

**Universidad Autónoma de Baja California**  
Instituto de Ingeniería  
Maestría y Doctorado en Ciencias e Ingeniería



**Reconocimiento de Movimientos Humanos  
utilizando Dispositivos Inteligentes mediante  
Técnicas de Minería de Datos**

*Tesis que para obtener el grado de:*

*MAESTRO EN CIENCIAS*

***Presenta***

Ing. Marcial Sandoval Gastelum

**Director de Tesis:**

Dr. Félix Fernando González Navarro

**Codirector de Tesis:**

Dra. Brenda Leticia Flores Ríos

---

# Dedicatorias

Para Dios en primer lugar por permitirme llegar y terminar esta etapa en mi formación académica.

A mis padres , Beatriz Gastelum Cáñez y Marcial L. Sandoval Banda, quienes siempre han estado conmigo a cada paso que doy en este camino llamado vida.

## Reconocimientos

Primeramente, al Consejo Nacional de Ciencia y Tecnología (CONACYT), que en conjunto con la Universidad Autónoma de Baja California (UABC) y el Instituto de Ingeniería ofrecieron los recursos y herramientas necesarias para llevar a cabo este proyecto.

Al Comité de Computación conformado el Dr. Félix Fernando González Navarro, la Dra. Brenda L. Flores Ríos , la Dra. Larysa Burtseva y el Dr. Gabriel A. López Morteo, por abrirme las puertas al Instituto de Ingeniería y darme la oportunidad de seguir preparándome académicamente y profesionalmente.

Al asesor y tutor, el Dr. Félix Fernando González Navarro, por haber visto en mí la capacidad necesaria para terminar el proyecto emprendido al inicio del programa y por todo su apoyo brindado durante el transcurso del mismo.

A mi co-tutora, la Dra. Brenda L. Flores Ríos , por siempre darse un espacio en su apretada agenda para poderme solucionar todas mis dudas e inquietudes.

A la Coordinadora del Programa MyDCI, la Dra. Jesus Eliana Rodríguez Burgueño por su disposición extraordinaria para brindar apoyo.

## Resumen

La cantidad de personas que trabajan en casa ha aumentado en los últimos años y profesionales con roles a distancia han surgido. Por esta razón, el desarrollo de tecnología para la detección de gestos y actividades humanas de la vida diaria es importante para este sector de la población en constante crecimiento. En este documento se propone un sistema detallado de reconocimiento de actividades humanas basado en Minería de Datos que funciona con datos provenientes de un acelerómetro y un giroscopio. El propósito del presente estudio es encontrar un modelo clasificador de cuatro actividades humanas (subir escaleras, bajar escaleras, ponerse de pie y sentarse) que pueda ser usado en cualquier dispositivo para muñeca que cuente con acelerómetro y giroscopio. Cuatro técnicas de minería de datos arrojaron los mejores resultados en reconocimiento: K Nearest Neighbors, Support Vector Machines, Bagged Trees Ensemble y una red neuronal de una sola capa oculta, alcanzando un reconocimiento máximo de 96.4%.

# Contenido

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Planteamiento del problema . . . . .	1
1.2	Objetivo General . . . . .	5
1.2.1	Objetivos Específicos . . . . .	5
1.3	Estructura del documento de tesis . . . . .	6
<b>2</b>	<b>Marco Teórico</b>	<b>7</b>
2.1	Sistemas de Reconocimiento de Actividades Humanas . . . . .	7
2.2	Activity Recognition Chain (ARC) . . . . .	8
<b>3</b>	<b>Materiales y Métodos</b>	<b>11</b>
3.1	Dispositivo utilizado . . . . .	11
3.1.1	Descripción Técnica . . . . .	12
3.1.2	Sensores . . . . .	13
3.2	Metodología . . . . .	15
3.2.1	Metodología utilizada para el desarrollo de software para captura de muestras . . . . .	16
3.2.2	Metodología utilizada para minería de datos . . . . .	18
3.2.3	Métodos de Validación . . . . .	31
3.2.4	Métodos de Visualización . . . . .	31
<b>4</b>	<b>Resultados y Discusión</b>	<b>32</b>
4.1	Fase 1: Desarrollo de Software para captura de muestras . . . . .	32
4.1.1	Requisitos de Software . . . . .	32
4.1.2	Diseño . . . . .	34

4.1.3	Requerimientos no funcionales . . . . .	37
4.1.4	Requerimientos funcionales . . . . .	38
4.1.5	Casos de uso . . . . .	39
4.1.6	Casos de prueba . . . . .	40
4.2	Fase 2: Captura de muestras para prueba y entrenamiento . . . . .	44
4.2.1	Descripción de los datos . . . . .	47
4.3	Fase 3: Minado de Datos . . . . .	50
4.3.1	Preprocesamiento . . . . .	50
4.3.2	Algoritmos de Clasificación . . . . .	53
4.3.3	Métodos de Visualización . . . . .	72
4.3.4	Comparación de Modelos encontrados . . . . .	73
4.3.5	Selección de mejor Modelo . . . . .	77
<b>5</b>	<b>Conclusiones</b>	<b>79</b>
5.1	Conclusiones . . . . .	79
5.2	Trabajo Futuro . . . . .	80
5.3	Publicación . . . . .	81
5.3.1	Encuentro Nacional de Computación . . . . .	81
<b>A</b>	<b>Gráficas de Movimientos</b>	<b>82</b>
A.0.1	Probador 2 . . . . .	82
A.0.2	Probador 3 . . . . .	84
<b>B</b>	<b>Detección de outliers</b>	<b>87</b>
B.1	Movimientos de Probador 1 . . . . .	87
B.1.1	DWN . . . . .	87
B.1.2	SIT . . . . .	89
B.1.3	STD . . . . .	91
B.2	Movimientos de Probador 2 . . . . .	93
B.2.1	UP . . . . .	93
B.2.2	DWN . . . . .	95
B.2.3	SIT . . . . .	97
B.2.4	STD . . . . .	99
B.3	Movimientos de Probador 3 . . . . .	101

## CONTENIDO

---

B.3.1	UP . . . . .	101
B.3.2	DWN . . . . .	103
B.3.3	SIT . . . . .	105
B.3.4	STD . . . . .	107
<b>C</b>	<b>Anexos del Software MyBand</b>	<b>109</b>
C.0.1	Escenarios de uso . . . . .	110
C.0.2	Flujos Básicos . . . . .	115
C.0.3	Casos de Prueba: Flujos Positivos . . . . .	117
C.0.4	Sublujos Alternativos . . . . .	122
C.0.4.1	Escenarios de uso . . . . .	122
	<b>Referencias</b>	<b>126</b>

# Lista de Figuras

2.1	Metodología ARC . . . . .	9
3.1	Dispositivo utilizado. . . . .	13
3.2	Metodología . . . . .	15
3.3	Dualidad Proceso-Producto en la Ingeniería de Software . . . . .	17
3.4	Metodología utilizada para la minería y entrenamiento de los datos	19
3.5	Método DeepInsight. (a) Conversión del vector de atributos a una matriz de atributos. (b) Conversión de matriz de atributos a ima- gen con píxeles. . . . .	27
4.1	Modelo de Datos. . . . .	34
4.2	Comprobación de comunicación con la MSBand . . . . .	39
4.3	Diagrama de caso de uso de MyBand . . . . .	40
4.4	Escenarios de uso relacionados FB4.1 al FB4.13 . . . . .	42
4.5	Escenarios de uso relacionados con FB4.15 . . . . .	43
4.6	Diagrama de Despliegue . . . . .	45
4.7	Posición de la MSBand en portador. . . . .	46
4.8	Muestras de los movimientos a reconocer . . . . .	46
4.9	Movimientos UP y DWN realizados por Probador 1 . . . . .	48
4.10	Movimientos SIT y STD realizados por Probador 1 . . . . .	49
4.11	Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 1. . . . .	51
4.12	Outlier detectado para el movimiento UP efectuado por el Probador 1 . . . . .	52
4.13	Gráfica de desempeño de KNN . . . . .	54

## LISTA DE FIGURAS

---

4.14	Curva ROC para Vecinos Cercanos . . . . .	55
4.15	Gráfica de desempeño de SVM . . . . .	57
4.16	Curva ROC para SVM . . . . .	58
4.17	Diagrama de Árbol de Decisión . . . . .	59
4.18	Gráfica de desempeño del Árbol de Decisión simple . . . . .	60
4.19	Curva ROC para Árbol de Decisión simple . . . . .	61
4.20	Gráfica de desempeño del Conjunto de Árboles de Decisión . . . . .	62
4.21	Curva ROC para Conjunto de Árboles de Decisión . . . . .	63
4.22	Arquitectura de la Red Neuronal . . . . .	64
4.23	Desempeño de la red neuronal durante su entrenamiento. . . . .	65
4.24	Curva ROC para Red Neuronal . . . . .	66
4.25	Datos de $G$ utilizando PCA con 2 componentes principales . . . . .	67
4.26	Envolvente convexa mínima . . . . .	67
4.27	Proceso de rotación de la envolvente convexa mínima para encontrar la posición final de los píxeles asignados a los atributos de cada muestra de $M$ . . . . .	68
4.28	Mapeo de muestra . . . . .	69
4.29	Arquitectura de la CNN . . . . .	70
4.30	Proceso de entrenamiento del modelo DeepInsight . . . . .	71
4.31	Gráfica ROC resultante de la metodología DeepInsight . . . . .	72
4.32	Movements Dataset . . . . .	73
4.33	Comparación de curvas ROC para UP . . . . .	74
4.34	Comparación de curvas ROC para DWN . . . . .	75
4.35	Comparación de curvas ROC para SIT . . . . .	76
4.36	Comparación de curvas ROC para STD . . . . .	77
4.37	Matrices de Confusión de Salida . . . . .	78
A.1	Movimientos UP y DWN realizados por Probador 2 . . . . .	83
A.2	Movimientos SIT y STD realizados por Probador 2 . . . . .	84
A.3	Movimientos UP y DWN realizados por Probador 3 . . . . .	85
A.4	Movimientos SIT y STD realizados por Probador 3 . . . . .	86
B.1	Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 1. . . . .	88

## LISTA DE FIGURAS

---

B.2	Outliers detectados para el movimiento DWN efectuado por el Probador 1 . . . . .	89
B.3	Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 1. . . . .	90
B.4	Outliers detectados para el movimiento SIT efectuado por el Probador 1 . . . . .	91
B.5	Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 1. . . . .	92
B.6	Outliers detectados para el movimiento STD efectuado por el Probador 1 . . . . .	93
B.7	Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 2. . . . .	94
B.8	Outliers detectados para el movimiento UP efectuado por el Probador 2 . . . . .	95
B.9	Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 2. . . . .	96
B.10	Outliers detectados para el movimiento DWN efectuado por el Probador 2 . . . . .	97
B.11	Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 2. . . . .	98
B.12	Outliers detectados para el movimiento SIT efectuado por el Probador 2 . . . . .	99
B.13	Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 2. . . . .	100
B.14	Outliers detectados para el movimiento STD efectuado por el Probador 2 . . . . .	101
B.15	Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 3. . . . .	102
B.16	Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 3. . . . .	104
B.17	Outliers detectados para el movimiento DWN efectuado por el Probador 3 . . . . .	105

## LISTA DE FIGURAS

---

B.18 Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 3. . . . .	106
B.19 Outliers detectados para el movimiento SIT efectuado por el Probador 3 . . . . .	107
B.20 Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 3. . . . .	108
C.1 Escenarios de uso relacionados con REQ3 y REQ4 . . . . .	110
C.2 Escenarios de uso relacionados con REQ5-10 . . . . .	111
C.3 Escenarios de uso relacionados con REQ11-12 . . . . .	112
C.4 Escenarios de uso relacionados con REQ12-17 . . . . .	113
C.5 Escenarios de uso relacionados con REQ18 . . . . .	114
C.6 Escenarios de uso relacionados con Subflujo FB2 . . . . .	116
C.7 Caso de prueba Frequency Based CSV - 1 hz . . . . .	119
C.8 Caso de prueba Frequency Based CSV - 2 hz . . . . .	119
C.9 Caso de prueba Time Based CSV . . . . .	120
C.10 Caso de prueba Sample Based CSV . . . . .	121
C.11 Casos de Prueba alternativos . . . . .	122
C.12 Usuario cancela configuraciones . . . . .	123

# Lista de Tablas

1.1	Características principales de los sistemas de reconocimiento de actividades humanas . . . . .	4
3.1	Descripción Técnica de la Microsoft Band 2. . . . .	12
3.2	Lista descriptiva de los sensores de la MSBand. . . . .	14
3.3	Descripción de roles involucrados . . . . .	15
3.4	Matriz de Confusión . . . . .	27
4.1	Descripción de tablas. . . . .	35
4.2	Contenido de sample_rate . . . . .	36
4.3	Contenido de sensor_table . . . . .	36
4.4	Requerimientos no funcionales . . . . .	37
4.5	Requerimientos Funcionales. . . . .	39
4.6	Subflujo FB4 . . . . .	41
4.7	Matriz de Confusión KNN . . . . .	54
4.8	Medidas de desempeño para KNN . . . . .	55
4.9	Matriz de Confusión SVM . . . . .	57
4.10	Medidas de desempeño para SVM . . . . .	58
4.11	Matriz de Confusión Árbol de Decisión simple . . . . .	60
4.12	Medidas de desempeño para Árbol de Decisión simple . . . . .	61
4.13	Matriz de Confusión del Conjunto de Árboles de Decisión . . . . .	63
4.14	Medidas de desempeño para Conjunto de Árboles de Decisión . . . . .	64
4.15	Medidas de desempeño para Red Neuronal . . . . .	65
4.16	Medidas de desempeño del modelo DeepInsight . . . . .	72
C.1	Subflujo FB1 . . . . .	115

## LISTA DE TABLAS

---

C.2	Subflujo FB2 . . . . .	115
C.3	Subflujo FB3 . . . . .	117
C.4	Defición de columnas en tabla exportada en formato Frequency Based y Time Based . . . . .	118
C.5	Defición de columnas en tabla exportada en formato Sample Based	119

# Capítulo 1

## Introducción

### 1.1 Planteamiento del problema

En el pasado, las actividades de la vida diaria eran relacionadas directamente en áreas del cuidado de la salud pero conforme la industria de dispositivos móviles y la computación ubicua fue creciendo, empezó a llamar más la atención de la comunidad de ciencias computacionales, ya que los sensores que estos contenían aportan una gran cantidad de información que se puede interpretar para darle un sentido y nuevas aplicaciones para mejorar el estilo de vida de las personas.

A su vez, la cantidad de personas que trabajan en casa ha aumentado en los últimos años y profesionales con roles a distancia han surgido. De acuerdo con [Kwok \(2017\)](#), este sector representa alrededor del 16% de la fuerza laboral en Estados Unidos y era alrededor del 10% una década antes. Entre esta población, las personas mayores que viven solas son las más vulnerables, ya que cualquier caída accidental puede ser mortal ([González-Canete & Casilari, 2020](#)). En México, las caídas son la principal causa de lesiones accidentales en los adultos mayores y son la segunda causa de muerte por lesiones accidentales o no intencionales a nivel mundial ([Perez-Zepeda \*et al.\*, 2014](#)). Por esta razón, el desarrollo de tecnología para la detección de gestos y actividades humanas de la vida diaria es importante para este sector de la población en constante crecimiento, no sólo para proveer herramientas que ayuden a hacer su trabajo de una manera más eficiente y cómoda, sino también por razones de seguridad, siendo esta la aplicación principal para estas herramientas.

## 1.1 Planteamiento del problema

---

El reconocimiento de actividades físicas, comúnmente llamado Reconocimiento de Actividades Humanas (HAR) ha surgido como una área clave de investigación en la interacciones humano/computadora (HCI) así como en la computación móvil y ubicua. Uno de los fines del reconocimiento de actividades es el de proveer información del comportamiento del usuario que le permite a los sistemas computacionales asistirle proactivamente en sus tareas (Bulling *et al.*, 2014).

En términos del tipo de sensor utilizado para el monitoreo de actividades, el reconocimiento de actividades humanas se clasifica generalmente en dos categorías: reconocimiento basado en visión y reconocimiento basado en sensores. El primero es basado en el uso de dispositivos para detección visual como video cámaras, para detectar cambios en el comportamiento humano y el ambiente. La segunda categoría se basa en el uso de una red de sensores novedosos para monitoreo de actividades (L. Chen, 2012) e.g. sensores inerciales vestibles contenidos en relojes inteligentes, estos últimos han tenido un gran éxito en la comunidad de investigadores, ya que permiten realizar estudios como el presente en donde se busca superar las limitantes de los ambientes controlados y poder producir resultados más robustos.

Las aplicaciones de este tipo de tecnologías se extienden a una gran cantidad de industrias. Desde las ya mencionadas en el área de la salud y el cuidado de adultos mayores como en la industria del entretenimiento. Por ejemplo, empresas desarrolladoras de consolas para videojuegos como Nintendo, Microsoft y Sony las han utilizado para crear experiencias inmersivas con reconocimiento de actividades y gestos de los jugadores. Resulta sorprendente que estas tecnologías que originalmente pensadas únicamente con el propósito de entretener hayan sido utilizadas con propósitos científicos y por desarrolladores aficionados, como es el caso del creador de contenidos del canal de Youtube [StuffMadeHere](#) (2020), que construyó un tablero robótico de baloncesto que reconoce la velocidad y la posición de donde el jugador realiza el tiro al aro, para provocar siempre la anotación del mismo utilizando los sensores Kinect de la consola Xbox de Microsoft, entre otras aplicaciones para las mismas tecnologías.

Las industrias de la salud y bienestar físico también han aprovechado estas posibilidades. La empresa Nike creó un dispositivo de reconocimiento de actividades llamado Nike+, que consiste en un sensor que se coloca en los zapatos

## 1.1 Planteamiento del problema

deportivos de la persona que realiza la actividad. Una vez que el portador realiza la actividad física, una aplicación móvil desarrollada por la misma empresa hace la interpretación de las lecturas emitidas por el sensor y se la presenta al usuario. Otras empresas como Apple, Samsung y Fitbit ya han incorporado este tipo de funcionalidades en sus smartwatch.

En su artículo 'A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors', [Bulling \*et al.\* \(2014\)](#) muestran las principales características con las que cuenta un sistema HAR, que se incluyen en la Tabla 1.1 extraída de la misma fuente.

Tipo	Características	Descripción
Ejecución	Offline	El sistema registra los datos arrojados por los sensores primero. El reconocimiento es efectuado después. Por lo general es utilizado para aplicaciones no interactivas como el monitoreo de la salud.
	Online	El sistema obtiene los datos arrojados por los sensores y los procesa en tiempo real. Utilizados generalmente para procesamientos basados en actividades y aplicaciones interactivas humano-máquina.
Generalización	Usuario independiente Usuario específico	El sistema es optimizado para trabajar con un número grande de usuarios. El sistema está personalizado para un usuario en específico. El desempeño es generalmente mayor que un caso usuario-independiente, pero no generaliza tan bien con otros usuarios.
	Temporal	El sistema debería ser robusto a variaciones temporales causadas por condiciones externas (mal posicionamiento de los sensores, retardos en las respuestas del sensor, etc.).
Reconocimiento	Continuo	El sistema señala automáticamente la detección de las actividades o gestos en los datos que están siendo emitidos por los sensores.
	Aislado (Segmentado)	El sistema asume que los datos arrojados por los sensores han sido segmentados de inicio a fin de cada gesto previamente por un experto. Únicamente clasifica cada segmento en una de las clases definidas de actividad. El experto puede ser un sistema externo o una persona experimentador cuando se califique el desempeño de clasificación en la fase de diseño.

Continúa en la siguiente página

## 1.1 Planteamiento del problema

Tipo	Características	Descripción
Actividades	Periódicas	Las actividades o gestos a reconocer exhiben periodicidad, como caminar, correr, remar, andar en bicicleta, etc. Generalmente se utilizan técnicas de ventanas deslizantes (sliding windows) y atributos en dominio de la frecuencia para efectuar la clasificación.
	Esporádicas	Las actividades o gestos a reconocer ocurren esporádicamente en conjunto con otras actividades o gestos. La segmentación juega un papel crucial para aislar el subconjunto de datos que contienen el gesto.
	Estáticas	El sistema trabaja en la detección de posturas estáticas o gestos de señalización estáticos.
Modelo de sistema	Sin estado	El sistema de reconocimiento no modela un estado real del mundo, sino que basa el reconocimiento de actividades en detectar patrones puntuales en los datos arrojados por los sensores. Actualmente es el modelo dominante para lidiar con el reconocimiento de actividades llamadas primitivas (e.g. alcanzar un objeto, tomar un objeto).
	Con estado	El sistema utiliza un modelo del entorno, como el contexto del usuario o un mapa del entorno con localización de objetos. Esto mejora el desempeño de clasificación de actividades, a costo de mayor tiempo y conocimiento en la fase de diseño y un sistema más complejo de reconocimiento.

Tabla 1.1: Características principales de los sistemas de reconocimiento de actividades humanas

Para poder obtener un modelo que clasifique satisfactoriamente y desarrollar un sistema de este tipo, [Bulling \*et al.\* \(2014\)](#) muestran las actividades que se necesitan superar:

- Similitudes de intraclase: En la tarea de reconocer una actividad, ocurre que una misma actividad que representa una clase, es realizada de diferente manera por diferentes personas e incluso, una misma persona varía su forma de actuar según su estado de ánimo o el nivel de estrés en el que se encuentra al momento de realizar los movimientos de interés.

- Similitudes de interclases: Este problema surge al momento de realizar dos actividades diferentes que comparten los mismos movimientos. Por ejemplo, un sistema HAR en una escuela que requiera identificar cuando una persona este dibujando o escribiendo utilizando los valores arrojados por los sensores inerciales de un smartwatch, es propenso a arrojar valores bajos en reconocimiento, dado que estas dos actividades comparten muchas similitudes con respecto a otras como correr o nadar.
- Problema de clase nula: Dentro de los datos arrojados por los sensores de un sistema HAR, constantemente existen actividades registradas por los mismos que no forman parte de la investigación en curso, estas clases son consideradas nulas, y gran parte del trabajo de investigación consiste en encontrar la forma de eliminarlas conservando la máxima información posible en los datos para poder obtener un modelo de clasificación que arroje resultados satisfactorios.

Por lo anteriormente expuesto, se considera que el impacto de este trabajo puede ayudar a continuar mejorando la interacción humano / computadora, ofreciendo una herramienta más robusta para el desarrollo de más aplicaciones orientadas a relojes inteligentes, una tecnología cada vez más adoptada por la sociedad en general.

A continuación, se listarán los objetivos generales y específicos que se persiguieron en este trabajo de investigación.

## 1.2 Objetivo General

El objetivo general consiste en identificar los mejores modelos de minería de datos para el reconocimiento de cuatro movimientos humanos (subir escaleras, bajar escaleras, ponerse de pie y sentarse) utilizando datos arrojados por relojes inteligentes.

### 1.2.1 Objetivos Específicos

1. Desarrollar un software para la colección, integración y exportación de datos de sensores hacia la plataforma de minería de datos.

2. Realizar un estudio de selección de atributos para la determinación de los sensores relevantes en la detección de movimientos humanos.
3. Determinar los mejores algoritmos de reconocimiento en la clasificación de movimientos humanos.

## 1.3 Estructura del documento de tesis

Esta tesis está organizada de la siguiente manera: en el capítulo 2 se presenta el marco teórico, en donde se presentan los antecedentes encontrados en la literatura y que sirven como sustento para llevar a cabo la presente investigación. En el capítulo 3 se presenta una descripción de las metodologías utilizadas para llevar a cabo los experimentos necesarios para alcanzar los objetivos del presente documento. En el capítulo 4 se describe el desarrollo de cada una de las etapas necesarias para alcanzar los resultados presentados, desde el desarrollo del software llamado *MyBand* necesario para recolectar las muestras de los movimientos propuestos hasta la comparación de los valores arrojados por los modelos entrenados por las diferentes técnicas de machine learning utilizadas, en donde se incluyen las tasas de reconocimiento, la especificidad, sensibilidad, precisión, entre otras medidas como el área bajo las curvas ROC que describen el desempeño en la clasificación de cada uno de ellos. Finalmente, en el capítulo 5 se presentan las conclusiones definidas en base a los resultados obtenidos y se presenta una visión de las posibilidades que existen para un trabajo futuro.

# Capítulo 2

## Marco Teórico

### 2.1 Sistemas de Reconocimiento de Actividades Humanas

Existe una vasta variedad de movimientos humanos que han sido estudiados previamente utilizando sensores inerciales gracias a su tamaño portable y su bajo costo (e.g. acelerómetros, giroscopios, magnetómetros), desde actividades generales como subir escaleras, bajar escaleras, pararse, sentarse, caminar o correr (Machado *et al.*, 2015; Panwar *et al.*, 2017; Zhu *et al.*, 2018) hasta actividades de oficina como escribir manuscritos, hablar por teléfono, beber agua o trabajar en una computadora (Wang *et al.*, 2005).

Investigaciones han demostrado que el tobillo, la muñeca, la cadera y el muslo son los lugares más utilizados para la recolección de datos para el monitoreo de actividades humanas (Sheng B., 2020), aunque han habido otros experimentos en donde los sensores son colocados alrededor de un ambiente controlado (Wang *et al.*, 2005).

La cantidad de sensores puede variar, Panwar *et al.* (2017) utilizaron un solo acelerómetro tridimensional, colocado en la muñeca de las personas de prueba, en contraste con Machado *et al.* (2015), quienes utilizaron dos acelerómetros tridimensionales, uno colocado en la cadera y el otro en la muñeca. En Huynh T. (2007) se utilizaron tres sensores, uno en la muñeca derecha, otro en la cintura y otro en un muslo para poder reconocer tres actividades de larga duración (entre 40 min. y 80 min.): Prepararse para trabajar, ir de compras y hacer quehaceres

---

## 2.2 Activity Recognition Chain (ARC)

del hogar. Dividieron estas actividades en otras más pequeñas para alimentar el algoritmo de clasificación como lavarse los dientes, bañarse, sentarse, pararse, caminar, lavar platos, etc.

En conjunto con el uso de sensores inerciales, las técnicas de aprendizaje máquina y minería de datos ya han sido también utilizadas ampliamente para el reconocimiento de actividades humanas. *Anguita et al. (2012)* trabajaron en el reconocimiento de seis diferentes actividades humanas: ponerse de pie, sentarse, caminar, acostarse, subir escaleras y bajar escaleras. En su estudio, implementaron una adaptación a la técnica Support Vector Machine (SVM) para que ésta se ajustara mejor a las capacidades de procesamiento de un smartphone, cuyos sensores inerciales fueron utilizados para la captura de las muestras de movimientos y tuvieron excelentes resultados.

Según StatCounter GlobalStats en Diciembre de 2019, el 40.47% de los teléfonos móviles del mercado utiliza el sistema operativo Android, lo cual resulta conveniente para la reproducción futura de los experimentos presentados en el presente documento. Razón por la cual, la aplicación móvil desarrollada para obtener y almacenar los valores arrojados por los sensores del dispositivo utilizado para la captura de muestras se hizo orientado para este sistema operativo.

Se ha encontrado que en la mayoría de los experimentos de la literatura, los movimientos se han efectuado en ambientes controlados utilizando sensores que no se encuentran fácilmente en el mercado. El propósito del presente estudio es encontrar un modelo clasificador de cuatro actividades humanas (subir escaleras, bajar escaleras, ponerse de pie y sentarse) que pueda ser usado en cualquier dispositivo para muñeca que cuente con acelerómetro y giroscopio, que sea utilizado en los ambientes de la vida diaria del portador.

## 2.2 Activity Recognition Chain (ARC)

*Bulling et al. (2014)* presentan un Framework para el diseño y evaluación de sistemas HAR llamado Activity Recognition Chain (ARC).

## 2.2 Activity Recognition Chain (ARC)

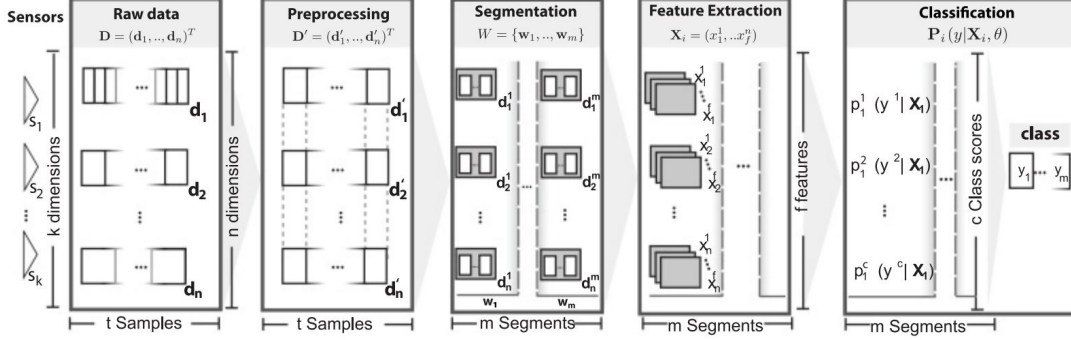


Figura 2.1: Metodología ARC

En la Figura 2.1 se muestra una cadena típica de reconocimiento de actividades (ARC) utilizando sensores de dispositivos vestibles. La metodología ARC comprime las etapas para adquisición de datos, preprocesamiento de señales y segmentación, selección y extracción de atributos, entrenamiento y clasificación. Las señales sin procesar de los sensores ( $D$ ) primero son pre-procesadas ( $D'$ ) y divididas en  $m$  segmentos ( $W_i$ ) de los cuales son extraídos los vectores de atributos ( $X_i$ ). Dados los atributos ( $X_i$ ), un modelo con parámetros  $\theta$  predice  $c$  clases con actividades  $Y_i = \{y_1, \dots, y_c\}$  con un vector de confianza  $p_i$ .

El vector  $S$  con los valores arrojados por los sensores del dispositivo vestible se define como:

$$s_i = \{d^1, d^2, d^3, \dots, d^t\}, i = 1, \dots, k \quad (2.1)$$

Donde  $k$  denota el número de sensores utilizados para reconocimiento de actividades y  $d_i$  los múltiples valores en el tiempo  $t$ .

Las tasas de muestreo de cada sensor puede variar, por lo que el vector  $s_1$  de un sensor que trabaje a una tasa de muestreo alta tendrá mayores dimensiones que un vector  $s_2$  de otro sensor cuya tasa de muestreo sea baja. La resolución de esta situación forma parte de la etapa de preprocesamiento de la metodología ARC. Esta etapa transforma la secuencia de datos crudos y no sincronizados a una secuencia preprocesada  $D'$ :

## 2.2 Activity Recognition Chain (ARC)

---

$$D' = \begin{pmatrix} d_1^1 & \dots & d_1^t \\ \dots & \dots & \dots \\ d_n^1 & \dots & d_n^t \end{pmatrix} \quad (2.2)$$

Donde  $d_i$  corresponde a una dimensión de la secuencia de datos preprocesados,  $n$  al total de dimensiones en los datos y  $t$  al número de muestras. En esta etapa también corresponden las técnicas de normalización, eliminación de outliers e imputación de datos en caso de que existan datos nulos en  $X$ .

En la etapa de segmentación se identifican aquellas porciones de datos que pueden contener información acerca de las actividades que se requieran clasificar. Esta etapa conduce a un conjunto de segmentos  $W$  que contienen una actividad potencial  $y$ :

$$W = \{w_1, w_2, w_3, \dots, w_m\} \quad (2.3)$$

La etapa de selección y extracción de atributos, reduce las señales a atributos discriminativos para las actividades de interés. Del conjunto de segmentos  $W$ , se extraen en forma de vectores de atributos  $X_i$ , definidos como:

$$X_i = F(D', W_i) \quad (2.4)$$

Donde  $F$  es la función de extracción de atributos. Conforme mayor sea el número de dimensiones en el vector  $X_i$ , mayor cantidad de datos se necesitarán para la estimación en los parámetros del modelo y mayor será el costo computacional para la clasificación.

Esta metodología se utilizó como base para los experimentos realizados en el presente estudio.

# Capítulo 3

## Materiales y Métodos

### 3.1 Dispositivo utilizado

El dispositivo inteligente elegido para recabar las muestras fue la banda para muñeca llamada Microsoft Band 2 (MSBand) de Microsoft que se muestra en la Figura 3.1a. Aunque actualmente ya se encuentra descontinuada por la empresa, no deja de ser una excelente opción para trabajos de investigación como el presente por la cantidad de sensores que posee (los cuales son especificados en la descripción técnica mostrada en la siguiente sección), la variedad de frecuencias de muestreo que ésta ofrece para sus diferentes sensores, así como por su kit de desarrollo fácil de utilizar ofrecido por la empresa (Microsoft, 2015). La descripción técnica de la banda se presenta en la Tabla 3.1.

### 3.1.1 Descripción Técnica

<b>Características Básicas</b>	
Dimensiones	12.8 x 32 x 8.7 mm
Peso	60 gramos
Batería	200 mAh Li-Po
Sistema operativo	Windows
Sensores	La descripción se encuentra en la Tabla 3.2 de la siguiente sección
<b>Procesador (SoC)</b>	
SoC	Cortex M4 MCU
Compañía	ARM
Núcleos	1x Cortex M4 a 0.2 GHz
GPU	a MHz
<b>Memoria</b>	
Memoria	- GB
Almacenamiento	- GB
<b>Pantalla</b>	
Tamaño	1.4 pulgadas
Resolución	128 x 320
Densidad de píxeles	247 ppp
Tipo	TFT
<b>Compatibilidad</b>	
iOS	sí, versión 8.1.2+
Android	sí, versión 4.4+
Windows Phone	sí, versión 8.1+
<b>Conectividad</b>	
Bluetooth	4.0 LE
Otros	GPS
<b>Actividad física</b>	Sirve para conteo de pasos, carrera, distancia, calorías, sueño, elevación, ritmo cardíaco, notificaciones, alarma inteligente

Tabla 3.1: Descripción Técnica de la Microsoft Band 2.

### 3.1 Dispositivo utilizado



(a) Microsoft Band 2

(b) Ejes de referencia.

Figura 3.1: Dispositivo utilizado.

Los inconvenientes que se encontraron de haber utilizado este dispositivo son primeramente, el hecho de estar discontinuado, complicando la adquisición y por supuesto, las actualizaciones del mismo. Después, como se puede apreciar en la Tabla 3.1, la pantalla puede resultar lo suficientemente larga para ser incómoda de portar para personas que tengan la muñeca angosta.

#### 3.1.2 Sensores

La Tabla 3.2 presenta los sensores contenidos en la MSBand, en conjunto con su descripción correspondiente y la frecuencia de muestreo a la que trabajan. Los sensores cuya frecuencia de muestreo aparece como 'Valor variable' se debe a que únicamente arrojan valores cuando ocurre algún cambio en los mismos. Los sensores cuyos valores dependen de la posición física del dispositivo, basan sus mediciones con los ejes de referencia mostrados en la Figura 3.1b.

Sensor	Detalles	Frecuencia
Acelerómetro	Arroja la aceleración X, Y y Z en unidades g. 1 $g = 9.81m/s^2$ .	62/31/8 Hz

Continúa en la siguiente página

### 3.1 Dispositivo utilizado

Sensor	Detalles	Frecuencia
Giroscopio	Arroja la velocidad angular X, Y y Z en unidades de grados por segundo ( $^{\circ}/\text{sec}$ )	62/31/8 Hz
Distancia	Arroja la distancia total en centímetros, velocidad actual en centímetros por segundo (cm/s), ritmo actual en milisegundos por metro (ms/m), y el modo de pedómetro actual (sin movimiento, caminando o corriendo).	1 Hz
Ritmo Cardíaco (HR)	Arroja el número de latidos por minuto, también indica si el sensor de HR está enlazado correctamente con el ritmo cardíaco del portador de la banda	1 Hz
Intervalo RR	Arroja el intervalo en segundos entre los últimos dos latidos continuos del corazón del portador.	Valor variable
Pedómetro	Arroja el número total de pasos que el portador de la banda ha dado desde el último reinicio de fábrica.	Valor variable
Temperatura de la piel	Arroja la temperatura actual de la piel del portador de la banda en grados Celsius.	1Hz
UV	Arroja la intensidad de exposición actual a radiación ultravioleta.	1 Hz
Contacto	Arroja el estado actual de la banda ya sea 'Banda puesta' o 'Banda no puesta'	Valor variable.
Calorías	Arroja el número total de calorías que el portador de la banda ha quemado desde el último reinicio de fábrica.	Valor variable
Respuesta Galvánica de la piel (GSR)	Arroja la resistencia en la piel del portador en kohms.	0.2/5 Hz
Luz ambiental	Arroja la intensidad actual de luz en luxes (Lumes por metro cuadrado).	2 Hz
Barómetro	Arroja el valor de la presión atmosférica en hPa y el valor de la temperatura en el ambiente en grados Celsius.	1 Hz
Altímetro	Arroja el valor actual de elevación como el total de ganancia / pérdida, pasos de ascenso / descenso , ritmo de cambio de elevación.	1 Hz

Tabla 3.2: Lista descriptiva de los sensores de la MSBand.

Los sensores de giroscopio, acelerómetro y respuesta galvánica de la piel cuenta con la posibilidad de modificar su frecuencia de muestreo con las opciones mostradas en la tabla.

## 3.2 Metodología

Con el fin de poder describir adecuadamente la metodología utilizada durante las próximas secciones, en la Tabla 3.3 se describen los roles y responsabilidades de los actores que colaboraron para llevar a cabo la presente investigación.

Actor	Roles	Responsabilidades
Marcial Sandoval Gastelum	Líder de proyecto Diseñador Programador Probador 1	Responsable del cumplimiento de cada uno de los requisitos y requerimientos funcionales del sistema, responsable del diseño de los casos uso y de prueba del software Realización de la interfaz de usuario, responsable de la usabilidad. Responsable sobre el código, de la integridad, responsable de la refactorización y simplicidad. Preparación y realización de pruebas funcionales.
Félix Fernando González	Cliente Evaluador	Proveedor de los requisitos de software Responsable de acreditar el buen funcionamiento del software así como el cumplimiento de cada uno de sus requisitos
Rodrigo Cota Soto	Probador 2	Preparación y realización de pruebas funcionales.
Josué Saúl Armenta Espinoza	Probador 3	Preparación y realización de pruebas funcionales.

Tabla 3.3: Descripción de roles involucrados

En los experimentos realizados en la presente investigación, se utilizó la metodología que se muestra en la Figura 3.2, que es basada en la metodología ARC explicada previamente en el capítulo 2.

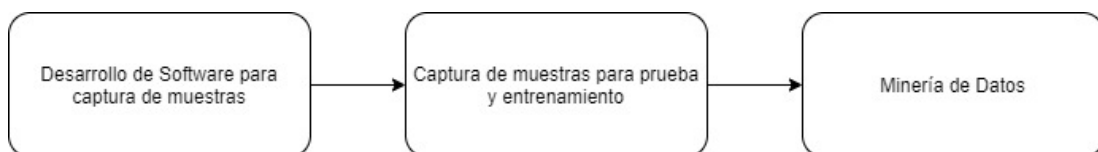


Figura 3.2: Metodología

La fase 1 correspondiente al desarrollo del software para captura de muestras, consiste en desarrollar una aplicación para sistema operativo Android con la cual se obtienen y almacenan los valores arrojados por los sensores de la MSBand. Posteriormente en la fase 2, los probadores efectúan los movimientos que se desean reconocer y se toman las muestras que servirán para entrenar los modelos de clasificación. Finalmente, en la fase 3 se utilizan diferentes técnicas de minería de datos sobre las muestras de entrenamiento, desde su preparación previa al entrenamiento, el entrenamiento mismo y la obtención de sus desempeños de reconocimiento que servirán de base para la selección de aquellos modelos con la mejor tasa de reconocimiento.

En virtud a que la metodología descrita previamente abarca dos grandes temas: desarrollo de software y minería de datos, se vio a la necesidad de utilizar una metodología independiente para cada uno ellos, correspondientes a la fase 1 y la fase 3 respectivamente, las cuales se describirán a continuación.

### 3.2.1 Metodología utilizada para el desarrollo de software para captura de muestras

Para el desarrollo del código de la aplicación móvil se utilizó el lenguaje de programación Java, siendo este el lenguaje utilizado en aplicaciones nativas del sistema operativo Android para la cual está orientada.

Algunas de las actividades son producto de la administración del proyecto mediante la metodología ágil ([Beck, 2005](#)), la cual mira el desarrollo de software como un proceso dinámico, evolutivo y orgánico en vez de estático, predefinido y mecánico como lo son otras metodologías y se promueve una iteración entre desarrollo y prueba, de esta forma se fomenta una comunicación estrecha dentro del equipo de desarrollo.

De esta manera, se muestran los artefactos generados para el análisis de requisitos, modelo de casos de uso, diseño de navegación de interfaces gráficas integrados en la [Figura 3.3](#).

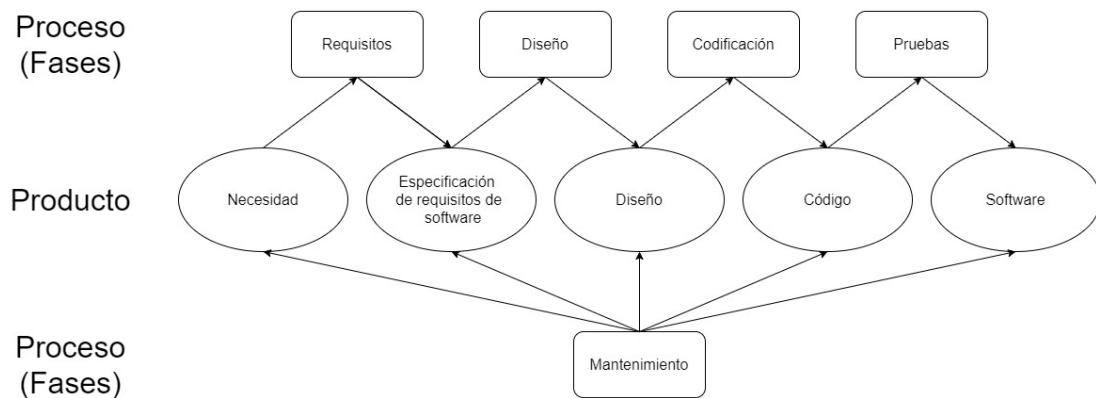


Figura 3.3: Dualidad Proceso-Producto en la Ingeniería de Software

A continuación se presenta la descripción correspondiente a cada una de las etapas de esta metodología.

1. **Requisitos.** Esta etapa tiene como finalidad la traducción satisfactoria de las necesidades del cliente para el sistema de software que se desea desarrollar a los requerimientos funcionales y no funcionales del mismo, que posteriormente serán utilizados en la etapa de diseño. Consta de cinco subprocesos: obtención de los requerimientos, el análisis, la negociación, la especificación y la validación de los mismos.
2. **Diseño.** A diferencia de la etapa de requisitos en la que se define qué debe hacer el sistema, en el diseño se decide cómo debe hacerlo. Para ello, se define la arquitectura del software y la metodología que será utilizada para construirlo. Los subprocesos de esta etapa incluyen la definición de clases y componentes como el modelo de datos, así como los elementos que se incluirán en la interfaz que utilizará el usuario.
3. **Codificación.** Según la definición brindada por el libro 'La Computación en México por especialidades académicas', esta etapa involucra actividades dependientes del lenguaje, para las cuales el responsable requiere del conocimiento de la sintáxis, la semántica y la pragmática del lenguaje de programación seleccionado para construir el producto. Tiene como principios minimizar la complejidad, anticiparse a los cambios, construir para verificar, reutilizar y utilizar estándares.

4. **Pruebas.** La etapa de pruebas tiene como propósito verificar la calidad del código y evaluar su comportamiento. Consta de cuatro subprocesos:
  - Pruebas unitarias: en las que se verifica el buen funcionamiento de cada uno de los componentes o módulos por separado.
  - Pruebas de integración: se verifica la buena comunicación entre módulos o componentes.
  - Pruebas del sistema: se revisa el desempeño del producto final como unidad.
  - Pruebas de aceptación: se realiza en conjunto con el cliente y se verifica que cada uno de los requisitos haya sido satisfecho.
  
5. **Mantenimiento.** El estándar IEEE 610.1218 define al Mantenimiento del Software como la modificación de un sistema software, o de un componente, después de que se ha entregado a los usuarios o clientes con el fin de corregir defectos, mejorar su rendimiento u otros atributos, o adaptarlo a un cambio en el entorno.

### 3.2.2 Metodología utilizada para minería de datos

Para llevar a cabo los experimentos de minería de datos y machine learning se utilizó la metodología que se muestra en la Figura 3.4.

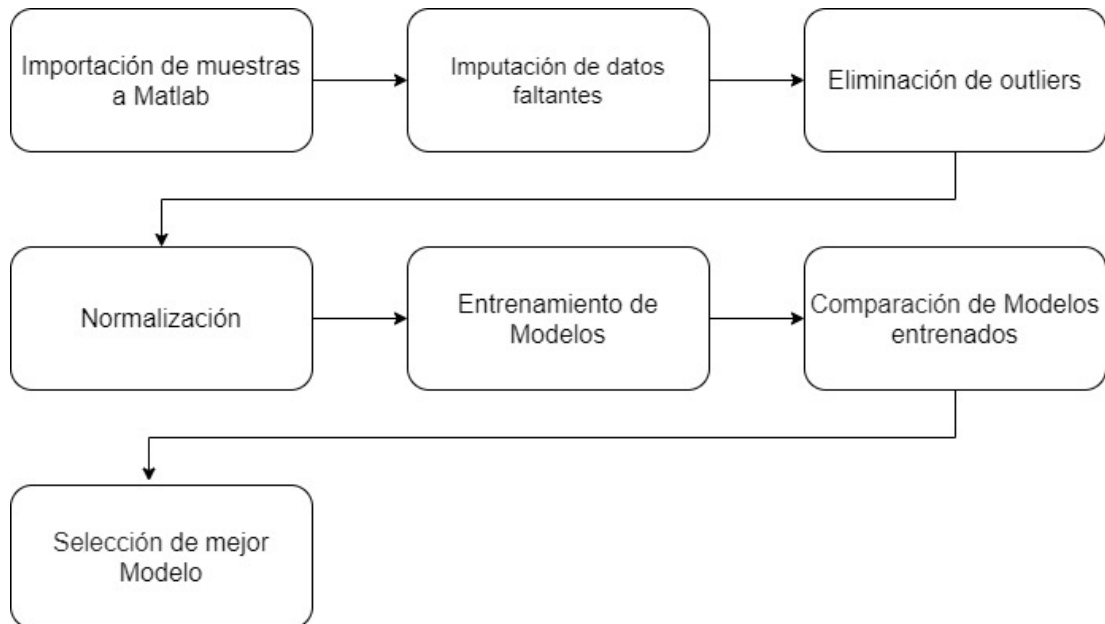


Figura 3.4: Metodología utilizada para la minería y entrenamiento de los datos

A continuación se presenta la descripción correspondiente a cada una de las fases de esta metodología.

1. **Importación de muestras a Matlab.** Se procesan los archivos tipo 'csv' con las mediciones de los sensores a un archivo tipo 'dat', tal que pueda ser manipulado fácilmente por el entorno de Matlab.

Para generar este archivo, se calculó el promedio de la cantidad de registros de cada archivo csv exportado por MyBand para determinar el tamaño del vector con cada movimiento, dando como resultado treinta registros por archivo. Considerando que cada movimiento que compone cada muestra se obtuvo utilizando el acelerómetro y giroscopio a una frecuencia de muestreo de ocho hertz durante cuatro segundos, los ciento ochenta atributos de cada muestra se componen de treinta valores arrojados por cada eje de cada uno de los sensores. Las muestras con registros faltantes se completaron con valores nulos, que más tarde fueron imputados en la Fase 'Imputación de datos faltantes' de la presente metodología.

Una vez hecho esto, se creó el dataset 'Movements.dat' que consta de una estructura de dos variables :

- Movements.X : Matriz de 180 x 360 con los valores de los atributos de cada muestra, en donde las columnas corresponden a las muestras y las filas a los atributos de cada muestra.
- Movements.Y: Vector de 1 x 360 con las etiquetas de clase de cada muestra

2. **Imputación de datos faltantes.** Se detectan datos faltantes en las muestras para después proceder a ser imputados utilizando técnicas de pre procesamiento de minería de datos.

Se utilizó la técnica de *Vecinos Cercanos* para la imputación de datos faltantes. El principio del funcionamiento de esta técnica consiste en clasificar los datos según la similitud entre el valor de los atributos en los datos. Para esta aplicación del algoritmo, la similitud se calculó en base a la distancia Euclideana (determinada por la Ecuación 3.1) de cada muestra con algún dato faltante con respecto al resto de las muestras en el conjunto datos.

$$D(X, Y) = \sqrt{\sum_{i=1}^n (X - Y_i)^2} \quad (3.1)$$

Como se mencionó anteriormente, en la Ecuación 3.1 se muestra la fórmula para calcular la distancia euclideana entre dos instancias, en donde  $X$  representa a la muestra con dato faltante  $j$ ,  $Y_i$  representa a la muestra sin datos faltantes y  $D$  representa el valor de la distancia entre ellos. Nótese que para poder calcular  $D$ , se requiere que ambas muestras tengan las mismas dimensiones, por lo que primero se extraen los atributos diferentes de  $j$  en  $X$  y estos se comparan con los mismos atributos de  $Y_i$ . Una vez realizada esta operación con cada una de las muestras sin datos faltantes, se imputa la media de los valores en los atributos  $j$  de las muestras  $Y$  cuya  $D$  sean las más pequeñas, osea, las muestras *más cercanas* a  $X$ . Para nuestro experimento, se utilizaron las tres muestras más cercanas para imputar cada dato faltante.

### 3. Eliminación de outliers.

Una vez obtenido el conjunto de datos sin datos faltantes, se procedió a retirar aquellas muestras que fueran consideradas *outliers*, que según [Hawkins \(1980\)](#) se refiere a aquellas observaciones que se desvían del resto de las observaciones tales que resultan sospechosas de haber sido generadas por un mecanismo diferente. La búsqueda de outliers se realizó con respecto a cada movimiento de cada probador.

Se utilizan técnicas de minería de datos para detectar aquellas muestras con valores atípicos para eliminarlas, de tal forma que no afecten negativamente el entrenamiento de los modelos.

Para medir el distanciamiento entre las muestras y poder eliminar los outliers, se utilizó la unidad de distancia *Mahalanobis*. La distancia Mahalanobis es una medida entre un vector  $y$  con respecto a una distribución con media  $\mu$  y covarianza  $\Sigma$ , y es representada por la Ecuación 3.2.

$$d_i = \sqrt{(y_i - \mu)\Sigma^{-1}(y_i - \mu)'} \quad (3.2)$$

El conjunto de las distancias  $d_i$  fue representado como una distribución tipo  $\chi^2$  y aquellas que rebasaran el nivel de cuantil del 97.5% de esta distribución fueron consideradas outliers.

Gracias a este pre procesamiento de datos en conjunto con el de imputación de datos, se elimina una parte de datos considerados ruidosos para los algoritmos de clasificación, que provocarían conclusiones erróneas ([Wang, 1999](#)).

4. **Normalización.** Una vez eliminados los outliers, los datos fueron estandarizados para tener cero media y varianza uno, siendo ésta la forma más usual de normalización según [Wang \(1999\)](#). Esto con el fin de eficientar el cálculo dentro de los algoritmos de entrenamiento tanto de las técnicas de minería de datos como de machine learning, se realiza una re estructuración de los valores en los datos con la mínima pérdida de información.

5. **Entrenamiento de modelos.** Se someten los datos a técnicas de clasificación de machine learning y minería de datos y se obtienen los modelos correspondientes. Los datos se entrenaron utilizando las siguientes técnicas de clasificación:

(a) *Vecinos Cercanos*

Como se explicó anteriormente, el principio del funcionamiento de esta técnica consiste en clasificar los datos según la similitud entre el valor de los atributos en los datos.

Para llevar a cabo este experimento se consideraron dos variables:

i. La medida de distancia utilizada para medir la similitud entre los datos. Sea X el conjunto de datos de entrenamiento y Y el conjunto de datos de prueba, las medidas de distancia utilizadas fueron las siguientes:

- Distancia Euclideana

$$D(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (3.3)$$

- Distancia Manhattan

$$D(X, Y) = \sum_{i=1}^n |X_i - Y_i| \quad (3.4)$$

- Distancia Chebyshev

$$D(X, Y) = \max(|X_{ik} - Y_{ik}|) \quad (3.5)$$

donde  $k$  es la dimensión  $k$  de la muestra  $i$ .

- Correlación Coseno

$$D(X, Y) = \cos\alpha = \frac{(X^T)(Y)}{\|X\| * \|Y\|} \quad (3.6)$$

- Distancia Spearman

$$D(X, Y) = \sum_{i=1}^n (X_i - Y_i)^2 \quad (3.7)$$

- ii. Cantidad de vecinos cercanos a encontrar para la clasificación en la predicción. La cantidad de vecinos cercanos se varió de distinción fina (un único vecino cercano), media (diez vecinos cercanos) y gruesa (cien vecinos cercanos).

El desarrollo de los experimentos consistió en realizar una búsqueda exhaustiva utilizando estos dos parámetros para encontrar aquella combinación que arrojará la mayor tasa de reconocimiento.

(b) *Support Vector Machine o SVM*

Las redes de vectores de soporte es una máquina de aprendizaje primordialmente para problemas de clasificación de dos clases. Conceptualmente se implementa la siguiente idea: los vectores de entrada que no son linealmente separables, se mapean a un espacio de mayor dimensionalidad en donde sí lo sean (Cortes C., 1995).

Dado que el método está orientado para resolver problemas de clasificación de únicamente dos clases, se utilizaron dos métodos diferentes para efectuar los experimentos de la presente investigación en donde se tiene un problema multiclase: el método 'uno contra uno', el cual consiste en entrenar un modelo por cada par de clases, por lo que la cantidad de modelos a entrenar será la cantidad de combinaciones posibles que se pueden formar con la cantidad de clases (para nuestro caso, se entrenaron seis clasificadores ya que estamos trabajando con cuatro clases), y el método 'uno contra todos', el cual consiste en entrenar un modelo por cada clase, los cuales discriminan si la muestra forma parte o no de la clase en cuestión.

Una vez hecho esto, se ajustaron dos parámetros para alcanzar el mejor modelo de SVM:

- i. El parámetro de tolerancia *Box constraint* 'C'. Los niveles de tolerancia  $C$  utilizados se muestran en la Ecuación 3.8.

$$C = \{1 \leq c \leq 16\} \quad (3.8)$$

Conforme mayor sea el valor para  $C$ , menor será la cantidad de vectores de soporte asignados por el algoritmo y mayor será el tiempo de entrenamiento.

- ii. El tipo kernel 'k' utilizado para el mapeo al espacio linealmente separable.

La documentación de Mathworks Matlab define la función  $G(x_j, x_k)$  como el elemento  $(j, k)$  de la matriz de Gram (espacio de atributos), donde  $x_j$  y  $x_k$  son vectores  $p$ -dimensionales que representan observaciones del conjunto de datos de entrada  $X$ . Tres diferentes kernel fueron probados en los experimentos de este algoritmo de clasificación:

A. Kernel Gaussiano o Función de base radial (RBF):

$$G(x_j, x_k) = e^{-\|x_j - x_k\|^2} \quad (3.9)$$

B. Kernel lineal:

$$G(x_j, x_k) = x_j' x_k \quad (3.10)$$

C. Kernel polinomial de tercer grado:

$$G(x_j, x_k) = (1 + x_j' x_k)^3 \quad (3.11)$$

Al igual que con los experimentos realizados para el algoritmo de vecinos cercanos, el desarrollo de los experimentos para SVM consistió en realizar una búsqueda exhaustiva utilizando estos dos parámetros para encontrar aquella combinación que arrojara la mayor tasa de reconocimiento.

- (c) Conjunto de árboles de decisión (*Tree Ensemble*). Los árboles de decisión tienen una forma muy simple de representarse, haciendo que el modelo obtenido sea relativamente fácil de entender por el usuario (Wang, 1999).

Los datos fueron entrenados por un conjunto de árboles de decisión siguiendo el algoritmo de Random Forest (Breiman, 2001). Los parámetros que se ajustaron para encontrar el mejor modelo utilizando esta técnica fueron los siguientes:

- i. La cantidad de árboles a entrenar.
- ii. La cantidad máxima de divisiones en cada árbol binario. Los valores para la cantidad máxima de árboles en el conjunto  $T$  utilizados se muestran en la Ecuación 3.12.

$$T = \{5 \leq t \leq 30\} \quad (3.12)$$

- iii. El método de decisión que varió entre bootstrap aggregation o *Bagging* y *AdaBoost*.
- (d) Red Neuronal. El conjunto de hiperparámetros ajustados para alcanzar los mejores resultados se conforma por:
- i. La cantidad de capas ocultas en la red.
  - ii. La cantidad de neuronas en cada capa oculta.
  - iii. La cantidad de iteraciones a través de los datos (*Epochs*).
  - iv. La tasa de aprendizaje utilizada durante el proceso de optimización.
- (e) DeepInsight. Se realizaron experimentos utilizando la metodología DeepInsight, la cual convierte muestras provenientes de los sensores de la MSBAnd a una matriz bien organizada con forma de imagen, en

donde cada posición dentro de la misma corresponde a un píxel en la imagen formada (Sharma, 2019).

DeepInsight utiliza el algoritmo de la herramienta *tsne* (Maaten & Hinton, 2008), con la opción de PCA a dos componentes principales para la visualización de datos de grandes dimensiones para agrupar los atributos de cada muestra que tengan mayor relación, de esta forma los atributos no se consideran variables independientes como en las técnicas de minería de datos y machine learning también utilizadas en el presente documento.

Con las muestras de los sensores de la MSBand representadas en forma de imagen, se ingresan a una red neuronal convolucional (CNN), de esta forma se aprovechan sus ventajas como la posibilidad de utilizar el GPU de la computadora para realizar los entrenamientos, y dado que los atributos con mayor relación ya se encuentran agrupados en imagen creada, dicha relación también es considerada al momento de hacer la clasificación.

Los variables de este conjunto fueron los siguientes:

- i. Dimensiones de la imagen generada.
- ii. Arquitectura de la CNN. Los parámetros modificados fueron los mismos que en el experimento realizado para obtener la arquitectura de la red neuronal explicada previamente:
  - A. La cantidad de capas ocultas en la red.
  - B. La cantidad de neuronas en cada capa.
  - C. La cantidad de iteraciones a través de los datos (*Epochs*) .

Las etapas involucradas en el proceso de la metodología de DeepInsight se muestran en la Figura 3.5, que fue tomada del artículo de Sharma (2019): 'DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture'.

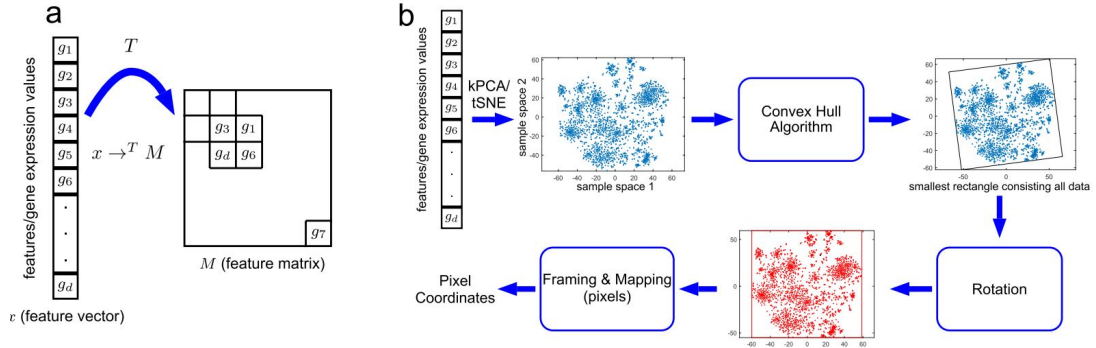


Figura 3.5: Método DeepInsight. (a) Conversión del vector de atributos a una matriz de atributos. (b) Conversión de matriz de atributos a imagen con píxeles.

6. **Comparación de Modelos encontrados.** Se someten los resultados obtenidos en la fase 5 en una comparación para determinar cuál de los modelos tiene mejor desempeño en reconocimiento de los movimientos contenidos en las muestras de entrenamiento.

Una de las herramientas utilizadas para este fin es la *Matriz de Confusión*. La matriz de confusión es una tabla en donde se incluye el conteo de cada predicción tanto positiva como negativa para cada clase realizada por el modelo clasificador. En la Fig. 3.4 se muestran los elementos que conforman una matriz de confusión.

Clase Real	Predicciones		
	0	1	$\Sigma$
0	Tn	Fp	Cn
1	Fn	Tp	Cp
$\Sigma$	Rn	Rp	N

Tabla 3.4: Matriz de Confusión

Sean:

**Fp** Positivos falsos. Aquellas predicciones que fueron catalogadas como positivas, cuando su valor real es negativo.

**Tp** Positivos verdaderos. Aquellas predicciones que fueron catalogadas como positivas, cuando su valor real es positivo.

**Fn** Negativos falsos. Aquellas predicciones que fueron catalogadas como negativas, cuando su valor real es positivo.

**Tn** Negativos verdaderos. Aquellas predicciones que fueron catalogadas como negativas, cuando su valor real es negativo.

**Cn** Cantidad de muestras negativas.

**Cp** Cantidad de muestras positivas.

**Rn** Cantidad de muestras predichas negativamente por el clasificador.

**Rp** Cantidad de muestras predichas positivamente por el clasificador.

**N** Número total de muestras clasificadas.

Utilizando la información aportada por la matriz de confusión, existen diferentes métricas para medir el desempeño de un algoritmo de clasificación, las que fueron utilizadas en el presente documento fueron las siguientes:

- Sensibilidad

$$SENS = \frac{Tp}{Cp} \quad (3.13)$$

- Especificidad

$$SPEC = \frac{Tn}{Cn} \quad (3.14)$$

- Precisión

$$PRE = \frac{Tp + Tn}{Cp + Cn} \quad (3.15)$$

- Recall

$$R = \frac{Tp}{Tp + Fn} \quad (3.16)$$

- F1 - Score

$$F1 = 2 \frac{PRE * R}{PRE + R} \quad (3.17)$$

- Matthews correlation coefficient

$$MCC = \frac{Tp(Tn) - Fp(Fn)}{\sqrt{(Tp + Fp)(Tp + Fn)(Tn + Fp)(Tn + Fn)}} \quad (3.18)$$

**Curva ROC** La curva característica del receptor operativo (Receiver Operating Characteristic o ROC) es útil para encontrar el punto de operación con las condiciones ideales para el mejor desempeño de un modelo de clasificación.

Todos los valores de las métricas de desempeño explicadas anteriormente, son válidas para un solo punto de operación, seleccionado para minimizar la probabilidad de error. Para poder evaluar más de un punto de operación y así encontrar el óptimo de operación se utiliza una función  $P(Fp)$  que este condicionada por  $P(Fn)$ , llamada método Neyman-Pearson. El punto de operación óptimo tal que minimice el coste de las clasificaciones erróneas utilizando el método Neyman-Pearson es graficando la sensibilidad contra especificidad conforme se varía la frontera de decisión. A esta gráfica se le conoce como la curva característica del receptor operativo (Receiver Operating Characteristic o ROC) (Bradley, 1997).

Dado que la curva ROC es dependiente de dos parámetros (sensibilidad y especificidad), es necesaria una sola medida que pueda determinar el desempeño general del modelo y para ello se utiliza el área bajo su curva (AUC).

Cuando el punto de operación se varía y se han obtenido el número de puntos que conforman la curva

$$P(Fp) = \alpha \quad (3.19)$$

$$P(Tp) = 1 - \beta \quad (3.20)$$

la forma más sencilla de obtener AUC es utilizar la integración trapezoidal mostrada en la ecuación 3.21 (Bradley, 1997).

$$AUC = \sum_{i=1}^n \left\{ (1 - \beta_i) \Delta\alpha + \frac{1}{2} (\Delta(1 - \beta) \Delta\alpha) \right\} \quad (3.21)$$

donde

$$\Delta(1 - \beta) \approx (1 - \beta_i) - (1 - \beta_{i-1}) \quad (3.22)$$

y

$$\Delta\alpha = \alpha_i - \alpha_{i-1} \quad (3.23)$$

7. **Selección de mejor Modelo.** Se selecciona el modelo con mayor tasa de reconocimiento para utilizarlo en las muestras de prueba.

La métrica de AUC en conjunto con la precisión y el cálculo de F1-Score se utilizaron en el presente estudio para hacer las comparaciones entre los algoritmos entrenados y hacer la selección del mejor modelo.

### 3.2.3 Métodos de Validación

Todos las técnicas de entrenamiento presentadas previamente fueron sometidas al siguiente método de validación:

**10 Fold Cross Validation** El conjunto de datos fue dividido de manera aleatoria en diez grupos sin traslape, generando diez dobleces o *folds*, asignando uno de ellos para prueba y el resto para entrenamiento (Seni & Elder, 2010). Una vez hecho esto, el resultado final presentado se refiere al valor promedio de las salidas arrojadas en la clasificación de las muestras de prueba de cada doblez.

### 3.2.4 Métodos de Visualización

En la presente investigación, se utilizaron dos métodos para visualizar los datos: PCA and t-SNE.

**PCA** Aunque el método de Análisis de Componentes Principales (PCA) fue desarrollado desde los inicios del siglo XX, ha resurgido en los últimos años gracias a sus aplicaciones en el análisis de la basta cantidad de datos que se han podido recabar gracias al desarrollo y evolución de la tecnología e internet. El PCA utiliza todas las variables originales del conjunto de datos para obtener un nuevo conjunto de datos con menos atributos o dimensiones llamados componentes principales (PC), que sirven como una aproximación a la información contenida en los datos originales. Conforme mayor sea el grado de relación entre los atributos, menor será la cantidad de nuevas variables requeridas. Cada uno de los PC no tienen relación entre sí y son ordenados de tal forma que los primeros conservan la mayor variación presente en el conjunto original de datos (Wang, 1999).

El PCA es dependiente de la escala de los datos, razón por la cual se realiza una normalización a los datos antes de someterlos al algoritmo.

**t SNE** Según van der Maaten & Hinton (2008), la técnica t-SNE (t-Distributed Stochastic Neighbor Embedding) provee una representación Y de baja dimensionalidad, que mapea y conserva la mayor estructura significativa posible del conjunto de datos original con mayores dimensiones.

# Capítulo 4

## Resultados y Discusión

En este capítulo se presentarán los resultados obtenidos de los experimentos efectuados utilizando las metodologías descritas en el capítulo anterior.

### 4.1 Fase 1: Desarrollo de Software para captura de muestras

#### 4.1.1 Requisitos de Software

Los requisitos de MyBand surgen de la necesidad de extraer los valores arrojados por los sensores inerciales de la MSBand y exportarlos a un formato que pueda ser fácilmente importado al entorno de desarrollo de software para cálculo técnico *MatLab* (Mathworks, 2020), en donde se realiza el análisis de los datos y se aplican las técnicas de clasificación de minería de datos y machine learning, actividades que formarán parte de las fases posteriores en la metodología.

Los requisitos principales de MyBand se describen a continuación:

#### **R1. Selección de sensores**

El usuario podrá elegir los sensores de los cuales se desea extraer los valores en la realización de movimientos a detectar. Las opciones posibles se mencionan a continuación:

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

- Monitor de ritmo cardíaco
- Intervalo RR
- Acelerómetro
- Altímetro
- Sensor de luz ambiental
- Barómetro
- Sensor de respuesta galvánica en la piel
- Contador de calorías
- Distancia recorrida
- Giroscopio
- Pedómetro
- Sensor de temperatura de la piel
- Sensor de UV
- Sensor de contacto

### R2. Lectura de valores de sensores

Se podrán visualizar en pantalla los valores extraídos de los sensores seleccionados.

### R3. Configuración de formato de archivo CSV a exportar

Se podrá elegir el formato deseado para el archivo CSV a exportar. Las opciones para el formato de salida se indican a continuación:

- **Frequency Based:** En este formato se exporta un archivo tipo csv independiente para cada frecuencia de muestreo a la que trabajan los sensores seleccionados para la captura de valores.
- **Time Based:** En este formato, se registran las capturas de los valores de los sensores seleccionados ordenados por su timestamp (cada lectura de sensor se registra en orden de aparición).
- **Sample Based:** En este formato se genera un solo registro en la tabla exportada en el archivo de salida tipo csv con los valores arrojados por todos los sensores seleccionados. Para lograrlo, se estandarizan las mediciones a una sola frecuencia de muestreo.

### R4. Escritura de valores de sensores en archivo CSV

Se guardarán las lecturas de los sensores localmente para posteriormente exportarlos a un archivo tipo CSV.

#### 4.1.2 Diseño

Con la finalidad de cumplir con los requisitos antes mencionados, se requirió del diseño e implementación de una base de datos que almacenara en sus registros las lecturas de los valores arrojadas por los sensores, así como las propiedades de cada sensor incluyendo: el nombre del sensor, la frecuencia de muestreo a la que trabaja y el valor capturado en la muestra.

Se eligió el modelo relacional para el diseño del modelo de bases de datos, considerando que varios sensores trabajan a la misma frecuencia de muestreo y cada registro del archivo CSV de salida puede contener la lectura de más de un sensor (según el formato elegido).

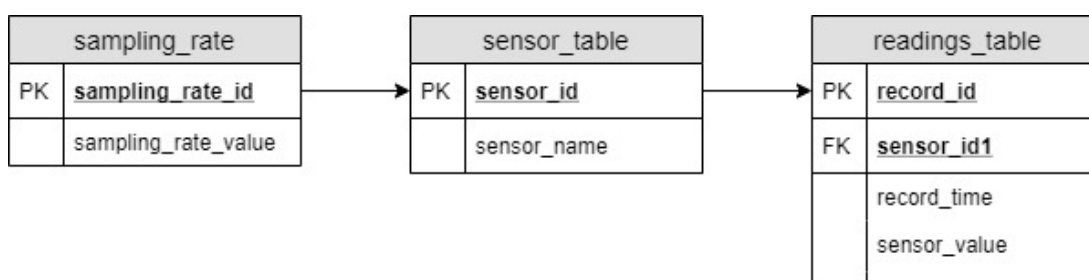


Figura 4.1: Modelo de Datos.

Como se puede apreciar en la Figura 4.1, el modelo de datos consta de tres tablas distintas, cada una de ellas orientadas a almacenar valores específicos competentes al registro de los valores arrojados por la MSBand. En la Tabla 4.1 se muestra la descripción del contenido, así como de cada campo que conforma cada una de las tablas del modelo de datos.

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

<b>Tabla: sampling_rate</b>		
<b>Descripción</b>	<b>Campos</b>	<b>Tipo</b>
Almacena la relación entre los valores de frecuencia de muestreo posibles de los sensores de la MSBand (samling_rate.value) con la clave única que lo identifica (sampling_rate.id).	sampling_rate.id samling_rate.value	int float
<b>Tabla: sensor_table</b>		
<b>Descripción</b>	<b>Campos</b>	<b>Tipo</b>
Almacena la relación entre las propiedades del sensor que incluyen su nombre (sensor.name) y frecuencia de muestreo a la que trabaja (indicado por su clave única de identificación sampling_rate.id1) con la clave única que lo identifica (sensor.id).	sensor.id sensor.name sampling_rate.id1	int string int
<b>Tabla: readings_table</b>		
<b>Descripción</b>	<b>Campos</b>	<b>Tipo</b>
Almacena la relación de las lecturas obtenidas de los sensores seleccionados con una clave única de identificación para la misma (record.id). Los datos que forman parte de cada registro de lecturas incluyen: la clave del sensor que arrojó el valor (sensor.id1), el valor arrojado por el sensor (sensor.value) y el timestamp del registro (record.time).	record.id sensor.id1 record.time sensor.value	int int long float

Tabla 4.1: Descripción de tablas.

Se buscó que las tablas no tuvieran campos tipo cadena de caracteres, para que sus valores fueran más ligeros. El único campo que se guardó en ese formato fue el campo de *sensor.name* dentro de la tabla *sensor\_table*.

El contenido de las tablas *sampling\_rate* y *sensor\_table* son constantes, dado que almacenan los datos competentes a las especificaciones de los sensores de la MSBand, y no de los valores que éstos arrojan. Dichos valores se basan en la información contenida en la Tabla 3.2 y muestran a continuación en las Tablas 4.2 y 4.3.

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

sampling_rate_id	sampling_rate_value
1	0.2
2	1
3	2
4	5
5	8
6	31
7	62
8	Value Change

Tabla 4.2: Contenido de sample\_rate

sensor_id	sensor_name	sampling_rate_id1
1	HEAR_RATE_SENSOR	2
2	RR_INTERVAL_SENSOR	8
3	ALTIMETER_SENSOR	2
4	AMBIENT_LIGHT_SENSOR	3
5	BAROMETER_SENSOR	2
6	CALORIES_SENSOR	8
7	DISTANCE_SENSOR	2
8	PEDOMETER_SENSOR	8
9	SKIN_TEMP_SENSOR	2
10	UV_LEVEL_SENSOR	2
11	BAND_CONTACT_SENSOR	8
12	ACCELEROMETER_SENSOR	7
13	ACCELEROMETER_SENSOR	6
14	ACCELEROMETER_SENSOR	5
15	GYROSCOPE_SENSOR	7
16	GYROSCOPE_SENSOR	6
17	GYROSCOPE_SENSOR	5
18	GSR_SENSOR	1
19	GSR_SENSOR	4

Tabla 4.3: Contenido de sensor\_table

El registro de las muestras de interés para los experimentos del presente documento se almacenarán localmente en el dispositivo smartphone dentro de la tabla readings\_table.

Para llevar acabo R4 se utilizó el gestor de base de datos utilizado en Android: *SQLite*. SQLite es un gestor de base de datos SQL transaccional sin servidor, que ya ha sido ampliamente utilizado en aplicaciones para dispositivos móviles con sistema operativo Android, en donde es usado como almacenamiento de bases

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

de datos tanto del usuario como del sistema, guardando información como los contactos, SMS, etc (Je-Min & J.W, 2012). Aprovechando estas ventajas y la fácil disponibilidad que aporta el sistema operativo, se utilizó en MyBand para almacenar los registros con los valores arrojados por los sensores de la MSBand, esto con la finalidad de generar el archivo CSV y a su vez, tener un respaldo de todas las muestras.

Dependiendo de la configuración del formato de salida, el modelo de la tabla con las muestras exportada es diferente.

La descripción detallada de los atributos de cada registro dentro de las tablas exportadas en formato Frequency Based, Time Based y Sample Based se presentan en el Anexo C del presente documento.

En las próximas subsecciones se describen las necesidades del sistema. El conjunto de requerimientos definen los aspectos funcionales y no funcionales. Los requerimientos no funcionales son aquellos que brindan soporte a las operaciones que el usuario desea que el sistema realice y permitirán identificar que la aplicación debe ser escalable y fácil de usar. Más adelante en la subsección de Requerimientos Funcionales, se presentan 22 requerimientos en la Tabla 4.5 con un identificador (id) para su trazabilidad con los escenarios de uso mostrados en el Anexo C del presente documento.

### 4.1.3 Requerimientos no funcionales

En la Tabla 4.4 se presentan los requerimientos no funcionales para MyBand.

Id	Descripción
REQN1	El smartphone debe trabajar con sistema operativo Android versión 4.1 (API 16 o <i>Jelly Bean</i> ) o superior
REQN2	El smartphone debe contar con un mínimo de 13MB para funcionar.
REQN3	Contar con la app Microsoft Band
REQN4	Utilizar dispositivo Microsoft Band 2
REQN5	Utilizar como mecanismo de comunicación sincronización bluetooth
REQN6	El smartphone debe contar con conexión a internet

Tabla 4.4: Requerimientos no funcionales

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

### 4.1.4 Requerimientos funcionales

En la Tabla 4.5 se presentan los requerimientos funcionales de MyBand app. Para esta sección se seleccionó el escenario de uso del requerimiento funcional *REQ19* para mostrar la pantalla principal de MyBand funcionando correctamente. Para poder ver el resto de los escenarios, refiérase al Anexo C del presente documento.

Id	Descripción	Escenario
REQ1	Cargar la banda Microsoft band 2	
REQ2	Configurar la hora en la banda	
REQ3	En el celular, entrar a Google Play Store	Fig. C.1a
REQ4	Descargar la App Microsoft Band	Fig. C.1b, Fig. C.1c
REQ5	Abrir la App Microsoft Band	Fig. C.2a
REQ6	Seleccionar opción iniciar sesión	
REQ7	Se requiere iniciar sesión con una cuenta de Microsoft	C.2b
REQ8	Sino se cuenta crear una en <a href="https://signup.live.com/signup">https://signup.live.com/signup</a>	
REQ9	Si se cuenta con una, iniciar sesión con cuenta existente	
REQ10	Poner la MSBand en modo de emparejamiento ( <i>Bluetooth</i> )	Fig. C.2d, Fig. C.3a
REQ11	Encender el Bluetooth del celular	Fig. C.3b
REQ12	Vincular la MSBand con el celular vía Bluetooth	Fig. C.3c
REQ13	Comprobar la comunicación entre MSBand y el celular: seleccionar la opción <i>My Microsoft band</i> ubicada en el menú de Microsoft Band App	Fig. C.4b
REQ14	¿Aparecen los detalles de la MSBand en pantalla?	
REQ15	Si es así, la comunicación con la banda ha sido exitosa	
REQ16	sino revisar el problema de comunicación y repetir pasos anteriores	
REQ17	En el celular, permitir la instalación de aplicaciones proveniente de fuentes desconocidas	Fig. C.4c
REQ18	Instalar la App MyBand	Fig. C.5a
REQ19	Comprobar la comunicación de la banda con la App MyBand seleccionando la opción <i>Start</i> hasta que el estado de la aplicación (indicada en la barra superior) cambie de <i>Select option</i> a <i>Band is connected</i>	Fig. 4.2
REQ20	¿El botón de start cambia a rojo?	
REQ21	si cambia, la banda está conectada	

Continúa en la siguiente página

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

Id	Descripción	Escenario
REQ22	sino revisar el problema de comunicación y repetir pasos anteriores	

Tabla 4.5: Requerimientos Funcionales.

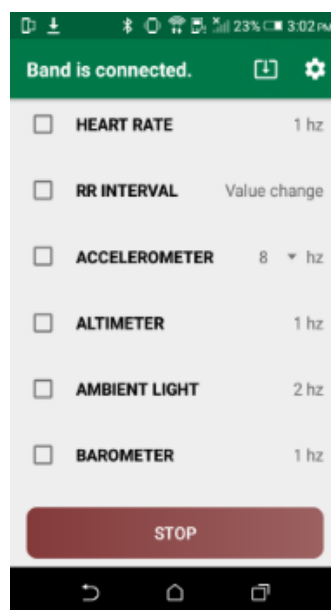


Figura 4.2: Comprobación de comunicación con la MSBand

### 4.1.5 Casos de uso

Siguiendo un proceso de desarrollo de software unificado, se proponen los casos de uso necesarios para capturar los requerimientos (Véase la Figura 4.3). Esta técnica ayudó a definir cuáles son los actores del sistema (usuario) y representar la funcionalidad.

El usuario de la aplicación tiene la opción de únicamente visualizar las lecturas de los sensores seleccionados o de registrar dichas lecturas en un archivo para ser exportado.

En el caso que se requiera de modificar la etiqueta, el nombre predeterminado para los archivos de salida contienen un timestamp con el siguiente formato : 'YYYYMMdd.HHmms' seguido por la extensión '.csv' (por ejemplo, un archivo

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

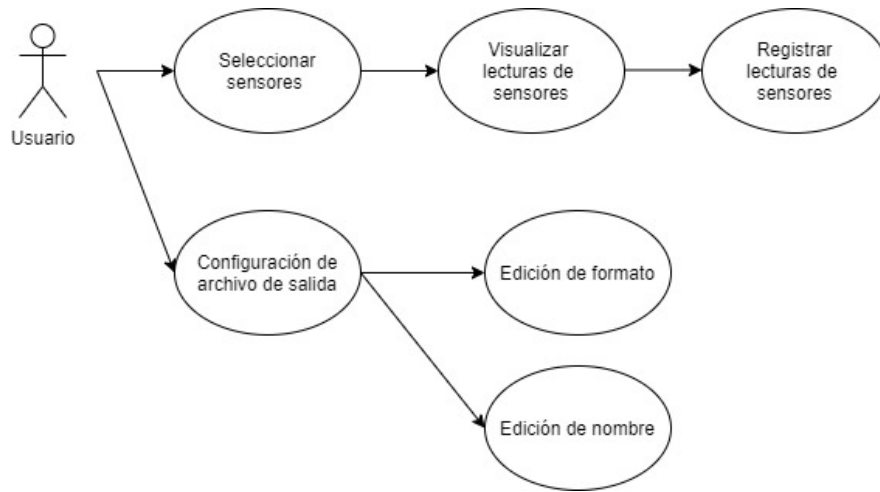


Figura 4.3: Diagrama de caso de uso de MyBand

exportado en la fecha 10 de agosto del 2020 a las 17 hrs 22 min 14 segundos, se le asignará la etiqueta predeterminada "20200810\_172217.csv").

### 4.1.6 Casos de prueba

Con la finalidad de verificar el buen funcionamiento en la implementación de los requerimientos funcionales y siguiendo el modelo presentado en el Diagrama de caso de uso en la Figura 4.3, se diseñaron dos tipos de pruebas: positivas (Flujos básicos) y negativas (Flujos alternativos), las primeras demuestran el flujo exitoso de la funcionalidad y las segundas lo contrario.

#### Flujos básicos

Los flujos básicos se refieren a aquellos flujos de funcionamiento del sistema de software tales que resultaron exitosos sin alguna complicación. El caso de uso inicia cuando el usuario desea realizar alguna de las siguientes funciones:

- Si desea editar el nombre del archivo csv a exportar, se ejecuta el subflujo FB1 (Tabla C.1).
- Si desea editar el formato del archivo csv a exportar, se ejecuta el subflujo FB2 (Tabla C.2).

## 4.1 Fase 1: Desarrollo de Software para captura de muestras

---

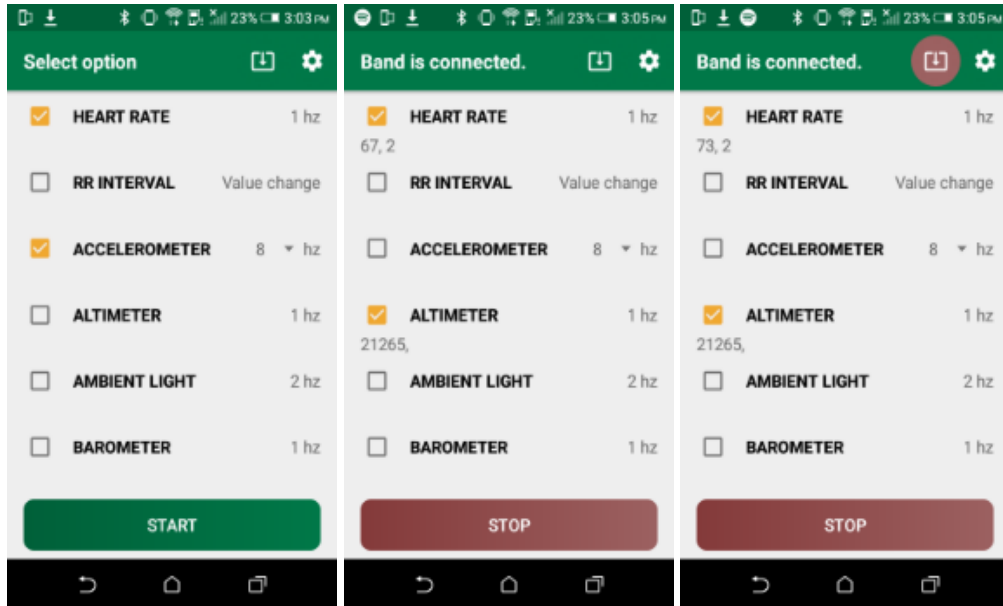
- Si desea visualizar las lecturas de los sensores, se ejecuta el subflujo FB3 (Tabla C.3).
- Si desea registrar las lecturas de los sensores, se ejecuta el subflujo FB4 (Tabla 4.6).

De manera demostrativa de la definición de los flujos básicos, se muestra el Flujo Básico FB4, que se sigue para registrar las lecturas de los sensores. Si se desean conocer los detalles correspondientes al resto de los subflujos mencionados anteriormente puede referirse al Anexo C.

Id	Descripción	Escenario
FB4.1	Colocar la MSBand en muñeca izquierda, asegurándose de posicionar la pantalla en la parte interna	Fig. 4.7
FB4.2	Elegir los sensores de interés de la pantalla principal de My band en los <i>checkbox</i> correspondientes	
FB4.3	¿Se visualizan en amarillo los checkbox de los sensores seleccionados?	
FB4.4	Si no, existe un problema técnico reportar a correo: marcial.sandoval@uabc.edu.mx	
FB4.5	Si, todo es correcto	Fig. 4.4a
FB4.6	Leer los valores arrojados por los sensores elegidos seleccionando la opción <i>Start</i>	
FB4.7	Después de 5 segundos, ¿Las lecturas de los sensores aparecieron abajo del nombre del sensor ?	
FB4.8	Si no, existe un problema técnico reportar a correo: marcial.sandoval@uabc.edu.mx	
FB4.9	Si, todo es correcto	Fig. 4.4b
FB4.10	Presionar botón <i>Save</i> , ubicado en la barra superior, a izquierda del menú de configuraciones señalado con un engrane	
FB4.11	¿La opción de resalta de rojo?	
FB4.12	Si no, existe un problema técnico reportar a correo: marcial.sandoval@uabc.edu.mx	
FB4.13	Si, todo es correcto	Fig. 4.4c
FB4.14	Cuando se requiera dejar de grabar las lecturas, presionar botón <i>Stop</i>	
FB4.15	Visualizar mensaje de confirmación de escritura de archivo, en donde se muestra el path de ubicación del mismo	Fig. 4.5a

Tabla 4.6: Subflujo FB4

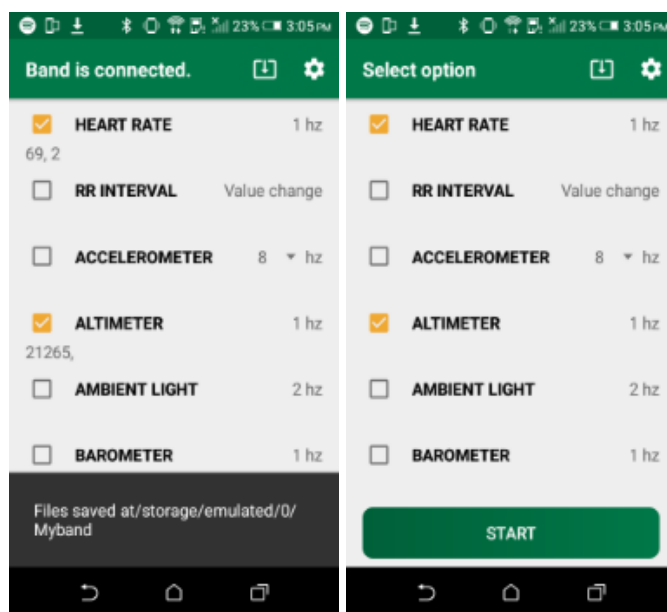
## 4.1 Fase 1: Desarrollo de Software para captura de muestras



(a) Selección de sensores a utilizar para muestras  
(b) Lectura de valores arrojados por los sensores seleccionados  
(c) Registro de valores arrojados por los sensores seleccionados

Figura 4.4: Escenarios de uso relacionados FB4.1 al FB4.13

## 4.1 Fase 1: Desarrollo de Software para captura de muestras



(a) Confirmación de registro de lecturas de los sensores seleccionados  
(b) Actividad principal después de haber presionado el botón *Start*

Figura 4.5: Escenarios de uso relacionados con FB4.15

Los casos de prueba correspondientes a los subflujos positivos de MyBand, se muestran y describen con detalle en el Anexo C.

A continuación se presentarán los flujos alternativos.

### Flujos alternativos

Los flujos alternativos surgen a partir de un incumplimiento de alguno de los requerimientos funcionales descritos en la Tabla 4.5, o bien, por la falta de alguno de los requerimientos no funcionales mostrados en la Tabla 4.4. Todos los escenarios correspondientes a los flujos que se describirán a continuación pueden encontrarse en el Anexo C.

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

---

La lista con los flujos alternativos es la siguiente:

1. Si no hay comunicación bluetooth entre el smartphone y la MSBand, MyBand mostrará el mensaje 'Band isn't paired with your phone' ('Banda no emparejada con su teléfono') en la barra de tareas como se muestra en la Figura [C.11a](#).
2. Si no hay comunicación entre la MSBand y la aplicación Microsoft Band, MyBand mostrará el mensaje 'Band is not bound to MS Health band communication service' ('Banda no enlazada con el servicio de comunicación de MS Health') en la barra de tareas como se muestra en la Figura [C.11b](#).
3. Si se pierde la comunicación entre la MSBand y MyBand durante el registro de valores de los sensores seleccionados, MyBand lo interpretará como si el usuario hubiera detenido el registro y continuará a partir del paso FB4.15 en adelante del Subflujo FB4.
4. Si se pierde la comunicación entre la MSBand y MyBand fuera del registro de valores de los sensores seleccionados, MyBand mostrará el mensaje 'Band is not bound to MS Health band communication service' en la barra de tareas como se muestra en la Figura [C.11b](#).
5. Si el usuario presiona *Save* en el menú de configuraciones sin haber modificado ningún campo, MyBand mostrará el mensaje 'No changes made' en un *Toast* como se muestra en la Figura [C.11c](#).
6. Si el usuario cancela la actividad de configuraciones, MyBand mostrará el mensaje 'Changes cancelled' en un *Toast* como se muestra en la Figura [C.12](#).

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

En la Figura [4.6](#) se muestra el Diagrama de Despliegue que muestra los detalles de los experimentos llevados a cabo en la presente investigación.

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

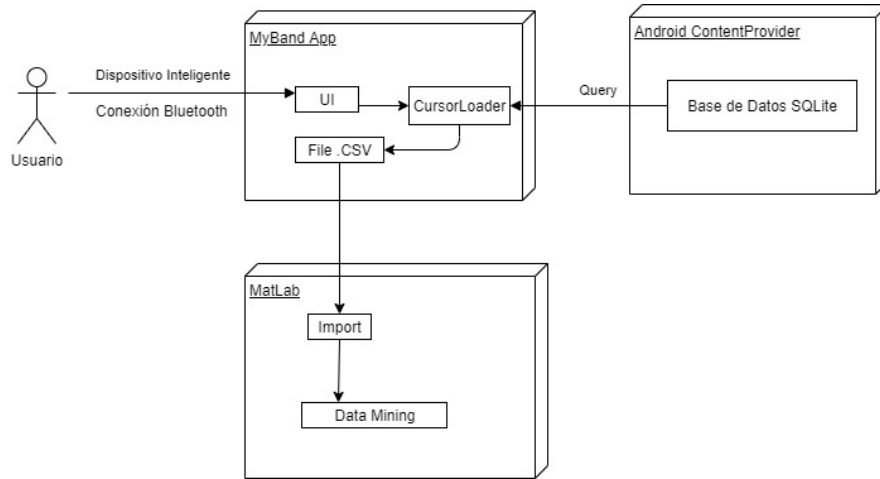


Figura 4.6: Diagrama de Despliegue

La persona que realiza los movimientos (el *Probador*) viste la MSBand en su muñeca izquierda como se muestra en la Figura 4.7, la cual tiene una comunicación por Bluetooth con MyBand. El Probador utiliza la aplicación para registrar los valores arrojados por el acelerómetro y el giroscopio de la MSBand a través de la aplicación MyBand, siguiendo el diagrama de caso de uso mostrado en la Figura 4.3. Al finalizar cada movimiento que constituye una muestra, MyBand arroja un archivo tipo CSV (Comma Separated Values). Una vez hecho esto, el archivo CSV es importado al entorno de desarrollo de software para cálculo técnico *MatLab* para su análisis utilizando técnicas de minería de datos y machine learning que serán descritas en la sección 3.2.3.

La etiqueta de los cuatro tipos de movimiento a clasificar se enumeran a continuación:

1. Subir escaleras (*UP*)
2. Bajar escaleras (*DWN*)
3. Sentarse (*SIT*)
4. Pararse (*STD*)

Una muestra de ellos se puede apreciar en la Figura 4.8.

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

---



Figura 4.7: Posición de la MSBand en portador.



(a) UP

(b) DWN

(c) SIT

(d) STD

Figura 4.8: Muestras de los movimientos a reconocer

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

---

Las personas con el rol de Probador mencionados en la Tabla 3.3 tienen las siguientes características: estudiantes hombres con edades de 28 y 29 años, peso entre 60 y 80 kgs y estatura entre 1.68 mts a 1.78 mts. Cada Probador realizó cada uno de los tipo de movimiento 30 veces durante 4 segundos, creando con conjunto de datos con 360 muestras.

### 4.2.1 Descripción de los datos

El acelerómetro y giroscopio fueron los sensores elegidos para obtener las muestras por trabajar a la misma frecuencia de muestreo, lo cual es conveniente para el minado de los datos y además pueden ser encontrados en otros dispositivos móviles para muñeca comerciales, esto facilita recrear los experimentos en un futuro. La frecuencia de muestreo a la que trabajan ambos sensores es lo suficientemente alta para alcanzar una tasa de reconocimiento aceptable. El resto de los sensores fueron descartados por no aportar información en las muestras, de entre ellos, se consideró utilizar el altímetro pero la velocidad de respuesta en su transductor era demasiado baja para poder generar un cambio en sus valores dentro de la ventana de tiempo de cuatro segundos considerada para cada movimiento, lo mismo sucedió con el sensor de ritmo cardíaco, también considerado al inicio de la presente investigación.

Sea  $X$  el conjunto de valores que conforman muestras capturadas por los sensores inerciales de la MSBand. Cada elemento en  $X$ ,  $x^i$  se define como:

$$x^i = [accX, accY, accZ, gyrX, gyrY, gyrZ] \quad (4.1)$$

Donde:

- $accX$ : Valor arrojado por el acelerómetro con respecto al eje X mostrado en la Figura 3.1b.
- $accY$ : Valor arrojado por el acelerómetro con respecto al eje Y mostrado en la Figura 3.1b.
- $accZ$ : Valor arrojado por el acelerómetro con respecto al eje Z mostrado en la Figura 3.1b.

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

- gyrX: Valor arrojado por el giroscopio con respecto al eje X mostrado en la Figura 3.1b.
- gyrY: Valor arrojado por el giroscopio con respecto al eje Y mostrado en la Figura 3.1b.
- gyrZ: Valor arrojado por el giroscopio con respecto al eje Z mostrado en la Figura 3.1b.

En las gráficas presentadas en las Figuras 4.9 y 4.10, se visualizan las mediciones de cada uno de los ejes de los sensores correspondientes a cada movimiento realizado por el Probador 1.

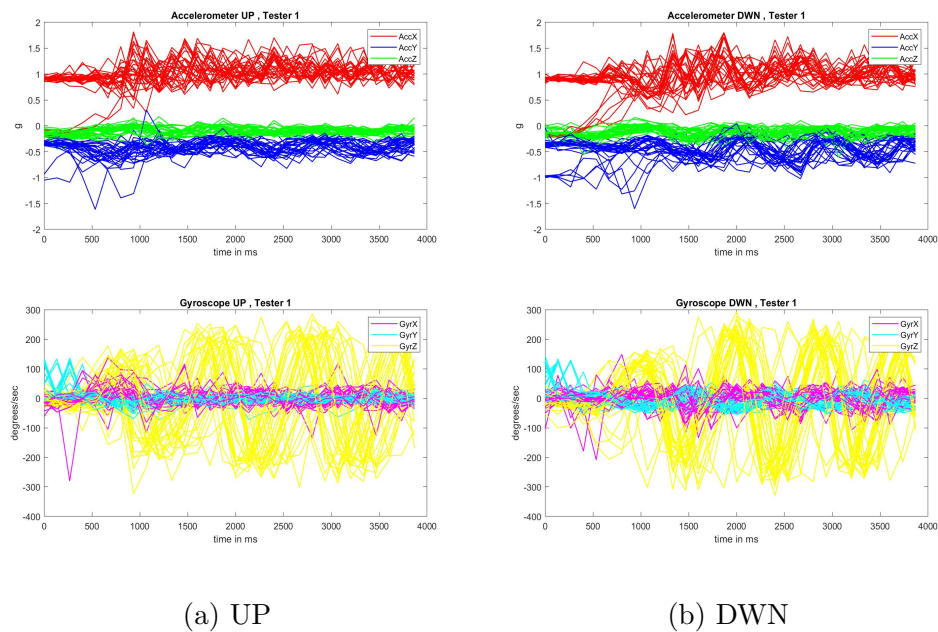


Figura 4.9: Movimientos UP y DWN realizados por Probador 1

Como se puede apreciar en las gráficas de los movimientos de la Figura 4.9, los valores arrojados por el eje Z del giroscopio es el que presenta un patrón de comportamiento periódico fácil de apreciar a simple vista, lo que significa que el Probador 1 sube y baja las escaleras con movimientos consistentes de sus brazos.

## 4.2 Fase 2: Captura de muestras para prueba y entrenamiento

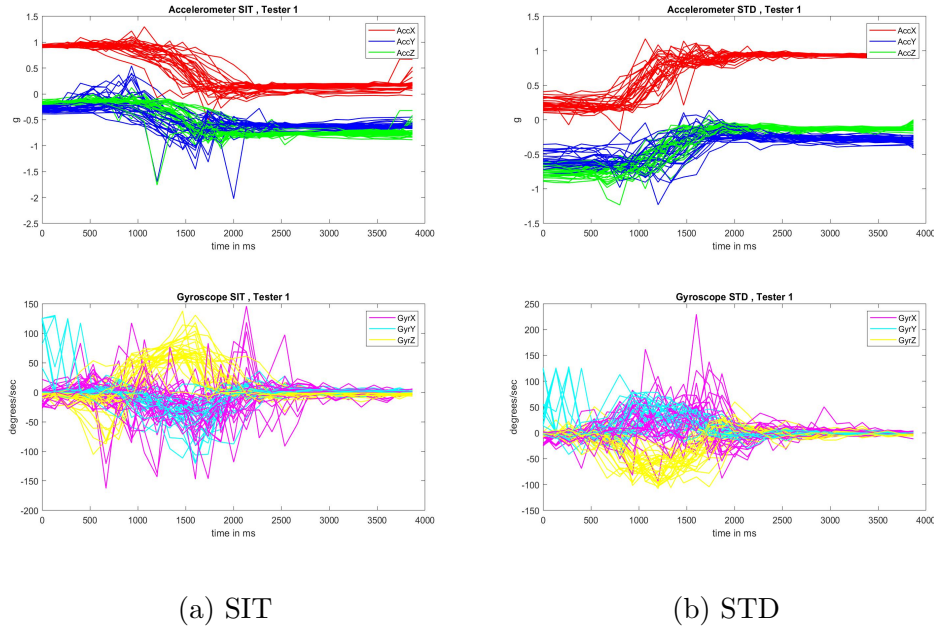


Figura 4.10: Movimientos SIT y STD realizados por Probador 1

A diferencia de los movimientos de UP y DWN de este probador, en el caso de los movimientos de SIT y STD resultó ser el acelerómetro el sensor que aporta más información por el claro patrón de comportamiento en los datos durante los movimientos. En el caso del giroscopio, el eje z presenta un patrón fácil de apreciar, dado que efectuaba de manera consistente los movimientos de muñeca al apoyarse del descansabrazos de la silla para ponerse de pie o sentarse en la misma.

Las gráficas de los movimientos efectuados por los probadores 2 y 3, se pueden apreciar en las Figuras A.1, A.2, A.3 y A.4, dentro del Anexo A del presente documento.

Cada muestra de movimiento es representada en forma de tabla, como se explicó en la sección 4.1.2. Dado que cada muestra contenía diferente cantidad de registros, se estandarizaron todas al valor promedio de renglones encontrado en todo el conjunto de tablas, que fue de 30 registros. En caso de que una

muestra tuviera más de 30 registros, se eliminaron los registros de diferencia (los últimos registrados). En caso de que una muestra tuviera menos de 30 registros, se agregaron renglones con valores nulos, que en las etapas siguientes fueron imputados.

Una vez hecho esto, se transformó cada matriz de 30 registros de 6 atributos ( $accX$ ,  $accY$ ,  $accZ$ ,  $gyrX$ ,  $gyrY$ ,  $gyrZ$ ) a renglones, dando como resultado la representación final de la muestra en un renglón de 180 atributos.

Al conjunto de datos de 360 muestras de 180 atributos se le denominó *Movement Dataset*.

### 4.3 Fase 3: Minado de Datos

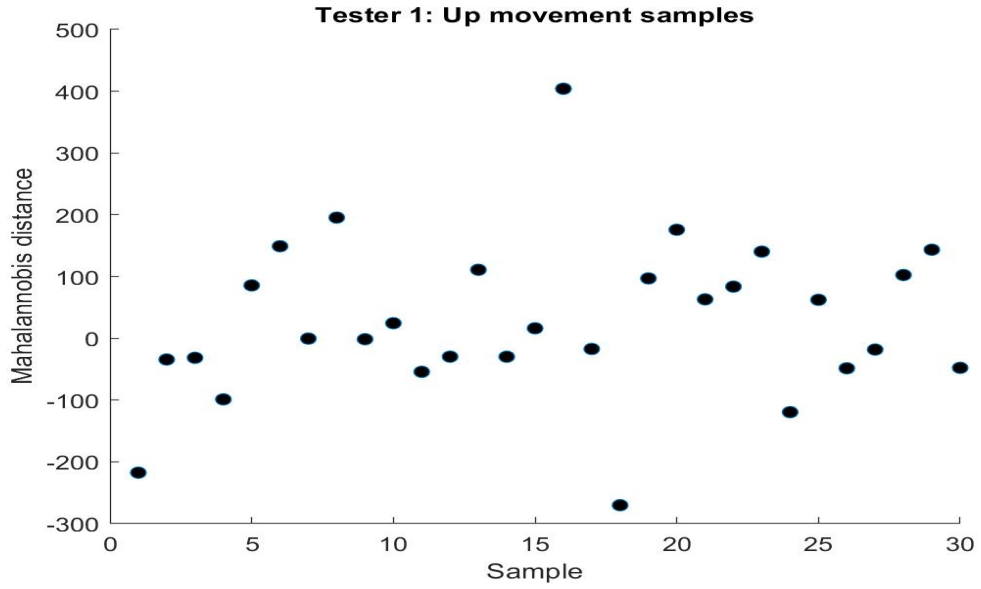
La minería de datos se define como el proceso de descubrimiento de patrones en un conjunto de datos (Witten *et al.*, 2011), generalmente este conjunto de datos suele ser de dimensionalidades altas que hacen que el descubrimiento de conocimiento en los mismos sea una tarea muy ardua sino imposible de analizar y comprender para una persona por más experiencia que esta tenga. Por esta razón, diferentes técnicas tanto automáticas como semiautomáticas se han desarrollado para ayudar en las investigaciones y búsqueda de conocimiento.

Las técnicas de minería de datos resultan propicias para el análisis de esta investigación, dada la dimensionalidad del conjunto de datos adquirido a partir de los sensores de la MSBand descrito previamente.

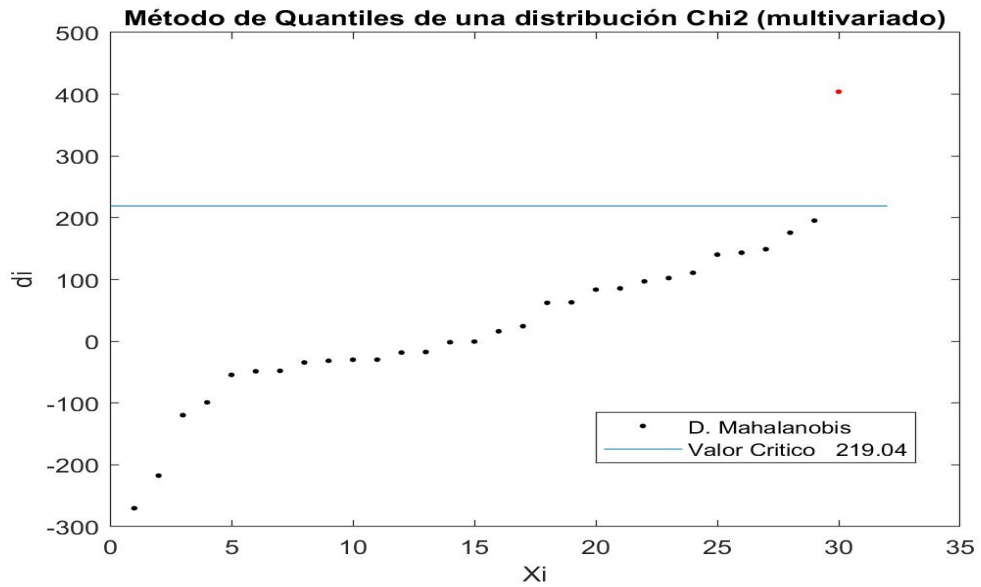
#### 4.3.1 Preprocesamiento

##### Detección de Outliers

En la Figura 4.11a, se muestran graficadas las distancias Mahalannobis correspondientes a cada uno de las muestras UP efectuadas por el Probador 1. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudo detectar un sólo outlier para este subconjunto de datos (Véase la Figura 4.11b).



(a) Distancias



(b) Outliers

Figura 4.11: Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 1.

En la Figura 4.12 se muestra el outlier detectado por el Método de Cuantiles de una distribución Chi cuadrada indicado con una línea roja.

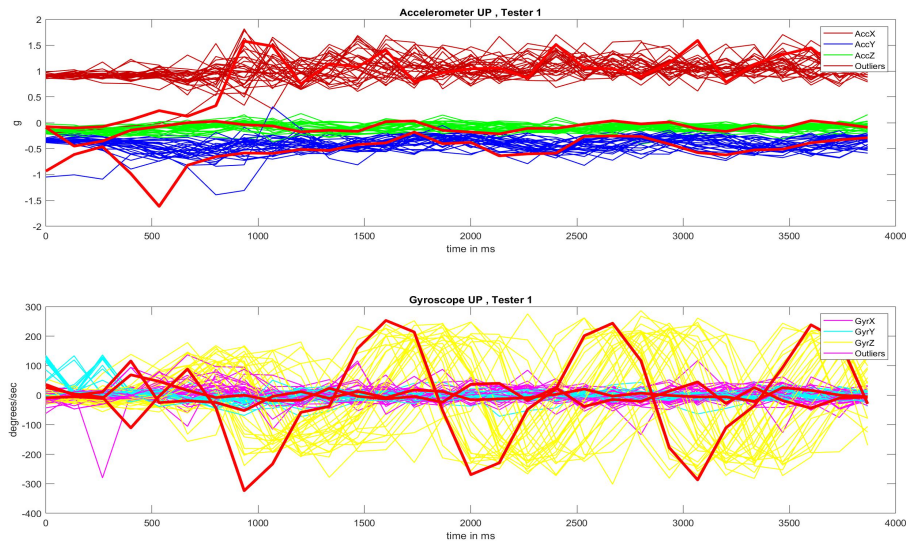


Figura 4.12: Outlier detectado para el movimiento UP efectuado por el Probador 1

Siguiendo este mismo proceso, se efectuó la detección de outliers para cada uno de los movimientos efectuados por cada uno de los probadores. Las gráficas correspondientes a ese análisis se muestran en el Anexo B.

Para nuestro experimento, se detectaron un total de 57 outliers, reduciendo las dimensiones de Movements dataset de 360 muestras a 303, representando una reducción de un 15.83%.

### Normalización

El resultado de los procesos de imputación de datos, eliminación de outliers y normalización descritos en la sección 3.2.2, se presenta en las gráficas mostradas en las Figuras 4.9, 4.10, A.1, A.2, A.3 y A.4.

### División Train / Test / Validation

Una vez habiendo construido el conjunto de datos normalizado, libre de outliers y de valores nulos, el conjunto de datos fue subdividido usando la proporción 90:10 para entrenamiento/prueba y validación respectivamente, es decir, de nuestro conjunto de datos con 303 muestras, se utilizaron 272 muestras para entrenamiento/prueba y 31 muestras para validación.

### 4.3.2 Algoritmos de Clasificación

Como se explicó en la sección 3.2.3, cada uno de los resultados obtenidos de cada técnica de clasificación presentadas a continuación en la fase 3, fue validado utilizando la metodología de 10 Fold Cross Validation. El conjunto de muestras de entrenamiento/prueba de *Movement Dataset* fue dividido de manera aleatoria en diez grupos sin traslape, generando diez dobleces o *folds*, asignando uno de ellos para prueba y el resto para entrenamiento. Se realizó esta partición aleatoria 10 veces, eligiendo una diferente distribución de muestras por cada iteración. El resultado presentado como final en cada algoritmo de clasificación se refiere al promedio de la salida de cada una de estas 10 iteraciones.

Se eligió el conjunto de parámetros tal que arrojara la mayor tasa de reconocimiento para cada algoritmo de clasificación para ser finalmente comparado contra los demás.

### Vecinos Cercanos

Después de realizar una búsqueda exhaustiva considerando cada uno de los parámetros explicados en la sección 3.2.2, en esta técnica se determinó que utilizando la medida de distancia Spearman y una distinción fina se alcanzó una tasa de reconocimiento de 89%. La matriz de confusión presentada en la Tabla 4.7 se obtuvo probando el modelo entrenado con esas características en el subconjunto de muestras asignadas para validación.

En la Figura 4.13 se muestran los resultados obtenidos durante el proceso de entrenamiento. Se puede ver claramente que independientemente de la medida de distancia utilizada para calcular la cercanía entre las instancias, conforme se iba aumentando la cantidad de vecinos cercanos utilizados para calcular la inferencia,

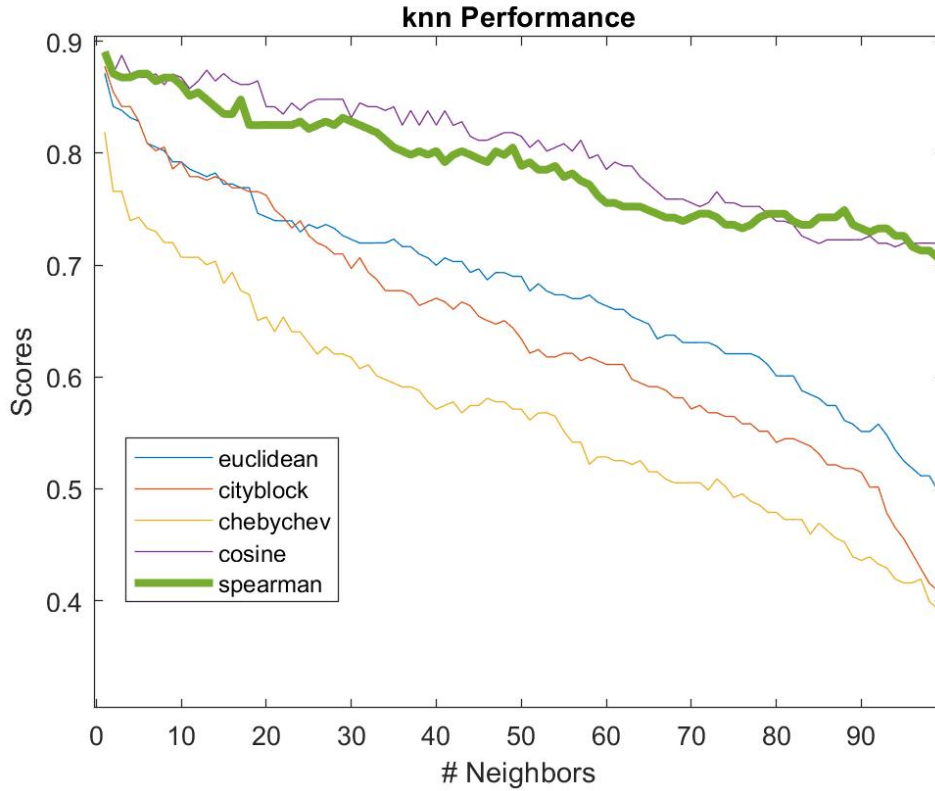


Figura 4.13: Gráfica de desempeño de KNN

la tasa de reconocimiento iba disminuyendo, razón por la cual, se decidió utilizar una distinción fina (un solo vecino cercano) en el modelo final. La medida de distancia con mejores resultados se muestra en un color verde.

Clase Real	Predicciones			
	UP	DWN	SIT	STD
UP	11	1	0	0
DWN	1	3	0	1
SIT	0	0	4	0
STD	0	0	0	9

Tabla 4.7: Matriz de Confusión KNN

### 4.3 Fase 3: Minado de Datos

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	0.9167	0.9444	0.9167	0.9167	0.8611	0.9306
DWN	0.6	0.96	0.75	0.6667	0.6139	0.78
SIT	1	1	1	1	1	1
STD	1	0.9524	0.9	0.9474	0.9258	0.9762

Tabla 4.8: Medidas de desempeño para KNN

La curva ROC del modelo obtenido de Vecinos Cercanos utilizando los parámetros mencionados previamente se muestra en la Figura 4.14.

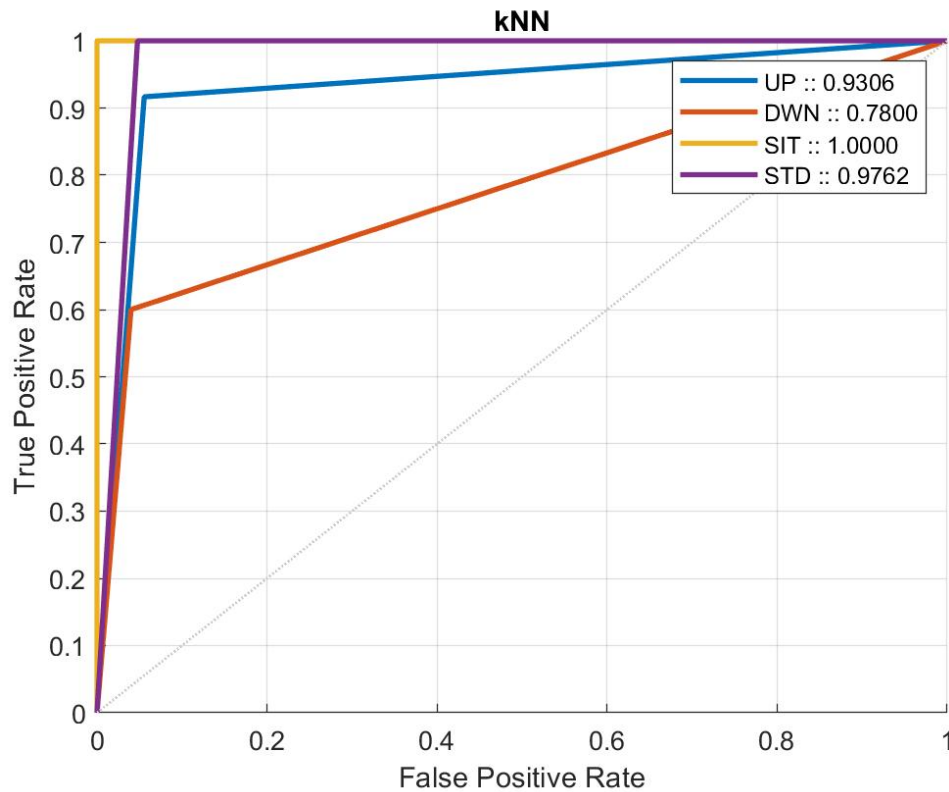


Figura 4.14: Curva ROC para Vecinos Cercanos

Se puede apreciar en la Tabla 4.8 que este modelo clasificador resulta muy preciso para distinguir las clases SIT y STD.

En la gráfica se puede observar el desempeño obtenido para cada una de las clases. La clase DWN fue la que arrojó mayores errores de clasificación, generando

un AUC de 0.78, mientras que en el caso de la clase SIT, tuvo un valor de 1.

#### **Support Vector Machine o SVM**

Se pudo observar que la tasa de reconocimiento no variaba a partir de un nivel de tolerancia  $C$  igual a diez, pero los vectores de soporte encontrados si lo hacían hasta un nivel de dieciséis, después de este valor, el algoritmo no determinaba valores distintos, razón por la cual se determinó diez como el nivel límite superior del conjunto de valores propuestos para el experimento.

Después de realizar los experimentos se determinó que se obtenían mejores resultados utilizando el método 'Uno contra uno'.

La tasa de reconocimiento máxima fue de 89.45% utilizando el kernel lineal con los niveles de  $C$  igual a 2.6 y 2.8, se eligió  $C = 2.6$  como la mejor opción. La matriz de confusión presentada en la Tabla 4.9 se obtuvo probando el modelo entrenado con esas características en el subconjunto de muestras asignadas para validación.

En la Figura 4.13 se muestran los resultados del desempeño del modelo obtenido durante el proceso de entrenamiento.

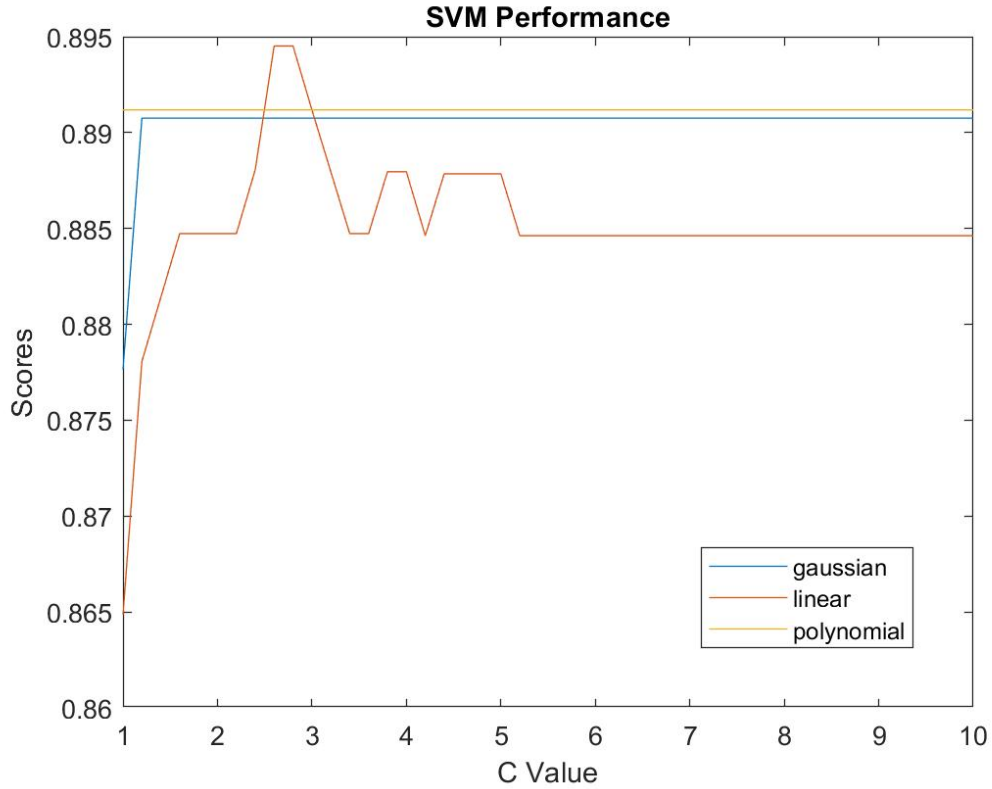


Figura 4.15: Gráfica de desempeño de SVM

Como se muestra en la gráfica de desempeño en la Figura 4.15, el valor asignado para  $C$  no tuvo un impacto considerable en la mayoría de los experimentos. Únicamente los experimentos realizados utilizando el kernel lineal muestran algún cambio significativo en los resultados, alcanzando un pico máximo para valores de 2.6 y 2.8.

Clase Real	Predicciones			
	UP	DWN	SIT	STD
UP	5	0	0	0
DWN	2	4	0	1
SIT	0	0	8	0
STD	0	0	0	11

Tabla 4.9: Matriz de Confusión SVM

La curva ROC del modelo obtenido de SVM utilizando los parámetros men-

cionados previamente se muestra en la Figura 4.16.

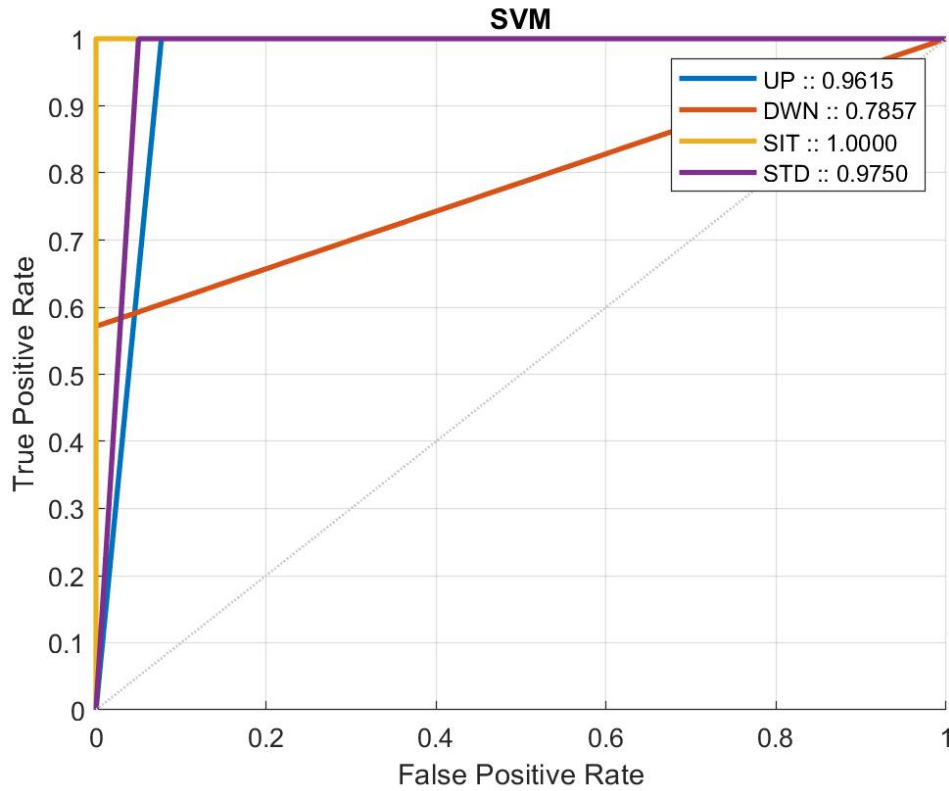


Figura 4.16: Curva ROC para SVM

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	1	0.9231	0.7143	0.8333	0.812	0.9615
DWN	0.5714	1	1	0.7273	0.7127	0.7857
SIT	1	1	1	1	1	1
STD	1	0.95	0.9167	0.9565	0.9332	0.975

Tabla 4.10: Medidas de desempeño para SVM

Este modelo clasificador resultó tener deficiencias al momento de reconocer el movimiento de DWN como lo muestran tanto la Tabla 4.10 con las medidas de desempeño y la Figura 4.16.

### Tree Ensemble

Inicialmente se entrenó un solo árbol de decisión variando únicamente la división máxima, alcanzando una tasa de reconocimiento de 80.6% a 20 divisiones máximas.

Con la finalidad de presentar un diagrama árbol de decisión que fuera más fácil de interpretar, cada uno de los 180 atributos fueron nombrados según el eje del sensor al que correspondían siguiendo el orden descrito en la sección 4.2.1 y agregando índice con el que fue se registrado al momento de realizar el movimiento, es decir:

accX0, accY0, accZ0, gyrX0, gyrY0, gyrZ0, ... , accX30, accX30, accX30, gyrX30, gyrY30, gyrZ30.

En la Figura 4.17 se muestra el Diagrama de árbol de decisión resultante.

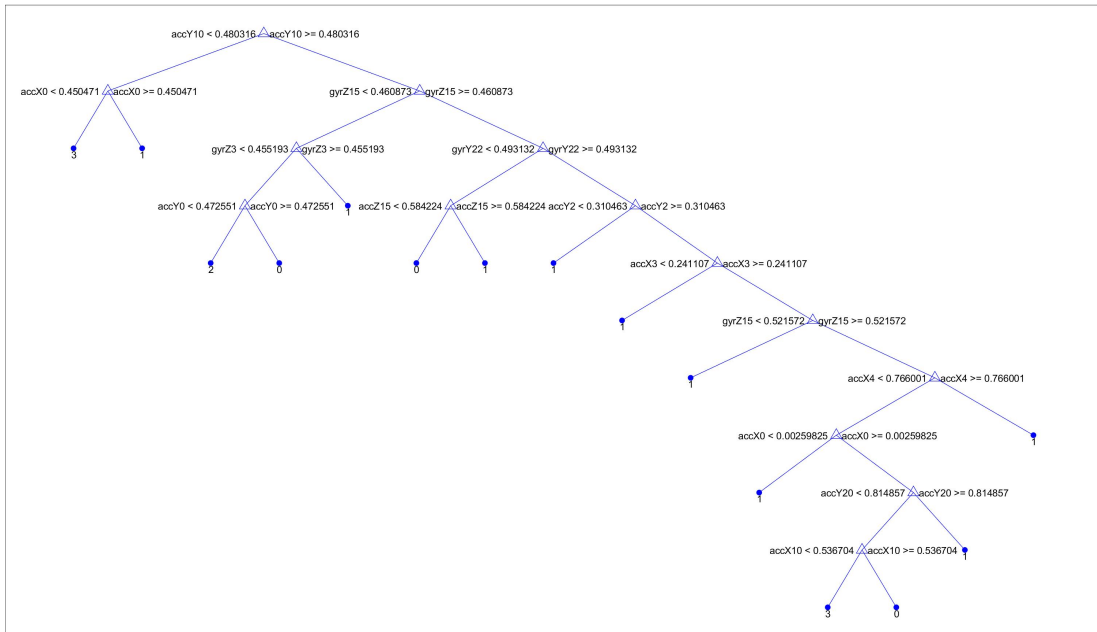


Figura 4.17: Diagrama de Árbol de Decisión

La matriz de confusión presentada en la Tabla 4.11 se obtuvo probando el modelo entrenado con esas características en el subconjunto de muestras asignadas

para validación. En la Figura 4.18 se puede apreciar la curva de desempeño de este modelo.

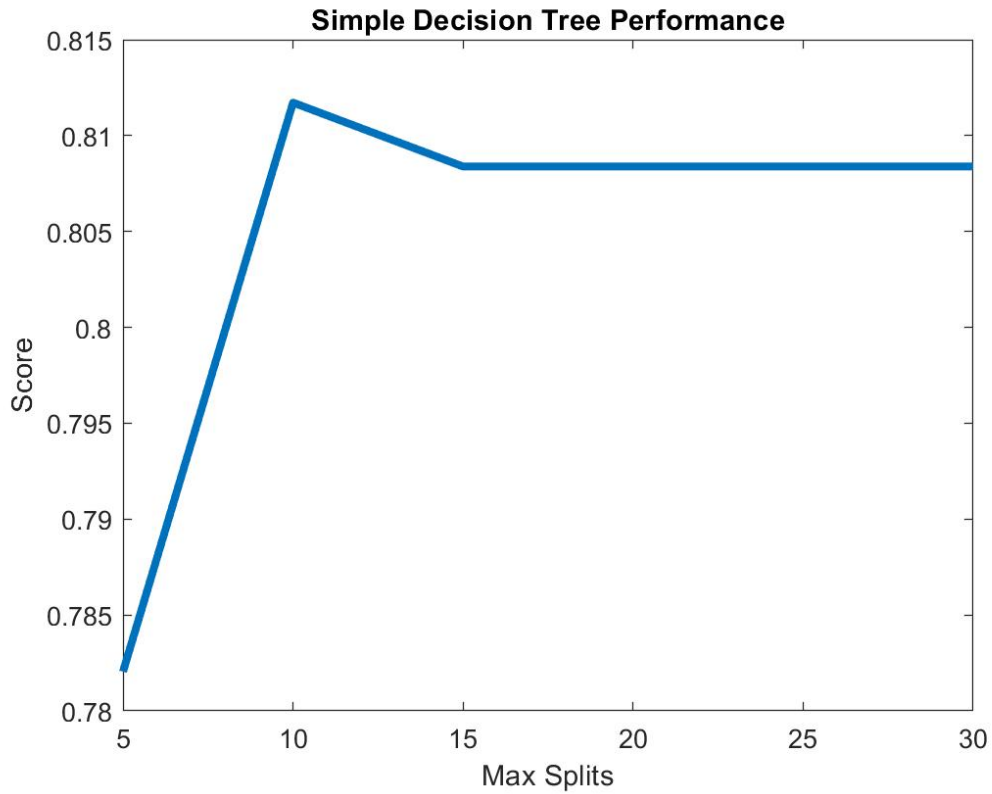


Figura 4.18: Gráfica de desempeño del Árbol de Decisión simple

Se puede apreciar en la Figura 4.18 que habiendo dividido el árbol de decisión quince veces, éste alcanzó la máxima cantidad posible permitida por los datos, razón por la cual ya no se obtuvieron resultados diferentes despues de alcanzar ese valor.

Clase Real	Predicciones			
	UP	DWN	SIT	STD
UP	6	4	0	0
DWN	1	6	0	0
SIT	0	0	7	1
STD	0	0	0	6

Tabla 4.11: Matriz de Confusión Árbol de Decisión simple

Utilizando la matriz de confusión presentada en la Tabla 4.11 se calcularon las métricas de especificidad y sensibilidad para obtener la curva ROC del modelo obtenido que se muestra en la Figura 4.19.

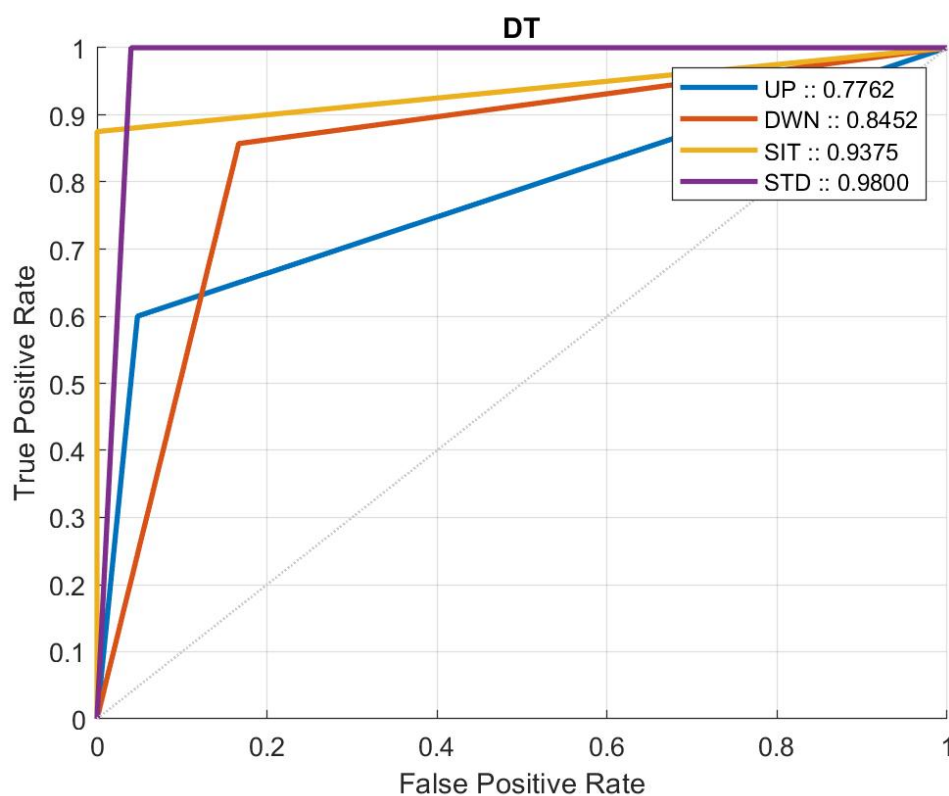


Figura 4.19: Curva ROC para Árbol de Decisión simple

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	0.6	0.9524	0.8571	0.7059	0.6176	0.7762
DWN	0.8571	0.8333	0.6	0.7059	0.6176	0.8452
SIT	0.875	1	1	0.9333	0.9157	0.9375
STD	1	0.96	0.8571	0.9231	0.9071	0.98

Tabla 4.12: Medidas de desempeño para Árbol de Decisión simple

Con la finalidad de aumentar la tasa de reconocimiento, se procedió al entrenamiento de un conjunto de árboles de decisión variando la división máxima como

se hizo en el experimento del árbol simple y además utilizando el dos diferentes métodos: *Bagging* y *Boosting*. Con estos parámetros de entrada, se realizó una búsqueda exhaustiva para encontrar la cantidad máxima de divisiones de la raíz de cada árbol de decisión, que por conveniencia, se realizó hasta 30 divisiones, siendo 25 la cantidad con mejores resultados. Se determinó que el método de *Bagging* era el mejor como se puede apreciar en la gráfica de desempeño mostrada en la Figura 4.20. Con estos valores se logró alcanzar una tasa de reconocimiento de 90%. La matriz de confusión presentada en la Tabla 4.13 se obtuvo probando el modelo entrenado con esas características en el subconjunto de muestras asignadas para validación.

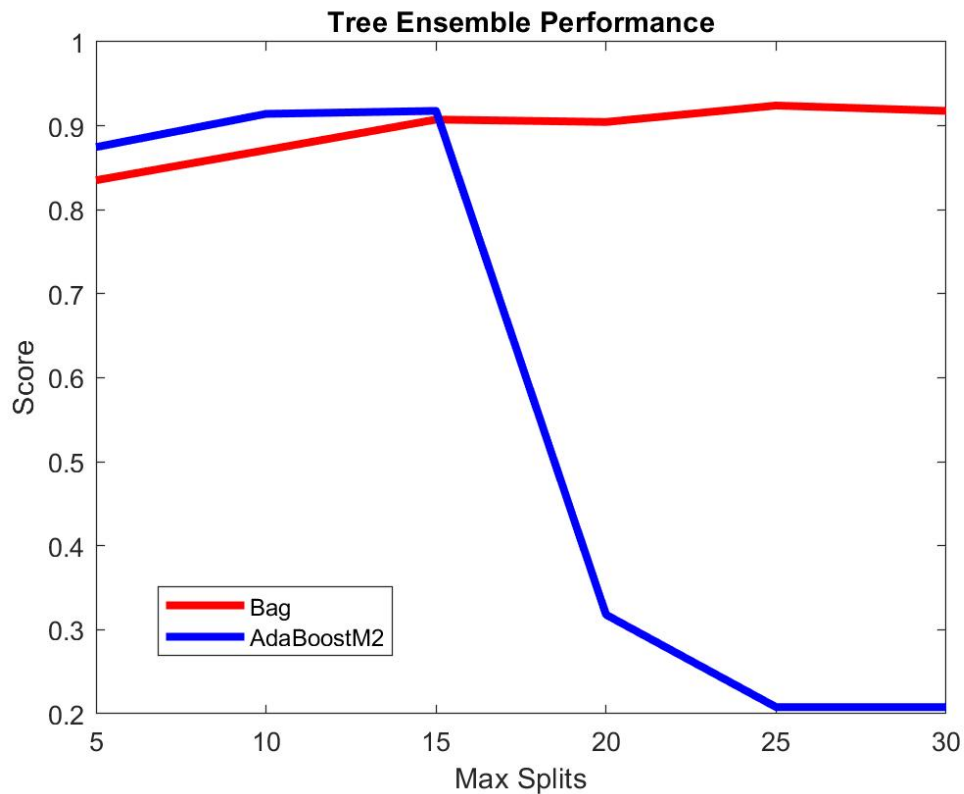


Figura 4.20: Gráfica de desempeño del Conjunto de Árboles de Decisión

Clase Real	Predicciones			
	UP	DWN	SIT	STD
UP	6	0	0	0
DWN	2	4	0	0
SIT	0	0	6	0
STD	1	0	0	11

Tabla 4.13: Matriz de Confusión del Conjunto de Árboles de Decisión

La curva ROC del modelo obtenido del conjunto de árboles de decisión utilizando los parámetros mencionados previamente se muestra en la Figura 4.21.

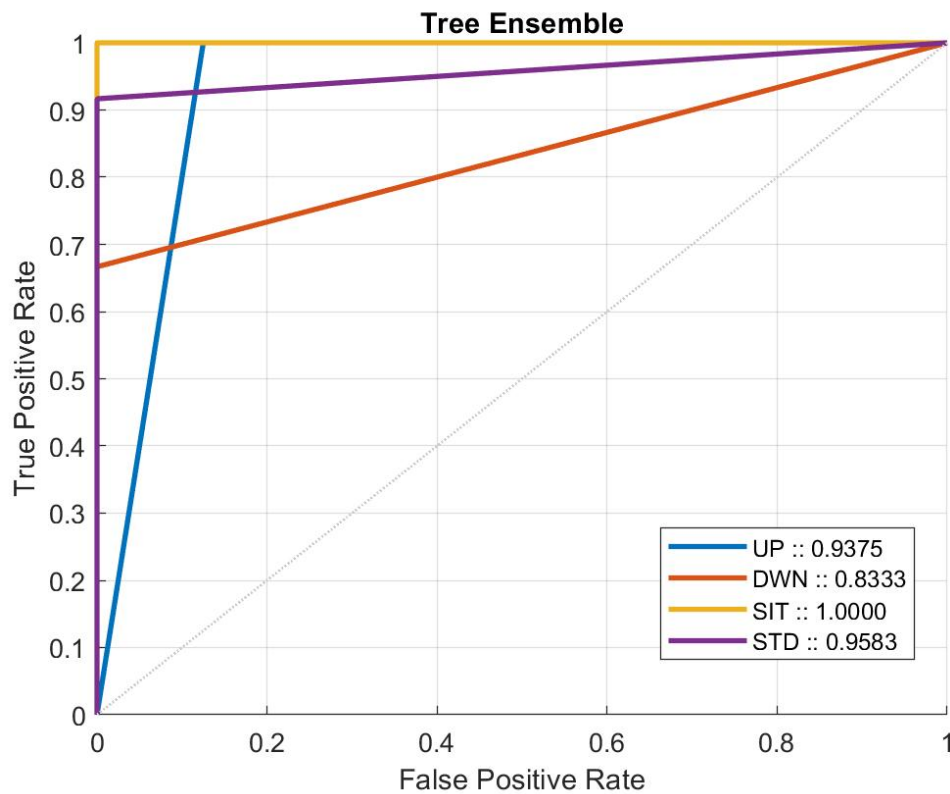


Figura 4.21: Curva ROC para Conjunto de Árboles de Decisión

Para el caso del modelo del conjunto de árboles de decisión, la clase SIT fue la que tuvo una mayor cantidad de predicciones erróneas con un AUC de 1. Por

su parte la clase DWN tuvo un AUC de 0.83, fue la que tuvo un desempeño más bajo.

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	1	0.875	0.6667	0.8	0.7638	0.9375
DWN	0.6667	1	1	0.8	0.7845	0.8333
SIT	1	1	1	1	1	1
STD	0.9167	1	1	0.9565	0.9319	0.9583

Tabla 4.14: Medidas de desempeño para Conjunto de Árboles de Decisión

### Red Neuronal

La arquitectura final utilizada en la red neuronal es mostrada en la Figura 4.22 con una sola capa oculta y una capa de salida. La cantidad de neuronas de la capa oculta (Hidden1) fue de diez unidades. Los valores de los pesos  $W$  y los valores de ajuste  $b$  fueron entrenados utilizando la función de pérdida de gradiente descendente. Ajustando la cantidad máxima de de epochs a 1000 se alcanzó una tasa de reconocimiento del 96.4% .

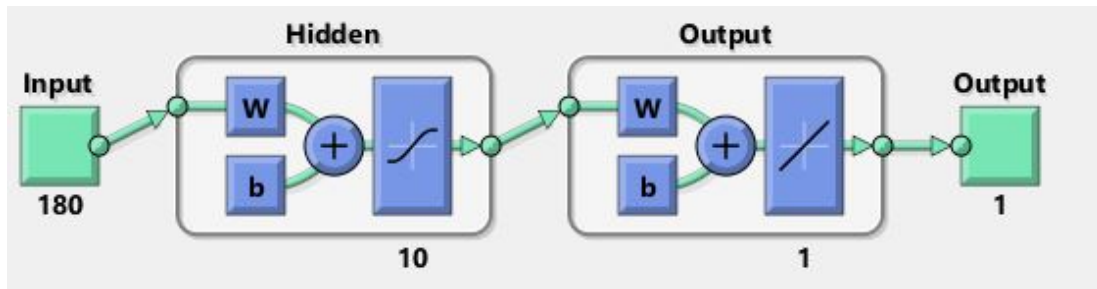


Figura 4.22: Arquitectura de la Red Neuronal

En la Figura 4.23 se muestra el desempeño del modelo durante su entrenamiento. Se puede ver que el modelo alcanzó el punto máximo de aprendizaje a partir de la epoch cien, cuando los valores del error sobre las muestras de prueba (mostrados en color rojo) se estabilizaron en su punto mínimo.

La curva ROC del modelo obtenido de la red neuronal utilizando los parámetros mencionados previamente se muestra en la Figura 4.24.

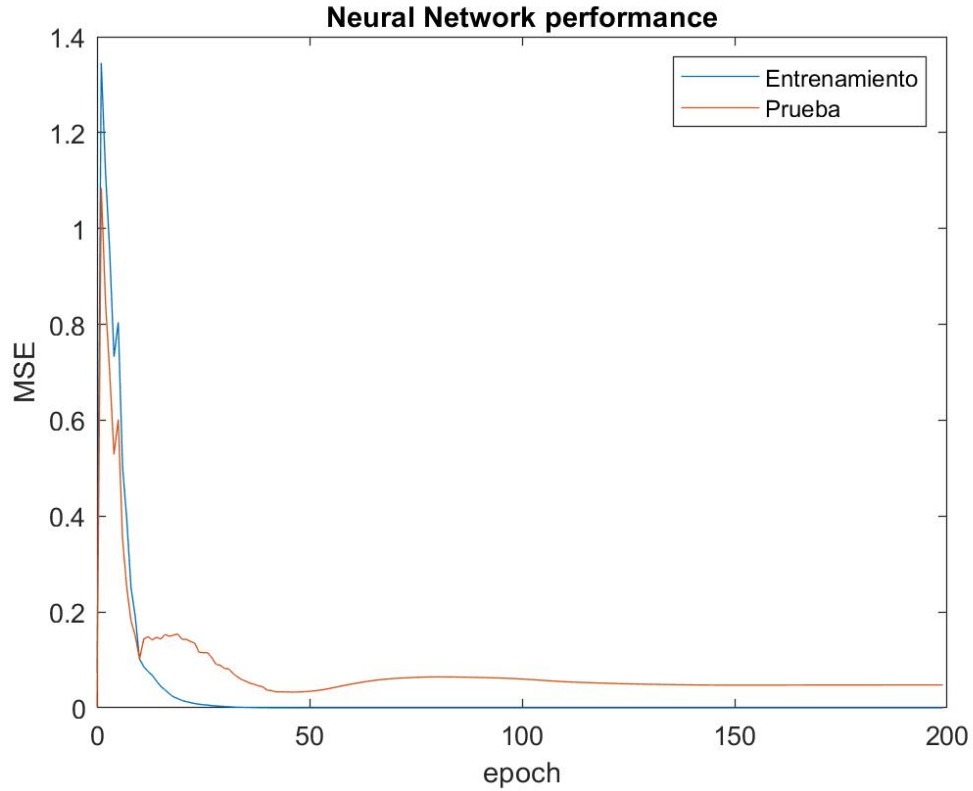


Figura 4.23: Desempeño de la red neuronal durante su entrenamiento.

En este caso, hubieron dos clases con una AUC con valor de 1, tanto la clase SIT como la clase STD. La clase con menor AUC resultó ser la clase UP, con un valor de 0.90.

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	0.8	1	1	0.8889	0.8777	0.9
DWN	1	0.9615	0.8333	0.9091	0.8951	0.9808
SIT	1	1	1	1	1	1
STD	1	1	1	1	1	1

Tabla 4.15: Medidas de desempeño para Red Neuronal

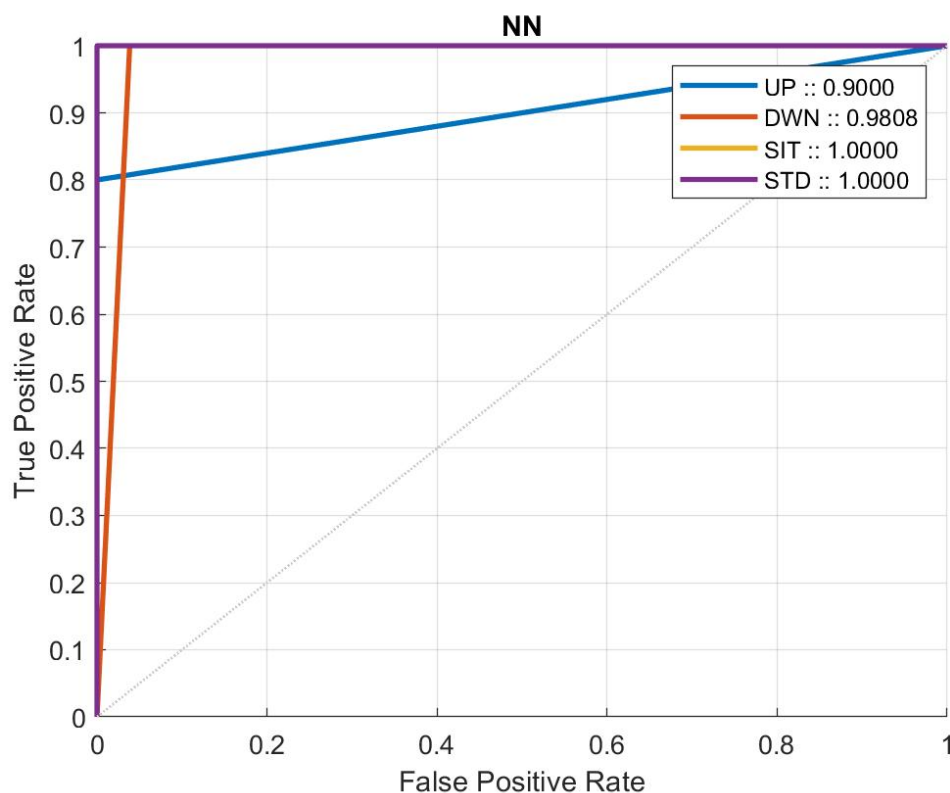


Figura 4.24: Curva ROC para Red Neuronal

### DeepInsight

Como se mencionó en la sección 3.2.2, la finalidad de esta metodología es representar un conjunto de datos que no son imágenes (como las mediciones provenientes de los sensores inerciales) a una forma que si lo sean, para después poder ingresarlas a una CNN. Para eso, primero se concatena cada vector con las muestras extraídas de los sensores de la MSBand a una matriz  $M$ . A la matriz traspuesta de  $M$  se le denomina matriz  $G$ , de tal forma que los renglones de  $G$  corresponden a los atributos de las muestras y las columnas, a cada una de las 303 muestras de  $M$ .

Para poder visualizar las muestras de  $G$  en un plano bidimensional, se hicieron experimentos con la técnica tSNE y PCA utilizando dos componentes principales. De tal forma que se graficaron cada uno de los 303 valores distintos correspondientes a cada muestra, en sólo dos dimensiones. El resultado de esta etapa se

muestra en la Figura 4.25. Para una mejor interpretación de la gráfica, los atributos correspondientes a valores del acelerómetro, aparecen en color rojo y los correspondientes al giroscopio en azul.

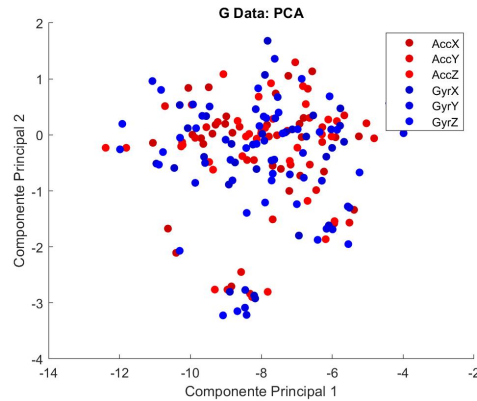


Figura 4.25: Datos de  $G$  utilizando PCA con 2 componentes principales

Una vez hecho esto, se determinó la envolvente convexa mínima necesaria para incluir todos los datos, como se muestra en la Figura 4.26. La envolvente convexa mínima se refiere al rectángulo con el área mínima tal que contenga todos los puntos en su interior.

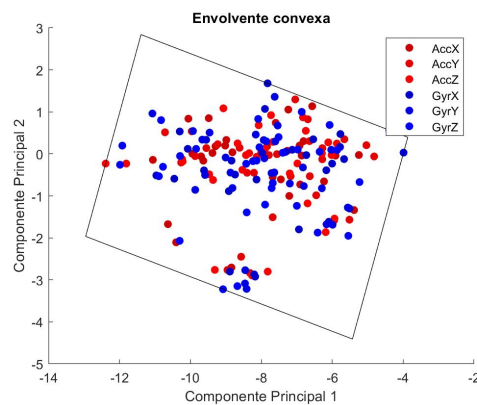


Figura 4.26: Envlovente convexa mínima

Después, se hace una rotación en el sentido de las manecillas del reloj de la envolvente convexa previamente determinada en conjunto con todos los datos

hasta obtener su posición vertical. Con esto, se determina la posición final correspondiente para cada uno de los atributos que forman una muestra de  $M$ . En la Figura 4.27 se puede apreciar el resultado de este proceso.

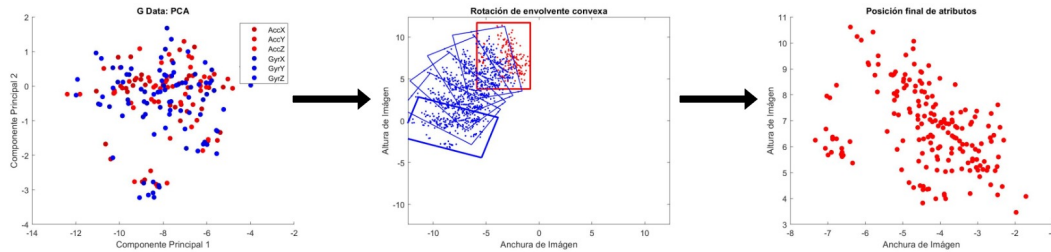
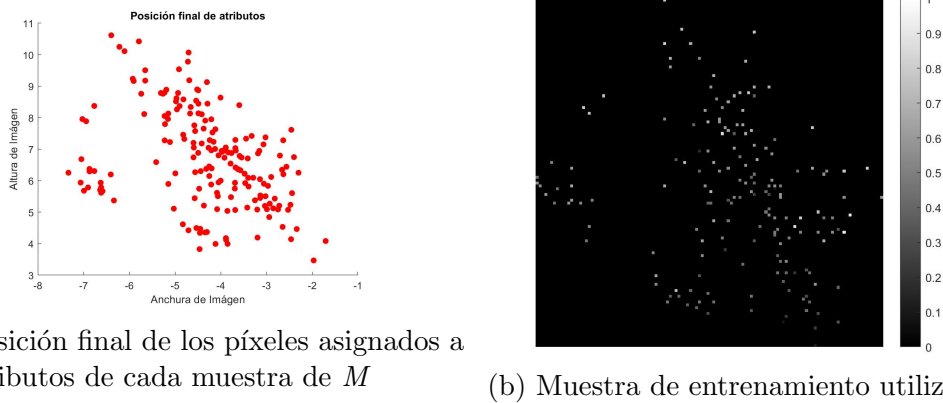


Figura 4.27: Proceso de rotación de la envoltura convexa mínima para encontrar la posición final de los píxeles asignados a los atributos de cada muestra de  $M$

Con la posición de cada píxel determinada para cada atributo, el siguiente paso es mapear la matriz  $M$  a una nueva Matriz de tamaño  $W \times H$ , donde:

$W$  es un número entero positivo que corresponde al ancho deseado para la imagen con los datos de cada muestra y  $H$ , es un número entero positivo que corresponde a la altura deseada para la imagen con los datos de cada muestra. El tamaño que arrojó mejores resultados mostrados en el presente documento fue de  $120 \times 120$  píxeles.

En la Fig. 4.28b se presenta una muestra en forma de imagen elegida aleatoriamente del conjunto de datos de entrenamiento. Dado que las muestras fueron normalizadas para tener varianza 1 y se utiliza únicamente un canal para cada imagen, los valores de los píxeles correspondientes a cada atributo oscila entre los colores blanco y negro, cuya tonalidad es proporcional al valor relativo del atributo en la muestra con respecto a la distribución del resto de las muestras para ese atributo. Al resto de los píxeles se le asignó un valor de 0 en los experimentos para que tuvieran un menor impacto al momento de ser procesados en la arquitectura de la red neuronal de convolución del paso siguiente de esta metodología.



(a) Posición final de los píxeles asignados a los atributos de cada muestra de  $M$  (b) Muestra de entrenamiento utilizada

Figura 4.28: Mapeo de muestra

Una vez transformadas las muestras a forma de imagen, se diseñó la arquitectura de la red neuronal de convolución. Ajustando las variables mencionadas en la Sección 3.2.2 se obtuvo la arquitectura final mostrada en la Figura 4.29, que fue generada utilizando la herramienta 'Nn-svg' de Alexander LeNail (LeNail, 2019). Como se puede apreciar, la arquitectura consta de tres capas ocultas representadas por bloques. Cada bloque mostrado está formado por una capa convolución, una de activación utilizando la función ReLU (Rectified linear unit function) y una capa Max Pooling que reduce el tamaño de la imagen a la mitad. Estas tres capas en conjunto seleccionan los píxeles de la imagen de su entrada que aporten más información. Para nuestro caso, las capas de Max Pooling seleccionan los píxeles con mayor valor (los más brillantes), reduciendo la imagen de entrada de la capa a una con la mitad de sus dimensiones (alto y ancho). La cantidad de filtros o kernels aplicados a cada bloque se muestran en las etiquetas mostradas en la parte inferior de cada capa oculta.

Una vez que se somete la imagen de 120 x 120 píxeles a los tres bloques 'ocultos', la información queda representada en una matriz de 14 x 14 x 64, que después se representa en forma de un vector de 12,244 atributos, reduciendo el tamaño inicial de 14,400 (120 x 120). Este vector después se ingresa a una red neuronal artificial convencional, cuya salida se somete a una función *softmax*.

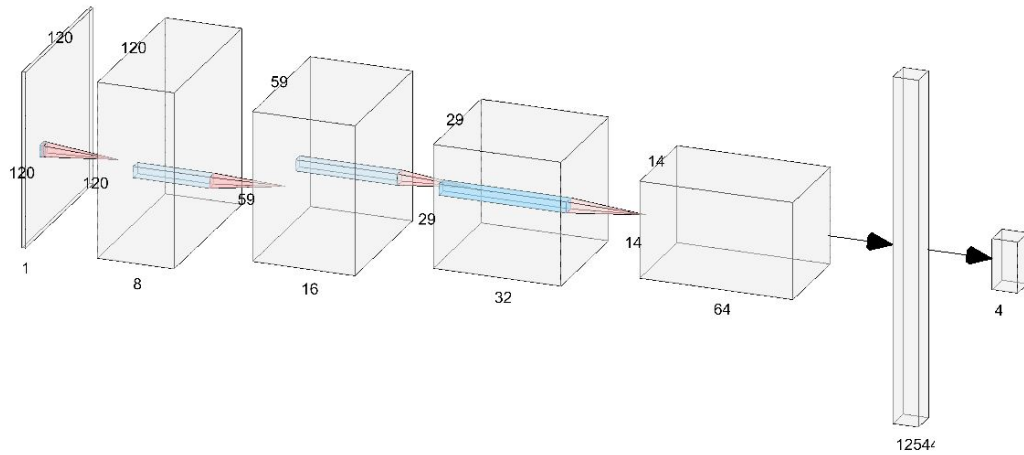


Figura 4.29: Arquitectura de la CNN

El proceso de entrenamiento de esta red se muestra en la Figura 4.30. Aunque la gráfica si alcanzó su nivel estable, los niveles de pérdida fueron los suficientemente altos como para alcanzar resultados favorables. Como se puede apreciar, la tasa de reconocimiento máxima fue de 51.61%.

### 4.3 Fase 3: Minado de Datos

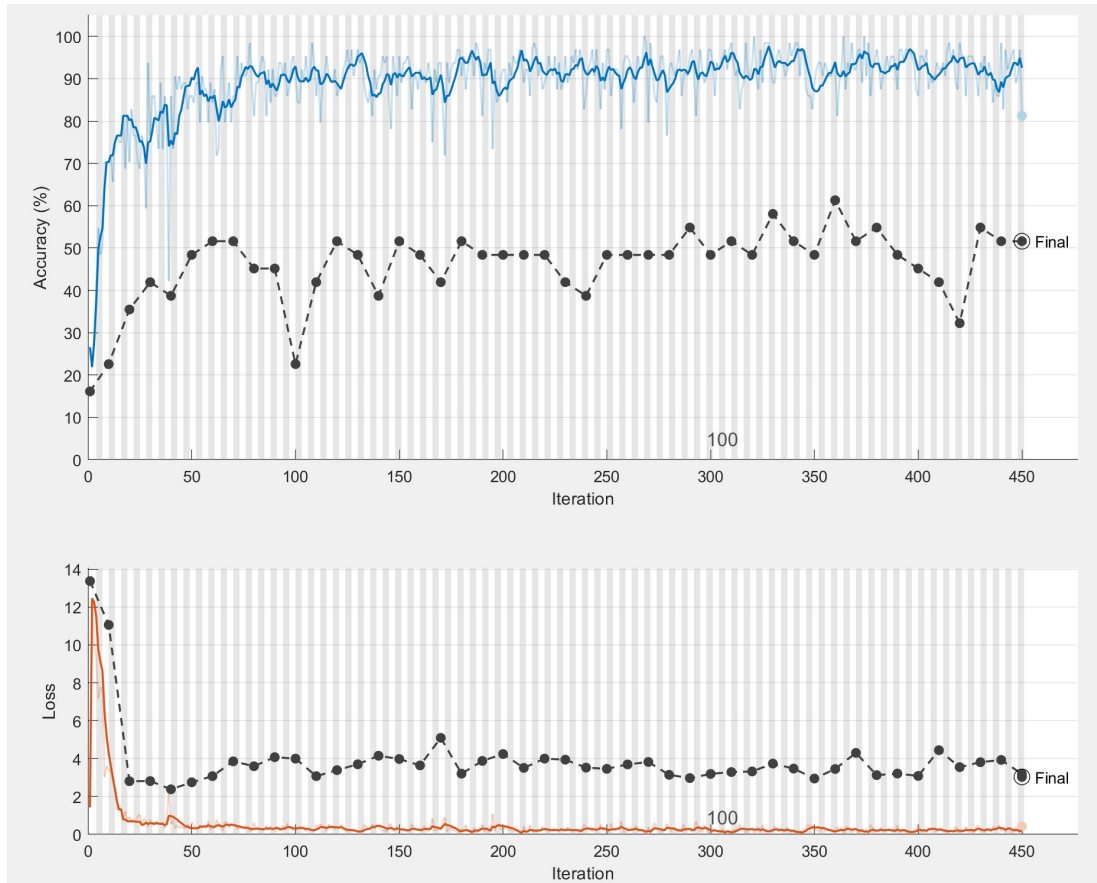


Figura 4.30: Proceso de entrenamiento del modelo DeepInsight

La gráfica ROC de este modelo se muestra en la Figura 4.31. La única clase que arrojó resultados satisfactorios fue la clase de UP.

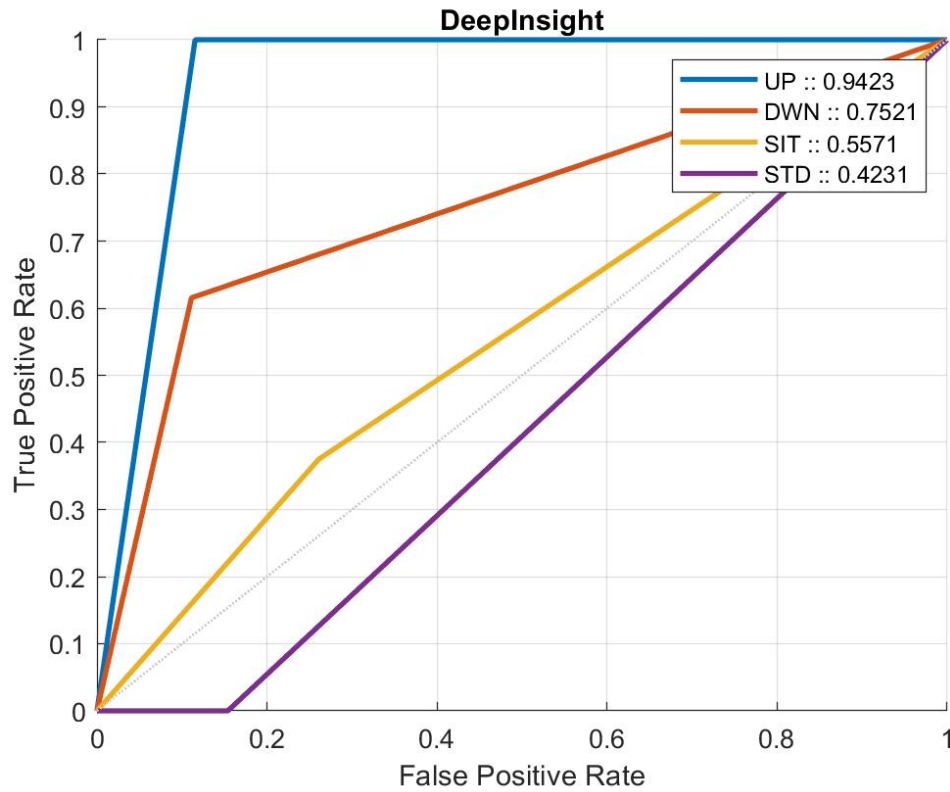


Figura 4.31: Gráfica ROC resultante de la metodología DeepInsight

Las métricas de desempeño del modelo entrenado se muestran en la Tabla 4.16.

Clase	Medida					
	SENS	SPEC	PRE	F1	MCC	AUC
UP	1	0.8846	0.625	0.7692	0.7436	0.9423
DWN	0.6154	0.8889	0.8	0.6957	0.5323	0.7521
SIT	0.375	0.7391	0.3333	0.3529	0.11	0.5571
STD	0	0.8462	0	0	-0.1688	0.4231

Tabla 4.16: Medidas de desempeño del modelo DeepInsight

### 4.3.3 Métodos de Visualización

En la presente investigación, se utilizaron dos métodos para visualizar los datos: PCA and t-SNE. El conjunto de datos de entrada *Movements* se puede visualizar

en la Figura 4.32 utilizando ambos métodos.

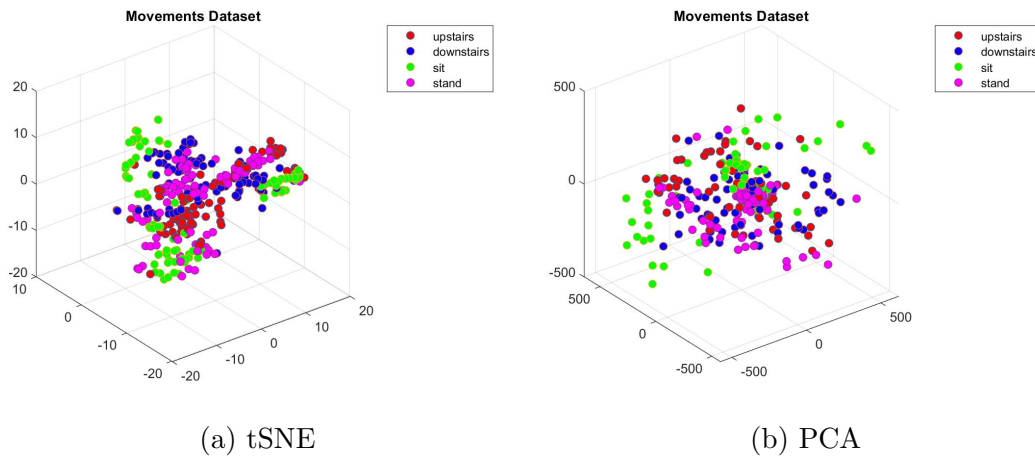


Figura 4.32: Movements Dataset

Para este conjunto de datos, se puede ver una mayor definición en los clusters correspondientes a cada clase utilizando la técnica de tSNE en comparación con la representación de la visualización utilizando PCA a dos componentes principales.

Aquí se puede apreciar que los puntos correspondientes a la clase DWN se encuentran muy dispersos a través de todo el universo de datos, razón por la cual la mayoría de los algoritmos de clasificación tuvieron dificultades para clasificar esa clase correctamente en comparación con las demás.

#### 4.3.4 Comparación de Modelos encontrados

En la Figura 4.33 se muestran las curvas ROC para UP. Esta clase fue la que se pudo clasificar con valores más satisfactorios a través de todos los algoritmos utilizados. Teniendo un área mínima debajo de la curva ROC de 0.9286 con el modelo resultante de vecinos cercanos.

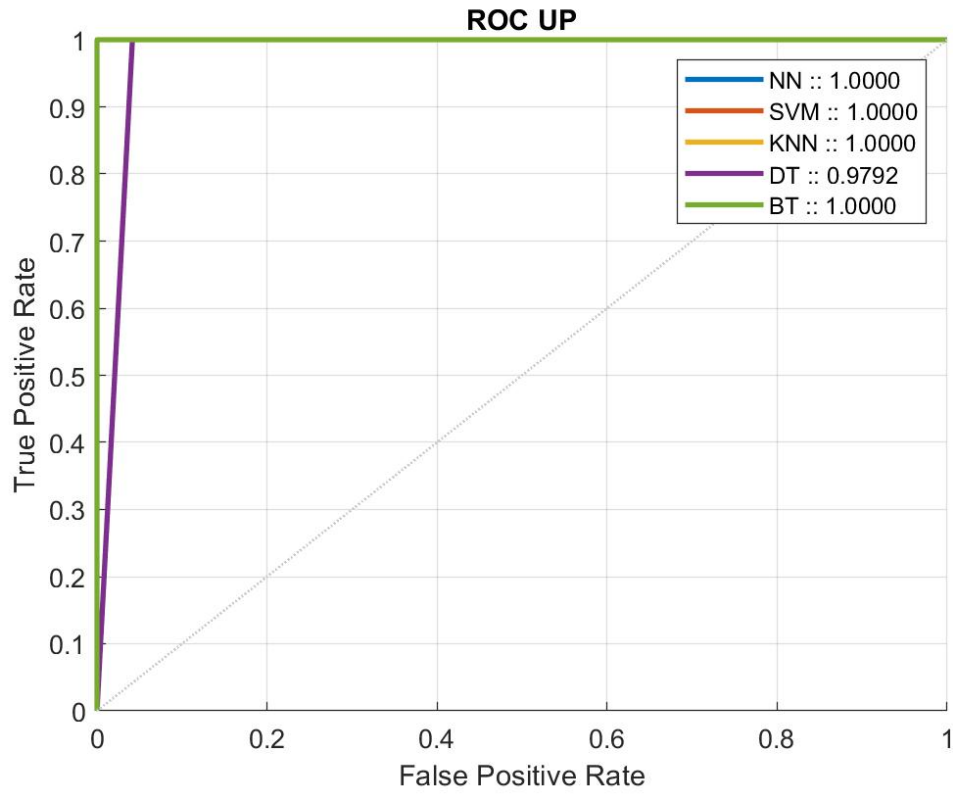


Figura 4.33: Comparación de curvas ROC para UP

En la Figura 4.34 se muestran las curvas ROC para DWN.

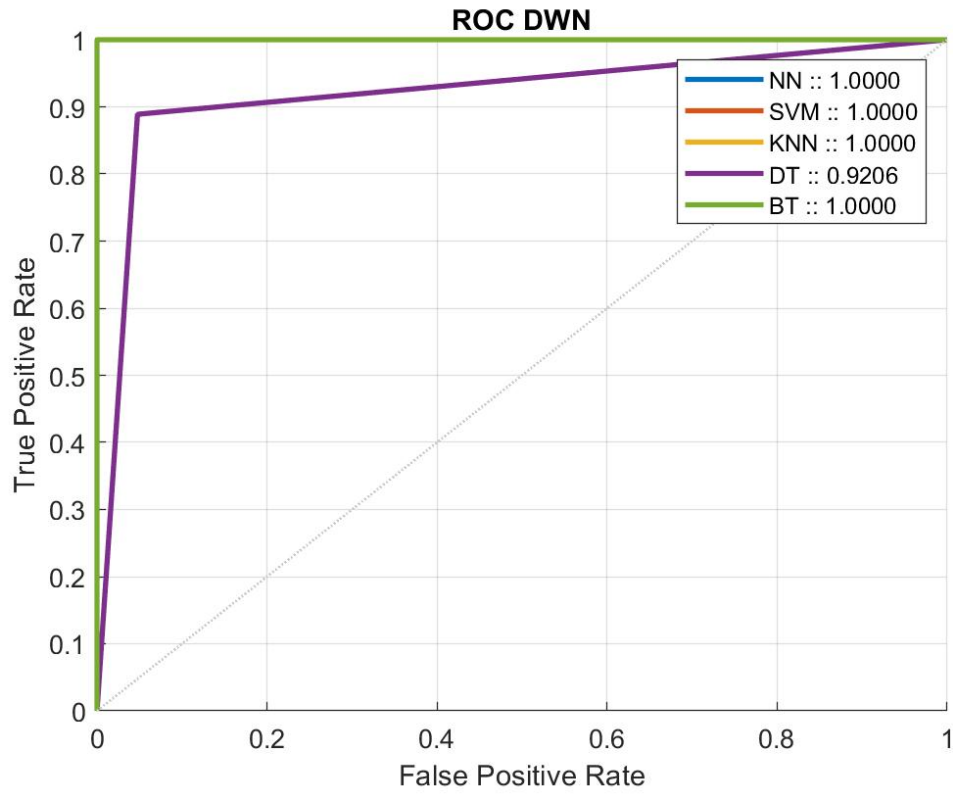


Figura 4.34: Comparación de curvas ROC para DWN

En la Figura 4.35 se muestran las curvas ROC para SIT.

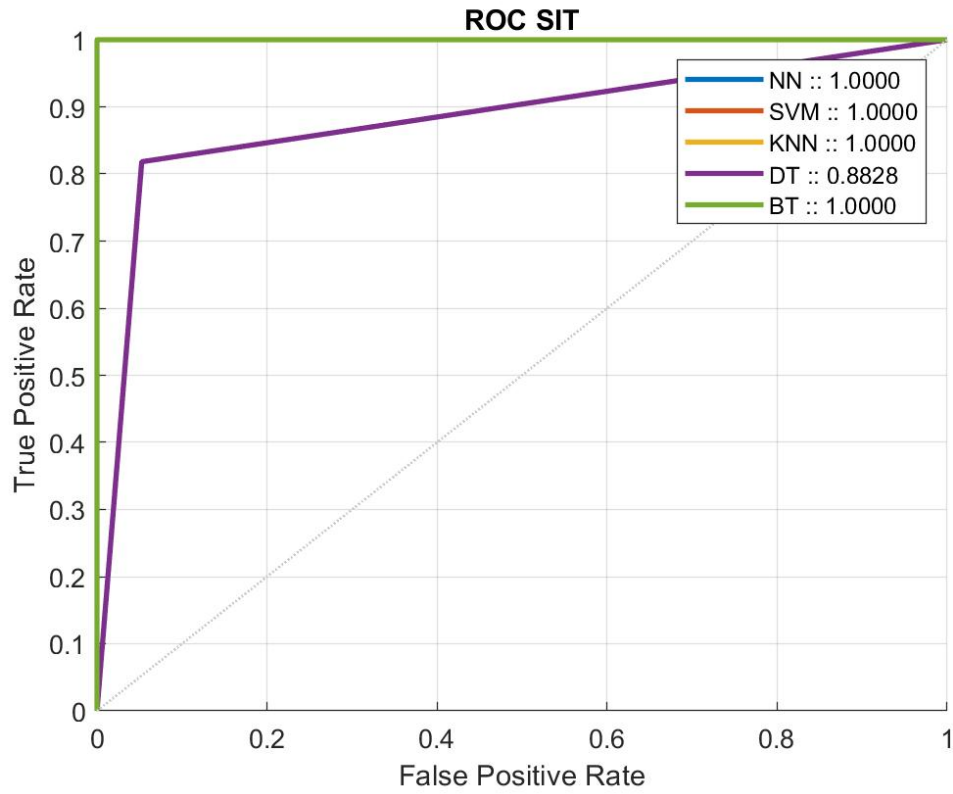


Figura 4.35: Comparación de curvas ROC para SIT

En la Figura 4.36 se muestran las curvas ROC para STD.

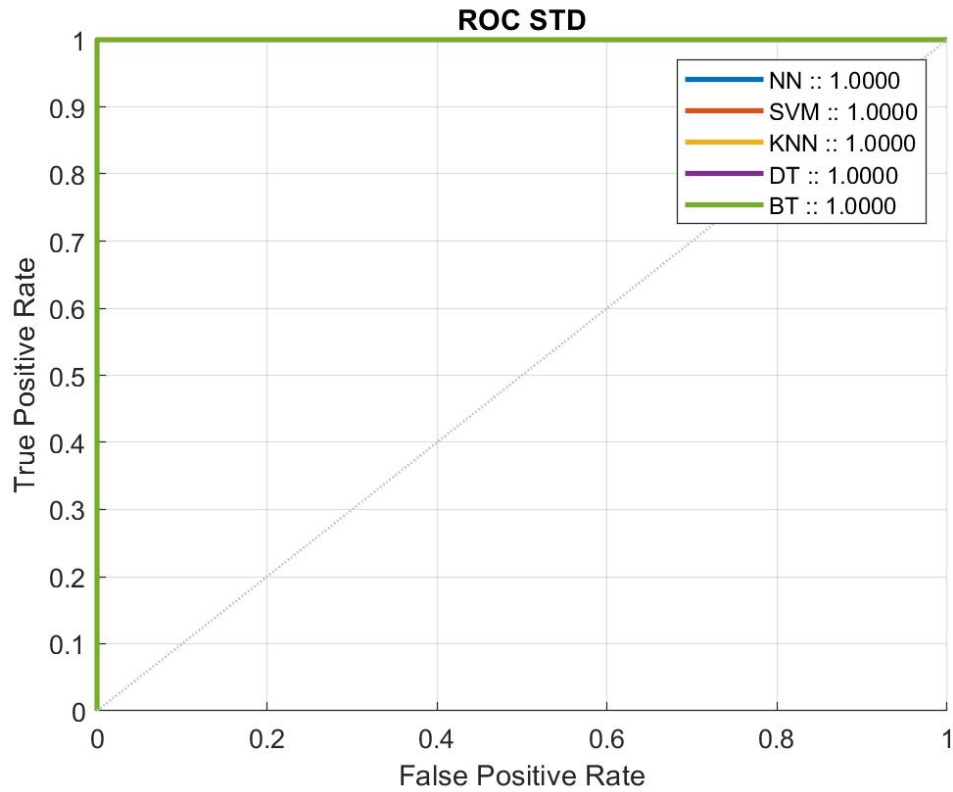


Figura 4.36: Comparación de curvas ROC para STD

### 4.3.5 Selección de mejor Modelo

Se seleccionó el modelo con la mayor tasa de reconocimiento para evaluar las muestras de validación.

Los valores de las matrices de confusión mostradas en la Figura 4.37 corresponden al valor relativo de los aciertos positivos con respecto al total de las muestras.

### 4.3 Fase 3: Minado de Datos

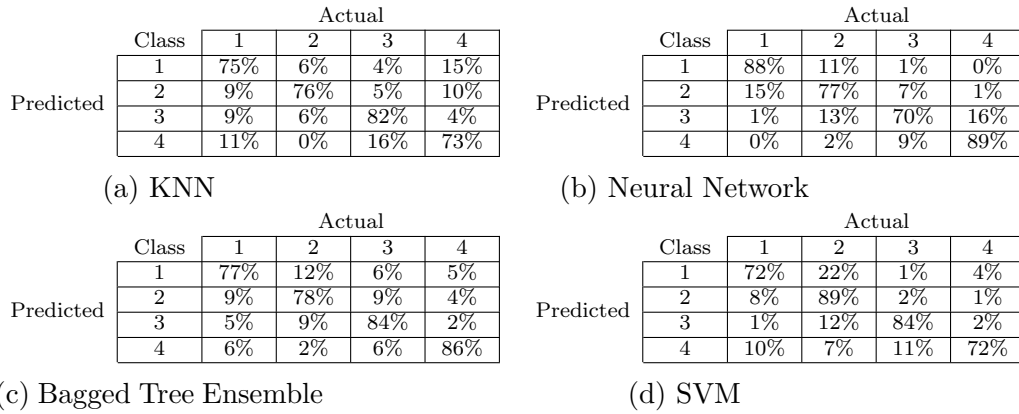


Figura 4.37: Matrices de Confusión de Salida

El mejor modelo es la red neuronal, por que logró una tasa de reconocimiento general del 96.4%. La configuración recomendada es utilizar una arquitectura de una sola capa oculta con diez neuronas, utilizando el método de optimización de gradiente descendente y mean squared error como medida de pérdida. Tomando en cuenta esa arquitectura se puede obtener un modelo clasificador con resultados satisfactorios para los movimientos de subir escaleras, bajar escaleras, pararse y sentarse, siempre y cuando el acelerómetro y el giroscopio utilizados en el dispositivo vestible de muñeca trabajen a una frecuencia de muestreo mínima de 8hz.

# Capítulo 5

## Conclusiones

### 5.1 Conclusiones

Después de haber realizado los experimentos presentados en este documento, se concluye que de entre todos los algoritmos de clasificación utilizados, la red neuronal es la que presenta mejores resultados para clasificar los movimientos propuestos (subir escaleras, bajar escaleras, pararse y sentarse). La configuración necesaria para obtener los resultados expuestos en el presente documento consta de una arquitectura de una sola capa oculta con diez neuronas, utilizando el método de optimización de gradiente descendente y mean squared error como medida de pérdida.

Las condiciones que se deben cumplir para poder recrear los experimentos y obtener los resultados presentados son las siguientes:

1. Que el dispositivo vestibular de muñeca utilizado para realizar los movimientos conste con un acelerómetro y un giroscopio.
2. El acelerómetro y el giroscopio utilizados en el dispositivo vestibular de muñeca trabajen a una frecuencia de muestreo mínima de 8hz.

Las complicaciones para obtener modelos más robustos con menores márgenes de error se listan a continuación:

1. De entre las razones que afectaron considerablemente los resultados en los experimentos fue la inconsistencia en los movimientos entre los participantes, e.g. la tasa de movimiento de brazos al subir o bajar las escaleras o si la persona se apoyaba en la silla al levantarse o al sentarse. Por esta razón, los valores absolutos mínimos y máximos en el conjunto de datos de cada participante variaban significativamente.
2. La baja cantidad de muestras para entrenamiento tuvo un gran impacto en el desempeño tanto de la red neuronal artificial como en los resultados obtenidos con la metodología DeepInsight en comparación con las otras técnicas de minería de datos a las que se sometieron los datos.

## 5.2 Trabajo Futuro

El ritmo de crecimiento en la industria de los dispositivos móviles vestibles, la capacidad de transmisión de información y el desarrollo de ambientes inteligentes dentro de las casas y/u oficinas, hace de este tema de investigación uno muy interesante y potencial para seguir desarrollando más investigación al respecto. Entre los posibles caminos a seguir se mencionan los siguientes:

1. Las técnicas de Aprendizaje Profundo o *Deep Learning* también pueden ser utilizadas en estos experimentos para alcanzar una tasa de reconocimiento mayor. Al momento de la escritura de este documento se están realizando experimentos utilizando la metodología *Deep Insight* publicada por Sharma et al. en (Sharma, 2019), aunque aún no se logra este objetivo, los resultados lucen prometedores, por ello, ese trabajo también puede ser considerado como una gran oportunidad para trabajo futuro.
2. La colección de más muestras provenientes de diferentes participantes también puede considerarse como trabajo futuro, de esta forma se obtendrían mejoras en los algoritmos de clasificación.

## 5.3 Publicación

### 5.3.1 Encuentro Nacional de Computación

Resultados de este trabajo fueron presentados en el Encuentro Nacional de Computación 2020 efectuado del 24 al 26 de Agosto del 2020, y agregados al libro "Aplicaciones de la Computación" (Primera edición, Agosto 2020) emitido por el Centro de Investigación en Matemáticas Avanzadas de la Universidad Autónoma de Coahuila.

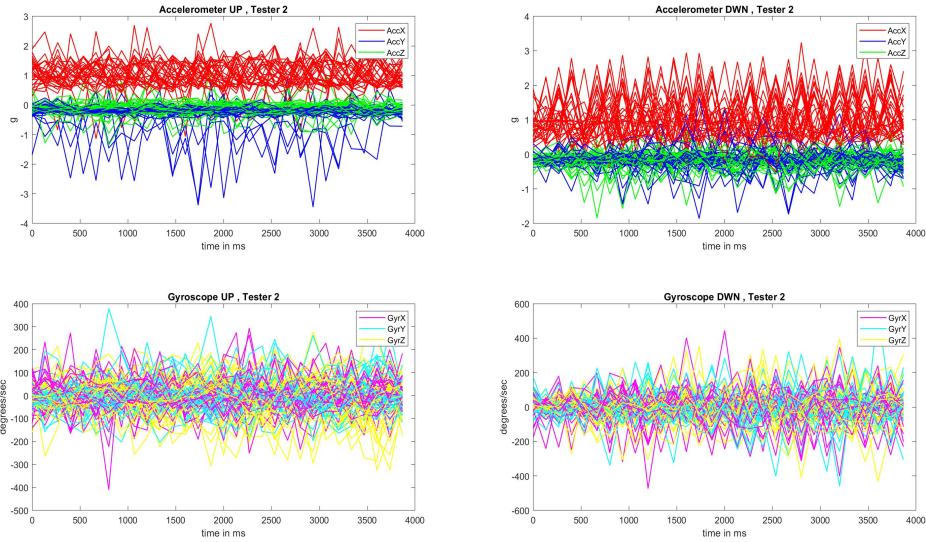
# Anexos A

## Gráficas de Movimientos

### A.0.1 Probador 2

En las gráficas presentadas en las Figuras [A.1](#) y [A.2](#), se visualizan las mediciones de cada uno de los ejes de los sensores correspondientes a cada movimiento realizado por el Probador 2.

En los movimientos de subir y bajar escaleras mostrados en la Figura [A.1](#), se muestra que no presenta un patrón fácil de apreciar a simple vista, y esto se debe a que el Probador 2 prácticamente no movía los brazos mientras efectuaba el movimiento, los mantuvo hacia abajo, razón por la cual, el efecto proveniente de las vibraciones al pisar los escalones se reflejan directamente en el eje X del acelerómetro, por ello las mediciones de ese eje son más altas que aquellas del Probador 1.



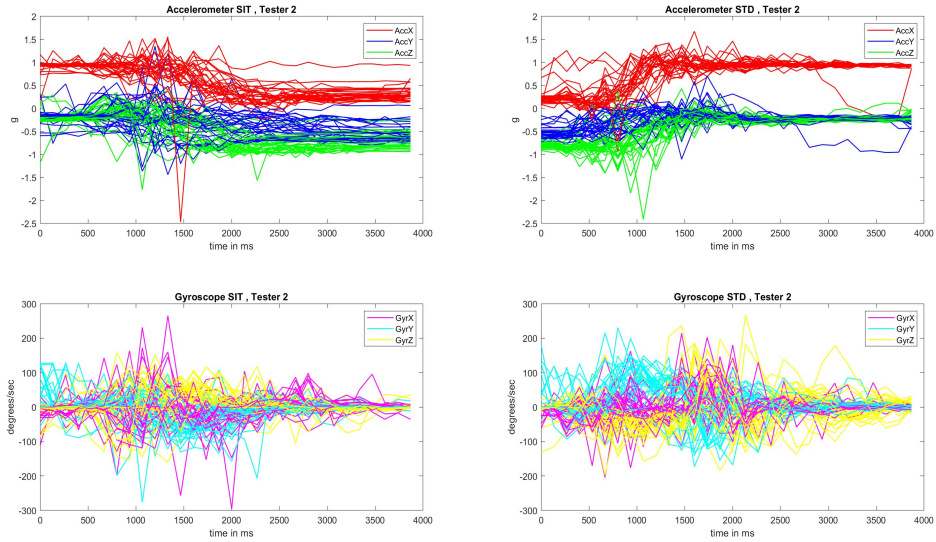
(a) UP

(b) DWN

Figura A.1: Movimientos UP y DWN realizados por Probador 2

En las gráficas de la Figura A.2 se puede apreciar un patrón en las mediciones de los ejes del acelerómetro, muy similar al reflejado en las mediciones correspondientes a los movimientos realizados por el Probador 1.

El ruido es considerablemente más notorio que aquel del Probador 1, esto muestra la posibilidad de que el Probador 2 haya portado la banda con más holgura en la muñeca. Este ruido se puede notar también en las mediciones efectuadas en el giroscopio, si se analizan sus valores, se alcanza a apreciar también un patrón en cada uno de sus ejes, pero el ruido evita que se noten a simple vista.



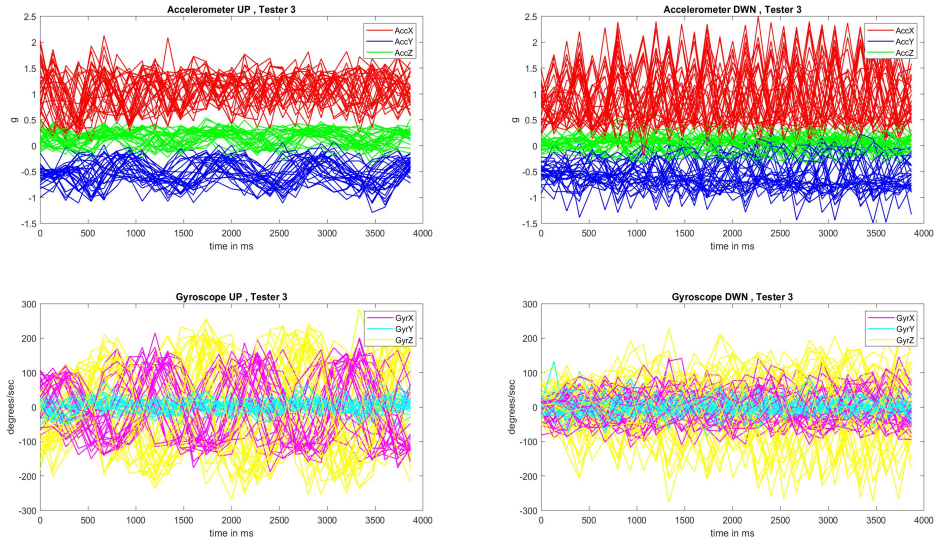
(a) SIT

(b) STD

Figura A.2: Movimientos SIT y STD realizados por Probador 2

### A.0.2 Probador 3

En las gráficas presentadas en las Figuras A.3 yd A.4, se visualizan las mediciones de cada uno de los ejes de los sensores correspondientes a cada movimiento realizado por el Probador 3.

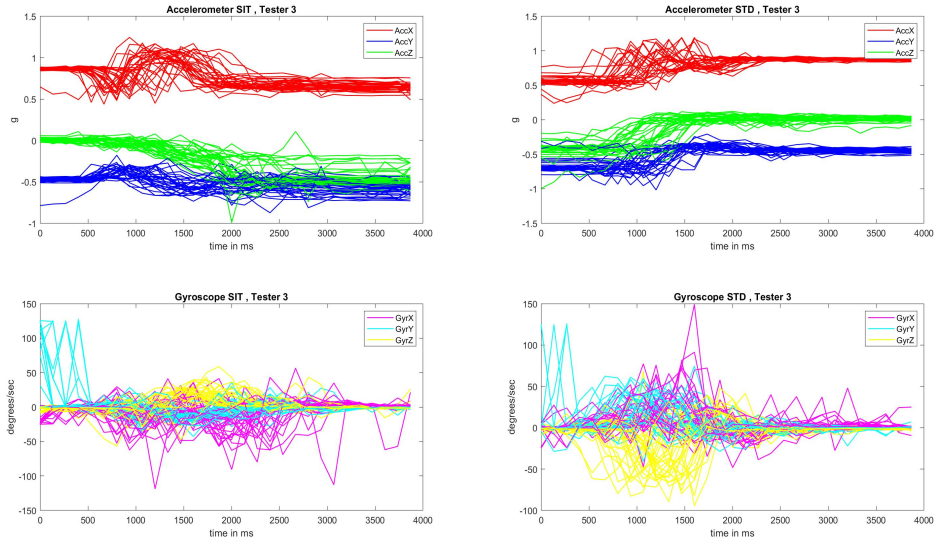


(a) UP

(b) DWN

Figura A.3: Movimientos UP y DWN realizados por Probador 3

Este participante realizó los movimientos de una manera más consistente, moviendo los brazos de arriba a bajo conforme subía y bajaba las escaleras, portando la banda ajustada a su muñeca, dando como resultado mediciones con patrones que se pueden apreciar fácilmente a simple vista, con mucho menor ruido comparado con las mediciones presentadas para estos movimientos correspondientes al Probador 2.



(a) SIT

(b) STD

Figura A.4: Movimientos SIT y STD realizados por Probador 3

# Anexos B

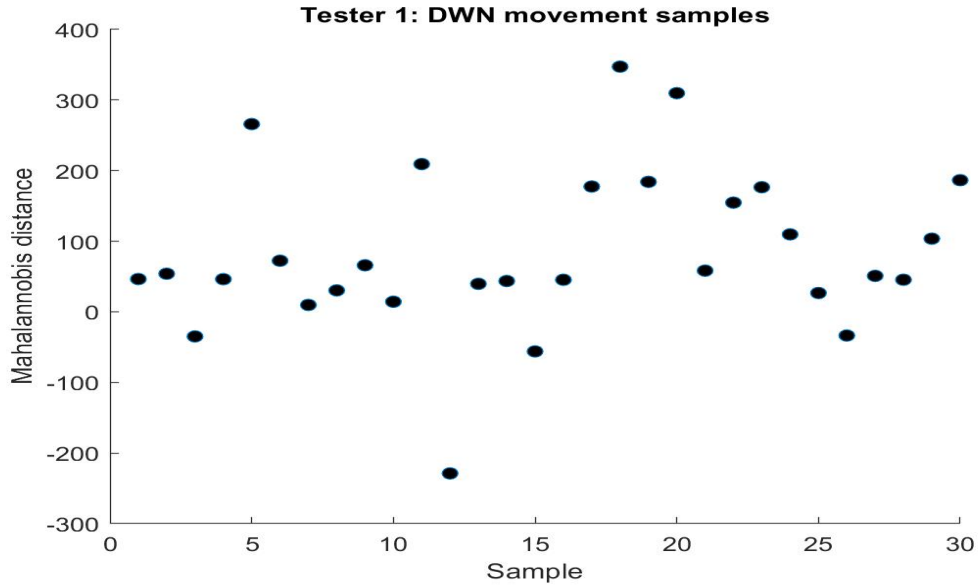
## Detección de outliers

### B.1 Movimientos de Probador 1

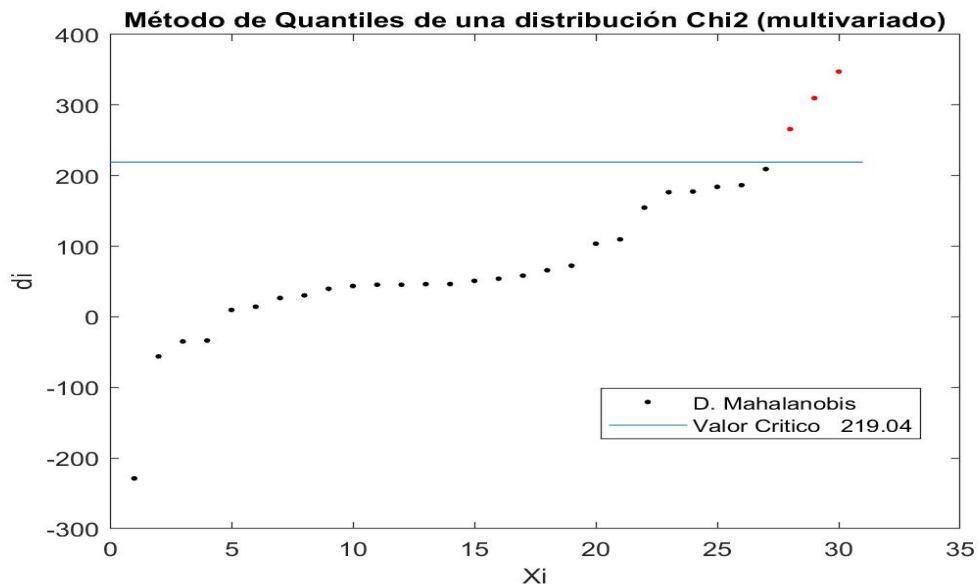
#### B.1.1 DWN

En la Figura [B.1a](#), se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras DWN efectuadas por el Probador 1. Una vez procesadas estas muestras por el Método de Quantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar tres outlier para este subconjunto de datos (Véase la Figura [B.1b](#)).

## B.1 Movimientos de Probador 1



(a) Distancias



(b) Outliers

Figura B.1: Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 1.

## B.1 Movimientos de Probador 1

Estas anomalías detectadas se muestran en la Figura B.2 con líneas rojas.

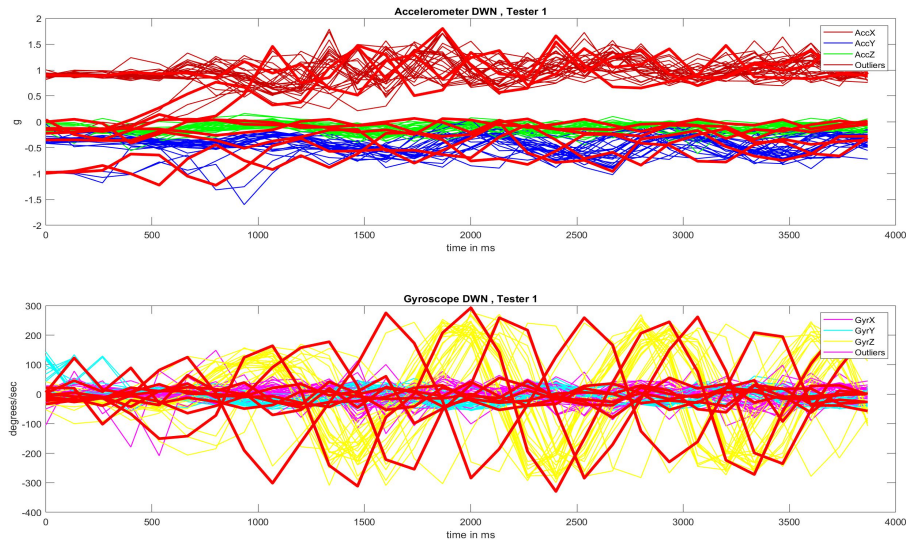
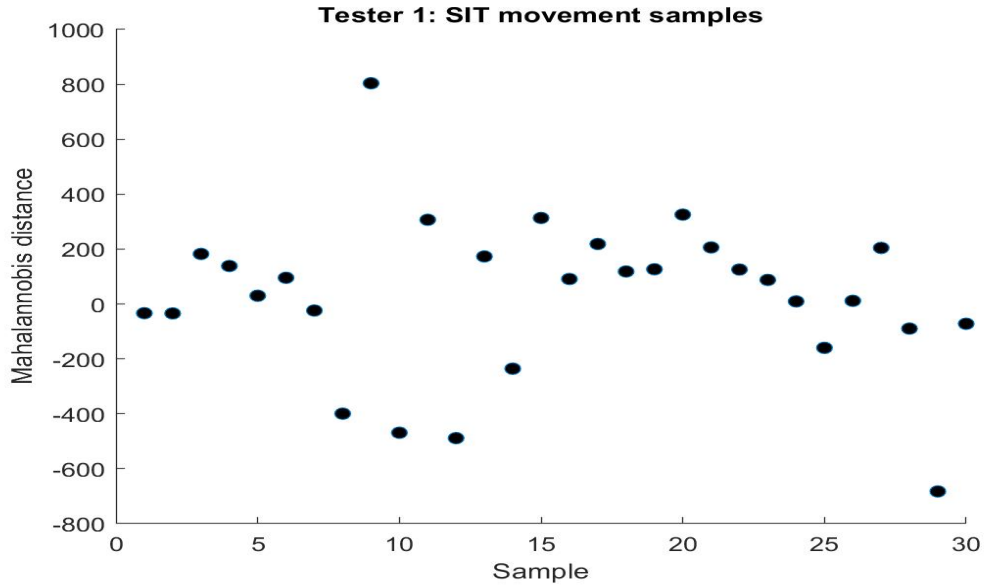


Figura B.2: Outliers detectados para el movimiento DWN efectuado por el Probador 1

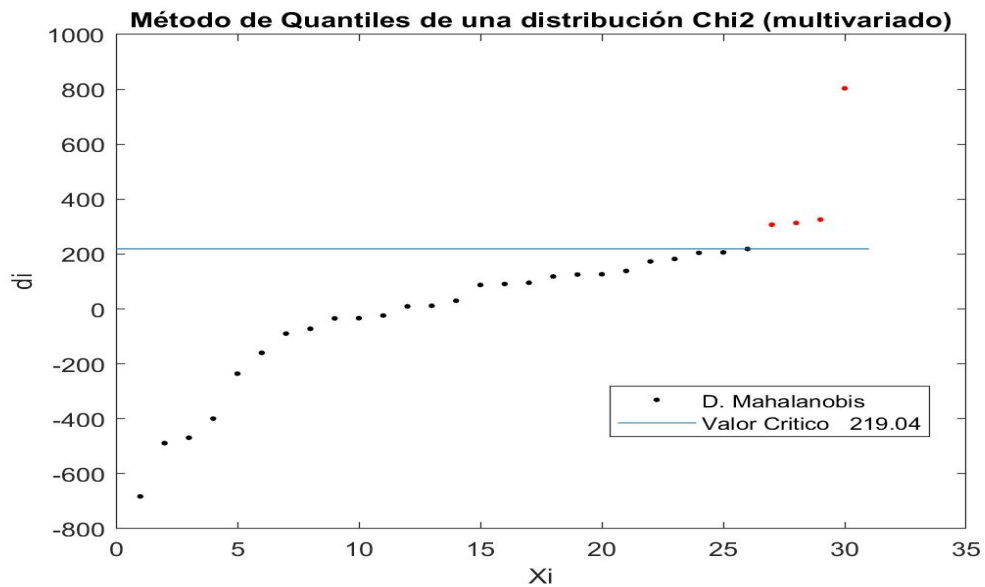
### B.1.2 SIT

En la Figura B.3a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras SIT efectuadas por el Probador 1. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar cuatro outlier para este subconjunto de datos (Véase la Figura B.3b).

## B.1 Movimientos de Probador 1



(a) Distancias



(b) Outliers

Figura B.3: Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 1.

Estas anomalías detectadas se muestran en la Figura B.2 con líneas rojas.

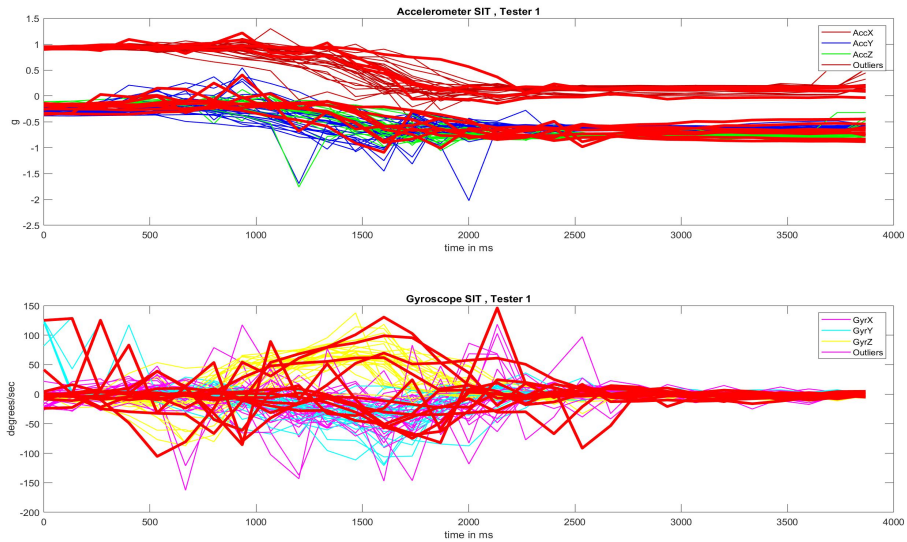
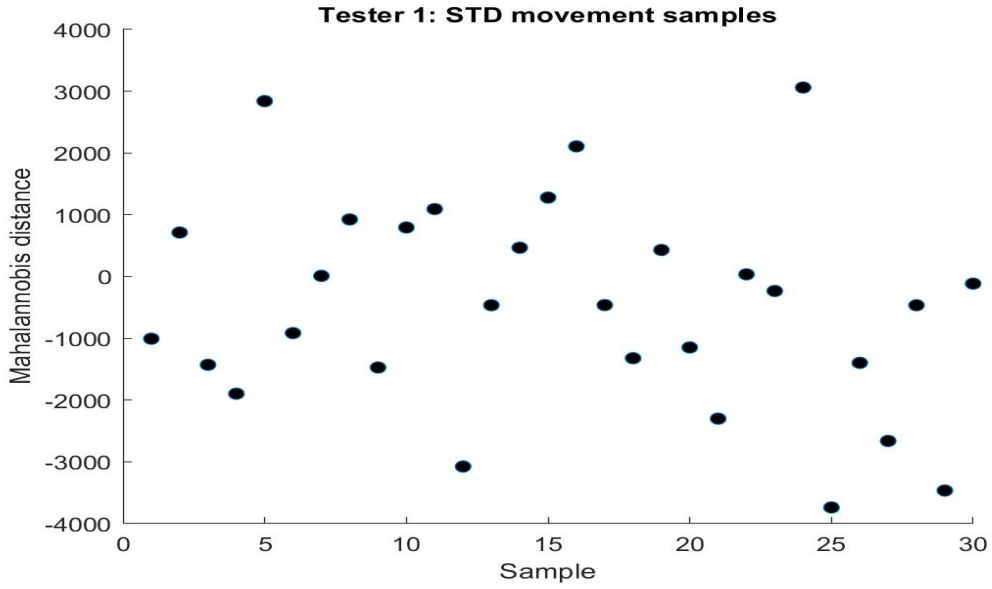


Figura B.4: Outliers detectados para el movimiento SIT efectuado por el Probador 1

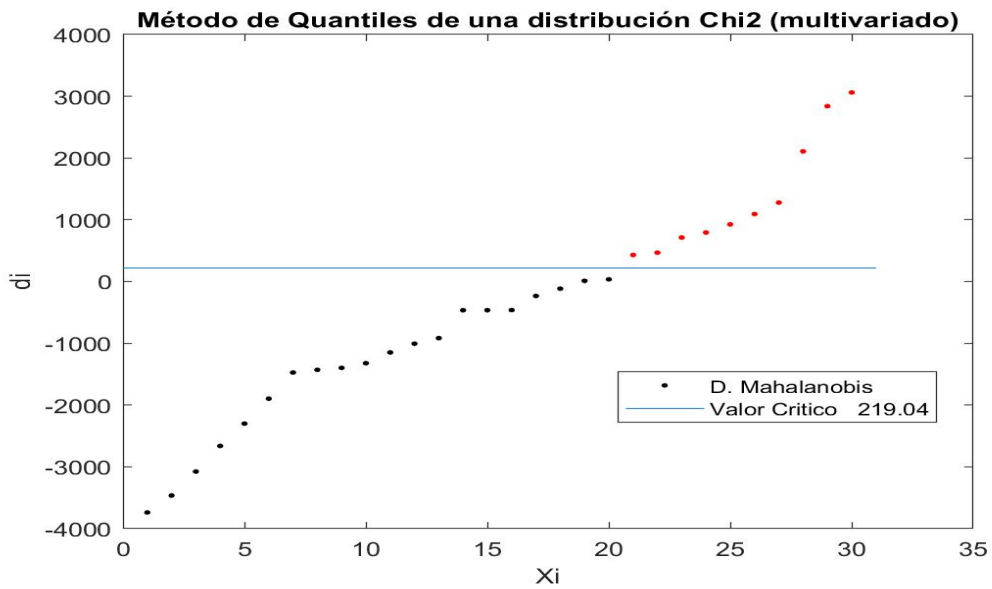
### B.1.3 STD

En la Figura B.5a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras STD efectuadas por el Probador 1. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar diez outlier para este subconjunto de datos (Véase la Figura B.5b).

## B.1 Movimientos de Probador 1



(a) Distancias



(b) Outliers

Figura B.5: Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 1.

Estas anomalías detectadas se muestran en la Figura B.2 con líneas rojas.

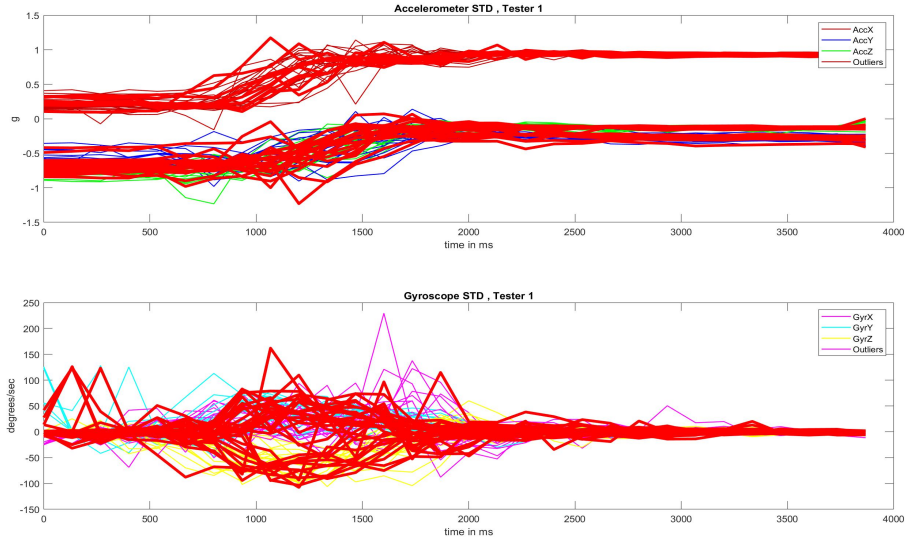
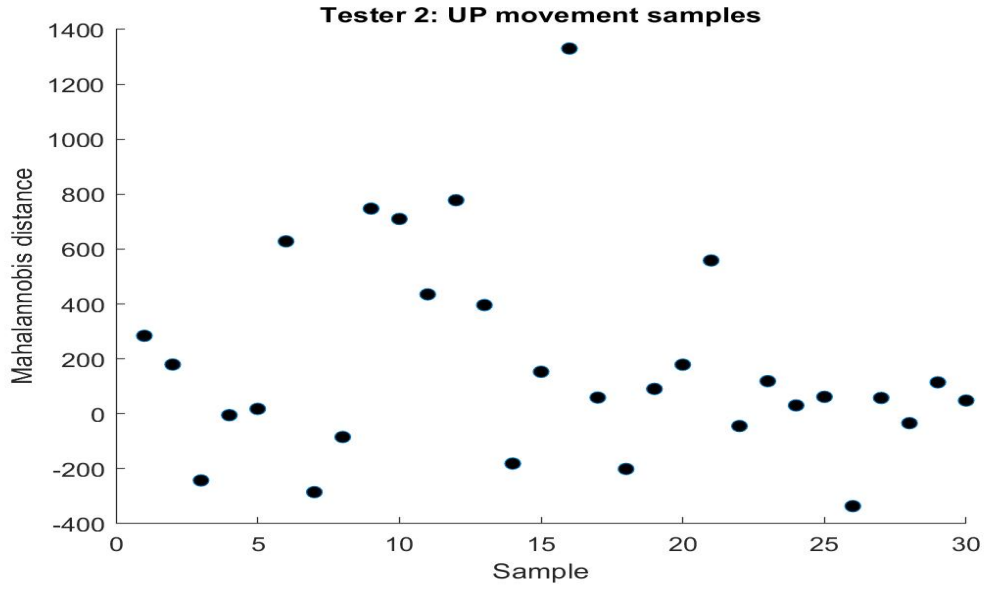


Figura B.6: Outliers detectados para el movimiento STD efectuado por el Probador 1

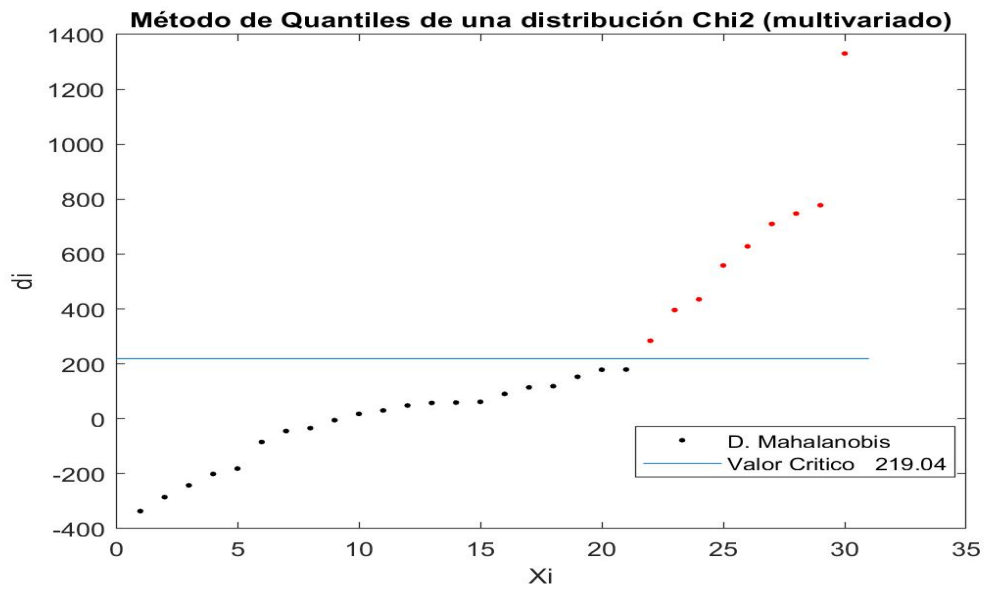
## B.2 Movimientos de Probador 2

### B.2.1 UP

En la Figura B.7a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras UP efectuadas por el Probador 2. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar nueve outlier para este subconjunto de datos (Véase la Figura B.7b).



(a) Distancias



(b) Outliers

Figura B.7: Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 2.

Estas anomalías detectadas se muestran en la Figura B.8 con líneas rojas.

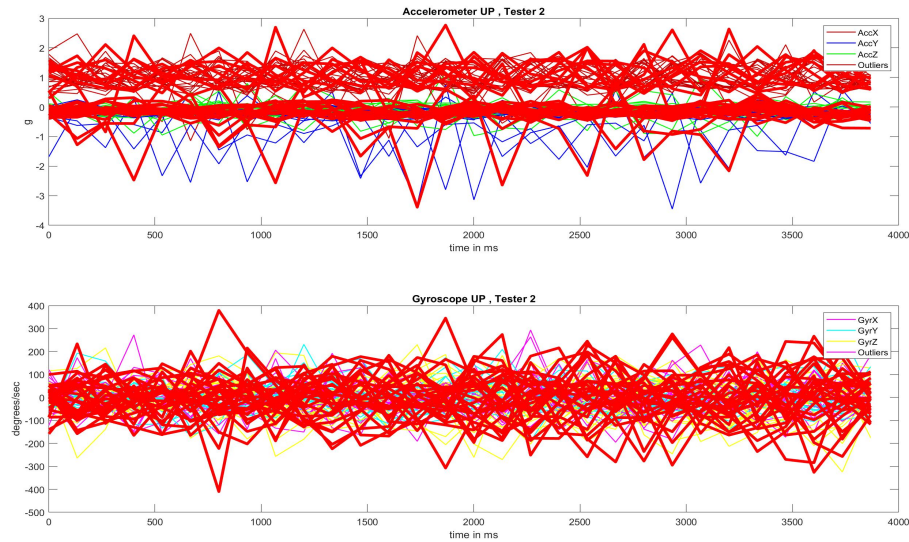
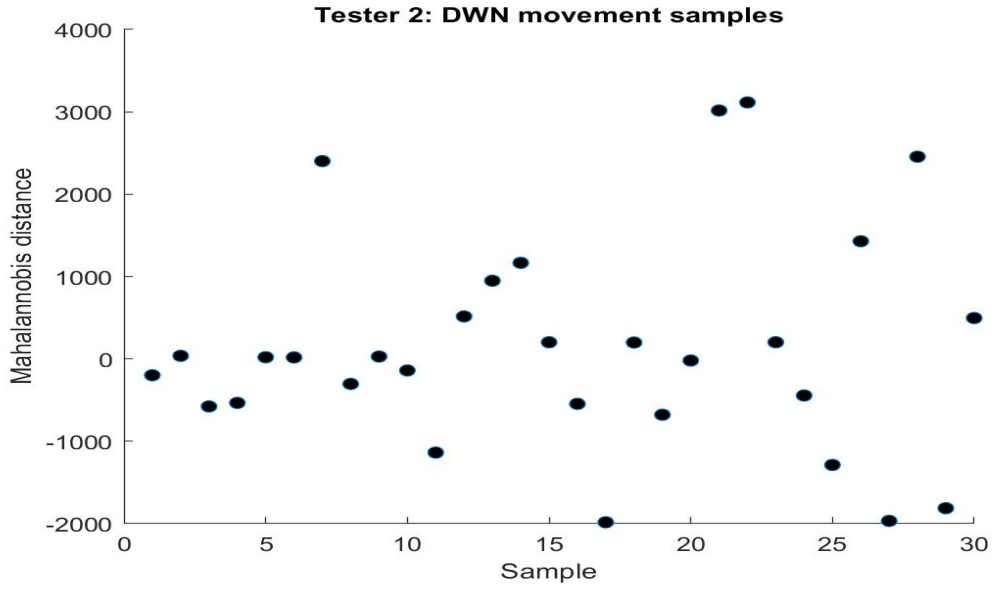


Figura B.8: Outliers detectados para el movimiento UP efectuado por el Probador 2

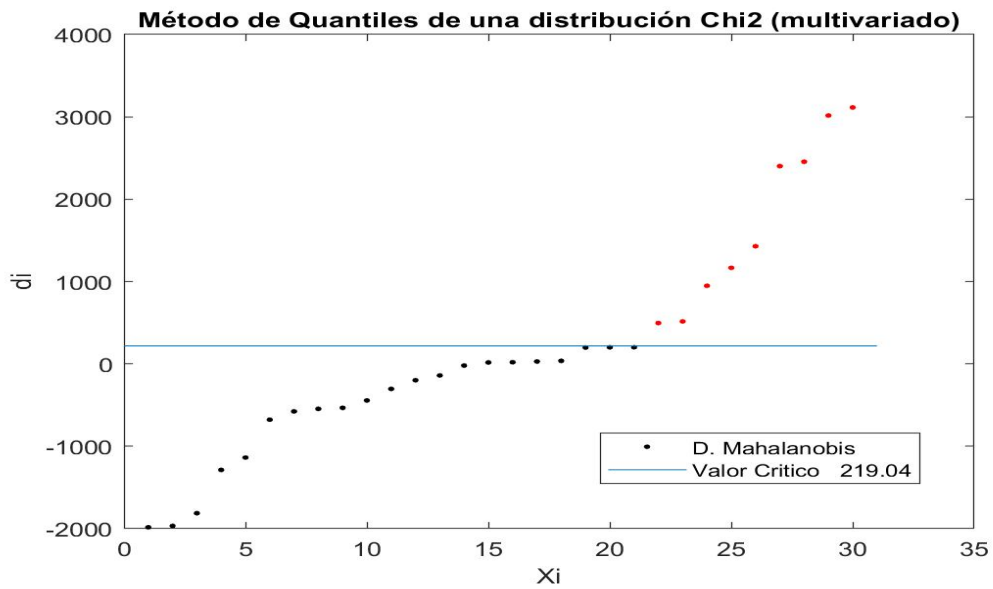
Este participante no realizaba movimientos periódicos con sus brazos al momento de subir las escaleras, el patrón en los movimientos no se pueden apreciar a simple vista y por consiguiente, se detectó una cantidad mayor de outliers con respecto a los demás probadores.

### B.2.2 DWN

En la Figura B.9a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras DWN efectuadas por el Probador 2. Una vez procesadas estas muestras por el Método de Quantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar nueve outlier para este subconjunto de datos (Véase la Figura B.9b).



(a) Distancias



(b) Outliers

Figura B.9: Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 2.

Estas anomalías detectadas se muestran en la Figura B.10 con líneas rojas.

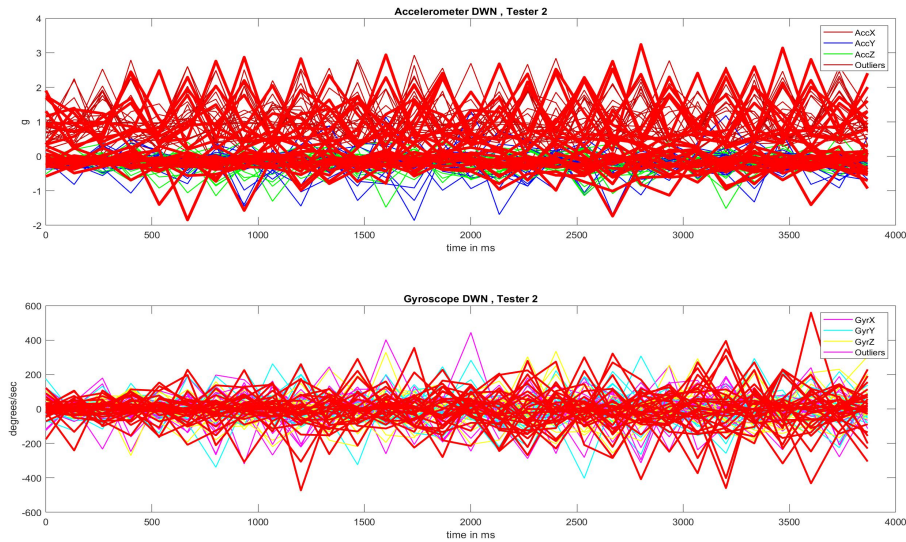
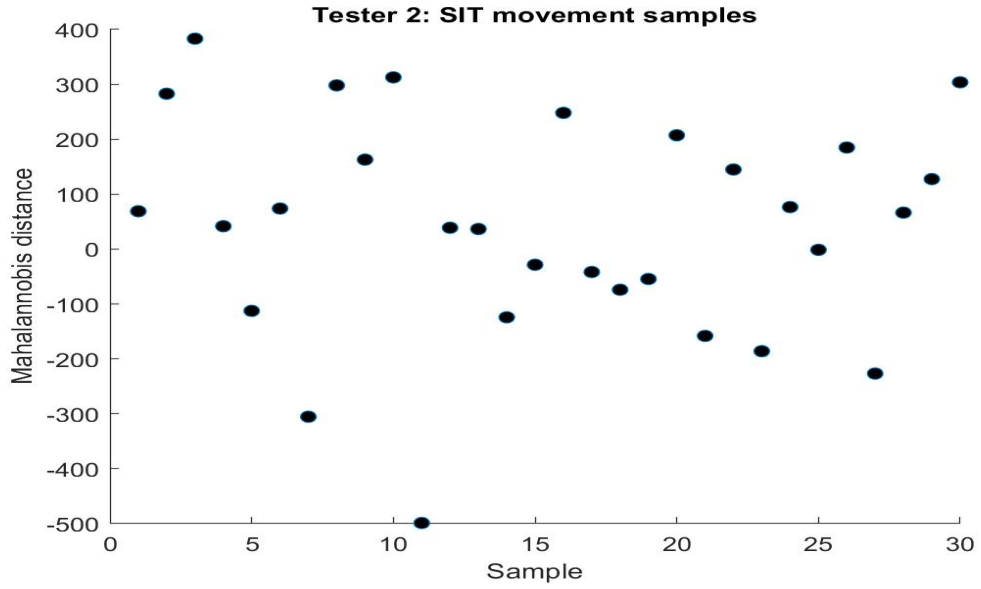


Figura B.10: Outliers detectados para el movimiento DWN efectuado por el Probador 2

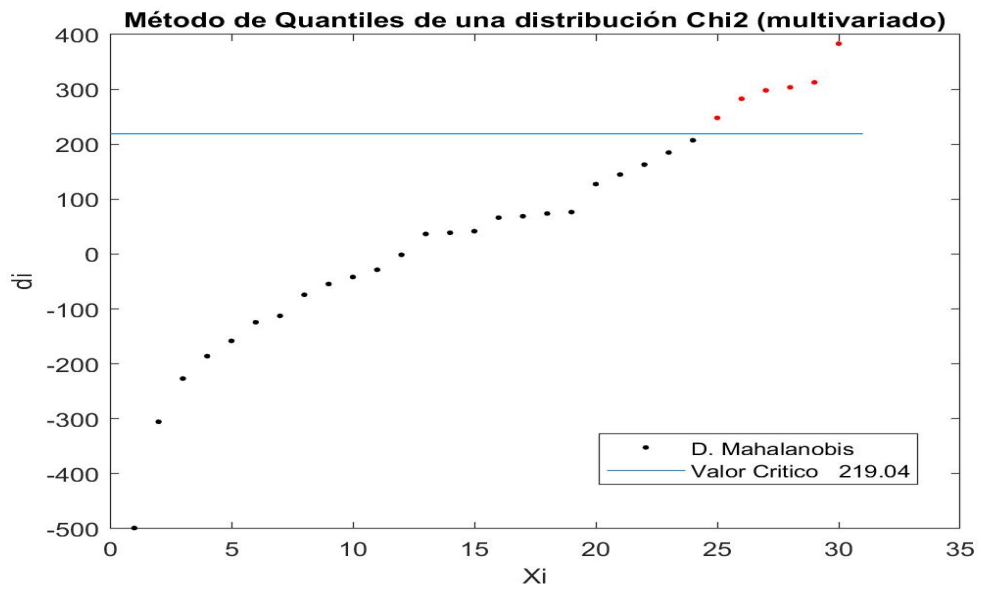
Al igual que con el movimiento de UP, este participante tampoco realizaba movimientos periódicos con sus brazos al momento de bajar las escaleras, el patrón en los movimientos no se pueden apreciar a simple vista y por consiguiente, se detectó una cantidad mayor de outliers con respecto a los demás probadores como se puede ver en la Figura B.10.

### B.2.3 SIT

En la Figura B.11a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras SIT efectuadas por el Probador 2. Una vez procesadas estas muestras por el Método de Quantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar seis outlier para este subconjunto de datos (Véase la Figura B.11b).



(a) Distancias



(b) Outliers

Figura B.11: Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 2.

Estas anomalías detectadas se muestran en la Figura B.12 con líneas rojas.

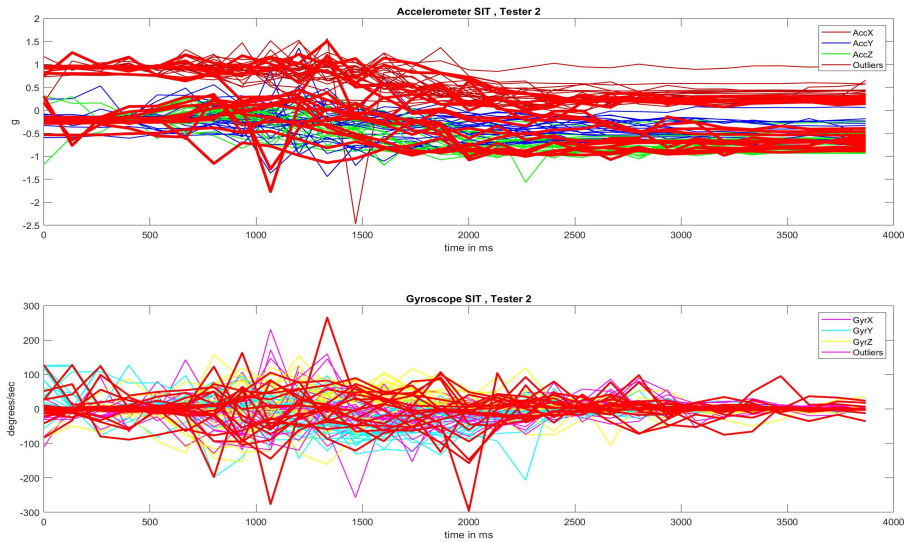
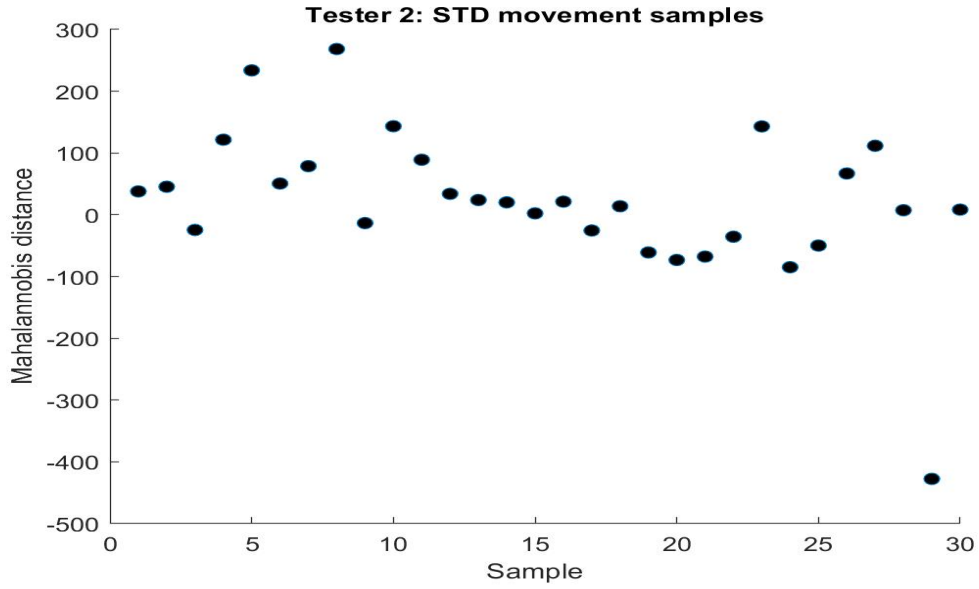


Figura B.12: Outliers detectados para el movimiento SIT efectuado por el Probador 2

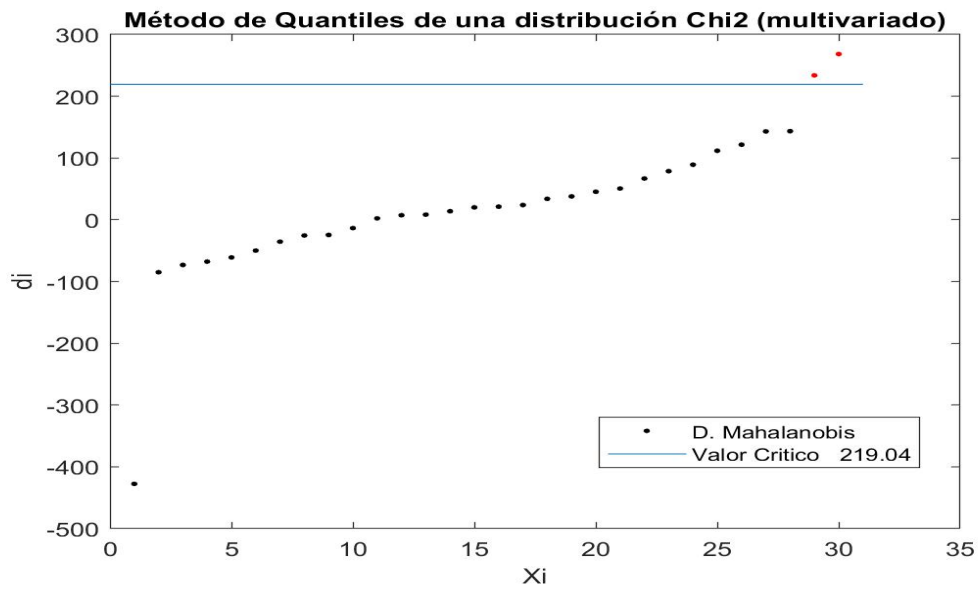
### B.2.4 STD

En la Figura B.13a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras STD efectuadas por el Probador 2. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar únicamente dos outlier para este subconjunto de datos (Véase la Figura B.13b).

## B.2 Movimientos de Probador 2



(a) Distancias



(b) Outliers

Figura B.13: Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 2.

Estas anomalías detectadas se muestran en la Figura B.14 con líneas rojas.

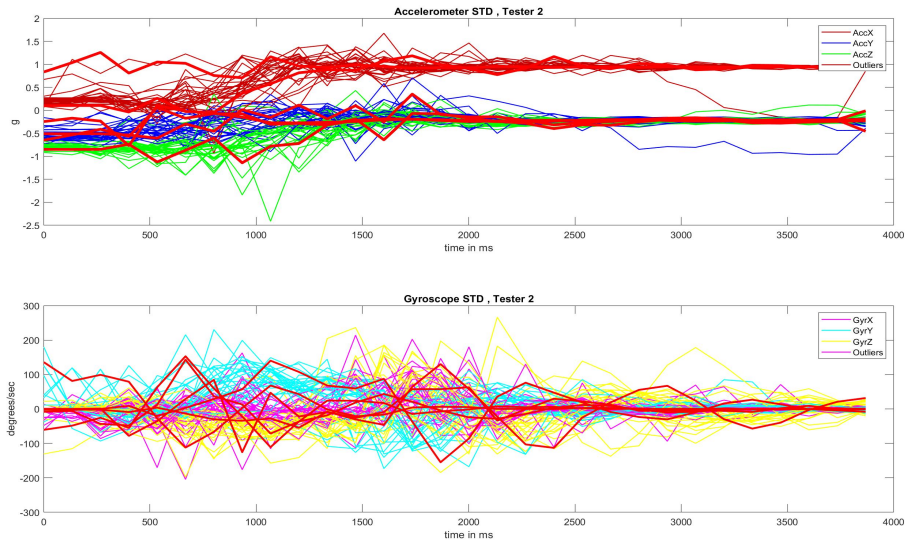
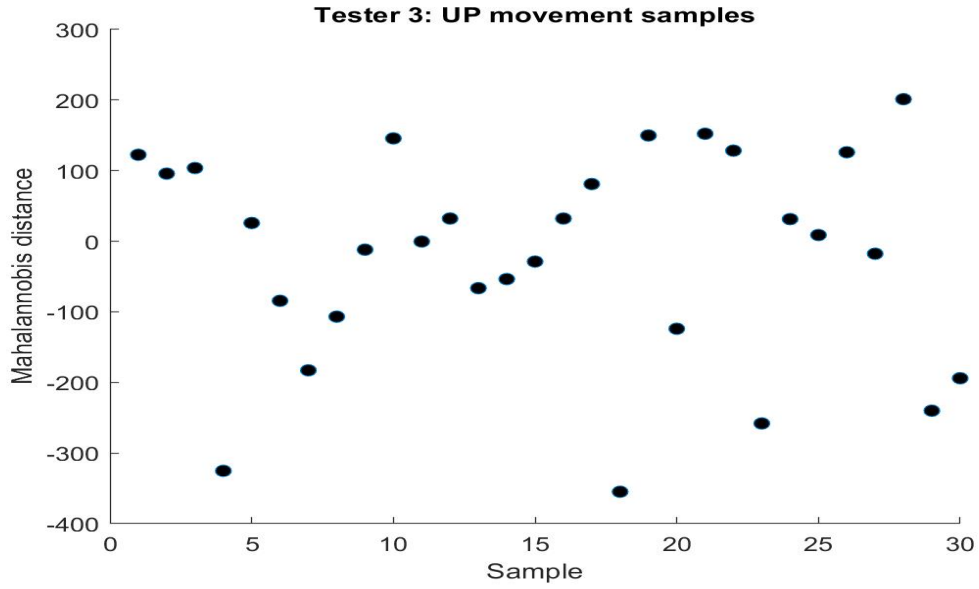


Figura B.14: Outliers detectados para el movimiento STD efectuado por el Probador 2

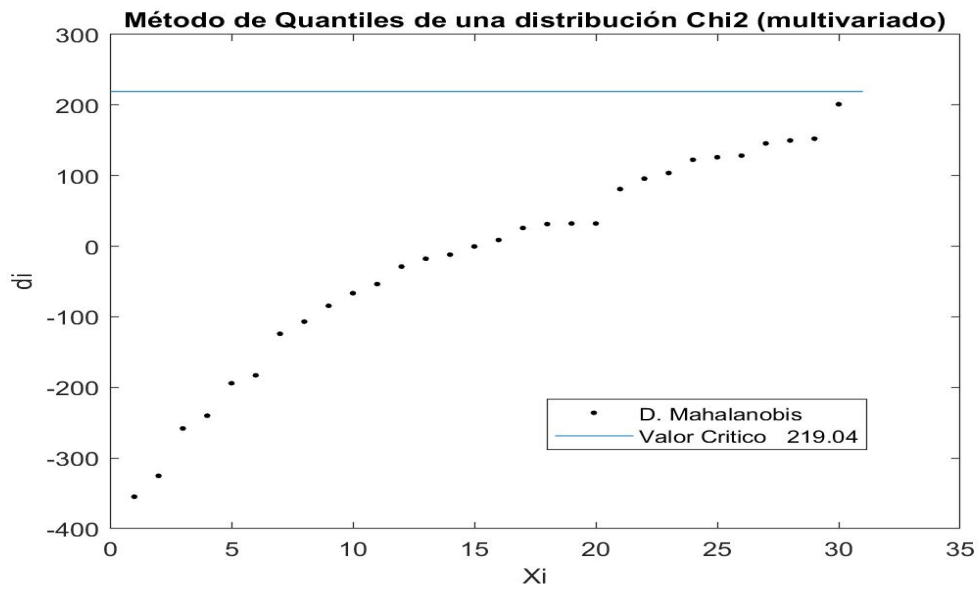
## B.3 Movimientos de Probador 3

### B.3.1 UP

En la Figura B.15a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras UP efectuadas por el Probador 3. Una vez procesadas estas muestras por el Método de Quantiles de una distribución Chi cuadrada para la detección de outliers, se puede apreciar que no se detectó ningún outlier para este subconjunto. (Véase la Figura B.15b).



(a) Distancias



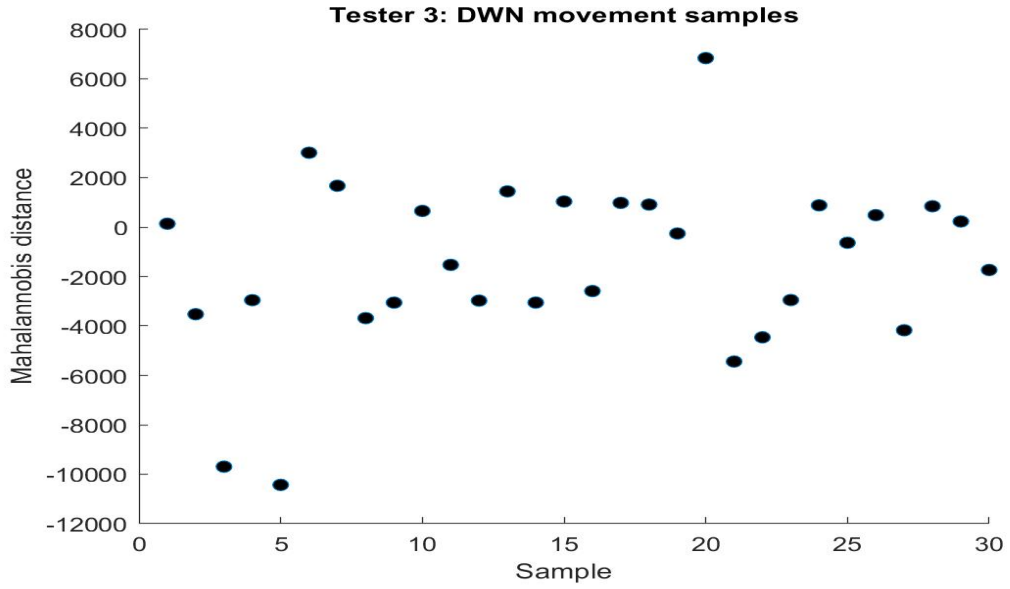
(b) Outliers

Figura B.15: Detección de outliers en el conjunto de muestras UP efectuadas por el Probador 3.

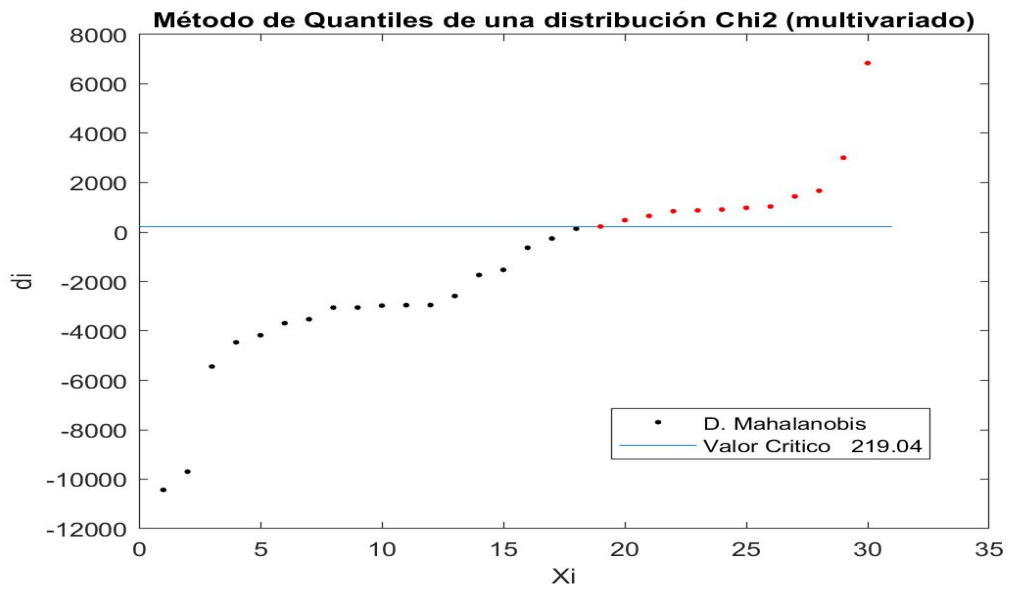
La consistencia en los movimientos realizados por este probador se manifiestan en una detección de outliers nula.

### B.3.2 DWN

En la Figura [B.16a](#), se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras DWN efectuadas por el Probador 3. Una vez procesadas estas muestras por el Método de Quantiles de una distribución Chi cuadrada para la detección de outliers, se pudieron detectar doce outlier para este subconjunto de datos (Véase la Figura [B.16b](#)).



(a) Distancias



(b) Outliers

Figura B.16: Detección de outliers en el conjunto de muestras DWN efectuadas por el Probador 3.

Estas anomalías detectadas se muestran en la Figura B.17 con líneas rojas.

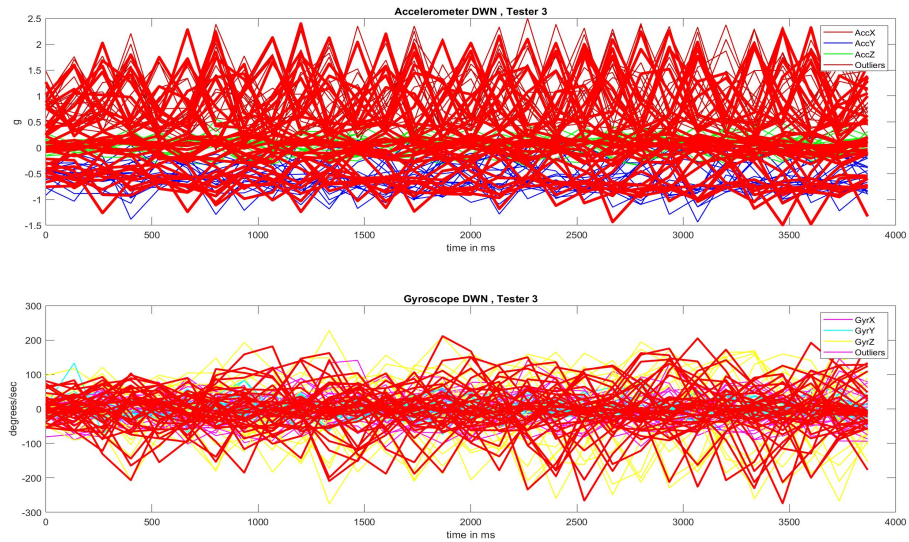
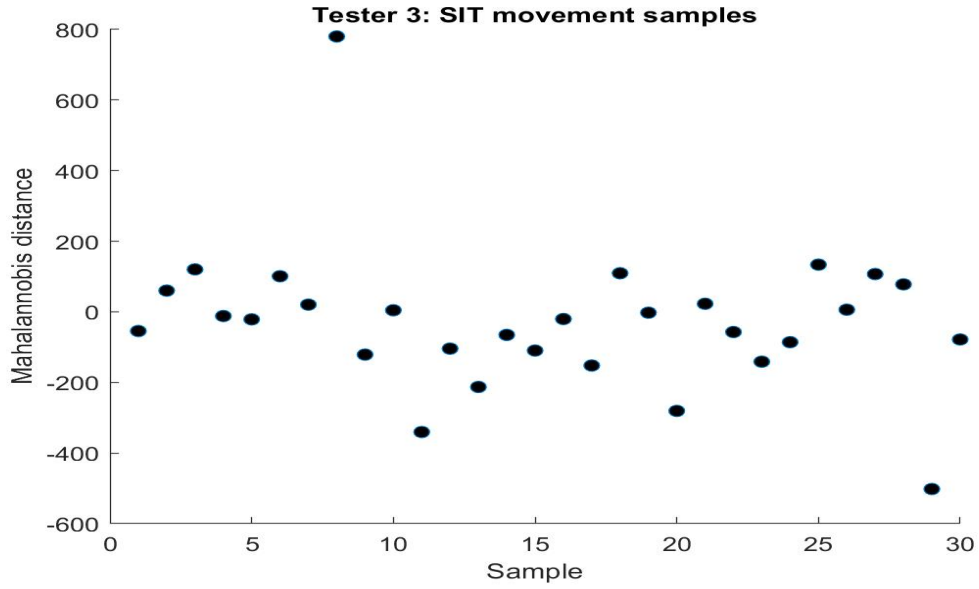


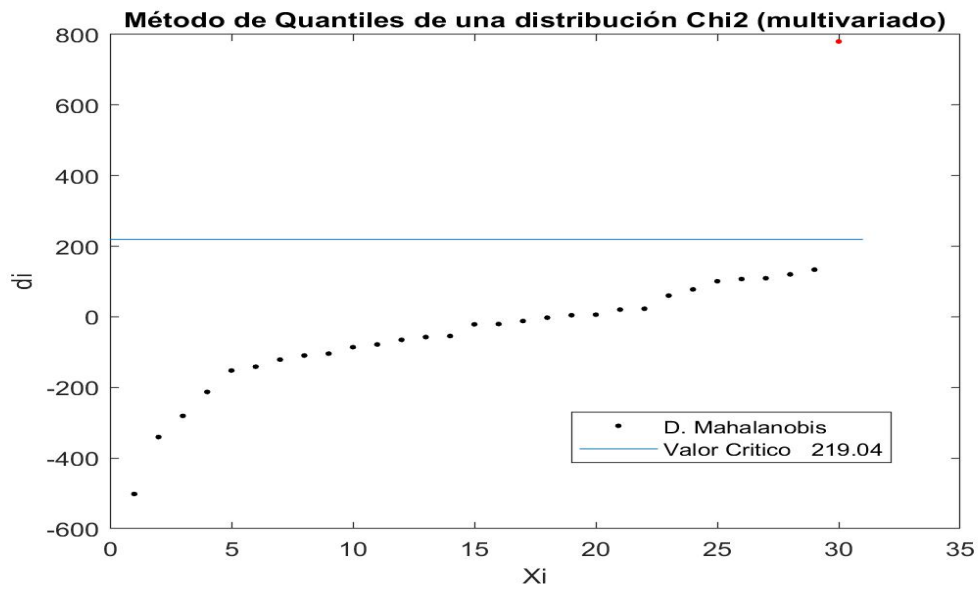
Figura B.17: Outliers detectados para el movimiento DWN efectuado por el Probador 3

### B.3.3 SIT

En la Figura B.18a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras SIT efectuadas por el Probador 3. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se pudo detectar únicamente un outlier para este subconjunto de datos (Véase la Figura B.18b).



(a) Distancias



(b) Outliers

Figura B.18: Detección de outliers en el conjunto de muestras SIT efectuadas por el Probador 3.

Estas anomalías detectadas se muestran en la Figura B.19 con líneas rojas.

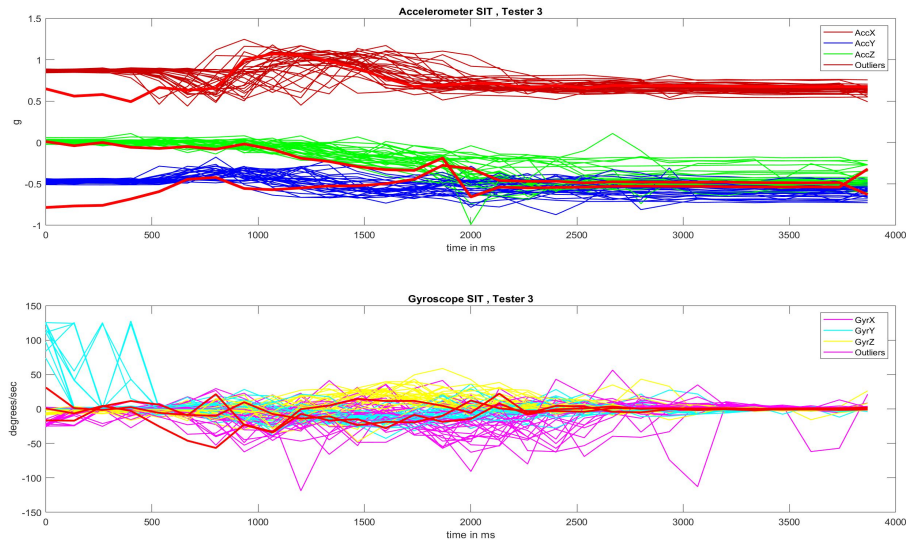
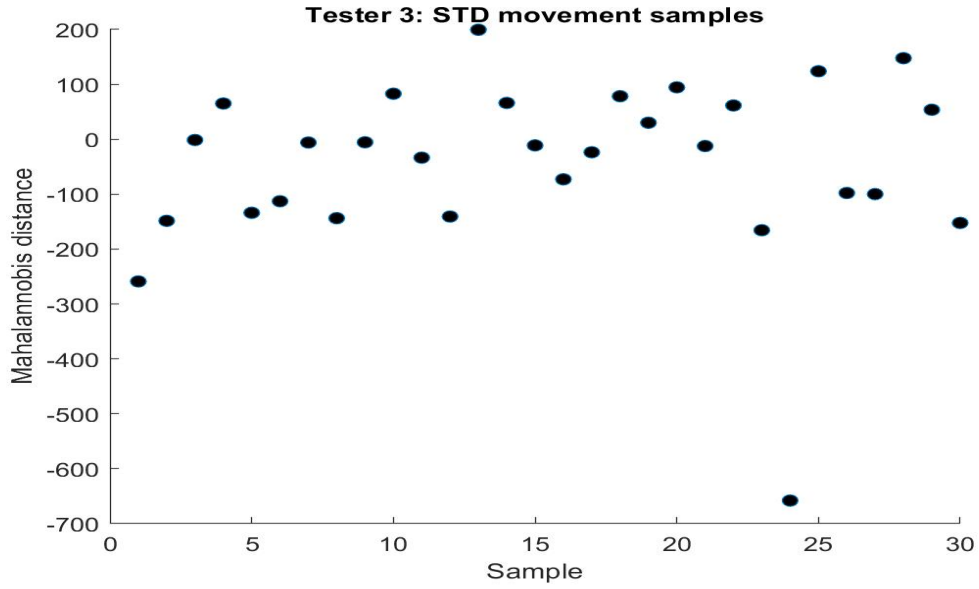


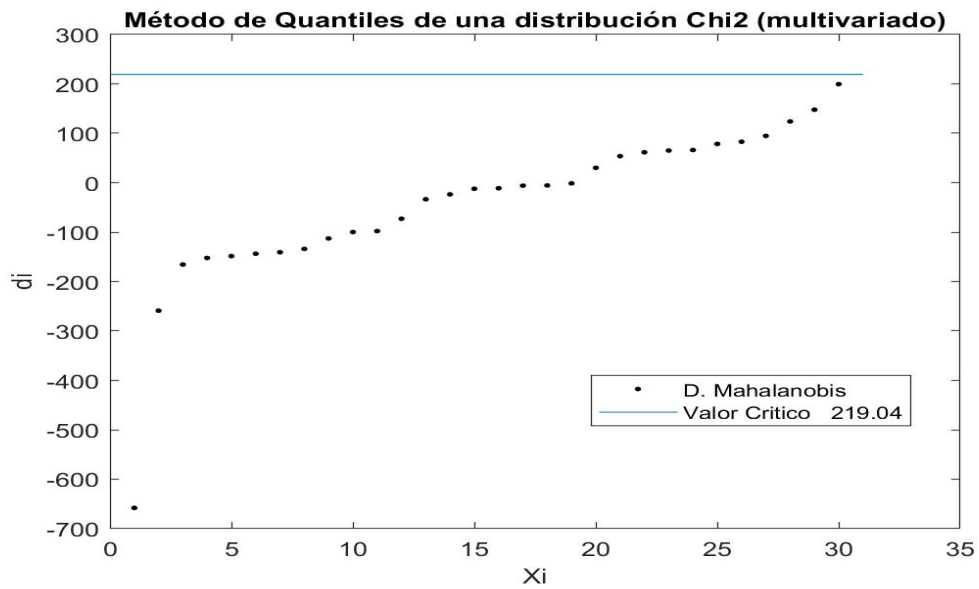
Figura B.19: Outliers detectados para el movimiento SIT efectuado por el Probador 3

### B.3.4 STD

En la Figura B.20a, se muestran graficadas las distancias Mahalannobis correspondientes a cada una de las muestras STD efectuadas por el Probador 3. Una vez procesadas estas muestras por el Método de Cuantiles de una distribución Chi cuadrada para la detección de outliers, se puede apreciar que igual que en la clase UP de este probador, no se detectó ningún outlier para este subconjunto. (Véase la Figura B.20b).



(a) Distancias



(b) Outliers

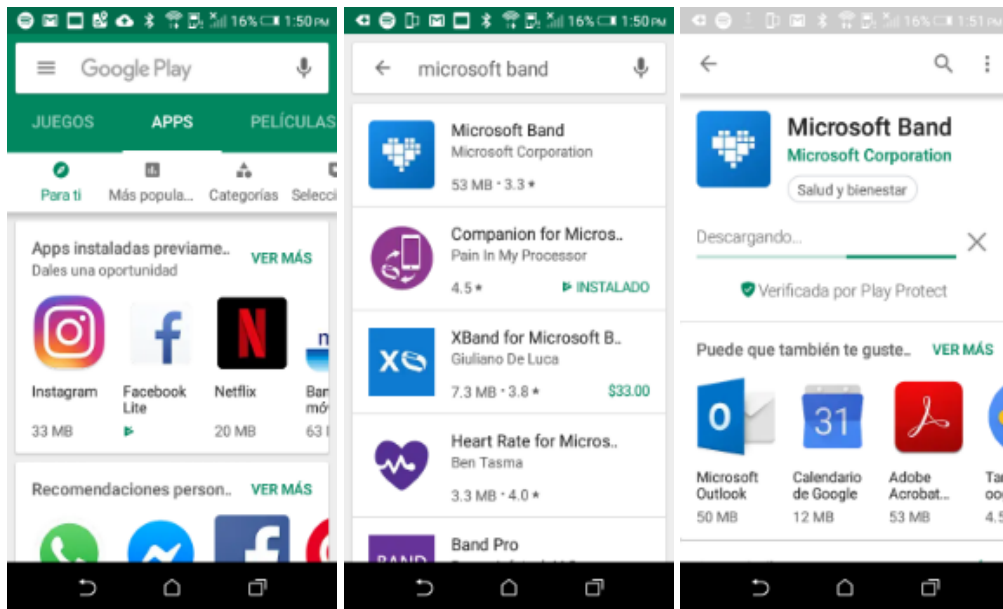
Figura B.20: Detección de outliers en el conjunto de muestras STD efectuadas por el Probador 3.

## Anexos C

# Anexos del Software MyBand

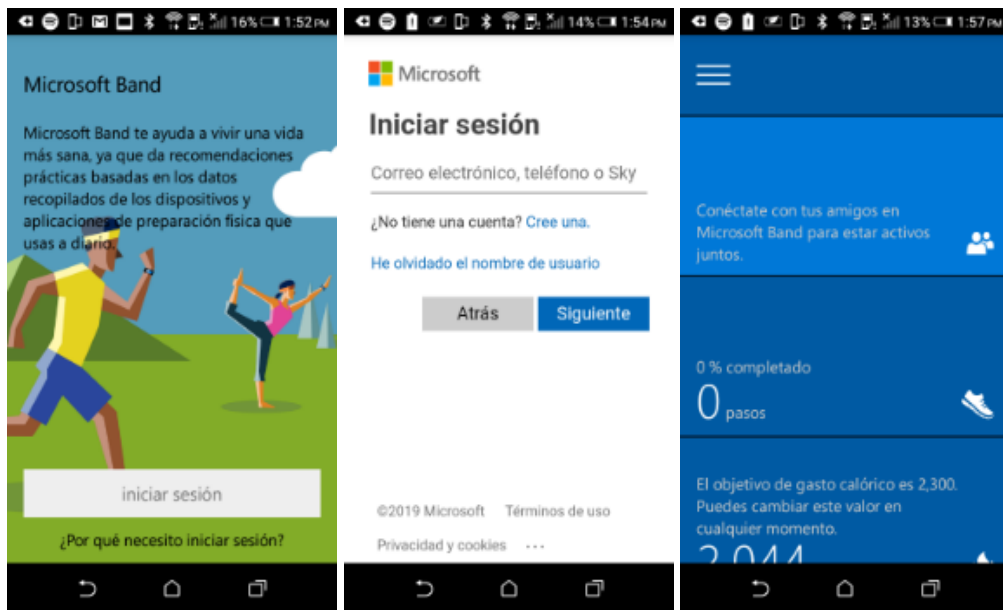
Se presenta el diseño de navegación que los usuarios realizarán con la aplicación. Esto permite tener conocimiento la vista del flujo de navegación entre las interfaces y la relación entre ellas. La Figura [C.1](#) presenta las interfaces más relevantes y módulos principales que permiten cubrir la funcionalidad.

## C.0.1 Escenarios de uso

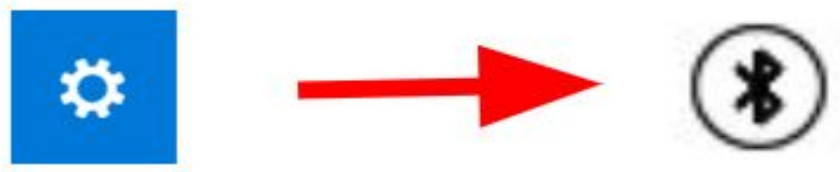


(a) Actividad principal de Google Play (b) Búsqueda de Microsoft Band App (c) Descarga de Microsoft Band App

Figura C.1: Escenarios de uso relacionados con REQ3 y REQ4

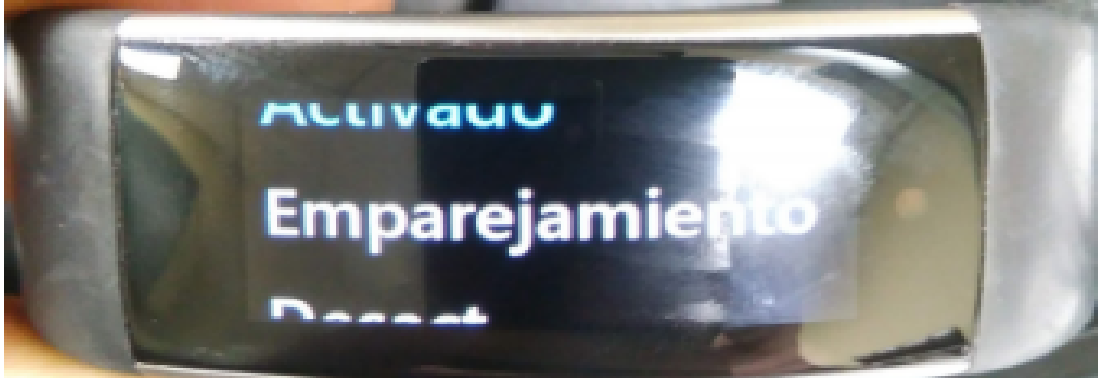


(a) Actividad de inicio de Microsoft Band App (b) Actividad para inicio de sesión de Microsoft Band App (c) Actividad principal de Microsoft Band App

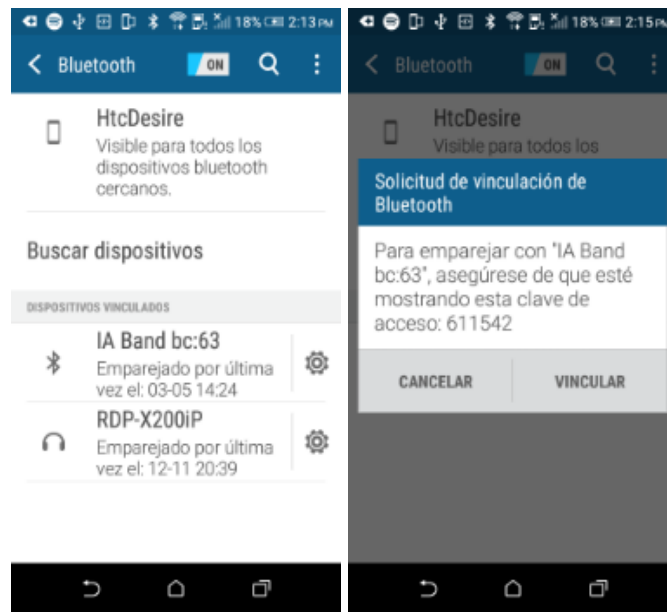


(d) Configuración bluetooth de la MSBand

Figura C.2: Escenarios de uso relacionados con REQ5-10

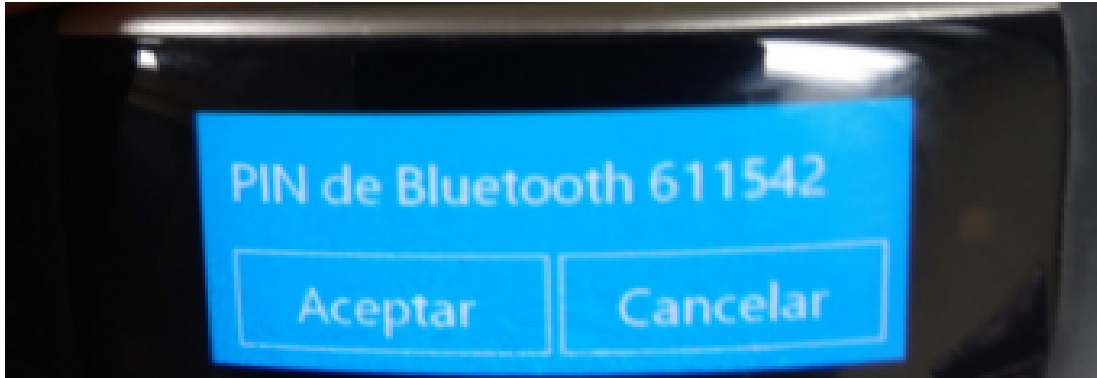


(a) Modo de emparejamiento en REQ10

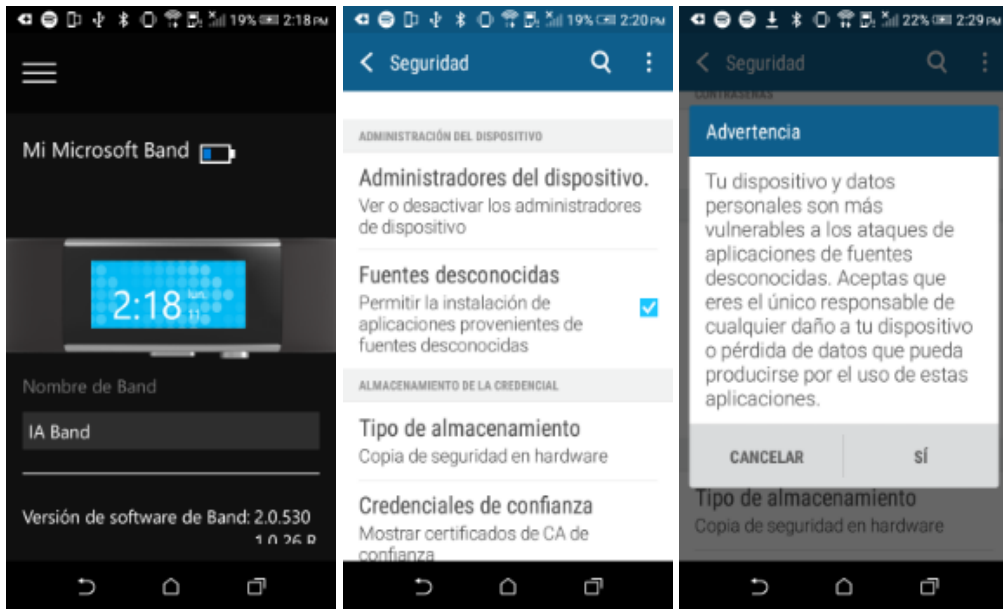


(b) Encendido de Bluetooth en celular  
(c) Vinculación de celular con MSBand

Figura C.3: Escenarios de uso relacionados con REQ11-12

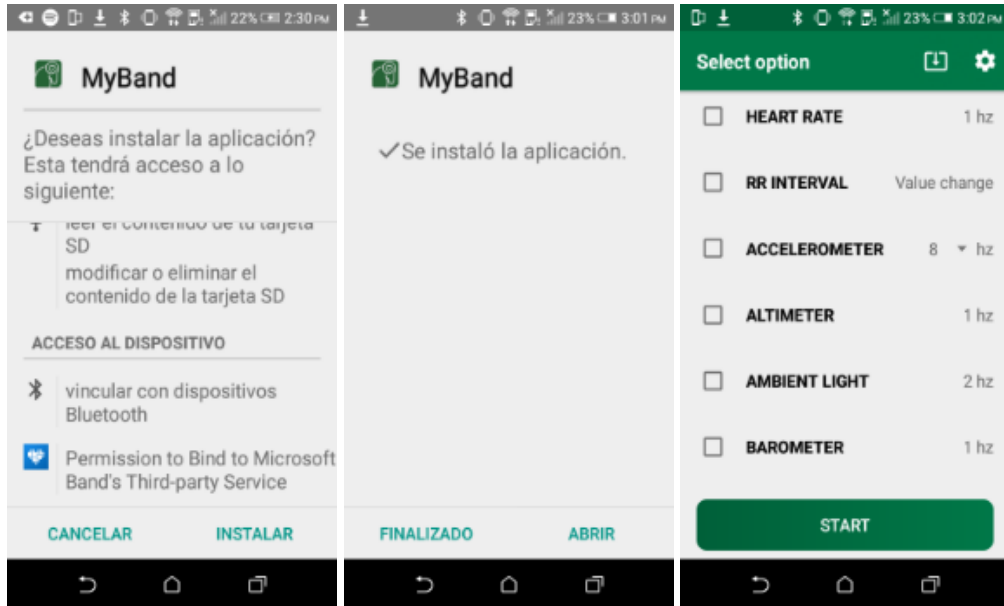


(a) Vinculación Bluetooth en MSBand



(b) Opción My Microsoft band en Microsoft Band App  
 (c) Opción de permitir la instalación de apps provenientes de fuentes desconocidas.  
 (d) Aprobación de permisos para instalación de apps provenientes de fuentes desconocidas.

Figura C.4: Escenarios de uso relacionados con REQ12-17



(a) Instalación de My- (b) Confirmación de insta- (c) Actividad principal de  
Band App lación de MyBand App MyBand App

Figura C.5: Escenarios de uso relacionados con REQ18

---

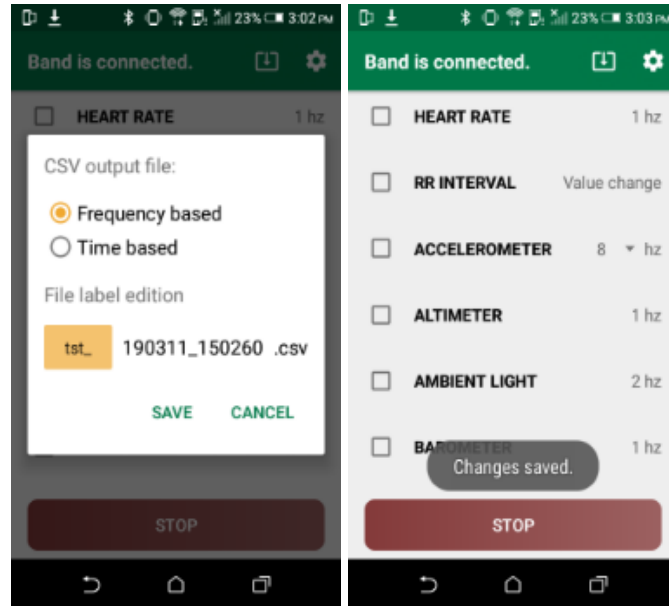
## C.0.2 Flujos Básicos

<b>Id</b>	<b>Descripción</b>	<b>Escenario</b>
FB1.1	En la actividad principal de MyBand, entrar al menú de configuraciones (Indicado con la imagen de un engrane en la barra superior)	Fig. C.5c
FB1.2	Dentro de la sección <i>File label edition</i> , configurar el nombre del archivo con las lecturas de los sensores a exportar	Fig. C.6a
FB1.3	Presionar el botón <i>Save</i>	
FB1.4	Visualizar mensaje de confirmación de configuración de archivo	Fig. C.6b

Tabla C.1: Subflujo FB1

<b>Id</b>	<b>Descripción</b>	<b>Escenario</b>
FB2.1	En la App My band, entrar al menú de configuraciones (Indicado con la imagen de un engrane en la barra superior)	Fig. C.5c
FB2.2	Dentro de la sección <i>CSV output file</i> , seleccionar la opción deseada para el formato del archivo con las lecturas de los sensores a exportar presionando el <i>radiobutton</i> que aparece a la izquierda del nombre con la opción.	Fig. C.6a
FB2.3	Presionar el botón <i>Save</i>	
FB2.4	Visualizar mensaje de confirmación de configuración de archivo	Fig. C.6b

Tabla C.2: Subflujo FB2



(a) Configuración de archivo CSV a exportar

(b) Confirmación de configuración de archivo CSV a exportar

Figura C.6: Escenarios de uso relacionados con Subflujo FB2

---

<b>Id</b>	<b>Descripción</b>	<b>Escenario</b>
FB3.1	Colocar la MSBand en muñeca izquierda, asegurándose de posicionar la pantalla en la parte interna	Fig. 4.7
FB3.2	Elegir los sensores de interés de la pantalla principal de My band en los <i>checkbox</i> correspondientes	
FB3.3	¿Se visualizan en amarillo los checkbox de los sensores seleccionados?	
FB3.4	Si no, existe un problema técnico reportar a correo: marcial.sandoval@uabc.edu.mx	
FB3.5	Si, todo es correcto	Fig. 4.4a
FB3.6	Leer los valores arrojados por los sensores elegidos seleccionando la opción <i>Start</i>	
FB3.7	Después de 5 segundos, ¿Las lecturas de los sensores aparecieron abajo del nombre del sensor ?	
FB3.8	Si no, existe un problema técnico reportar a correo: marcial.sandoval@uabc.edu.mx	
FB3.9	Si, todo es correcto	Fig. 4.4b

Tabla C.3: Subflujo FB3

### C.0.3 Casos de Prueba: Flujos Positivos

La descripción detallada de los atributos de cada registro dentro de las tablas exportadas en formato Frequency Based y Time Based se presenta a continuación en la Tabla C.4.

---

Columna	Descripción
A	Clave de identificación única de cada registro
B	Año en el que fue tomada la muestra
C	Mes en el que fue tomada la muestra
D	Día en el que fue tomada la muestra
E	Hora en el que fue tomada la muestra
F	Minuto en el que fue tomada la muestra
G	Segundo en el que fue tomada la muestra
H	Clave de identificación única de la frecuencia de muestreo a la que trabaja el sensor que arrojó la muestra
I	Clave de identificación única del sensor que arrojó la muestra
J	Valor 1 arrojado por el sensor
K	Valor 2 arrojado por el sensor (En caso de aplicar)
L	Valor 3 arrojado por el sensor (En caso de aplicar)

Tabla C.4: Defición de columnas en tabla exportada en formato Frequency Based y Time Based

Las columnas K y L aplican en los valores arrojados por los sensores con mas de un valor de salida, e.g. acelerómetro (con tres valores, aceleración en los ejes X, Y y Z), giroscopio (con tres valores, la velocidad angular en los ejes X, Y y Z), etc.

Dado que para el caso del formato Sample Based, los valores arrojados por todos los sensores seleccionados se concatenarán cada registro, se definió el orden de aparición de éstos siguiendo su clave única de identificación señalada en la Tabla 4.3. La descripción detallada de los atributos de cada registro dentro de las tablas exportadas en formato Sample Based se presenta a continuación en la Tabla C.5.

Columna	Descripción
A	Tiempo transcurrido desde el comienzo del registro de valores
B-Z	Valores arrojados por los sensores seleccionados

Tabla C.5: Defición de columnas en tabla exportada en formato Sample Based

En la Figura C.7, se muestra el archivo exportado para Frequency Based CSV, el cual corresponde a los valores arrojados por sensores cuya frecuencia de muestreo es de 1hz y en la Figura C.8, de 2 hz, como se indica en la terminación del nombre del archivo 'SR1' (Sample Rate 1) y 'SR2' (Sample Rate 2) .

	A	B	C	D	E	F	G	H	I	J	K
1	690	2019	04	01	16	09	27	1	1	61	2
2	693	2019	04	01	16	09	28	1	1	69	2
3	696	2019	04	01	16	09	29	1	1	65	2
4	699	2019	04	01	16	09	30	1	1	65	2
5	702	2019	04	01	16	09	31	1	1	65	2
6	705	2019	04	01	16	09	32	1	1	65	2
7	708	2019	04	01	16	09	33	1	1	65	2
8	711	2019	04	01	16	09	34	1	1	66	2
9	714	2019	04	01	16	09	35	1	1	65	2
10	717	2019	04	01	16	09	36	1	1	67	2
11	720	2019	04	01	16	09	37	1	1	67	2
12	723	2019	04	01	16	09	38	1	1	67	2
13	726	2019	04	01	16	09	39	1	1	72	2
14	729	2019	04	01	16	09	40	1	1	71	2

Figura C.7: Caso de prueba Frequency Based CSV - 1 hz

	A	B	C	D	E	F	G	H	I	J
1	689	2019	04	01	16	09	27	2	5	173
2	691	2019	04	01	16	09	27	2	5	173
3	692	2019	04	01	16	09	28	2	5	173
4	694	2019	04	01	16	09	28	2	5	172
5	695	2019	04	01	16	09	29	2	5	172
6	697	2019	04	01	16	09	29	2	5	170
7	698	2019	04	01	16	09	30	2	5	170
8	700	2019	04	01	16	09	30	2	5	170
9	701	2019	04	01	16	09	31	2	5	168
10	703	2019	04	01	16	09	31	2	5	172
11	704	2019	04	01	16	09	32	2	5	172
12	706	2019	04	01	16	09	32	2	5	172
13	707	2019	04	01	16	09	33	2	5	172
14	709	2019	04	01	16	09	33	2	5	172

Figura C.8: Caso de prueba Frequency Based CSV - 2 hz

En la Figura C.9, se muestra el archivo exportado para Time Based CSV, como se indica en la terminación del nombre del archivo 'TB' (Time Based), cuyos registros corresponden a los valores arrojados por sensores por orden de

identificador único. Los registros contenidos siguen el modelo presentado en la Tabla C.4.

	A	B	C	D	E	F	G	H	I	J	K
1	736	2019	04	01	16	09	57	2	5	178	
2	737	2019	04	01	16	09	58	2	5	180	
3	738	2019	04	01	16	09	58	1	1	72	2
4	739	2019	04	01	16	09	58	2	5	178	
5	740	2019	04	01	16	09	59	2	5	178	
6	741	2019	04	01	16	09	59	1	1	72	2
7	742	2019	04	01	16	09	59	2	5	178	
8	743	2019	04	01	16	10	00	2	5	176	
9	744	2019	04	01	16	10	00	1	1	71	2
10	745	2019	04	01	16	10	00	2	5	176	
11	746	2019	04	01	16	10	01	2	5	176	
12	747	2019	04	01	16	10	01	1	1	69	2
13	748	2019	04	01	16	10	01	2	5	176	
14	749	2019	04	01	16	10	02	2	5	178	

Figura C.9: Caso de prueba Time Based CSV

En el caso de Sample Based CSV, para producir un registro en la tabla guardada en el archivo de salida, lo que el algoritmo hace es extraer los registros guardados en la tabla `readings.table` después del momento en que se presionó el botón 'Start' y antes del momento en el que se dejó de guardar los registros presionando el botón 'Stop'. Una vez hecho esto, obtiene la frecuencia de muestreo mas alta, calcula el promedio de los valores arrojados por los sensores seleccionados que trabajan a esta frecuencia (produciendo un *downsampling*) y lo concatena con los valores arrojados por los sensores que trabajan a una frecuencia de muestreo más baja.

En el formato de salida de este modo, se tiene la clave de identificación único de cada registro (`_id`), el timestamp de cada muestra representado como la cantidad de milisegundos transcurridos desde el momento que se presionó el botón de 'guardar muestra' (Columna A), los valores arrojados por los sensores seleccionados (columnas B-D), los cuales aparecerán en el orden establecido por su `sensor_id`, señalados en la Tabla 4.2.

Para el ejemplo mostrado en la Figura C.10, aparecen los valores del sensor `ALTIMETER_SENSOR` con `sensor_id` 4 en la columna B, los valores del sensor `AMBIENT_LIGHT_SENSOR` con `sensor_id` 5 en la columna C y los valores del sensor `GSR_SENSOR` con `sensor_id` 7 en la columna D (las claves únicas de identificación de los sensores antes mencionadas están contenidas en la la Tabla C.4).

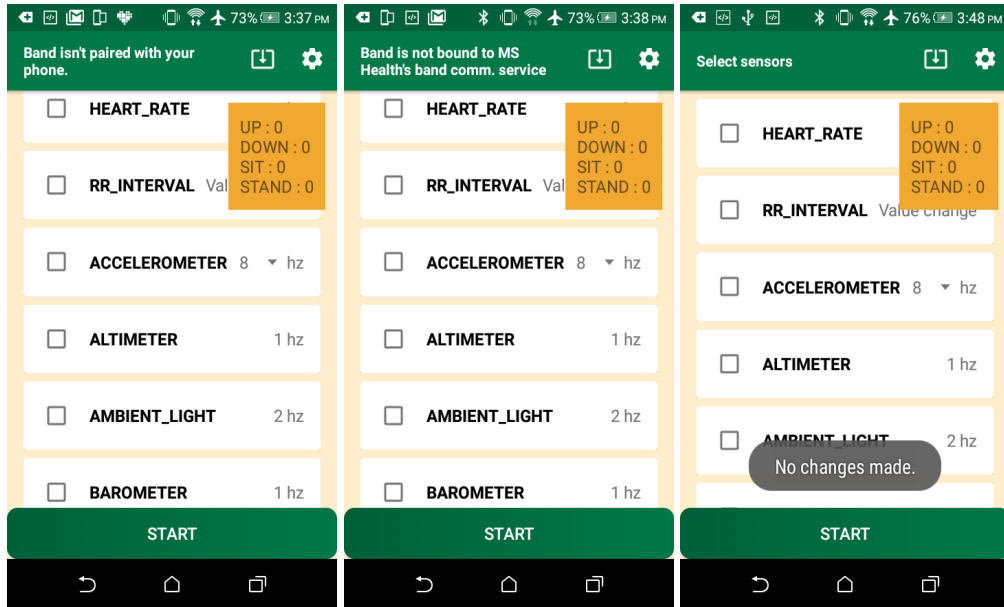
---

	A	B	C	D	E
1	0	25161	235	null	
2	523	null	187	null	
3	1020	25161	274	null	
4	1553	null	251	null	
5	2077	25161	240	null	
6	2501	null	240	null	
7	3043	25161	243	null	
8	3516	null	245	340330	
9	4015	25161	246	null	
10	4529	null	240	null	
11	5009	25161	243	null	
12	5508	null	243	null	
13	6037	25161	244	null	
14	6503	null	243	null	
15	7004	25161	243	null	
16	7532	null	244	null	
17	8000	25161	246	null	
18	8553	null	244	340330	
19	9035	25161	246	null	

Figura C.10: Caso de prueba Sample Based CSV

## C.0.4 Sublujos Alternativos

### C.0.4.1 Escenarios de uso



(a) Falla en comunicación (b) Falla en comunicación (c) Usuario guarda configuración sin efectuar cambios entre el smart-phone y la MSBand entre la MSBand y la aplicación Microsoft Band sin algún cambio

Figura C.11: Casos de Prueba alternativos

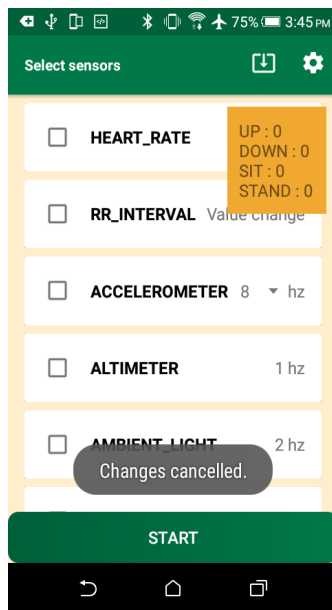


Figura C.12: Usuario cancela configuraciones

# Referencias

- ANGUITA, D., GHIO, A., ONETO, L., PARRA, X. & REYES-ORTIZ, J. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Elementary particle theory*, Springer. 8
- BECK, K. (2005). Extreme programming explained, embrace change. 27–28, Addison–Wesley Professional. 16
- BRADLEY, A.P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, **30**, 1145–1159. 29, 30
- BREIMAN, L. (2001). Random forests. In *Machine Learning 45*, 5–32, Almqvist and Wiksell. 25
- BULLING, A., BLANKE, U. & SCHIELE, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, **46**. 2, 3, 4, 8
- CORTES C., V.V. (1995). Support–vector networks. In *Mach Learn 20*, 273–297. 23
- GONZÁLEZ-CANETE, F.J. & CASILARI, E. (2020). Fconsumption analysis of smartphone based fall detection systems with multiple external wireless sensors. In *Sensors*. 1
- HAWKINS, D.M. (1980). In *Identification of outliers*, Chapman and Hall. 21
- HUYNH T., S.B., BLANKE U. (2007). Scalable recognition of daily activities with wearable sensors. *Springer*, **4718**, 50–67. 7

- JE-MIN, K. & J.W, K. (2012). Androbench: Benchmarking the storage performance of android-based mobile devices. [37](#)
- KWOK, R. (2017). Flexible working: Science in the gig economy. In *Nature* 550, 419—421. [1](#)
- L. CHEN, E.A., J. HOEY (2012). Sensor-based activity recognition. In *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 6, 790–808. [2](#)
- LENAIL (2019). Nn-svg: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, **4(33)**. [69](#)
- MAATEN, L. & HINTON, G. (2008). Visualizing high-dimensional data using t-sne. vol. 9, 2579–2605, Microtome Publishing, pagination: 27. [26](#)
- MACHADO, I.P., GOMES, A.L., GAMBOA, H., AO, V.P. & COSTA, R.M. (2015). Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization. In *Information Processing and Management Volume 51, Issue 2*, 204–214. [7](#)
- MATHWORKS (2020). Matlab homepage. Last accessed 3 September 2020. [32](#)
- MICROSOFT (2015). Microsoft support, microsoft band 2 product specifications. Last accessed 3 February 2019. [11](#)
- PANWAR, M., DYUTHI, S. & PRAKASH, K. (2017). Cnn based approach for activity recognition using a wrist-worn accelerometer. In *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2438–2441. [7](#)
- PEREZ-ZEPEDA, M., CRUZ-ARENAS, E., LOPEZ, M., GONZALEZ LARA, M. & GODOY-CASTELLANOS, I. (2014). Caídas: revisión de nuevos conceptos. *Hospital (Rio de Janeiro, Brazil)*, **13**, 86–95. [1](#)
- SENI, G. & ELDER, J. (2010). Ensemble methods in data mining: Improving accuracy through combining predictions. In *Synthesis Lectures on Data Mining and Knowledge Discovery*. [31](#)

- SHARMA, A.E.A. (2019). Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. In *Sci Rep* 9, 11399. 26, 80
- SHENG B., E.A. (2020). A comparison of different machine learning algorithms, types and placements of activity monitors for physical activity classification. In *Measurement (02632241)*. 7
- STUFFMADEHERE (2020). My robotic basketball hoop wont let you miss. Last accessed 3 August 2020. 2
- VAN DER MAATEN, L. & HINTON, G. (2008). Viualizing data using t-sne. *Journal of Machine Learning Research*, 9, 2579–2605. 31
- WANG, S., YANG, J., CHEN, N., CHEN, X. & ZHANG, Q. (2005). Human activity recognition with user-free accelerometers in the sensor networks. In *2005 International Conference on Neural Networks and Brain*, 1212–1217. 7
- WANG, X.Z. (1999). *Multivariate Statistical Analysis for Data Analysis and Statistical Control*, 61–84. Springer London, London. 21, 25, 31
- WITTEN, I.H., FRANK, E. & HALL, M.A. (2011). Chapter 1 - what's it all about? In I.H. Witten, E. Frank & M.A. Hall, eds., *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, 3 – 38, Morgan Kaufmann, Boston, third edition edn. 50
- ZHU, XIAO, Z., CHENG, M., ZHOU, L., YAN, B. & LIN, S.W.H. (2018). Deep ensemble learning for human activity recognition using smartphone. 1–5. 7