes características correspondientes a los resultados de la validación arrojados por la aplicación de Matlab, de donde se puede verificar la efectividad de cada uno de los sistemas. Para poder realizar una comparación precisa de estos sistemas, se realizó una prueba de cada clasificador tomando las muestras tomadas en ese mismo momento, donde se pudo evidenciar diferentes comportamientos que se expondrán de manera resumida en la Tabla de comparación 3.3.

**Tabla 3.11** Tabla comparativa de los resultados obtenidos en los diferentes clasificadores

Tipo de Análisis	Movimiento	SVM Lineal	KNN coseno	Red Neuronal Amplia
Matriz de confusión	Movimientos principales	La máxima confusión ocurrió con la pronación con un total de 111 erróneos.	La máxima confusión ocurrió con la pronación con un total de 115 erróneos.	La máxima confusión ocurrió con la pronación con un total de 80 erróneos.
	Cambios entre movimientos	La máxima confusión ocurrió con la extensión-flexión con un total de 5 erróneos.	La máxima confusión ocurrió con la flexión - extensión con un total de 20 erróneos.	La máxima confusión ocurrió con la extensión-flexión con un total de 69 erróneos.
Curva ROC	Movimientos principales	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos son casi del 100%.	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos son casi del 100%.	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos son casi del 100%.
	Cambios entre movimientos	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos indican una eficiencia promedio de 47.5%.	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos indican una eficiencia promedio de 43.5%.	Las curvas ROC indican que los falsos positivos tienden a ser nulos y los positivos verdaderos indican una eficiencia promedio de 61.5%.
Prueba en la aplicación	Todos	Se presentaron los siguientes casos del clasificador al momento del uso con datos que se iban obteniendo en la interfaz: <ul style="list-style-type: none"> <li>La señal de pronación se confundía con la de supinación en varios puntos.</li> <li>La señal de flexión tendía a confundirse con el cambio entre</li> </ul>	Se presentaron los siguientes casos del clasificador al momento del uso con datos que se iban obteniendo en la interfaz: <ul style="list-style-type: none"> <li>La señal de flexión se confunde con la de extensión en varios puntos.</li> <li>Detecta de manera adecuada la supinación y pronación.</li> </ul>	Se presentaron los siguientes casos del clasificador al momento del uso con datos que se iban obteniendo en la interfaz: <ul style="list-style-type: none"> <li>Todo a lo que corresponde a los cambios de movimiento, tiende a no dar una respuesta y da prioridad a los movimientos principales.</li> </ul>

		flexión- extensión. <ul style="list-style-type: none"> <li>• La extensión, solía confundirse con la flexión.</li> <li>• La supinación acertaba la mayor parte del tiempo.</li> <li>• Detectaba de manera acertada los cambios entre supinación y pronación, pero no acierta en los cambios entre flexión y extensión.</li> </ul>	<ul style="list-style-type: none"> <li>• Detecta en mayor proporción de los cambios entre los movimientos de flexión y extensión, que de los de supinación y pronación.</li> <li>• Existe confusión entre la extensión y el cambio entre flexión-extensión.</li> </ul>	<ul style="list-style-type: none"> <li>• Existe confusión en ciertos puntos entre flexión y supinación.</li> <li>• El brazo debe estar siempre completamente estirado para que el sistema lo detecte de manera acertada.</li> <li>• Detecta, en su mayoría, de manera adecuado los movimientos principales.</li> </ul>
--	--	---	--	--

Con la ayuda de esta tabla comparativa se tiene como resultado final que los tres clasificadores poseen mayor porcentaje de eficiencia para ciertos movimientos o cambios de estos. Esto puede deberse al tipo de algoritmo que está utilizando cada clasificador, ya que el utilizar datos que se van obteniendo a medida que se va realizando el movimiento por parte del maniquí, influye directamente en el resultado final debido a la cantidad de datos que se está obteniendo, así como la condición en que se encuentran estos datos. Con este preámbulo, analizando las condiciones y las diferentes respuestas de cada clasificador, se concluye que el clasificador que mejor se adapta para la aplicación del presente proyecto es la red neuronal, ya que presenta una alta calidad al momento de la detección de los datos principales, así como, a pesar de poseer falsos positivos en los cambios de movimiento, se da prioridad a los movimientos principales, lo cual permite tener la información más importante que en este caso es determinar la posición final de la extremidad superior emulada por el maniquí.

## **4. CONCLUSIONES Y RECOMENDACIONES**

A continuación, se presentarán las conclusiones y recomendaciones finales del presente proyecto que van orientadas a determinar la funcionalidad y aplicabilidad de este, dentro de la visión de la comunidad, determinando los objetivos cumplidos.

### **4.1. CONCLUSIONES**

- Se investigó diferentes conceptos relacionados a la fisiología de la extremidad superior, permitiendo determinar los movimientos principales de estudio, así como, el posicionamiento adecuado de los sensores a lo largo de la misma. Con esta investigación, se concluye que la ubicación apropiada para cada uno de los sensores es en cada segmento de la extremidad superior que corresponde al brazo, antebrazo y mano, ya que, al tener los sensores ubicados en tres distintos puntos, estos permiten obtener valores específicos de cada sensor facilitando la diferenciación entre ellos.
- Se determinaron los principales movimientos y posiciones de la extremidad superior, considerando que la pronación, supinación, extensión y flexión permiten determinar la biomecánica general de esta, teniendo en cuenta que trabajan los tres segmentos de la extremidad a la vez. Bajo este preámbulo, se concluye que los sensores de unidad de medición inercial son ideales para trabajar en este tipo de proyectos ya que, cuentan con un acelerómetro, giroscopio y magnetómetro, los cuales al realizar un tratamiento de sus datos permiten determinar movimiento y posición.
- Partiendo del análisis de las señales de los sensores de unidad de medición inercial, se investigó las características necesarias a considerar en un sistema microprocesado para la adquisición y envío de datos, las cuales deben permitir la comunicación IIC, comunicación serial asincrónica y que cuente con puertos de entrada y salida. Se concluye que el microcontrolador Arduino Nano cumple con los requerimientos mínimos para la implementación en el proyecto, ya que, como ventajas adicionales presenta la facilidad de adquisición comercial dentro del país, bajo costo y robustez en el diseño de su placa.
- Se implementaron diferentes métodos de preprocesamiento y procesamiento de las señales, partiendo desde la comunicación entre los sensores y el microprocesador, hasta la obtención de los datos en el computador, concluyendo, que los métodos de filtrado digital, eliminación de datos anómalos y extracción de características de

las señales permiten obtener una base de datos robusta alcanzando resultados de mayor precisión al momento de utilizarla en el entrenamiento y validación en los diferentes clasificadores.

- Se analizó e investigó un conjunto de clasificadores de señales, que presentan mayor eficiencia y mejores resultados al trabajar con señales obtenidas a partir de acelerómetros, giroscopios y magnetómetros. Por esta razón, se concluye que los clasificadores que mejor se adaptan a este tipo de datos son Redes Neuronales, Máquina de Vectores de Estado (SVM) y K – Vecinos más cercanos (KNN), los cuales facilitan la detección del tipo de movimiento y posición final de la extremidad superior y presentan una alta precisión en la determinación de resultados.
- Una vez obtenido los datos de los clasificadores se tiene una gran cantidad de información que requiere ser presentada de manera estructurada para una mejor interpretación de resultados. Por lo tanto, se concluye que es importante crear una interfaz gráfica dinámica con el usuario, que permita obtener información del hardware y software implementado, presentar las señales obtenidas por los sensores y permitir elegir el tipo de clasificador con los resultados respectivos.
- Finalmente, se concluye que el monitoreo del movimiento y posición de la extremidad superior es importante, debido a que permite identificar el estado de la estructura biomecánica del miembro superior, analizando si este cumple con los parámetros estándares a nivel anatómico para determinar si existe un funcionamiento adecuado de la misma.

## **4.2. RECOMENDACIONES**

- Se recomienda que, para obtener resultados más precisos y eficientes al momento de utilizar clasificadores, sin importar cuáles sean estos, se trabaje con una base de datos lo más amplia posible, que permita una partición adecuada de estos, para lo que conlleva entrenamiento y validación. Ya que, a mayor cantidad de datos, los algoritmos de cada clasificador pueden obtener patrones más adecuados que se ajusten a las necesidades deseadas.
- Se recomienda que a nivel de diseño de interfaz se trabaje con pausas de tiempo cuando se realicen bucles de toma de datos, ya que, en algunos casos, debido a la velocidad de procesamiento de la computadora, estos suelen ignorar las funcionalidades de otros elementos como botones y cambios entre ventanas de operador.

- Se sugiere realizar una investigación amplia de los clasificadores que se pueden implementar en sistemas que posean datos biomecánicos, ya que, por más que la eficiencia sea muy alta, debido al algoritmo utilizado, los resultados al momento de su implementación pueden reducirse de manera significativa.
- Se recomienda como futuro trabajo realizar este proyecto con personas y no mediante emulación de movimiento en un maniquí y que, se aumenten mayores características a monitorear como, por ejemplo, el uso de señales electromiografías para la detección del estado muscular y así crear un sistema de monitoreo más completo y que cuente con más funciones para determinar afectaciones en la extremidad superior. Debido a que más información permite mejorar el aprendizaje de los clasificadores.

## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] Á. Zárraga López-Quiles, “Diseño y Desarrollo de un sistema para la valoración de la movilidad articular de la muñeca mediante tecnología inercial”, 2016, Universitat Politècnica de València, Accedido: Oct. 27, 2021, [Online-PDF], Disponible: [https://riunet.upv.es/bitstream/handle/10251/89000/TFG\\_A%20Ivaro\\_15047696490784563027531439537224.pdf?sequence=2](https://riunet.upv.es/bitstream/handle/10251/89000/TFG_A%20Ivaro_15047696490784563027531439537224.pdf?sequence=2).
- [2] D. Knudson, “Fundamentals of Biomechanics”, 2007, Editorial Springer, 2nd ed, Accedido: Oct. 27, 2021, [Online], Disponible: <https://tinyurl.com/yzzvz2g3>
- [3] Medline, “Lesiones y enfermedades del brazo” 2020, Medlineplus.gov, Accedido: Nov. 14, 2021, [Online], Disponible: <https://medlineplus.gov/spanish/arminjuriesanddisorders.html>
- [4] H. Ricardo, “Miembro superior: Osteología, miología y artrología. Proporciones y módulos”, 2018, Universidad Complutense Madrid, Accedido: Oct. 27, 2021, [Online], Disponible: <https://eprints.ucm.es/id/eprint/45915/1/TEMA%205.%20Miembro%20superior.pdf>.
- [5] Marketing SEO S.L, “Anatomía Topográfica,” Anatomía Topográfica, Sep. 02, 2021. Accedido Oct. 27, 2021, [Online], Disponible: <https://anatomiatopografica.com/extremidad-superior/> .
- [6] A.I.Kapandji, "Codo, La Pronosupinación" Fisiología Articular, Editorial Medica Panamericana, Tomo I, 6th ed. Madrid, 2006.
- [7] T. Seel, Jörg Raisch, and T. Schauer, “IMU-Based Joint Angle Measurement for Gait Analysis,” ResearchGate, Apr. 2014. [Online]. Accedido: Oct. 29, 2021 [https://www.researchgate.net/publication/261756955\\_IMU-Based\\_Joint\\_Angle\\_Measurement\\_for\\_Gait\\_Analysis](https://www.researchgate.net/publication/261756955_IMU-Based_Joint_Angle_Measurement_for_Gait_Analysis) .
- [8] A. Bose, S.Puri, P. Banerjee, "Accelerometers, Gyros", Modern Inertial Sensors and Systems, Nueva Delhi, India, 2014, ch 3 y 4, pp 59 . 100
- [9] L. Ardilla and Dialnet, “Diseño, calibración y prueba de una unidad de medida inercial (IMU) inalámbrica para cohetes,” Ingenium, vol. 13, no. 25, pp. 36–43, 2012, doi: <https://dialnet.unirioja.es/descarga/articulo/5038483.pdf>.

- [10] E. R. Bachmann, X. Yun, and A. Brumfield, "Limitations of Attitude Estimation Algorithms for Inertial/Magnetic Sensor Modules," Nps.edu, 2021, doi: <http://hdl.handle.net/10945/46086>.
- [11] InvenSense, "MPU-9250 Product Specification Revision 1.1 MPU-9250 Product Specification", Jun, 20, 2016, InvenSense INC, 1ra ed, Accedido: Oct. 29, 2021. [Online]. Disponible: <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>.
- [12] A. Moreno, "EL BUS I2C", 2004, 4ta ed, Universidad de Córdoba-España, Accedido: Oct. 30, 2021. [Online]. Disponible: <http://www.uco.es/~el1mofer/Docs/IntPerif/Bus%20I2C.pdf>.
- [13] Share-Alike, "Arduino Nano User Manual", Arduino SRL, 2da ed, Accedido: Jul. 20, 2021 [Online-PDF], Disponible : <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>.
- [14] G. Wetzstein, "Inertial Measurement Units I", 2021, Universidad de Standford, Accedido: Oct. 30, 2021. [Online], Disponible: <https://stanford.edu/class/ee267/lectures/lecture9.pdf>.
- [15] MathWorks, "MATLAB - El lenguaje del cálculo técnico," Mathworks.com, 2021. [Online]. Accedido Nov. 14, 2021. Disponible: <https://la.mathworks.com/products/matlab.html>
- [16] M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, "BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms," *Source Code for Biology and Medicine*, vol. 8, no. 1, Apr. 2013, doi: 10.1186/1751-0473-8-11.
- [17] Jingwei Too, Abdul Rahim Abdullah, and Norhashimah Mohd Saad, "A Comparative Analysis of Wavelet Families for the Classification of Finger Motions," *ResearchGate*, May 2019. [Online]. Accedido Nov. 03, 2021. Disponible: [https://www.researchgate.net/publication/334192564\\_A\\_Comparative\\_Analysis\\_of\\_Wavelet\\_Families\\_for\\_the\\_Classification\\_of\\_Finger\\_Motions](https://www.researchgate.net/publication/334192564_A_Comparative_Analysis_of_Wavelet_Families_for_the_Classification_of_Finger_Motions).
- [18] P. Faraldo y B. Pateiro, "Estadística y metodología de la investigación Tema 1. Estadística Descriptiva," 2012, Universidad de Santiago de Compostela, Accedido: Oct. 25, 2021 [Online]. Disponible:



[http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/MATERIALES/Mat\\_G2021103104\\_EstadisticaTema1.pdf](http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/MATERIALES/Mat_G2021103104_EstadisticaTema1.pdf).

- [19] K. S. Kim, H. H. Choi, C. S. Moon, y C. W. Mun, "Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions," *Current Applied Physics*, vol. 11, no. 3, pp. 740–745, May. 2011, doi: 10.1016/j.cap.2010.11.051, Accedido: Oct. 20, 2021, [PDF].
- [20] Scala Higher Education, "ESTADÍSTICA DESCRIPTIVA.", 2021, Scala Higher Education, 1ra ed, Accedido: Nov. 04, 2021. [Online]. Disponible: [https://repositorio.scalahed.com/recursos/files/r176r/w34389w/EstadisticaForense\\_Ant\\_B2\\_S.pdf](https://repositorio.scalahed.com/recursos/files/r176r/w34389w/EstadisticaForense_Ant_B2_S.pdf).
- [21] Matlab, "Comparison with Euclidean Norm," Mathworks.com, 2013. [Onlines]. Accedido Nov. 04, 2021. Disponible: <https://la.mathworks.com/help/signal/ref/bandpower.html> ().
- [22] P. Tárrega, "Herramientas Imprescindibles para Machine Learning Video-MATLAB", 2016. [Online]. Disponible : <https://www.mathworks.com/videos/essential-tools-for-machine-learning-1488913197554.html>.
- [23] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018, doi: 10.1109/tccn.2018.2881442, Accedido: Ene. 22, 2022, [PDF]
- [24] L. Quintanilla, "Entrenamiento y evaluación de un modelo - ML.NET", Oct. 11, 2021 Microsoft.com, Accedido: Ene. 22, 2021, [Online], Disponible: <https://docs.microsoft.com/es-es/dotnet/machine-learning/how-to-guides/train-machine-learning-model-ml-net>
- [25] Eijaz Allibhai, "Holdout vs. Cross-validation in Machine Learning," *Medium*, Oct. 03, 2018. <https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f>.
- [26] G. Betancourt, "LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs)", Apr. 2005, editorial Scientia Et Technica, 1ra ed, Vol.1, Accedido: Nov. 07, 2021, [PDF], Disponible: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/6895>

- [27] A. Zapata-Tapasco, S. Pérez-Londoño, and J. Mora-Flórez, "Método basado en clasificadores k-NN parametrizados con algoritmos genéticos y la estimación de la reactancia para localización de fallas en sistemas de distribución", Mar. 2014, Revista Facultad de Ingeniería de la Universidad de Antioquia-Colombia, Accedido: Nov. 07, 2021. [Online]. Disponible: <https://www.redalyc.org/articulo.oa?id=43030033020>.
- [28] A. Moujahid, I. Inza, and P. Larrañaga, "Tema 5. Clasificadores K-NN", 2021, Universidad del País Vasco-Euskal Herriko, 1ra ed, Accedido: Nov. 07, 2021. [Online]. Disponible: <http://www.sc.ehu.es/ccwbyes/docencia/mmcc/docs/t9knn.pdf>.
- [29] S. Beltrán, "Qué son las redes neuronales y sus funciones," *ATRIA Innovation*, Oct. 22, 2019, Accedido: Oct 30, 2021, [Online], Disponible: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>.
- [30] Arduino, "Arduino Nano", Ene. 15, 2019, Arduino.cl, Accedido: Ene. 22, 2022, [Online], Disponible: <https://arduino.cl/arduino-nano/#:~:text=Tiene%2014%20pines%20de%20entrada,y%20un%20bot%C3%B3n%20de%20reseteo>
- [31] Microchip, "SearchMICROCHIP Datasheet", 2018, Microchip.com, 1ra edición, Accedido: Ene. 22, 2022, [Online], Disponible: <https://www.microchip.com/sitesearch/search/All/164p>
- [32] Parallax, "P8X32A Propeller Datasheet - Parallax", Oct. 14, 2020, Parallax, Accedido: Ene. 22, 2022, [Online], Disponible: <https://www.parallax.com/package/p8x32a-propeller-datasheet/>
- [33] NXP Semiconductors, "UM10204 I2C-bus specification and user manual", 2015, 1ra edición, NXP Semiconductors, Accedido: Oct. 2021, [Online], Disponible: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>.
- [34] R. Arora, "I2C Bus Pullup Resistor Calculation Application Report I2C Bus Pullup Resistor Calculation", 2015, Texas Instruments, Accedido: Ene. 22, 2022, [Online], Disponible: <https://www.ti.com/lit/an/slva689/slva689.pdf>
- [35] Atmel, "ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET" Ene. 2015, Atmel, 1ra Edición, Accedido: Ene. 22, 2022, [Online], Disponible: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf).

- [36] Naylamp Meckatronics, "MÓDULO BLUETOOTH HC06," Naylamp Meckatronics, 2021. [Online]. Accessed Nov. 17, 2021. Disponible: <https://naylampmechatronics.com/inalambrico/24-modulo-bluetooth-hc06.html>
- [37] S. García, S. Ramírez-Gallego, J. Luengo, and F. Herrera, "Big Data monografía monografía Big Data: Preprocesamiento y calidad de datos," Oct. 2016, Revista Universidad de Granada-España, 237va ed, Accessed: Nov. 16, 2021. [Online-PDF], Disponible: [https://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2133\\_Nv237-Digital-sramirez.pdf](https://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2133_Nv237-Digital-sramirez.pdf).
- [38] P. Cetta, "FILTROS DIGITALES Principios de los filtros digitales.", 2009, editorial FACM-UCA, Universidad Católica de Argentina, Accedido: Oct. 14, 2021, [Online], Disponible: <https://www.pablocetta.com/pdfs/publicaciones/filtros.pdf>.
- [39] B. Taylor, "Bolder Flight Systems / software / mpu9250", Jun. 10, 202, GitLab, Accedido: Ene. 23, 2022, [Online], Disponible: <https://gitlab.com/bolderflight/software/mpu9250/>
- [40] MatLab, "App Classification Learner - MATLAB & Simulink - MathWorks América Latina," [la.mathworks.com](https://la.mathworks.com), 2021. [Online]. Accedido: Nov. 18, 2021. Disponible: <https://la.mathworks.com/help/stats/classification-learner-app.html>
- [41] A. Mannini and A. M. Sabatini, "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, Feb. 2010, doi: 10.3390/s100201154.
- [42] J. I. Pilataxi Piltaxi, M. F. Trujillo Guerrero, V. C. Benavides Laguapillo, and J. A. Rosales Acosta, "Human Activity Recognition Using an Accelerometer Magnitude Value," *Communications in Computer and Information Science*, pp. 462–472, 2020, doi: 10.1007/978-3-030-42520-3\_37.
- [43] Hospital Universitario Ramón y Cajal SDA, "Curvas ROC", 2021, Salud Madrid - Hospital Universitario Ramón y Cajal, Madrid-España, Accedido: Oct. 13, 2021, [Online], Disponible: [http://www.hrc.es/bioest/roc\\_1.html](http://www.hrc.es/bioest/roc_1.html).
- [44] J. Chavez, "HDU 5095 linealización de las funciones del kernel en SVM (simulación)", 2021, Programmer Click, Accedido: Oct. 17, 2021, [Online], Disponible: <https://programmerclick.com/article/33542297035/> (accessed Nov. 29, 2021).
- [45] J. Chavez, "Minería de datos-KNN-K algoritmo vecino más cercano", 2021, Programmer Click, Accedido: Nov. 29, 2021, [Online], Disponible: <https://programmerclick.com/article/5615574574/> .

# **ANEXOS**

Anexo A. Manual de usuario.

## **ANEXO A**

### **MANUAL DE USUARIO DE LA INTERFAZ GRÁFICA**








## **CONTENIDO**


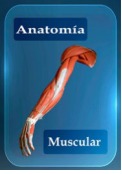





- A1. Descripción de la interfaz gráfica
- A2. Botoneras principales
- A3. Pantallas de la interfaz gráfica
- A4. Ejecución de la interfaz
  - A4.1. Instalación de la aplicación
  - A4.2. Inicio de la toma de datos

## A1. DESCRIPCIÓN DE LA INTERFAZ GRÁFICA

La interfaz gráfica de usuario implementada en el proyecto técnico titulado “SISTEMA DE EMULACIÓN DE MOVIMIENTO DE LA EXTREMIDAD SUPERIOR PARA DETERMINAR POSICIÓN” fue desarrollado en el software Matlab2021b de MathWorks®, con el objetivo de realizar una interfaz dinámica que permita navegar entre pantallas para obtener información relevante sobre el proyecto y sus utilidades, así como la presentación del resultado final del proyecto, se optó por utilizar la herramienta App Designer de Matlab, esta herramienta permite crear aplicaciones con un interfaz gráfico de alta calidad y dinamizar la interacción con el usuario, finalmente se presentan dos secciones importantes en el interfaz, la primera es la información sobre el proyecto, como la ubicación de los sensores y el sistema anatómico que permite el movimiento de la extremidad superior, mientras que la otra sección presenta en una ventana los datos recibidos por los sensores y la detección de movimiento, permitiendo elegir el tipo de clasificador utilizado para trabajar con las señales de los sensores.

## A2. BOTONERAS PRINCIPALES

Botón	Función
	Se encuentra en la pantalla principal, y permite dirigirse a la ventana llamada menú donde se pueden elegir las dos secciones de la interfaz.
	Permite dirigirse a la ventana de inicio.
	Permite dirigirse a la ventana de información del proyecto.
	Permite dirigirse a la ventana donde se encuentra el resultado final del proyecto.
	Permite dirigirse a la ventana donde se encuentra la información de la ubicación de los sensores.
	Permite dirigirse a la ventana donde se presenta información sobre la biomecánica de la extremidad superior.
	Permite dirigirse a la ventana donde se presenta información del proceso del proyecto.

	Permite dirigirse a la ventana donde se presenta información sobre las articulaciones y movimientos de la extremidad superior.
	Permite dirigirse a la ventana donde se presenta información sobre los músculos y movimientos de la extremidad superior.
	Permite dirigirse a la ventana de inicio.
	Permite dirigirse a la ventana de menú.
	Permite dirigirse a la ventana anterior a la que se encuentra.
	Permite iniciar la toma de datos de los sensores.
	Permite detener la toma de datos de los sensores.

### A3. PANTALLAS DE LA INTERFAZ GRÁFICA

El interfaz gráfico cuenta con 9 ventanas interactivas las cuales será descritas a continuación.

La primera ventana mostrada en la Figura A1 es la ventana de inicio, esta cuenta con la presentación del proyecto y solamente tiene un botón que permite dirigirse a la siguiente ventana que es el menú.



Figura A 1: Ventana de inicio.



La siguiente ventana es la del menú aquí se encuentran tres botoneras, una permite regresar al inicio de la aplicación, otra que permite ir a la ventana de información y la última permite dirigirnos a la ventana de aplicación, esta ventana es central y permite que se navegue por las 9 ventanas con mayor facilidad.



**Figura A2:** Ventana de menú.

La ventana mostrada en la Figura A3 muestra la información necesaria sobre el procedimiento, en ella se encuentra 6 botoneras, tres de ellas son pequeñas y se encuentran en la parte inferior, permiten movilizarse en las otras ventanas con facilidad son los botones de regreso, menú e inicio. Mientras que hay otros botones que permite dirigirse a la ventana de ubicación de los sensores, de información del proceso y de información sobre la biomecánica.



**Figura A3:** Ventana de información general

En la Figura A4 se muestra la ventana de la ubicación de los sensores, esta muestra la información de donde se deben colocar los sensores para determinar el movimiento y posición de la extremidad superior, cuenta con tres botoneras que permiten navegar entre las otras ventanas del interfaz.



**Figura A4:** Ventana de Ubicación de los Sensores.

En la Figura A5, se muestra la ventana de biomecánica, aquí se cuenta con una pequeña información sobre la biomecánica y la importancia de analizarla para este proyecto y cuenta con 5 botones, tres de ellos permiten navegar entre las otras ventanas del interfaz y dos botones que permiten obtener una información más específica sobre la anatomía y el movimiento estos son el botón de Anatomía de la Articulación y el botón de Anatomía Muscular.



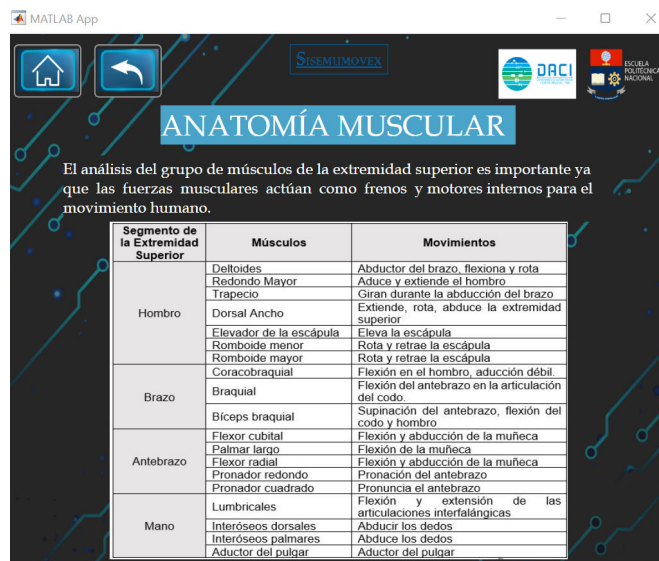
**Figura A5:** Ventana de Información de Biomecánica.

En la Figura A6, se muestra la ventana de información de la anatomía de la articulación, en esta ventana se encuentran datos de las articulaciones que se utilizan en el movimiento de la extremidad superior y que tipos de movimiento permiten realizar las mismas, esto permite mejorar la interpretación de los resultados obtenidos, también cuenta con dos botones que facilitan la navegación entre ventanas.



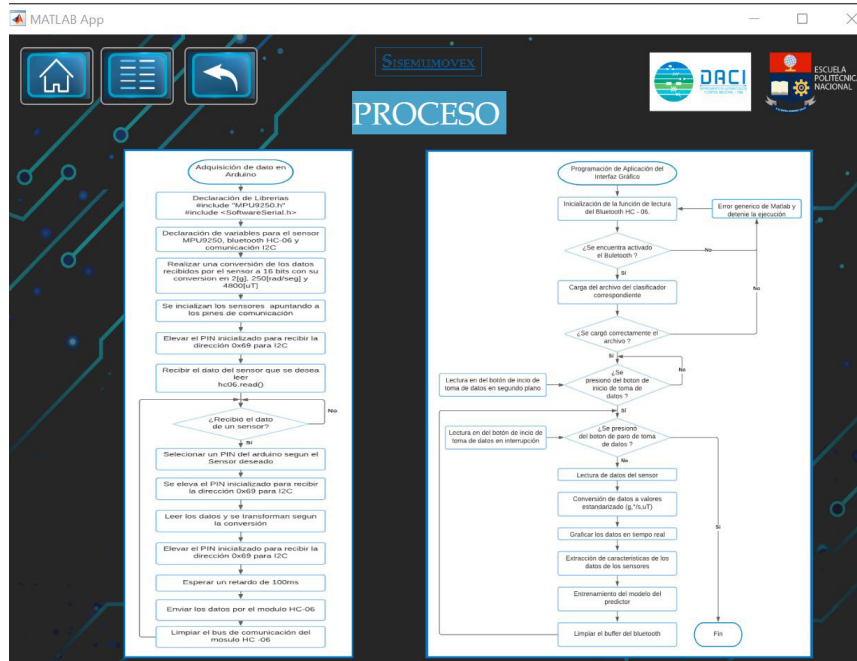
**Figura A6:** Ventana de Información de Anatomía de las Articulaciones.

En la Figura A7 se muestra la ventana de información de la anatomía muscular, en esta se muestra una tabla que indica los músculos por segmentos de la extremidad superior y el movimiento que permiten realizar en el miembro superior facilitando la interpretación de los resultados entregados por el proyecto. También cuenta con dos botones que facilitan la navegación entre ventanas del interfaz.



**Figura A7:** Ventana de Información de Anatomía Muscular.

En la Figura A8 se muestra la ventana de información del proceso, esta se encuentra ubicada en la ventana de información, aquí se presentan los diagramas de flujo que se utilizan para la programación del microprocesador y el diseño del procesamiento y preprocesamiento de la señal. También cuenta con tres botones que facilitan la navegación entre ventanas del interfaz.



**Figura A8:** Ventana de Información del Proceso.

En la Figura A9 se muestra la ventana de aplicación, en esta ventana se inician la toma de datos de los sensores y entrega un resultado del movimiento y posición de la extremidad superior.

Cuenta con cuatro botones, dos de ellos permiten navegar entre varias ventanas de manera dinámica y dos botones uno para iniciar la toma de datos y otro para detener la toma de datos, también cuenta con una lista de botones donde una permite elegir el tipo de clasificador que se desea utilizar para el procesamiento de señal tomando en cuenta que solo puede elegir uno y que por defecto se elige el clasificador de RED NEURONAL, y otra lista de botones para elegir que tipo de señal se desea observar del sensor IMU, donde solo se puede elegir una sola opción y que por defecto se muestran los datos del acelerómetro de los tres sensores.

Cuenta con tres Axes que permiten graficar las señales de los sensores y gracias a estas se pueden tomar los datos de movimiento en cada extremidad, hay tres gráficas ya que

cada una representa un eje espacial (x, y, z) y en esta se grafican los datos de los tres sensores.

Y finalmente se muestra una imagen con el tipo de movimiento realizado y detectado, tomando en cuenta que estas pueden ser supinación, pronación, flexión, y extensión.

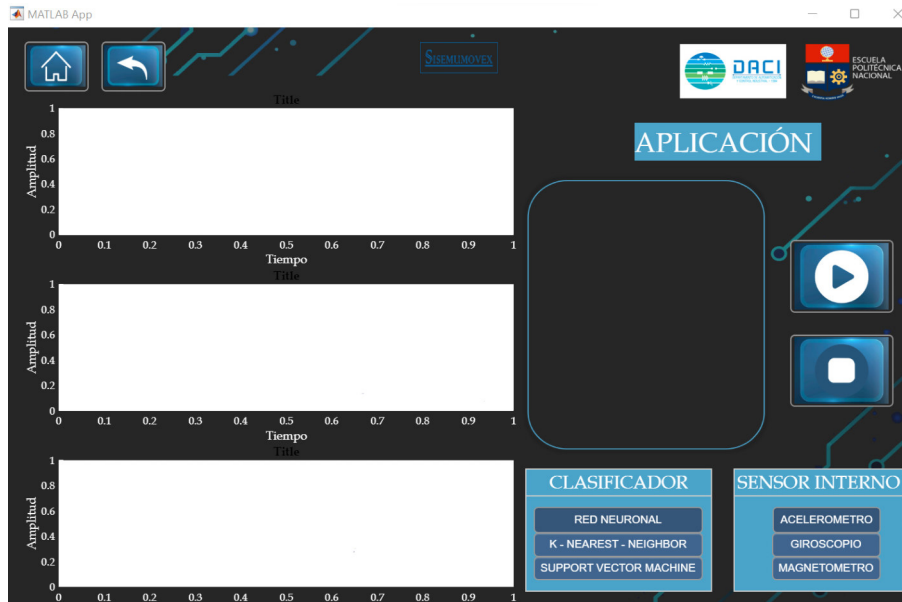
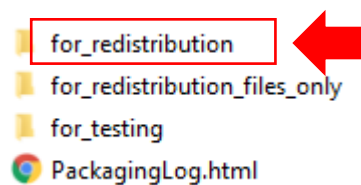


Figura A9: Ventana de la Aplicación

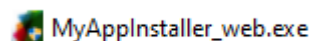
## A4. Ejecución de la interfaz

### A4.1 Instalación de la aplicación

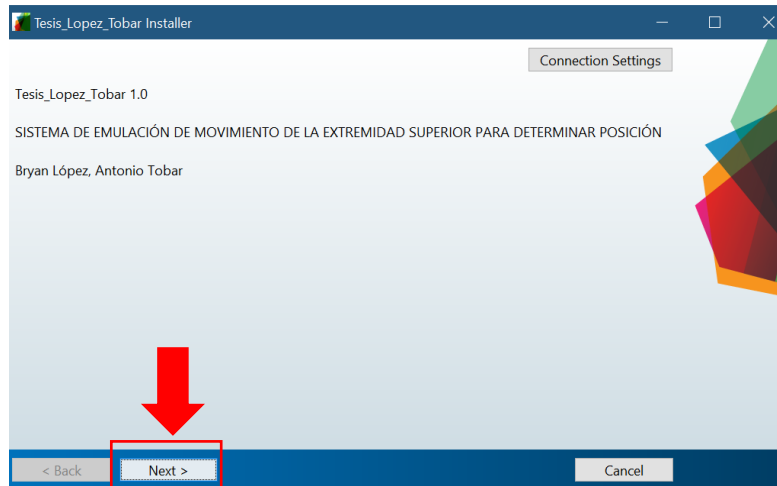
1. Abrir la carpeta llamada **Ventana 1** y se desplegarán tres carpetas, se debe abrir la carpeta **for\_redistribution**.



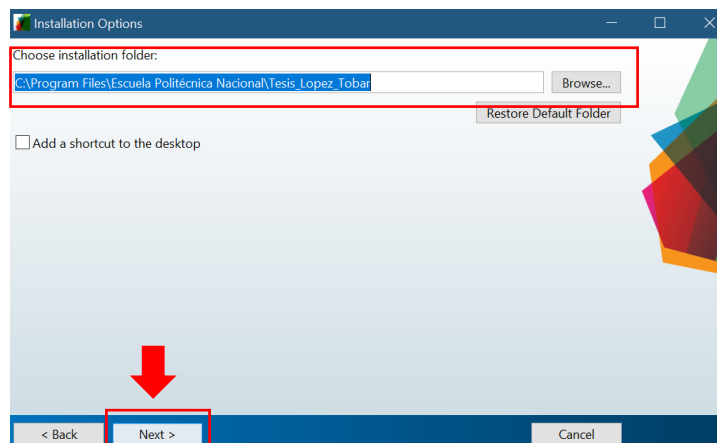
2. Al abrir se debe ejecutar el único archivo presente llamado **MyApplnstaller\_web.exe**



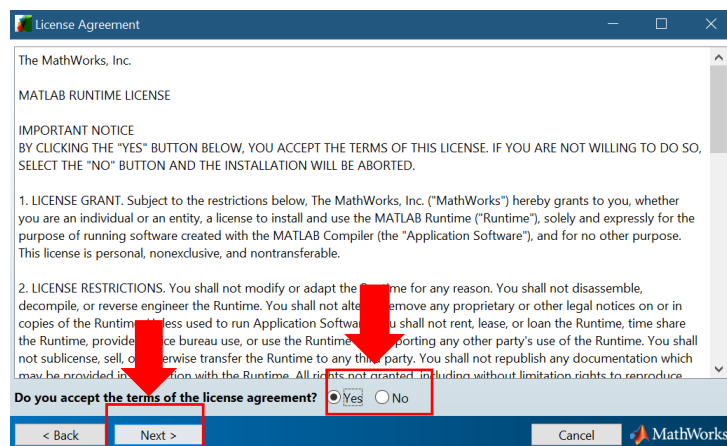
3. Al ejecutar el archivo se puede se pedirán permisos de instalación del sistema y abrirá la ventana de presentación dela aplicación donde se debe seleccionar el botón de **Next**.



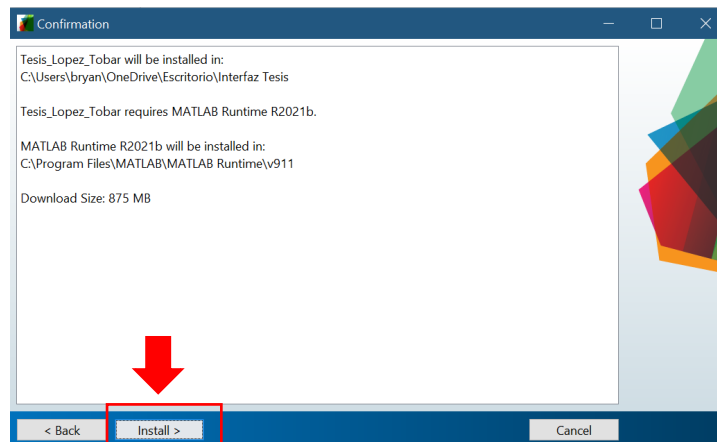
4. Luego se abrirá la ventana donde se debe guardar la aplicación, se recomienda que la carpeta en la que se guardará la aplicación no contenga caracteres especiales. Una vez seleccionada la ventana se debe presionar en **Next**.



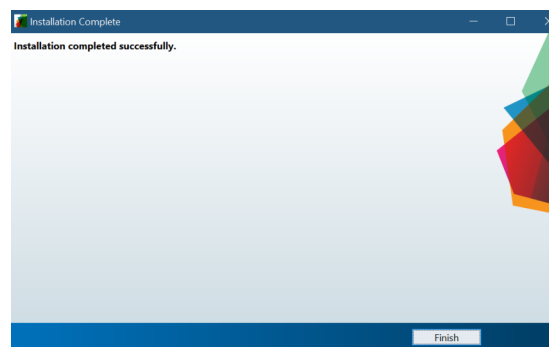
5. Luego se deben aceptar las condiciones de instalación de la aplicación y presionar el botón **Next**.



6. Finalmente se muestra una ventana con el resumen de la instalación. Y al dar clic en ***Install*** empezará la instalación.



7. Al terminar la descarga y la instalación se mostrará la siguiente ventana.



8. Finalmente se puede revisar en el escritorio o en todos los programas la aplicación y se la puede ejecutar.



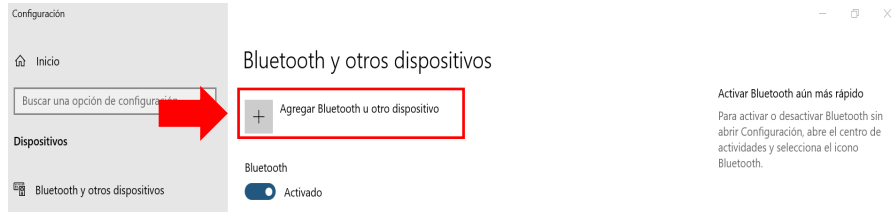
## **A4.2 Inicio de la toma de datos**

Para la toma de datos, se debe tomar en cuenta que la computadora en la que se ejecute la aplicación debe contar con Bluetooth ya que esto permitirá la toma de datos de los sensores.

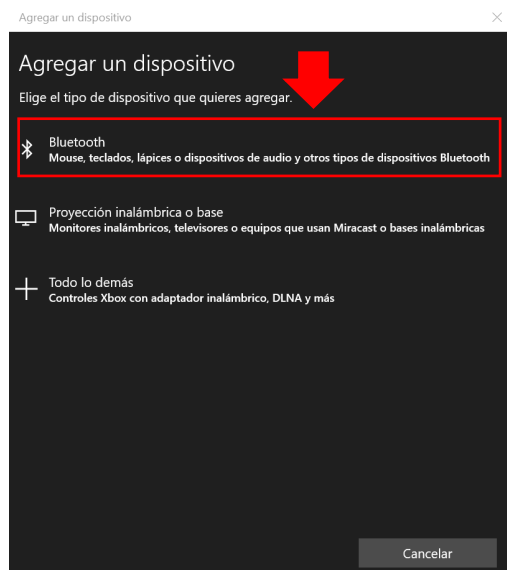
Una vez verificado que el computador cuenta con Bluetooth se debe conectar el mismo con la computadora, para el caso de trabajar con Windows 10 se presenta un breve paso a paso para la conexión de la PC con el dispositivo implementado.



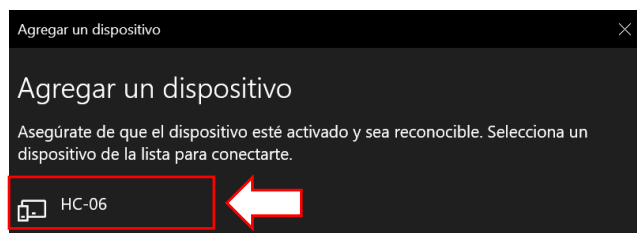
1. Acceder a la configuración *Bluetooth y otros dispositivos de la PC*, esta se encuentra en configuraciones. Y se debe seleccionar la opción *Agregar Bluetooth u otro dispositivo*.



2. Una vez dado clic en esa opción se procede a elegir la opción Bluetooth



3. Se debe esperar un momento hasta que se detecte el dispositivo y se lo debe elegir. En este caso el dispositivo Bluetooth del hardware diseñado para el proyecto se llama **HC-06**.

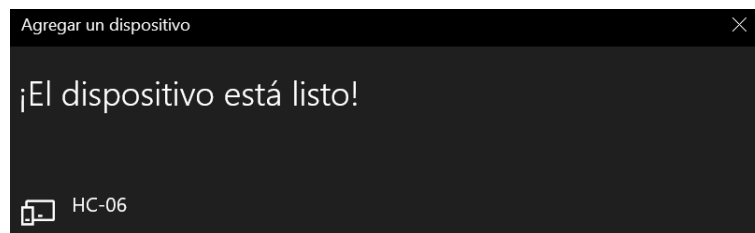


4. Luego de seleccionar se aparecerá una ventana para colocar un código el cual es **1234** para poder vincular el dispositivo con la PC.





5. Finalmente se comprueba que el dispositivo se encuentra vinculado.



6. Nota: En algunos dispositivos se muestra en el estado no conectado el bluetooth, si no solo aparece vinculado, es porque al arrancar en el interfaz la toma de datos con un comando de la programación se activa el bluetooth.

Una vez vinculado el bluetooth con la PC en la ventana 11 se muestran los botones, estos permiten arrancar o parar la toma de datos, estos datos se mostrarán en el Axes y el resultado en una imagen.

