



UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS
DEPARTAMENTO DE FÍSICA
MÉRIDA ESTADO MÉRIDA

**“Hacia una Detección Temprana de la Enfermedad de Parkinson a
Través del Análisis de Componentes Principales (PCA) y Redes
Neuronales Artificiales (RNA)”.**

[Una Propuesta Experimental]

Trabajo Especial de Grado como requisito para optar al
Título de Licenciado en Física

Autor: Br. Stephanie Carrillo Vielma

Tutor: Dr. Misael D. Rosales

Marzo, 2021

“Hacia una Detección Temprana de la Enfermedad de Parkinson a Través del Análisis de Componentes Principales (PCA) y Redes Neuronales Artificiales (RNA)”.

[Una Propuesta Experimental]

La defensa de esta tesis fue realizada en forma virtual a través de una presentación grabada en un vídeo que fue enviado al jurado junto con el documento de la tesis, debido a que la pandemia generada por el COVID-19 no permitió cumplir con el protocolo formal de presentación estándar que exige el Departamento de Física de la Universidad de los Andes. Las preguntas formuladas por el jurado fueron debidamente respondidas y se anexan en el apéndice **preguntas y observaciones del jurado** de este documento.

Jurado:

Dr. John Ferreira Rodríguez
C.I. V - 8.042.037
Proyectos CIAE-TELMAG
correo: jdferreira1@protonmail.com

Dr. Marcos Rodríguez Morales
C.I. V- 6.019.789
CITEC-ULA
correo: mrrrodmar@gmail.com

*Somos polvo de estrellas que piensa acerca de las estrellas.
Somos la forma en la que el Universo se piensa a sí mismo.*
Carl Sagan

www.bdigital.ula.ve

Agradecimientos

A mis padres, por su apoyo incondicional, dedicación y fortaleza, ustedes son el motor de mis proyectos. Gracias mamá, gracias papá por todo el amor. Los amo.

A mi hermano José Leonardo, por todos los consejos dados, ser un ejemplo a seguir y estar siempre presente incluso en la distancia. Te quiero.

A mi abuela Dora, por sus cuidados y oraciones que me han acompañado. Gracias.

A mi tía Thais, sin su ayuda y cariño todo sería más difícil.

A mi tutor el profesor Dr. Misael D. Rosales por guiarme en este trabajo, por la paciencia, las enseñanzas y las contribuciones. Muchas gracias por la confianza y el tiempo dedicado.

Al profesor Carlos Guada, por sus consejos y ayuda en el manejo de los equipos.

Al jurado, profesores Marcos Rodríguez y John Ferreira, gracias por las aportaciones.

A la Universidad de los Andes, por ser el Alma Mater y la casa de estudio de la cual estoy orgullosa de pertenecer.

A Antonio por estar siempre allí y ser parte significativa de mi vida. Gracias por tu amistad.

A mis amigos: Gustavo, Samuel, Moisés y José Francisco, con quienes compartí tantos momentos especiales durante este camino.

A mis compañeros del laboratorio: Sandra, Flor, Ignacio, Neptali y Juan Carlos, gracias por sus valiosas aportaciones y por el apoyo recibido estos últimos años.

A las personas que formaron parte de este estudio, ustedes son la razón del mismo.

A todos aquellos que de alguna u otra manera estuvieron presentes, muchas gracias por todo.

Stephanie



**“Hacia una detección temprana de la Enfermedad de Parkinson a través del Análisis de Componentes Principales (PCA) y Redes Neuronales Artificiales (RNA).
[Una Propuesta Experimental]”.**

Trabajo especial de grado como requisito para optar al Título de Licenciado en Física

Autor: Stephanie Carrillo Vielma

Tutor: Dr. Misael D. Rosales

Mérida-Venezuela, marzo de 2021

Resumen

La enfermedad de Parkinson es un trastorno neurodegenerativo que afecta predominantemente a personas de edad adulta, entre 50 y 60 años. La complejidad en el diagnóstico de la enfermedad representa un desafío para los médicos, ya que en muchos casos es difícil de diagnosticar con precisión en las primeras etapas. Se presenta un sistema para detectar tempranamente la enfermedad de Parkinson a través de redes neuronales artificiales (RNA), haciendo uso de un perceptrón multicapa de propagación hacia atrás, para la búsqueda y clasificación de patrones de la dinámica corporal. El sistema integra 5 etapas: en la primera se construye una base de datos que se obtiene con un sistema microelectromecánico, constituido por un acelerómetro, un giróscopo, un magnetómetro y un pulsómetro, conectados a una placa Arduino Due; a través de ocho tareas que incorporan movimientos lineales y circulares previamente seleccionadas se extraen las características dinámicas de 60 sujetos sanos en edades comprendidas entre los 20 y 74 años y de 6 pacientes con la enfermedad de Parkinson en sus primeros estadios (I y II) con un promedio de edad de 60 años. La segunda etapa corresponde al tratamiento de los datos empleando el software MATLAB, reduciéndolos a una misma longitud, se estandarizan para regularizar las escalas dinámicas, se eliminan los valores atípicos para pasar a la tercera etapa que corresponde a la reducción de la dimensionalidad aplicando un análisis de componentes principales (PCA), etiquetando a los sujetos para crear entonces la base de conocimiento, siendo ahora estos los datos de entrada a la red neuronal. En la cuarta etapa se establece la arquitectura de la red, se inicia el entrenamiento de la misma, se obtiene la matriz de confusión y los parámetros de la red, que se utilizan para la proyección del grupo patológico. En la quinta etapa se presentan los dendrogramas de clasificación en términos de distancia de Mahalanobis por tarea y combinaciones entre ellas. Se obtiene una eficiencia del 100 % cuando se emplean todas las tareas y todos los parámetros. Finalmente, se proyectan 15 nuevos sujetos sanos y se obtiene una clasificación del 73.3 % para sujetos con características dinámicas similares al grupo sano, mientras que el restante posee características dinámicas similares al grupo patológico, siendo sujetos potenciales que pueden padecer Parkinson en algún momento de su vida.

Palabras claves: Enfermedad de Parkinson, Arduino, sensores, análisis de componentes principales (PCA), redes neuronales artificiales (RNA), perceptrón multicapa, dendrograma, distancia de Mahalanobis.



**“Towards early detection of Parkinson’s disease through Principal Component Analysis (PCA) and Artificial Neural Networks (ANN).
[An Experimental Proposal]”.**

Degree thesis as a requirement to qualify for the Bachelor’s Degree in Physics.

Author: Stephanie Carrillo Vielma

Tutor: Dr. Misael D. Rosales

Mérida-Venezuela, marzo de 2021

Abstract

Parkinson’s disease is a neurodegenerative disorder that predominantly affects people in adulthood, between 50 and 60 years of age. The complexity in diagnosing the disease represents a challenge for clinicians, as in many cases it is difficult to diagnose accurately in the early stages. A system is presented to detect Parkinson’s disease early through artificial neural networks (ANN), using a multilayer perceptron, for search and classification patterns presented body dynamics. The system integrates 5 stages: in the first, a database is built that is obtained with a microelectromechanical system (MEMS), consisting of an accelerometer, a gyroscope, a magnetometer and a heart rate monitor, connected to an Arduino Due board; through eight tasks that incorporate previously selected linear and circular movements, the dynamic characteristics of 60 healthy subjects between 20 and 74 years of age and 6 patients with Parkinson’s disease in its first stages (I and II) with an average age of 60 years. The second stage corresponds to the treatment of the data using the MATLAB software, reducing them to the same length, they are standardized to regularize the dynamic scales, the outliers are eliminated to pass to the third stage that corresponds to the reduction of dimensionality applying an analysis of principal components (PCA), labeling the subjects to then create the knowledge base, these being now the input data to the neural network. In the fourth stage, the network architecture is established, its training begins, the confusion matrix and the network parameters are obtained, which are used for the projection of the pathological group. In the fifth stage, the classification dendrograms are presented in terms of Mahalanobis distance per task and combinations between them. A 100 % efficiency is obtained when all the tasks and all the parameters are used. Finally, 15 new healthy subjects are projected and a classification of 73.3 % is obtained for subjects with dynamic characteristics similar to the healthy group, while the rest have dynamic characteristics similar to the pathological group, being potential subjects who may suffer from Parkinson’s at some point in their life.

Keywords: Parkinson’s disease, Arduino, sensors, Principal Component Analysis (PCA), Artificial Neural Networks (ANN), Multilayer Perceptron (MLP), dendrogram, Mahalanobis distance.

Introducción

La enfermedad de Parkinson (EP), es la segunda enfermedad neurodegenerativa más común después de la enfermedad de Alzheimer. Las neuronas dopaminérgicas producen una sustancia química denominada dopamina, la cual envía mensajes a la parte del cerebro que controla los movimientos y la coordinación, ésta también actúa como un mensajero entre dos áreas del cerebro, la sustancia negra y el cuerpo estriado, responsable de los movimientos suaves controlados. Los signos clínicos iniciales de la EP aparecen después de la degeneración de aproximadamente el 60 % de las neuronas dopaminérgicas de la sustancia negra, dando lugar a la disfunción motora que se manifiesta con los cuatro síntomas primarios: temblor en reposo, bradicinesia, rigidez muscular e inestabilidad postural [1, 2, 3].

Según la OMS, se estima que cerca de 7 millones de personas alrededor del mundo padecen la enfermedad de Parkinson, afectando principalmente a personas de edades entre los 50 y 60 años, sin embargo actualmente están apareciendo cada vez más casos en personas menores de 50 años, debido a factores externos como el estrés. A pesar de que se han identificado los factores de riesgo genéticos y ambientales, que contribuyen al desarrollo de la enfermedad, aún no se entiende los mecanismos patogénicos exactos que subyacen a la pérdida selectiva de células dopaminérgicas. El diagnóstico de la EP se basa en la observación clínica, no hay marcadores biológicos ni de imágenes definidos, lo que implica que la exactitud del diagnóstico en etapas tempranas está relacionado directamente con las habilidades y la experiencia del médico tratante [1, 4].

Para el diagnóstico el neurólogo analiza el historial médico completo del paciente y realiza numerosas evaluaciones clínicas para confirmar la presencia de EP en ese sujeto y para descartar otras enfermedades. Después de un riguroso estudio de la historia clínica a veces se puede tardar hasta un año en diagnosticar la EP, además debido a la falta de medidas objetivas existe una alta posibilidad de hacer un mal diagnóstico. Se ha encontrado que la tasa de diagnósticos erróneos es de alrededor del 25 %, mientras que aproximadamente el 40 % de los casos de enfermedad de Parkinson se pasan por alto como otros trastornos neurológicos [5].

Un diagnóstico temprano y preciso de la EP es necesario para un mejor tratamiento que controle con mayor eficiencia los efectos de los síntomas. En los últimos años los avances tecnológicos han permitido el desarrollo de varios sistemas y dispositivos para la detección y medición de los temblores característicos de la EP. Muchos sensores utilizados para adquirir bioseñales incluyen: acelerómetros, giróscopos y magnetómetros. Por otro lado, el uso de clasificadores en el diagnóstico médico se está incrementando gradualmente. Los sistemas expertos y diferentes técnicas de inteligencia artificial (IA) como algoritmos basados en redes neuronales tienen el potencial de ser buenas herramientas de apoyo para el experto. Los sistemas de clasificación pueden ayudar a aumentar la precisión y fiabilidad de los diagnósticos y minimización de posibles errores [6].

El objetivo de la siguiente investigación es el de realizar la detección temprana de la enfermedad de Parkinson a través de la dinámica corporal del paciente cuando realizan una serie de tareas siguiendo patrones lineales y circulares, los datos de 45 sujetos con edades comprendidas entre los 25 hasta los 74 años son adquiridos a través de un sistema microelectromecánico conformado por un acelerómetro, un magnetómetro, un giróscopo y un pulsómetro conectados todos a una tarjeta Arduino Due. Estos datos se estandarizan, se les realiza un Análisis de Componentes Principales (PCA) para la reducción de dimensionalidad y se construye una base de conocimiento de sujetos sanos que se usa de entrada para el algoritmo de Redes Neuronales Artificiales (RNA), específicamente de un perceptrón multicapa de propagación hacia atrás.

Una vez finalizado el entrenamiento y obtenidos los parámetros de la red, se realiza la proyección del conjunto de datos de 6 pacientes diagnosticados con la enfermedad de Parkinson en sus primeros estadios y empleando técnicas de visualización como los dendrogramas se busca con cual combinación de tareas y parámetros se logra la mejor eficiencia de clasificación entre los dos grupos. Ya obtenida la mejor clasificación se proyecta uno a uno los sujetos del grupo denominado prueba que está conformado por 15 sujetos sanos desconocidos para la red neuronal, con edades comprendidas entre los 25 y 67 años, con el fin de identificar a cual grupo (sano o patológico) es más similar en cuanto a las características dinámicas estudiadas.

www.bdigital.ula.ve

Índice general

1	Planteamiento del problema.	1
1.1	Planteamiento del problema.	1
1.2	Objetivos.	2
1.2.1	Objetivo general.	2
1.2.2	Objetivos específicos.	2
1.3	Justificación.	3
1.4	Estado del arte.	4
1.5	Distribución por capítulos.	8
2	Marco Teórico.	9
2.1	Enfermedad de Parkinson.	9
2.1.1	Causas de la enfermedad.	10
2.1.2	Manifestaciones Clínicas.	11
2.1.3	Diagnóstico.	14
2.1.4	Pronóstico.	15
2.1.5	Tratamiento de la enfermedad.	16
2.1.6	Prevención.	17
2.2	Redes Neuronales Artificiales.	18
2.2.1	Redes neuronales de tipo biológico.	18
2.2.2	Redes Neuronales Artificiales.	19
2.2.3	El Perceptrón.	20
2.2.4	El Perceptrón Multicapa.	21
2.2.5	Arquitectura.	21
2.2.6	La retropropagación.	22
2.2.7	Deducción del algoritmo Backpropagation.	23
2.2.8	Entrenamiento de la red neuronal.	25
2.2.9	Dimensiones de la red.	26
2.2.10	Función de activación de las neuronas.	27
2.2.11	Aplicaciones de un perceptrón multicapa.	28
2.2.12	Problemas de clasificación.	28
2.3	Análisis de Componentes Principales.	29
2.3.1	Autovalores y autovectores.	29
2.3.2	Cálculo de las componentes principales.	30
2.3.3	Proceso de extracción de factores.	30
2.3.4	Gráfico de sedimentación.	33
3	Marco Metodológico.	34

3.1	Diagrama de flujo.	34
3.2	I. Adquisición de los Datos.	36
3.2.1	Hardware.	36
3.2.1.1	Sistemas Microelectromecánicos.	36
3.2.1.2	Sensor: Tarjeta GY-80.	36
3.2.1.3	Giróscopo L3G4200D.	37
3.2.1.4	Acelerómetro ADXL345.	38
3.2.1.5	Magnetómetro HMC5883L.	38
3.2.1.6	Pulsómetro: <i>Pulse sensor</i> .	39
3.2.1.7	Arduino.	39
3.2.1.8	Arduino Due.	40
3.2.2	Software.	40
3.2.2.1	MATLAB.	40
3.2.2.2	Entorno de desarrollo en Arduino.	42
3.2.2.3	Aqevent.	43
3.2.2.4	Características del computador.	44
3.2.3	Montaje Experimental.	44
3.2.4	Población de estudio.	47
3.2.5	Criterios de inclusión y de exclusión.	48
3.2.6	Recogida de los datos.	48
3.2.7	Mediciones básicas.	49
3.2.8	Calibración.	51
3.2.9	Adquisición de los datos.	52
3.3	II. Tratamiento de los datos.	52
3.3.1	Preparación de la data.	52
3.3.2	Normalización y depuración de los datos.	54
3.3.3	Creación de ejemplos.	59
3.3.4	Formato de los datos.	60
3.3.5	Reducción de la dimensionalidad.	61
3.4	III. Aplicación de PCA.	62
3.4.1	Etapa de entrenamiento.	62
3.4.2	Etapa de reconocimiento.	64
3.4.2.1	Matriz de confusión PCA.	66
3.4.2.2	Base de datos y conocimiento.	67
3.5	IV. Entrenamiento de la Red Neuronal.	68
3.6	V. Método de validación de la red.	70
3.6.1	Función de la red.	70
3.6.2	Matriz de confusión RNA.	70
3.6.3	Proyección del paciente desconocido.	71
3.6.4	Clasificación.	72
3.6.5	Dendrograma.	73
4	Resultados y análisis de los datos.	76
4.1	Distribución de los sujetos sanos por grupos etarios.	76
4.2	Calibración del acelerómetro.	77
4.3	Visualización de la data en una tarea.	78
4.4	Preparación de la data.	78

4.5	Análisis dinámico por cada tarea.	80
4.5.1	Frecuencia central de los parámetros por todas las tareas por pacientes.	80
4.5.2	Frecuencia central de los parámetros por tarea y todos los pacientes.	83
4.5.3	Valores medios de los parámetros para todos los pacientes por tarea.	91
4.5.4	Sincronización del movimiento.	94
4.5.4.1	Sincronización para los sujetos sanos.	95
4.5.4.2	Sincronización para los pacientes con Parkinson.	97
4.5.5	Espectro de Fourier por parámetro para cada tarea.	99
4.6	Análisis de Componentes Principales (PCA).	102
4.6.1	Gráfico de sedimentación.	103
4.6.2	Proyección de la imagen de cada sujeto sano sobre el espacio V.	104
4.6.3	Correlación entre los sujetos prueba.	105
4.6.4	Matriz de confusión de PCA.	107
4.7	Entrenamiento de la red neuronal.	108
4.7.1	Arquitectura de la red.	108
4.7.2	Matriz de confusión de RNA.	109
4.8	Clasificación de todos los pacientes por tareas.	110
4.8.1	Dendrograma para la tarea 1.	110
4.8.2	Dendrograma para la tarea 2.	111
4.8.3	Dendrograma para la tarea 3.	112
4.8.4	Dendrograma para la tarea 4.	113
4.8.5	Dendrograma para la tarea 5.	114
4.8.6	Dendrograma para la tarea 6.	115
4.8.7	Dendrograma para la tarea 7.	116
4.8.8	Dendrograma para la tarea 8.	117
4.8.9	Dendrograma para todas las tareas.	118
4.9	Clasificación empleando solo magnetómetro.	123
4.10	Proyección del grupo prueba.	124
5	Discusiones y conclusiones.	140
5.1	Discusiones.	140
5.2	Comparativa con otros trabajos.	142
5.3	Conclusiones.	144
5.4	Recomendaciones.	146
5.5	Futuras líneas de investigación.	146
	Apéndice	147
A	Programas Matlab.	147
A.1	Concatena los archivos txt.	147
A.2	Análisis Componentes Principales.	148
A.3	Redimensiona los datos.	153
A.4	Combinación entre parámetros.	155
A.5	Depuración de datos.	156
A.6	Formato de los datos.	157
A.7	Data de entrenamiento y test.	160
A.8	Etiquetas.	161

A.9 Red Neuronal Backpropagation.	161
A.10 Red Neuronal Perceptronica BP.	163
A.11 Fase de test y producción.	165
A.12 Matriz de Confusión.	166
A.13 Proyección del paciente desconocido.	168
A.14 Dendrograma Parkinson.	171
A.15 Clasificación Parkinson.	171
A.16 Dinámica de los Pacientes.	172
B Programas Arduino.	175
B.1 Todos los sensores.	175
B.2 Calibración acelerómetro ADXL345.	176
C Abreviaturas.	182
D Glosario.	183
E Fichas técnicas (Datasheets).	185
F Preguntas y observaciones del jurado.	194
Bibliografía	198

www.bdigital.ula.ve

Índice de figuras

2.1	Corte transversal del Mesencéfalo.	9
2.2	Ilustración etiquetada que muestra niveles del neurotransmisor dopamina	10
2.3	Estructura general de una neurona.	18
2.4	Estructura general de una neurona artificial.	19
2.5	Modelo de McCulloch-Pitts para una neurona artificial.	21
2.6	Arquitectura de un perceptrón multicapa	22
2.7	Función tangente hiperbólica y función sigmoideal	27
2.8	Gráfico de sedimentación para 10 autovalores.	33
3.1	Diagrama de flujo.	35
3.2	Tarjeta GY-80.	37
3.3	Pulsómetro para Arduino.	39
3.4	Placa Arduino Due.	40
3.5	Entorno de desarrollo en Arduino	42
3.6	Ejemplo de ejecución de Aqevent.	43
3.7	Montaje experimental.	44
3.8	Vista desde arriba del montaje experimental y posición del sensor GY-80	45
3.9	Fotografía del circuito empleado	46
3.10	Diagrama del circuito empleado	46
3.11	Toma de datos para un paciente	47
3.12	Ejemplo de un conjunto de mediciones de un magnetómetro sin interferencias.	50
3.13	Ejemplo gráfico de la data sin depurar.	53
3.14	Ejemplo gráfico de la data sin depurar matriz fila.	54
3.15	Ejemplo de la data normalizada para una tarea y un paciente.	55
3.16	Ejemplo de la data normalizada para una tarea y un paciente con <i>outliers</i>	55
3.17	Arreglo multidimensional de la matriz G.	56
3.18	Matriz G	56
3.19	Depuración de la data por parámetro.	58
3.20	Superposición de la data no depurada y depurada.	59
3.21	Ejemplo de los parámetros para una tarea.	59
3.22	Matriz Dr, conformada por varios ejemplos de la data.	60
3.23	Muestra del paso de la matriz Dr a la matriz A.	60
3.24	Vista de la matriz A: Todos los ejemplos de todas las tareas y pacientes.	61
3.25	Ejemplo de la matriz de covarianza.	62
3.26	Matriz de autovalores.	62
3.27	Matriz de autovectores.	63
3.28	Gráfico de sedimentación para 100 autovalores.	63
3.29	Proyección de la imagen media de cada paciente sobre el espacio V.	64

3.30	Correlación entre los pacientes y su respectiva clasificación	66
3.31	Matriz de confusión con PCA para 10 sujetos.	67
3.32	Base de conocimiento con sus etiquetas (Pcf).	68
3.33	Entrenamiento de la red neuronal.	70
3.34	Matriz de confusión con Red Neuronal para 10 sujetos.	71
3.35	Propuesta de una posible clasificación haciendo uso de la proyección	72
3.36	Diagrama de dispersión y su dendrograma.	73
3.37	Asignación de observaciones a conglomerados.	74
3.38	Dendrograma de clasificación con distancia de Mahalanobis.	74
3.39	Diagrama resumen desde la entrada de los datos hasta la clasificación	75
4.1	Ejemplo antes y después de calibrar el acelerómetro.	77
4.2	Datos concatenados para cada sensor.	78
4.3	Ejemplo de la data sin depurar.	79
4.4	Ejemplo de la data normalizada para una tarea y un paciente.	79
4.5	Frecuencia central de todas las tareas para un sujeto sano	80
4.6	Frecuencia central de todas las tareas para un sujeto sano	81
4.7	Frecuencia central de todas las tareas para un paciente con Parkinson	81
4.8	Frecuencia central de todas las tareas para un paciente con Parkinson	82
4.9	Frecuencia central para la tarea 1	83
4.10	Frecuencia central para la tarea 2	84
4.11	Frecuencia central para la tarea 3	85
4.12	Frecuencia central para la tarea 4	86
4.13	Frecuencia central para la tarea 5	87
4.14	Frecuencia central para la tarea 6	88
4.15	Frecuencia central para la tarea 7	89
4.16	Frecuencia central para la tarea 8	90
4.17	Gráfico de barras del valor medio de la aceleración por grupo	91
4.18	Gráfico de barras del valor medio del campo magnético por grupo	92
4.19	Gráfico de barras del valor medio de la velocidad angular por grupo	92
4.20	Gráfico de barras del valor medio del pulso por grupo	93
4.21	Ejemplo de sincronización para el sujeto sano 1	95
4.22	Ejemplo de sincronización para el sujeto sano 2	95
4.23	Ejemplo de sincronización para el sujeto sano 3	96
4.24	Ejemplo de sincronización para el sujeto sano 4	96
4.25	Ejemplo de sincronización para el paciente con Parkinson 1	97
4.26	Ejemplo de sincronización para el paciente con Parkinson 2	97
4.27	Ejemplo de sincronización para el paciente con Parkinson 3	98
4.28	Ejemplo de sincronización para el paciente con Parkinson 4	98
4.29	Descomposición de la aceleración en componentes de frecuencias	99
4.30	Descomposición del campo magnético en componentes de frecuencias	100
4.31	Descomposición de la velocidad angular en componentes de frecuencias	101
4.32	Descomposición del pulso en componentes de frecuencias	102
4.33	Gráfico de sedimentación para 150 autovalores.	103
4.34	Gráfico de sedimentación para 15 autovalores	104
4.35	Proyección de la imagen media de cada paciente sobre el espacio V.	105
4.36	Correlación entre los sujetos pruebas con su respectiva clasificación.	106

4.37	Ejemplo de correlación para un sujeto con respecto al grupo.	106
4.38	Matriz de confusión de PCA para los 45 sujetos de entrenamiento.	107
4.39	Entrenamiento de la red neuronal con 500 epoches	108
4.40	Matriz de confusión de RNA para los 45 sujetos de entrenamiento	109
4.41	Dendrograma de clasificación para la tarea 1	110
4.42	Distancia de Mahalanobis en función de la edad para la tarea 1	110
4.43	Dendrograma de clasificación para la tarea 2	111
4.44	Distancia de Mahalanobis en función de la edad para la tarea 2	111
4.45	Dendrograma de clasificación para la tarea 3	112
4.46	Distancia de Mahalanobis en función de la edad para la tarea 3	112
4.47	Dendrograma de clasificación para la tarea 4	113
4.48	Distancia de Mahalanobis en función de la edad para la tarea 4	113
4.49	Dendrograma de clasificación para la tarea 5	114
4.50	Distancia de Mahalanobis en función de la edad para la tarea 5	114
4.51	Dendrograma de clasificación para la tarea 6	115
4.52	Distancia de Mahalanobis en función de la edad para la tarea 6	115
4.53	Dendrograma de clasificación para la tarea 7	116
4.54	Distancia de Mahalanobis en función de la edad para la tarea 7	116
4.55	Dendrograma de clasificación para la tarea 8	117
4.56	Distancia de Mahalanobis en función de la edad para la tarea 8	117
4.57	Dendrograma de clasificación empleando todas las tareas.	118
4.58	Distancia de Mahalanobis en función de la edad para todas las tareas	118
4.59	Dendrograma de clasificación para todas las tareas solo el magnetómetro	123
4.60	Distancia de Mahalanobis en función de la edad solo magnetómetro	124
4.61	Dendrograma de clasificación para un sujeto femenino de 25 años.	125
4.62	Distancia de Mahalanobis en función de la edad para todas las tareas F-25	125
4.63	Dendrograma de clasificación para un sujeto masculino de 25 años.	126
4.64	Distancia de Mahalanobis en función de la edad para todas las tareas M-25	126
4.65	Dendrograma de clasificación para un sujeto femenino de 29 años.	127
4.66	Distancia de Mahalanobis en función de la edad para todas las tareas F-29	127
4.67	Dendrograma de clasificación para un sujeto masculino de 29 años.	128
4.68	Distancia de Mahalanobis en función de la edad para todas las tareas M-29	128
4.69	Dendrograma de clasificación para un sujeto masculino de 30 años.	129
4.70	Distancia de Mahalanobis en función de la edad para todas las tareas M-30	129
4.71	Dendrograma de clasificación para un sujeto masculino de 31 años.	130
4.72	Distancia de Mahalanobis en función de la edad para todas las tareas M-31	130
4.73	Dendrograma de clasificación para un sujeto femenino de 34 años.	131
4.74	Distancia de Mahalanobis en función de la edad para todas las tareas F-34	131
4.75	Dendrograma de clasificación para un sujeto masculino de 39 años.	132
4.76	Distancia de Mahalanobis en función de la edad para todas las tareas M-39	132
4.77	Dendrograma de clasificación para un sujeto femenino de 42 años.	133
4.78	Distancia de Mahalanobis en función de la edad para todas las tareas F-42	133
4.79	Dendrograma de clasificación para un sujeto femenino de 48 años.	134
4.80	Distancia de Mahalanobis en función de la edad para todas las tareas F-48	134
4.81	Dendrograma de clasificación para un sujeto masculino de 48 años.	135
4.82	Distancia de Mahalanobis en función de la edad para todas las tareas M-48	135
4.83	Dendrograma de clasificación para un sujeto masculino de 51 años.	136

4.84	Distancia de Mahalanobis en función de la edad para todas las tareas M-51	136
4.85	Dendrograma de clasificación para un sujeto masculino de 53 años.	137
4.86	Distancia de Mahalanobis en función de la edad para todas las tareas M-53	137
4.87	Dendrograma de clasificación para un sujeto femenino de 62 años.	138
4.88	Distancia de Mahalanobis en función de la edad para todas las tareas F-62	138
4.89	Dendrograma de clasificación para un sujeto femenino de 67 años.	139
4.90	Distancia de Mahalanobis en función de la edad para todas las tareas F-67	139

www.bdigital.ula.ve

Índice de tablas

2.1	Escala de Hoehn y Yahr.	16
3.1	Especificaciones técnicas de la placa Arduino Due.	41
3.2	Características principales de los pacientes con Parkinson.	49
3.3	Variables registradas con sus unidades.	51
3.4	Índices de las combinaciones posibles de los parámetros.	53
3.5	Módulos y estandarización de cada una de la variables.	57
3.6	Directrices sobre la fuerza de relación.	66
4.1	Distribución de los sujetos sanos según la edad y sexo	76
4.2	Distribución de los sujetos sanos para el entrenamiento de la red.	76
4.3	Valores frecuenciales para la tarea 1 por grupo etario, patológico y grupo control.	83
4.4	Valores frecuenciales para la tarea 2 por grupo etario, patológico y grupo control	84
4.5	Valores frecuenciales para la tarea 3 por grupo etario, patológico y grupo control	85
4.6	Valores frecuenciales para la tarea 4 por grupo etario, patológico y grupo control	86
4.7	Valores frecuenciales para la tarea 5 por grupo etario, patológico y grupo control	87
4.8	Valores frecuenciales para la tarea 6 por grupo etario, patológico y grupo control	88
4.9	Valores frecuenciales para la tarea 7 por grupo etario, patológico y grupo control	89
4.10	Valores frecuenciales para la tarea 8 por grupo etario, patológico y grupo control	90
4.11	Valores medios del módulo de la aceleración por tarea	91
4.12	Valores medios del módulo del campo magnético	92
4.13	Valores medios del módulo de la velocidad angular	93
4.14	Valores medios del módulo del pulso por tarea	93
4.15	Eficiencia para cada una de las tareas y algunas combinaciones entre ellas	122
5.1	Resumen de los valores frecuenciales promedios de los grupos: sanos, patológicos y prueba, por tarea.	141
5.2	Resumen de los valores promedios de la aceleración, campo magnético, velocidad angular y pulso cardíaco de los grupos sanos y patológicos, por cada tarea.	141

Capítulo 1

Planteamiento del problema.

1.1. Planteamiento del problema.

La enfermedad de Parkinson es un trastorno crónico neurodegenerativo que se caracteriza por el temblor, la rigidez y la lentitud en los movimientos corporales, que afecta aproximadamente al 1 % de la población mundial mayor de 60 años. El Parkinson es la segunda enfermedad de tipo neurodegenerativa de causa desconocida y curso crónico progresivo y no reversible después de la enfermedad de Alzheimer [1, 7]. La prevalencia de la EP varía ampliamente a lo largo del mundo. En general, en Europa y Estados Unidos es más alta que en el resto de países, mientras que en Latinoamérica, Asia y África es inferior, especialmente en este último continente. En Latinoamérica se estima una prevalencia de 300 a 600 casos por cada 100000 habitantes [7]. En Venezuela, aun cuando no existen hasta la actualidad estadísticas oficiales se calcula por extrapolación de las cifras a nivel mundial, que hay entre 30000 y 45000 personas con la enfermedad.

El desarrollo de la EP está relacionado con la edad ya que generalmente aparece a partir de los 50 años y la media de inicio de los síntomas se sitúa en los 60 años. Sin embargo, se registra recientemente un aumento considerable de casos en personas más jóvenes, denominándose Parkinson precoz y se estima que cerca del 15 % de los pacientes tienen menos de 50 años, los médicos atribuyen estos casos a factores como la contaminación y al estrés [8]. Por lo tanto, es esencial un diagnóstico temprano de la EP, ya que cada tipo de parkinsonismo tiene un tratamiento diferente para mejorar los síntomas y hacer más lento el progreso de la enfermedad. Este diagnóstico se basa en la historia clínica del paciente, un examen neurológico y en la experiencia del médico, por lo que es fundamentalmente clínico, representando una tarea compleja ya que los signos y síntomas pueden ser confundidos con los efectos del envejecimiento normal, lo que conlleva a un diagnóstico tardío debido a que no existe una prueba sencilla y universalmente aplicable [9].

Se estima que alrededor del 25 % de los casos son diagnosticados incorrectamente, por lo que se convierte en una necesidad encontrar una herramienta que pueda brindar un diagnóstico más certero [5]. Recientemente existe en el área de la medicina el interés por encontrar y aplicar modelos de clasificación y predicción concretos que ayuden a obtener diagnósticos y tratamientos acertados para la prevención y cura de enfermedades. Una de las múltiples alternativas que prometen las tecnologías para el análisis de la información son los algoritmos de Inteligencia Artificial (IA), que ofrecen un potencial de posibilidades en el diagnóstico de distintas enfermedades y su tratamiento.

Las redes neuronales artificiales (RNA), árboles de decisión (AD) y redes bayesianas (RB) son los algoritmos más usados en el sector de la salud [6]. Se han realizado estudios previos sobre el diagnóstico de EP basándose en síntomas no motores como la pérdida del habla, otros utilizan características de movimiento y marcha, con tareas de agarre y elevación, pruebas de golpeteo de dedos, movimiento de manos y dedos y escritura a mano; ya que el Parkinson disminuye la amplitud de los movimientos, a la vez que provoca rigidez en los músculos. Hasta la fecha, todos estos estudios han mostrado limitaciones en una o más facetas (en la sensibilidad y especificidad de los resultados, el nivel de supervisión o intervención de especialistas necesarios o el requisito de equipo especializado) que han impedido su aplicación en general como herramientas para diagnosticar la EP.

Debido a la complejidad de la patología a estudiar en la presente investigación, se propone hacer uso de las RNA, especialmente de un perceptrón multicapa de propagación hacia atrás, para la búsqueda y clasificación de patrones que permita la detección temprana de la enfermedad de Parkinson, a partir de señales de movimientos de los brazos del paciente obtenidas con un sistema electromecánico (MEMS) el cual está constituido por un acelerómetro, un giróscopo, un magnetómetro y un pulsómetro. Además, algunos estudios han demostrado que los algoritmos de IA se ven influenciados negativamente por información irrelevante y redundante, por lo que es necesario reducir la dimensionalidad de los datos normalizándolos y aplicándoles un análisis de componentes principales (PCA).

1.2. Objetivos.

1.2.1. Objetivo general.

- Emplear el análisis de componentes principales (PCA) y redes neuronales artificiales (RNA) para la búsqueda y separación de clases que permitan la detección temprana de la enfermedad de Parkinson a partir de la dinámica corporal del paciente.

1.2.2. Objetivos específicos.

- Construir una base de datos que serán obtenidos mediante un sistema microelectromecánico (MEMS) a través de tareas previamente seleccionadas para extraer características dinámicas de sujetos sanos y de pacientes con la enfermedad de Parkinson.
- Estudiar la dinámica corporal de los sujetos sanos y de los pacientes con la enfermedad de Parkinson a partir de los datos obtenidos para justificar el empleo de redes neuronales para la clasificación.
- Utilizar el análisis de componentes principales para la reducción de la dimensionalidad de los datos y la obtención de una primera clasificación basada en la correlación de los mismos.
- Desarrollar una red neuronal del tipo perceptrón multicapa de propagación hacia atrás para la búsqueda y separación de clases que permita una detección temprana de la enfermedad de Parkinson.

1.3. Justificación.

La enfermedad de Parkinson es un trastorno neurodegenerativo con manifestaciones motoras y no motoras. El proceso inicia mucho tiempo antes de la aparición de los síntomas no motores. Aun cuando se ha avanzado en el conocimiento de los síntomas no motores y en la evolución de la enfermedad, el diagnóstico se basa en las características clínicas, sin embargo los estudios clínico-patológicos han demostrado que hasta un 25 % de los pacientes diagnosticados con EP durante la vida tienen un diagnóstico diferente en la autopsia.

Los errores diagnósticos más frecuentes son: temblor esencial, parkinsonismo vascular y otros cuadros atípicos. Lo anterior también determina que diferentes metodologías empleadas para el diagnóstico generan distintos datos sobre la epidemiología de la enfermedad, lo cual se traduce en variabilidad en las cifras reportadas sobre la mayoría de los ámbitos de la EP. Además, un paciente puede tener la enfermedad hasta 10 años antes de ser diagnosticada [7] y, en el momento del diagnóstico, típicamente el 70 % de las neuronas en la parte afectada del cerebro (la sustancia negra) ya han sido perdidas.

Se estima que la enfermedad de Parkinson afecta actualmente aproximadamente a 7 millones de personas mayores de 50 años, calculándose que para el año 2030 sea duplicada la cifra, convirtiéndose en un problema de salud pública [1]. Las proyecciones de la Organización Mundial de la Salud (OMS) también estiman un incremento de la prevalencia de esta enfermedad por el proceso natural de envejecimiento de la población. Actualmente no hay cura o medicación que reduzca o detenga la progresión de la enfermedad de Parkinson, aunque es posible reducir los síntomas en la etapa temprana, permitiéndole al paciente mejorar su calidad de vida durante más tiempo.

Los tratamientos para la EP se han dividido en dos grupos: farmacológico y quirúrgico. Los fármacos terapéuticos para tratar la EP se dividen en fármacos terapéuticos sintomáticos y terapéuticos dirigidos; incluyen principalmente la Levodopa (dopamina). Sin embargo, el éxito terapéutico alcanzado al inicio va disminuyendo a los pocos años de tratamiento debido al desarrollo de complicaciones motoras y psiquiátricas, que son consecuencia del avance de la enfermedad. Mientras que el tratamiento quirúrgico como la estimulación cerebral profunda del globo pálido interno o del núcleo subtalámico, presentan el inconveniente de que tan solo alrededor del 10 % de los pacientes son candidatos para estas cirugías a la vez que son invasivas [1, 8].

Con respecto a los diagnósticos de enfermedades en general, existen varios biomarcadores físicos y funcionales que se pueden utilizar para proporcionar información tanto diagnóstica como predictiva. La interacción hombre-computadora (HCI) investiga las interfaces entre las personas y las computadoras, produciendo marcadores que pueden usarse para medir el estado del usuario, por ejemplo, estados fisiológicos, cognitivos y mentales. En principio, cualquier dispositivo con el que interactúen los usuarios y que produzca una salida que pueda medirse y almacenarse, podría utilizarse. Sin embargo, las evaluaciones basadas en la tecnología también deben proporcionar resultados válidos y precisos, ser independientes del entrenamiento del evaluador y permitir un uso fácil y repetitivo.

El desarrollo tecnológico en los últimos años ha traído avances importantes particularmente en el campo de la salud. Las redes neuronales artificiales, los sensores y las técnicas de análisis de datos

contribuyen en mejorar los servicios en cualquier campo de estudio especialmente en la medicina, proporcionando herramientas para monitorear enfermedades, identificar patrones que apoyan los diagnósticos médicos y detectar condiciones anómalas, fácilmente aplicables éstas en áreas rurales o países en vías de desarrollo, donde además no hay especialistas o los servicios médicos son muy costosos [10].

La inteligencia artificial puede ayudar a los médicos a realizar un diagnóstico en poco tiempo y con una alta precisión, por lo tanto el enfoque de esta investigación será el de emplear las redes neuronales artificiales, en especial un perceptrón multicapa de propagación hacia atrás, para realizar una detección temprana de la enfermedad de Parkinson en sus primeros estadios, a través de la búsqueda y separación de clases de los datos de la dinámica corporal, en especial del movimiento de los brazos siguiendo patrones, sirviendo como un método que brinde apoyo al médico en la toma de decisiones concernientes al diagnóstico de esta enfermedad.

1.4. Estado del arte.

La evaluación médica constituye una herramienta primordial para obtener información sobre el estado de salud de los pacientes. En las últimas décadas se han desarrollado una serie de sensores tanto fisiológicos como biocinéticos que pueden extraer diversos datos de importancia médica. Los registros médicos contienen información valiosa que puede ser utilizados para entrenar algoritmos, pero debido a la gran cantidad de datos, extraer conocimientos de ellos a través de los métodos de análisis tradicionales es una tarea compleja, por este motivo se han incorporado redes neuronales, redes bayesianas, árboles de decisión, entre otros algoritmos que contribuyen en la toma de decisiones con el propósito mejorar la calidad de esas evaluaciones. Estos sistemas expertos de ayuda tienen tres componentes fundamentales: base de datos (información del paciente), algoritmos de inteligencia artificial y usuarios (médicos), logrando importantes contribuciones en el diagnóstico y seguimiento de enfermedades.

Con respecto al diagnóstico de la enfermedad de Parkinson empleando redes neuronales y sensores se encuentran los siguientes estudios:

Belén A. (2019) [11], propone el diagnóstico de la enfermedad de Parkinson mediante grabaciones telefónicas. En total cuenta con 24160 grabaciones obtenidas del proyecto mPower de 1670 participantes: 425 diagnosticados con EP (8267 grabaciones) y 1245 de control (15893 grabaciones). Se crearon dos bases de datos: la primera incluye 62 características de sonido como frecuencias f_0 , f_1 , f_2 , f_3 y f_4 , volumen, datos normativos de la voz e índices de perturbación, extraídas a través del software OpenSmile y la segunda base de datos contiene las grabaciones de voz en crudo. Para la red neuronal emplearon la librería Keras con 7 capas densas y un *dropout* de 0.3, registra los siguientes resultados: Base de datos 1: Con una selección aleatoria del conjunto test una precisión del 71.1 % mientras que para una selección consecutiva del conjunto test una precisión de 40.5 %. Para la base de datos 2 (análisis en crudo) con una selección aleatoria del conjunto test una precisión del 62.3 % mientras que para una selección consecutiva del conjunto test una precisión de 57.3 %. Concluye que se obtienen mejores resultados con una aleatoriedad de los datos de test y con el método de valorización cruzada.

Sivaranjini S. y Sujatha C. (2019) [12], intentaron clasificar las imágenes de resonancia magnéti-

ca (IRM) del control saludable y de los sujetos con enfermedad de Parkinson utilizando la red neuronal de aprendizaje profundo. La arquitectura de la red neuronal convolucional *AlexNet* se utilizó para refinar el diagnóstico de la enfermedad de Parkinson. Las imágenes de RM están entrenadas por la red de transferencia aprendida y probadas para dar las medidas de precisión. Obtuvieron una precisión del 88.9 % con el sistema propuesto. Concluyendo que los modelos de aprendizaje profundo pueden ayudar a los clínicos en el diagnóstico de la enfermedad de Parkinson y producir una clasificación objetiva de un grupo de pacientes.

Caliskan et al. (2017) [13], proponen el diagnóstico de la enfermedad de Parkinson a través de la interpretación de los datos de registro de voz. Utilizando las deficiencias del habla, que es uno de los primeros indicadores de la enfermedad de Parkinson. El diagnóstico constó de tres pasos principales que incluyeron pre-procesamiento, extracción de características y clasificación, utilizó una red neuronal que combina el *Autoencoder (AE)* y *Softmax Classifier* y además empleó dos bases de datos, la primera extraída del repositorio de la Universidad de California, que consta de los registros de voz de 31 pacientes, 23 con la EP y registra 23 atributos y la segunda creada en el Departamento de Neurología en la Facultad de Medicina, Universidad de Estambul, que consta de 40 pacientes, 20 con la EP y registra 26 atributos. Se compararon los resultados del clasificador propuesto con los resultados del método de clasificación de vanguardia como redes bayesianas (RB) y árboles de decisión (AD), obteniendo que el clasificador de la red neuronal profunda es un clasificador muy eficiente, con un 93.7 % para el diagnóstico de la enfermedad de Parkinson.

Rustempasic I. y Can M. (2013) [14], tuvieron como objetivo principal diferenciar entre personas sanas y personas con la enfermedad de Parkinson mediante el empleo de redes neuronales artificiales y utilizaron el conjunto de datos tomados de la base de registros de la voz de la Universidad de California, con 48 casos normales y 147 casos de EP. Para entrenar los ejemplos que se utilizaron se reforzaron mediante la técnica de filtrado con siete Committee Machine y el Análisis de Componentes Principales (PCA) para la reducción de datos. Los resultados experimentales demostraron que la combinación de estos métodos propuestos obtuvieron muy buenos resultados con un valor positivo del 92 % en la clasificación de la enfermedad de Parkinson.

Soleimanian F. y Mohammadi P. (2013) [15], utilizaron dos tipos de redes neuronales para realizar el diagnóstico efectivo de la enfermedad de Parkinson: un perceptrón multicapa con aprendizaje de propagación hacia atrás (MLP) y redes neuronales con funciones de base radial (RBF); para diferenciar las variables clínicas de las muestras ($n=195$) de pacientes que sufrían de la enfermedad de Parkinson y de quienes no. Para este propósito, el conjunto de datos fue tomado de la base de datos del repositorio de aprendizaje automático de la Universidad de California. Se utilizó la media de los datos para la normalización de la data para el entrenamiento y durante los cálculos. Se determinó que el MLP tiene la mejor clasificación con un 93,22 % de precisión para el conjunto de datos, mientras que para el RBF se obtuvo una precisión del 86,44 %. Concluyendo que el MLP es más efectivo y esta técnica puede ayudar a los neurólogos a hacer sus decisiones definitivas sin vacilación.

Geman et al. (2013) [16], mostraron en su trabajo como las herramientas modernas que incluyen sensores como el Nintendo Wii y el Xbox pueden adquirir señales biomédicas, en especial de parámetros dinámicos de la marcha parkinsoniana y del temblor, que pueden ser analizadas posteriormente para el diagnóstico médico. El objetivo fue encontrar un instrumento para la detección de la EP, usando señales de la marcha y el temblor. La base de datos del estudio constó de 32

pacientes con EP, 58 sujetos sanos y 14 sujetos sospechosos. Los datos del temblor adquiridos en este estudio se registraron utilizando acelerómetros, sensores de presión y un microcontrolador que ejecuta la adquisición de datos. Realizaron un análisis lineal y no lineal de la data, donde emplearon análisis de Fourier y paquetes de análisis no lineal como *TISEAN* y *NLzyer*, además hacen uso del algoritmo *K-means*. Concluyeron que los parámetros no lineales pueden proporcionar información esencial en el diagnóstico diferencial en comparación con los parámetros lineales clásicos, además establecen que la enfermedad de Parkinson puede detectarse con exactitud de hasta el 98 % utilizando el algoritmo *K-means*.

Khemphila A. y Boonjing V. (2012) [17], utilizaron un perceptrón multicapa (MLP) para hacer un diagnóstico de la enfermedad de Parkinson. El objetivo fue clasificar la presencia de la enfermedad en una base de datos que consistió en los registros de voz de 31 personas, de las cuales 23 tienen la enfermedad. Esta base de datos fue creada por Max Little de la Universidad de Oxford. Utilizando atributos obtenidos inicialmente, entre los cuales se encuentran: frecuencia vocal fundamental (mínima, media y máxima), exponente de escala fractal de la señal, medidas de la relación de ruido de componentes tonales en la voz; se obtuvo para la fase de entrenamiento un 91,5 % y para la fase de validación 80,7 %. Mientras que al realizar una reducción de las características, el número de atributos final es de 16 y se obtuvo para la fase de entrenamiento un 82,0 % y para la fase de validación 83,3 %, concluyendo que la reducción de características ayuda a incrementar la eficiencia en la fase de validación.

Para el diagnóstico de otras enfermedades empleando redes neuronales se encuentran los siguientes estudios:

Boston C. (2018) [18], afirma en su artículo que un equipo de científicos de Estados Unidos y China han desarrollado una herramienta que utilizó *Big Data y Machine Learning* para diagnosticar la neumonía y han demostrado que es capaz de distinguir entre las infecciones pulmonares provocadas por bacterias que requieren tratamiento con antibiótico de inmediato, y las causadas por virus. Además, esta herramienta puede diagnosticar la degeneración macular y edema macular diabético, enfermedades de la retina que constituyen dos de las causas más frecuentes de ceguera irreversible. Aunque ambas se pueden tratar y mejorar si el diagnóstico se realiza temprano. En el estudio, se compararon los resultados de la herramienta con el diagnóstico realizado por cinco oftalmólogos expertos en retina que revisaron los mismos escáneres. Observaron que la inteligencia artificial diagnosticaba igual de bien que el experto y era capaz de tomar una decisión en 3 segundos con una eficiencia de 95 %.

Ortiz et al. (2016) [19], para el diagnóstico de la enfermedad de Alzheimer obtuvieron imágenes de tomografía por emisión de positrones (PET) del cerebro afectado. Esta prueba consiguió generar una imagen del cerebro a nivel funcional. Por otro lado necesitaron la imagen generada por resonancia magnética (IRM), lo que ofreció una vista del cerebro a nivel estructural. Estas dos imágenes se introdujeron en el sistema que las divide en diferentes regiones y se estudia cada una de ellas en una red neuronal de aprendizaje profundo especializada en cada área, para posteriormente combinar los resultados y emitir un diagnóstico. Aseguraron que mediante un sistema de IA basado en aprendizaje profundo consiguieron diagnosticar con un acierto superior al 90 % a enfermos de Alzheimer y con un porcentaje superior al 80 % consiguieron una detección precoz.

Bostan V. y Pantelimon B. (2015) [20], utilizaron un perceptrón multicapa (MLP) con el ob-

jetivo principal de diseñar un método no invasivo para diagnosticar pacientes con cirrosis hepática utilizando únicamente datos de laboratorio. Como resultado del estudio tras comparar las RNA en términos de rendimiento, el modelo fue capaz de reconocer 3 tipos de valores: cirrosis, no cirrosis y los no identificables con una tasa de éxito de casi el 90 %.

En cardiología, se han aplicado RNA para determinar el diagnóstico de la enfermedad coronaria. **Atkov et al.** (2012) [21], realizaron un estudio en el que se utilizó un MLP que incluyó tanto factores genéticos como no genéticos relacionados con la enfermedad, el resultado tuvo una precisión del 93 %, a pesar de estos buenos resultados, hay que señalar que para algunos modelos la precisión fue inferior al 90 %. Las RNA también se han aplicado en otras enfermedades del corazón, tales como los defectos de las válvulas cardíacas. En el estudio realizado por **Uğuz H.** (2012) [22], se desarrolló un sistema para la clasificación de las señales cardíacas, obtenidas a partir 120 pacientes con estenosis pulmonar, mitral y sin estenosis, usando el estetoscopio. La correcta clasificación se logró para el 95 % de los diferentes sonidos cardíacos.

Abdul et al. (2012) [23], proponen el uso de un MLP con el propósito de ayudar al dermatólogo en el diagnóstico de seis tipos de enfermedades. Estas enfermedades cutáneas son psoriasis, dermatitis seborreica, liquen plano, pitiriasis rosea, dermatitis crónica y pitiriasis rubra pilaris. Realizar un diagnóstico acertado es importante ya que estas enfermedades comparten rasgos clínicos similares de eritema y descamación sin mucha diferencia. Además, una enfermedad puede mostrar la característica de otra enfermedad en las primeras etapas. Los resultados mostraron que la red y el uso de selección de características alcanzó el 91 % de eficiencia.

Aeinfar et al. (2009) [24], proponen entrenar un perceptrón multicapa con propagación hacia atrás para reconocer el patrón de diagnóstico y predicción de cinco desordenes de la sangre: la anemia megaloblástica, talasemia, púrpura trombocitopénica idiopática (PTI), leucemia mielógena crónica y linfoproliferativa, a través de los resultados de exámenes de sangre. Los exámenes de sangre utilizados correspondieron a 450 casos de pacientes del hospital Taleghani en Kermanshah, Irán, se utilizan en el entrenamiento supervisado y en la actualización de los parámetros de la red. Este método obtuvo un porcentaje de error de 0,78 % por lo que concluye que la red neuronal artificial es aplicable debido a su alta precisión, rápida convergencia y bajo uso de memoria para el diagnóstico y predicción del desorden.

Finalmente, se puede indicar que los algoritmos de inteligencia artificial, sus diferentes métodos y otras herramientas han sido ampliamente usadas en el campo de la medicina con gran éxito. Sin embargo, en esta revisión no se han encontrado trabajos relacionados directamente con el tema de investigación tratado en esta tesis, basado en emplear un perceptrón multicapa de propagación hacia atrás sobre una base de datos de la dinámica corporal de pacientes aplicando previamente un análisis de componentes principales en la disminución de la dimensionalidad de los datos para el diagnóstico temprano de la enfermedad de Parkinson.

1.5. Distribución por capítulos.

Esta tesis está estructurada en cinco capítulos:

- **Capítulo 1: Planteamiento del problema.** Se realiza el planteamiento del problema, la presentación de los objetivos, la justificación de la investigación y el estado del arte relacionado con el tema a estudiar.
- **Capítulo 2: Marco Teórico.** Se presenta una revisión teórica de la enfermedad de Parkinson, las causas, manifestaciones clínicas, su diagnóstico, pronóstico, tratamiento y prevención. Aspectos relacionados con las redes neuronales artificiales (RNA): arquitectura y entrenamiento; perceptrón multicapa: su estructura, la retropropagación, deducción del algoritmo *backpropagation* y aplicaciones del perceptrón. Finalmente, se describe el tema de análisis de componentes principales (PCA): autovalores y autovectores, cálculo de las componentes principales, proceso de extracción de factores y gráfico de sedimentación.
- **Capítulo 3: Marco Metodológico.** Se hace una presentación de un diagrama de flujo que contiene las 5 etapas para la adquisición y análisis de los datos. En la primera se describen las bases fundamentales del hardware y software, se presenta el montaje experimental, las mediciones básicas y la población de pacientes con EP en estudio. La segunda etapa corresponde al tratamiento de los datos. La tercera está conformada por la aplicación de PCA y la creación de la base de conocimiento. En la cuarta etapa se realiza el entrenamiento de la red neuronal y en la última la validación del método y la presentación de los dendrogramas de clasificación.
- **Capítulo 4: Resultados y análisis de los datos.** Se centra en el análisis de los resultados obtenidos.
- **Capítulo 5: Discusiones y conclusiones.** Se comentan las discusiones de los resultados, se hace una comparativa con los trabajos encontrados en la literatura, se presentan las conclusiones y recomendaciones para futuros trabajos.

Finalmente, se encuentra el apartado de apéndice, conformado por:

- Programas de Matlab.
- Programas Arduino.
- Abreviaturas.
- Glosario.
- Fichas técnicas de los dispositivos (*Datasheets*).

Capítulo 2

Marco Teórico.

2.1. Enfermedad de Parkinson.

La Enfermedad de Parkinson (EP) es un trastorno degenerativo del sistema nervioso central que pertenece a un grupo de afecciones conocidas como trastornos del movimiento. La constituye una entidad clínica neurodegenerativa caracterizada por la presencia de temblor en reposo, bradicinesia y rigidez muscular. Se presenta como una patología crónica, de larga duración, irreversible y con una sintomatología que empeora con el tiempo [1, 3, 4]. A pesar de los muchos estudios y esfuerzos en investigación, al día de hoy se desconocen las causas exactas que desencadenan esta enfermedad, se postula que la respuesta podría estar en la combinación de una predisposición genética junto a factores ambientales. La EP es el trastorno del movimiento más frecuente y la segunda enfermedad neurodegenerativa más común en occidente, tras la enfermedad de Alzheimer, lo cual hace que sea una de las enfermedades más estudiadas desde su descubrimiento por parte del médico inglés James Parkinson en el año 1817 [25].

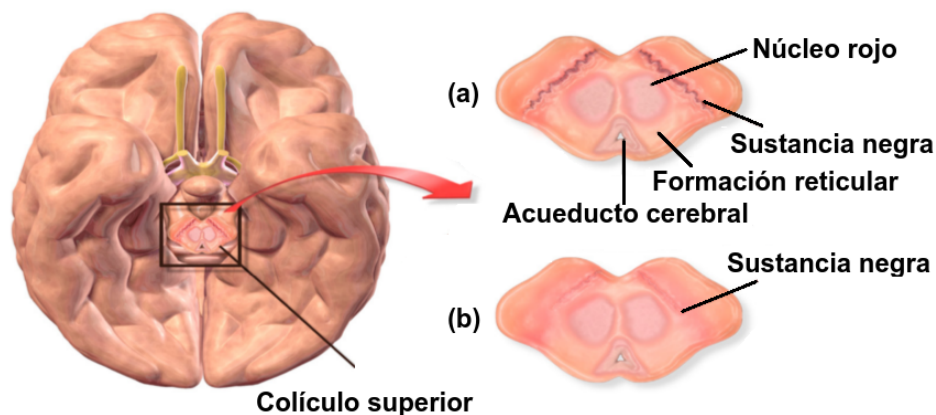


Figura 2.1: Corte transversal del Mesencéfalo que muestra la evidente pigmentación de la sustancia negra de un sujeto sano (a) y la despigmentación en un paciente con enfermedad de Parkinson (b).

Desde un punto de vista fisiopatológico, la EP se caracteriza por la pérdida progresiva de neuronas dopaminérgicas en determinadas regiones del sistema nervioso central como se muestra en la figura 2.1. También se encuentran afectadas las neuronas colinérgicas, catecolaminérgicas y serotoninérgicas. En un individuo sano, la generación de los movimientos motores coordinados por parte del

cerebro, radica en un equilibrio entre la activación de los movimientos voluntarios o vía directa y la acción de la vía inhibitoria o vía indirecta. En ambas vías participa el neurotransmisor dopamina, así como sus receptores. Por lo tanto, en condiciones normales están implicados los receptores tipo 1 de dopamina (D1), situados en el globo pálido interno y en la parte reticular de la sustancia negra, y también participan las neuronas dopaminérgicas que proyectan al globo pálido externo, donde se localizan los receptores D2, como se muestra en la figura 2.2 [26]. En los pacientes con EP, debido a la presencia de una degeneración dopaminérgica, estas vías se encuentran afectadas, produciéndose una hipoactividad en la vía directa y una hiperactividad en la vía inhibitoria, todo ello genera una incapacidad para la realización de los movimientos coordinados.

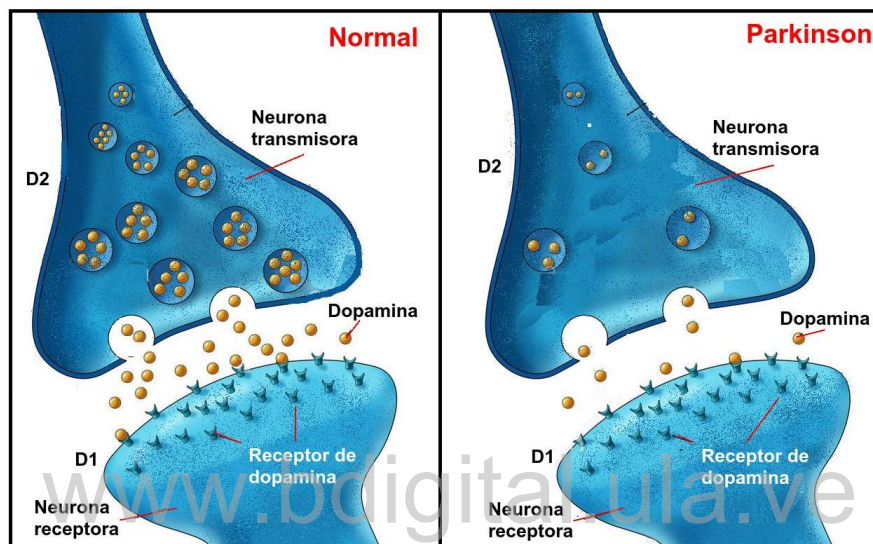


Figura 2.2: Ilustración etiquetada que muestra niveles bajos del neurotransmisor dopamina en una neurona afectada por la EP (derecha) y niveles normales (izquierda).

2.1.1. Causas de la enfermedad.

Se desconoce la causa exacta de la enfermedad, aunque algunos casos son hereditarios y pueden ser debidos a mutaciones genéticas específicas. Sin embargo, la mayoría de los casos son esporádicos, lo que quiere decir que la enfermedad generalmente no es hereditaria. Se cree que la enfermedad es el resultado de una combinación de susceptibilidad genética y exposición a uno o más factores ambientales desconocidos que desencadenan la enfermedad.

- **Factores genéticos.** Hasta el momento han sido identificados nueve loci y ocho genes involucrados en la EP familiar. El primer gen fue el de la α -sinucleína, identificado en una familia italo-americana [1]. Posteriormente, se han encontrado otras formas de EP de etiología monogénica, producidas por mutaciones en los genes LRRK2, Parkin, PINK1, DJ-1 y ATP13A2. Actualmente, se considera que en la mayoría de los casos la susceptibilidad genética para desarrollar EP es producto de variantes normales de algunos genes que, bajo la influencia de ciertos factores ambientales aun no identificados, participan en la patogénesis de la enfermedad [27].

- **Factores ambientales.** A pesar de que hasta ahora no se ha sugerido ningún factor ambiental esencialmente asociado a la EP, numerosos estudios epidemiológicos han relacionado diversos factores ambientales con un mayor o menor riesgo de padecer la EP. Así, por ejemplo, parece claro que la incidencia de la EP es superior en sociedades occidentales y más aún en ambientes industriales y agrícolas. Atendiendo a esta realidad, hay autores que citan como factor predispuesto al desarrollo de la EP el hecho de vivir en un medio rural [4, 25, 27]. La asociación entre exposición a pesticidas y mayor riesgo de EP es fuerte y no parece ser explicada sólo por azar y por sesgos. Sin embargo, estos resultados provienen fundamentalmente de estudios observacionales, por lo que se puede concluir que la relación sea causal.

2.1.2. Manifestaciones Clínicas.

La EP afecta predominantemente a personas de edad adulta, entre 50 y 65 años, actualmente es cada vez mayor el número de personas jóvenes diagnosticadas, siendo 1 de cada 5 menores de 50 años [28]. La EP se manifiesta con una variedad de signos y síntomas que suelen desorientar al médico general que normalmente no se encuentra familiarizado con el cuadro característico. Los primeros síntomas se engloban en la fase premotora de la enfermedad, entre los más citados se encuentran: sensación de astenia, flexión de un brazo con ausencia de balanceo, lenguaje monótono, trastornos del estado del ánimo y enlentecimiento general [4].

En la etapa de los síntomas de tipo motor, se encuentran los cuatro principales síntomas de la EP:

1. **Temblor:** El temblor asociado con la enfermedad tiene una apariencia característica. Es el resultado de una contracción lenta de los músculos tanto agonistas como antagonistas. Es la forma más frecuente de presentación de la enfermedad. Típicamente, el temblor toma la forma de un movimiento rítmico hacia adelante y hacia atrás a una velocidad de 4-6 Hz [1]. El temblor afecta sobre todo a las extremidades superiores y, con menor frecuencia, a labios, mentón, lengua y extremidades inferiores. Es característico que desaparezca con el movimiento para reaparecer de nuevo cuando el miembro vuelve a estar en reposo. También es típico que cese durante el sueño y aumente con la ansiedad.

Se han descrito diferentes formas de presentación del temblor en la EP:

- **Tipo I o clásico:** Temblor de reposo o reposo y postural con la misma frecuencia.
- **Tipo II:** Temblor de reposo y postural/cinético pero con diferente frecuencia.
- **Tipo III:** Sólo temblor postural/cinético. Temblor de reposo monosintomático.

La ausencia de temblor no descarta el diagnóstico de EP.

2. **Rigidez:** Es la resistencia al movimiento y afecta a la mayoría de las personas con EP. Los músculos permanecen constantemente tensos y contraídos y la persona tiene dolor o se siente tiesa. La rigidez se vuelve obvia cuando alguien más trata de mover o extender el brazo de la persona. Clínicamente se evalúa mediante el movimiento pasivo de varias articulaciones, mientras se le pide al paciente que intente relajar su extremidad. Se considera que hay rigidez cuando es necesaria más fuerza de la esperada para movilizar la articulación, en un cierto rango de movimiento.

3. **Bradicinesia:** Es la lentitud en la ejecución de cualquier movimiento, con una reducción progresiva de la velocidad y la amplitud de los movimientos repetitivos. Esta disminución del movimiento espontáneo y automático es particularmente frustrante debido a que puede hacer que las tareas sencillas se vuelvan difíciles. La persona no puede realizar rápidamente movimientos rutinarios. Las actividades que antes realizaba de forma rápida y sencilla, como hacerse el aseo personal o vestirse, pueden tomar mucho más tiempo. A menudo hay una disminución de las expresiones faciales [27]
4. **Inestabilidad postural:** Las alteraciones posturales se suelen manifestar con posturas en flexión de miembros y tronco. Están alterados los reflejos posturales y de enderezamiento. En ocasiones, esto puede provocar caídas y una marcha en la que parece que el paciente busca su centro de gravedad, con el cuerpo inclinado hacia adelante y pasos cortos.

Si bien es uno de los signos cardinales de la EP, en las etapas iniciales se manifiesta solo en forma de una ligera desestabilización ante la prueba del empujón. En esta fase tiene una buena respuesta a la terapia y se hace más evidente con los años de evolución [7].

La enfermedad de Parkinson no afecta a todas las personas de la misma manera y la tasa de progresión y los síntomas particulares difieren entre los individuos. Los síntomas de la enfermedad comienzan típicamente en un lado del cuerpo. Sin embargo, con el tiempo la enfermedad afecta ambos lados. A medida que la enfermedad progresa, los síntomas de la EP pueden comenzar a interferir con las actividades diarias [1]. Una cantidad de otros síntomas denominados “no motores”, los cuales afectan al proceso de pensamiento, la sensibilidad y la sensación de bienestar, pueden acompañar la enfermedad de Parkinson y algunos pueden tratarse con medicamentos o fisioterapia. Entre los más comunes se encuentran en las siguientes categorías: [1, 8, 25]

1. Neuro-psiquiátricos:

- Trastornos afectivos.
- Alteraciones cognitivas.
- Alucinaciones.
- Delirios.
- Demencia.
- Depresión.
- Alteración de la memoria.
- Trastornos del control de impulsos.
- Psicosis (en relación al tratamiento farmacológico).

2. Del sueño:

- Somnolencia diurna.
- Sueños vívidos.

- Insomnio.
- Sueño fragmentado.
- Síndrome de piernas inquietas.

3. Autonómicos:

- Hipotensión ortostática.
- Reducción de la frecuencia cardíaca.
- Sudoración excesiva.
- Seborrea.
- Disfunción sexual.
- Alteración de la micción.

4. Digestivos:

- Dificultad para masticar y tragar.
- Gastroenteritis.
- Estreñimiento.
- Náuseas.

5. Sensoriales:

- Dolor.
- Alteraciones visuales.
- Parestesia: sensación de hormigueo.
- Hiposmia: reducción parcial del olfato.
- Anosmia: pérdida total del olfato.
- Alteración en el sentido del gusto.

6. Otros:

- Alteraciones de la voz.
- Fátiga.
- Pérdida de peso.
- Disminución de la masa ósea.
- Alteraciones respiratorias.
- Calambres musculares.

2.1.3. Diagnóstico.

El diagnóstico de la enfermedad de Parkinson es difícil de precisar en sus inicios cuando los síntomas no motores predominan y en el estadio motor (temblor, rigidez y lentitud) cuando se presentan datos atípicos que pudieran estar causados por otro tipo de enfermedad neurodegenerativa. Actualmente no existen pruebas de sangre o de laboratorio que diagnostiquen la EP. Por ello, el diagnóstico se basa en la historia clínica y en un examen neurológico del paciente [1, 14]. En algunos casos, la EP puede ser difícil de diagnosticar con precisión al inicio de la enfermedad, ya que los signos y síntomas tempranos a veces se pueden descartar como los efectos del envejecimiento normal. Un diagnóstico rápido depende en gran parte de la experiencia del neurólogo.

La ausencia de una prueba clínica universalmente aplicable, conduce a un diagnóstico tardío de la enfermedad. Ya que muchas otras enfermedades tienen características similares pero requieren tratamientos diferentes, es fundamental hacer un diagnóstico preciso para que las personas puedan recibir el tratamiento adecuado. Por su sencillez y rapidez las escalas clínicas son las que se emplean con mayor frecuencia. El diagnóstico actual se basa en medidas subjetivas derivadas de observaciones visuales por el neurólogo para generar una puntuación de UPDRS [5].

A veces, podría tomar hasta un año para diagnosticar la EP después de una cuidadosa consideración de la historia neurológica del paciente y evaluaciones clínicas. Además, debido a la falta de medidas objetivas, también hay una alta posibilidad de realizar diagnósticos erróneos de la EP. Se ha encontrado que la tasa de diagnóstico erróneo de la EP es de alrededor del 25 % y aproximadamente el 40 % de los casos de la EP se pasan por alto como otros trastornos neurológicos. Según los expertos, el diagnóstico de EP requiere la presencia de dos o más de los cuatro síntomas motores principales de la EP [5].

Las pruebas neurodiagnósticas son poco útiles en la evaluación de un paciente con sospecha de EP, una revisión sistemática de la Academia Americana de Neurología no considera beneficioso el uso de pruebas auxiliares como: resonancia magnética, tomografía por emisión de positrones y ecografía transcraneal, dado que el diagnóstico EP continua siendo predominantemente clínico, con alta sensibilidad en cualquier estadio de la enfermedad y especificidad elevada en estadio avanzado y de un 67 % en estadio temprano [9].

Los criterios más reconocidos por la comunidad neurológica son los propuestos por el Banco de Cerebros de Londres. Éstos se basan en observaciones y correlaciones clínico-patológicos de un amplio número de casos [7].

Se distinguen tres pasos para constituir el diagnóstico de EP :

- **Primer paso:** Establecer el diagnóstico de síndrome parkinsoniano.
 - Bradicinesia

Al menos uno de los siguientes puntos:

- Rigidez muscular.
- Temblor de reposo.

- Inestabilidad postural.
- **Segundo paso:** Criterios de exclusión para Enfermedad de Parkinson. La presencia de cualquiera de ellos excluye el diagnóstico:
 - Historia de accidentes vasculares.
 - Historia de traumas de cráneo.
 - Historia de encefalitis.
 - Tratamiento con neurolépticos al inicio de los síntomas.
 - Más de un paciente con parkinsonismo en la familia.
 - Remisión sustancial espontánea
 - Signos cerebelosos.
 - Presencia de un tumor cerebral.
 - Presentación temprana y severa de demencia.
 - Signo de Babinski.
- **Tercer paso:** Criterios positivos que soportan en forma prospectiva el diagnóstico de EP. Debe existir al menos tres de ellos para establecer el diagnóstico definitivo:
 - Inicio unilateral.
 - Temblor de reposo presente.
 - Enfermedad progresiva.
 - Persistencia de asimetría afectando más el lado por el que se inició la enfermedad.
 - Excelente respuesta por la levodopa (70 -100 %).
 - Respuesta a la levodopa por cinco años o más.
 - Curso clínico de diez años o más.

2.1.4. Pronóstico.

La expectativa de vida promedio de una persona con Parkinson generalmente es la misma que para las personas que no padecen la enfermedad. Afortunadamente, hay muchas opciones de tratamiento disponibles. Sin embargo, en las etapas tardías, la enfermedad de Parkinson puede ya no responder a los medicamentos y puede llegar a asociarse con complicaciones graves como la asfixia, la neumonía y las caídas [1].

Para categorizar la situación clínica y la respuesta terapéutica en la EP se han diseñado numerosos métodos de cuantificación [29]. Entre estos métodos se pueden distinguir los que analizan síntomas, signos o incapacidad funcional que se corresponden a la mayoría de las escalas clínicas, y los que miden otros hechos clínicos con medidas físicas cuantitativas y que pueden requerir en determinados casos instrumentos más o menos sofisticados, como acelerómetros, giróscopos y programas informáticos.

La enfermedad de Parkinson es un trastorno que progresa lentamente. No es posible predecir qué curso tomará la enfermedad en cada persona. Una escala comúnmente utilizada por los neurólogos para describir cómo han avanzado los síntomas de la enfermedad en un paciente es la escala de Hoehn y Yahr presentada en la tabla 2.1 [30].

Estadio	Síntomas
0	No hay síntomas
1	Síntomas unilaterales y leves. Generalmente se presenta con temblor en una extremidad
2	Síntomas bilaterales. Mínima incapacidad. La postura y la marcha están afectadas.
3	Significativo enlentecimiento de los movimientos corporales.
4	Síntomas severos. Rigidez y bradicinesia. Incapacidad grave; aún capaz de permanecer de pie sin ayuda.
5	No puede mantenerse de pie ni caminar. Invalidez completa.

Tabla 2.1: Escala de Hoehn y Yahr.

Otra escala que se emplea comúnmente es la Escala Unificada de Valoración de la Enfermedad de Parkinson de la Sociedad de Trastornos del Movimiento (MDS-UPDRS, por sus siglas en inglés). Esta escala de cuatro partes mide el movimiento motor en la EP: experiencias no motrices de la vida diaria, experiencias motrices de la vida diaria, evaluación motriz y complicaciones motrices [3, 29, 31]. Tanto la escala de Hoehn y Yahr como la de MDS-UPDRS se usan para medir cómo están las personas y evaluar la respuesta a los tratamientos.

2.1.5. Tratamiento de la enfermedad.

Actualmente no existe una cura para la enfermedad de Parkinson, pero a veces los medicamentos, la cirugía u otras terapias pueden mejorar los síntomas motores.

Tratamientos farmacológicos: Los medicamentos para la enfermedad de Parkinson comprenden tres categorías [1]

- La primera categoría incluye medicamentos que incrementan el nivel de dopamina en el cerebro. Los medicamentos más comunes para la enfermedad son precursores de la dopamina, sustancias como levodopa que cruzan la barrera hematoencefálica y luego cambian a dopamina. Otros medicamentos imitan a la dopamina o previenen o retrasan su descomposición.
- La segunda categoría de medicamentos para la enfermedad de Parkinson afecta a otros neurotransmisores en el organismo con el fin de aliviar algunos síntomas de la enfermedad. Por ejemplo, los medicamentos anticolinérgicos interfieren con la producción o la captación del neurotransmisor acetilcolina. Estos pueden ser eficaces en la reducción de los temblores.
- La tercera categoría de medicamentos recetados para la enfermedad incluye medicamentos que ayudan a controlar los síntomas no motores de la enfermedad. Por ejemplo, se pueden recetar antidepresivos a las personas con depresión relacionada con la enfermedad. El médico

evaluará cuánto trastornan los síntomas la vida de la persona y ajustará el tratamiento al caso particular de la persona. Aun así, tal vez los síntomas no se alivien por completo.

Cirugía: Antes del descubrimiento de la levodopa, la cirugía era una opción para el tratamiento de la enfermedad de Parkinson. Los estudios en las últimas décadas han llevado a grandes progresos en las técnicas quirúrgicas y la cirugía vuelve a considerarse para las personas con Parkinson para quienes la farmacoterapia ya no es suficiente [4].

- **Palidotomía y talamotomía:** Los primeros tipos de cirugía para la EP implicaron la destrucción selectiva de partes específicas del cerebro que contribuyen a los síntomas de la enfermedad. Las técnicas quirúrgicas se han refinado y pueden ser muy eficaces para los síntomas motores de la EP. La cirugía de lesión más común es la palidotomía. En este procedimiento, el cirujano destruye selectivamente una porción del cerebro llamada globo pálido. Otro procedimiento, llamado talamotomía, implica la destrucción quirúrgica de parte del tálamo. Este enfoque se usa principalmente para reducir el temblor.
- **Estimulación cerebral profunda:** Usa un electrodo que se implanta quirúrgicamente en el cerebro, normalmente en el núcleo subtalámico o en el globo pálido. Cuando se activa usando un tubo externo, el generador de pulsos y los electrodos estimulan el cerebro de forma indolora de tal manera que ayuda a bloquear las señales que causan muchos de los síntomas motores de la enfermedad. La estimulación cerebral profunda está aprobada por la Administración de Alimentos y Medicamentos de los Estados Unidos y se usa extensamente como tratamiento para la enfermedad de Parkinson.

Terapias de apoyo y complementarias: Puede utilizarse una amplia variedad de terapias de apoyo y complementarias para la enfermedad de Parkinson. Entre éstas se encuentran las técnicas fisioterapéuticas, ocupacionales y del lenguaje normales, que pueden ayudar con problemas tales como los trastornos de la marcha y de la voz, temblores y rigidez, y deterioro cognitivo. Otras terapias de apoyo incluyen cuidados en la dieta, ejercicio, masajes terapéuticos, yoga, hipnosis y acupuntura; lo que implica mejoras en la realización de actividades diarias y logra en el paciente independencia y mejoramiento en la calidad de vida [1].

2.1.6. Prevención.

En la mayoría de los casos, no hay forma de prevenir la enfermedad de Parkinson debido a que se desconoce la causa de la enfermedad. Sin embargo, los investigadores están buscando un biomarcador, es decir, una anormalidad biológica que compartan todas las personas con Parkinson, que podría ser detectado con técnicas de evaluación o con una prueba química simple administrada a las personas que todavía no tienen síntomas parkinsonianos. Esto podría ayudar a los médicos a identificar a las personas con riesgo de tener la enfermedad. También puede permitirles encontrar tratamientos que detengan el proceso de la enfermedad en las etapas tempranas [1, 27].

2.2. Redes Neuronales Artificiales.

Las redes neuronales artificiales (RNA) son una rama de la inteligencia artificial (IA), las cuales surgen de la necesidad de obtener un sistema de procesamiento de información que imite al cerebro humano [32], el cual tiene un funcionamiento paralelo y no lineal, por lo que no puede representarse mediante modelos lineales. En efecto, las RNA aplicadas están inspiradas en las redes neuronales biológicas, aunque poseen otras funcionalidades y estructuras de conexión, constituyen una excelente herramienta en el modelado de sistemas no lineales.

2.2.1. Redes neuronales de tipo biológico.

El cerebro es el principal elemento del sistema nervioso humano y está compuesto por un tipo especial de célula llamada neurona. Una neurona es una célula viva y como tal posee todos los elementos comunes de las células biológicas. A su vez las neuronas tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas. La figura 2.3 muestra la estructura típica de una neurona biológica [33].

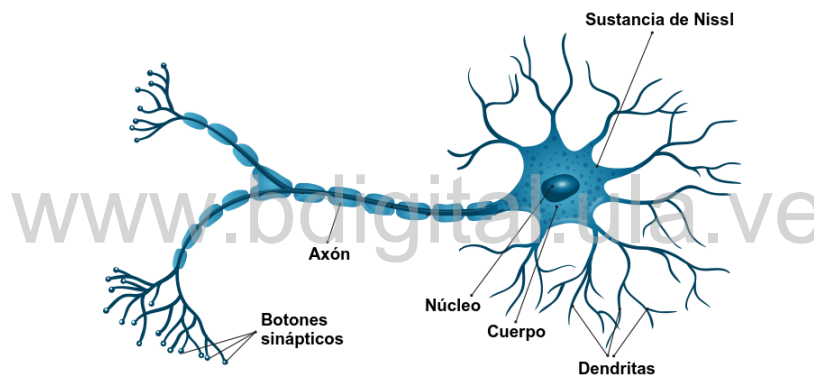


Figura 2.3: Estructura general de una neurona.

El objetivo principal de las redes neuronales de tipo biológico es desarrollar operaciones de síntesis y procesamiento de información, relacionadas con los sistemas biológicos. Las neuronas y las conexiones entre ellas constituyen la clave para el procesamiento de la información. La mayor parte de las neuronas poseen una estructura de árbol, llamada dendritas, que reciben las señales de entrada procedentes de otras neuronas a través de las sinapsis. Una neurona consta de tres partes:

1. El cuerpo de la neurona.
2. Las dendritas, que reciben las entradas.
3. El axón, que lleva la salida de la neurona a las dendritas de otras neuronas a través de los botones sinápticos.

La comunicación entre neuronas se lleva a cabo de la siguiente manera: en el cuerpo celular de las neuronas transmisoras o pre-sinápticas se genera un pulso eléctrico llamado potencial de acción. Este pulso se propaga a través del axón en dirección a las sinapsis. La información se transmite a las neuronas vecinas utilizando un proceso químico, mediante la liberación de neurotransmisores.

Estos neurotransmisores se transmiten a través de las dendritas y a su vez al cuerpo celular hacia la neurona receptora. Esta neurona receptora o post-sináptica toma la señal enviada por cientos de neuronas a través de las dendritas y la transmite al cuerpo celular. Estas señales pueden ser excitadoras (positivas) o inhibitoras (negativas). El soma es el encargado de integrar la información proveniente de las distintas neuronas. Si la señal resultante supera un determinado umbral (umbral de disparo) el soma emite un pulso que se transmite a lo largo del axón dando lugar a la transmisión eléctrica a lo largo de la neurona. Al llegar la señal al extremo del axón se liberan neurotransmisores que permiten transmitir la señal a las neuronas vecinas [34].

2.2.2. Redes Neuronales Artificiales.

Las redes neuronales artificiales son modelos matemáticos que intentan reproducir el funcionamiento del sistema nervioso. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada. Además, gracias a la posibilidad de implementación en paralelo y su respuesta relativamente rápida, constituyen un incentivo para la investigación en problemas que involucren sistemas dinámicos no lineales.

Las características principales de las RNA son las siguientes:

1. **Auto-organización y adaptabilidad:** utilizan algoritmos de aprendizaje adaptativo y auto-organización, por lo que ofrecen mejores posibilidades de procesamiento robusto y adaptativo.
2. **Procesado no lineal:** aumenta la capacidad de la red para aproximar funciones, clasificar patrones y aumenta su inmunidad frente al ruido.
3. **Procesado paralelo:** normalmente se usa un gran número de nodos de procesamiento, con alto nivel de interconectividad.

El elemento básico de computación, modelo de neurona artificial, como se muestra en la figura 2.4 se le llama habitualmente nodo o unidad. Recibe señales de entrada a las que se denominará *input*, desde otras unidades o de una fuente externa de datos. Cada *input* tiene un peso asociado w , que se va modificando en el llamado proceso de aprendizaje [33].

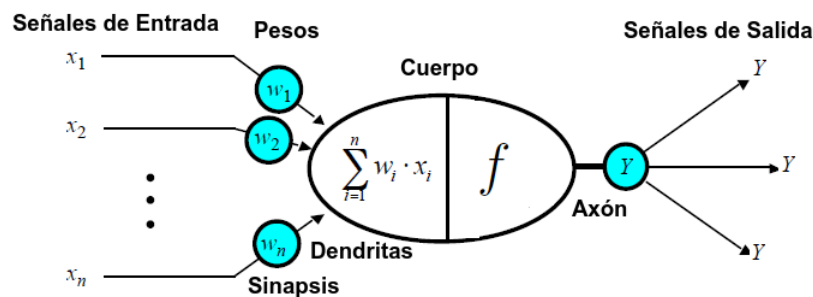


Figura 2.4: Estructura general de una neurona artificial.

En todo modelo de red neuronal artificial interviene los siguientes elementos que conforman su estructura [6]:

- **Señales de entradas** ($x_1, x_2, \dots, x_n$): pueden ser datos originales o provenientes de otras neuronas.
- **Señales de salida** ($y_1, y_2, \dots, y_m$): que corresponde con los resultados de la red (clasificación o predicción en esta investigación).
- **Los pesos sinápticos** ($w_{1k}, w_{2k}, \dots, w_{nk}$): A las entradas que provienen de otras neuronas se les asigna un peso, es decir una determinada importancia. El peso es el número que se modifica durante el entrenamiento de la red neuronal y que permite a la red adaptarse para conseguir la respuesta correcta.
- **Regla de propagación:** Se realizan diferentes operaciones con las entradas y los pesos. Con esto se busca obtener el valor del potencial que posteriormente se utilizará para realizar el procesamiento. Una de las operaciones más comunes es la suma ponderada. Otra regla puede ser la distancia euclídea, que es utilizada por los mapas auto-organizados y las redes de función de base radial.
- **La función de activación:** El valor obtenido en la regla de propagación se envía a través de la función de activación y esta define el nuevo estado o la salida de la capa. Según el entrenamiento de la red neuronal se elige la función de activación que pueden ser: lineal, escalón, sigmoidea o gaussiana.

Las características de las RNA juegan un importante papel, por ejemplo, en el procesamiento de señales e imágenes. Se usan arquitecturas que comprenden elementos de procesamiento adaptativo paralelo, combinados con estructuras de interconexiones jerárquicas [33]. Hay dos fases en la modelización con redes neuronales:

- **Fase de entrenamiento:** se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos que definen el modelo de red neuronal. Se calculan de manera iterativa, de acuerdo con los valores de los valores de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red neuronal y la salida deseada.
- **Fase de prueba:** en la fase anterior, el modelo puede que se ajuste demasiado a las particularidades presentes en los patrones de entrenamiento, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos.

Las RNA se pueden clasificar en diferentes formas dependiendo de los elementos básicos: según el número de capas pueden ser monocapa o multicapa. Según el tipo de conexión son recurrentes o no recurrentes. Según el mecanismo de aprendizaje supervisadas, no supervisadas o híbridas. Según el tipo de conexión, totalmente conectadas o parcialmente conectadas.

2.2.3. El Perceptrón.

El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples se denomina perceptrón, fue presentado el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts [35]. Un ejemplo de modelo neuronal con dos entradas x e y es representado en la figura 2.5.

Las entradas x e y son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida z es la respuesta a tal estímulo. La neurona se adapta al medio circundante y aprende de él

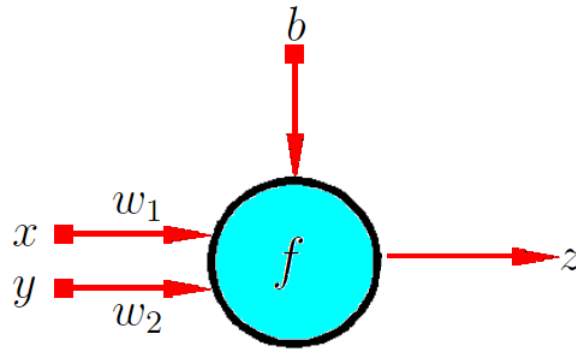


Figura 2.5: Modelo de McCulloch-Pitts para una neurona artificial.

modificando el valor de sus pesos sinápticos w_1 y w_2 y su término aditivo b . Estos son conocidos como los parámetros libres del modelo, pues los mismos pueden ser modificados y adaptados para realizar una tarea determinada. En este modelo, la salida neuronal z está dada por

$$z = f(w_1x + w_2y + b) \quad (2.1)$$

La función de activación f es seleccionada de acuerdo a la tarea realizada por la neurona. Dentro de las redes neuronales supervisadas el perceptrón multicapa (MLP) es la arquitectura más usada dentro de este tipo.

2.2.4. El Perceptrón Multicapa.

El perceptrón multicapa es una generalización del perceptrón simple y surgió como consecuencia de las limitaciones de dicha arquitectura en lo referente al problema de la separabilidad no lineal. Minsky y Papert mostraron en 1969 que la combinación de varios perceptrones simples podía resultar una solución adecuada para tratar ciertos problemas no lineales. Sin embargo, los autores no presentaron una solución al problema de cómo adaptar los pesos de la capa de entrada a la capa oculta, pues la regla de aprendizaje del perceptrón simple no puede aplicarse en este escenario. No obstante, la idea de combinar varios perceptrones sirvió de base para estudios posteriores realizados por Rumelhart, Hinton y Williams en 1986 [33, 36]. Estos autores presentaron una manera de retropropagación de los errores medidos en la salida de la red hacia las neuronas ocultas, dando lugar a la llamada regla delta generalizada. Diferentes autores han demostrado independientemente que el perceptrón multicapa es un aproximador universal, en el sentido de que cualquier función continua en un espacio R^n puede aproximarse con un perceptrón multicapa, con al menos una capa oculta de neuronas.

2.2.5. Arquitectura.

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta.

1. **Capa de entrada:** Les corresponde este nombre por ser las encargadas de recibir la información de fuentes externas a la propia red. Si la red está conectada a otras redes, las unidades de entrada reciben los datos de las salidas de las otras redes; en otros casos, simplemente,

las unidades de entrada reciben los datos que el usuario de la red introduce manualmente en el computador.

2. **Capas ocultas:** Este tipo de neuronas no se consiguen en todas las redes, algunas mínimamente complicadas las incluyen, estas no tienen relación directa ni con la información de entrada ni con la de salida, por lo que no son “visibles” al ambiente exterior de la red, de ahí su nombre, su función es procesar la información en niveles más complejos para de esta forma hacer los cálculos más eficientes.
3. **Capa de salida:** Ofrecen las señales o información al exterior de la red, dan la respuesta del sistema, son las que ofrecen datos al usuario para su posterior tratamiento.

En la figura 2.6 se puede observar un perceptrón multicapa formado por una capa de entrada, dos capas ocultas y una de salida.

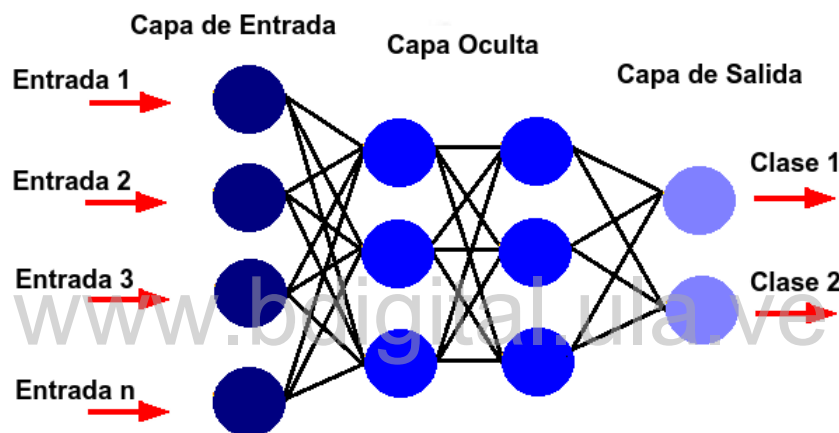


Figura 2.6: Arquitectura de un perceptrón multicapa con dos capas ocultas y dos neuronas de salida

Las conexiones entre neuronas son siempre hacia delante: las conexiones van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa; no hay conexiones laterales ni conexiones hacia atrás. Por tanto, la información siempre se transmite desde la capa de entrada hacia la capa de salida. Se considera w_{ji} como el peso de conexión entre la neurona de entrada i y la neurona oculta j , y v_{kj} como el peso de conexión entre la neurona oculta j y la neurona de salida k [33].

Normalmente, los pesos óptimos se obtienen optimizando alguna función de energía. Por ejemplo, un criterio muy utilizado en el llamado entrenamiento supervisado, es minimizar el error cuadrático medio entre el valor de salida y el valor real esperado.

2.2.6. La retropropagación.

La retropropagación es el algoritmo más utilizado en redes con conexión hacia adelante, es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas

que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de la salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias luego cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido se ajustan los errores de los pesos sinápticos de cada neurona [34].

2.2.7. Deducción del algoritmo Backpropagation.

Cuando se presenta un patrón p de entrada $X_p : x_{p1}, \dots, x_{pi}, \dots, x_{pn}$, éste se transmite a través de los pesos w_{ji} desde la capa de entrada hacia la capa oculta. Las neuronas de la capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos v_{kj} hacia la capa de salida, donde aplicando la misma operación que en el caso anterior, las neuronas de esta última capa proporcionan la salida de la red [33, 34].

La regla de propagación más simple consiste en realizar una suma de las entradas ponderadas con sus pesos sinápticos correspondientes. Este proceso se resume en lo siguiente:

$$h_{i(t)} = \sum_j w_{ij} * x_{j(t)} \quad (2.2)$$

Etapa de aprendizaje

En esta etapa el objetivo es hacer mínimo el error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de patrones denominado grupo de entrenamiento. Así, el aprendizaje en las redes backpropagation es de tipo supervisado. La función de error que se pretende minimizar para cada patrón p viene dada por:

$$e_p = \frac{1}{2} \sum_{k=1}^M (D_{pk} - Y_{pk})^2 \quad (2.3)$$

Se compara la salida obtenida Y_p con la salida deseada D_p , donde k es el índice de la neurona para las neuronas de la ultima capa, y M el total de neuronas de la misma.

A partir de esta expresión se puede obtener una medida general de error mediante:

$$e = \frac{1}{P} \sum_{p=1}^P e_p \quad (2.4)$$

Siendo p el índice de ejemplo, y P el número total de ejemplos.

La base del algoritmo backpropagation para la modificación de los pesos es la técnica conocida como gradiente decreciente.

Como e_p es función de todos los pesos de la red, el gradiente de e_p es un vector igual a la derivada parcial de e_p respecto a cada uno de los pesos. El gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta, es decir, la dirección negativa, determina el decremento más rápido en el error.

$$\Delta w_{ji} = -\alpha \frac{\partial e_p}{\partial w_{ji}} \quad (2.5)$$

Si aplicamos la regla de la cadena a la ecuación 2.5 obtenemos que:

$$\frac{\partial e_p}{\partial w_{ji}} = \frac{\partial e_p}{\partial h_j} \frac{\partial h_j}{\partial w_{ji}} \quad (2.6)$$

La ecuación 2.6 expresa la derivada del error en función de dos derivadas. La derivada del error respecto al potencial resultante h_j indica como varia el error al variar la entrada de la neurona j , mientras que la derivada con respecto al peso sináptico w_{ji} indica como varia la entrada de la neurona j al variar el peso de la conexión que va desde la neurona i hasta la neurona j .

El segundo término de la ecuación 2.6 lo podemos expresar a partir de la ecuación 2.2 de la siguiente manera:

$$\frac{\partial h_j}{\partial w_{ji}} = \frac{\partial \sum_i w_{ji} Y_{pi}}{\partial w_{ji}} = Y_{pi} \quad (2.7)$$

Si escribimos el primer termino de la ecuación 2.6 como

$$\frac{\partial e_p}{\partial h_j} = -\delta_{pj} \quad (2.8)$$

tenemos que

$$\frac{\partial e_p}{\partial w_{ji}} = -\delta_{pj} Y_{pi} \quad (2.9)$$

y por lo tanto la ecuación 2.5 queda expresada de la siguiente manera

$$\Delta w_{ji} = -\alpha \delta_{pj} Y_{pi} \quad (2.10)$$

Para calcular el valor de δ se vuelve a aplicar la regla de la cadena

$$\delta_{pj} = -\frac{\partial e_p}{\partial h_j} = -\left[\frac{\partial e_p}{\partial Y_{pj}} \frac{\partial Y_{pj}}{\partial h_j} \right] \quad (2.11)$$

El cálculo del segundo término de la ecuación 2.11 se hace por definición:

$$\frac{\partial Y_{pj}}{\partial h_j} = \frac{\partial f_j(h_j)}{\partial h_j} = f'_j(h_j) \quad (2.12)$$

Sin embargo, para el cálculo del primer término de la ecuación 2.11 es necesario distinguir entre dos casos diferentes.

- **La neurona j es una neurona de salida:** En este caso podemos obtener el segundo término a partir de la ecuación 2.3 ya que el subíndice j es igual al subíndice k

$$\frac{\partial e_p}{\partial Y_{pj}} = \frac{\partial \frac{1}{2} \sum_{j=1}^M (D_{pj} - Y_{pj})^2}{\partial Y_{pj}} = -(D_{pj} - Y_{pj}) \quad (2.13)$$

Así, la variación de los pesos de una conexión que va hacia la capa externa de la red se calcula como:

$$\delta w_{ji} = \alpha(D_{pj} - Y_{pj})f'_j(h_j)Y_{pi} \quad (2.14)$$

- **La neurona j es una neurona oculta:** En este caso es necesario aplicar nuevamente la regla de la cadena

$$\frac{\partial e_p}{\partial Y_{pj}} = \sum_k \left[\frac{\partial e_p}{\partial h_k} \frac{\partial h_k}{\partial Y_{pj}} \right] \quad (2.15)$$

Donde k es el subíndice de las neuronas que pertenecen a la próxima capa. La ecuación 2.15 la podemos reescribir utilizando la ecuación 2.2

$$\frac{\partial e_p}{\partial Y_{pj}} = \sum_k \left[\frac{\partial e_p}{\partial h_k} \frac{\partial \left[\sum_j w_{kj} Y_{pj} \right]}{\partial Y_{pj}} \right] = \sum_k \left[\frac{\partial e_p}{\partial h_k} w_{kj} \right] \quad (2.16)$$

y por la ecuación 2.9 tenemos que

$$\frac{\partial e_p}{\partial Y_{pj}} = \sum_k -\delta_{pk} w_{kj} = - \sum_k \delta_{pk} w_{kj} \quad (2.17)$$

Así, la variación de los pesos de una conexión que va desde una capa hacia otra capa de la red que no sea la externa se calcula como:

$$\Delta w_{ji} = \alpha \sum_k (\delta_{pk} w_{kj}) f'_j(h_j) Y_{pi} \quad (2.18)$$

En la implementación del algoritmo, se toma una amplitud de paso que viene dado por la tasa de aprendizaje α . A mayor tasa de aprendizaje el proceso será más rápido. Sin embargo, si la tasa de aprendizaje es muy alta puede dar lugar a oscilaciones en torno a un mínimo local. Es posible disminuir el impacto de dichas oscilaciones mediante la adición de un momento β quedando la expresión 2.10 de la siguiente manera:

$$\Delta w_{ji(t+1)} = \alpha \delta_{pj} Y_{pi} + \beta \Delta w_{ji(t)} \quad (2.19)$$

De esta manera el momento β determina el efecto en el instante $t + 1$ del cambio de los pesos realizado en el instante t . Con este momento se consigue la convergencia de la red en menor número de iteraciones, ya que si la modificación de los pesos en los instantes t y $t + 1$ es en la misma dirección, entonces el descenso por la superficie de error en $t + 1$ es mayor. En cambio, si la modificación en los pesos en los instantes t y $t + 1$ es más pequeño, lo que es adecuado, ya que esto significa que se ha pasado por un mínimo.

2.2.8. Entrenamiento de la red neuronal.

Para el entrenamiento de la red neuronal se necesitan un conjunto predefinido de pares de entradas y salidas deseadas, a los que se les denomina par o patrones de entrenamiento. La red aprende el conjunto de patrones dados como ejemplo, empleando un ciclo de dos fases: propagación y adaptación.

Fase de propagación: Después de suministrarle el conjunto de patrones de entrada a la primera capa de unidades de red, este se va propagando a través de todas las capas siguientes, hasta generar una salida. La salida obtenida se compara con la salida deseada y se calcula el error para cada unidad de salida.

Fase de adaptación: Las señales de error son transmitidas hacia atrás, partiendo de la capa de salida, hacia todos los nodos de la capa intermedia que contribuyen directamente a la salida. Las unidades de las capas intermedias solo reciben una parte de la señal total de error, basándose en una aproximación relativa de la contribución que aporte la unidad a la salida original. El proceso se repite en cada una de las capas hasta que todos los nodos de la red hayan recibido una señal de error que describa su contribución relativa al error total. Se actualizan los pesos de conexión de cada unidad, basándose en la señal de error recibida, haciendo que la red converja hacia un estado que permita codificar el grupo de patrones de entrenamiento.

Los nodos de las capas intermedias se organizan así mismo de tal modo que los distintos nodos aprendan a reconocer diferentes características del espacio total de entradas. Luego de entrenar la red, al presentarle un conjunto arbitrario de patrones de entrada que estén incompletos o presenten ruido, las unidades de las capas ocultas de la red responderán con una salida activa si la nueva entrada contiene un grupo de patrones que se asemejan a aquellas características que las unidades individuales hayan aprendido a reconocer durante su entrenamiento; pero si los patrones de entrada no contienen características de entrada para las cuales han sido entrenadas las unidades de las capas ocultas tendrán una tendencia a inhibir sus salidas. Cuando el error entre la salida de la red y la salida deseada sea lo suficientemente pequeño, se dice que la red está entrenada y se dice que la red se encuentra en la fase de producción. En esta fase la red realiza el proceso de propagación de la señal hacia adelante hasta conseguir la señal de salida, pero los pesos ya no serán ajustados [33, 34].

2.2.9. Dimensiones de la red.

Una de las principales dificultades que surge a la hora de trabajar con redes neuronales, es la de seleccionar el número de capas y nodos por cada capa para que ésta funcione satisfactoriamente. El número de nodos en la capa de entrada suele venir determinado por la naturaleza del problema, el tamaño de la capa de salida es posible determinarlo decidiendo si se desean valores analógicos o binarios en las unidades de salida.

En general, es suficiente que la red tenga tres capas, aunque a veces parece que un problema es más fácil de resolver con más de una capa oculta. Existen resultados experimentales que sugieren utilizar el menor número posible de unidades en las capas ocultas, y en caso de no lograr la convergencia de la red durante su entrenamiento, se debe aumentar el número de nodos hasta que se llegue a un tamaño final que logre el rendimiento global del sistema. Si al incrementar el número de nodos en una capa no se observan mejoras en la convergencia de la red, es conveniente dividir esta capa en dos capas sucesivas [34].

Pesos de la red: Los pesos de la red deberían recibir unos valores iniciales pequeños y aleatorios, generalmente entre ± 1 .

Tasa de aprendizaje: Controla el tamaño del cambio de los pesos en cada iteración. Se deben

evitar dos extremos: un ritmo de aprendizaje demasiado pequeño puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar atrapado en un mínimo local; en cambio, un ritmo de aprendizaje demasiado grande puede conducir a inestabilidades en la función de error, lo cual evitará que se produzca la convergencia debido a que se darán saltos en torno al mínimo sin alcanzarlo [33, 37]. Por tanto, se recomienda elegir un ritmo de aprendizaje lo más grande posible sin que provoque grandes oscilaciones. En general, el valor de la tasa de aprendizaje suele estar comprendida entre 0.01 y 0.25.

2.2.10. Función de activación de las neuronas.

La función f de la ecuación 2.1 es la llamada función de activación. Para el perceptrón multicapa, las funciones de activación mas utilizadas son la función sigmoideal y la función tangente hiperbólica. Dichas funciones poseen como imagen un intervalo continuo de valores dentro de los intervalos $[0, 1]$ y $[-1, 1]$ respectivamente, y vienen dadas por las siguientes ecuaciones:

Función sigmoideal:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.20)$$

Función tangente hiperbólica:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.21)$$

Ambas son funciones crecientes con dos niveles de saturación: el máximo, que proporciona salida 1, y el mínimo, salida 0 para la función sigmoideal y salida -1, para la tangente hiperbólica, como se observa en la figura 2.7 [38]

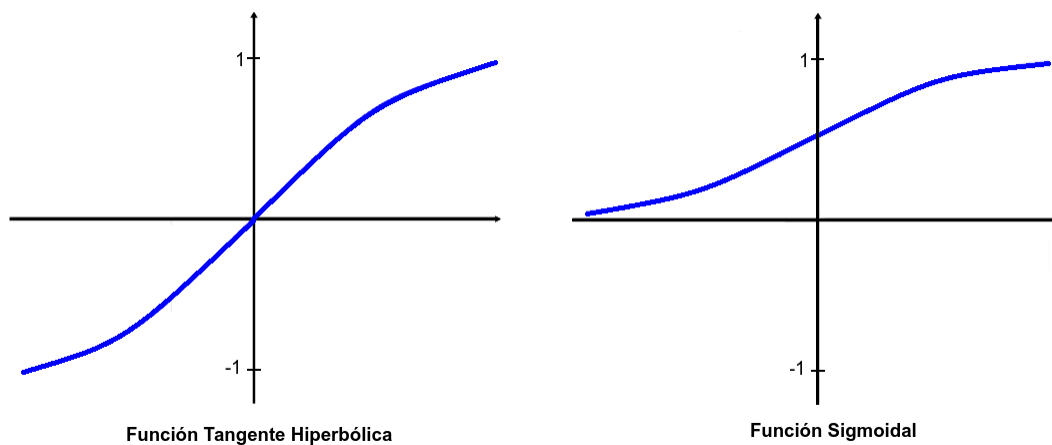


Figura 2.7: Función tangente hiperbólica y función sigmoideal

Si bien ninguna de las funciones usuales, tales como: sigmoideal, lineal, tangente hiperbólica, contienen en ningún momento inestabilidades en el umbral de decisión, es cierto que desde el punto de vista fisiológico se justifica la incorporación de una fluctuación extendida en un rango cercano a la región donde la función toma el valor (1) o (-1). La función de activación tangente hiperbólica

sugiere un comportamiento monótono en la clasificación de clases dado que a partir de un límite dado de atributos la función se hace (1) o (-1) independientemente del conjunto de entradas [34].

En algunas ocasiones, la función de activación en el perceptrón multicapa es común a todas las neuronas de la red. En otras ocasiones y dependiendo de la naturaleza del problema, las neuronas de salida se distinguen del resto de neuronas de la red, utilizando otro tipo de función de activación. En este caso, las más usadas son la función identidad y la función escalón.

2.2.11. Aplicaciones de un perceptrón multicapa.

Las redes neuronales pueden aplicarse con éxito en la visión artificial, en el procesamiento de señales e imágenes, reconocimiento del habla y de caracteres, sistemas expertos, análisis de imágenes médicas, control remoto, control de robots e inspección industrial. La mayoría de estas aplicaciones consisten en realizar un reconocimiento de patrones, como: buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado. Desde el punto de vista de los casos de aplicación, la ventaja de las redes neuronales reside en el procesamiento paralelo, adaptativo y no lineal.

En general, una red del tipo perceptrón multicapa intenta resolver dos tipos de problemas:

- **Problemas de predicción:** que consisten en la estimación de una variable continua de salida, a partir de la presentación de un conjunto de variables predictoras de entrada.
- **Problemas de clasificación:** que consisten en la asignación de la categoría de pertenencia de un determinado patrón a partir de un conjunto de variables predictoras de entrada.

2.2.12. Problemas de clasificación.

Las técnicas de clasificación se enmarcan dentro del aprendizaje supervisado porque para cada dato se tiene información sobre sus variables de entrada y de salida; es decir, cada dato u objeto está etiquetado. Los datos son objetos caracterizados por atributos que pertenecen a diferentes clases (etiquetas discretas). El problema más simple de clasificación es la decisión binaria que se ocupa de decidir si una muestra pertenece a una clase o a otra. Como se puede inferir, este tipo de problemas son mejor implementados ocupando únicamente una neurona de salida. La red será entrenada para producir una salida de activación alta para una decisión y una salida baja para otra.

Cuando una red entrenada se pone a prueba con una muestra desconocida, se decidirá a que categoría corresponde basados en el nivel de activación logrado por la neurona de salida. Si la activación es al menos tan grande como un umbral predefinido, entonces estamos en capacidad de hacer la decisión. En muchos casos, el umbral se ubica justo en medio de los dos posibles límites. Esto tiene sentido lógico y en la práctica es usualmente efectivo en la ausencia de información adicional.

2.3. Análisis de Componentes Principales.

El análisis de componentes principales PCA (por sus siglas en inglés, *Principal Component Analysis*) es una técnica estadística de síntesis de la información o reducción del número de variables perdiendo la menor cantidad de información posible. Estas técnicas fueron desarrolladas inicialmente por Pearson a finales del siglo XIX y posteriormente fueron estudiadas por Hotelling en los años 30 del siglo XX [39]. Sin embargo, no se popularizaron hasta la aparición de las computadoras. Para estudiar las relaciones que se presentan entre p variables correlacionadas (que miden información común) se puede transformar el conjunto original de variables en otro conjunto de nuevas variables incorreladas entre sí (que no tenga redundancia en la información) llamado conjunto de componentes principales.

Las nuevas variables son combinaciones lineales de las anteriores y se van construyendo según el orden de importancia en cuanto a la variabilidad total que recogen de la muestra. De modo ideal, se busca $m < p$ variables que sean combinaciones lineales de las p originales y que estén incorreladas, recogiendo la mayor parte de la información o la variabilidad de los datos [39, 40].

2.3.1. Autovalores y autovectores.

En esta sección se hace una descripción intuitiva de dos de los conceptos matemáticos que se aplican en el PCA para facilitar el entendimiento del cálculo de componentes principales [41].

■ Autovectores (*eigenvectors*)

Los *eigenvectors* de una matriz son todos aquellos vectores que, al multiplicarlos por dicha matriz, resultan en el mismo vector o en un múltiplo entero del mismo. Estos tienen una serie de propiedades matemáticas específicas:

- Los *eigenvectors* solo existen para matrices cuadradas y no para todas. En el caso de que una matriz $n \times n$ tenga *eigenvectors*, el número de ellos es n .
- Si se escala un *eigenvector* antes de multiplicarlo por la matriz, se obtiene un múltiplo de mismo *eigenvector*. Esto se debe a que si se escala un vector multiplicándolo por cierta cantidad, lo único que se consigue es cambiar su longitud pero la dirección es la misma.
- Todos los *eigenvector* de una matriz son ortogonales entre ellos, independientemente de las dimensiones que tengan.

Dada la propiedad de que multiplicar un *eigenvector* solo cambia su longitud pero no su naturaleza, es frecuente escalarlos de tal forma que su longitud sea 1. De este modo se consigue que todos ellos estén estandarizados.

■ Autovalores (*eigenvalues*)

Cuando se multiplica una matriz por alguno de sus *eigenvectors* se obtiene un múltiplo del vector original, es decir, el resultado es ese mismo vector multiplicado por un número. Al valor por el que se multiplica el *eigenvector* resultante se le conoce como *eigenvalue*. A todo

eigenvector le corresponde un *eigenvalue* y viceversa.

En el método de PCA, cada una de las componentes se corresponde con un *eigenvector*, y el orden de componente se establece por orden decreciente de *eigenvalue*. Así la primera componente es el *eigenvector* con el *eigenvalue* asociado más alto.

2.3.2. Cálculo de las componentes principales.

Se considera una serie de variables $(x_1, x_2, \dots, x_p)$ sobre un grupo de objetos o individuos y se trata de calcular, a partir de ellas, un nuevo conjunto de variables $(y_1, y_2, \dots, y_p)$, incorreladas entre sí, cuyas varianzas vayan decreciendo progresivamente [39].

Cada y_j (donde $j = 1, \dots, p$) es una combinación lineal de las $(x_1, x_2, \dots, x_p)$ originales, es decir

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jp}x_p = \mathbf{a}'_j \mathbf{x} \quad (2.22)$$

Siendo $\mathbf{a}'_j = (a_{1j}, a_{2j}, \dots, a_{pj})$ un vector de constantes y

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$$

Si se quiere maximizar la varianza, una forma será aumentar los coeficientes a_{ij} . Por ello, para mantener la ortogonalidad de la transformación se impone que el módulo del vector $\mathbf{a}'_j = (a_{1j}, a_{2j}, \dots, a_{pj})$ sea 1, es decir

$$\mathbf{a}'_j \mathbf{a}_j = \sum_{k=1}^p a_{kj}^2 = 1 \quad (2.23)$$

El primer componente se calcula eligiendo $\mathbf{a}'_1$ de modo que y_1 tenga la mayor varianza posible, sujeta a la restricción de que $\mathbf{a}'_1 \mathbf{a}_1 = 1$. El segundo componente principal se calcula obteniendo $\mathbf{a}'_2$ de modo que la variable obtenida, y_2 esté incorrelada con y_1 . Del mismo modo se eligen $y_1, y_2, \dots, y_p$, incorrelados entre sí, de manera que las variables aleatorias obtenidas vayan teniendo cada vez menor varianza.

2.3.3. Proceso de extracción de factores.

Se quiere elegir $\mathbf{a}_1$ de modo que se maximice la varianza de y_1 sujeta a la restricción de que $\mathbf{a}'_1 \mathbf{a}_1 = 1$
 $\Rightarrow g$

$$\text{Var}(y_j) = \text{Var}(\mathbf{a}'_1 \mathbf{x}) = \mathbf{a}'_1 \Sigma \mathbf{a}_1 \Rightarrow f \quad (2.24)$$

El método habitual para maximizar una función de varias variables sujeta a restricciones es el método de los multiplicadores de Lagrange, definido por: $L = f + \lambda g$

El problema consiste en maximizar la función $\mathbf{a}'_1 \Sigma \mathbf{a}_1$ sujeta a la restricción $\mathbf{a}'_1 \mathbf{a}_1 = 1$. Se puede observar que la incógnita es $\mathbf{a}_1$, por lo tanto la función L será:

$$L(\mathbf{a}_1) = \mathbf{a}_1' \Sigma \mathbf{a}_1 - \lambda(\mathbf{a}_1' \mathbf{a}_1 - 1) \quad (2.25)$$

y se busca el máximo derivando e igualando a cero la ecuación 2.25

$$\frac{\partial L(\mathbf{a}_1)}{\partial \mathbf{a}_1} = 2\Sigma \mathbf{a}_1 - 2\lambda \mathbf{I} \mathbf{a}_1 = 0 \Rightarrow (\Sigma - \lambda \mathbf{I}) = 0 \quad (2.26)$$

Teniéndose un sistema lineal de ecuaciones. Por el teorema de Rouché-Frobenius, para que el sistema tenga una solución distinta de 0 la matriz tiene que ser singular. Esto implica que el determinante debe ser igual a cero:

$$|\Sigma - \lambda \mathbf{I}| = 0 \quad (2.27)$$

y de este modo, λ es un autovalor de Σ . La matriz de covarianzas Σ es de orden p y si además es definida positiva, tendrá p autovalores distintos, $\lambda_1, \lambda_2, \dots, \lambda_p$ tales que, por ejemplo, $\lambda_1 > \lambda_2 > \dots > \lambda_p$.

Se tiene que, desarrollando la expresión 2.27

$$(\Sigma - \lambda \mathbf{I}) \mathbf{a}_1 = 0 \quad (2.28)$$

$$\Sigma \mathbf{a}_1 - \lambda \mathbf{I} \mathbf{a}_1 = 0 \quad (2.29)$$

$$\Sigma \mathbf{a}_1 = \lambda \mathbf{I} \mathbf{a}_1 \quad (2.30)$$

entonces la ecuación 2.24

$$\text{Var}(y_1) = \text{Var}(\mathbf{a}_1' \mathbf{x}) = \mathbf{a}_1' \Sigma \mathbf{a}_1 = \mathbf{a}_1' \lambda \mathbf{I} \mathbf{a}_1 = \lambda \mathbf{a}_1' \mathbf{a}_1 = \lambda \cdot 1 = \lambda \quad (2.31)$$

Luego, para maximizar la varianza de y_1 se tiene que tomar el mayor autovalor, digamos λ_1 , y el correspondiente autovector $\mathbf{a}_1$. En realidad, $\mathbf{a}_1$ es un autovector que da la combinación de las variables originales que tiene mayor varianza, esto es, si $\mathbf{a}_1' = (a_{11}, a_{12}, \dots, a_{1p})$, entonces

$$y_1 = \mathbf{a}_1' \mathbf{x} = a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p \quad (2.32)$$

La segunda componente principal, $y_2 = \mathbf{a}_2' \mathbf{x}$, se obtiene mediante un argumento parecido. Además, se quiere que y_2 esté incorrelado con la componente anterior y_1 , es decir, $\text{Cov}(y_2, y_1) = 0$.

$$\text{Cov}(y_2, y_1) = \text{Cov}(\mathbf{a}_2' \mathbf{x}, \mathbf{a}_1' \mathbf{x}) \quad (2.33)$$

Por definición,

$$\text{Cov}(X, Y) = \Xi[(X - \mu)(Y - \mu)] = \Sigma \quad (2.34)$$

Por lo tanto, sustituyendo la ecuación 2.34 en la ecuación 2.33 queda

$$\text{Cov}(y_2, y_1) = \mathbf{a}_2' \cdot \Xi[(\mathbf{x} - \mu)(\mathbf{x} - \mu)'] \cdot \mathbf{a}_1 = \mathbf{a}_2' \Sigma \mathbf{a}_1 \quad (2.35)$$

Es decir, se requiere que $\mathbf{a}_2' \Sigma \mathbf{a}_1 = 0$.

Como se tenía que $\Sigma \mathbf{a}_1 = \lambda \mathbf{I} \mathbf{a}_1$, lo anterior es equivalente a

$$\mathbf{a}_2' \Sigma \mathbf{a}_1 = \mathbf{a}_2' \lambda \mathbf{I} \mathbf{a}_1 = \lambda \mathbf{a}_2' \mathbf{a}_1 = 0, \quad (2.36)$$

esto es equivalente a que $\mathbf{a}_2' \mathbf{a}_1 = 0$, es decir, que los vectores sean ortogonales.

De este modo, se tendrá que maximizar la varianza de y_2 , es decir, $\mathbf{a}_2' \Sigma \mathbf{a}_2$, sujeta a las siguientes restricciones

$$\begin{aligned}\mathbf{a}_2' \Sigma \mathbf{a}_2 &= 1 \\ \mathbf{a}_2' \Sigma \mathbf{a}_1 &= 0\end{aligned}$$

Se toma la función:

$$L(\mathbf{a}_2) = \mathbf{a}_2' \Sigma \mathbf{a}_2 - \lambda(\mathbf{a}_2' \mathbf{a}_2 - 1) - \delta \mathbf{a}_2' \mathbf{a}_1 \quad (2.37)$$

y derivando la ecuación 2.37,

$$\frac{\partial L(\mathbf{a}_2)}{\partial \mathbf{a}_2} = 2\Sigma \mathbf{a}_2 - 2\lambda \mathbf{a}_2 - \delta \mathbf{a}_1 = 0 \quad (2.38)$$

si se multiplica la ecuación 2.38 por $\mathbf{a}_1'$, entonces

$$2\mathbf{a}_1' \Sigma \mathbf{a}_2 - \delta = 0 \quad (2.39)$$

Luego,

$$\delta = 2\mathbf{a}_1' \Sigma \mathbf{a}_2 = 2\mathbf{a}_2' \Sigma \mathbf{a}_1 = 0, \quad (2.40)$$

ya que $\text{Cov}(y_2, y_1) = 0$. De este modo, la ecuación 2.38 queda finalmente como:

$$\frac{\partial L(\mathbf{a}_2)}{\partial \mathbf{a}_2} = 2\Sigma \mathbf{a}_2 - 2\lambda \mathbf{a}_2 - \delta \mathbf{a}_1 = 2\Sigma \mathbf{a}_2 - 2\lambda \mathbf{a}_2 = 0 \quad (2.41)$$

la cual puede escribirse como $(\Sigma - \lambda I)\mathbf{a}_2 = 0$

Usando los mismos razonamientos que antes, se elige λ como el segundo mayor autovalor de la matriz Σ con su autovector asociado $\mathbf{a}_2$.

Los razonamientos anteriores se pueden extender, de modo que al j -ésimo componente le correspondería el j -ésimo autovalor.

Entonces todos los componentes $\mathbf{y}$ (en total p) se pueden expresar como el producto de una matriz formada por los autovectores, multiplicada por el vector $\mathbf{x}$ que contiene las variables originales $x_1, x_2, \dots, x_p$

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (2.42)$$

donde

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}, \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$$

como

$$\begin{aligned}\text{Var}(y_1) &= \lambda_1 \\ \text{Var}(y_2) &= \lambda_2 \\ &\vdots \\ \text{Var}(y_p) &= \lambda_p\end{aligned}$$

la matriz de covarianza de $\mathbf{y}$ será

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_p \end{pmatrix}$$

Se tiene que

$$\Lambda = (Y) = A' \text{Var}(X) A = A' \Sigma A$$

o bien $\Sigma = A \Lambda A'$

ya que A es una matriz ortogonal, entonces $AA' = I$.

Calcular las componentes principales equivale a aplicar una transformación ortogonal A a las variables $\mathbf{x}$ (ejes originales) para obtener unas nuevas variables $\mathbf{y}$ incorreladas entre sí.

2.3.4. Gráfico de sedimentación.

Una forma gráfica para determinar el número de componentes principales a retener es un gráfico de sedimentación. La idea y nombre pertenece a Cattell [1966]. Se obtiene al representar en las ordenadas las raíces características (los autovalores) y en las abscisas el índice de las mismas. En algunos casos, se ploteará el logaritmo de los autovalores cuando el primer autovalor es muy grande.

Uniando todos los puntos se obtiene una figura que en general se parece al perfil de una montaña con una fuerte pendiente hasta llegar a la base como se muestra en la figura 2.8. De acuerdo con el criterio se retiene todas aquellas componentes previas a la zona de sedimentación.

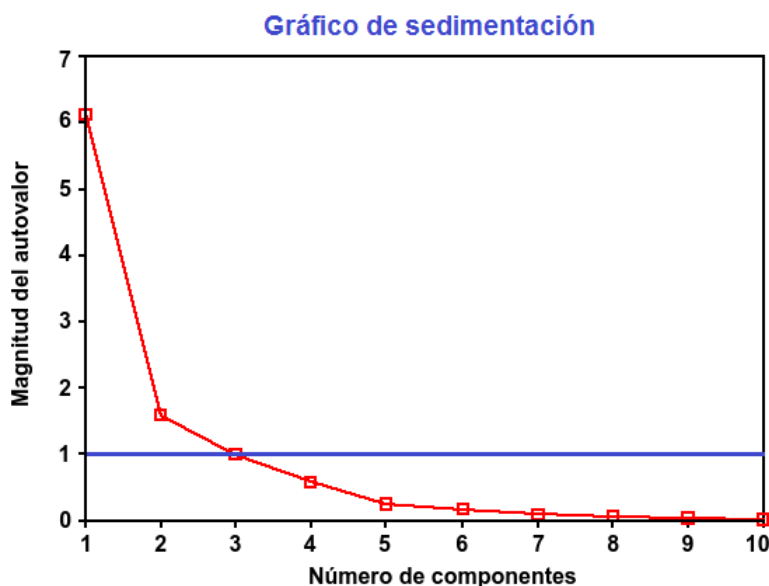


Figura 2.8: Gráfico de sedimentación para 10 autovalores.

Capítulo 3

Marco Metodológico.

3.1. Diagrama de flujo.

En la figura 3.1 se presenta un diagrama de flujo donde se identifican 5 etapas para la adquisición y análisis de los datos, las cuales son las siguientes:

- **I. Adquisición de los datos:** Se describe el hardware (tarjeta GY-80, giróscopo L3G4200D, acelerómetro ADXL345, magnetómetro HMC5883L, pulsómetro, Arduino Due) y software (MATLAB, el entorno de desarrollo de Arduino, Aqevent y las características del computador) empleados. Se presenta el montaje experimental, las mediciones básicas y la población en estudio con los criterios de inclusión y exclusión.
- **II. Tratamiento de los datos:** Se muestra la preparación previa de la data, la reducción a una misma longitud, la normalización, depuración de los datos y la creación del nuevo formato de los mismos para la aplicación de PCA.
- **III. Aplicación de PCA:** Comienza con la construcción de la matriz de covarianza y la obtención de los autovalores y autovectores (espacio V), se realiza el gráfico de sedimentación y se proyectan los sujetos en el nuevo espacio creando una llave única de identificación, se presentan los sujetos a reconocer y se determina la correlación existente entre ellos, se gráfica la matriz de confusión y se crea la base de conocimiento.
- **IV. Entrenamiento de la red neuronal:** Se diseña la arquitectura de la red neuronal, se inicia el entrenamiento (fase test), se gráfica la matriz de confusión y se obtiene la función de la red que contiene los pesos y bias (sesgo de la neurona para enviar un 0 o un 1) del entrenamiento, estos se guardan en una estructura llamada parámetros de la red.
- **V. Validación del método:** Se proyectan los nuevos sujetos a clasificar sobre el espacio V (fase de producción) y con los datos de los parámetros de la red se construyen los dendrogramas que visualizan la clasificación por tarea y combinaciones entre ellas.

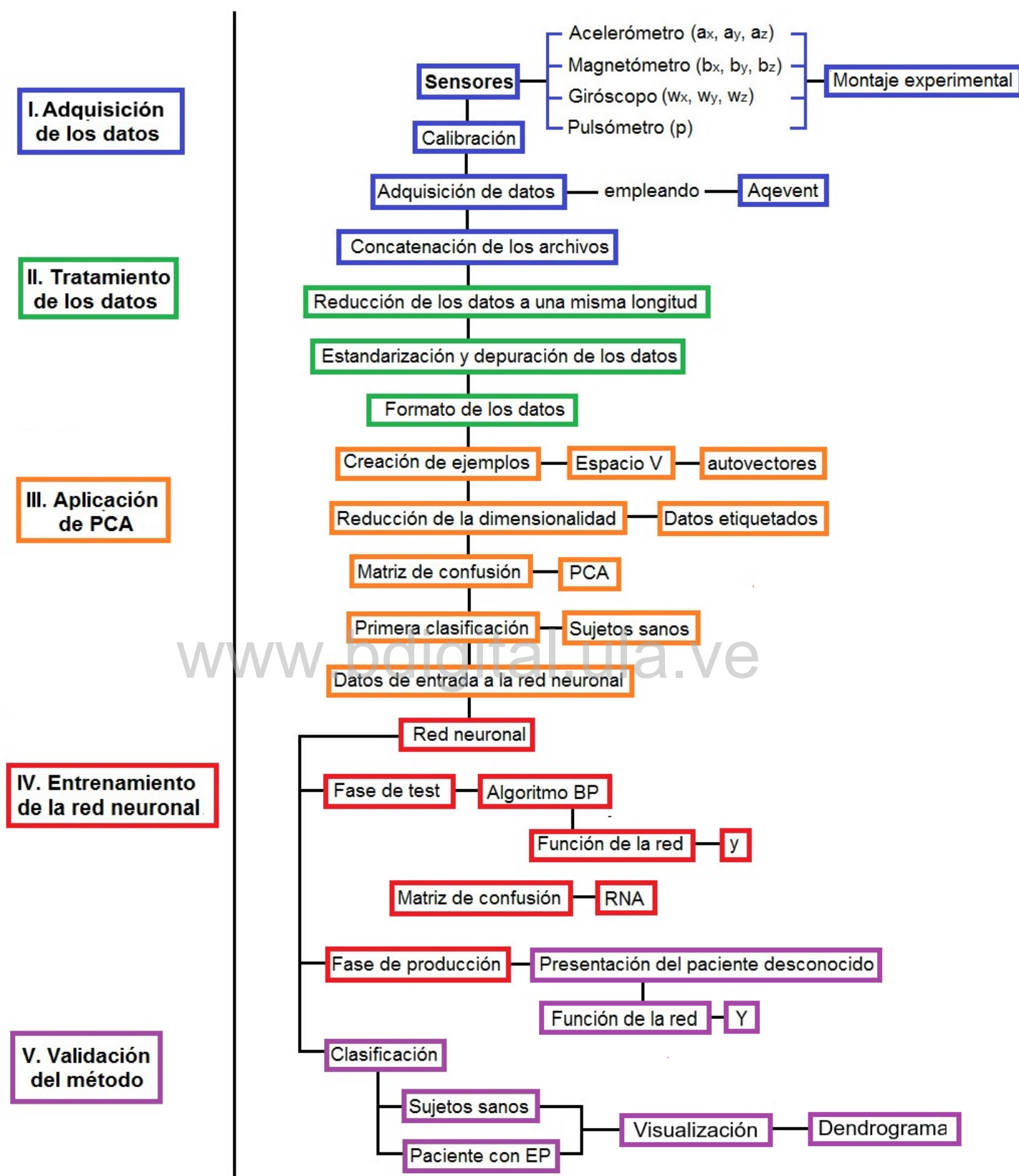


Figura 3.1: Diagrama de flujo.

3.2. I. Adquisición de los Datos.

3.2.1. Hardware.

3.2.1.1. Sistemas Microelectromecánicos.

Los sistemas microelectromecánicos (MEMS, por sus siglas en inglés *Microelectromechanical systems*). Son definidos típicamente como dispositivos de pequeñas dimensiones compuestos por elementos activos y pasivos microfabricados, que realizan diferentes funciones como percepción, procesamiento de datos, comunicación y actuación sobre el entorno. Los tipos de dispositivos MEMS pueden variar desde estructuras relativamente simples que no tienen ninguna parte móvil, hasta sistemas electromecánicos muy complejos en la que múltiples elementos se mueven bajo el control de la electrónica integrada.

Gracias a los avances en el campo de los semiconductores, los MEMS son una tecnología que puede aplicarse utilizando una gran diversidad de materiales y técnicas de fabricación; los sensores actuales son capaces de medir hasta tres ejes simultáneamente con voltajes de alimentación tan pequeños que una batería pequeña es suficiente para activarlos y con rangos de medición y resolución suficientes para aplicaciones de medición de movimiento del cuerpo humano. Varios investigadores han realizado diversos estudios relacionados con la evaluación de algunos parámetros de movimiento humano muy específicos con aplicaciones para rehabilitación, diagnóstico de enfermedades y valoración de efectos por consumo de medicamentos [42].

Además, existen unidades que combinan tres tipos de sensores (acelerómetros, giróscopos y magnetómetros) en un mismo encapsulado lo que impacta en tamaño del equipo final, el consumo de energía y el incremento en la precisión de la medición. La motivación para el uso de esta tecnología con respecto a los dispositivos utilizados hasta ahora es que ofrece, además de un precio módico, un menor consumo, un peso más reducido y un alto desempeño.

3.2.1.2. Sensor: Tarjeta GY-80.

Es un módulo compacto como se muestra en la figura 3.2 que incluye un giróscopo, un acelerómetro, un magnetómetro y un sensor de presión barométrica y temperatura. Todos los sensores individuales son accesibles a través del protocolo I2C por lo que solo necesita 4 conexiones para acceder a todos los sensores: SDA (datos I2C), SCL (Clock I2C), GND y VCC (3.3 V o 5 V) [43].

Entre sus principales características destacan:

- Giróscopo L3G4200D
- Acelerómetro ADXL345
- Magnetómetro HMC5883L
- Barómetro BMP085
- Protocolo de comunicación I2C
- Voltaje de operación: 3.3 V – 5 V

- Peso: 5 g
- Dimensiones: 2.5 cm x 1.7 cm x 0.2 cm
- Precio: 10 \$

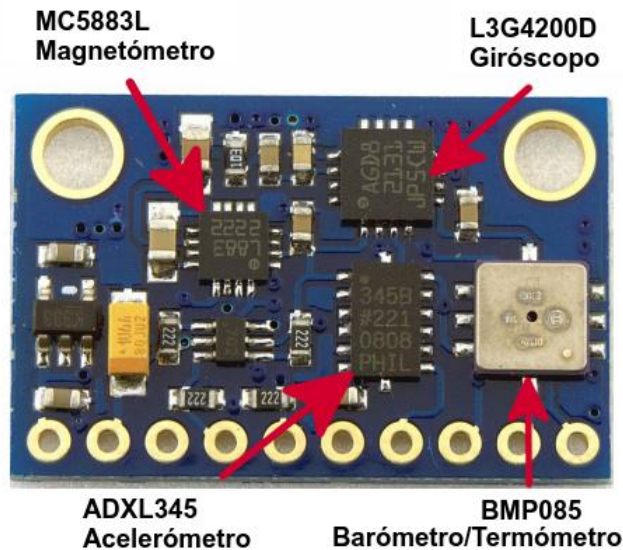


Figura 3.2: Tarjeta GY-80.

www.bdigital.ula.ve

3.2.1.3. Giróscopo L3G4200D.

El L3G4200D es un sensor de velocidad angular de 3 ejes y de baja potencia. Incluye un sensor y una interfaz IC capaz de proporcionar la velocidad angular medida al exterior a través de una interfaz digital (I2C o SPI). El elemento sensor se fabrica utilizando un proceso micromecanizado desarrollado por STMicroelectronics. La interfaz IC se fabrica utilizando un CMOS, proceso que permite un alto nivel de integración para diseñar un mejor circuito [44].

Características:

- 3 escalas full seleccionables (250/500/2000 dps).
- Interfaz de salida I2C/SPI.
- 16 bits en los datos de salida.
- Filtros pasa bajos y pasa alto integrados, con banda ancha seleccionable por el usuario.
- Amplio voltaje de alimentación: 2.4 V a 3.6 V.
- Ultra estable sobre la temperatura y el tiempo.
- Tecnología FIFO.

3.2.1.4. Acelerómetro ADXL345.

El ADXL345 es un acelerómetro MEMS capacitivo que detecta la aceleración en los ejes X, Y, Z. El dispositivo mide la aceleración dinámica resultante del movimiento o choque y la aceleración estática, como la gravedad. El sensor es una estructura micromecanizada de superficie de polisilicio construido en la parte superior de una oblea de silicio. Pequeños resortes se suspenden sobre la superficie de la oblea y proporcionan una resistencia contra las fuerzas debidas a la aceleración aplicada, la deflexión de la estructura se mide utilizando capacitores diferenciales que consisten en placas fijas independientes y placas unidas a la masa móvil, la aceleración desvía la masa de prueba y desbalancea el condensador diferencial, dando como resultado una salida del sensor cuya amplitud es proporcional a la aceleración [45].

Características:

- Bajo consumo de corriente.
- Alta resolución de hasta 13 bits.
- Sensibilidad de 40 mg/LSB
- Rango de medición ajustable entre $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$
- Protocolo de comunicación SPI o I2C.
- Tensión de alimentación entre 2.0 V a 3.6 V.
- Tecnología FIFO.

3.2.1.5. Magnetómetro HMC5883L.

El HMC5883L es un soporte de superficie del módulo multi-chip diseñado para la detección magnética de campo bajo una interfaz digital. Este sensor consta de 3 ejes y utiliza interfaz I2C. El HMC5883L utiliza la tecnología de Honeywell Anisotrópico Magnetorresistivo (AMR) que proporciona ventajas sobre otras tecnologías de sensores magnéticos. Estos sensores anisótropicos disponen de precisión en la sensibilidad de cada eje. Están contruidos con tecnología del estado sólido, está diseñado para medir la magnitud de campos terrestres desde los miligauss hasta 8 gauss [46].

Características:

- 3 ejes sensores.
- 12 bits ADC junto con un bajo nivel de ruido AMR.
- Operaciones con bajo voltaje: 2.16 V a 3.6 V
- Bajo poder de consumo ($100 \mu A$)
- Interfaz de comunicación I2C
- Tecnología FIFO.

3.2.1.6. Pulsómetro: *Pulse sensor*.

El *Pulse sensor* es un sensor para Arduino inmediatamente utilizable que permite medir el pulso de una persona. El sensor está perfectamente adecuado para estudiantes, artistas, deportistas y desarrolladores, que quieren incorporar datos creados del pulso de una persona en un proyecto. Este sensor combina un led infrarrojo y un fototransistor que permite medir el ritmo cardíaco en el dedo o cualquier otra parte del cuerpo, amplificado con un circuito de cancelación de ruido que permite una lectura rápida y fiable de los datos como se muestra en la figura 3.3. Su consumo de corriente es solo de 4 mA y el voltaje de operación es de 5 V [47].

- Voltaje: 3 V a 5 V.
- Consumo: 4 mA a 5 V.
- Diámetro: 16 mm.
- Grosor: 3 mm.
- Precio: 10 \$



Figura 3.3: Pulsómetro para Arduino.

3.2.1.7. Arduino.

Arduino es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador. Con software y hardware flexibles y fáciles de utilizar, Arduino ha sido diseñado para adaptarse a las necesidades de todo tipo de público, desde aficionados, hasta expertos en robótica o equipos electrónicos. También consta de un simple entorno de desarrollo, que permite la interacción con la plataforma de manera muy sencilla. Se puede definir por tanto como una sencilla herramienta de contribución a la creación de prototipos, entornos u objetos interactivos destinados a proyectos multidisciplinarios [48].

A través de Arduino se puede recopilar multitud de información del entorno sin excesiva complejidad. Gracias a sus pines de entrada, se puede jugar con toda una gama de sensores que brindan la capacidad de controlar o actuar sobre ciertos factores del entorno que le rodean. Este dispositivo potencia el proceso de aprendizaje mediante la experimentación, además de que provee un soporte

fácil y flexible para estudiantes, profesores e investigadores, de tal manera que puedan contar con una base para poder desarrollar múltiples y diversos proyectos en ciencias, tecnologías e ingeniería.

3.2.1.8. Arduino Due.

El Arduino Due es una placa electrónica basada en la CPU Atmel SAM3X8E ARM Cortex-M3. Es la primera placa Arduino basada en un microcontrolador de núcleo ARM de 32 bits. Cuenta con 54 pines digitales de entrada/salida (E/S), 12 entradas analógicas, 4 UARTs (puertos serie de hardware), un reloj de 84 MHz, una conexión capaz USB OTG, 2 DAC (de digital a analógico), 2 TWI, un conector de alimentación, un conector de SPI, un conector JTAG, un pulsador de reset y un pulsador de borrado. La placa contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a una computadora con un cable micro-USB o alimentarlo con un adaptador de CA a CC o una batería. El Due es compatible con todos los escudos de Arduino que funcionan a 3.3 V y cumplan con el estándar 1.0 Arduino pinout. A diferencia de la mayoría de las placas Arduino, la placa Arduino Due funciona a 3.3 V. La tensión máxima que los pines de E/S pueden tolerar es 3.3 V. La aplicación de tensiones superiores a 3.3 V a cualquier pin de E/S podría dañar la placa [49].

En la figura 3.4, se puede observar la placa Arduino Due y en la tabla 3.1 sus especificaciones técnicas.

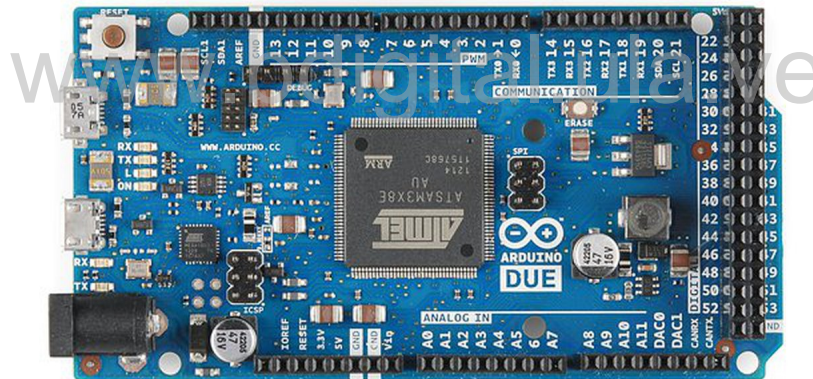


Figura 3.4: Placa Arduino Due.

3.2.2. Software.

3.2.2.1. MATLAB.

MATLAB (*Matrix Laboratory*) es un lenguaje de alto nivel, diseñado para proveer facilidades de cálculos numéricos, visualización y programación en un entorno muy sencillo de utilizar. MATLAB ofrece un entorno de desarrollo integrado (IDE) muy versátil con un lenguaje de programación propio (lenguaje M). Proporciona una interfaz con líneas de comando para resolver problemas lineales y no lineales, además de otros experimentos en forma numérica. Utiliza una notación matemática simple para tratar los problemas y resolverlos. Su poder radica en el manejo de matrices en forma eficiente, también incluye su propio compilador lo cual permite extender su uso permitiendo al

Elemento	Información
Microcontrolador	AT91SAM3X8E
Voltaje de operación	3.3 V
Voltaje recomendado de entrada	7-12 V
Pines de entrada y salida digitales	54 pines I/O (12 proveen salida PWM)
Corriente de salida total en los pines E/S	130 mA
Corriente DC máxima en el pin de 3.3 V	800 mA
Corriente DC máxima en el pin de 5 V	800 mA
Memoria Flash	512 KB toda disponible
SRAM	96 KB (en dos bancos de: 64 KB y 32 KB)
Velocidad de reloj	84 MHz
Dimensiones	101.52 mm x 53.3 mm
Peso	36 g
Precio	25 \$

Tabla 3.1: Especificaciones técnicas de la placa Arduino Due.

usuario crear sus propios comandos, clases y funciones. Es compatible con los lenguajes de programación: C y Fortran [50].

Se trata de un software matemático para cálculo científico (aritmético y simbólico) basado en matrices. Entre sus prestaciones básicas se hallan:

- Realizar cálculos aritméticos (como una calculadora).
- Realizar cálculo simbólico (con la posibilidad de hacer operaciones como derivar funciones, calcular primitivas, etc.) y exacto.
- Programar en un lenguaje no compilado (script).
- Realizar gráficos en dos y tres dimensiones.
- La representación de datos y funciones.
- La implementación de algoritmos.
- La creación de interfaces de usuario (GUIDE).
- La comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Las capacidades de MATLAB se pueden ampliar con las llamadas cajas de herramientas o *toolboxes*. Así, por ejemplo, se pueden utilizar *toolboxes* de matemáticas y optimización, estadística y análisis de datos, diseño de sistemas de control y análisis, procesamiento de señales y comunicaciones, procesamiento de imagen, pruebas y medidas, biología computacional, redes neuronales artificiales, modelado y análisis financiero, desarrollo de aplicaciones o informes y conexión a bases de datos, lo que facilita el estudio de sistemas dinámicos y su regulación.

3.2.2.2. Entorno de desarrollo en Arduino.

El entorno de desarrollo integrado, llamado IDE (sigla en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware a través del puerto serie [51]. Los programas de Arduino están compuestos por un solo fichero con extensión “.ino”, aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

En la figura 3.5 se puede observar que el IDE de Arduino se compone de:

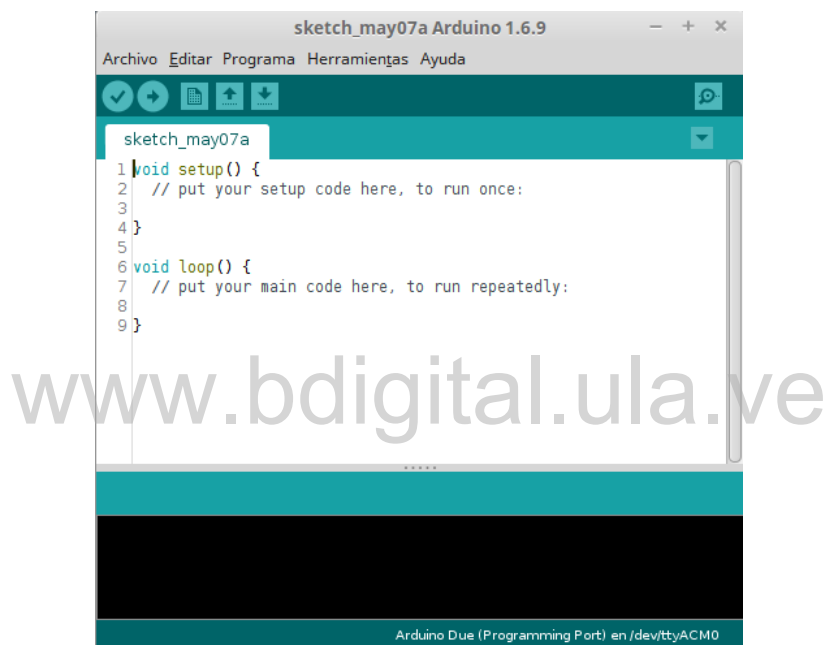


Figura 3.5: Entorno de desarrollo en Arduino

- **Un editor de texto:** donde se escribe el código del programa el cual admite las posibilidades de cortar, pegar, buscar y remplazar texto.
- **Un área de mensajes:** se muestra, tanto la información mientras se cargan los programas, como los posibles errores que se tengan a la hora de compilar, ya sea por problemas en el código del *sketch*, por fallo en la detección de Arduino en el puerto USB, o por cualquier otro problema que sea detectado.
- **Una barra de herramientas:** donde se puede acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino.

Desde la barra de herramientas se tiene acceso directo a las principales funcionalidades que ofrece el IDE de Arduino, como por ejemplo: verificar el proceso de carga, crear un nuevo *sketch*, abrir un *sketch* ya existente, guardar los programas, abrir el monitor serial, etc.

Dentro del monitor serial hay otra pestaña para establecer la tasa de baudios (*baudrate*), que marca el número de unidades de señal transmitidas por segundo. Este valor ha de estar sincronizado con el *baudrate* en el que esté trabajando el Arduino, el cual puede ser establecido en el código del *sketch* mediante el comando “`serial.begin`”, o de no ser así, se establecerá un valor por defecto. Si el monitor serial y Arduino no están sincronizados con la misma tasa de baudios, la información que aparezca en la ventana será completamente ilegible.

Dentro de los menús, cabe mencionar la existencia de librerías, que pueden proporcionar funcionalidades adicionales para la utilización en *sketches*, por ejemplo para trabajar con hardware o manipular datos. Se debe tener en cuenta que al cargar un código que incluya librerías, éstas también se vuelcan en la placa junto con el resto del *sketch*, incrementando la ocupación del programa y reduciendo el espacio disponible en la memoria de Arduino [52].

3.2.2.3. Aqevent.

Aqevent es un programa que está desarrollado en el lenguaje de programación C y permite enviar directamente la salida del puerto serial a un archivo de texto de extensión “.txt”, se ejecuta desde la terminal de Linux como se observa en la figura 3.6 siguiendo los siguientes comandos:

1. `cd/home/Escritorio/Aqevent` (ubicación de la carpeta).
2. `ls` (para comprobar que el archivo se encuentra en la carpeta).
3. `make` (para compilar).
4. `./aqevent` (para ejecutar el programa y por defecto creará un archivo .txt cada 5 s).
5. `./aqevent -t 10` (para ejecutar el programa y creará un archivo .txt cada 10 s, por ejemplo).

```

Terminal
porthos@misael ~ $ cd /home/porthos/Escritorio/Stephanie/Aqevent/
porthos@misael ~/Escritorio/Stephanie/Aqevent $ ls
aqevent  Base_De_Datos  filename.h  kbhit.h  Toma_Data
aqevent.c  Carpeta sin título  filename.o  kbhit.o
aqevent.o  filename.c      kbhit.c    Makefile
porthos@misael ~/Escritorio/Stephanie/Aqevent $ make
gcc -Wall -c -o aqevent.o aqevent.c
gcc -o aqevent -Wall aqevent.o kbhit.o filename.o
porthos@misael ~/Escritorio/Stephanie/Aqevent $ ./aqevent
difsec = 5      modal = q
2019_05_07_12:52:24_q.txt
2019_05_07_12:52:30_q.txt
2019_05_07_12:52:36_q.txt
2019_05_07_12:52:42_q.txt
2019_05_07_12:52:48_q.txt
2019_05_07_12:52:54_q.txt
2019_05_07_12:53:00_q.txt
2019_05_07_12:53:06_q.txt
2019_05_07_12:53:12_q.txt
2019_05_07_12:53:18_q.txt
2019_05_07_12:53:24_q.txt
2019_05_07_12:53:30_q.txt
2019_05_07_12:53:36_q.txt
2019_05_07_12:53:42_q.txt

```

Figura 3.6: Ejemplo de ejecución de Aqevent.

3.2.2.4. Características del computador.

- **Sistema Operativo:** Linux Mint 17.1 Cinnamon 64-bit.
- **Versión de Cinnamon:** 2.4.8
- **Procesador:** Intel® Core™ i5-2400 CPU @ 3.10 GHz x 4.
- **Memoria RAM:** 3.3 GB.
- **Capacidad del disco duro:** 1 TB.
- **Tarjeta gráfica:** Intel Corporation 2nd Generation Core Processor Family Integrated Graphics Controller.

3.2.3. Montaje Experimental.

Se utilizó un montaje como el de la figura 3.7, compuesto por un vidrio cuyas medidas son 75.0 x 65.0 cm, el cual tiene dibujado un patrón circular de 44.0 cm de diámetro, en conjunto con una línea horizontal y una vertical, que pasan por el centro del mismo, como se muestra en la figura 3.8; una placa Arduino Due a la que se le conecta la tarjeta sensor GY-80 que está protegida por una caja de acrílico, el pulsómetro y una corneta que es controlada por un potenciómetro, dispuestos como se muestra en el circuito de la figura 3.9 y a su vez esta placa se conecta a la computadora mediante un cable micro USB.

Este vidrio es ubicado sobre una mesa plana, a la cual se le ha fijado un imán circular (de 5 cm de diámetro) concéntrico con el círculo a una distancia de 20 cm por debajo, con el propósito de crear un punto de asimetría para el estudio del magnetómetro. Además, es necesario contar con una silla cómoda, cuya altura le permita a la persona que realizará la prueba, un buen apoyo de sus brazos sobre la mesa para así evitar cansancio que puede traer una alteración de los resultados.

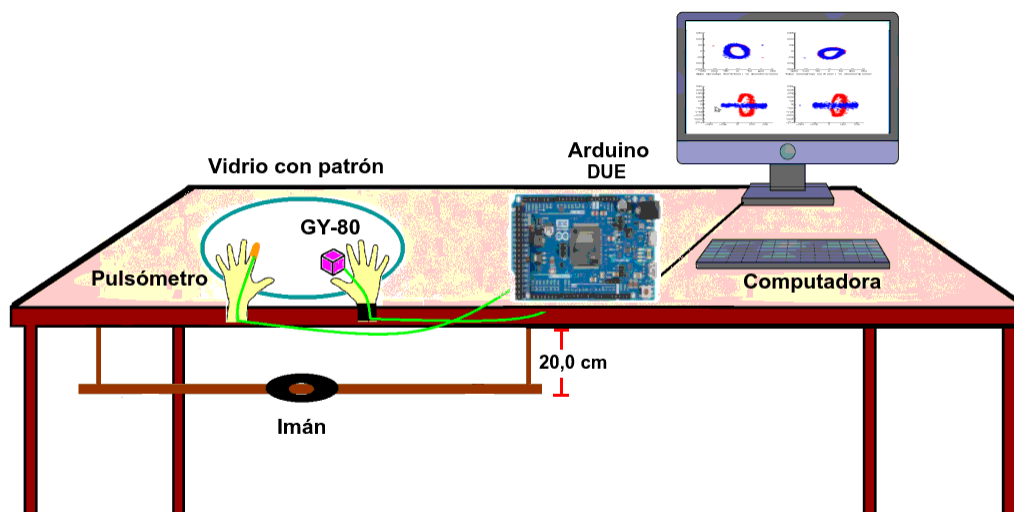


Figura 3.7: Montaje experimental donde se observa la mesa con el imán, el vidrio con patrón, los sensores, tarjeta GY-80 y computadora.

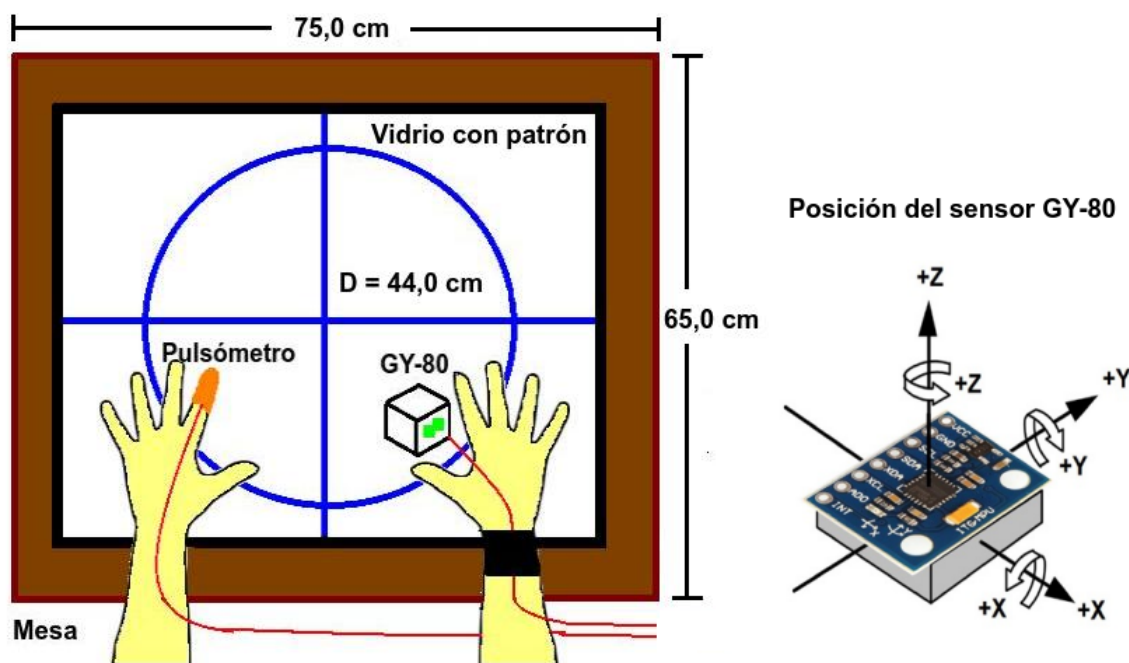


Figura 3.8: Vista desde arriba del montaje experimental donde se muestra el vidrio con el patrón con sus respectivas medidas y la posición empleada para el sensor GY-80.

Componentes:

- Vidrio con los patrones.
- Mesa plana.
- Imán circular.
- Computadora con MATLAB, Aqevent y el IDE de Arduino instalados.
- Placa Arduino Due.
- Tarjeta GY-80.
- Pulsómetro.
- Corneta (sonido del pulsómetro).
- Potenciómetro (para aumentar o disminuir el volumen de la corneta).
- Placa de prueba o Protoboard.
- Cables puente para prototipos.
- Cable micro USB.

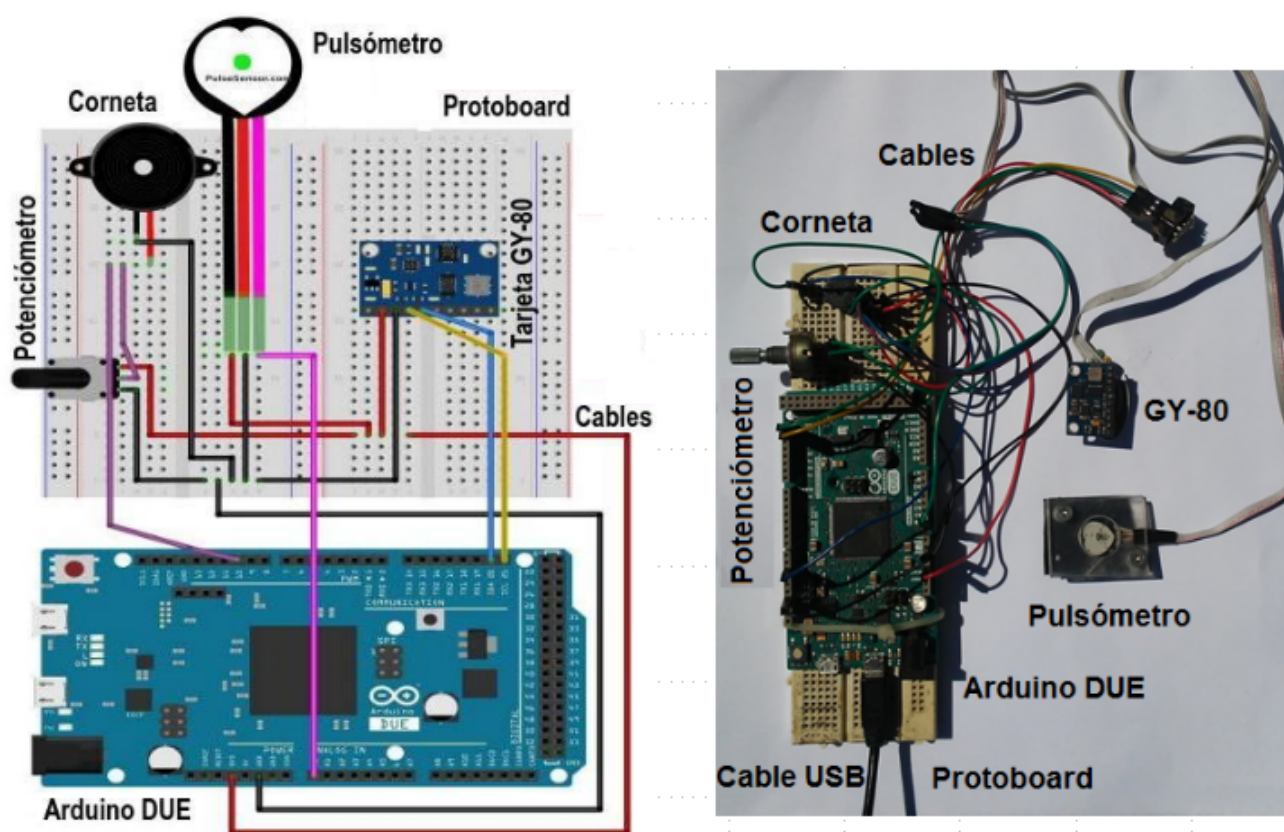


Figura 3.9: Circuito empleado: Tarjeta Arduino DUE, sensor GY-80 y pulsómetro conectados a un protoboard. Izquierda: Diagrama. Derecha: Fotografía del circuito.

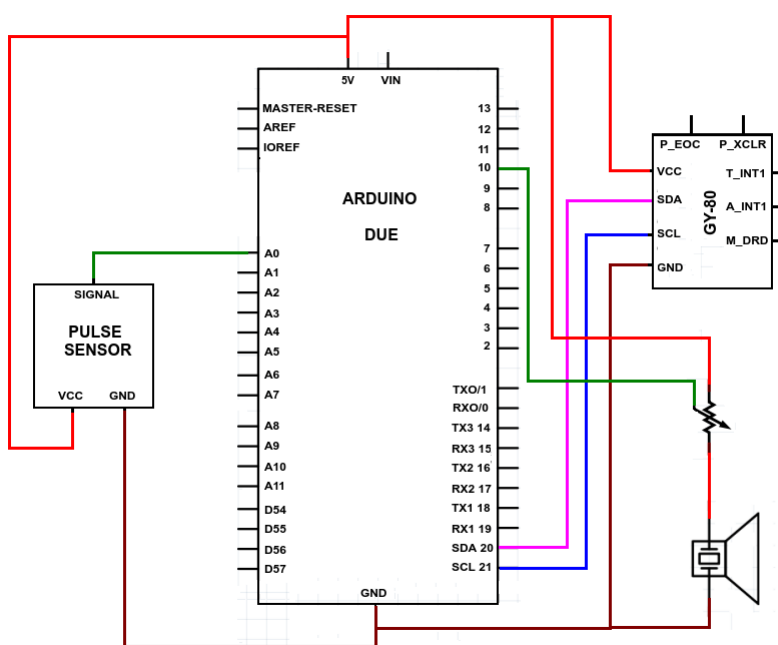


Figura 3.10: Diagrama del circuito empleado: Tarjeta Arduino DUE, sensor GY-80, pulsómetro, corneta y su potenciómetro.

En la figura 3.11 se muestra un sujeto prueba realizando una tarea. El sujeto toma en una de sus manos el sensor GY-80, el cual registrará la dinámica de su movimiento durante la tarea, mientras que en la otra mano, en el dedo índice, se sitúa el pulsómetro que registra el pulso cardíaco. Estos datos son digitalizados por la tarjeta Arduino DUE para posteriormente ser procesados.



Figura 3.11: Toma de datos de un sujeto sano, se observa que el sistema no es invasivo y que el montaje experimental es sencillo.

3.2.4. Población de estudio.

La población de estudio está conformada por dos grupos:

- El primero constituido por 60 sujetos sanos, 24 mujeres y 36 hombres, con edades comprendidas entre los 24 y 74 años, este grupo a la vez será dividido en dos: un subgrupo corresponderá a los sujetos con los que se entrenará la red neuronal, para ello se seleccionaron 45 de los 60 sujetos, divididos en 5 grupos etarios y los 15 restantes serán utilizados para la validación del método.
- El segundo grupo corresponde a personas que hayan sido diagnosticadas con la Enfermedad de Parkinson y que además su enfermedad se encuentre en los estadios I y II de la escala de Hoehn y Yahr, para este estudio se contó con 6 pacientes, 2 mujeres y 4 hombres, con edades comprendidas entre los 56 y 63 años, este grupo corresponderá a la fase de producción de la red neuronal.

3.2.5. Criterios de inclusión y de exclusión.

Criterios de inclusión.

- Edad comprendida entre 20 a 80 años.
- Género: masculino y femenino.
- Paciente que cumpla con los criterios diagnósticos de probable enfermedad de Parkinson realizada por un neurólogo.
- Paciente cuya enfermedad se encuentre en los estadios I y II de la escala de Hoehn y Yahr.
- Paciente motivado a participar y que acepte las condiciones informadas.

Criterios de exclusión.

- Personas que presenten síntomas o signos atípicos.
- Paciente cuya enfermedad se encuentre en los estadios III, IV y V de la escala de Hoehn y Yahr.
- Paciente que no esté motivado a participar y que no acepte las condiciones informadas.

3.2.6. Recogida de los datos.

Los datos se recogieron desde febrero de 2019 hasta marzo de 2020. Previamente se les informó a los sujetos sanos qué se pretendía alcanzar con la investigación y además en qué consistía la toma de datos. Una parte del estudio se realizó en la Facultad de Ciencias de la Universidad de los Andes, mientras que la otra en un sitio céntrico de la ciudad de Mérida a modo de facilitar el acceso. Para los pacientes con Parkinson se contó con la colaboración de 2 estudiantes del quinto y sexto año de medicina, quienes tenían el vínculo con ellos y sirvieron de mediadores. A los pacientes con Parkinson se les explicó con más detalle en qué consistía la toma de datos, además se les aclaró que no iban a recibir algún beneficio para su enfermedad pero que podrían ayudar a futuros pacientes. Luego de aplicar los criterios de inclusión y exclusión, así como la disposición de los pacientes para ir hasta el sitio o recibirnos en sus casas, se tomaron los datos de 6 pacientes cuyas principales características se muestran en la tabla 3.2.

N	Edad	Sexo	Ocupación	Estadio	Síntomas	Medicamento
1	56 años	F	Docente de primaria	I	- Temblor en una mano - Bradicinesia (leve) - Cansancio - Escritura menos legible	Levodopa + carbidopa 25/250mg (si lo toma)
2	58 años	M	Obrero y chófer	I	- Temblor en una mano - Bradicinesia - Fatiga - Dolor en las articulaciones	Levodopa + carbidopa 25/250mg (no lo toma)
3	59 años	F	Ama de casa	I	- Temblor en las manos - Bradicinesia (leve) - Fatiga	Levodopa + carbidopa 25/250mg (si lo toma)
4	61 años	M	Vendedor	II	- Temblor en las manos - Inestabilidad postural - Somnolencia - Dolores musculares	Levodopa + benserazida 200/50mg (si lo toma)
5	62 años	M	Obrero	II	- Temblor en ambas manos - Cambios en la locomoción - Insomnio - Voz ronca	Levodopa + carbidopa 25/250mg (si lo toma)
6	63 años	M	Agricultor	II	- Temblor en una mano - Alteración de la marcha - Pérdida del tono de la voz - Desánimo	Levodopa + carbidopa 25/250mg (no lo toma)

Tabla 3.2: Características principales de los pacientes con Parkinson.

3.2.7. Mediciones básicas.

1. **Aceleración (a):** La aceleración es la variación de la velocidad por unidad de tiempo es decir, la razón de cambio en la velocidad respecto al tiempo:

$$a = \frac{dV}{dt}$$

Así mismo la segunda ley de Newton indica que en un cuerpo con masa constante, la aceleración del cuerpo es proporcional a la fuerza que actúa sobre él mismo:

$$a = \frac{F}{m}$$

Este segundo concepto es utilizado por los acelerómetros para medir la aceleración. Los acelerómetros internamente tienen un MEMS que de forma similar a un sistema masa resorte permite medir la aceleración.

2. **Velocidad angular (ω):** La velocidad angular es la tasa de cambio del desplazamiento angular por unidad de tiempo, es decir que tan rápido gira un cuerpo alrededor de su eje.

$$\omega = \frac{d\theta}{dt}$$

Los giroscopios utilizan un MEMS para medir la velocidad angular usando el efecto Coriolis. El efecto observa en un sistema de referencia en rotación cuándo un cuerpo se encuentra en movimiento respecto a dicho sistema de referencia. Consiste en una aceleración relativa del cuerpo en dicho sistema de rotación. El objeto en este caso sufre una aceleración desde el punto de vista del observador en rotación, como si existiera una fuerza irreal sobre el objeto que lo acelera [53].

3. **Campo magnético (b):** La Tierra genera un campo magnético que crea disturbios magnéticos medibles en la atmósfera. Un magnetómetro es un instrumento que mide este fenómeno en términos de densidad de flujo magnético. Cuando hay materiales magnéticos presentes, un magnetómetro detecta la cantidad de distorsión que estos materiales causan en el campo de la Tierra.

Un magnetómetro no solo dice como afectan el flujo magnético ciertos materiales magnéticos en particular, sino que también pueden medir la fuerza de los campos magnéticos. Esta información puede utilizarse para discernir la dirección, la rotación y el ángulo de los campos magnéticos, así como la ubicación de objetos específicos dentro de ellos [54].

En un ambiente sin interferencias, cada terna de mediciones del magnetómetro generan una nube de puntos que se ubican en una esfera centrada en el origen de un sistema de referencia tridimensional con radio igual a la magnitud de la intensidad del campo magnético que se está midiendo (Figura 3.12).

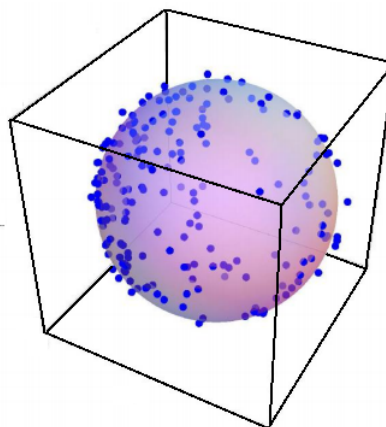


Figura 3.12: Ejemplo de un conjunto de mediciones de un magnetómetro sin interferencias.

Cuando el campo magnético se ve alterado por la presencia de otros campos magnéticos u objetos ferromagnéticos, las mediciones de las componentes del mismo se ven alteradas

haciendo que las mismas dejen de estar sobre la superficies de la esfera ideal. Para esta investigación nos interesa las interferencias tipo *Hard-Iron* [55].

Se denominan así a las interferencias producidas por elementos ferromagnéticos que se encuentran magnetizados de forma permanente como imanes. Normalmente estos elementos se encuentran en posiciones fijas respecto al magnetómetro por lo que las interferencias del tipo *Hard-Iron* se manifiestan como un vector de campo constante que se suma al vector de campo magnético que se está midiendo.

4. **Frecuencia cardiaca:** Es el número de contracciones del corazón por minuto. En un sujeto sano la frecuencia cardíaca mínima es de 0.8 Hz o 50 latidos/min y la máxima es de 1.7 Hz o 100 latidos/min.

Si bien es cierto que una frecuencia cardiaca normal no garantiza que una persona esté libre de problemas de salud, es un punto de referencia útil para identificar una variedad de enfermedades [56].

En la tabla 3.3 se especifican las unidades en que las diferentes variables medidas están expresadas:

Variable	Unidades
Acelerómetro	en unidades de g
Giróscopo	°/s
Magnetómetro	μT
Pulsómetro	pulsos/s

Tabla 3.3: Variables registradas con sus unidades.

3.2.8. Calibración.

Las medidas obtenidas por cualquier tipo de sensor pueden tener errores, éstos pueden ser constantes o variables y dependiendo del tipo pueden generar problemas en la utilización de los resultados obtenidos. Todo sensor necesita ser calibrado para reducir los errores, para calibrar un sensor lo que se hace es tomar unas medidas en una situación determinada en la que se conoce qué resultado se debería obtener. A partir de la diferencia entre lo que se ha medido y lo que se debería haber medido se modifica posteriormente matemáticamente las mediciones para que se acerquen más a su valor real.

Los errores constantes se pueden deber a un *offset*, este error supone que cada medición está desviada del valor real un cierto valor constante y la solución es calcular este valor constante y restarlo en todas las mediciones futuras. Los errores constantes también pueden ser debidos a un error en la ganancia, lo cual se solucionaría multiplicando las mediciones futuras por una ganancia que se debe calcular. Estos 2 tipos de errores son los que se tratarán de reducir mediante las pruebas de calibración.

Los errores variables son debidos al ruido propio de los sensores, este ruido hace que la medición se desvíe indistintamente hacia arriba o hacia abajo en un determinado rango. Este tipo de errores

se suelen eliminar mediante un filtro, que reduce las desviaciones.

El acelerómetro utilizado (ADXL345) tiene una buena calibración de fábrica, pero aun así será necesario realizar una para modificar la ganancia en cada eje. Para cuantificar el *offset* inicial se pone el sensor en el centro de la plataforma y se debe asegurar que la calcomanía que está sobre la caja protectora de acrílico, quede de frente al paciente de modo que los ejes del acelerómetro estén perpendiculares a la superficie terrestre, por lo tanto en ese eje, el z, la aceleración debe de ser 1g.

Cuando se ejecuta el programa “**Todos los sensores.ino**” el cual se encuentra en el apéndice B.1, la biblioteca < GY80.h > contiene instrucciones para realizar una calibración tanto para el magnetómetro como para el giróscopo, suficientes para el propósito de esta investigación.

3.2.9. Adquisición de los datos.

Después de haber calibrado el sensor, se carga en el entorno de desarrollo de Arduino el programa “**Todos los sensores.ino**”. A través del software Aqevent se adquieren los datos durante 120 segundos por cada tarea. El paciente realiza 8 tareas siguiendo los patrones que están dibujados sobre el vidrio, las tareas son las siguientes: Movimiento circular en sentido horario y antihorario y movimiento lineal en sentido vertical y horizontal, con la mano derecha e izquierda.

Para la toma de datos de una tarea para un paciente, se crean nueve archivos cada uno de 15 segundos, los cuales se guardan con el siguiente formato:

2020_02_03_16:41:45_q.txt
2020_02_03_16:42:01_q.txt
2020_02_03_16:42:17_q.txt
2020_02_03_16:42:33_q.txt

Los archivos son guardados en la carpeta **Aqevent** para ser concatenados desde MATLAB con el programa “**Concatena los archivos txt.m**” que se encuentra en el apéndice A.1.

3.3. II. Tratamiento de los datos.

3.3.1. Preparación de la data.

Una vez adquirida la data es necesario redimensionar a una misma longitud los datos de todos los pacientes, para ello se utiliza el programa “**Redimensiona los datos.m**”, que se encuentra en el apéndice A.3. Este programa hace uso de la función “**Combinación entre parámetros.m**”, apéndice A.4, que hace combinaciones entre las componentes de las variables de acuerdo a un índice óptimo como se muestra en la tabla 3.4.

Hay 15 combinaciones posibles, se puede escoger la combinación que se requiera y el resto de los parámetros serán vaciados. Seguidamente, se reescriben los datos de todos los pacientes a la misma longitud tomando en cuenta la mínima longitud.

Índice óptimo	Etiqueta	Combinación
1	1 1 1 0 0 0 0 0 0	$a_x a_y a_z$
2	0 0 0 1 1 1 0 0 0	$b_x b_y b_z$
3	0 0 0 0 0 0 1 1 1 0	$w_x w_y w_z$
4	0 0 0 0 0 0 0 0 0 1	p
5	1 1 1 1 1 1 0 0 0 0	$a_x a_y a_z b_x b_y b_z$
6	1 1 1 0 0 0 1 1 1 0	$a_x a_y a_z w_x w_y w_z$
7	1 1 1 1 1 1 0 0 0 1	$a_x a_y a_z b_x b_y b_z p$
8	1 1 1 0 0 0 1 1 1 1	$a_x a_y a_z w_x w_y w_z p$
9	0 0 0 1 1 1 1 1 1 0	$b_x b_y b_z w_x w_y w_z$
10	1 1 1 0 0 0 0 0 0 1	$a_x a_y a_z p$
11	0 0 0 1 1 1 1 1 1 1	$b_x b_y b_z w_x w_y w_z p$
12	0 0 0 1 1 1 0 0 0 1	$b_x b_y b_z p$
13	0 0 0 0 0 0 1 1 1 1	$w_x w_y w_z p$
14	1 1 1 1 1 1 1 1 1 0	$a_x a_y a_z b_x b_y b_z w_x w_y w_z$
15	1 1 1 1 1 1 1 1 1 1	$a_x a_y a_z b_x b_y b_z w_x w_y w_z p$

Tabla 3.4: Índices de las combinaciones posibles de los parámetros.

En la figura 3.13 se observa un ejemplo de la data sin depurar de una tarea para un paciente con todos los parámetros seleccionados (índice óptimo = 15).

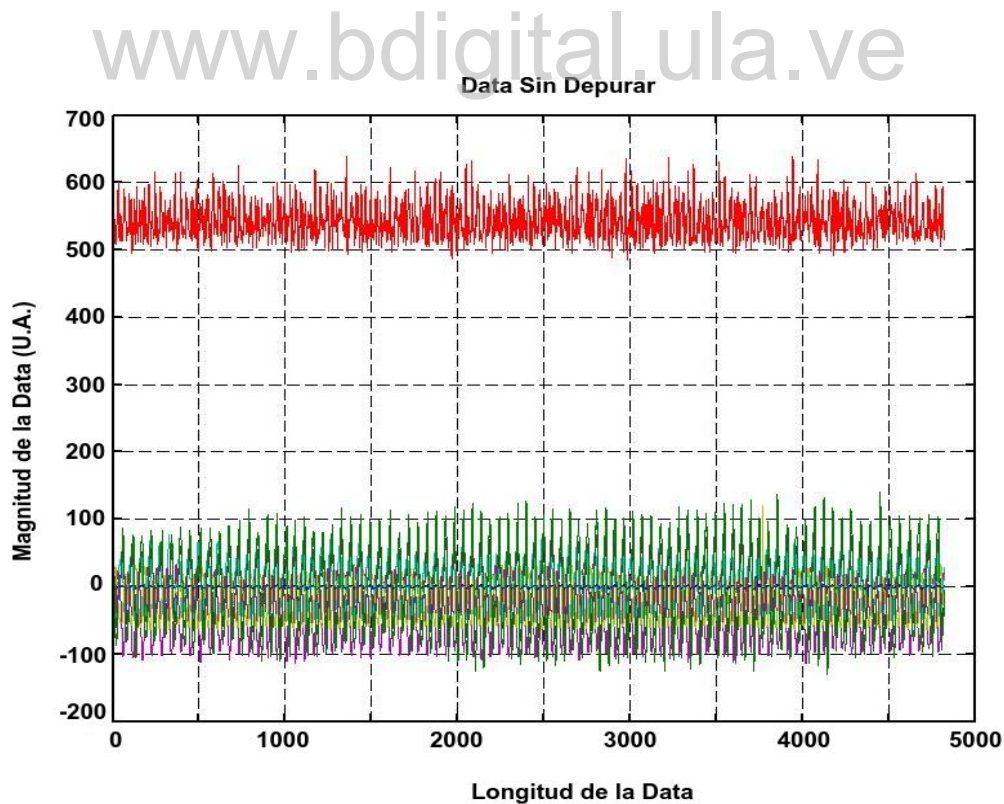


Figura 3.13: Ejemplo gráfico de la data sin depurar.

Formando una matriz fila, la data graficada queda como en la figura 3.14

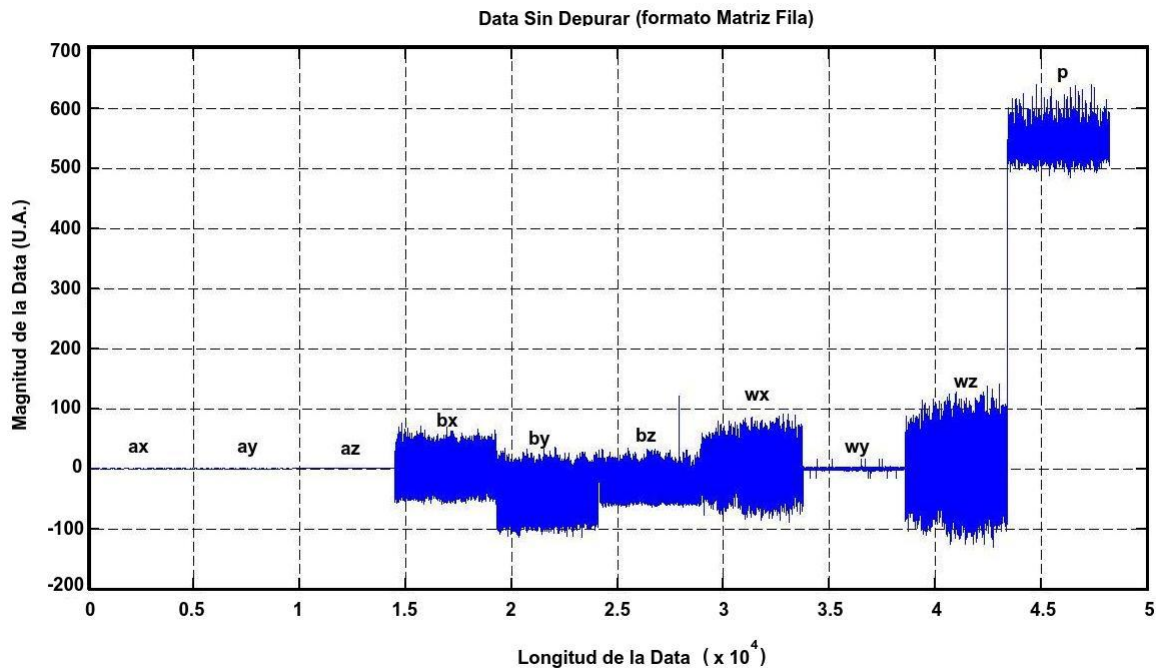


Figura 3.14: Ejemplo gráfico de la data sin depurar matriz fila.

3.3.2. Normalización y depuración de los datos.

Una de las herramientas comúnmente utilizada en el análisis de datos es la normalización. Ésta se hace por dos razones:

1. Se requiere cambiar la línea base o rango dinámico de todos los sensores a la desviación de cada sensor, esto regulariza las escalas dinámicas de los datos.
2. El centro de los datos debe ser situado a un centro común, lo que se logra eliminando el *offset* de cada instrumento, esto es quitando su línea base.

Existen varios tipos de normalización y dependerá de los datos y del problema a solucionar. En esta investigación se hará uso de la siguiente transformación lineal:

$$X = \frac{x - \bar{x}}{\sigma} \quad (3.1)$$

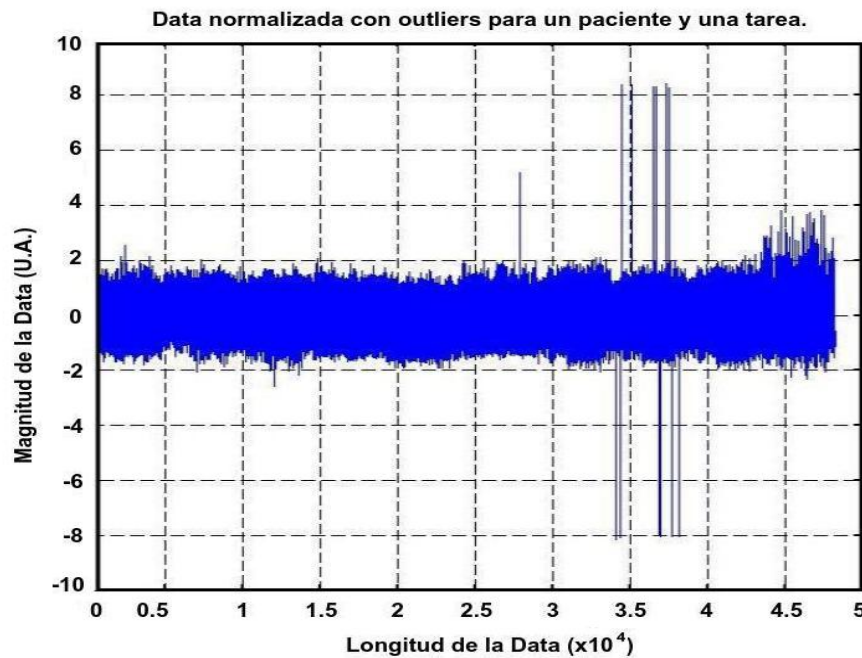
Siendo X el nuevo vector de datos normalizados, x la variable de entrada, $\bar{x}$ su valor medio y σ la desviación estándar. Este tipo de normalización tendrá valor medio cero y desviación estándar uno, con lo que se evitarán tendencias de las entradas fuera de los rangos considerados como normales, debido a los desplazamientos de la línea base de los instrumentos de medición. Por otra parte, al desplazar los datos a media cero y dividirlos por su desviación, los rangos dinámicos son los más apropiados en el entrenamiento de las RNA [57].

Para una tarea y un paciente la data estandarizada se ve como en el ejemplo de la figura 4.4.

ax	ay	az	bx	by	bz	wx	wy	wz	p
0.2193	-0.8414	-0.1933	0.8380	-0.8269	-0.7131	-0.7751	-0.7293	-0.7410	-0.3383
0.0358	-0.8414	-0.0117	0.8380	-0.8269	-0.7131	-0.3973	-0.3467	-0.3795	-0.2731
-0.5951	-0.6594	0.6542	0.4432	-1.0540	-0.8693	-0.2133	-0.2460	-0.2178	-0.0286
-0.9851	-0.6594	1.1385	0.4432	-1.0540	-0.8693	-0.2326	-0.1453	-0.1618	0.4931
-1.1457	-0.3560	1.1385	0.3079	-1.1367	-0.8894	-0.0680	-0.0647	-0.0840	1.0474
-1.3407	-0.5380	1.4412	0.3079	-1.1367	-0.8894	-0.0873	-0.1453	-0.8468	1.4549
-1.6389	0.4328	1.6228	0.2909	-0.9509	-0.8945	-2.2183	-2.3001	-1.9233	1.4060
-1.5357	0.9182	1.5017	0.2909	-0.9509	-0.8945	-0.7363	-0.6890	-0.7350	0.9822
-1.6045	1.4035	1.4412	0.2909	-0.9509	-0.8945	-0.2520	-0.2863	-0.2677	0.4116
-1.8339	2.7384	1.6833	0.3756	-0.7855	-0.8743	0.2420	0.2373	0.2236	-0.1427
-0.8016	3.8912	0.5331	0.3756	-0.7855	-0.8743	0.3098	0.2977	0.3075	-0.5176
-0.5033	-0.3560	0.5331	0.6632	-0.6411	-0.7786	0.4551	0.4186	0.4672	-0.7948
-1.6389	0.0687	1.6833	0.6632	-0.6411	-0.7786	0.6391	0.6200	0.6070	-0.9252
-1.1457	0.1901	1.1385	0.6632	-0.6411	-0.7786	0.8135	0.7811	0.8027	-0.7295
-1.3866	0.9182	1.4412	1.2891	-0.2894	-0.3505	0.8425	0.8213	0.8507	-0.5665
-1.1113	0.6755	0.7753	1.2891	-0.2894	-0.3505	0.8619	0.7206	0.7987	-0.3383
-1.0769	0.6148	1.0779	1.8023	0.3508	1.0548	0.8716	1.0026	1.0244	-0.2242
-0.9966	0.6755	0.9569	1.8023	0.3508	1.0548	1.2397	1.2442	1.3479	-0.1753
-0.8933	0.1901	1.0779	0.1444	0.8056	2.3392	1.2978	1.3651	1.3939	-0.2731
-0.6754	0.4934	0.6542	0.1444	0.8056	2.3392	1.1234	1.0026	1.0763	-0.3383
-0.4575	0.1901	0.4120	0.1444	0.8056	2.3392	0.9975	0.9422	1.0044	-0.4687
-0.5951	0.6755	0.6542	-1.4458	0.3717	0.9490	1.0460	1.0026	1.0464	-0.4524
-0.5951	0.7361	0.5937	-1.4458	0.3717	0.9490	1.1138	1.1033	1.0843	-0.4035
-0.1936	0.4934	0.1699	-1.1920	-0.4134	-0.2196	0.4938	0.4991	0.5251	-0.4361
0.1849	0.7968	-0.3144	-1.1920	-0.4134	-0.2196	-0.1648	-0.1654	-0.1219	-0.3709
0.1734	1.5856	-0.3749	-1.1920	-0.4134	-0.2196	0.3098	0.2776	0.2955	-0.2894
0.4143	0.9788	-0.4355	-0.8932	-0.8269	-0.5318	0.4163	0.3783	0.4453	-0.2568
0.8387	0.7361	-0.7381	-0.8932	-0.8269	-0.5318	0.2129	0.1769	0.1777	-0.3220
1.6990	-0.1133	-1.7067	-0.8142	-0.8682	-0.5973	1.0556	1.5665	1.5337	-0.2568
1.6875	-0.5380	-1.7067	-0.8142	-0.8682	-0.5973	0.2710	0.3179	0.0578	-0.2242

Figura 3.15: Ejemplo de la data normalizada para una tarea y un paciente.

Para términos de reducción de la dimensionalidad es necesario tener la data estandarizada como se muestra en la figura 4.4. Sin embargo, al graficarla como se ve en la figura 3.16 esta data está llena de *outliers* debidos a errores de comunicación provocados por el hardware, por lo tanto es necesario realizar una depuración de los datos.

Figura 3.16: Ejemplo de la data normalizada para una tarea y un paciente con *outliers*.

De manera general se tiene una matriz G donde se encuentra la data normalizada de todas las tareas ($T_1 \cdots T_8$) por todos los pacientes ($P_1 \cdots P_k$), siendo k el número de pacientes. Esta matriz es un arreglo multidimensional como muestra la figura 3.17 y con dimensiones $G = (n \times m \times l \times k)$. Donde:

- n : longitud de la data.
- m : número de parámetros.
- l : tareas.
- k :pacientes.

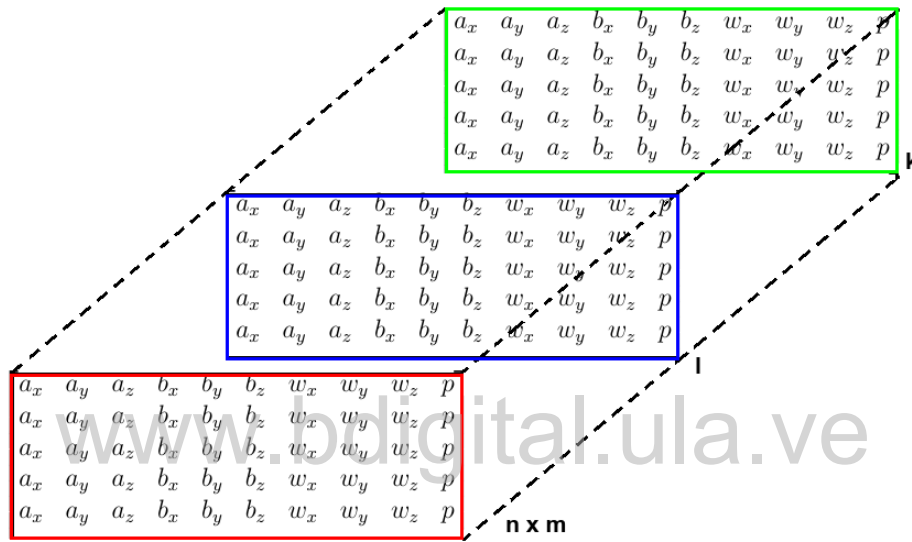


Figura 3.17: Arreglo multidimensional de la matriz G .

En dos dimensiones la matriz G se vería como en la figura 3.18, siendo $[a_x \ a_y \ a_z \ b_x \ b_y \ b_z \ w_x \ w_y \ w_z \ p]$ en cada caso una matriz de n filas:

$$G = \begin{bmatrix} [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \end{bmatrix} \begin{bmatrix} T_1 P_1 \\ T_2 P_1 \\ T_3 P_1 \\ \vdots \\ T_8 P_1 \\ T_1 P_k \\ T_2 P_k \\ T_3 P_k \\ \vdots \\ T_8 P_k \end{bmatrix}$$

Figura 3.18: Matriz G

Para la depuración de los datos se utiliza el programa “**Depuración de los datos.m**”, apéndice A.5. A partir de la matriz G, se desglosan los parámetros que se tomaron para el estudio, en este caso se tomó $\text{índice_óptimo} = 15$, que incluye todos los parámetros : $a_x, a_y, a_z, b_x, b_y, b_z, w_x, w_y, w_z, p$.

Se calculan los módulos de la aceleración, campo magnético, velocidad angular y pulso. Esos módulos se estandarizan aplicando la ecuación 3.1, para cada como se muestra en la tabla 3.5.

Módulos	Estandarización
$a = \sqrt{a_x^2 + a_y^2 + a_z^2}$	$a^* = (a - \bar{a})/\sigma_a$
$b = \sqrt{b_x^2 + b_y^2 + b_z^2}$	$b^* = (b - \bar{b})/\sigma_b$
$w = \sqrt{w_x^2 + w_y^2 + w_z^2}$	$w^* = (w - \bar{w})/\sigma_w$
$p = p$	$p^* = (p - \bar{p})/\sigma_p$

Tabla 3.5: Módulos y estandarización de cada una de la variables.

Con el valor de las variables estandarizadas se crea el vector $K = [a \quad b \quad w \quad p]$.

Se determina un umbral que depende de dos factores: El primero es σ_o , valor que es dado de entrada, para este caso los datos con una desviación estándar superior a $\pm 3\sigma_o$ fueron descartados y el segundo factor es la desviación estándar de cada variable:

$$a_o = \sigma_o * \sigma_a$$

$$b_o = \sigma_o * \sigma_b$$

$$w_o = \sigma_o * \sigma_w$$

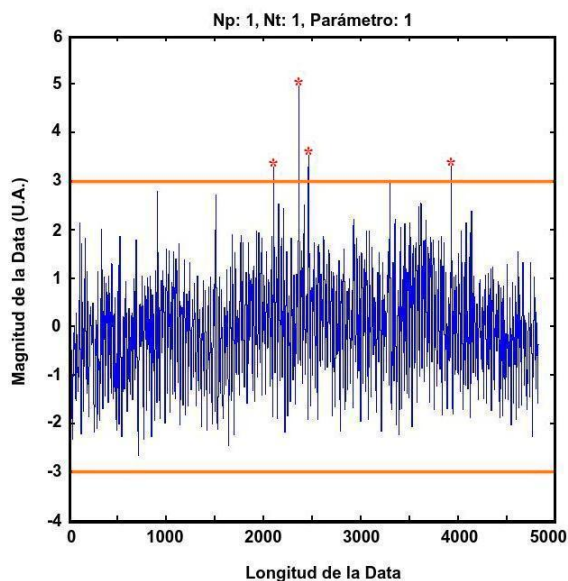
$$p_o = \sigma_o * \sigma_p$$

Se obtiene un vector K_o con el valor de la desviación total para cada variable: $K_o = [a_o \quad b_o \quad w_o \quad p_o]$.

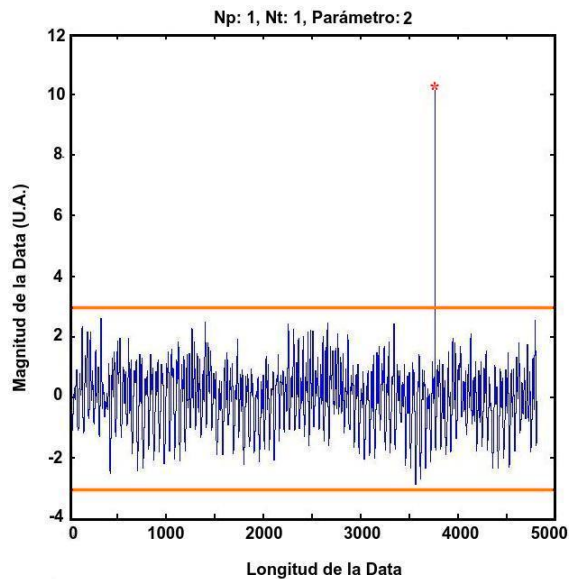
Ahora se buscan los índices donde $k \geq k_o$, como se muestra en la figura 3.19, esto se hace por cada parámetro, por tarea y por paciente y esos valores son descartados del vector original de datos.

La variable DATA ya depurada se reescribe y se crea un nuevo archivo que va escrito como “**Data_Depurada_Paciente_(i)_Tarea_(j)**”, donde i corresponde al paciente y j a la tarea respectiva. Seguidamente se llevan a una misma longitud, tomando en cuenta la longitud mínima. Se estandariza nuevamente y se obtiene un arreglo multidimensional H normalizado por paciente y por tarea.

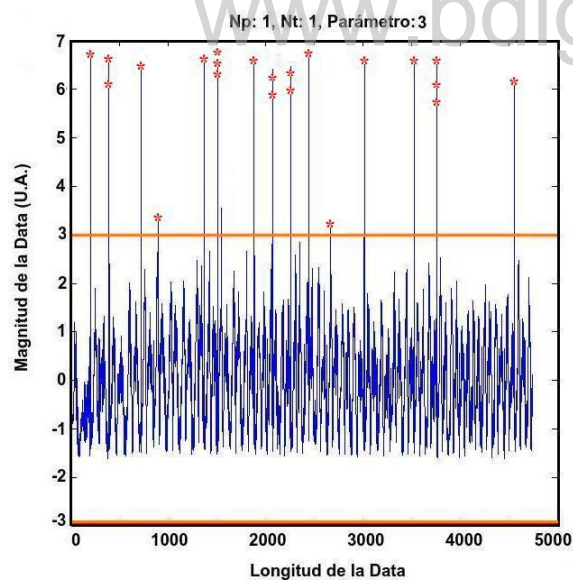
En la figura 3.20 se observa la superposición de la data no depurada (azul) y la data depurada (rojo), una data limpia sin *outliers*.



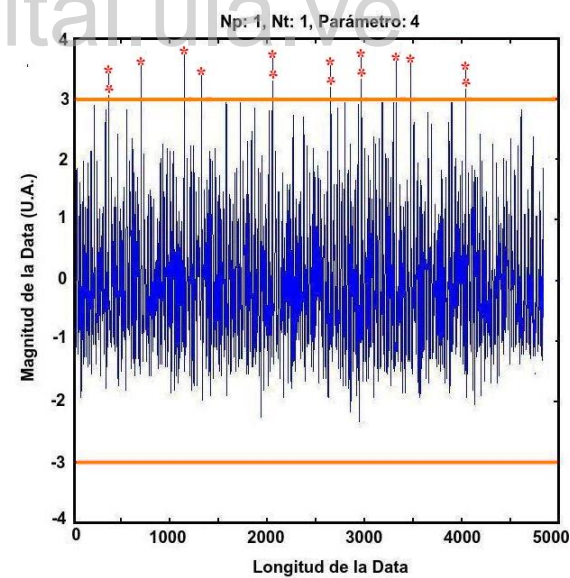
(a) Depuración parámetro 1.



(b) Depuración parámetro 2.



(c) Depuración parámetro 3.



(d) Depuración parámetro 4.

Figura 3.19: Depuración de la data por parámetro.

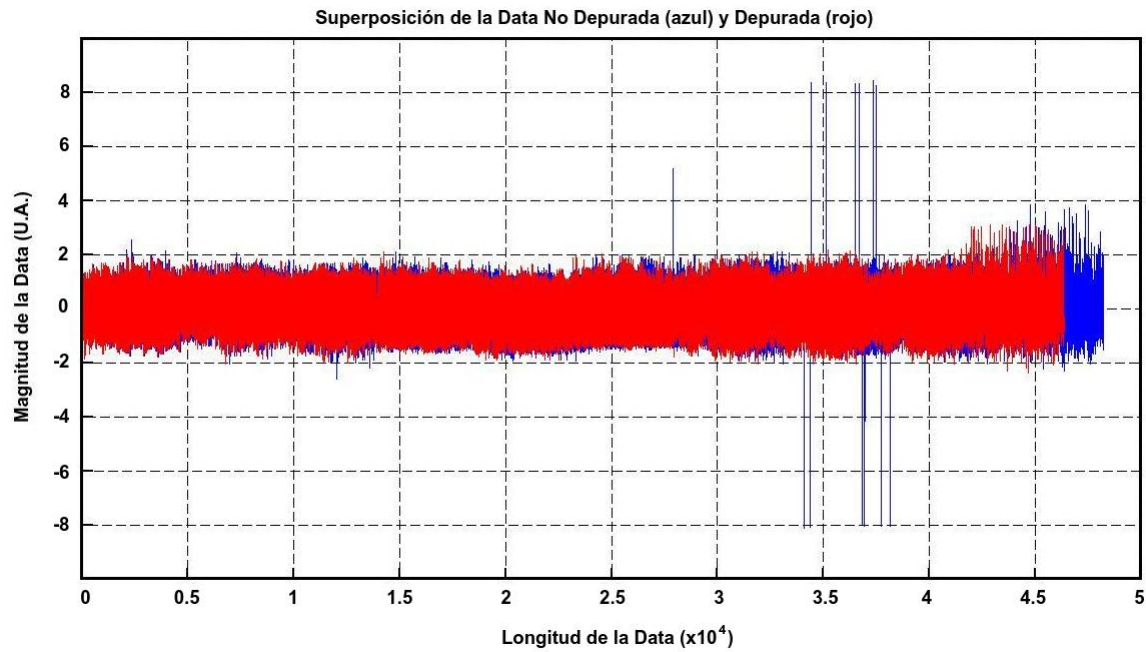


Figura 3.20: Superposición de la data no depurada y depurada.

3.3.3. Creación de ejemplos.

El arreglo H se divide en ventanas (N_v) del mismo intervalo de tiempo (D_t), se construye un arreglo D_s que es múltiplo del número de ventanas, este se redimensiona utilizando la función *reshape* y se obtiene una nueva matriz denominada D_r que está conformada por N_v ejemplos de la data por cada tarea y por paciente. En la figura 3.21 se muestra el gráfico para el ejemplo de una tarea y de un paciente.

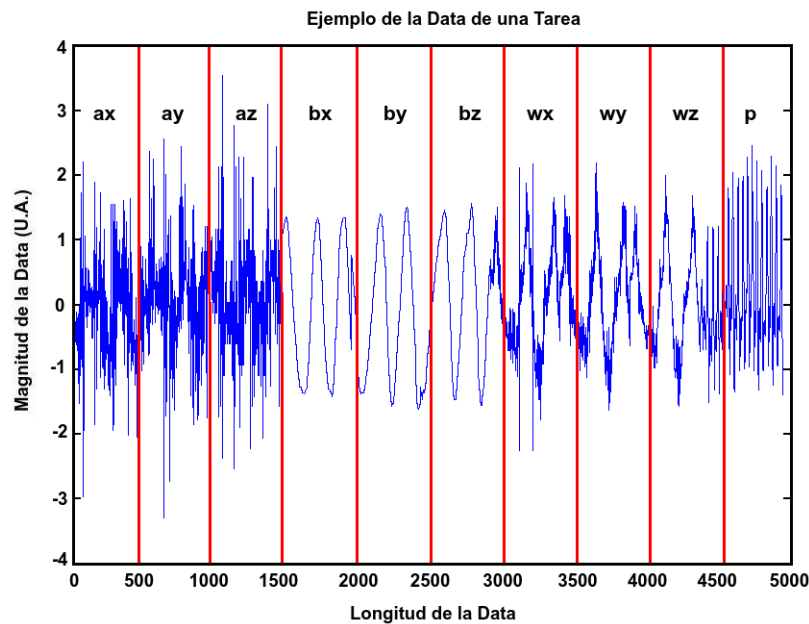


Figura 3.21: Ejemplo de los parámetros para una tarea.

En dos dimensiones la matriz Dr se ve como en la figura 3.22. Una colección de ejemplos por tareas y por paciente.

$$Dr = \begin{bmatrix} [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p] \end{bmatrix} \begin{matrix} \text{Ejemplo}_1 T_1 P_1 \\ \text{Ejemplo}_2 T_1 P_1 \\ \text{Ejemplo}_3 T_1 P_1 \\ \vdots \\ \text{Ejemplo}_n T_1 P_1 \\ \vdots \\ \text{Ejemplo}_1 T_8 P_1 \\ \text{Ejemplo}_2 T_8 P_1 \\ \text{Ejemplo}_3 T_8 P_1 \\ \vdots \\ \text{Ejemplo}_n T_8 P_1 \end{matrix}$$

Figura 3.22: Matriz Dr, conformada por varios ejemplos de la data.

3.3.4. Formato de los datos.

Empleando el programa “**Formato de los datos.m**”, apéndice A.6, se obtiene la matriz A a partir de la matriz Dr como se ve en la figura 3.23.

$$Dr = \begin{bmatrix} a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \end{bmatrix}$$

$$A = \begin{bmatrix} a_x & a_y & a_z & b_x & b_y & b_z & w_x & w_y & w_z & p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x] & [a_y] & [a_z] & [b_x] & [b_y] & [b_z] & [w_x] & [w_y] & [w_z] & [p] \\ [a_x] & [a_y] & [a_z] & [b_x] & [b_y] & [b_z] & [w_x] & [w_y] & [w_z] & [p] \\ [a_x] & [a_y] & [a_z] & [b_x] & [b_y] & [b_z] & [w_x] & [w_y] & [w_z] & [p] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [a_x] & [a_y] & [a_z] & [b_x] & [b_y] & [b_z] & [w_x] & [w_y] & [w_z] & [p] \end{bmatrix}$$

Figura 3.23: Muestra del paso de la matriz Dr a la matriz A.

La matriz A contiene todos los ejemplos de todas las tareas por todos los pacientes, representado en la figura 3.24, es un formato adecuado de tener los datos para su posterior análisis, ya que $A = (n \times m)$ es bidimensional, donde:

- n: ejemplos * tareas * pacientes.
- m: longitud de los datos.

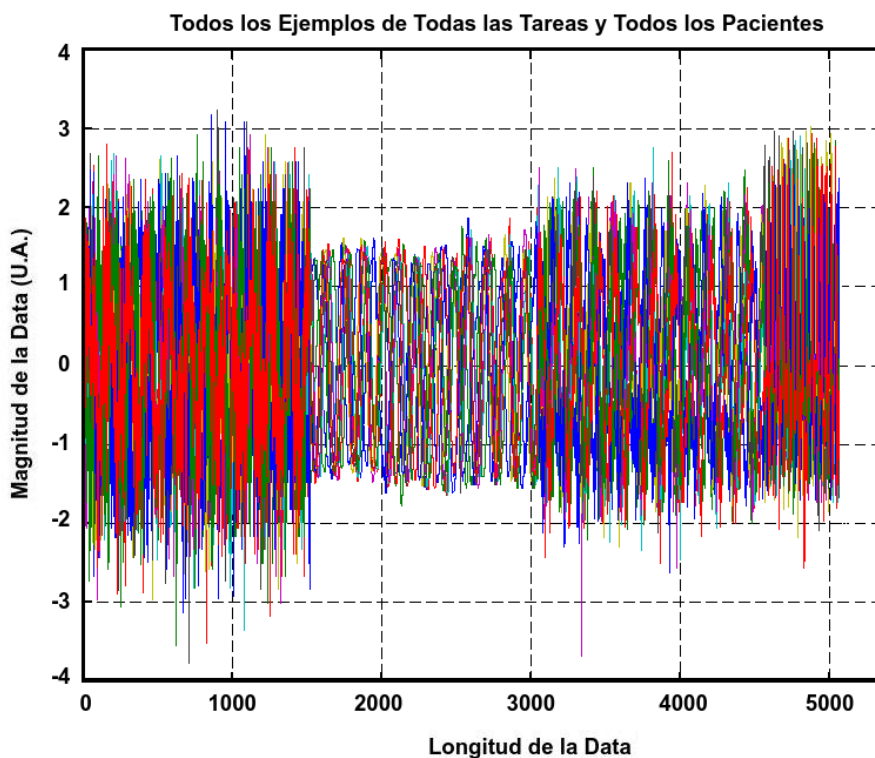


Figura 3.24: Vista de la matriz A: Todos los ejemplos de todas las tareas y pacientes.

A esta matriz se le hace un filtrado suave utilizando la función *smooth* con un factor de depuración $h_o = 3$. La matriz A se divide en data de entrenamiento (A_e) y data de test (A_t), de acuerdo con un porcentaje de entrenamiento dado (pe), en este caso se utilizó $pe = 70\%$.

3.3.5. Reducción de la dimensionalidad.

La reducción de la dimensionalidad es el proceso de encontrar un espacio dimensionalmente inferior adecuado para representar los datos originales, con la esperanza de que la representación alternativa de los datos permita:

- Explorar datos de alta dimensión con el objetivo de descubrir una estructura o patrones que conducen a la formación de hipótesis estadísticas.
- Visualización de los datos utilizando diagramas de dispersión cuando la dimensionalidad es reducida a 2D o 3D.
- Analizar los datos utilizando métodos estadísticos, como la agrupación, suavizado, estimación de densidad de probabilidad, o clasificación.

Un posible método para la reducción de la dimensionalidad sería simplemente seleccionar subconjuntos de las variables para su procesamiento y análisis en grupos. Sin embargo, en algunos casos, eso significaría perder mucha información útil. Una alternativa sería crear nuevas variables que sean funciones (por ejemplo, combinaciones lineales) de las variables originales. Un método que permite hacer esto es el conocido como análisis de componentes principales (PCA) [58].

3.4. III. Aplicación de PCA.

3.4.1. Etapa de entrenamiento.

Una vez culminada la fase del tratamiento de la data, la cual se hace a su vez dentro del programa “**Análisis Componentes Principales.m**”, apéndice A.2, se estandariza la data de entrenamiento (Ae) aplicando la ecuación 3.1, obteniendo una nueva matriz G, a partir de la cual se calcula la matriz de covarianza (figura 3.25) y la diagonal de la misma, usando las siguientes funciones de MATLAB:

- $C = \text{cov}(G) \Rightarrow$ matriz de covarianza.
- $S = \text{diag}(C) \Rightarrow$ diagonal de la matriz de covarianza.

1.0000	0.8998	0.7778	0.7044	0.6696	0.5976	0.5098	0.4680	0.4277	0.4009
0.8998	1.0000	0.9589	0.8635	0.7708	0.6790	0.5954	0.5642	0.5266	0.5006
0.7778	0.9589	1.0000	0.9448	0.8459	0.7390	0.6587	0.6325	0.5983	0.5710
0.7044	0.8635	0.9448	1.0000	0.9479	0.8544	0.7555	0.7138	0.6807	0.6349
0.6696	0.7708	0.8459	0.9479	1.0000	0.9512	0.8605	0.7769	0.7444	0.6971
0.5976	0.6790	0.7390	0.8544	0.9512	1.0000	0.9394	0.8614	0.7965	0.7483
0.5098	0.5954	0.6587	0.7555	0.8605	0.9394	1.0000	0.9505	0.8837	0.7757
0.4680	0.5642	0.6325	0.7138	0.7769	0.8614	0.9505	1.0000	0.9566	0.8586
0.4277	0.5266	0.5983	0.6807	0.7444	0.7965	0.8837	0.9566	1.0000	0.9414
0.4009	0.5006	0.5710	0.6349	0.6971	0.7483	0.7757	0.8586	0.9414	1.0000

Figura 3.25: Ejemplo de la matriz de covarianza.

El siguiente paso consiste en la obtención de los autovalores y autovectores de la matriz de covarianza, utilizando la función *eigs*. Con el número de autovalores (Na) ya fijado.

La función $[V, D] = \text{eigs}(C, Na, 'lm')$, devuelve la matriz diagonal D que son los autovalores (figura 3.26) y la matriz V cuyas columnas son los autovectores (figura 3.27).

283.3078	0	0	0	0	0	0	0	0	0
0	230.8371	0	0	0	0	0	0	0	0
0	0	213.2194	0	0	0	0	0	0	0
0	0	0	192.0595	0	0	0	0	0	0
0	0	0	0	162.8947	0	0	0	0	0
0	0	0	0	0	157.4967	0	0	0	0
0	0	0	0	0	0	129.9116	0	0	0
0	0	0	0	0	0	0	123.9571	0	0
0	0	0	0	0	0	0	0	118.8502	0
0	0	0	0	0	0	0	0	0	108.6264

Figura 3.26: Matriz de autovalores.

-0.0039	-0.0044	0.0058	0.0160	-0.0181	-0.0174	0.0161	-0.0035	-0.0095	-0.0149
-0.0051	-0.0070	0.0065	0.0165	-0.0272	-0.0135	0.0221	-0.0090	-0.0099	-0.0165
-0.0070	-0.0089	0.0057	0.0162	-0.0297	-0.0126	0.0247	-0.0119	-0.0094	-0.0169
-0.0091	-0.0108	0.0046	0.0167	-0.0261	-0.0160	0.0263	-0.0137	-0.0092	-0.0159
-0.0116	-0.0093	0.0047	0.0163	-0.0253	-0.0166	0.0270	-0.0137	-0.0062	-0.0160
-0.0128	-0.0060	0.0049	0.0159	-0.0259	-0.0146	0.0293	-0.0141	-0.0033	-0.0145
-0.0130	-0.0066	0.0018	0.0157	-0.0278	-0.0112	0.0286	-0.0159	-0.0008	-0.0129
-0.0122	-0.0057	-0.0012	0.0174	-0.0273	-0.0111	0.0300	-0.0196	0.0016	-0.0149
-0.0122	-0.0073	-0.0053	0.0203	-0.0260	-0.0122	0.0308	-0.0217	0.0040	-0.0144
-0.0128	-0.0066	-0.0084	0.0232	-0.0247	-0.0103	0.0337	-0.0210	0.0068	-0.0170

Figura 3.27: Matriz de autovectores.

Se ordenan los autovalores en orden descendente utilizando la función `sort(S, 'descend')` y se construye el gráfico de sedimentación como se muestra en la figura 3.28.

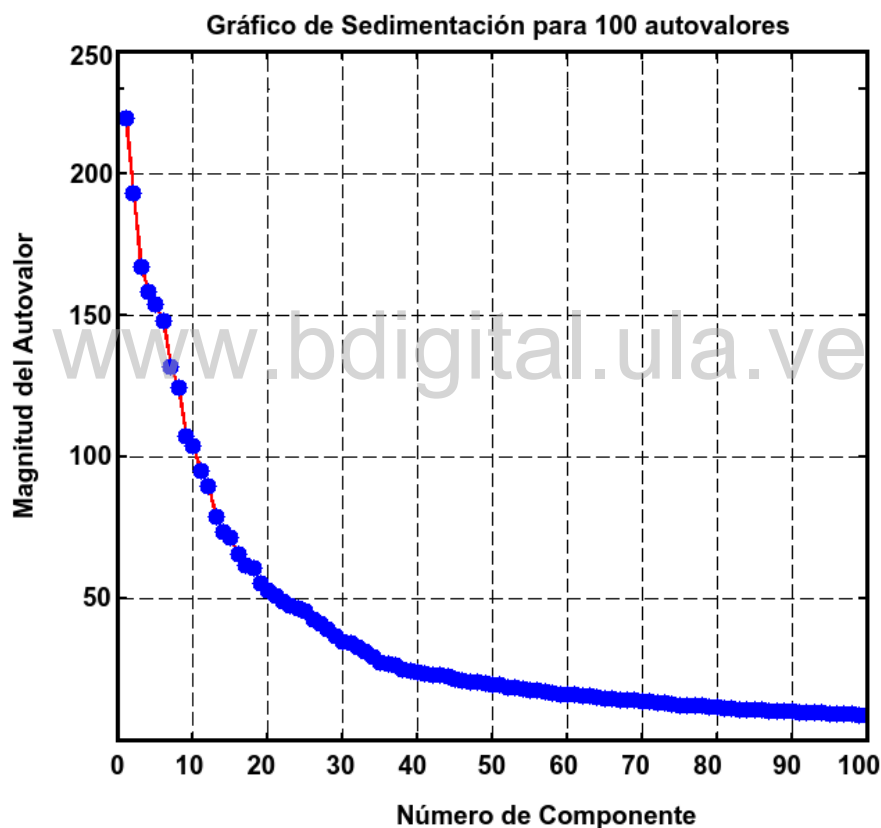


Figura 3.28: Gráfico de sedimentación para 100 autovalores.

V corresponde al nuevo espacio donde serán proyectados los pacientes promedios.

$$P = I * V$$

Donde:

I : pacientes promedios.

V: nuevo espacio (conjunto de autovectores)

La parte de entrenamiento termina con la creación de una llave única para cada paciente, la cual contiene la proyección de la media de los datos (imagen media) de cada paciente sobre el espacio V como se observa en la figura 3.29.

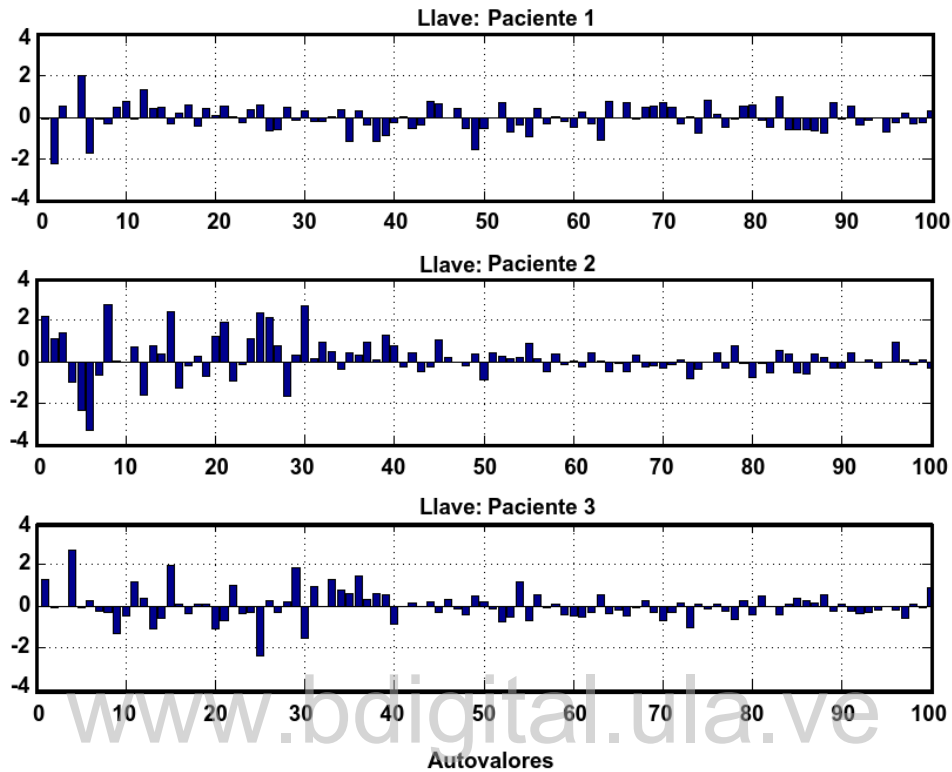


Figura 3.29: Proyección de la imagen media de cada paciente sobre el espacio V .

3.4.2. Etapa de reconocimiento.

Se toma la imagen media de los pacientes a reconocer, estos son etiquetados con la función “**Etiquetas.m**”, apéndice A.8, se proyecta sobre el espacio V de características y se determina la mínima distancia entre dos puntos: la nueva proyección y la proyección de los pacientes de entrenamiento.

Es importante recordar las siguientes definiciones sobre distancia, tipos y coeficiente de correlación:

Distancia: Se denomina distancia a la longitud del camino más corto entre dos entidades. Desde un punto de vista formal, para un conjunto de elementos X , se define distancia como cualquier función binaria $d(a, b)$ de $X \times X$ en $\mathbb{R}$ que verifique las siguientes condiciones [59]:

- No negatividad: $d(a, b) \geq 0 \forall a, b \in X$
- Simetricidad: $d(a, b) = d(b, a) \forall a, b \in X$
- Desigualdad triangular: $d(a, b) \leq d(a, c) + d(c, b) \forall a, b, c \in X$

Distancia Euclídea: Se denomina distancia euclídea entre dos puntos $A(x_1, y_1)$ y $B(x_2, y_2)$ a la longitud del segmento de recta que tiene por extremos A y B . Se expresa matemáticamente como:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

La distancia euclídea, a pesar de su sencillez de cálculo y de que verifica algunas propiedades interesantes tiene dos inconvenientes:

- Es una distancia sensible a las unidades de medida de las variables. Como consecuencia de ello, los cambios de escala determinarán, también cambios en la distancia entre los individuos.
- El segundo inconveniente se deriva de la naturaleza de las variables: Si las variables están correlacionadas, estas nos darán en gran medida, una información redundante. Como consecuencia la distancia euclídea inflará la disimilaridad o divergencia entre los individuos.

Distancia de Mahalanobis: Fue introducida por Mahalanobis en 1936. Se expresa matemáticamente como:

$$d = \sqrt{(x_2 - x_1)^2 \Sigma^{-1} (y_2 - y_1)^2}$$

donde Σ^{-1} es la matriz de covarianza. Esta distancia presenta las ventajosas propiedades de solventar los dos inconvenientes de la aplicación de la distancia euclídea. Por una parte es invariante ante los cambios de escala y no depende, por tanto de las unidades de medida.

La distancia de Mahalanobis proporciona un método muy poderoso para saber si un determinado conjunto de condiciones similares es en realidad un conjunto de condiciones ideales, está dada en términos de desviación estándar [59].

Coefficiente de correlación: La correlación lineal es un método estadístico que estudia la relación lineal existente entre dos variables. Para estudiar la relación lineal existente entre dos variables continuas es necesario disponer de parámetros que permitan cuantificar dicha relación. Uno de estos parámetros es la covarianza, que indica el grado de variación conjunta de dos variables aleatorias. Para poder hacer comparaciones se estandariza la covarianza, generando los coeficientes de correlación (r) [60].

Existen diferentes tipos, de entre los que destacan el coeficiente de Pearson, Rho de Spearman y Tau de Kendall. Todos ellos varían entre +1 y -1. Siendo +1 una correlación positiva perfecta y -1 una correlación negativa perfecta.

Como norma general en la tabla 3.6 se muestra las directrices sobre la fuerza de relación entre las variables [61].

La correlación es solamente apropiada para examinar la relación entre datos cuantificables significativos en vez de datos categóricos. En la figura 3.30 se muestra la correlación existente para un sujeto con respecto a todo el grupo y por lo tanto su clasificación con el sujeto más cercano, en este caso el más parecido. Este procedimiento se repite al menos 5 veces empleando cada vez una imagen media diferente para cada sujeto, se guardan los datos para la construcción de la matriz de confusión y la creación de la base de conocimiento.

Valor de r	Fuerza de relación
-1.0 a -0.5 o 0.5 a 1.0	Fuerte
-0.5 a -0.3 o 0.3 a 0.5	Moderada
-0.3 a -0.1 o 0.1 a 0.3	Débil
-0.1 a 0.1	Muy débil

Tabla 3.6: Directrices sobre la fuerza de relación.

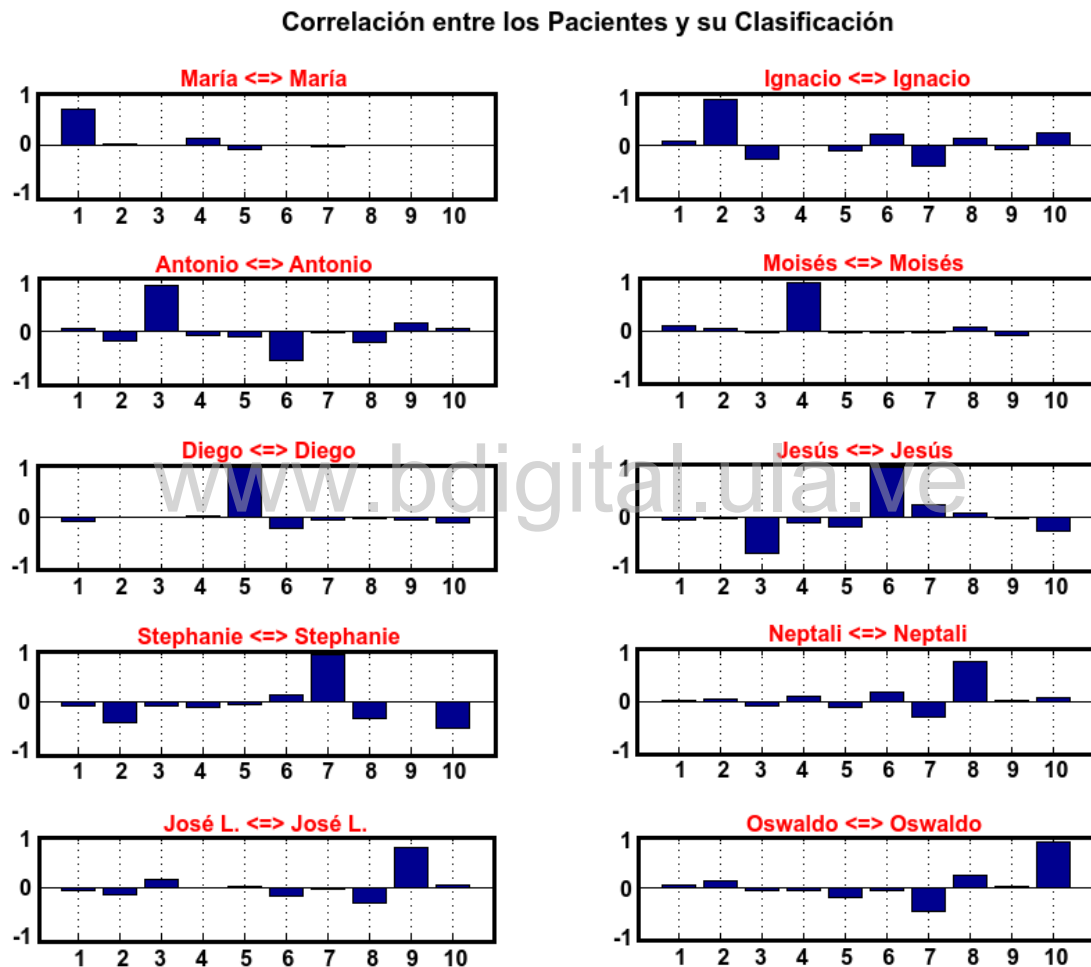


Figura 3.30: Correlación entre los pacientes y su respectiva clasificación, en el eje x se presentan los sujetos y en el eje y la correlación entre ellos.

3.4.2.1. Matriz de confusión PCA.

Es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado. La matriz de confusión de un problema de clase n es una matriz $n \times n$ en la que las filas se nombran según las clases reales (etiquetas dadas) y las columnas, según las clases previstas por el modelo (etiquetas encontradas). Sirve para mostrar de forma explícita cuándo una clase es

confundida con otra.

En la figura 3.31 se muestra la matriz de confusión obtenida durante el PCA para 10 sujetos, m representa la pendiente de la recta y se puede observar que una pendiente igual a 1 equivale a una clasificación con una exactitud del 100 %.

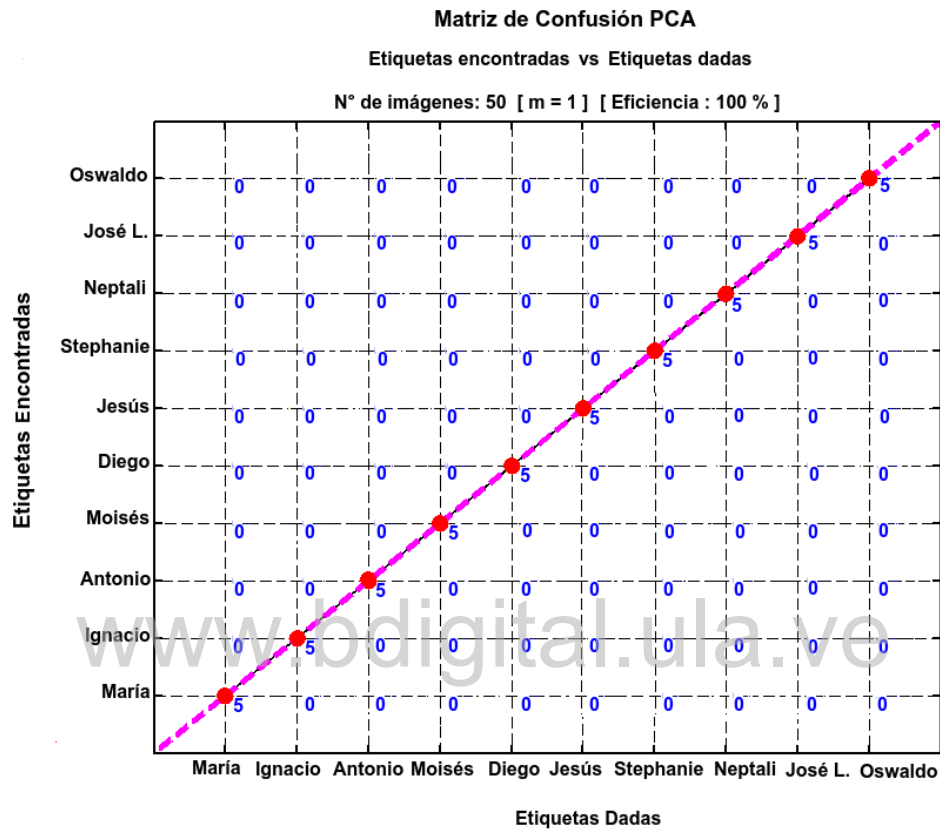


Figura 3.31: Matriz de confusión con PCA para 10 sujetos.

3.4.2.2. Base de datos y conocimiento.

- **Base de datos (DB):** Almacena grandes cantidades de datos organizados siguiendo un determinado esquema o modelo de datos que facilite su almacenamiento, recuperación y modificación.
- **Base de conocimiento (KB):** Es la evolución lógica de los sistemas de base de datos tradicionales, es un intento de plasmar no ya cantidades muy grandes de datos, sino elementos de conocimiento así como la manera en que éste ha de ser utilizado [62].

En la figura 3.32 se muestra un fragmento de la base de conocimiento con las etiquetas para cada sujeto, la cual será denominada Pcf. Esta se crea empleando la función “**Data de entrenamiento y test.m**”, apéndice A.7. Pcf será dividido de acuerdo al porcentaje de entrenamiento en dos partes: *Pcf_ent* que corresponde a los patrones de entrada para entrenar la red neuronal y *Pcf_test* que son los patrones usados durante el test.

Base de conocimiento											
0.1748	0.8910	-0.3287	-0.1897	0	0	0	0	0	0	0	1.0000
-0.2382	0.2471	0.2072	0.2600	0	0	0	0	0	0	1.0000	0
0.0819	0.5122	-0.3821	0.0252	0	0	0	0	0	0	1.0000	1.0000
0.2536	0.4141	0.4470	0.0153	0	0	0	0	0	1.0000	0	0
-0.5047	0.0325	-0.0959	0.2683	0	0	0	0	0	1.0000	0	1.0000
0.3909	0.4579	0.1060	-0.2936	0	0	0	0	0	1.0000	1.0000	0
-0.2716	0.0405	0.2833	0.1540	0	0	0	0	0	1.0000	1.0000	1.0000
0.1073	-0.0143	0.1316	0.0081	0	0	0	0	1.0000	0	0	0
0.2525	-0.0362	0.1093	0.1412	0	0	0	0	1.0000	0	0	1.0000
0.0957	-0.4656	0.0800	-0.1968	0	0	0	0	1.0000	0	1.0000	0
DATA				Etiquetas							

Figura 3.32: Base de conocimiento con sus etiquetas (Pcf).

3.5. IV. Entrenamiento de la Red Neuronal.

Ya obtenida la base de conocimiento se continúa con el entrenamiento de la red neuronal, para ello se emplea el programa “**Red Neuronal Backpropagation.m**”, apéndice A.9. En este apartado se siguen los siguientes pasos:

■ Entrenamiento

Dentro del programa “**Red Neuronal Perceptrónica BP.m**”, apéndice A.10, inicialmente se hace una asignación aleatoria de los patrones de entrenamiento. La red cuenta con 4 capas cada una con un número de neuronas previamente fijada y una quinta capa o capa salida que está conformada por 8 *bits*, lo que corresponde a la longitud de las etiquetas. Se utiliza la función *randsn* para determinar en el primer *epoch* los pesos (W1, W2, W3, W4, W5) y los bias (B1, B2, B3, B4, B5), al azar para cada capa.

En la fase de presentación de los patrones se emplea la función *tansig* (función de transferencia tangente hiperbólica) que devuelve la salida de cada capa usando los datos de la capa anterior (A1, A2, A3, A4, A5). Se determina la diferencia (E) entre la salida (A5) y el objetivo que corresponde a la etiqueta original, luego se hace la suma cuadrática de ese error (SSE) con la función *sumsq*.

■ Aprendizaje

En esta fase se realiza el cálculo del gradiente descendiente en retropropagación para cada capa, para ello se emplea la función *deltatan*, se empieza desde la última capa hacia la primera, para cada capa se utiliza la salida correspondiente a esa capa y el delta y peso de la capa anterior, para así poder modificar los pesos y bias en la dirección del gradiente mínimo empleando la función *learnbp* y la tasa de aprendizaje (*lr*).

Se reasignan los pesos y los bias en la dirección del gradiente mínimo, los nuevos valores serán la suma de los anteriores y los actuales, se guarda el vector de errores (SSE). Al finalizar ese

ciclo se repite nuevamente para cada fila de la matriz Pcf_ent hasta obtener el error en el entrenamiento ($Error_ent$) que corresponde a la suma cuadrática del vector de errores.

■ Test

Para calcular el error en el test, se determina la salida de la red (Y) empleando la función *tansig*, cuyas entradas son los pesos y los bias previamente calculados y la data de test.

$$Y = \text{tansig}(W5 * \text{tansig}(W4 * \text{tansig}(W3 * (\text{tansig}(W2 * \text{tansig}(W1 * test, B1), B2), B3), B4), B5)$$

Y es una matriz $n \times m$, donde n = número de sujetos y m = número de neuronas de la capa de salida. De la matriz Pcf_test se extraen solo las etiquetas y se crea una nueva matriz T_test , con la que se hace la diferencia con la matriz Y , obteniendo el vector de errores test ($Error_test$).

Se determinan las etiquetas encontradas (Ee) convirtiendo de binario a decimal la salida ($A5$) y a la vez se hace lo mismo para las etiquetas dadas (Ed) convirtiendo de binario a decimal la matriz T_test . Se hace la diferencia entre estas para realizar el gráfico del error en función de los *epoches*. Se verifica para cada presentación el conjunto de pesos asignados que minimizan la suma cuadrática del error, la presentación continúa hasta completar todos los *epoches* o cuando la $SSE \geq err_goal$.

En la figura 3.33 se observa el gráfico descrito anteriormente. En este se muestra el entrenamiento de la red, la gráfica roja representa el logaritmo del error del test ($Error_test$) mientras que la gráfica azul representa el logaritmo del error del entrenamiento ($Error_ent$), ambas en función del *epoch*. Para lograr un buen entrenamiento lo ideal es que estas gráficas converjan por debajo del eje x, cuando el error entre la salida de la red y la salida deseada es lo suficientemente pequeño (casi nulo) como muestra la gráfica verde, se dice que la red está entrenada. Además es importante que durante el entrenamiento la gráfica roja esté por encima de la azul para evitar un sobre-entrenamiento de la red.

Finalmente, se guardan los parámetros de la red en una estructura que contiene los pesos y los bias por capa.

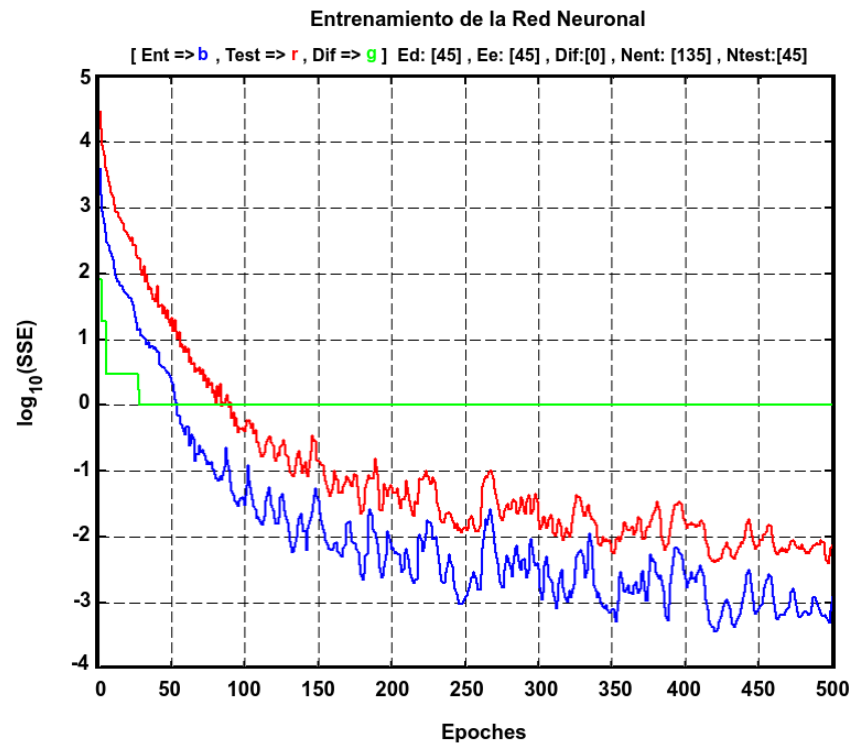


Figura 3.33: Entrenamiento de la red neuronal.

3.6. V. Método de validación de la red.

3.6.1. Función de la red.

Una vez entrenada la red, esta se encuentra en fase de producción. En esta fase se realiza el proceso de propagación de la señal hacia adelante hasta conseguir la salida pero los pesos ya no serán ajustados.

La primera capa se calcula empleando la función *tansig* con los pesos y bias correspondientes y usando los patrones de entrada que está compuesto por todos los ejemplos, tanto de entrenamiento como de test (Pcf), y así se hace sucesivamente por cada capa hasta llegar a la salida de la última capa. Este procedimiento se realiza dentro del programa “**Fase de test y producción.m**”, apéndice [A.11](#)

3.6.2. Matriz de confusión RNA.

Siguiendo el mismo procedimiento de la sección de PCA, empleando el programa “**Matriz de confusión.m**”, apéndice [A.12](#), se construye la matriz de confusión para la red neuronal para poder visualizar que el entrenamiento haya sido eficaz y en caso contrario repetirlo hasta obtener un valor superior a 90 % de eficiencia. En la figura 3.34 se muestra la matriz de confusión RNA para 10 sujetos, en las abscisas se tienen las etiquetas dadas y en las ordenadas las etiquetas encontradas. La pendiente es $m=1$, lo que representa una eficiencia del 100 %.

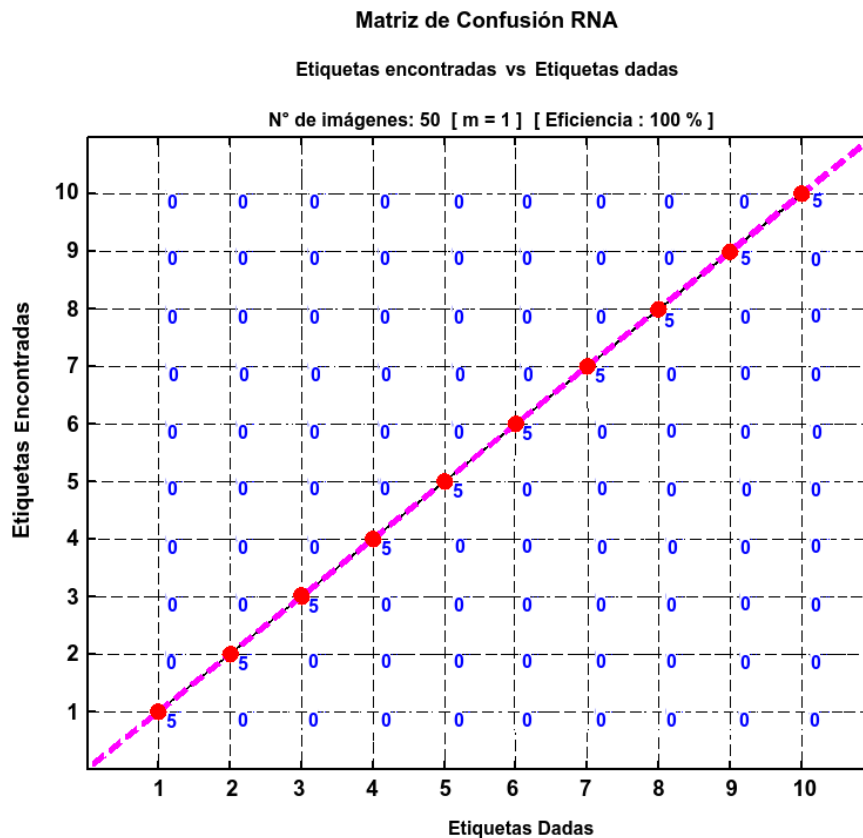


Figura 3.34: Matriz de confusión con Red Neuronal para 10 sujetos.

3.6.3. Proyección del paciente desconocido.

El programa “**Proyección del paciente desconocido.m**”, apéndice A.13, contiene el denominado “Paciente_test” que corresponde al grupo de sujetos de entrenamiento más los nuevos pacientes desconocidos. Los datos de entrada como: índice óptimo, número de autovalores, tiempo, sigma, número de tarea, frecuencia de muestreo, son los mismos valores que se emplearon durante el PCA. Además son necesarias las variables V (espacio de características) y $Proj$ (proyección de los sujetos entrenamiento), todas las salidas del programa “**Análisis Componentes Principales.m**”.

Siguiendo la misma rutina que se hizo durante el PCA, los datos son redimensionados a una misma longitud, se eliminan los *outliers*, se crean las ventanas de datos y se obtiene una nueva matriz Dr , empleando la función “**Redimensiona los datos.m**”. Esta matriz se reescribe de manera que quede bidimensional, esta nueva matriz llamada At contiene todos los ejemplos de todas las tareas por cada paciente_test. Se determina la media de cada uno y se proyecta sobre el espacio de características V . Se calcula la correlación de los pacientes_test con respecto a los de entrenamiento y se guardan esas proyecciones con su respectiva etiqueta. Se crea la Base_Paciente_Test que contiene la proyección (PCF) y la posición de los atributos y etiquetas.

3.6.4. Clasificación.

Como propuesta de una posible clasificación se hace la proyección de estos datos sobre la matriz de confusión obtenida durante el entrenamiento de la red neuronal, para ello se utilizan los pesos y bias ya calculados y se emplea el programa “Fase de test y producción.m” para la obtención de la función de la red (Y). En la figura 3.35 se muestran cuatro cuadrantes, el que está en la izquierda superior es la zona prohibida, allí no debe aparecer ningún sujeto ya que indicaría que el entrenamiento de la red no fue el adecuado, en la izquierda inferior es la red neuronal entrenada con 100 % de eficiencia, el de la derecha inferior es la proyección del paciente_test: la línea verde corresponde a los pacientes con los que entrenó la red (en primera instancia pacientes sanos) mientras que los puntos rojos son los pacientes nuevos a clasificar, por último el cuadrante que queda corresponde a una zona desconocida, la teoría es que si un sujeto cae en ese lugar puede ser patológico. Sin embargo, no existe algún criterio que valide esto por lo que es importante introducir otro método de clasificación más especializado.

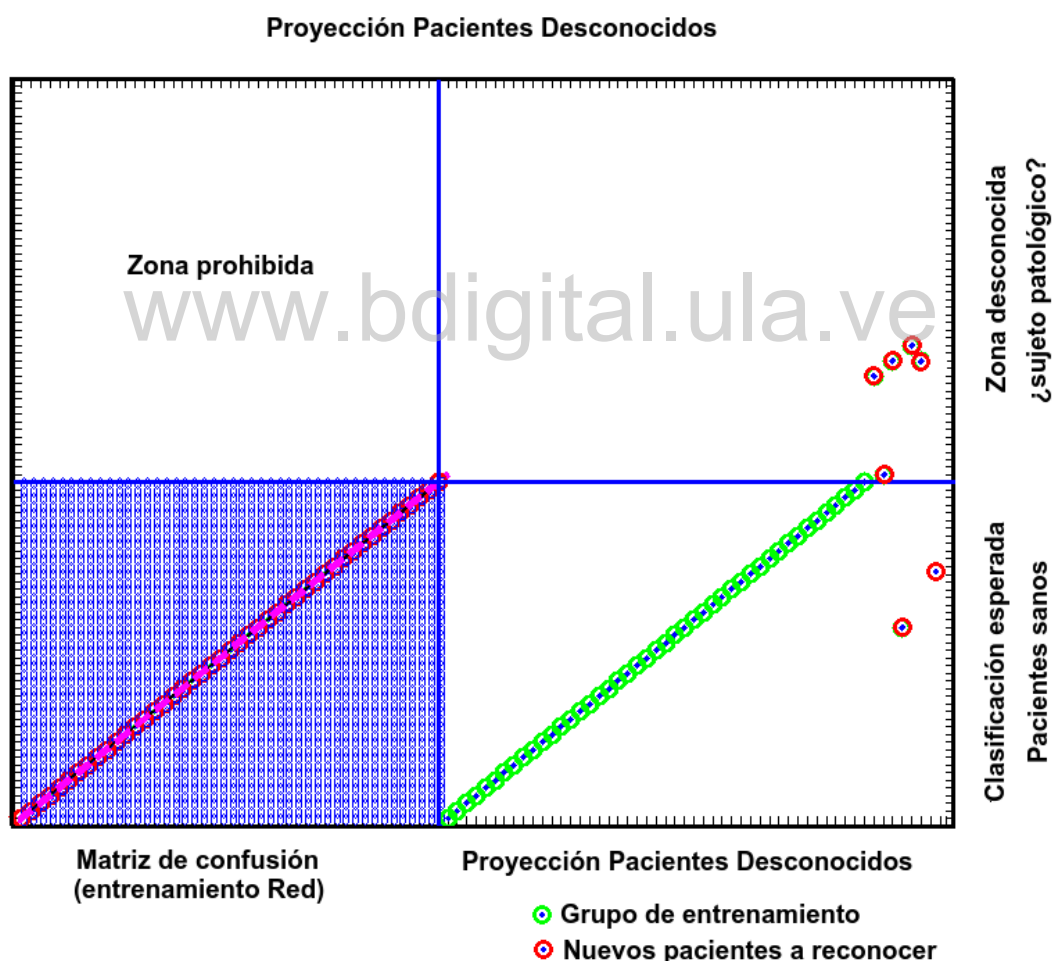


Figura 3.35: Propuesta de una posible clasificación donde se hace la proyección de los pacientes desconocidos sobre la matriz de confusión RNA, se tienen dos zonas de clasificación: la primera se corresponde con los sujetos sanos, ya que si un paciente se ubica en ésta es porque tiene características muy similares a los sujetos de entrenamiento, mientras que la segunda zona denominada desconocida, puede indicar que el paciente tiene características diferentes al grupo sano.

3.6.5. Dendrograma.

Un dendrograma es un diagrama que muestra la relación jerárquica entre objetos. Por lo general, se crea como resultado de un agrupamiento jerárquico. El uso principal de un dendrograma es encontrar la mejor manera de asignar objetos a grupos. El dendrograma siguiente, figura 3.36 muestra la agrupación jerárquica de seis observaciones que se muestran en el diagrama de dispersión de la izquierda [63].

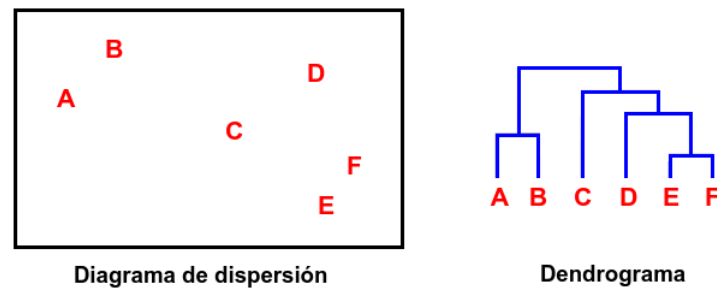


Figura 3.36: Diagrama de dispersión y su dendrograma.

La clave para interpretar un dendrograma es centrarse en la altura a la que se unen dos objetos. En el ejemplo anterior, podemos ver que E y F son más similares, ya que la altura del enlace que los une es la más pequeña. Los siguientes dos objetos más similares son A y B.

En el dendrograma anterior, la altura indica el orden en el que se unieron los grupos. Se puede crear un dendrograma más informativo donde las alturas reflejan la distancia entre los grupos, como se muestra en la figura 3.36. En este caso, el dendrograma muestra que la gran diferencia entre los conglomerados está entre el conglomerado de A y B versus el de C, D, E y F.

Es importante apreciar que el dendrograma es un resumen de la matriz de distancias y, como ocurre con la mayoría de los resúmenes, se pierde información. Por ejemplo, el dendrograma sugiere que C y D están mucho más cerca entre sí que C a B, pero los datos originales (que se muestran en el diagrama de dispersión) muestran que esto no es cierto. La consecuencia de la pérdida de información es que los dendrogramas son más precisos en la parte inferior, mostrando qué elementos son muy similares [63].

Asignar observaciones a conglomerados

Las observaciones se asignan a los conglomerados trazando una línea horizontal a través del dendrograma. Las observaciones que se unen por debajo de la línea están agrupadas. En la figura 3.37 se tienen dos grupos. Un grupo combina A y B, y un segundo grupo combina C, D, E y F.

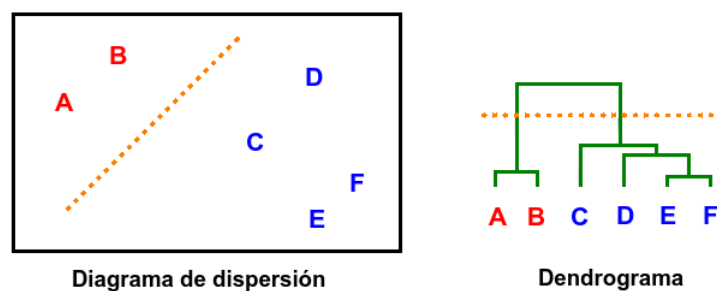


Figura 3.37: Asignación de observaciones a conglomerados.

Un error común que se comete al leer dendrogramas es asumir que la forma del dendrograma da una pista de cuántos grupos existen. En el ejemplo anterior, la interpretación (incorrecta) es que el dendrograma muestra que hay dos grupos, ya que la distancia entre los grupos (los segmentos verticales del dendrograma) es mayor entre dos y tres grupos. Los dendrogramas a menudo sugieren un número correcto de agrupaciones cuando no hay evidencia real que apoye la conclusión [63].

A partir de la función de la red (Y) y utilizando el programa “**Dendrograma Parkinson.m**”, apéndice A.14, se crea el dendrograma de clasificación, el cual emplea la distancia de Mahalanobis para su cálculo, un ejemplo de este gráfico se ve en la figura 3.38.

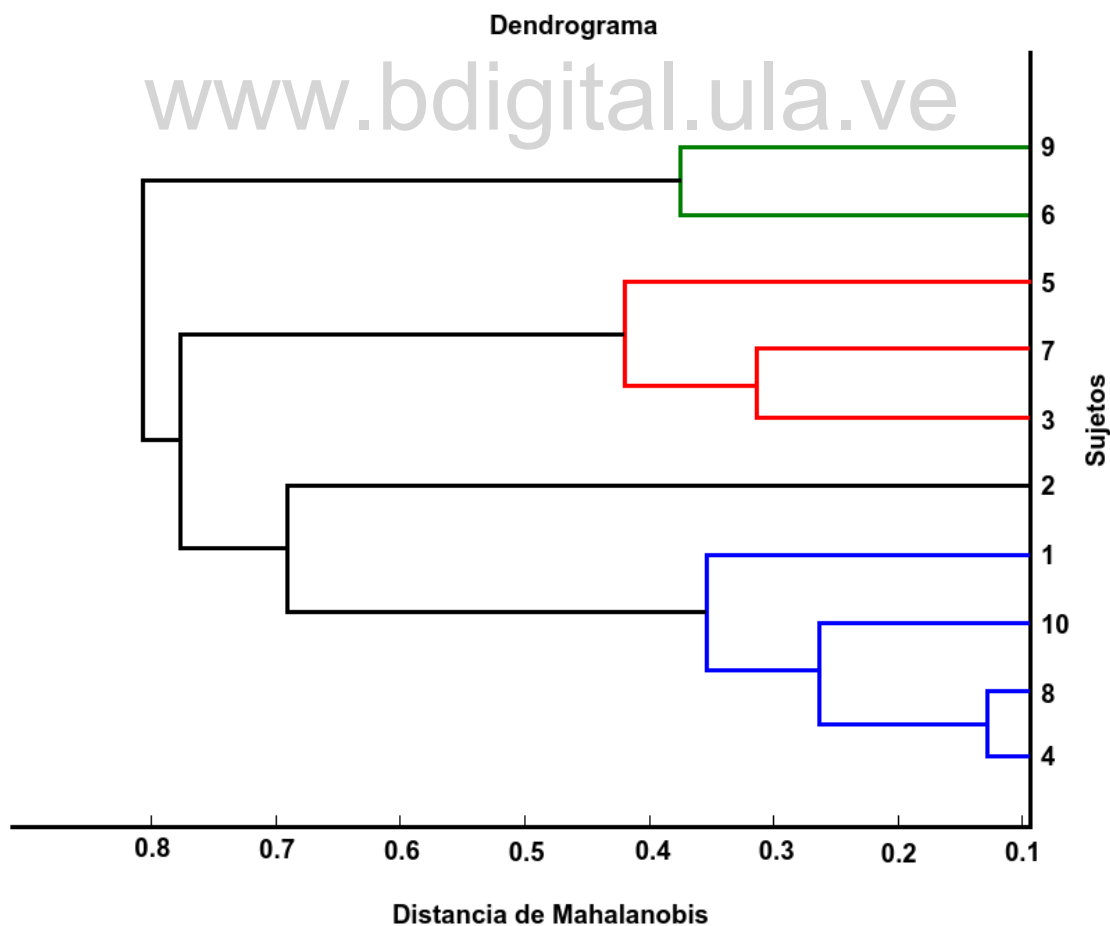


Figura 3.38: Dendrograma de clasificación con distancia de Mahalanobis.

Finalmente, se muestra un diagrama resumen desde la adquisición de los datos hasta la visualización de la clasificación a través de los dendrogramas.

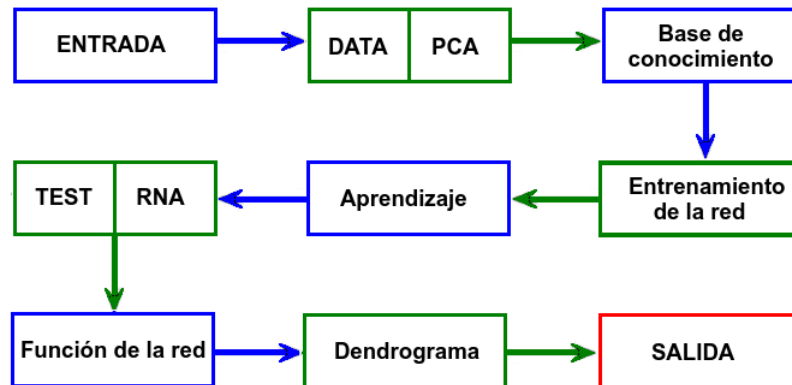


Figura 3.39: Diagrama que muestra un resumen desde la entrada de los datos hasta la clasificación.

www.bdigital.ula.ve

Capítulo 4

Resultados y análisis de los datos.

4.1. Distribución de los sujetos sanos por grupos etarios.

Inicialmente se tiene la data de 60 sujetos sanos con edades comprendidas entre los 20 y 74 años, se establecen cinco grupos etarios para poder estudiar cómo es el comportamiento dinámico a través de las décadas, en la tabla 4.1 se puede apreciar la cantidad, edad, distribución por sexo, edad promedio y desviación estándar para cada grupo.

Grupo Etario	Cantidad	%	M	F	Edad promedio	Desviación
(20-29)	13	21.6	9	4	26.2	2.0
(30-39)	13	21.6	9	4	34.2	2.7
(40-49)	12	20.0	8	4	43.3	3.0
(50-59)	11	18.3	6	5	54.5	3.5
(60-74)*	11	18.3	4	7	69.9	4.2
Total	60		36	24		

Tabla 4.1: Distribución de los sujetos sanos según la edad y sexo. *Se hace una excepción al agregar un sujeto de edad mayor al grupo etario.

Para el entrenamiento de la red se requiere una distribución equitativa por grupo etario y a la vez se requiere un grupo pequeño para la validación del método, por lo tanto se escogieron 9 sujetos por grupo. En la tabla 4.2 se observa la distribución final por sexo y promedio de las edades de los sujetos para el entrenamiento de la red, en total 45 sujetos para entrenamiento y 15 para la validación del método.

Grupo Etario	Cantidad	M	F	Edad promedio	Desviación
(20-29)	9	7	2	25.9	1.9
(30-39)	9	6	3	34.6	2.1
(40-49)	9	7	2	42.4	2.5
(50-59)	9	4	5	55.0	3.6
(60-74)	9	4	5	63.8	4.6
Total	45	28	17		

Tabla 4.2: Distribución de los sujetos sanos para el entrenamiento de la red.

4.2. Calibración del acelerómetro.

Para iniciar la adquisición de datos es necesario calibrar el acelerómetro utilizando el programa “Calibración acelerómetro ADXL345.ino”, el cual se encuentra en el apéndice B.2. El programa se ejecuta desde el entorno de desarrollo en Arduino, se deja correr unos 15 segundos de manera que se registren una cantidad alta de datos y desde el teclado se envía el comando ‘c’, que representa la calibración *offset* y ya el acelerómetro estará calibrado. En la figura 4.1 se observa el antes y después de la calibración, el acelerómetro se encuentra calibrado cuando la componente a_z es equivalente a 1g.

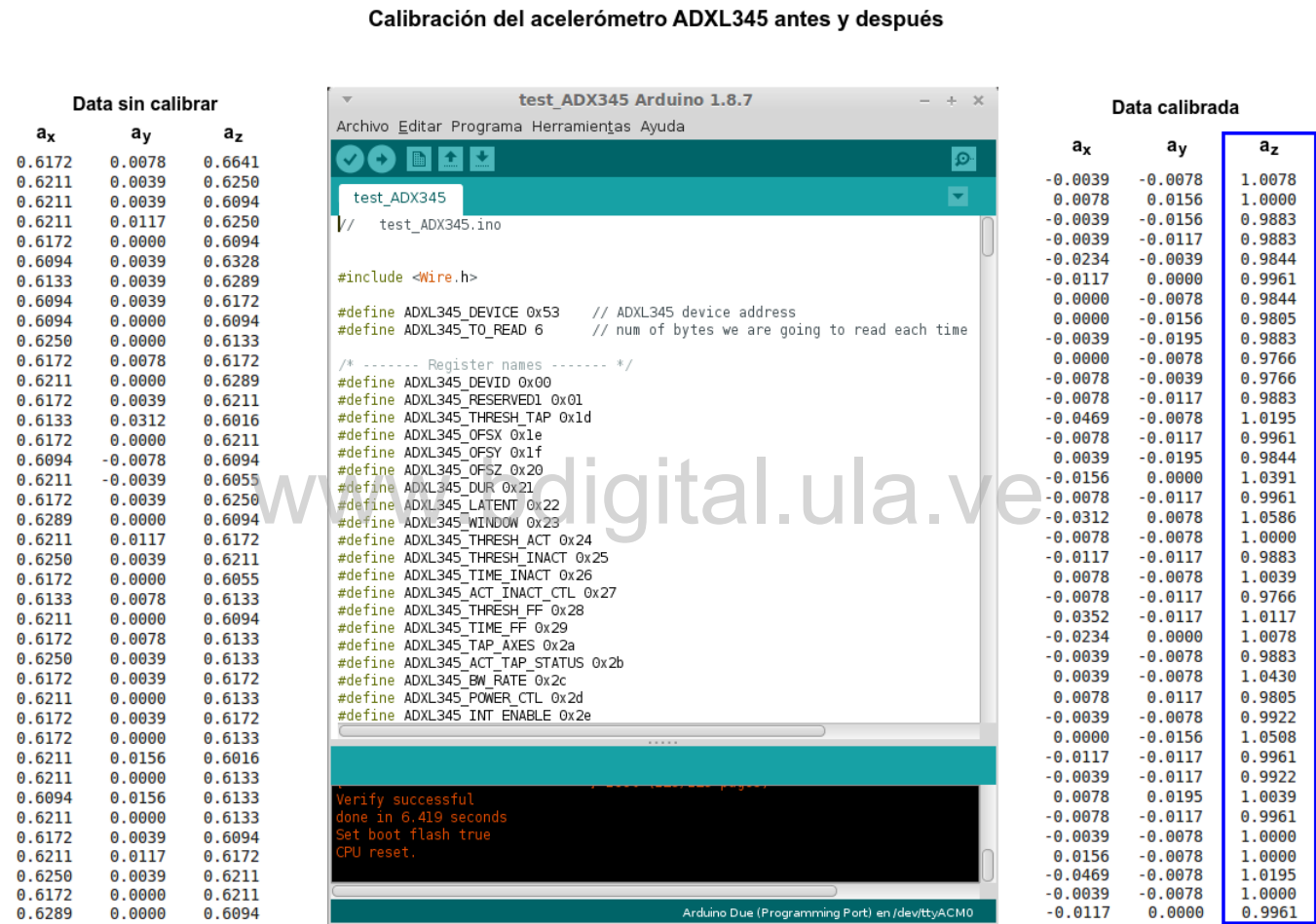


Figura 4.1: Ejemplo antes y después de calibrar el acelerómetro.

4.3. Visualización de la data en una tarea.

La figura 4.2 muestra la vista preliminar de los datos concatenados en función del tiempo para cada sensor: acelerómetro, giróscopo, magnetómetro y pulsómetro. Es importante revisar este gráfico al finalizar la adquisición de datos en cada tarea para cerciorarse de que no existan errores de comunicación entre los sensores y la computadora.

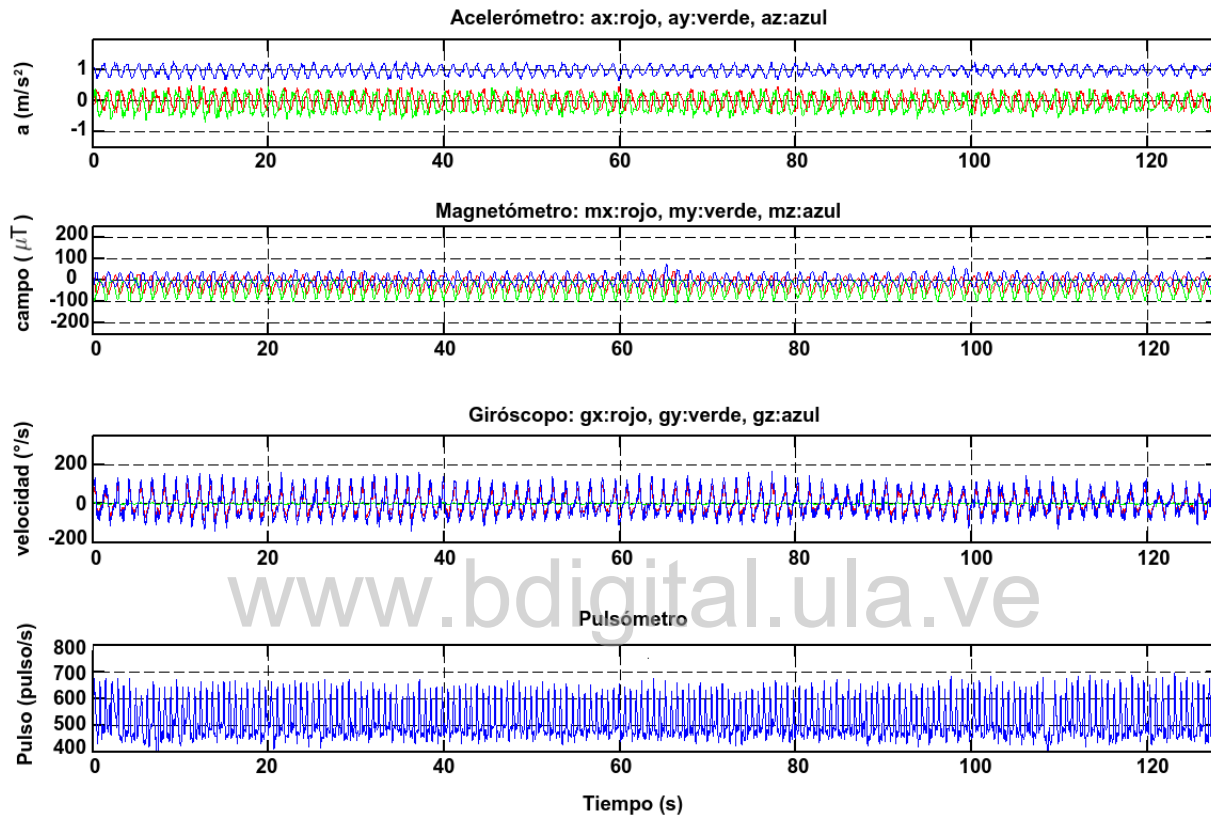


Figura 4.2: Datos concatenados para cada sensor.

4.4. Preparación de la data.

Una vez adquirida la data esta se ve como en la figura 4.3, es necesario para el entrenamiento de la red neuronal que esta se encuentre en las mismas escalas dinámicas, para ello se normaliza, la nueva normalización tendrá media cero y desviación estándar uno. La data estandarizada para una tarea y un paciente se ve en la figura 4.4.

ax	ay	az	bx	by	bz	wx	wy	wz	p
-0.1280	0.0120	1.0720	37.3900	-23.9100	-26.3400	12.1000	2.1000	63.0000	495.0000
-0.2240	-0.0680	1.1040	31.7400	-33.0400	-23.4100	7.8000	3.1000	65.7000	494.0000
-0.2200	-0.0400	1.1040	31.7400	-33.0400	-23.4100	10.8000	2.6000	63.7000	492.0000
-0.2280	-0.0640	1.1000	31.7400	-33.0400	-23.4100	12.0000	3.1000	67.9000	496.0000
-0.1840	0.0320	1.0760	25.2200	-40.8700	-22.4400	11.0000	2.6000	67.6000	496.0000
-0.0960	0.2480	1.0760	25.2200	-40.8700	-22.4400	13.2000	2.5000	69.1000	493.0000
0.0360	0.4240	1.0360	25.2200	-40.8700	-22.4400	10.5000	1.7000	66.3000	496.0000
-0.0880	0.3160	1.0600	25.2200	-40.8700	-22.4400	10.2000	2.3000	62.2000	494.0000
0.0320	0.3880	1.0400	17.3900	-46.9600	-20.4900	8.7000	2.6000	56.3000	489.0000
-0.0680	0.2440	1.0800	17.3900	-46.9600	-20.4900	8.4000	2.3000	50.8000	495.0000
-0.0400	0.2400	1.0560	17.3900	-46.9600	-20.4900	10.2000	2.1000	48.8000	505.0000
-0.0840	0.2040	1.0680	9.5700	-50.4300	-18.0500	10.3000	1.6000	49.9000	511.0000
0.0040	0.1160	1.0520	9.5700	-50.4300	-18.0500	9.4000	2.2000	46.1000	524.0000
-0.0160	0.2600	1.0480	9.5700	-50.4300	-18.0500	9.0000	1.3000	36.7000	543.0000
0.0200	0.3000	1.0440	9.5700	-50.4300	-18.0500	5.0000	1.1000	30.9000	555.0000
0.1200	0.1080	1.0360	-9.5700	-53.4800	-11.7100	0.2000	0.2000	6.0000	598.0000
0.1880	0.2560	1.0200	-9.5700	-53.4800	-11.7100	0.3000	0.1000	4.8000	583.0000
0.1120	0.2200	1.0400	-9.5700	-53.4800	-11.7100	3.1000	-0.9000	2.8000	567.0000
-0.0240	0.2680	1.0520	-9.5700	-53.4800	-11.7100	0.7000	0.2000	4.5000	550.0000
0.0400	0.2400	1.0560	-20.0000	-50.8700	-8.2900	1.5000	-0.1000	1.5000	538.0000
-0.0320	0.2160	1.0560	-20.0000	-50.8700	-8.2900	-2.1000	-0.5000	-7.1000	519.0000
0.0200	0.2640	1.0320	-20.0000	-50.8700	-8.2900	-2.4000	-0.7000	-15.4000	511.0000
0.0720	0.1440	1.0520	-30.4300	-46.5200	-4.8800	-3.4000	-1.2000	-24.8000	495.0000
0.0520	0.1560	1.0520	-30.4300	-46.5200	-4.8800	-6.6000	-1.1000	-27.3000	474.0000
0.1400	0.0560	1.0440	-30.4300	-46.5200	-4.8800	-5.9000	-1.2000	-29.1000	472.0000
0.1400	0.1160	1.0320	-30.4300	-46.5200	-4.8800	-3.9000	-2.2000	-25.9000	472.0000
0.1360	0.1800	1.0280	-39.1300	-39.1300	-1.9500	-4.2000	-1.5000	-17.3000	476.0000
0.1920	0.1400	1.0080	-39.1300	-39.1300	-1.9500	-2.6000	-1.3000	-16.7000	482.0000
0.0520	0.2640	1.0480	-39.1300	-39.1300	-1.9500	-4.6000	-0.9000	-20.5000	494.0000
0.2360	0.0760	1.0240	-39.1300	-39.1300	-1.9500	-5.4000	-1.6000	-28.9000	500.0000

Figura 4.3: Ejemplo de la data sin depurar.

www.bdigital.ula.ve

ax	ay	az	bx	by	bz	wx	wy	wz	p
0.2193	-0.8414	-0.1933	0.8380	-0.8269	-0.7131	-0.7751	-0.7293	-0.7410	-0.3383
0.0358	-0.8414	-0.0117	0.8380	-0.8269	-0.7131	-0.3973	-0.3467	-0.3795	-0.2731
-0.5951	-0.6594	0.6542	0.4432	-1.0540	-0.8693	-0.2133	-0.2460	-0.2178	-0.0286
-0.9851	-0.6594	1.1385	0.4432	-1.0540	-0.8693	-0.2326	-0.1453	-0.1618	0.4931
-1.1457	-0.3560	1.1385	0.3079	-1.1367	-0.8894	-0.0680	-0.0647	-0.0840	1.0474
-1.3407	-0.5380	1.4412	0.3079	-1.1367	-0.8894	-0.0873	-0.1453	-0.8468	1.4549
-1.6389	0.4328	1.6228	0.2909	-0.9509	-0.8945	-2.2183	-2.3001	-1.9233	1.4060
-1.5357	0.9182	1.5017	0.2909	-0.9509	-0.8945	-0.7363	-0.6890	-0.7350	0.9822
-1.6045	1.4035	1.4412	0.2909	-0.9509	-0.8945	-0.2520	-0.2863	-0.2677	0.4116
-1.8339	2.7384	1.6833	0.3756	-0.7855	-0.8743	0.2420	0.2373	0.2236	-0.1427
-0.8016	3.8912	0.5331	0.3756	-0.7855	-0.8743	0.3098	0.2977	0.3075	-0.5176
-0.5033	-0.3560	0.5331	0.6632	-0.6411	-0.7786	0.4551	0.4186	0.4672	-0.7948
-1.6389	0.0687	1.6833	0.6632	-0.6411	-0.7786	0.6391	0.6200	0.6070	-0.9252
-1.1457	0.1901	1.1385	0.6632	-0.6411	-0.7786	0.8135	0.7811	0.8027	-0.7295
-1.3866	0.9182	1.4412	1.2891	-0.2894	-0.3505	0.8425	0.8213	0.8507	-0.5665
-1.1113	0.6755	0.7753	1.2891	-0.2894	-0.3505	0.8619	0.7206	0.7987	-0.3383
-1.0769	0.6148	1.0779	1.8023	0.3508	1.0548	0.8716	1.0026	1.0244	-0.2242
-0.9966	0.6755	0.9569	1.8023	0.3508	1.0548	1.2397	1.2442	1.3479	-0.1753
-0.8933	0.1901	1.0779	0.1444	0.8056	2.3392	1.2978	1.3651	1.3939	-0.2731
-0.6754	0.4934	0.6542	0.1444	0.8056	2.3392	1.1234	1.0026	1.0763	-0.3383
-0.4575	0.1901	0.4120	0.1444	0.8056	2.3392	0.9975	0.9422	1.0044	-0.4687
-0.5951	0.6755	0.6542	-1.4458	0.3717	0.9490	1.0460	1.0026	1.0464	-0.4524
-0.5951	0.7361	0.5937	-1.4458	0.3717	0.9490	1.1138	1.1033	1.0843	-0.4035
-0.1936	0.4934	0.1699	-1.1920	-0.4134	-0.2196	0.4938	0.4991	0.5251	-0.4361
0.1849	0.7968	-0.3144	-1.1920	-0.4134	-0.2196	-0.1648	-0.1654	-0.1219	-0.3709
0.1734	1.5856	-0.3749	-1.1920	-0.4134	-0.2196	0.3098	0.2776	0.2955	-0.2894
0.4143	0.9788	-0.4355	-0.8932	-0.8269	-0.5318	0.4163	0.3783	0.4453	-0.2568
0.8387	0.7361	-0.7381	-0.8932	-0.8269	-0.5318	0.2129	0.1769	0.1777	-0.3220
1.6990	-0.1133	-1.7067	-0.8142	-0.8682	-0.5973	1.0556	1.5665	1.5337	-0.2568
1.6875	-0.5380	-1.7067	-0.8142	-0.8682	-0.5973	0.2710	0.3179	0.0578	-0.2242

Figura 4.4: Ejemplo de la data normalizada para una tarea y un paciente.

4.5. Análisis dinámico por cada tarea.

En esta sección se mostrará el estudio correspondiente a la dinámica corporal de los sujetos sanos y los pacientes con Parkinson, se quiere ver si existen diferencias entre ambas dinámicas en términos de valores medios de los parámetros, frecuencias, sincronización en los movimientos y análisis de Fourier sobre el módulo de los parámetros. Para ello se emplea el programa “**Dinámica de los pacientes.m**” que se encuentra en el apéndice [A.16](#) por cada tarea.

Se estudian las siguientes secciones:

- Frecuencia central de los parámetros por todas las tareas por pacientes.
- Frecuencia central de los parámetros por tarea y todos los sujetos.
- Valores medios de los parámetros para todos los pacientes por tarea.
- Sincronización del movimiento.
- Espectro de Fourier por parámetro y por tarea.

4.5.1. Frecuencia central de los parámetros por todas las tareas por pacientes.

Sujetos sanos

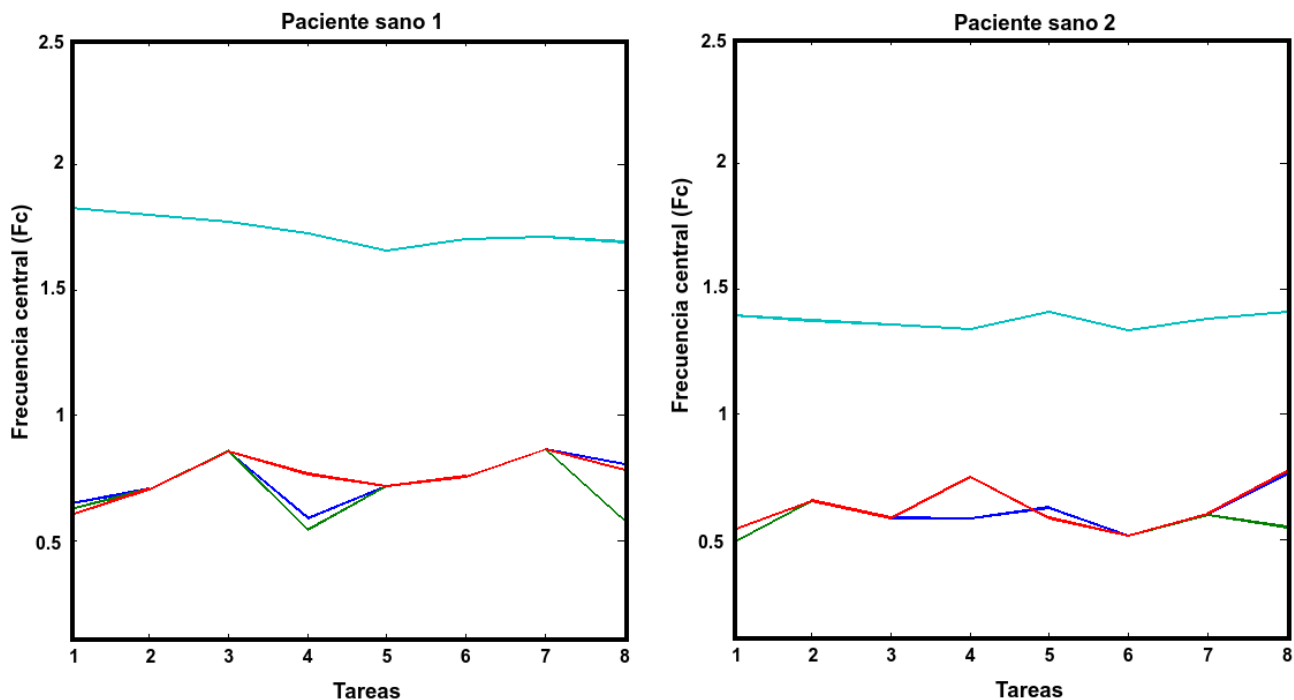


Figura 4.5: Frecuencia central de todos los parámetros y todas las tareas para un sujeto sano, aceleración (azul), campo magnético (verde), velocidad angular (rojo) y pulso cardíaco (cian).

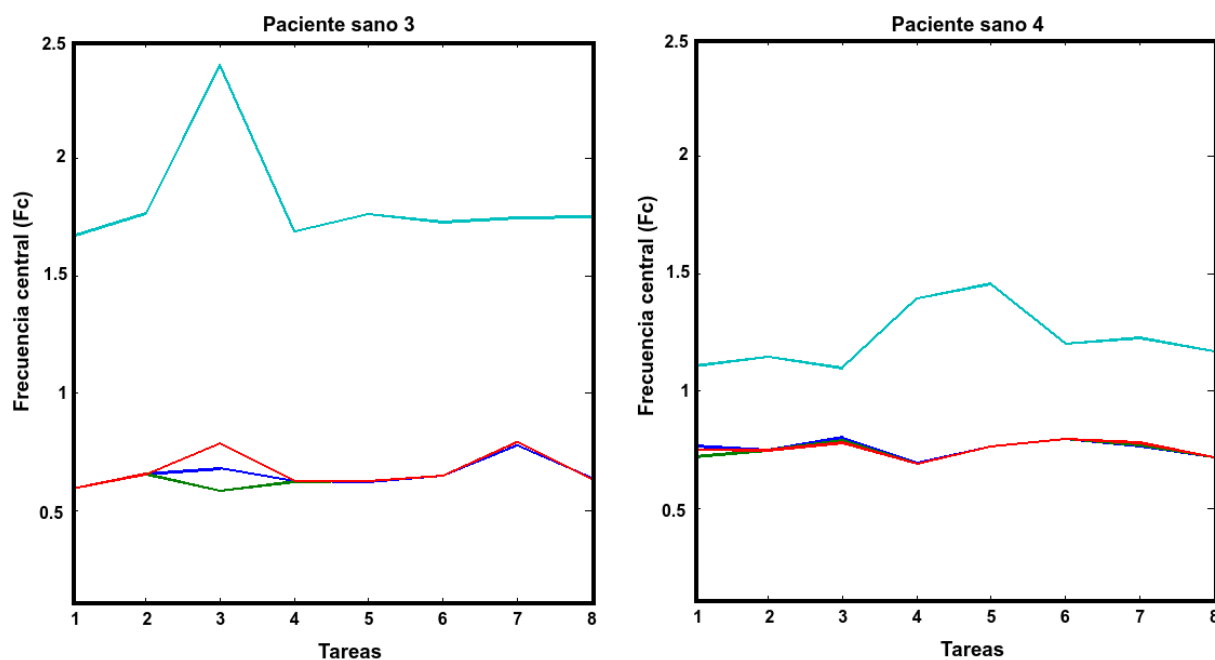


Figura 4.6: Frecuencia central de todos los parámetros y todas las tareas para un sujeto sano, aceleración (azul), campo magnético (verde), velocidad angular (rojo) y pulso cardíaco (cian).

Pacientes con Parkinson

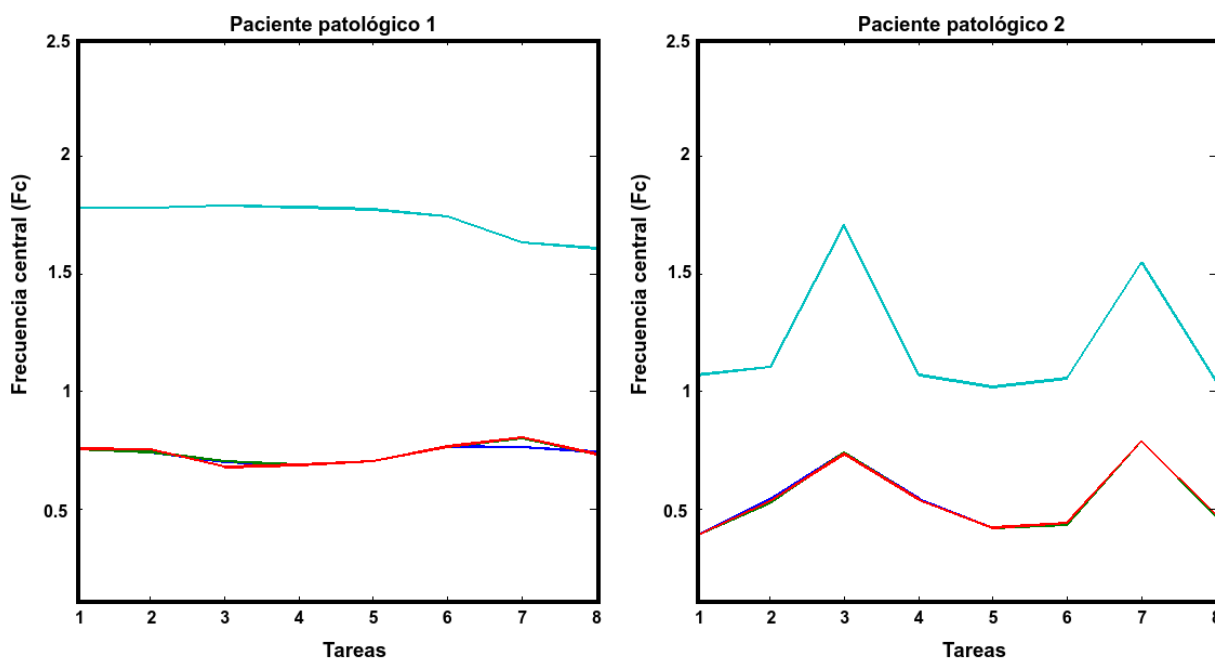


Figura 4.7: Frecuencia central de todos los parámetros y todas las tareas para un paciente con Parkinson, aceleración (azul), campo magnético (verde), velocidad angular (rojo) y pulso cardíaco (cian).

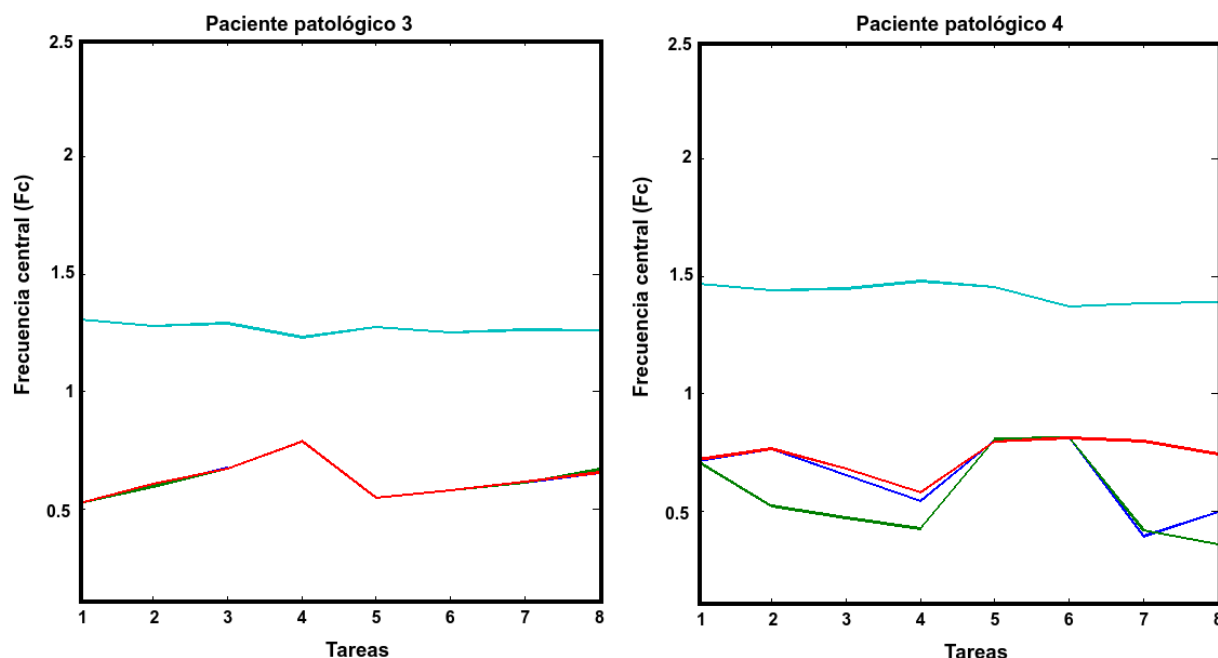


Figura 4.8: Frecuencia central de todos los parámetros y todas las tareas para un paciente con Parkinson, aceleración (azul), campo magnético (verde), velocidad angular (rojo) y pulso cardíaco (cian).

En las figuras 4.5 y 4.6 se muestran los gráficos de la frecuencia central de todos los parámetros (aceleración (azul), campo magnético (verde), velocidad angular (rojo) y pulso cardíaco (cian)) y todas las tareas para los sujetos sanos y en las figuras 4.7 y 4.8 se muestran para los pacientes con Parkinson. Se observa una regularidad en la frecuencia central para todos los casos, entre 0.5 Hz y 1 Hz, la variabilidad es muy pequeña para que dé indicios de alguna pre-clasificación de los grupos a través de la dinámica corporal con las tareas empleadas.

4.5.2. Frecuencia central de los parámetros por tarea y todos los pacientes.

Movimiento circular en el sentido horario con la mano derecha.

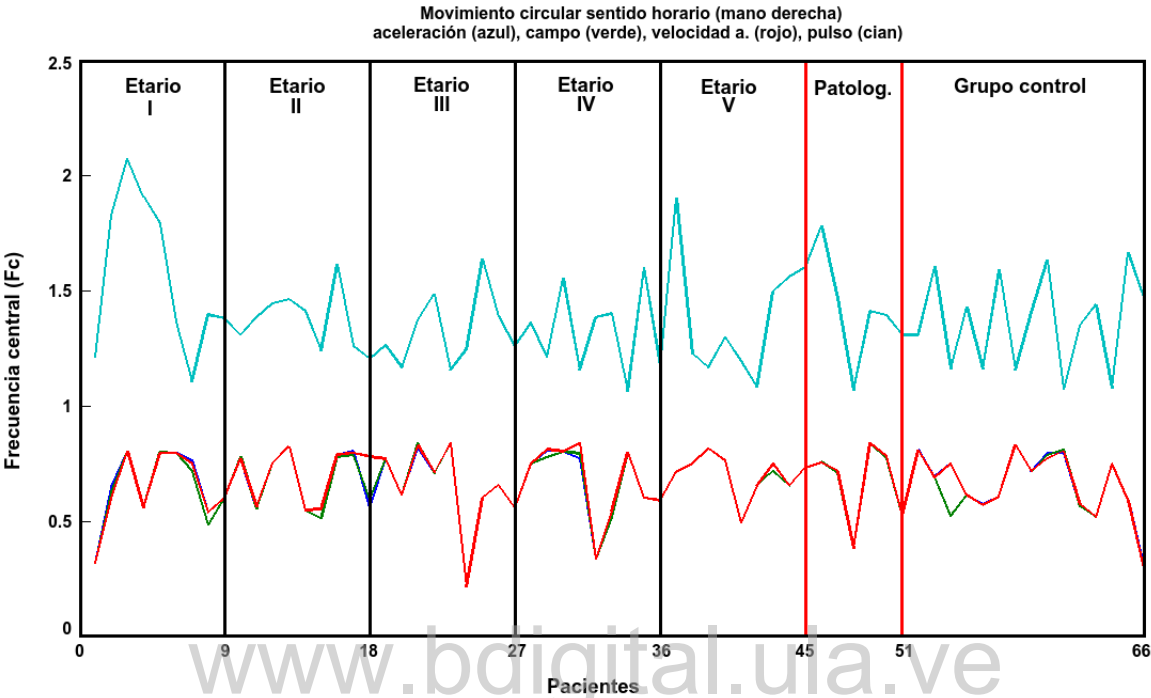


Figura 4.9: Frecuencia central para la tarea movimiento circular en el sentido horario con la mano derecha, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.65	0.68	0.64	0.66	0.69	0.66	0.65
campo m.	0.63	0.68	0.64	0.66	0.69	0.66	0.64
velocidad a.	0.64	0.70	0.64	0.67	0.70	0.66	0.65
pulso	1.56	1.36	1.32	1.32	1.39	1.40	1.36

Tabla 4.3: Valores frecuenciales para la tarea 1 por grupo etario, patológico y grupo control.

Para la tarea movimiento circular en el sentido horario con la mano derecha, se observa en la figura 4.9 un solapamiento en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular. En la tabla 4.3 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.66 ± 0.02) Hz, para el campo magnético un valor medio (0.66 ± 0.02) Hz, para la velocidad angular un valor medio (0.67 ± 0.02) Hz y para el pulso un valor medio de (1.39 ± 0.08) Hz.

Movimiento circular en el sentido antihorario con la mano derecha.

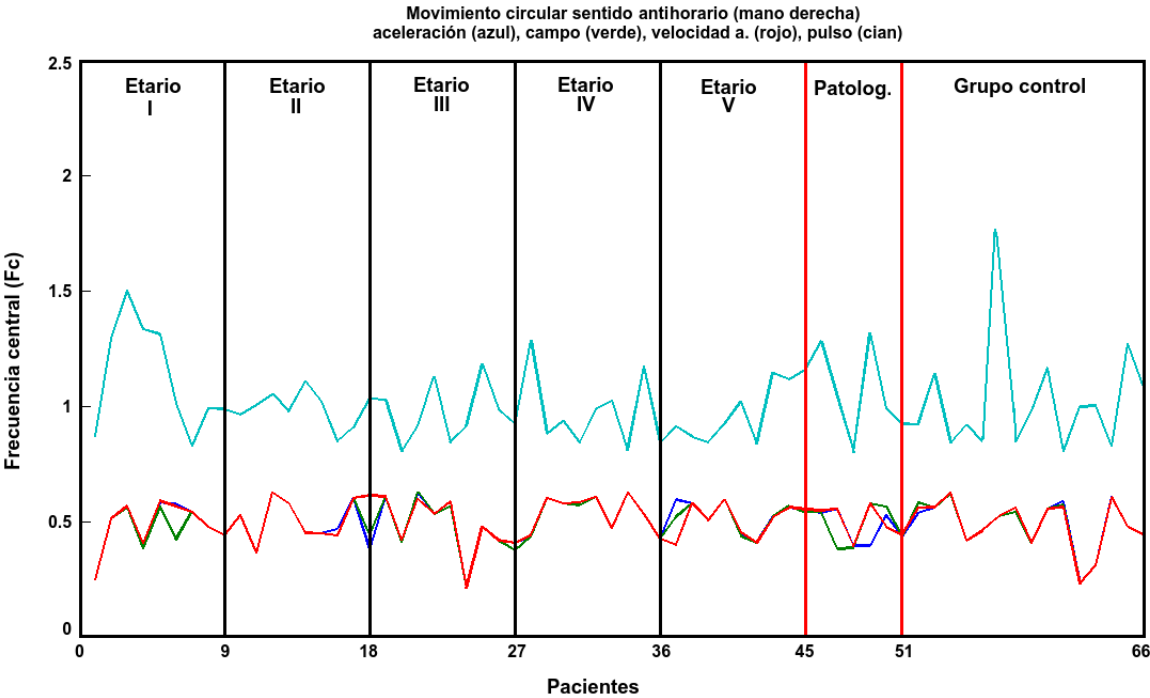


Figura 4.10: Frecuencia central para la tarea movimiento circular en el sentido antihorario con la mano derecha, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.66	0.67	0.65	0.74	0.73	0.65	0.67
campo m.	0.63	0.68	0.64	0.73	0.71	0.65	0.66
velocidad a.	0.66	0.70	0.65	0.74	0.70	0.68	0.67
pulso	1.56	1.39	1.45	1.34	1.36	1.47	1.48

Tabla 4.4: Valores frecuenciales para la tarea 2 por grupo etario, patológico y grupo control

Para la tarea movimiento circular en el sentido antihorario con la mano derecha, se observa en la figura 4.10 un solapamiento en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular en los grupos etarios II, III, IV y el grupo control, en los demás grupos hay leves diferencias. En la tabla 4.4 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.68 ± 0.03) Hz, para el campo magnético un valor medio (0.67 ± 0.03) Hz, para la velocidad angular un valor medio (0.69 ± 0.03) Hz y para el pulso un valor medio de (1.44 ± 0.07) Hz.

Movimiento lineal en dirección vertical con la mano derecha.

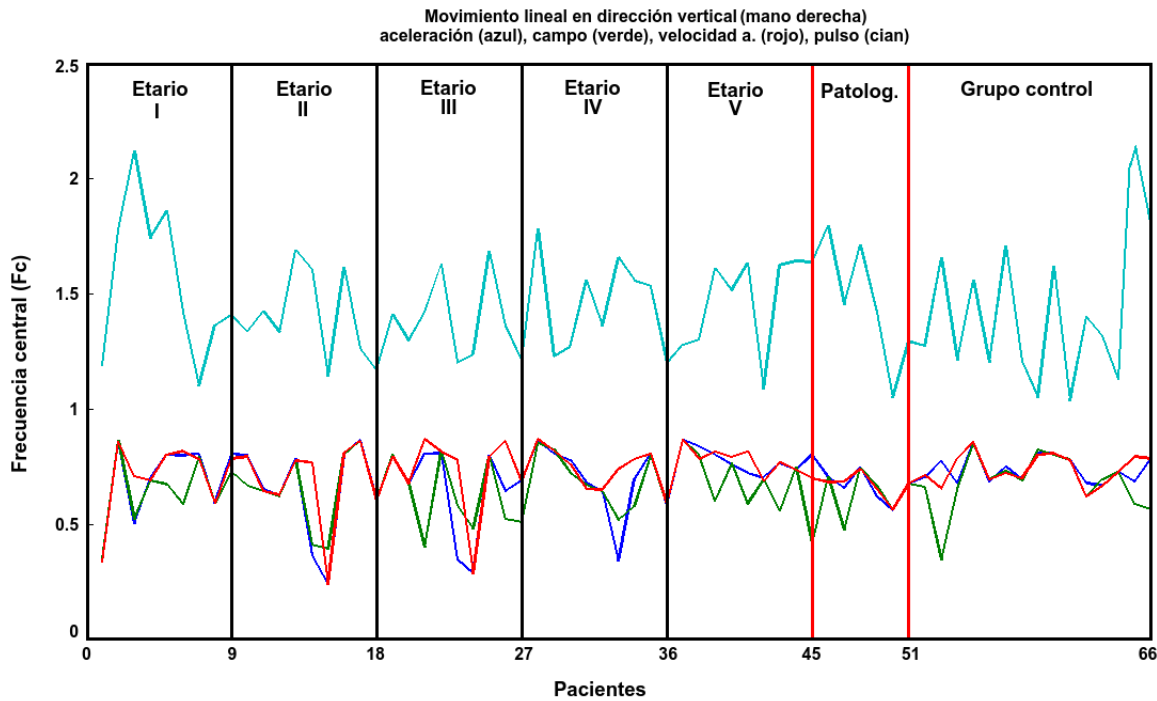


Figura 4.11: Frecuencia central para la tarea movimiento lineal en dirección vertical con la mano derecha, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.68	0.63	0.64	0.68	0.77	0.66	0.73
campo m.	0.64	0.64	0.61	0.68	0.66	0.63	0.67
velocidad a.	0.70	0.67	0.72	0.73	0.76	0.66	0.73
pulso	1.54	1.38	1.37	1.44	1.47	1.44	1.42

Tabla 4.5: Valores frecuenciales para la tarea 3 por grupo etario, patológico y grupo control

Para la tarea movimiento lineal en dirección vertical con la mano derecha, se observa en la figura 4.11 una variación en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular, sin embargo en general se tienen valores similares entre los grupos. En la tabla 4.5 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.68 ± 0.05) Hz, para el campo magnético un valor medio (0.65 ± 0.02) Hz, para la velocidad angular un valor medio (0.71 ± 0.03) Hz y para el pulso un valor medio de (1.44 ± 0.05) Hz.

Movimiento lineal en dirección horizontal con la mano derecha.

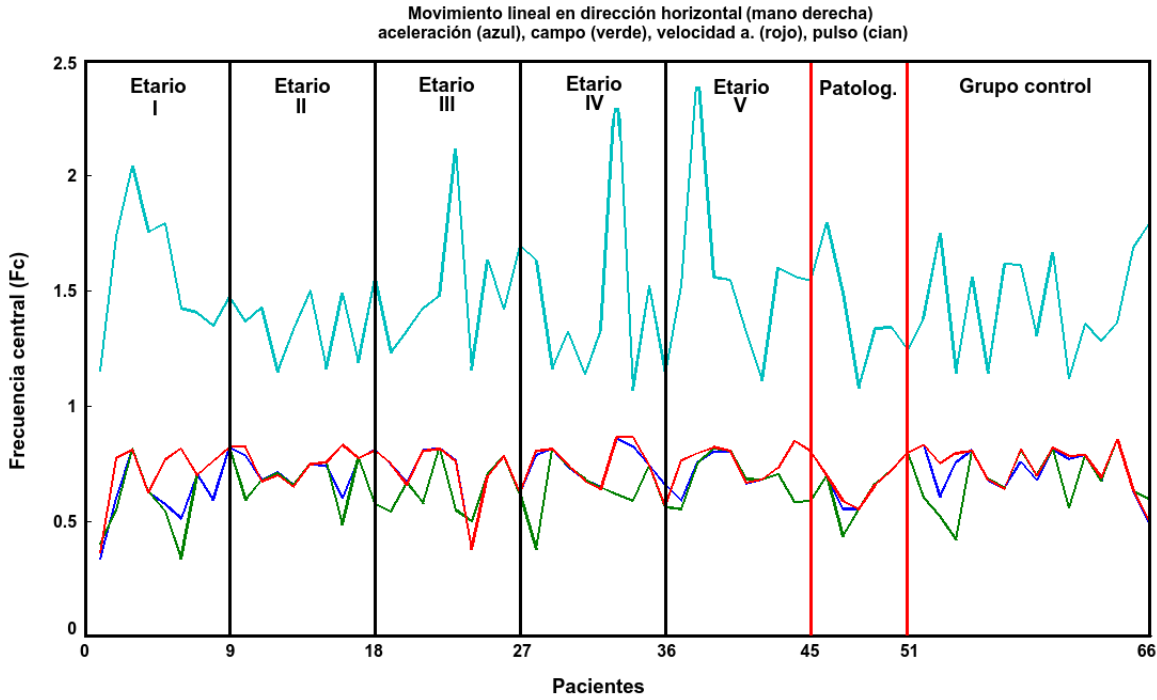


Figura 4.12: Frecuencia central para la tarea movimiento lineal en dirección horizontal con la mano derecha, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.61	0.71	0.69	0.73	0.73	0.65	0.71
campo m.	0.61	0.65	0.63	0.63	0.67	0.63	0.66
velocidad a.	0.70	0.74	0.69	0.73	0.75	0.66	0.73
pulso	1.56	1.35	1.46	1.27	1.56	1.34	1.44

Tabla 4.6: Valores frecuenciales para la tarea 4 por grupo etario, patológico y grupo control

Para la tarea movimiento lineal en dirección horizontal con la mano derecha, se observa en la figura 4.12 una variación en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular, sin embargo en general se tienen valores similares entre los grupos. En la tabla 4.6 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.69 ± 0.04) Hz, para el campo magnético un valor medio (0.64 ± 0.02) Hz, para la velocidad angular un valor medio (0.71 ± 0.03) Hz y para el pulso un valor medio de (1.42 ± 0.10) Hz.

Movimiento circular en el sentido horario con la mano izquierda.

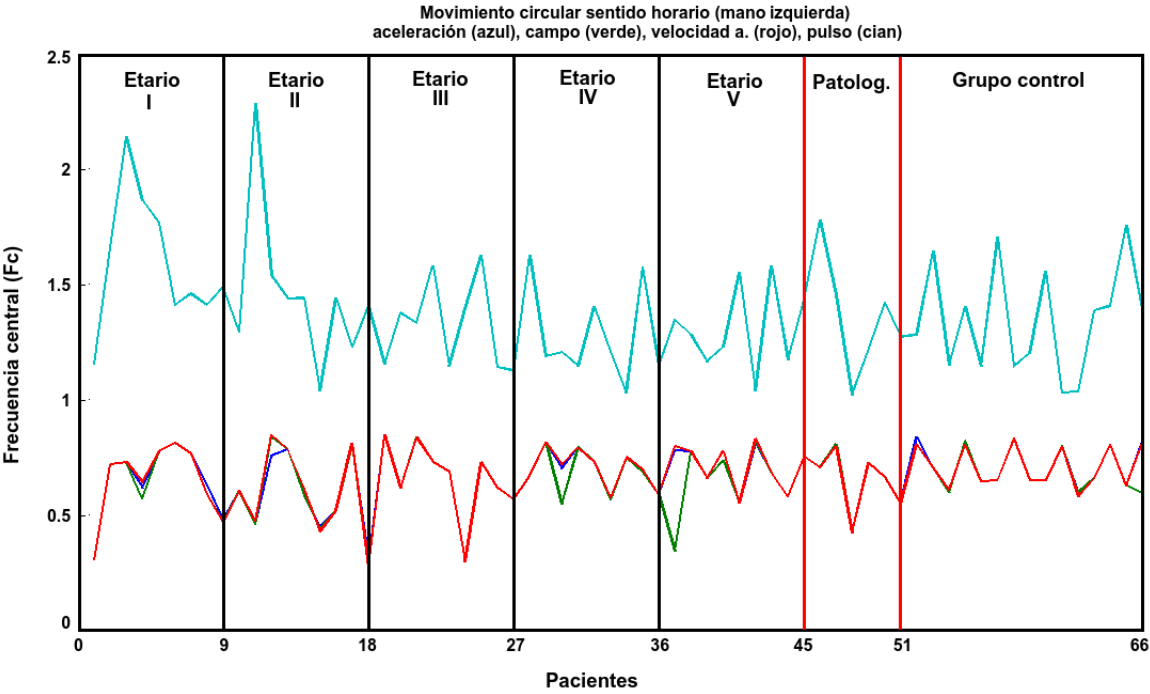


Figura 4.13: Frecuencia central para la tarea movimiento circular en el sentido horario con la mano izquierda, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.64	0.58	0.65	0.69	0.70	0.64	0.71
campo m.	0.63	0.59	0.65	0.67	0.65	0.64	0.69
velocidad a.	0.64	0.59	0.65	0.70	0.70	0.64	0.70
pulso	1.59	1.48	1.31	1.27	1.30	1.35	1.41

Tabla 4.7: Valores frecuenciales para la tarea 5 por grupo etario, patológico y grupo control

Para la tarea movimiento circular en el sentido horario con la mano izquierda, se observa en la figura 4.13 un solapamiento en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular. En la tabla 4.7 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.66 ± 0.04) Hz, para el campo magnético un valor medio (0.65 ± 0.03) Hz, para la velocidad angular un valor medio (0.66 ± 0.04) Hz y para el pulso un valor medio de (1.39 ± 0.11) Hz.

Movimiento circular en el sentido antihorario con la mano izquierda.

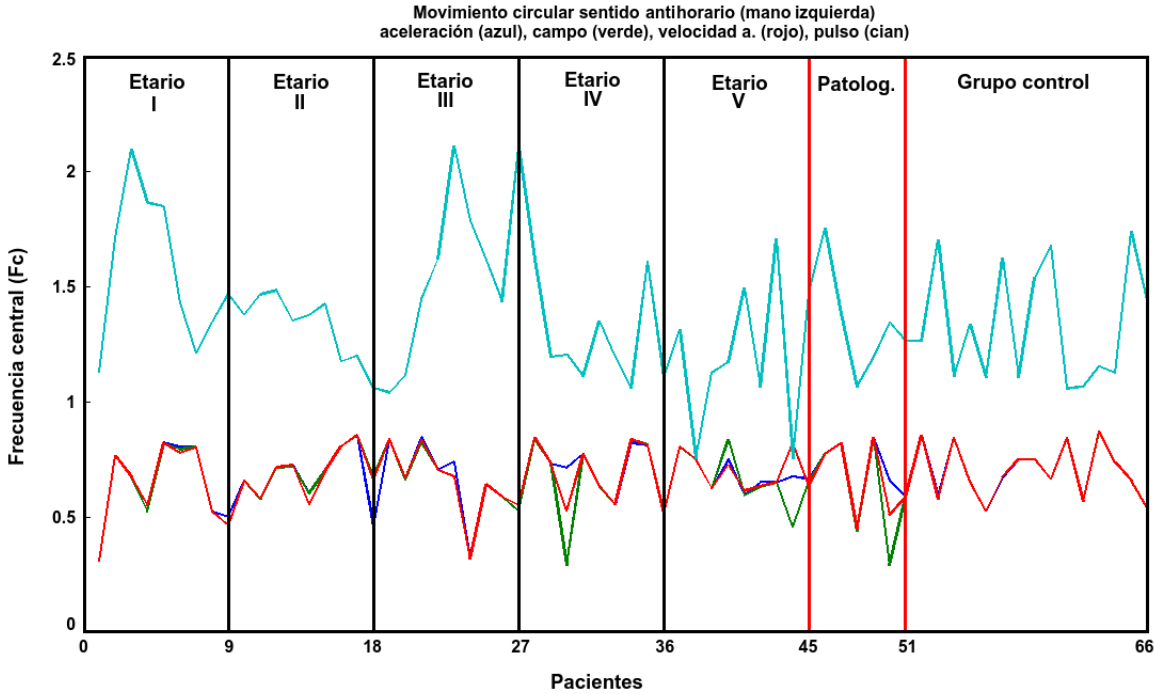


Figura 4.14: Frecuencia central para la tarea movimiento circular en el sentido antihorario con la mano izquierda, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.63	0.66	0.64	0.70	0.67	0.68	0.69
campo m.	0.62	0.69	0.63	0.65	0.65	0.62	0.69
velocidad a.	0.62	0.68	0.64	0.68	0.68	0.65	0.69
pulso	1.55	1.31	1.54	1.26	1.18	1.32	1.32

Tabla 4.8: Valores frecuenciales para la tarea 6 por grupo etario, patológico y grupo control

Para la tarea movimiento circular en el sentido antihorario con la mano izquierda, se observa en la figura 4.14 un solapamiento en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular, para los grupos etarios I, II y grupo control, en los demás grupos hay leves diferencias. En la tabla 4.8 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.67 ± 0.02) Hz, para el campo magnético un valor medio (0.65 ± 0.03) Hz, para la velocidad angular un valor medio (0.66 ± 0.02) Hz y para el pulso un valor medio de (1.35 ± 0.13) Hz.

Movimiento lineal en dirección vertical con la mano izquierda.

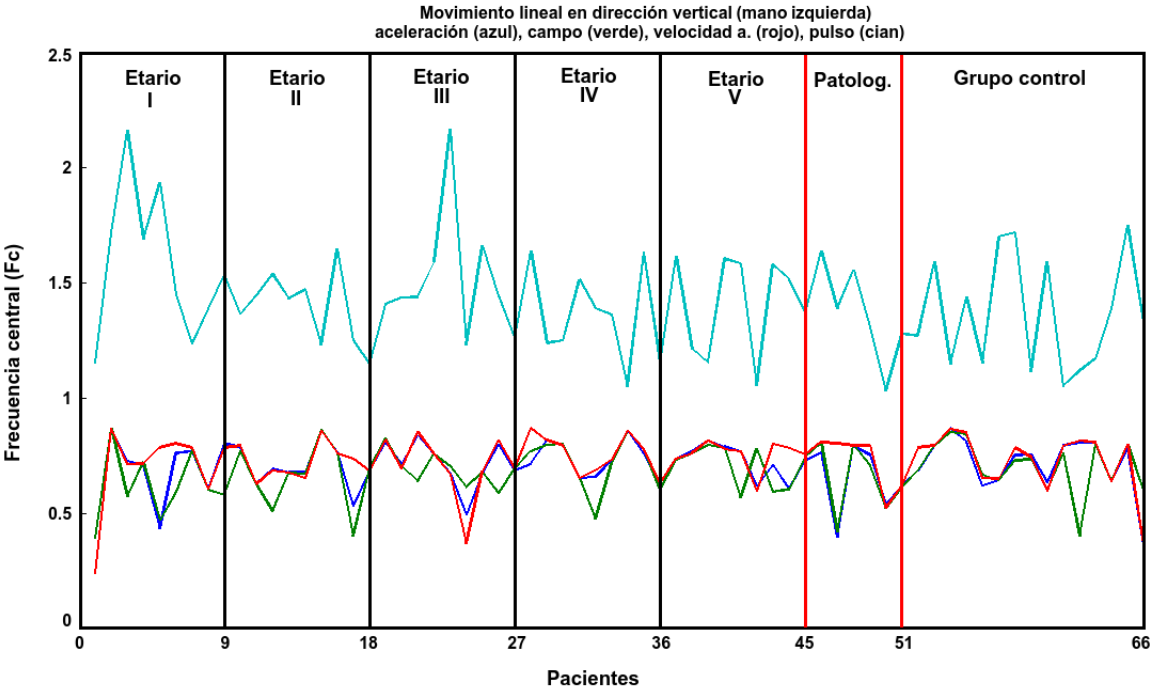


Figura 4.15: Frecuencia central para la tarea movimiento lineal en dirección vertical con la mano izquierda, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.65	0.69	0.70	0.72	0.72	0.64	0.71
campo m.	0.61	0.65	0.68	0.71	0.70	0.64	0.69
velocidad a.	0.69	0.71	0.70	0.75	0.74	0.72	0.72
pulso	1.57	1.39	1.49	1.35	1.40	1.37	1.38

Tabla 4.9: Valores frecuenciales para la tarea 7 por grupo etario, patológico y grupo control

Para la tarea movimiento lineal en dirección vertical con la mano izquierda se observa en la figura 4.15 una variación en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular. En la tabla 4.9 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.69 ± 0.03) Hz, para el campo magnético un valor medio (0.67 ± 0.03) Hz, para la velocidad angular un valor medio (0.72 ± 0.02) Hz y para el pulso un valor medio de (1.42 ± 0.07) Hz.

Movimiento lineal en dirección horizontal con la mano izquierda.

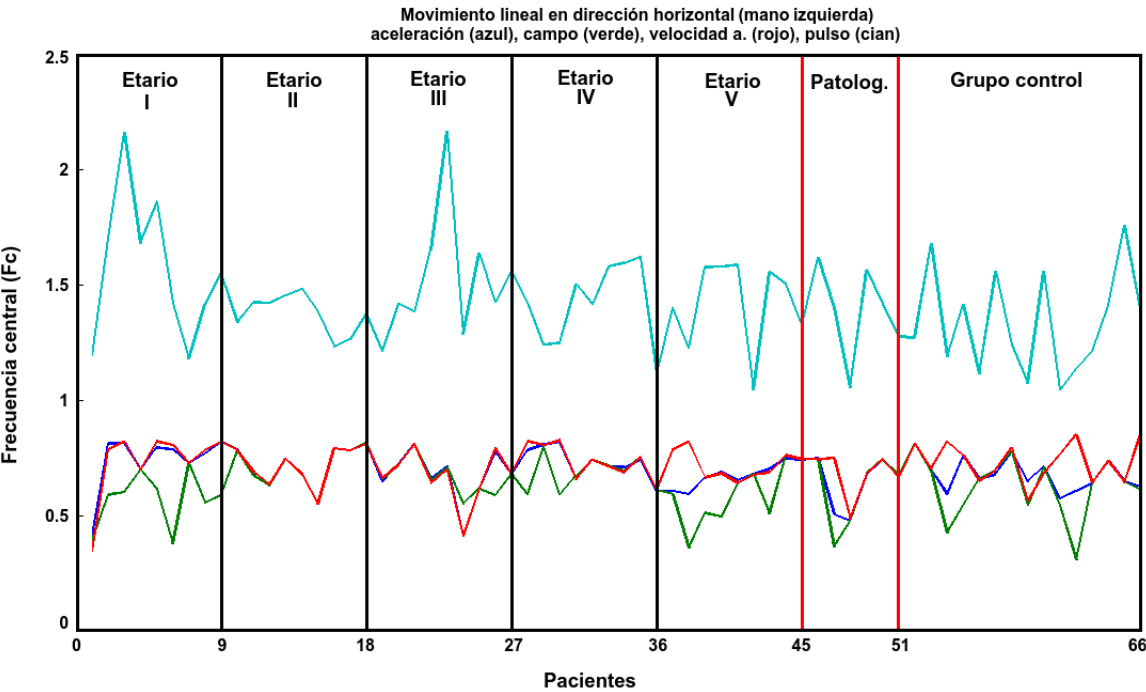


Figura 4.16: Frecuencia central para la tarea movimiento lineal en dirección horizontal con la mano izquierda, para todos los pacientes por grupo etario, patológico y grupo control.

Parámetro	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)	Control (Hz)
aceleración	0.72	0.70	0.66	0.72	0.66	0.63	0.67
campo m.	0.56	0.70	0.65	0.67	0.57	0.66	0.72
velocidad a.	0.72	0.70	0.66	0.72	0.70	0.67	0.72
pulso	1.56	1.36	1.53	1.37	1.41	1.37	1.32

Tabla 4.10: Valores frecuenciales para la tarea 8 por grupo etario, patológico y grupo control

Para la tarea movimiento lineal en dirección horizontal con la mano izquierda, se observa en la figura 4.16 una variación en el valor de la frecuencia para los módulos de la aceleración, campo magnético y velocidad angular, salvo para el grupo etario II. En la tabla 4.10 se tienen los valores de las frecuencias centrales de todos los parámetros para cada grupo etario, el grupo patológico y el grupo control. Para la aceleración se tiene un valor medio (0.68 ± 0.03) Hz, para el campo magnético un valor medio (0.65 ± 0.06) Hz, para la velocidad angular un valor medio (0.70 ± 0.02) Hz y para el pulso un valor medio de (1.42 ± 0.08) Hz.

4.5.3. Valores medios de los parámetros para todos los pacientes por tarea.

En la figura 4.17 se presenta un diagrama de barras del valor medio de la aceleración para cada tarea por grupo etario de los sujetos sanos y el grupo patológico, estos valores se muestran en la tabla 4.11. Se observa muy poca diferencia entre los grupos, el valor promedio con su desviación estándar por cada tarea es la siguiente: Tarea 1 = (1.08 ± 0.03) g, Tarea 2 = (1.09 ± 0.02) g, Tarea 3 = (1.08 ± 0.01) g, Tarea 4 = (1.10 ± 0.03) g, Tarea 5 = (1.12 ± 0.02) g, Tarea 6 = (1.10 ± 0.01) g, Tarea 7 = (1.10 ± 0.02) g y Tarea 8 = (1.08 ± 0.03) g. Los valores de la aceleración no son suficientes para establecer una clasificación entre los sujetos sanos y patológicos.

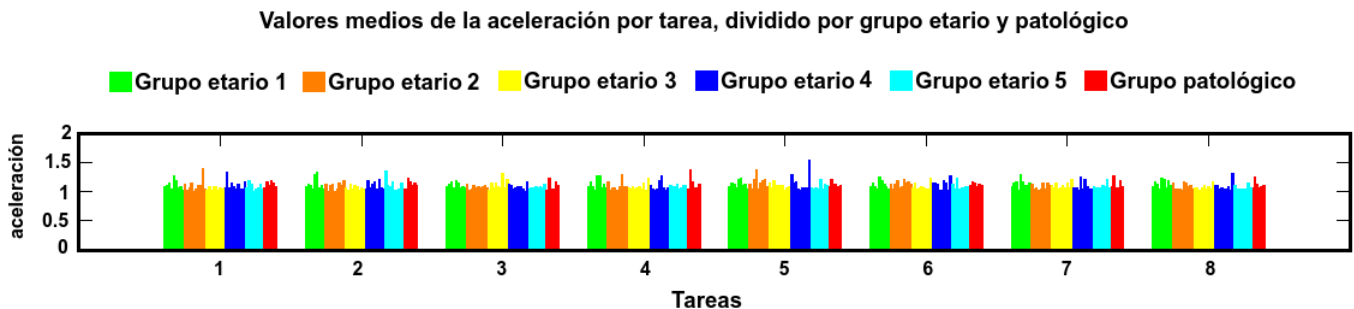


Figura 4.17: Gráfico de barras del valor medio de la aceleración por grupo etario de los sujetos sanos y el grupo patológico.

Tarea	Etario I (unid de g)	Etario II (unid de g)	Etario III (unid de g)	Etario IV (unid de g)	Etario V (unid de g)	Patológico (unid de g)
Tarea 1	1.10	1.09	1.04	1.10	1.05	1.11
Tarea 2	1.12	1.08	1.05	1.09	1.10	1.12
Tarea 3	1.10	1.07	1.09	1.07	1.08	1.09
Tarea 4	1.12	1.09	1.06	1.10	1.07	1.13
Tarea 5	1.12	1.15	1.09	1.10	1.13	1.11
Tarea 6	1.11	1.11	1.07	1.10	1.10	1.11
Tarea 7	1.13	1.07	1.08	1.10	1.07	1.12
Tarea 8	1.12	1.07	1.05	1.06	1.08	1.11

Tabla 4.11: Valores medios del módulo de la aceleración por tarea de los grupos etarios y patológico.

En la figura 4.18 se presenta un diagrama de barras del valor medio del campo magnético para cada tarea por grupo etario de los sujetos sanos y el grupo patológico, estos valores se muestran en la tabla 4.12. Se observa una diferencia marcada entre las tareas de movimiento circular (tareas 1,2,5,6) y las tareas de movimiento lineal (3,4,7,8). El valor promedio con su desviación estándar por cada tarea es la siguiente: Tarea 1 = (60 ± 3) μ T, Tarea 2 = (63 ± 5) μ T, Tarea 3 = (119 ± 7) μ T, Tarea 4 = (113 ± 7) μ T, Tarea 5 = (58 ± 4) μ T, Tarea 6 = (58 ± 4) μ T, Tarea 7 = (120 ± 4) μ T y Tarea 8 = (114 ± 4) μ T. Los valores del campo magnético pueden introducir una previa clasificación inter-tarea, lo que justifica la clasificación en términos de espacialidad, aún así no es suficiente para establecer una clasificación entre los sujetos sanos y patológicos.

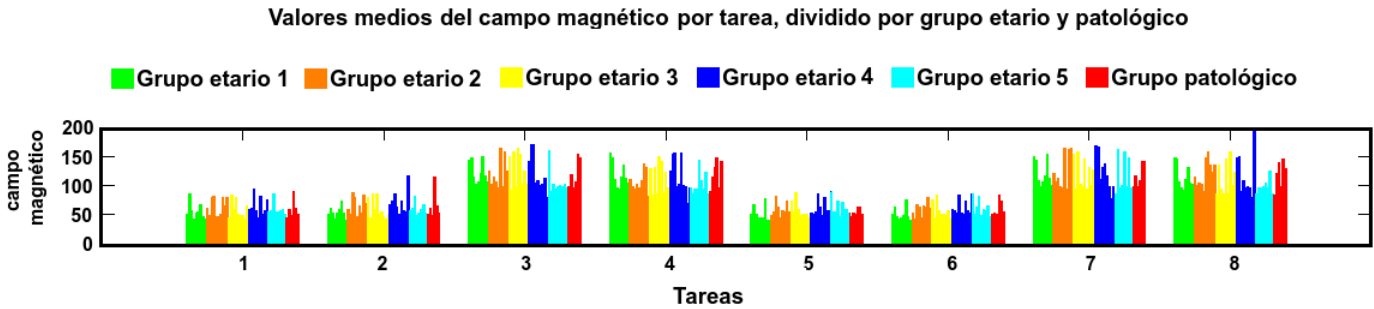


Figura 4.18: Gráfico de barras del valor medio del campo magnético por grupo etario de los sujetos sanos y el grupo patológico.

Tarea	Etario I (μT)	Etario II (μT)	Etario III (μT)	Etario IV (μT)	Etario V (μT)	Patológico (μT)
Tarea 1	55.20	62.02	59.39	63.24	60.46	58.32
Tarea 2	53.93	67.79	60.54	63.57	66.18	65.28
Tarea 3	123.01	121.19	127.19	120.83	104.61	119.67
Tarea 4	118.78	106.05	114.35	117.43	102.17	119.84
Tarea 5	50.88	58.76	57.58	60.96	62.34	54.54
Tarea 6	50.89	59.48	56.60	60.04	61.52	60.91
Tarea 7	120.98	123.78	119.12	125.47	115.27	117.17
Tarea 8	115.25	119.34	113.78	109.83	108.90	119.39

Tabla 4.12: Valores medios del módulo del campo magnético por tarea de los grupos etarios y patológico.

En la figura 4.19 se presenta un diagrama de barras del valor medio de la velocidad angular para cada tarea por grupo etario de los sujetos sanos y el grupo patológico, estos valores se muestran en la tabla 4.13. Se observa una diferencia entre las tareas de movimiento circular (tareas 1,2,5,6) y las tareas de movimiento lineal (3,4,7,8) y a la vez se tiene una desigualdad entre los grupos por tarea. El valor promedio con su desviación estándar por cada tarea es la siguiente: Tarea 1 = $(62 \pm 5)^\circ/\text{s}$, Tarea 2 = $(67 \pm 7)^\circ/\text{s}$, Tarea 3 = $(31 \pm 3)^\circ/\text{s}$, Tarea 4 = $(40 \pm 4)^\circ/\text{s}$, Tarea 5 = $(64 \pm 5)^\circ/\text{s}$, Tarea 6 = $(59 \pm 8)^\circ/\text{s}$, Tarea 7 = $(28 \pm 5)^\circ/\text{s}$ y Tarea 8 = $(37 \pm 5)^\circ/\text{s}$. Los valores de la velocidad angular pueden introducir una previa clasificación inter-tarea, aún así no es suficiente para establecer una clasificación entre los sujetos sanos y patológicos.

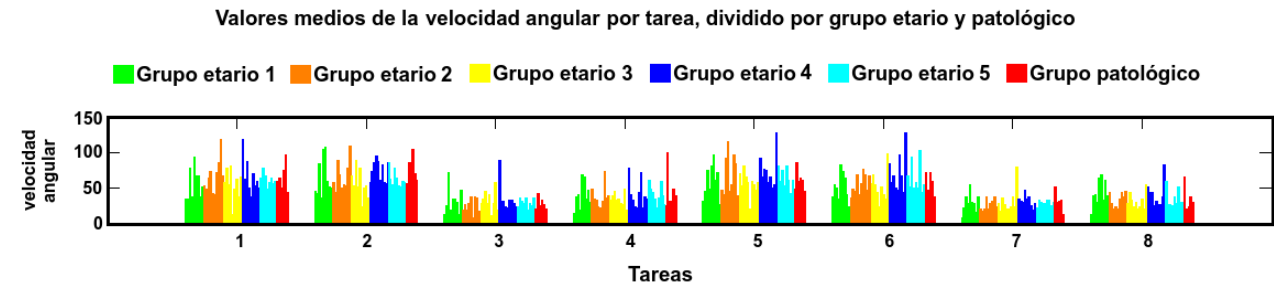


Figura 4.19: Gráfico de barras del valor medio de la velocidad angular por grupo etario de los sujetos sanos y el grupo patológico.

Tarea	Etario I (°/s)	Etario II (°/s)	Etario III (°/s)	Etario IV (°/s)	Etario V (°/s)	Patológico (°/s)
Tarea 1	55.95	66.59	55.78	67.13	60.87	64.70
Tarea 2	64.60	66.99	55.55	72.94	63.12	77.31
Tarea 3	32.21	25.76	33.05	35.29	29.47	28.31
Tarea 4	39.93	38.18	34.27	41.44	42.17	46.05
Tarea 5	59.98	68.45	56.82	66.02	71.24	60.62
Tarea 6	51.32	54.45	49.23	68.13	71.97	57.35
Tarea 7	29.47	26.51	26.16	38.08	24.25	31.86
Tarea 8	43.33	33.35	28.76	38.17	40.93	35.12

Tabla 4.13: Valores medios del módulo de la velocidad angular por tarea de los grupos etarios y patológico.

En la figura 4.20 se presenta un diagrama de barras del valor medio del pulso por grupo etario de los sujetos sanos y el grupo patológico para cada tarea, estos valores se muestran en la tabla 4.14. Se observa muy poca diferencia entre los grupos, el valor promedio con su desviación estándar por cada tarea es la siguiente: Tarea 1 = (1.40 ± 0.08) Hz, Tarea 2 = (1.43 ± 0.07) Hz, Tarea 3 = (1.44 ± 0.06) Hz, Tarea 4 = (1.42 ± 0.11) Hz, Tarea 5 = (1.38 ± 0.11) Hz, Tarea 6 = (1.36 ± 0.13) Hz, Tarea 7 = (1.43 ± 0.08) Hz y Tarea 8 = (1.43 ± 0.08) Hz. Los valores del pulso no son suficientes para establecer una clasificación entre los sujetos sanos y patológicos.

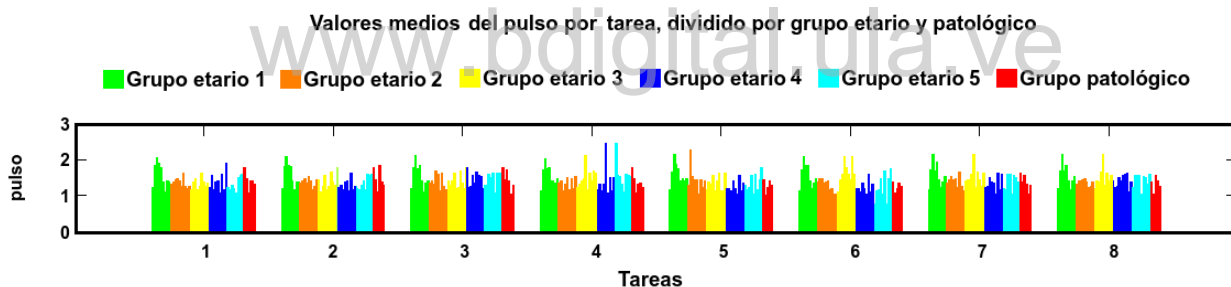


Figura 4.20: Gráfico de barras del valor medio del pulso por grupo etario de los sujetos sanos y el grupo patológico.

Tarea	Etario I (Hz)	Etario II (Hz)	Etario III (Hz)	Etario IV (Hz)	Etario V (Hz)	Patológico (Hz)
Tarea 1	1.56	1.37	1.33	1.32	1.39	1.40
Tarea 2	1.56	1.39	1.45	1.34	1.36	1.47
Tarea 3	1.54	1.38	1.37	1.44	1.47	1.44
Tarea 4	1.56	1.35	1.46	1.27	1.56	1.34
Tarea 5	1.59	1.48	1.31	1.27	1.30	1.35
Tarea 6	1.55	1.31	1.54	1.26	1.18	1.32
Tarea 7	1.57	1.39	1.49	1.35	1.40	1.37
Tarea 8	1.56	1.36	1.53	1.37	1.41	1.37

Tabla 4.14: Valores medios del módulo del pulso por tarea de los grupos etarios y patológico.

Los parámetros aislados no son suficientes para establecer una clasificación entre los sujetos sanos y patológicos, por lo que es necesario introducir una herramienta que realice una clasificación en simultáneo.

4.5.4. Sincronización del movimiento.

En el siguiente apartado se mostrará un análisis cualitativo de la sincronización en la trayectoria de cada una de las tareas realizadas. Este gráfico se crea plotando la componente b_y en función de la componente b_x , este se encuentra dividido en 8 gráficos donde se superponen las siguientes tareas:

- Gráfico 1: Movimiento circular en sentido horario y antihorario de la mano derecha.
- Gráfico 2: Movimiento circular en sentido horario y antihorario de la mano izquierda.
- Gráfico 3: Movimiento circular en sentido horario de la mano derecha e izquierda.
- Gráfico 4: Movimiento circular en sentido antihorario de la mano derecha e izquierda.
- Gráfico 5: Movimiento lineal en sentido vertical y horizontal de la mano derecha.
- Gráfico 6: Movimiento lineal en sentido vertical y horizontal de la mano izquierda.
- Gráfico 7: Movimiento lineal en sentido vertical de la mano derecha e izquierda.
- Gráfico 8: Movimiento lineal en sentido horizontal de la mano derecha e izquierda.

Se observa tanto para los sujetos sanos y los pacientes patológicos la recuperación de la trayectoria de las tareas realizadas, círculos y líneas, en algunos casos como en la figura 4.25 hay una mayor densidad de puntos lo cual está directamente relacionado con la lentitud al ejecutar los movimientos, también se observan zonas que tienen una mayor densidad de puntos con respecto a su otra mitad, como en las figuras 4.22, 4.24, 4.26 y 4.28, esto puede deberse a que la ejecución fue con la mano no dominante. En la mayoría de los casos se ven unos puntos aislados que son *outliers* debidos al protocolo de comunicación entre los equipos.

Finalmente, se esperaría que estos gráficos sean círculos y líneas perfectos, ya que se sigue un patrón por cada tarea, sin embargo como se evidencia en las siguientes figuras desde la 4.21 hasta la 4.28, esta idea no es cierta, ya que la dinámica corporal es diferente en cada persona incluso siguiendo un patrón previamente establecido. Lo que ilustra una complejidad del problema estudiado y justifica la necesidad de aplicar otras herramientas de análisis.

4.5.4.1. Sincronización para los sujetos sanos.

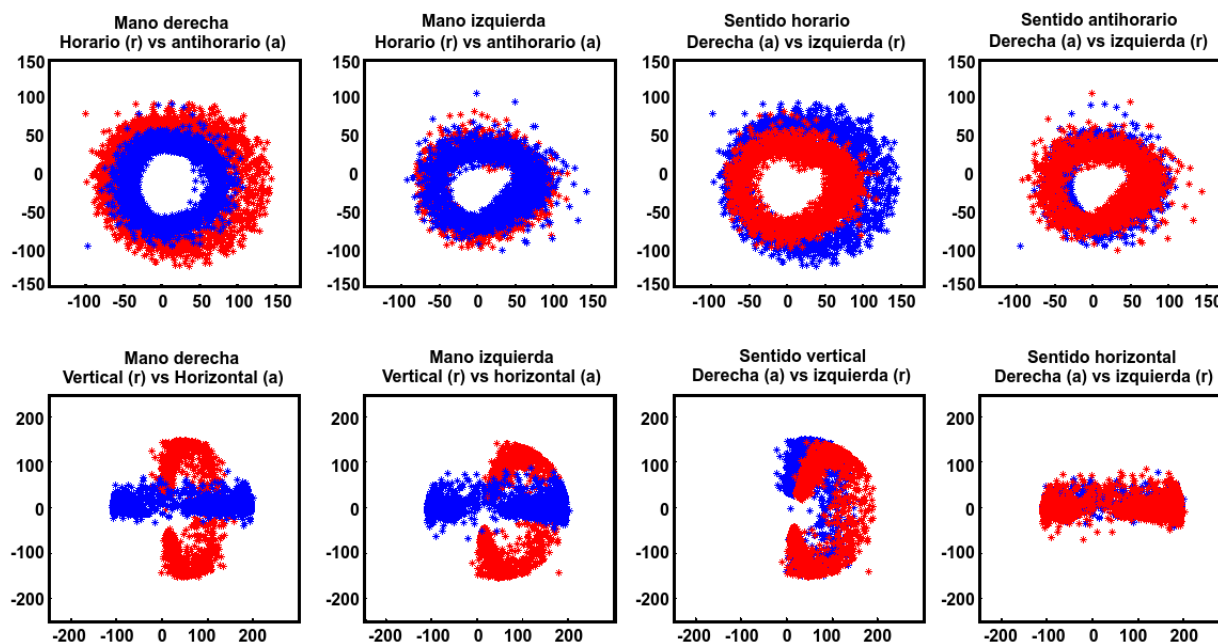


Figura 4.21: Ejemplo de sincronización para el sujeto sano 1, se observa que la ejecución con la mano derecha fue más lenta que con la mano izquierda para el movimiento circular. Mientras que para los movimientos de las líneas se ve mayor regularidad con la mano derecha.

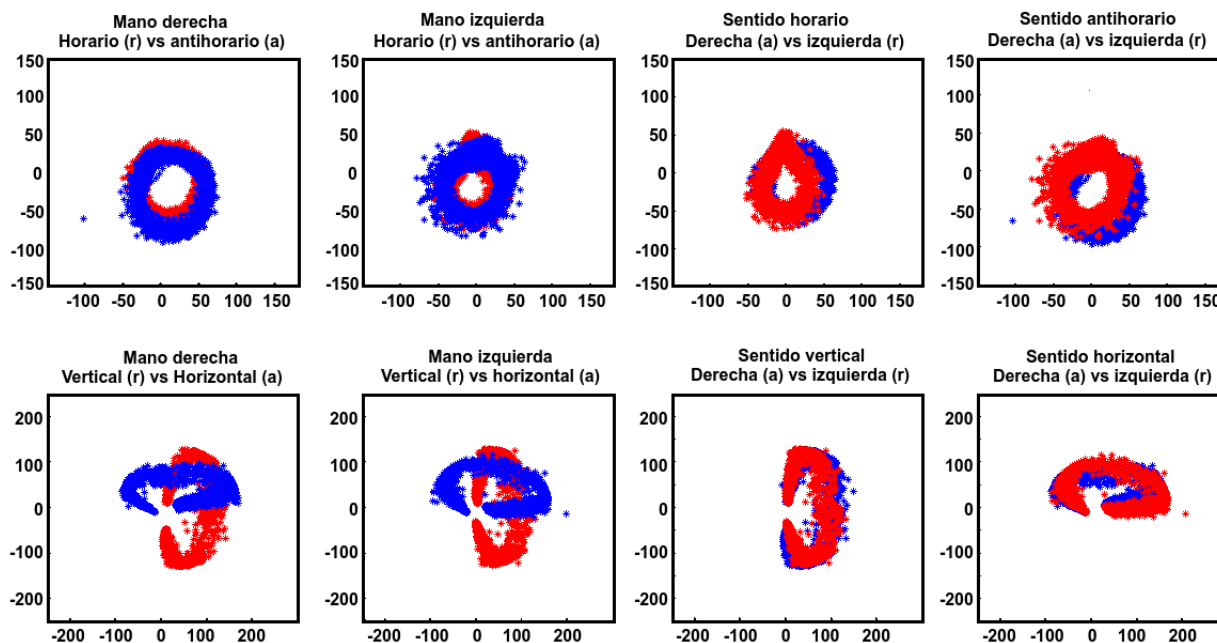


Figura 4.22: Ejemplo de sincronización para el sujeto sano 2, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular. Mientras que para los movimientos de las líneas se ve mayor regularidad en el movimiento vertical.

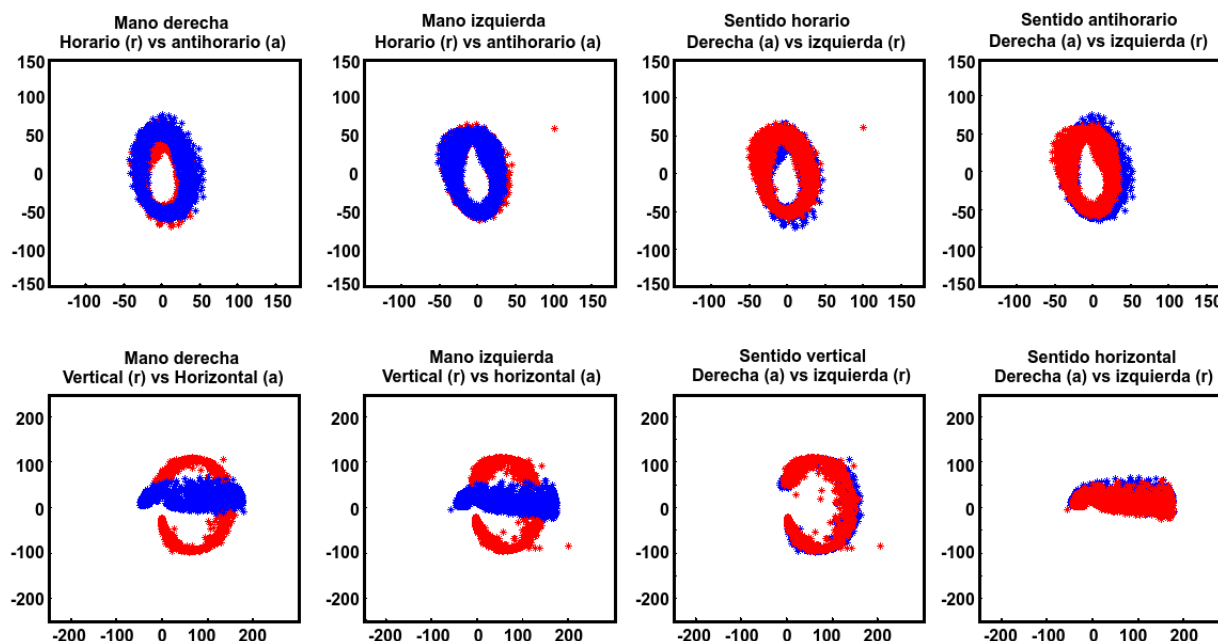


Figura 4.23: Ejemplo de sincronización para el sujeto sano 3, se observa que la ejecución fue regular con ambas manos para el movimiento circular. De igual manera para el movimiento de las líneas, existe una regularidad.

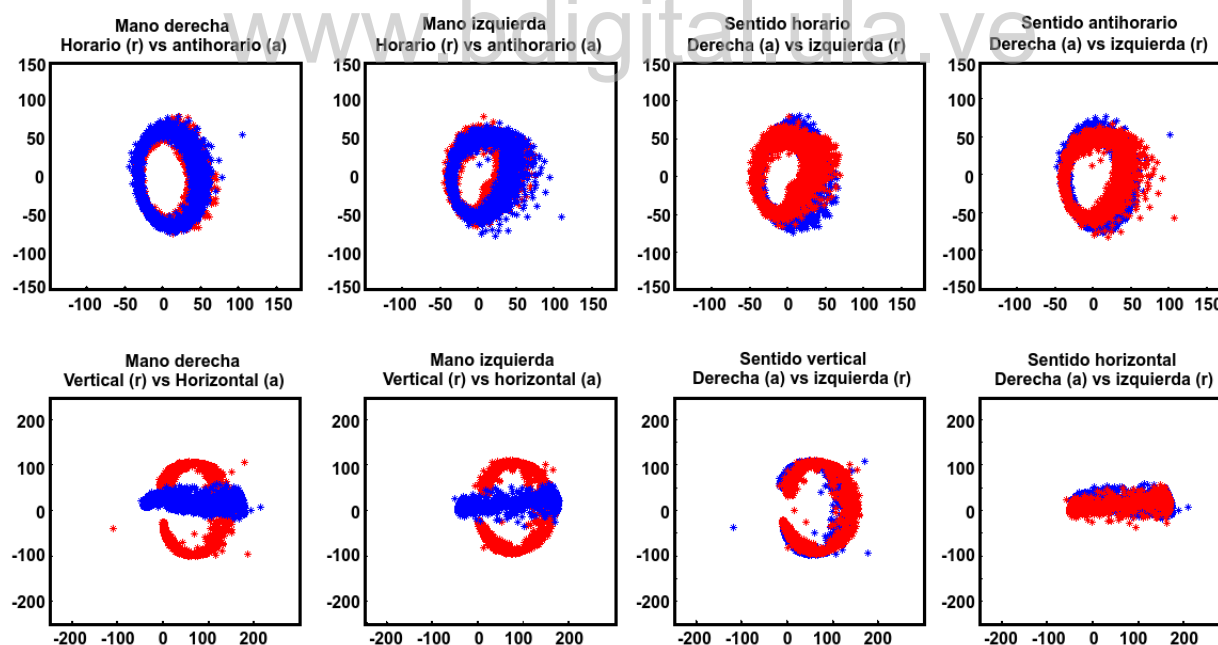


Figura 4.24: Ejemplo de sincronización para el sujeto sano 4, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular. Mientras que para los movimientos de las líneas se ve una regularidad conservada.

4.5.4.2. Sincronización para los pacientes con Parkinson.

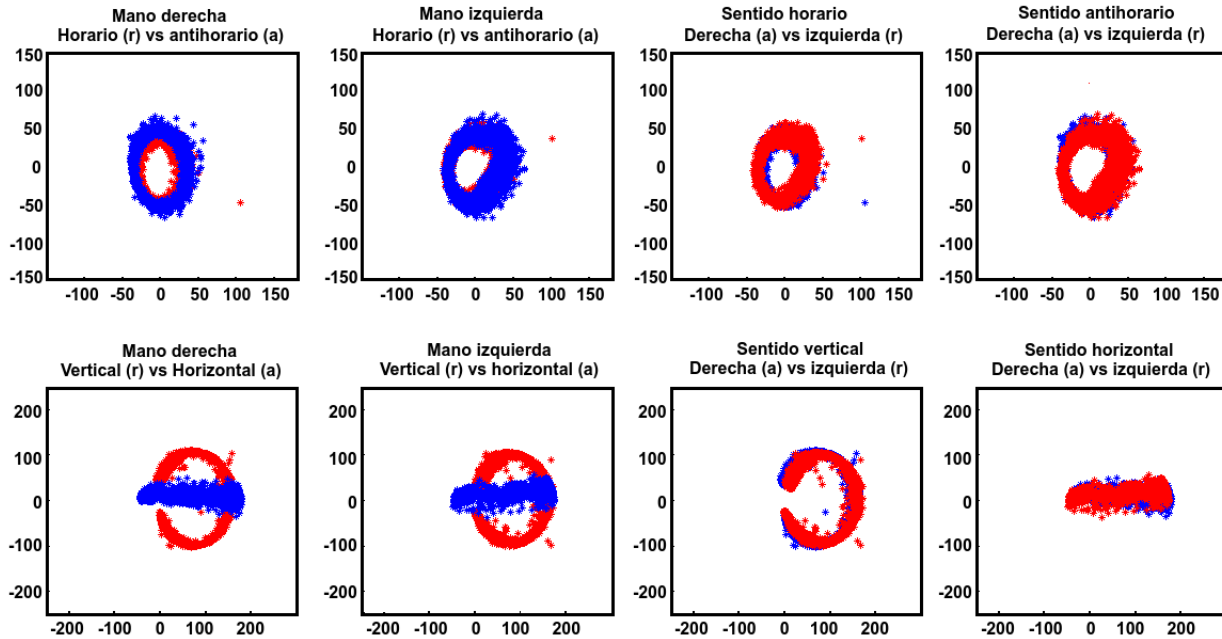


Figura 4.25: Ejemplo de sincronización para el paciente con Parkinson 1, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular. De igual forma para los movimientos de las líneas se mantiene la regularidad.

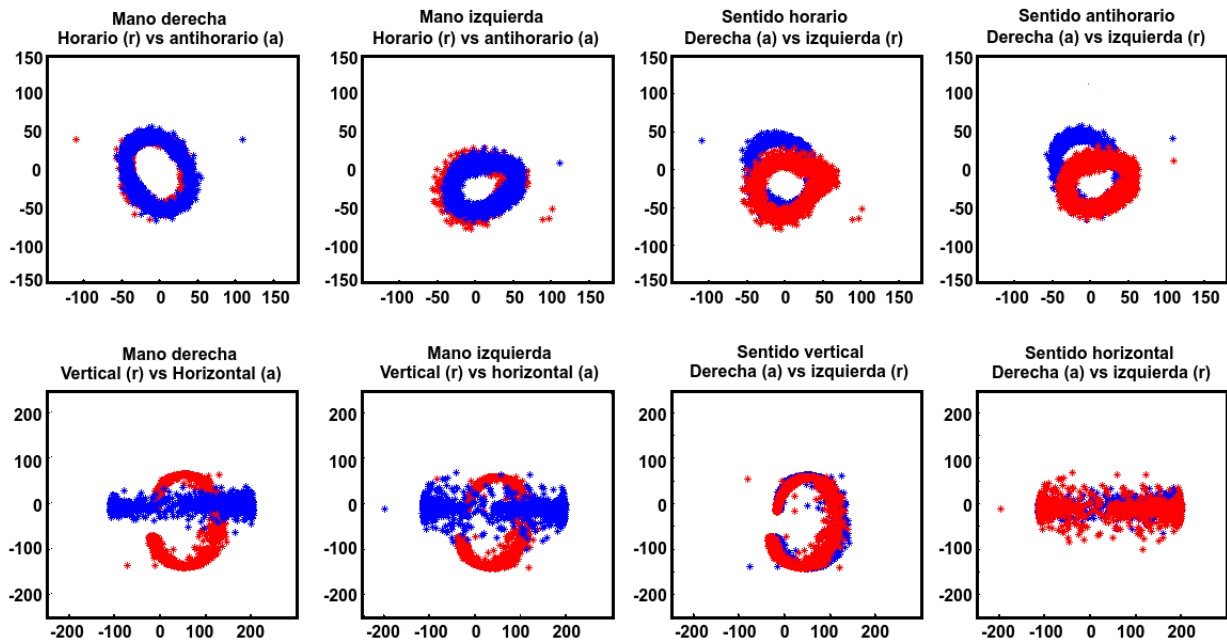


Figura 4.26: Ejemplo de sincronización para el paciente con Parkinson 2, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular. Al superponer esos movimientos se observa un leve desfase. Mientras que para los movimientos lineales se mantiene la regularidad.

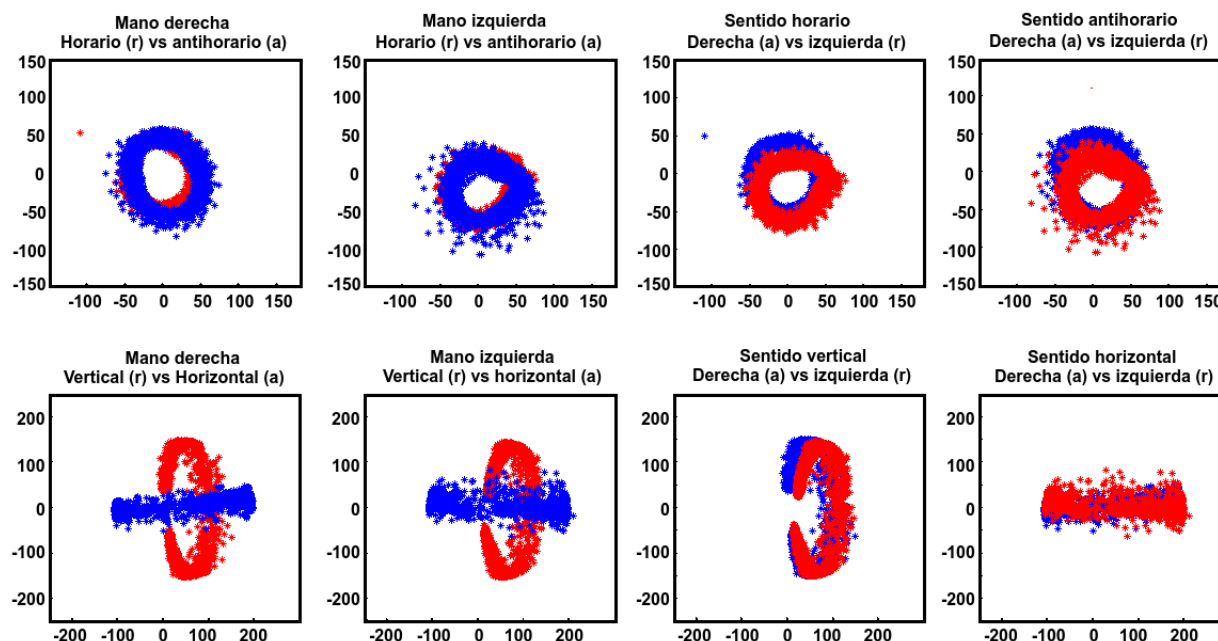


Figura 4.27: Ejemplo de sincronización para el paciente con Parkinson 3, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular, hay un leve desfase al superponer los movimientos horarios y antihorarios. Para los movimientos de las líneas se mantiene la regularidad.

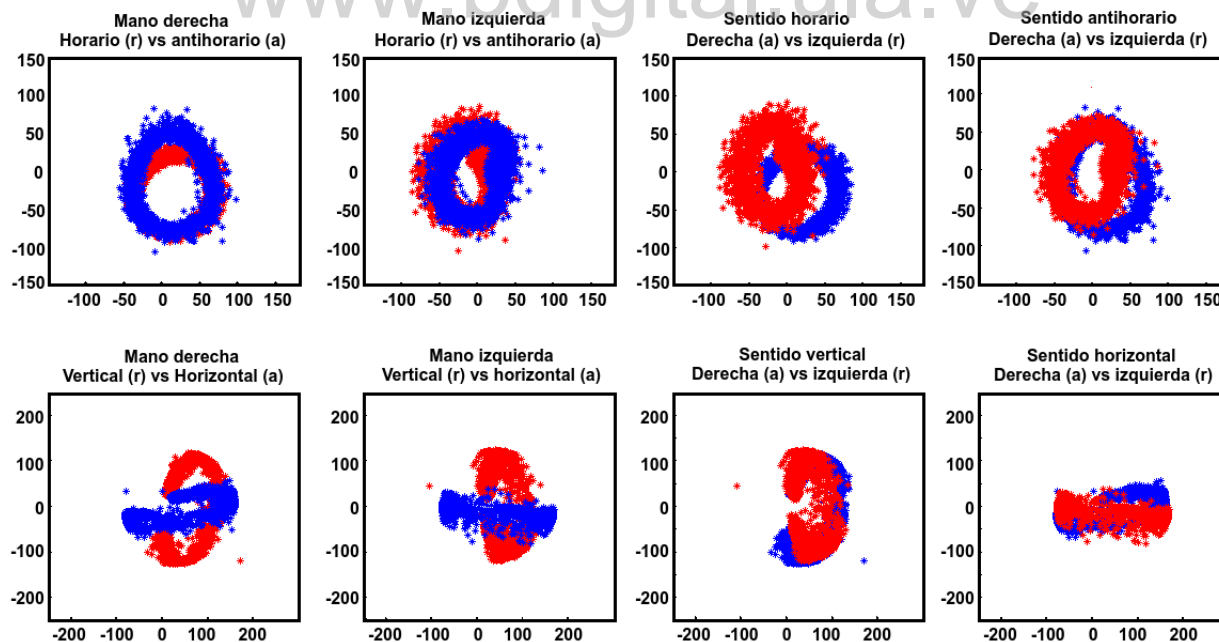


Figura 4.28: Ejemplo de sincronización para el paciente con Parkinson 4, se observa que la ejecución con la mano derecha fue más regular que con la mano izquierda para el movimiento circular, hay un leve desfase al superponer los movimientos horarios y antihorarios. Para los movimientos de las líneas no se mantiene la regularidad y el desfase es notable.

4.5.5. Espectro de Fourier por parámetro para cada tarea.

El análisis de Fourier nos permite determinar la amplitud y la fase de cada una de las componentes de frecuencia que tiene una señal. Se aplica sobre el módulo de cada parámetro y de cada tarea para los sujetos sanos y los pacientes patológicos para ver si existe una composición frecuencial que pueda sugerir diferencias entre ellos a partir de la dinámica corporal. El gráfico representa la frecuencia en el eje de las abscisas y la amplitud en el eje de las ordenadas.

Aceleración.

Para la aceleración se muestra en la figura 4.29 que se dividió por tareas similares para ambas manos. Se tiene que en los movimientos circulares el primer modo de vibración es de 1 Hz y claramente se aprecia un solapamiento entre ambos grupos, la amplitud es igual para ambas tareas, mientras que para los movimientos lineales se tienen varios modos de vibración, manteniéndose un claro solapamiento de los grupos, sin embargo la amplitud es diferente entre las tareas.

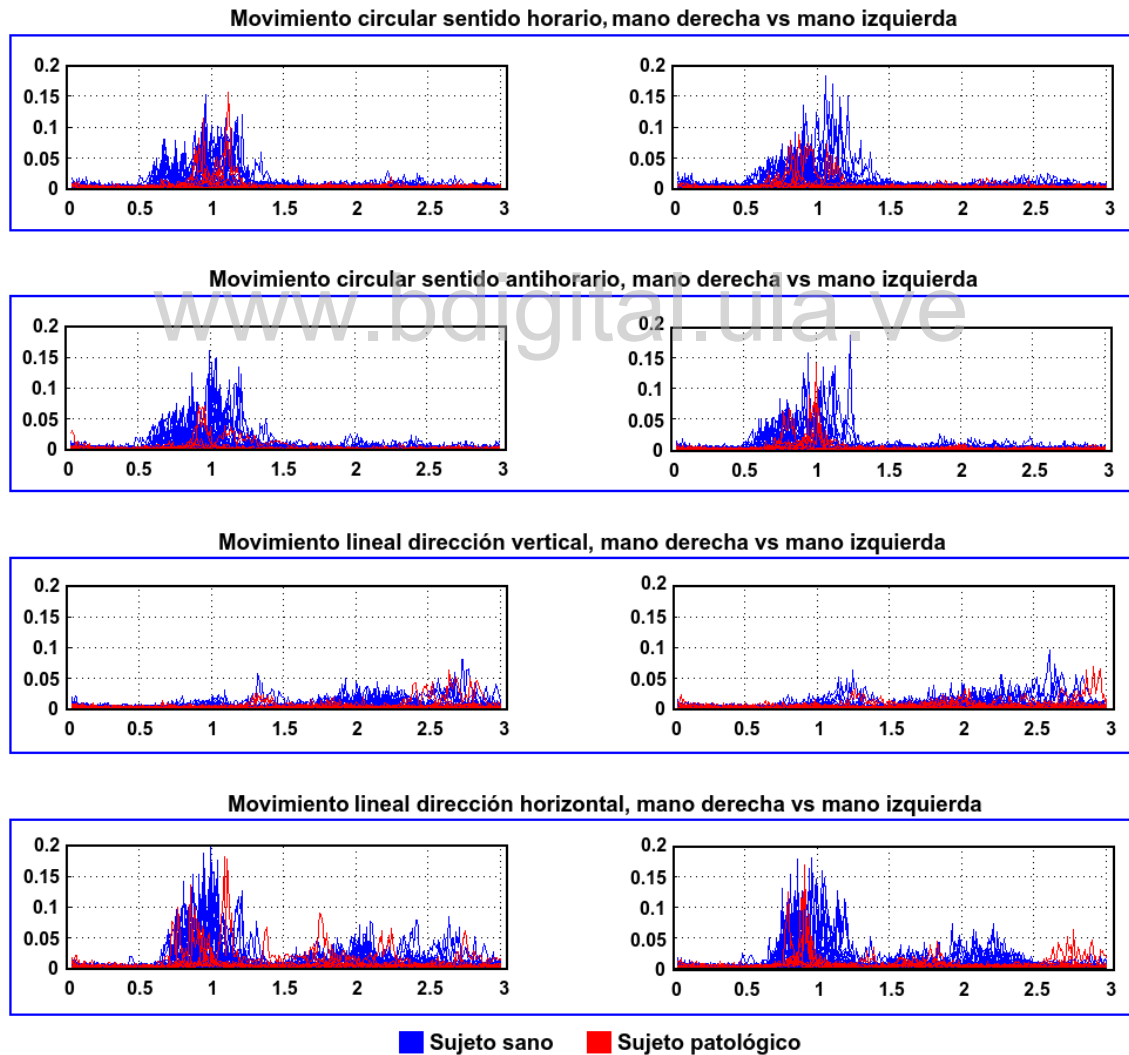


Figura 4.29: Descomposición de la aceleración en componentes de frecuencias, superposición de los sujetos sanos (azul) y los sujetos patológicos (rojo).

Campo magnético.

Para el campo magnético se muestra en la figura 4.30 que también se dividió en tareas similares para ambas manos. Se tiene que en los movimientos circulares el primer modo de vibración es de 1 Hz y claramente se aprecia un solapamiento entre ambos grupos, la amplitud es igual para ambas tareas, mientras que para los movimientos lineales se tienen varios modos de vibración, manteniéndose de igual manera solapamiento de los grupos, en este caso la amplitud es la misma para las tareas.

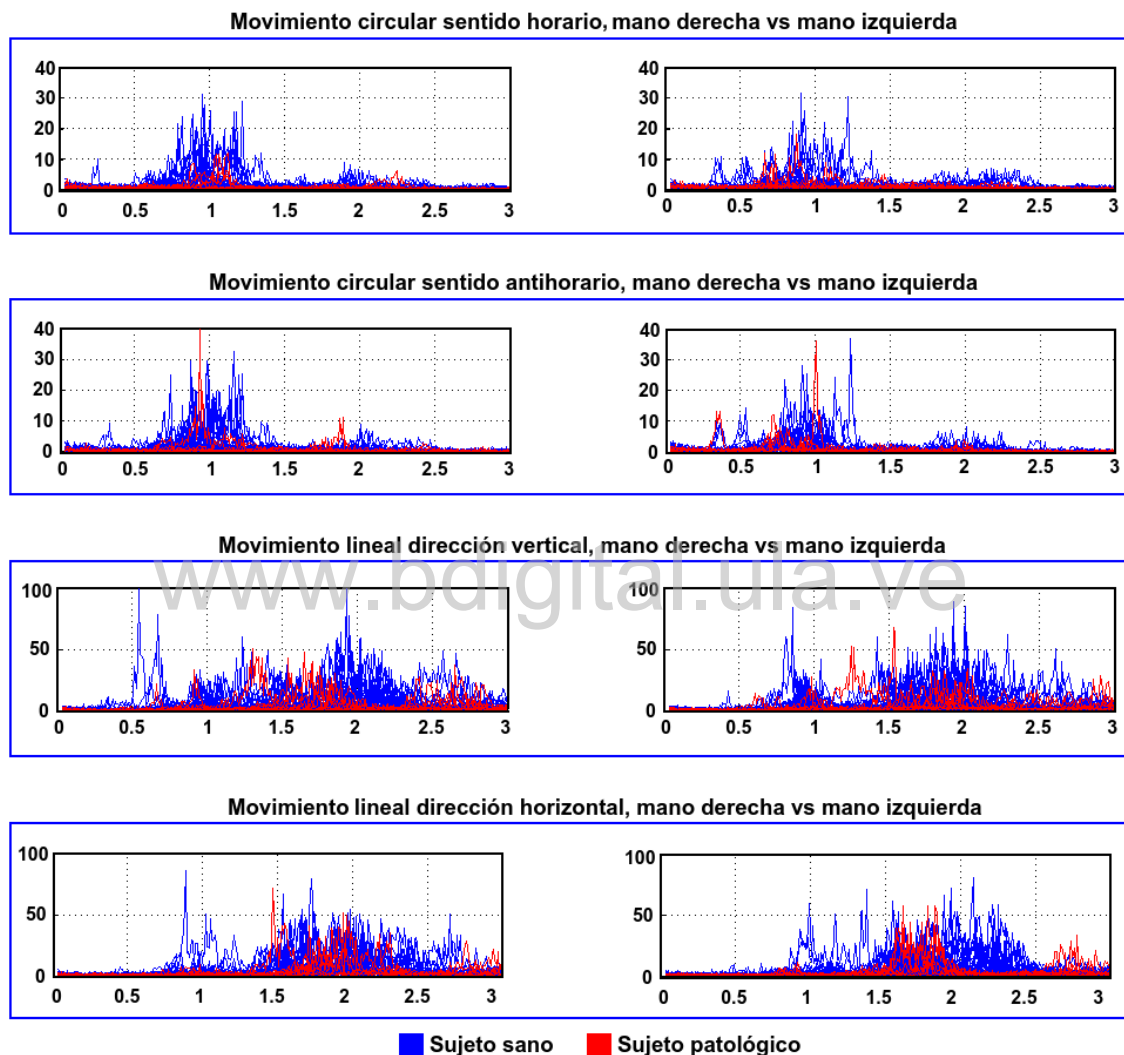


Figura 4.30: Descomposición del campo magnético en componentes de frecuencias, superposición de los sujetos sanos (azul) y los sujetos patológicos (rojo).

Velocidad angular.

Para la velocidad angular se muestra en la figura 4.31 que también se dividió en tareas similares para ambas manos. Se tiene para los movimientos circulares diferentes modos de vibración, claramente se aprecia un solapamiento entre los grupos, la amplitud es igual para todas la tareas, lo mismo ocurre para los movimiento lineales.

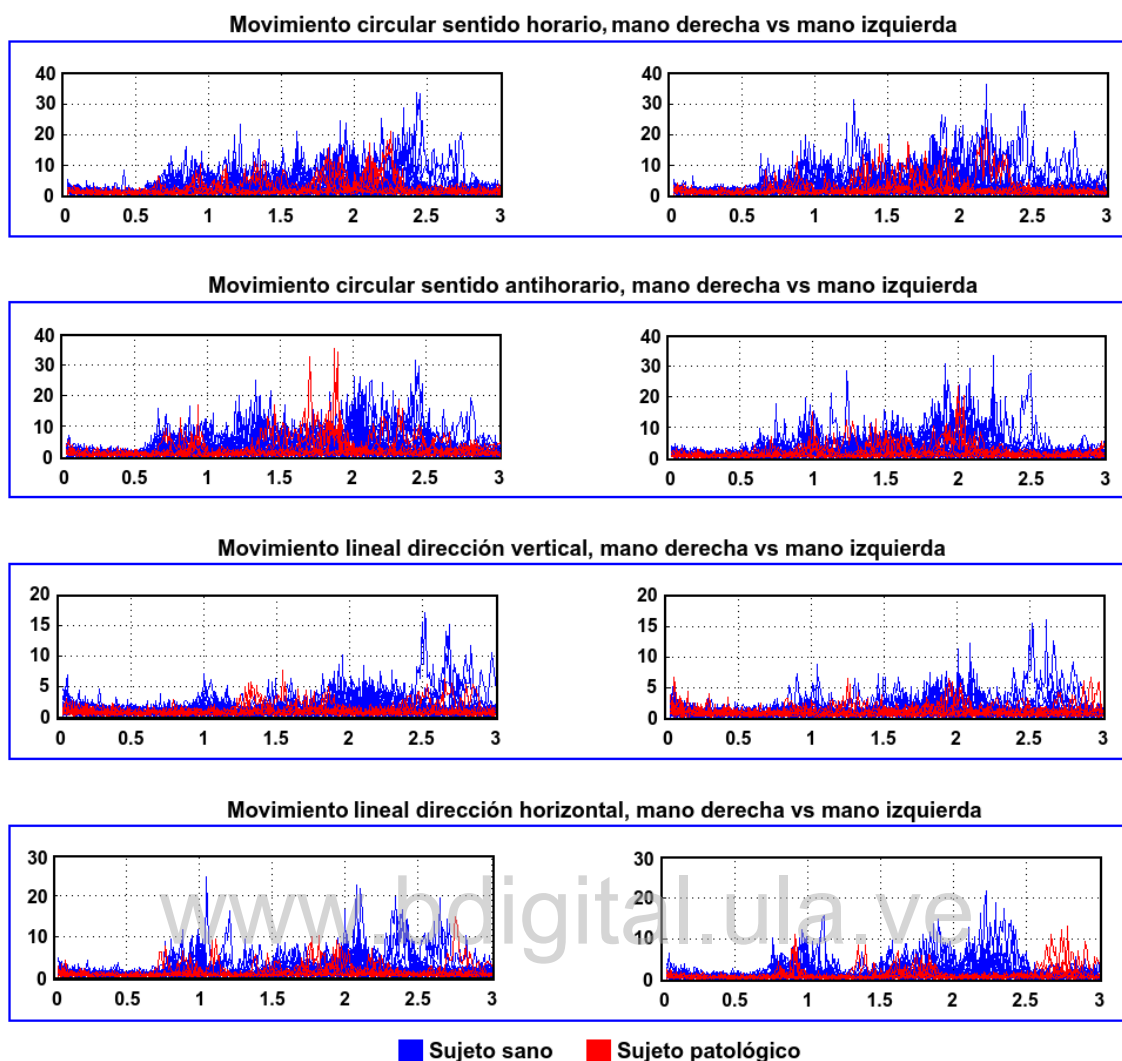


Figura 4.31: Descomposición de la velocidad angular en componentes de frecuencias, superposición de los sujetos sanos (azul) y los sujetos patológicos (rojo).

Pulso cardíaco.

Para el pulso cardíaco se muestra en la figura 4.32 que también se dividió en tareas similares para ambas manos. Se tiene para los movimientos circulares y lineales un primer modo de vibración de 2 Hz y claramente se aprecia un solapamiento entre ambos grupos, la amplitud es igual para todas las tareas.

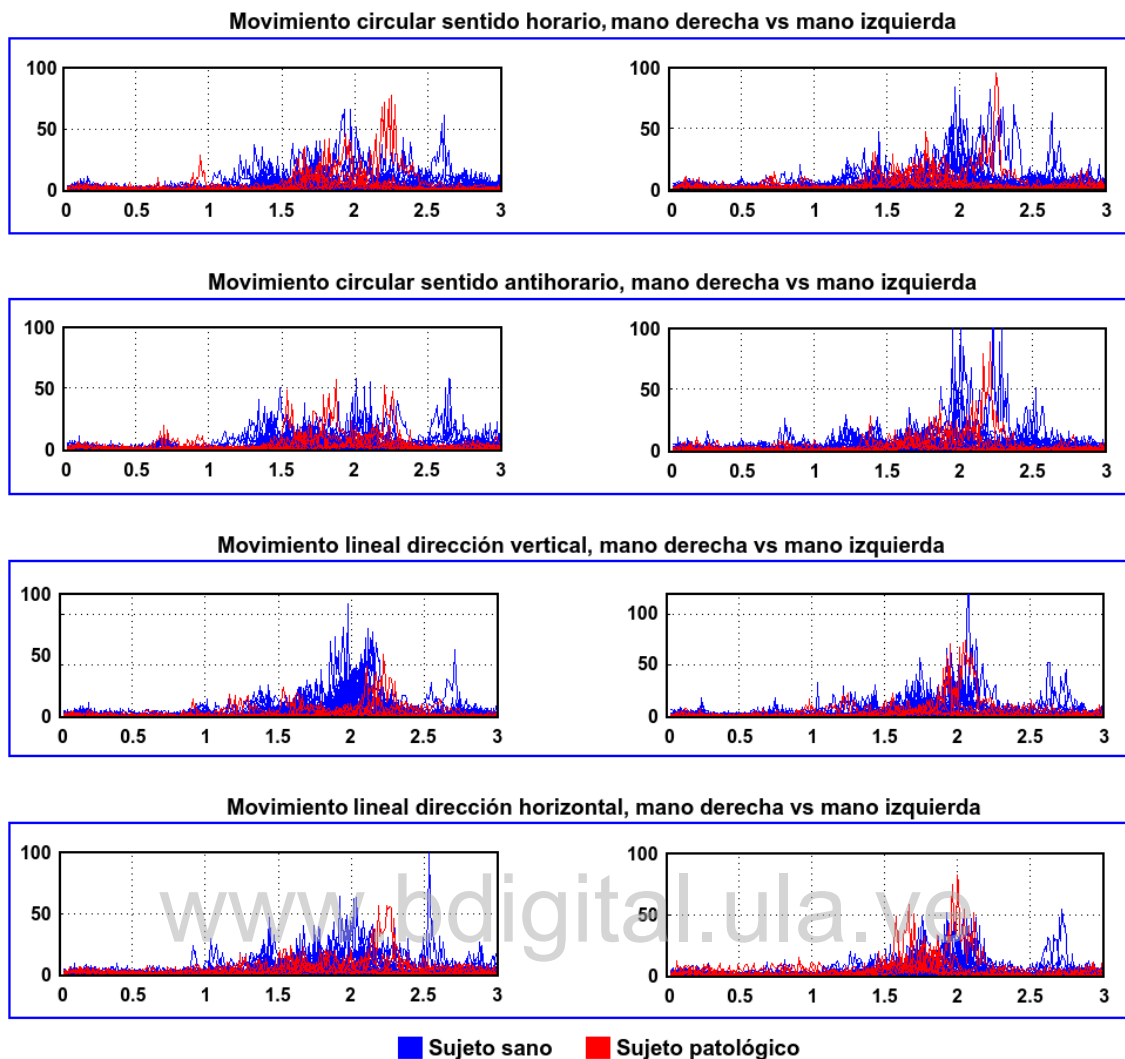


Figura 4.32: Descomposición del pulso en componentes de frecuencias, superposición de los sujetos sanos (azul) y los sujetos patológicos (rojo).

En general la descomposición de los parámetros: aceleración, campo magnético, velocidad angular y pulso, en componentes de frecuencias a través del análisis de Fourier, indica un solapamiento entre los grupos de sujetos sanos y pacientes con Parkinson para todas las tareas, por lo tanto es necesario otro método con el que se pueda hacer una separación de clases entre ellos, un algoritmo de aprendizaje automático.

4.6. Análisis de Componentes Principales (PCA).

PCA es un algoritmo de aprendizaje automático no supervisado que intenta reducir la dimensionalidad (número de características) dentro de un conjunto de datos mientras retiene toda la información posible.

Para el análisis de PCA se empleó el programa “Análisis Componentes Principales.m”, con los siguientes datos:

- Índice óptimo: 15 (todos los parámetros).

- Número de tareas: [1:8] (todas las tareas).
- Número de autovalores: 150.
- Frecuencia de muestreo: 50 Hz.
- Ancho de ventana: 10 s
- Sigma = 3.
- Porcentaje de entrenamiento: 70 %

4.6.1. Gráfico de sedimentación.

El gráfico de sedimentación se obtiene al representar las raíces características en las ordenadas y el número de componentes en las abscisas. Uniendo todos los puntos se obtiene una figura que en general se parece al perfil de una montaña con una fuerte pendiente hasta llegar a la base, conocida como zona de sedimentación. La gráfica de sedimentación ordena los valores propios desde el más grande hasta el más pequeño. Los valores propios de la matriz de correlación son iguales a las varianzas de las componentes principales.

Gráfico de sedimentación para 150 autovalores.

El patrón ideal es una curva pronunciada, seguida de una inflexión y luego una línea recta, como se observa en la figura 4.33. Se han dado diversos criterios para determinar el número de factores a conservar. Uno de los más conocidos y utilizados es el criterio o regla de Kaiser (1960) que indica lo siguiente: “Conservar solamente aquellos factores cuyos valores propios (autovalores) son mayores a la unidad”.



Figura 4.33: Gráfico de sedimentación para 150 autovalores.

Si un autovalor se aproxima a cero, esto significa que el factor correspondiente a ese autovalor es incapaz de explicar una cantidad relevante de la varianza total. Por lo tanto, un factor al que corresponde un autovalor próximo a cero se considera un factor residual y carente de sentido en el análisis.

Es importante resaltar que el gráfico de sedimentación no varía con el número de factores seleccionado, como se muestra a continuación para 15 autovalores.

Gráfico de sedimentación para 15 autovalores.

La gráfica de la figura 4.34 muestra los primeros 15 autovalores, a pesar de que la magnitud de estos es alta, estos empiezan a formar una línea recta después de la undécima componente principal, estos representan aproximadamente el 80 % de la información, por lo tanto los restantes explican una proporción pequeña de la variabilidad.

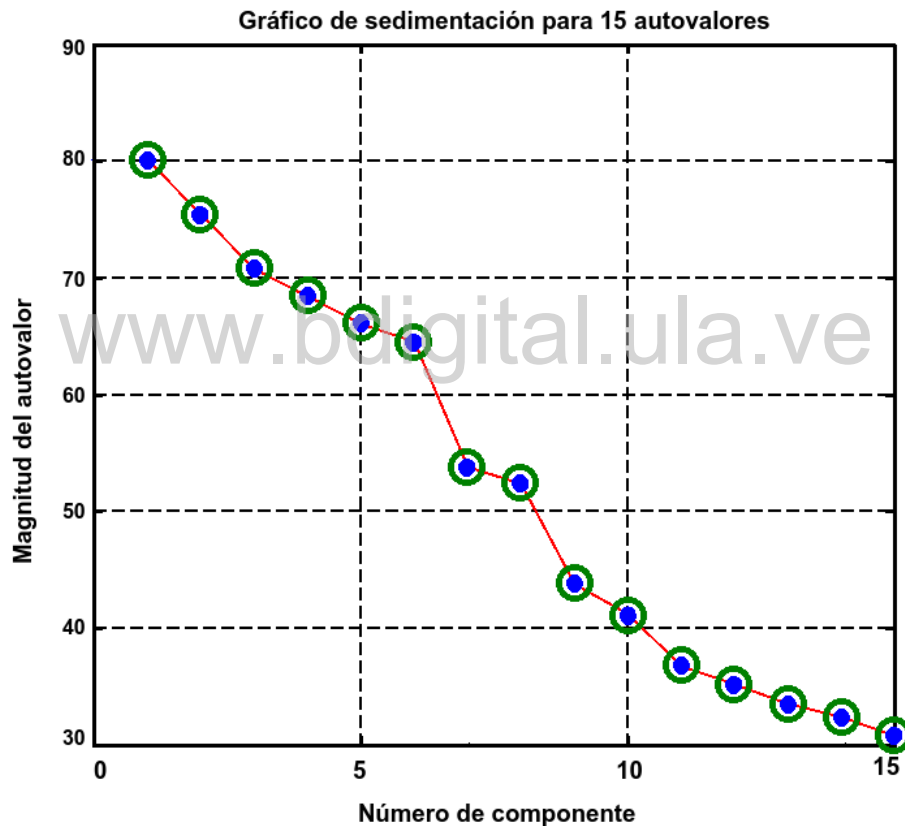


Figura 4.34: Gráfico de sedimentación para 15 autovalores, a partir de la undécima componente se empieza a formar una línea recta.

4.6.2. Proyección de la imagen de cada sujeto sano sobre el espacio V.

Una vez calculados los autovalores y autovectores de la matriz de covarianza, se hace la proyección de cada paciente sobre el espacio de autovectores V, para este caso este está compuesto por 150 elementos. La proyección crea una imagen única de cada paciente, una “llave” que lo identifica del resto. En la figura 4.35 se muestra la imagen proyectada de cada sujeto sano sobre el espacio V, el eje x corresponde al número de los autovalores mientras que en el eje y se encuentran los valores

de la proyección P.

Esta identificación es necesaria para la etapa de reconocimiento, además en este punto se le agrega una etiqueta en número binarios a cada sujeto que será su identificación para la creación de la base de datos.

Proyección de la imagen media de cada sujeto sano sobre el espacio V

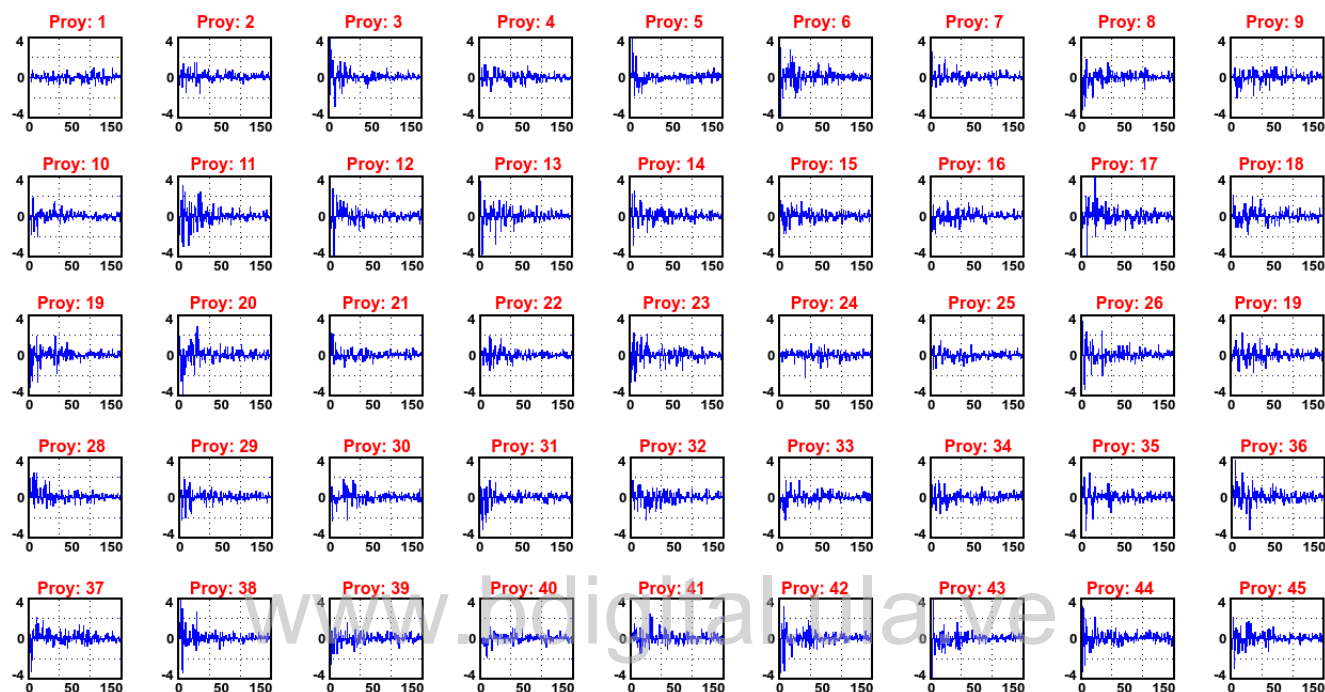


Figura 4.35: Proyección de la imagen media de cada paciente sobre el espacio V.

4.6.3. Correlación entre los sujetos prueba.

Para poder estudiar la relación lineal entre los sujetos se encuentra el grado de relación con respecto al grupo, a manera de ver quien es el sujeto más parecido. En la figura 4.36 se muestra la correlación que tiene cada sujeto con respecto a los 44 restantes. El eje x corresponde al número de sujetos, mientras que el eje y indica el grado de correlación, donde 1 indica una gran correlación. Para cada caso se presenta en rojo el sujeto estudiado y a la vez se muestra en la parte superior del gráfico cual es el sujeto más parecido.

En la figura 4.37 se amplía un cuadrante donde se aprecia lo dicho anteriormente, para este caso se estudia la correlación existente entre el sujeto 45 con respecto a los demás, se observa que no existe otro sujeto que tenga una correlación fuerte con él ya que ninguno tiene un valor superior de 0.5, según indica la tabla 3.6, solo hay dos sujetos con correlación moderada con valores de 0.3, los cuales son los sujetos 8 y 43.

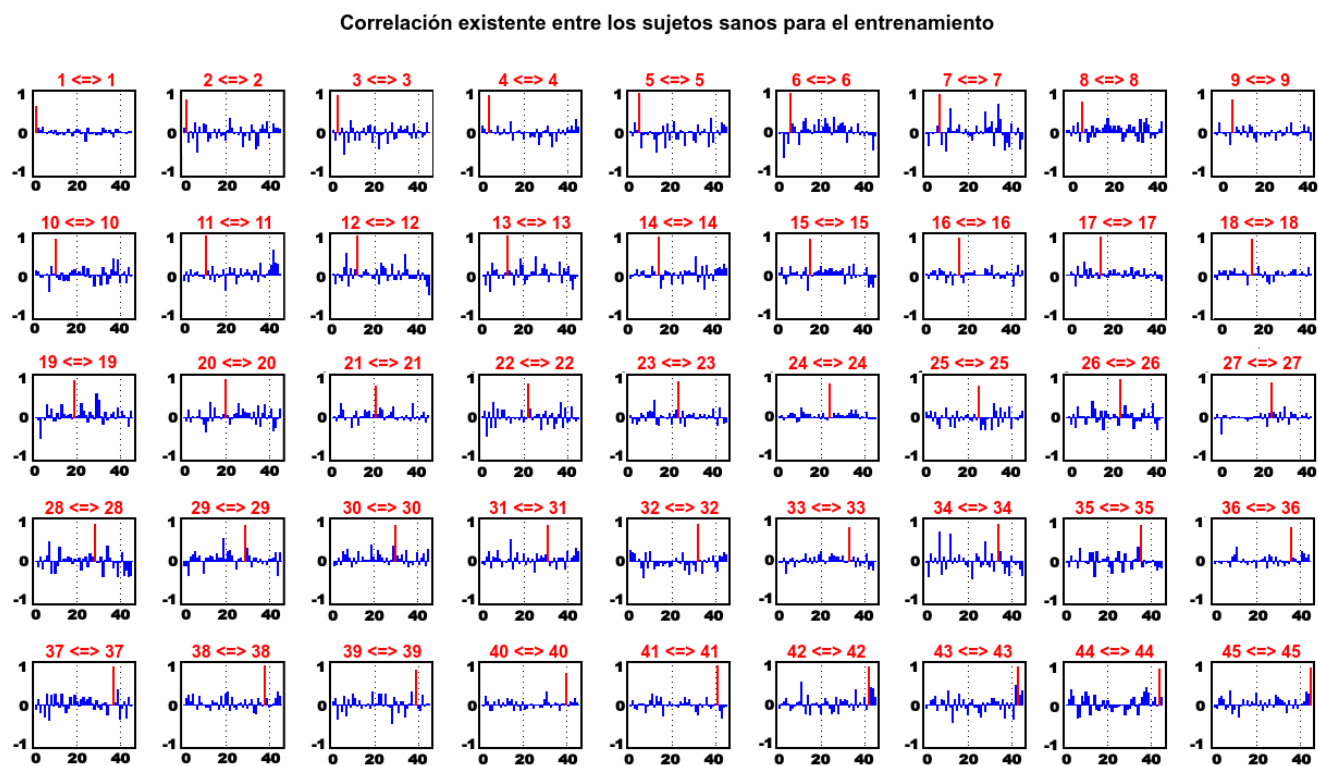


Figura 4.36: Correlación entre los sujetos pruebas con su respectiva clasificación.

www.bdigital.ula.ve

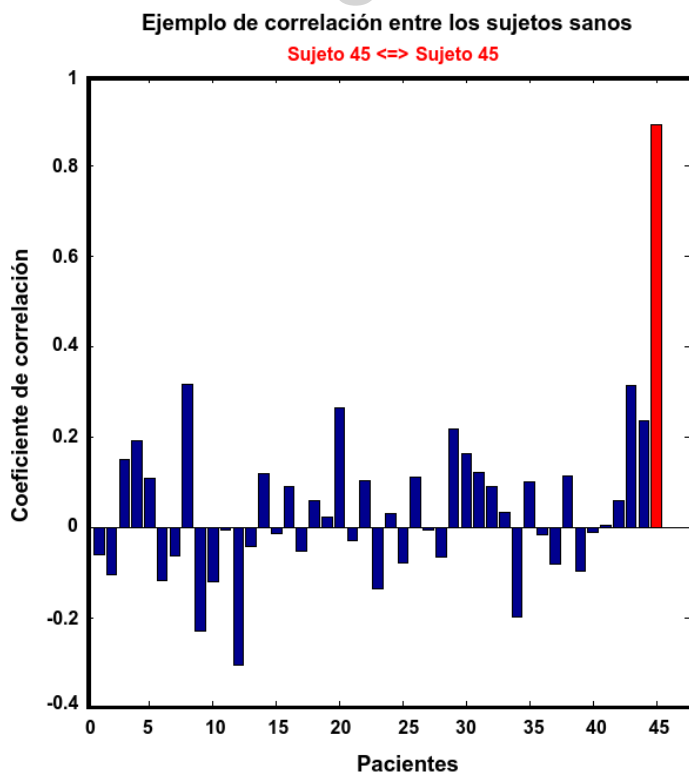


Figura 4.37: Ejemplo de correlación para un sujeto con respecto al grupo.

4.6.4. Matriz de confusión de PCA.

El conjunto original de datos fue dividido en dos partes. Se dejó un 70 % de los datos para el entrenamiento y se reservó el 30 % restante para el test. Se presenta 5 veces los datos del conjunto test de manera aleatoria y se calcula la correlación existente entre los datos. Gracias a que estos están etiquetados se hace una búsqueda entre las etiquetas dadas y las etiquetas encontradas, se gráfica una en contra de la otra y se obtienen dos diagonales: la diagonal principal representa los valores estimados de forma correcta por el modelo. Se observa en la figura 4.38 que la diagonal principal tiene una pendiente $m = 0.72$, lo que representa una eficiencia del 72 %, la otra diagonal no coincide con la diagonal principal, indicando que hubo confusión entre algunos los sujetos y da una idea de que las variables empleadas son suficientes para poder distinguir entre los sujetos a pesar de que el estudio de la dinámica concluyó que no había diferencias marcadas entre ellos, sin embargo se requiere otro método que logre una mejor clasificación para ello se propone el uso de redes neuronales artificiales.

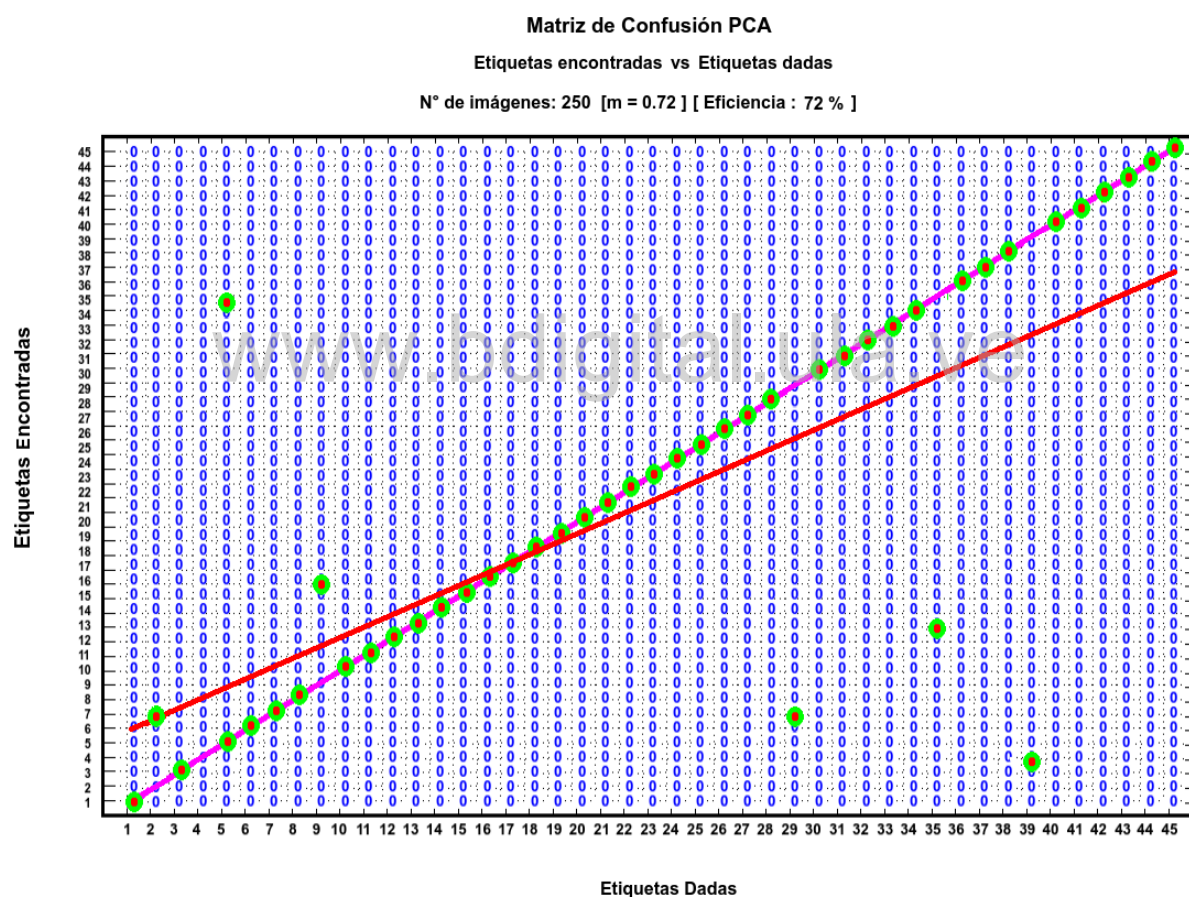


Figura 4.38: Matriz de confusión de PCA para los 45 sujetos de entrenamiento.

Con los resultados obtenidos para la matriz de confusión para PCA, se construye la base de conocimiento, la cual contiene los patrones de entrenamiento y los de test, en conjunto con las etiquetas.

4.7. Entrenamiento de la red neuronal.

4.7.1. Arquitectura de la red.

- La red tiene 4 capas con el siguiente número de neuronas por capa:
 - $N_1 = 25$ neuronas.
 - $N_2 = 24$ neuronas.
 - $N_3 = 23$ neuronas.
 - $N_4 = 22$ neuronas.
- Rata de aprendizaje: $lr = 0,0225$.
- Criterio de error para detener el entrenamiento: $erro_goal = 10^{-4}$.
- Epoch máximo para el entrenamiento = 500 epoches.

Se inicia el entrenamiento de la red neuronal el cual fue explicado en la sección 3.5.2, una vez finalizada la etapa de test como se muestra en la figura 4.39, la red está entrenada. Una vez entrenada se inicia la fase de producción donde se realiza el proceso de propagación hacia adelante y se obtiene la función de la red la cual contiene los pesos y bias. Con esta información se construye la matriz de confusión siguiendo los mismos pasos que se emplearon para la matriz de confusión de PCA.

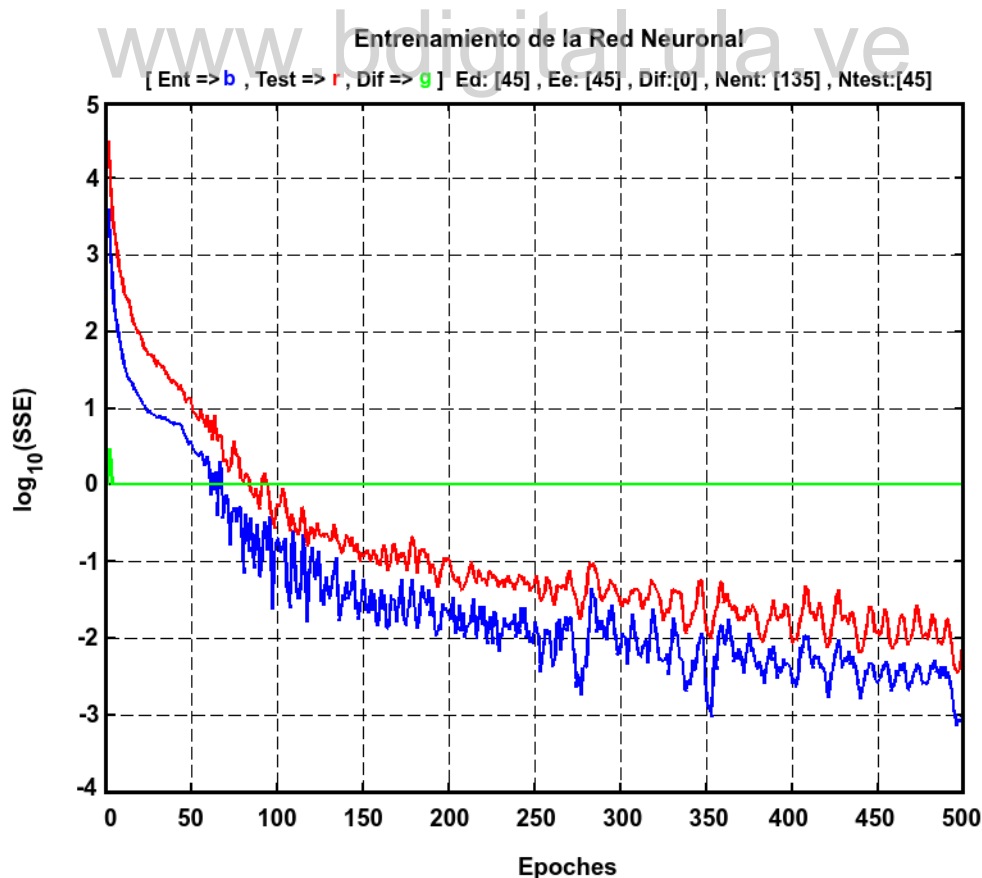


Figura 4.39: Entrenamiento de la red neuronal con 500 epoches

4.7.2. Matriz de confusión de RNA.

La matriz de confusión para RNA se muestra en la figura 4.40, de igual manera que para el caso de PCA, la diagonal principal representa los valores estimados de forma correcta por el modelo, esta tiene una pendiente $m = 1$, lo que representa una eficiencia del 100 %, la otra diagonal coincide con la diagonal principal, indicando que no hubo confusión entre los sujetos

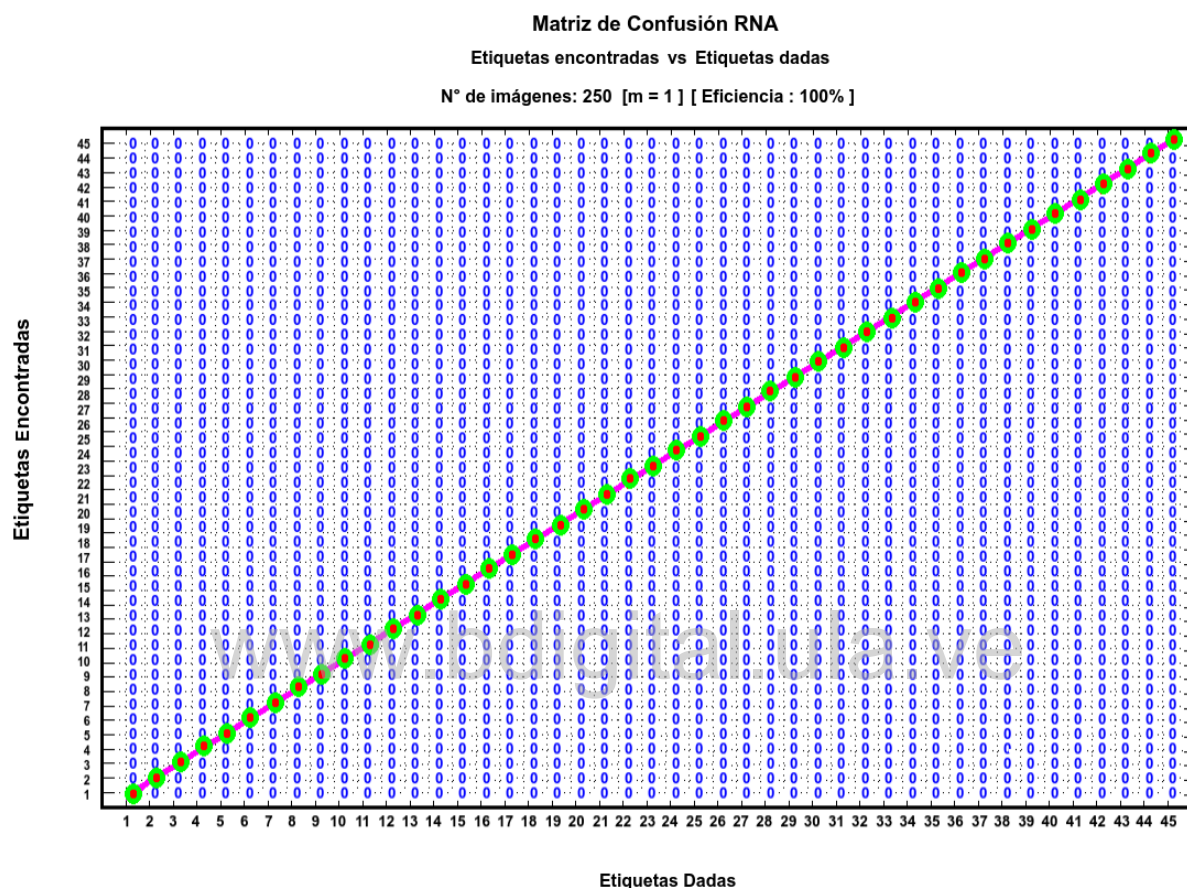


Figura 4.40: Matriz de confusión de RNA para los 45 sujetos de entrenamiento

A pesar de que en el análisis dinámico se encontró que no hay distinción entre los sujetos sanos y patológicos por tarea, se quiso verificar si con las redes neuronales sucede lo mismo. Para ello se realizó el mismo procedimiento desde la creación de la base de conocimiento hasta la obtención de la función de la red por cada tarea. A partir de ahora se analizará cada caso con su respectivo dendrograma, el cual es una técnica de visualización del tipo diagrama de árbol que muestra los grupos que se forman al crear conglomerados (conjuntos homogéneos) y su nivel de similitud. Este nivel se medirá en el eje horizontal empleando la distancia de Mahalanobis que indica el grado de separación en términos de desviación estándar.

Finalmente se quiere ver la clasificación en un diagrama bidimensional que esté ordenado de menor a mayor según la variable edad. Este se construirá a partir de la matriz de distancias para cada sujeto, utilizando el programa “**Clasificacion Parkinson.m**” que se encuentra en el apéndice A.15.

4.8. Clasificación de todos los pacientes por tareas.

4.8.1. Dendrograma para la tarea 1.

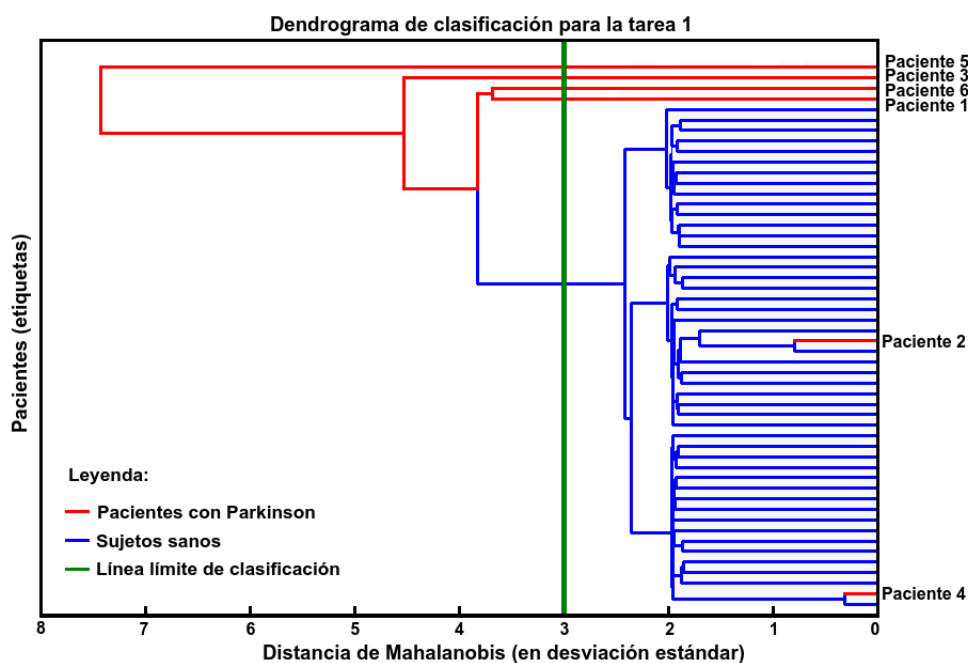


Figura 4.41: Dendrograma de clasificación para la tarea movimiento circular en sentido horario con la mano derecha (tarea 1).

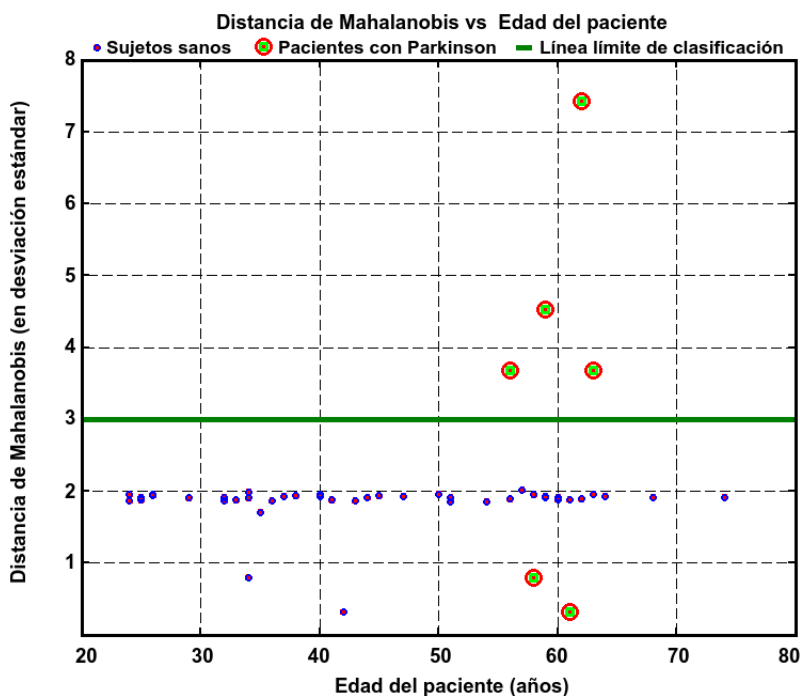


Figura 4.42: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 1.

4.8.2. Dendrograma para la tarea 2.

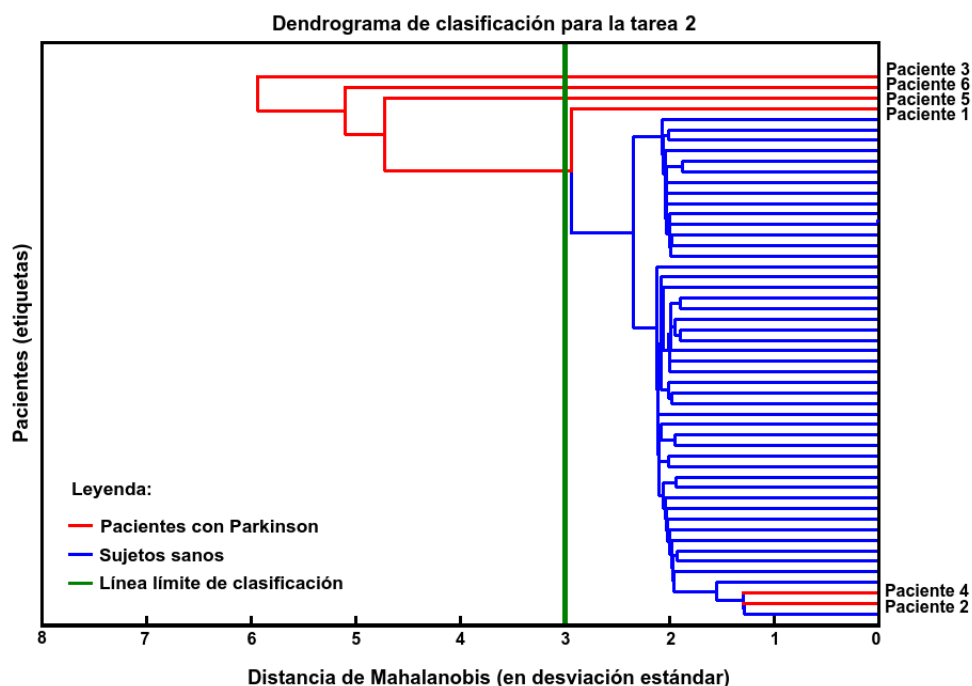


Figura 4.43: Dendrograma de clasificación para la tarea movimiento circular en sentido antihorario con la mano derecha (tarea 2).

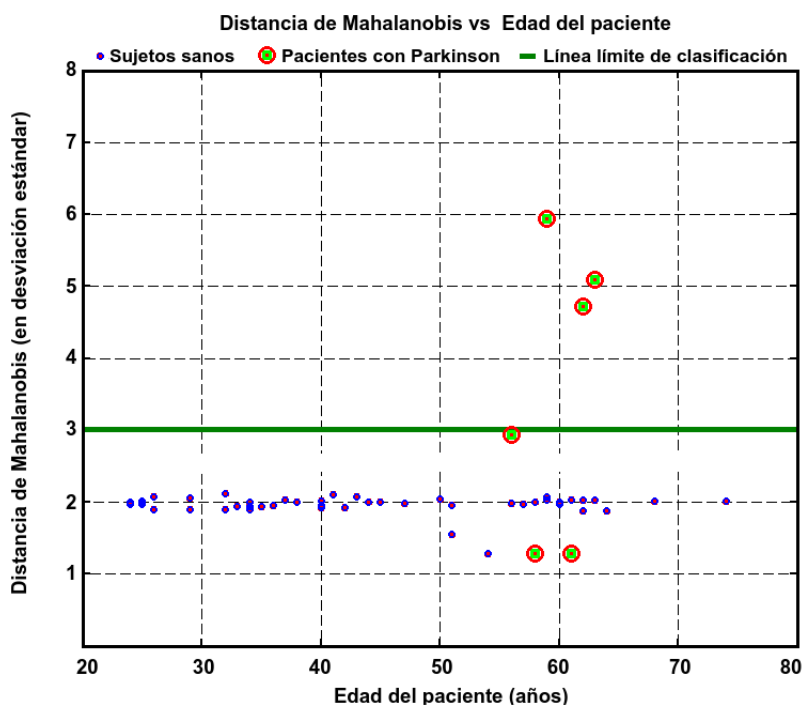


Figura 4.44: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 2.

4.8.3. Dendrograma para la tarea 3.

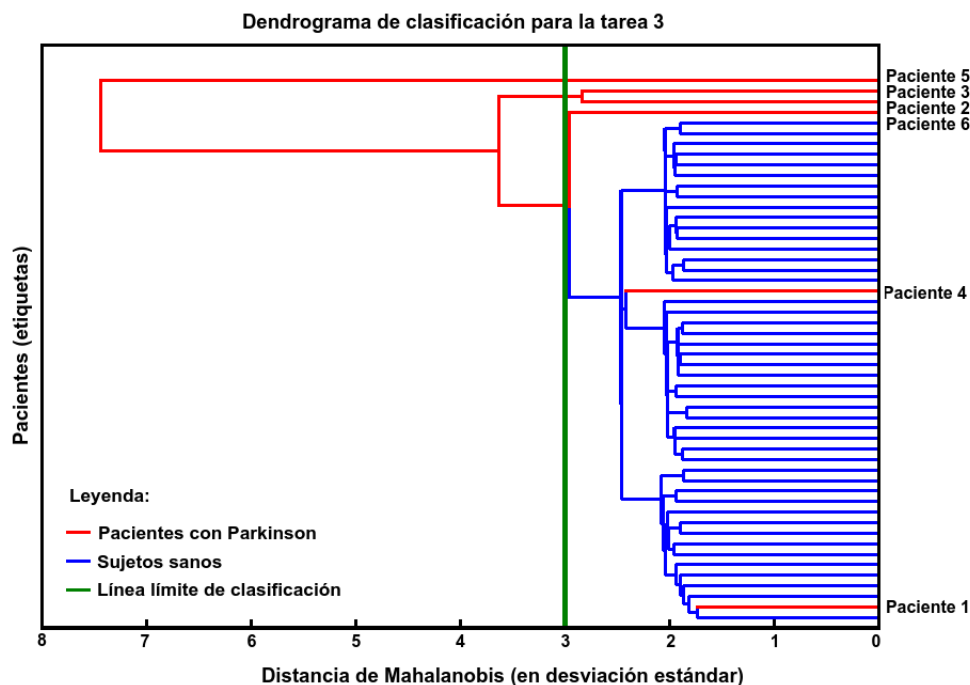


Figura 4.45: Dendrograma de clasificación para la tarea movimiento lineal en sentido vertical con la mano derecha (tarea 3).

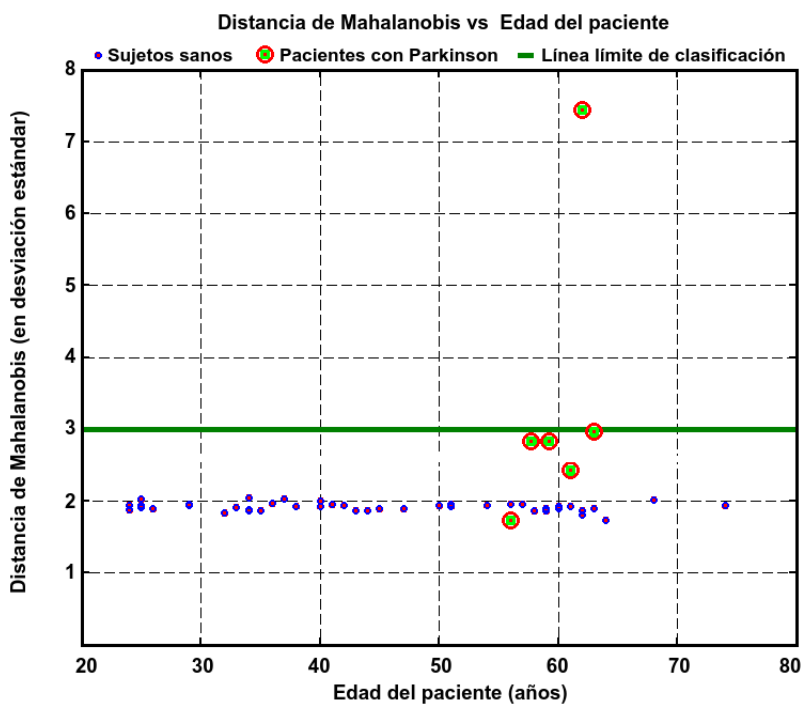


Figura 4.46: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 3.

4.8.4. Dendrograma para la tarea 4.

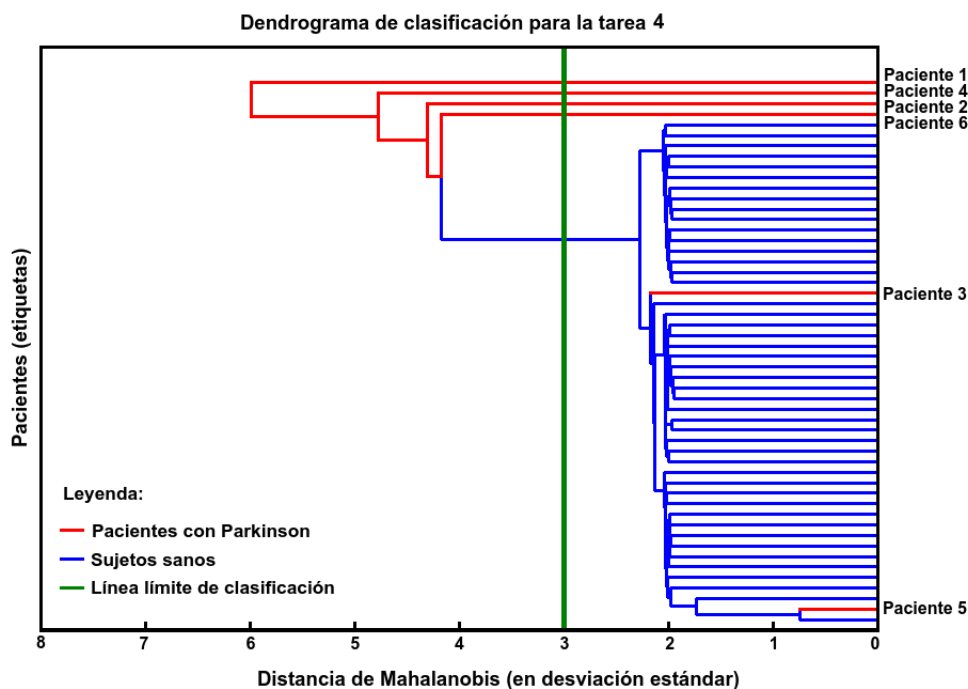


Figura 4.47: Dendrograma de clasificación para la tarea movimiento lineal en sentido horizontal con la mano derecha (tarea 4).

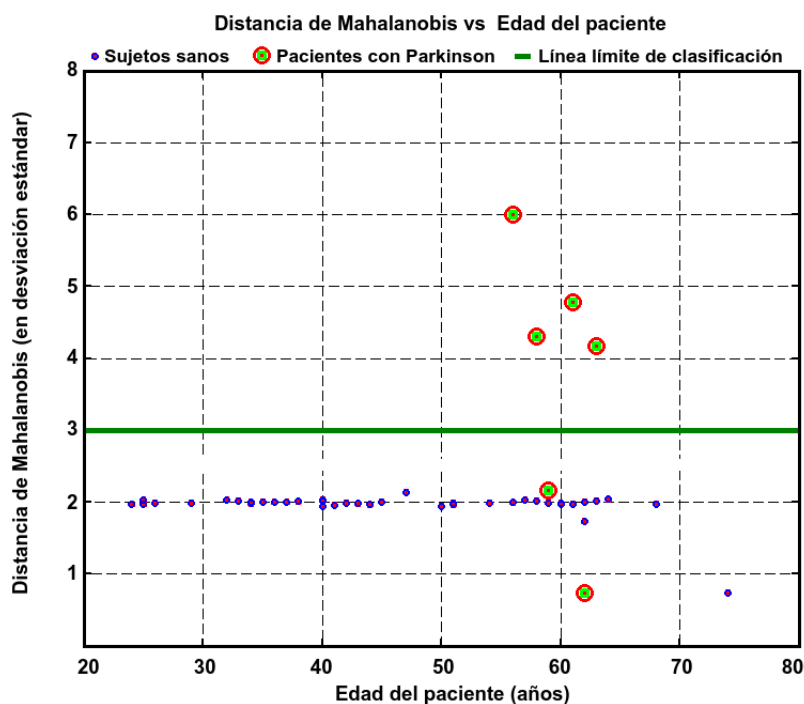


Figura 4.48: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 4.

4.8.5. Dendrograma para la tarea 5.

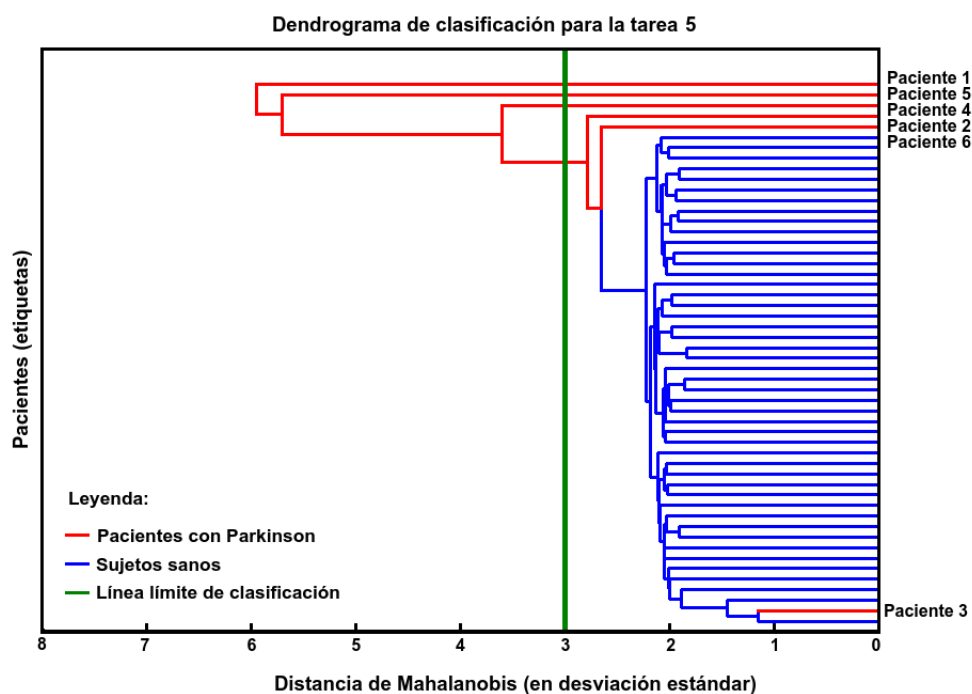


Figura 4.49: Dendrograma de clasificación para la tarea movimiento circular en sentido horario con la mano izquierda (tarea 5).

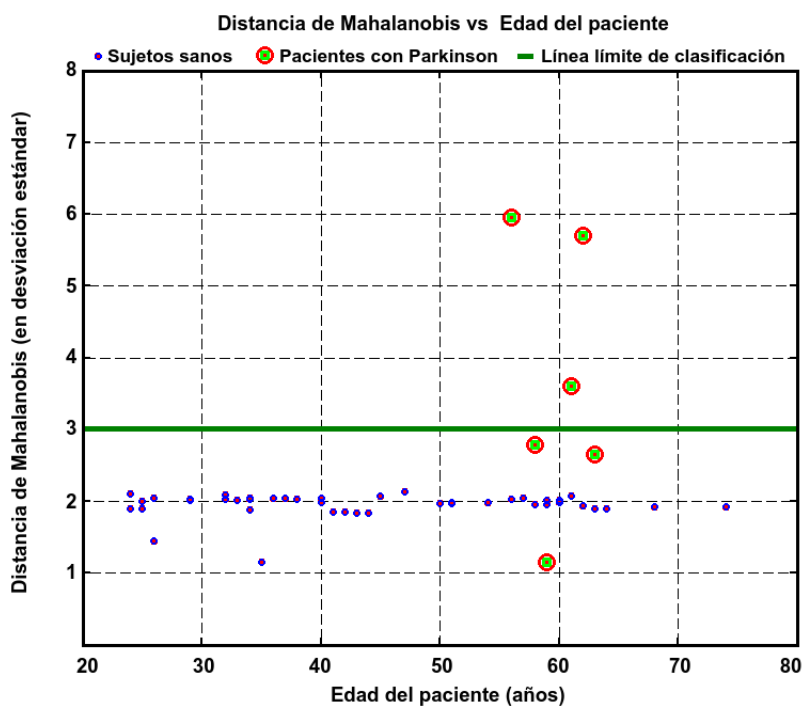


Figura 4.50: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 5.

4.8.6. Dendrograma para la tarea 6.

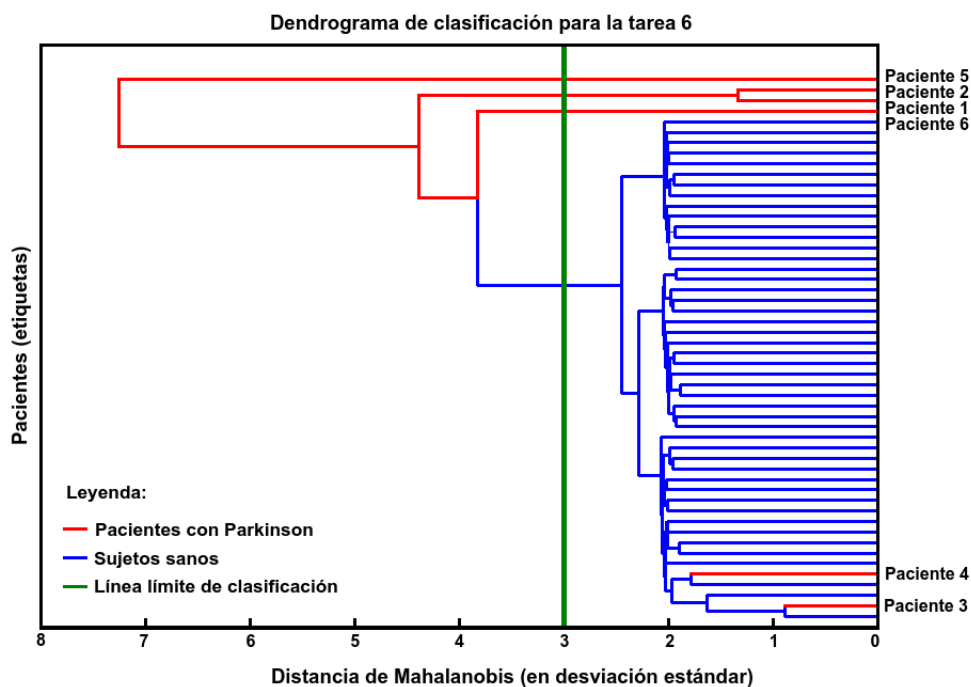


Figura 4.51: Dendrograma de clasificación para la tarea movimiento circular en sentido antihorario con la mano izquierda (tarea 6).

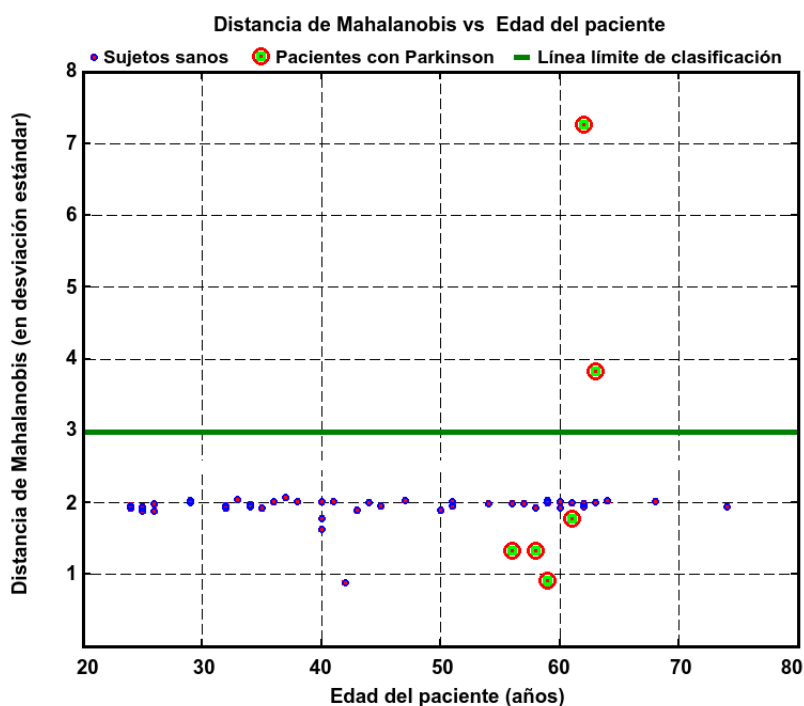


Figura 4.52: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 6.

4.8.7. Dendrograma para la tarea 7.

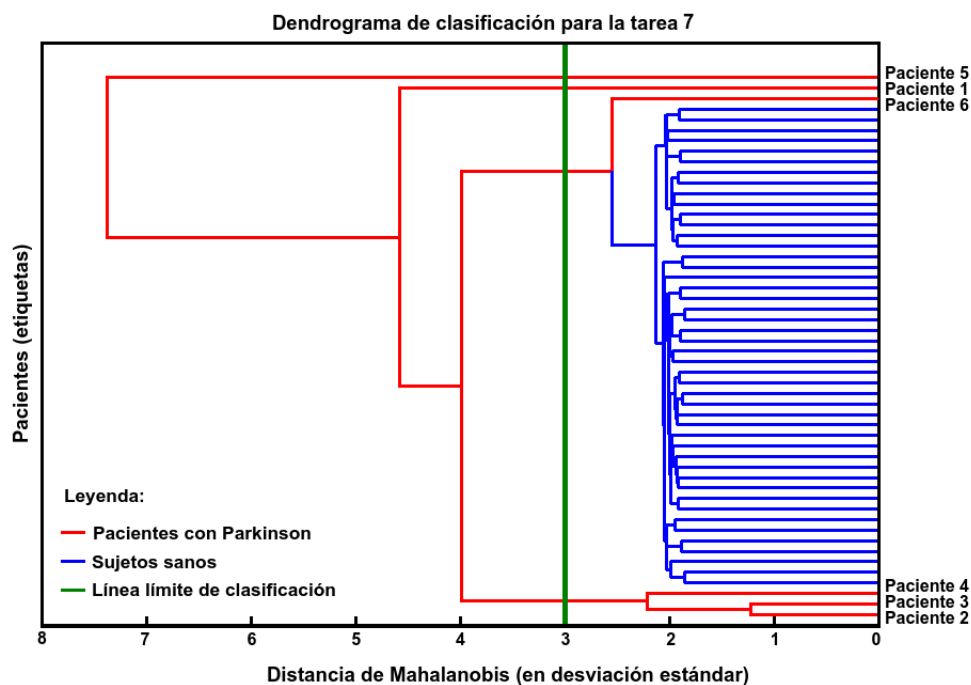


Figura 4.53: Dendrograma de clasificación para la tarea movimiento lineal en sentido vertical con la mano izquierda (tarea 7).

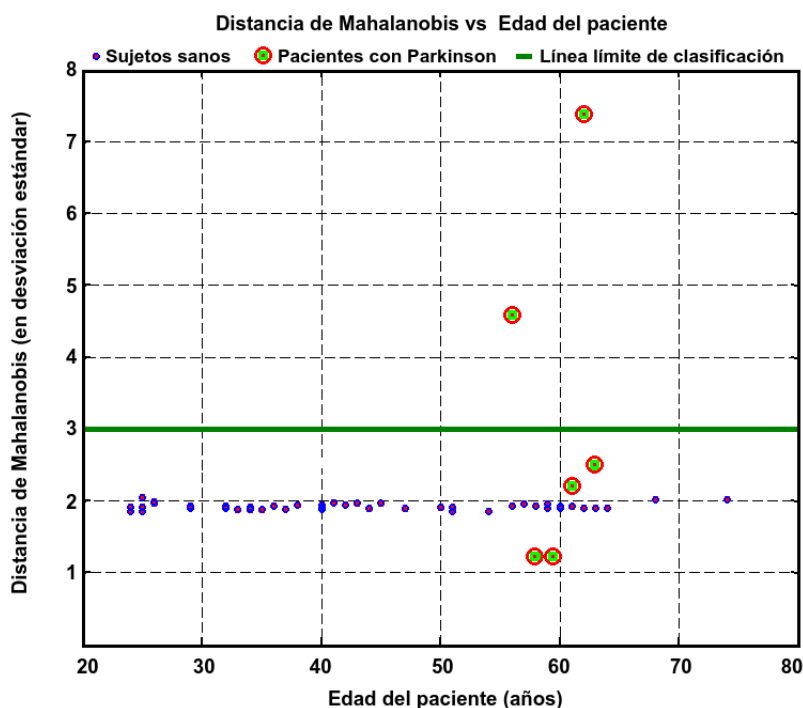


Figura 4.54: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 7.

4.8.8. Dendrograma para la tarea 8.

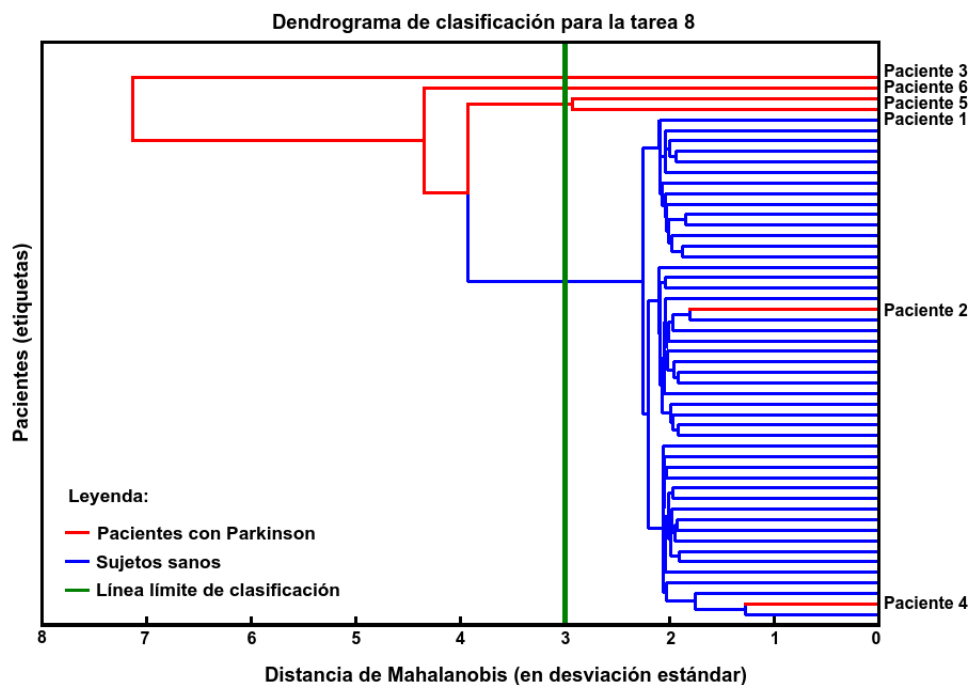


Figura 4.55: Dendrograma de clasificación para la tarea movimiento lineal en sentido horizontal con la mano izquierda (tarea 8).

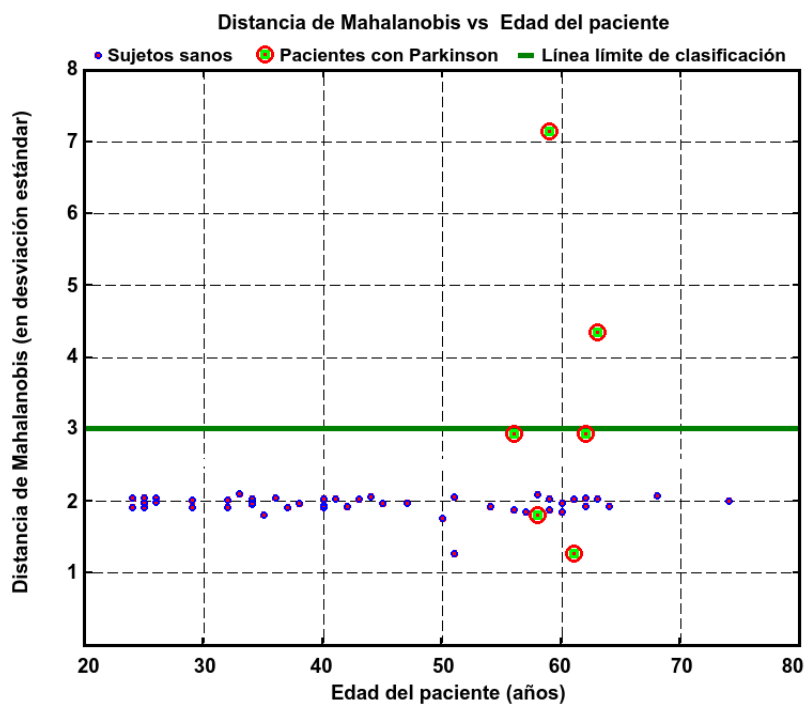


Figura 4.56: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para la tarea 8.

4.8.9. Dendrograma para todas las tareas.

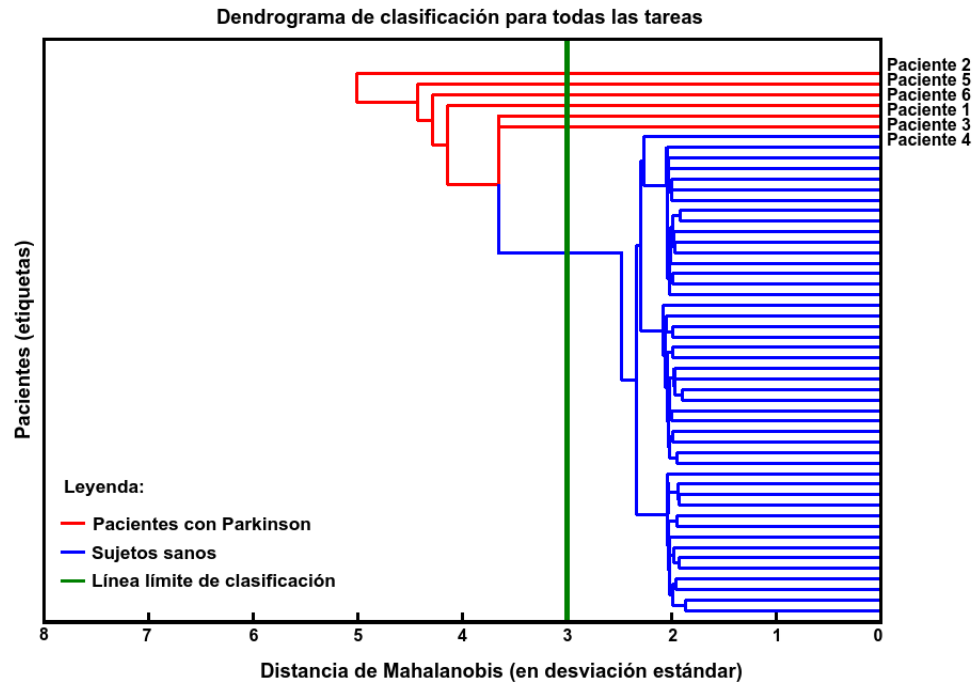


Figura 4.57: Dendrograma de clasificación empleando todas las tareas.

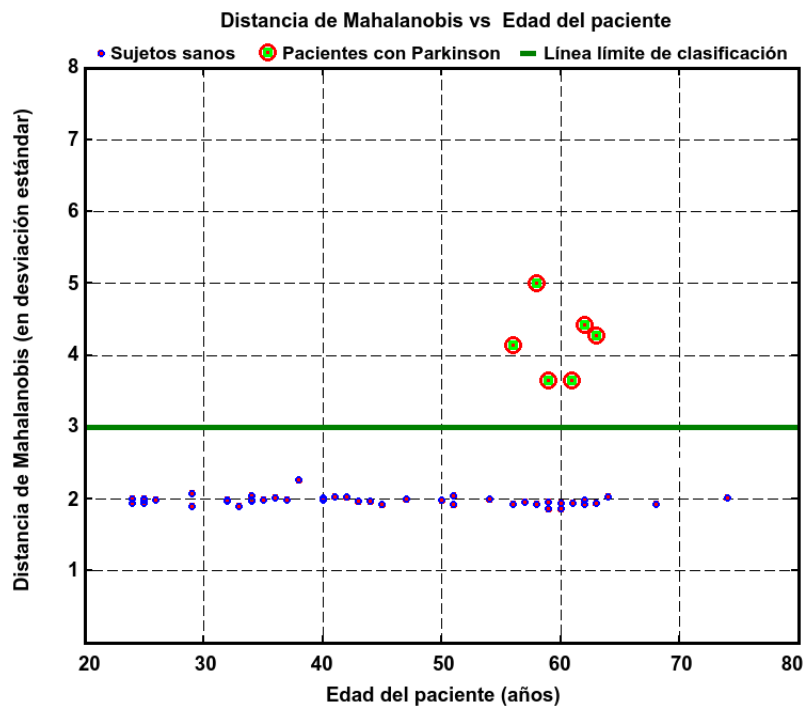


Figura 4.58: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas.

Para el estudio de los dendrogramas, éstos deben ser cortados, lo que corresponde a trazar una línea a lo largo del dendrograma para especificar la agrupación final y obtener el número de clusters en que se divide el conjunto de objetos. La decisión sobre el número de clusters es subjetiva, aún así para tomar una decisión sobre el número de clusters se suelen representar los distintos pasos del algoritmo y la distancia a la que se produce la fusión. En los primeros pasos el salto en las distancias es pequeño, mientras que en los últimos el salto entre los pasos será mayor. El punto de corte será aquel en el que comienzan a producirse saltos bruscos.

Para mantener una congruencia para todas las tareas, se tomó el promedio del valor de estos saltos, se puede observar que este valor está entre 2 y 4, así que se decidió tomar el valor medio entre estos (igual a 3) el cual sería el más adecuado para todos los casos.

Análisis de los dendrogramas:

- **Dendrograma para la tarea 1:** La figura 4.41 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia sobre 4 hasta 7.5, sin embargo dos de estos pacientes se encuentran dentro del grupo sano (con distancias menores a 3). Por lo tanto la tarea movimiento circular en sentido horario con la mano derecha no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.42, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 2 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 2:** La figura 4.43 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 2 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia sobre 3 hasta 6, sin embargo tres de estos pacientes se encuentran dentro del grupo sano (con distancias menores a 3). Por lo tanto la tarea movimiento circular en sentido antihorario con la mano derecha no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.44, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 3 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 3:** La figura 4.45 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas)

está a una distancia sobre 3 hasta 7.5, sin embargo cinco de estos pacientes se encuentran dentro del grupo sano (con distancias menores a 3). Por lo tanto la tarea movimiento lineal en sentido vertical con la mano derecha no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.46, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 5 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 4:** La figura 4.47 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 2 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia sobre 4.5 hasta 6.5, sin embargo dos de estos pacientes se encuentran dentro del grupo sano. (con distancias menores a 3). Por lo tanto la tarea movimiento lineal en sentido horizontal con la mano derecha no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.48, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 2 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 5:** La figura 4.49 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia cercana a 3 hasta 6, sin embargo uno de estos pacientes se encuentran dentro del grupo sano. Por lo tanto la tarea movimiento circular en sentido horario con la mano izquierda no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.50, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 3 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 6:** La figura 4.51 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia desde 1.5 hasta 7.5, sin embargo uno de estos pacientes se encuentran dentro del grupo sano. Por lo tanto la tarea movimiento circular en sentido antihorario con la

mano izquierda no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.52, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 4 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 7:** La figura 4.53 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 2 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está dividido en dos grupos a una distancia desde 1 hasta 7.5, sin embargo la mayor parte de estos sujetos están por debajo de la distancia 3 que fue elegido como valor de clasificación. Por lo tanto la tarea movimiento lineal en sentido vertical con la mano izquierda no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente distante del grupo sano.

En la figura 4.54, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 4 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para la tarea 8:** La figura 4.55 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está a una distancia desde 1 hasta 7.5, sin embargo la mayor parte de estos sujetos están por debajo de la distancia 3 que fue elegido como valor de clasificación, además 2 pacientes están dentro del grupo de sujetos sanos. Por lo tanto la tarea movimiento lineal en sentido horizontal con la mano izquierda no es suficiente por sí sola para realizar una correcta separación, ya que se espera que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.56, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que 4 de los pacientes con Parkinson quedan por debajo de la línea de clasificación.

- **Dendrograma para todas las tareas:** La figura 4.57 muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está dividido en un solo grupo a una distancia desde 3.5 hasta 5, la mayor parte de estos sujetos están por encima de la distancia 3 que fue elegida como valor de clasificación. Por lo tanto todas las tareas en conjunto son suficientes para realizar una correcta separación,

ya que el grupo de pacientes con Parkinson quede completamente separado del grupo sano.

En la figura 4.58, se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se observa que los 6 pacientes con Parkinson quedan por encima de la línea de clasificación, formando un cluster.

Eficiencia de las tareas:

En la tabla 4.15 se muestra la eficiencia de las tareas por separado y de algunas combinaciones entre ellas. Las tareas de 1 a 8 y todas las tareas se mostraron en los dendrogramas anteriores, mientras que para las otras combinaciones presentadas en la tabla no se muestra su dendrograma pero si se presenta su eficiencia. Se observa que la eficiencia de las tareas individuales para separar las clases varía mucho y es baja, no hay una simetría con respecto a mano izquierda o mano derecha. De igual manera, la combinación entre tareas similares no aumenta mucho la eficiencia, esta llega a ser del 100 % para todas las tareas con ambas manos.

Tareas	Porcentaje de eficiencia
Movimiento circular sentido horario mano derecha	66.7
Movimiento circular sentido antihorario mano derecha	50.0
Movimiento lineal sentido vertical mano derecha	16.7
Movimiento lineal sentido horizontal mano derecha	66.7
Movimiento circular sentido horario mano izquierda	50.0
Movimiento circular sentido antihorario mano izquierda	33.3
Movimiento lineal sentido vertical mano izquierda	33.3
Movimiento lineal sentido horizontal mano izquierda	33.3
Tarea círculos ambos movimientos mano derecha	66.7
Tarea círculos ambos movimientos mano izquierda	66.7
Tarea líneas ambos movimientos mano derecha	50.0
Tarea líneas ambos movimientos mano izquierda	33.3
Todas las tareas con la mano derecha	83.3
Todas las tareas con la mano izquierda	66.7
Todas las tareas con ambas manos	100.0

Tabla 4.15: Eficiencia para cada una de las tareas y algunas combinaciones entre ellas.

4.9. Clasificación empleando solo magnetómetro.

Debido a que en el estudio de dinámica corporal se obtuvo un resultado de previa clasificación inter-tarea empleando solo magnetómetro, es necesario revisar si bajo las redes neuronales existe una clasificación entre los dos grupos con todas las tareas. Se crea una base de conocimiento empleando solo el parámetro magnetómetro y luego se entrena la red neuronal hasta obtener la función de la red, se genera su respectivo dendrograma figura 4.59.

Se muestra que el grupo de los sujetos sanos (líneas azules) se encuentra en su mayoría a una distancia igual a 2, a pesar de que se tiene 3 conglomerados importantes, mientras que el grupo de pacientes con Parkinson (líneas rojas) está dividido en uno solo a una distancia desde 3 hasta 5, a pesar de que se tiene una clasificación de grupos, el corte se realiza en el límite seleccionado. Aun así el solo uso de magnetómetro introduce una idea de clasificación y futuramente se puede diseñar una aplicación que genere las tareas y registre los movimientos realizados con el *mouse* de manera que cualquier persona con acceso a una computadora pueda realizar el test para aumentar la base de conocimiento y mejoramiento del algoritmo para una detección temprana de la enfermedad de Parkinson.

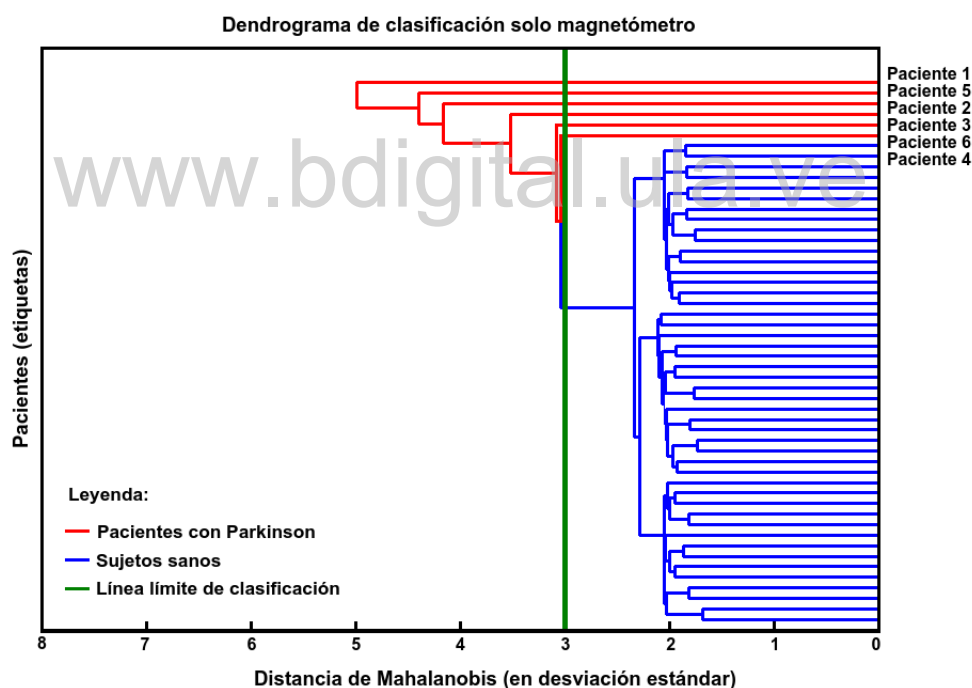


Figura 4.59: Dendrograma de clasificación para todas las tareas empleando solo el magnetómetro.

En la figura 4.60 se observa mejor explicado lo anteriormente descrito. En la representación por edades los sujetos sanos (puntos azules) se encuentran a una distancia 2 y la tendencia se mantiene constante sin importar la edad, se ve claramente la ubicación de los pacientes con Parkinson, se tiene que los 6 pacientes con Parkinson quedan por encima de la línea de clasificación, formando un *cluster*.

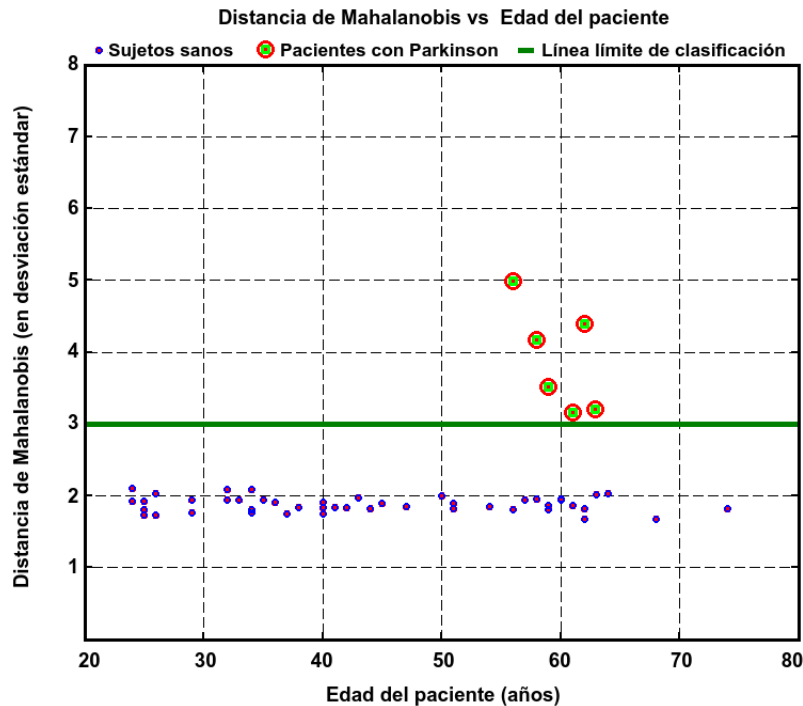


Figura 4.60: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas solo magnetómetro.

4.10. Proyección del grupo prueba.

Ya obtenida una clasificación entre los sujetos sanos y los pacientes con Parkinson empleando todos los parámetros y todas las tareas se realiza la siguiente propuesta:

Sobre la base de sujetos sanos y pacientes con la enfermedad de Parkinson y haciendo uso de los parámetros de la red con la que se obtuvo 100 % de eficiencia se proyecta uno a uno los sujetos del grupo prueba conformado por 15 sujetos: 7 mujeres y 8 hombres con edades comprendidas entre los 25 años y los 67 años. Se proyecta usando el programa “**Proyección del paciente desconocido.m**”, luego se obtiene la nueva función de la red **Y** que será la entrada para la construcción de los dendrogramas de clasificación. Se propone que si el sujeto queda por debajo de la línea de clasificación sus características dinámicas son más similares a la de los sujetos sanos, mientras que si la clasificación queda por encima, sus características dinámicas son más similares a la de los pacientes con Parkinson, siendo el sujeto prueba una persona que no presenta síntomas relacionados a la enfermedad.

Se observa en las figuras 4.65, 4.71, 4.81 y 4.89 que los sujetos quedan por encima de la línea de clasificación ($\sigma = 3$). En teoría estos sujetos tienen sus características dinámicas más similares hacia los pacientes con Parkinson, ya que su distancia de Mahalanobis es más pequeña en comparación con el grupo de sujetos sanos. En total se presentaron 15 sujetos, de los cuales 4 quedaron por encima de la línea de clasificación por lo tanto el algoritmo tuvo la capacidad de clasificar a 11 sujetos más cercanos al grupo sanos, lo que representa un 73.3 % de efectividad. Aún así, se está lejos de verificar que los otros sujetos puedan desarrollar la enfermedad, ya que habría que hacer un seguimiento durante años.

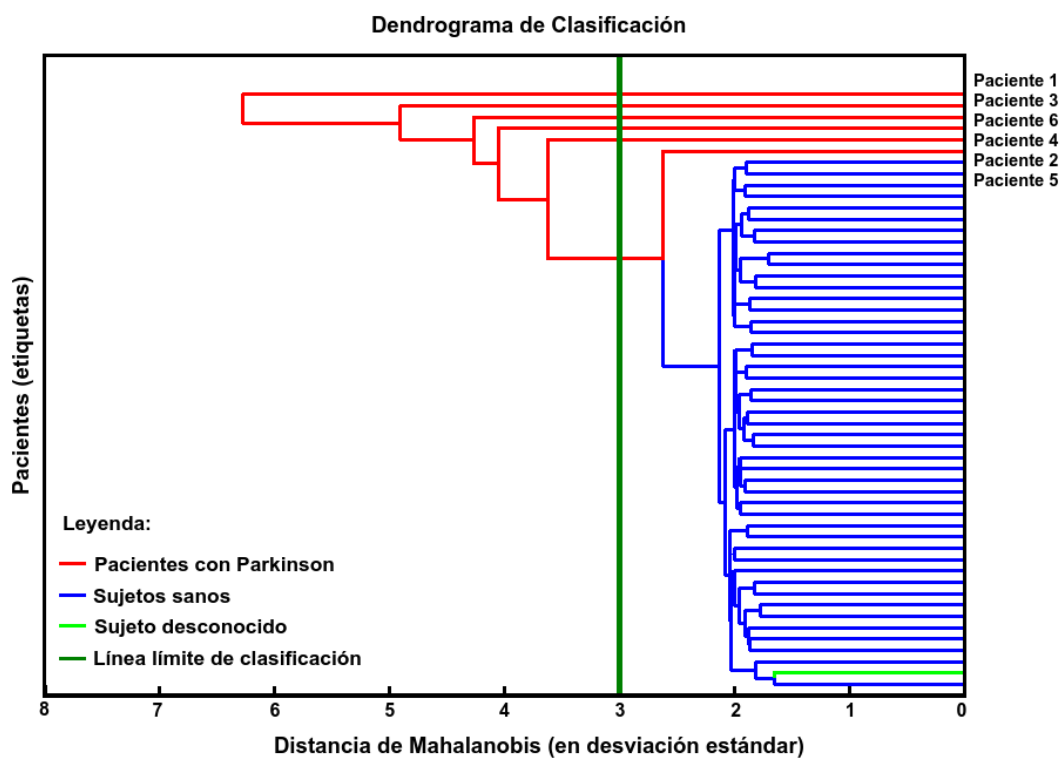


Figura 4.61: Dendrograma de clasificación para un sujeto femenino de 25 años.

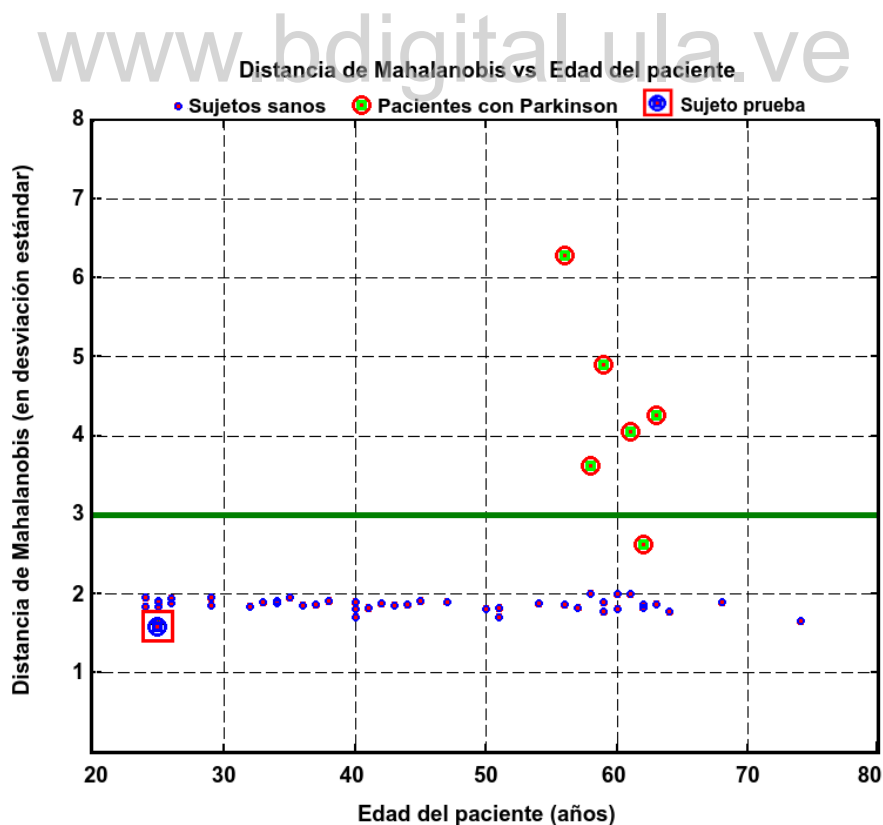


Figura 4.62: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 25 años.

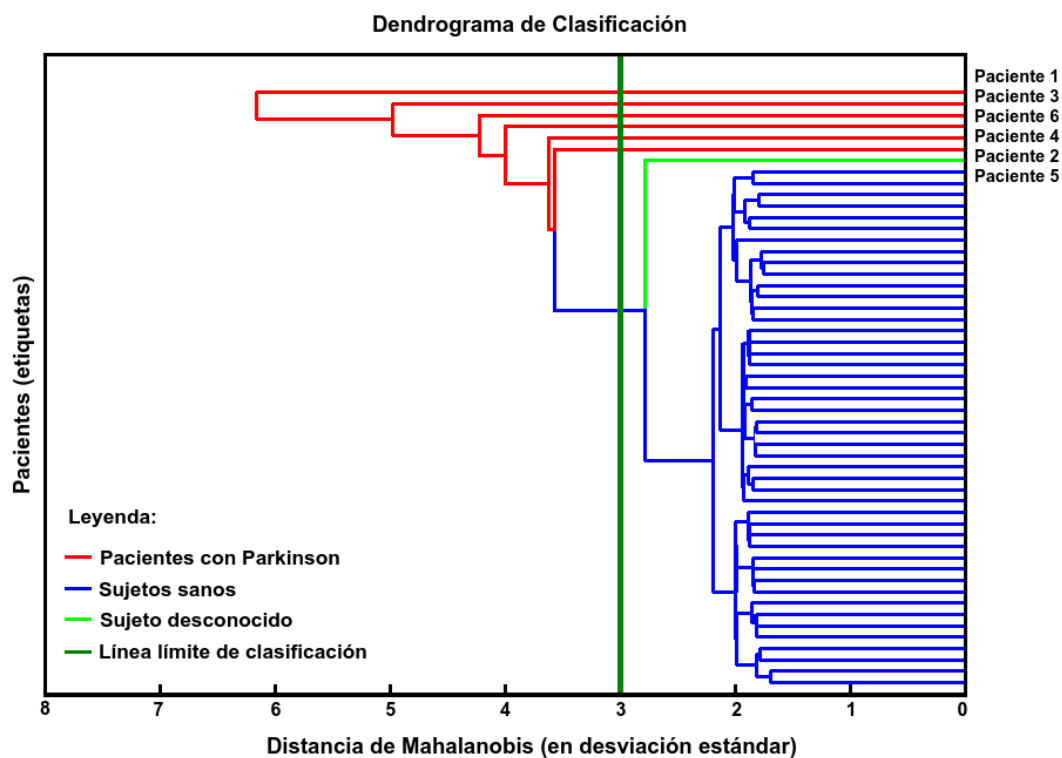


Figura 4.63: Dendrograma de clasificación para un sujeto masculino de 25 años.

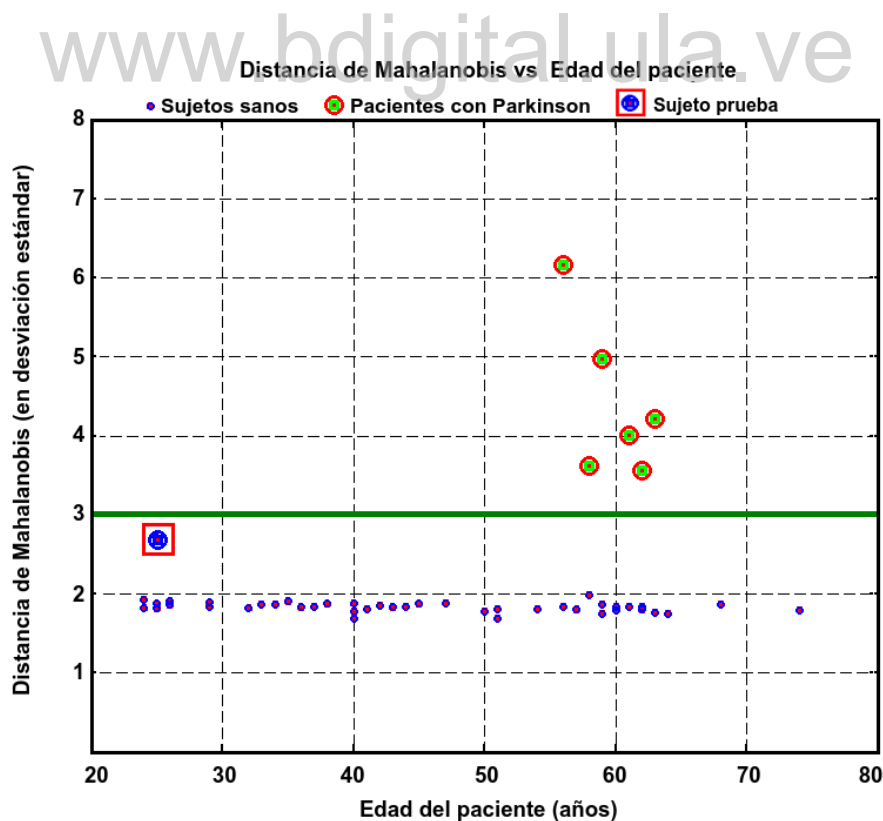


Figura 4.64: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 25 años.

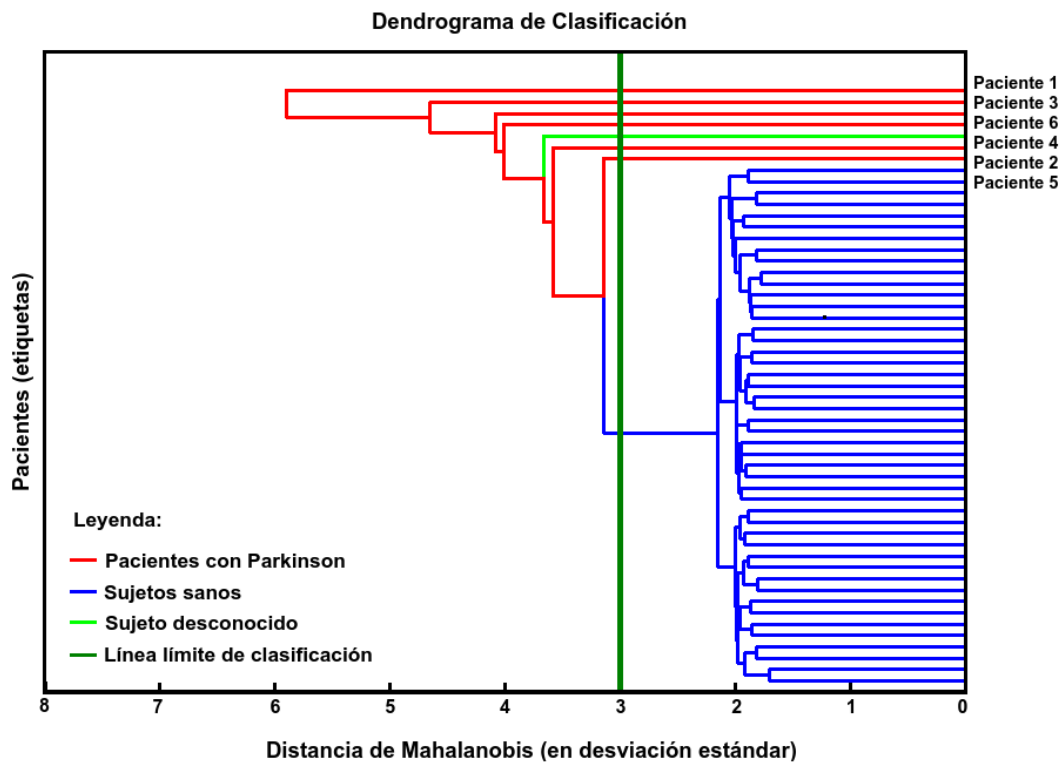


Figura 4.65: Dendrograma de clasificación para un sujeto femenino de 29 años.

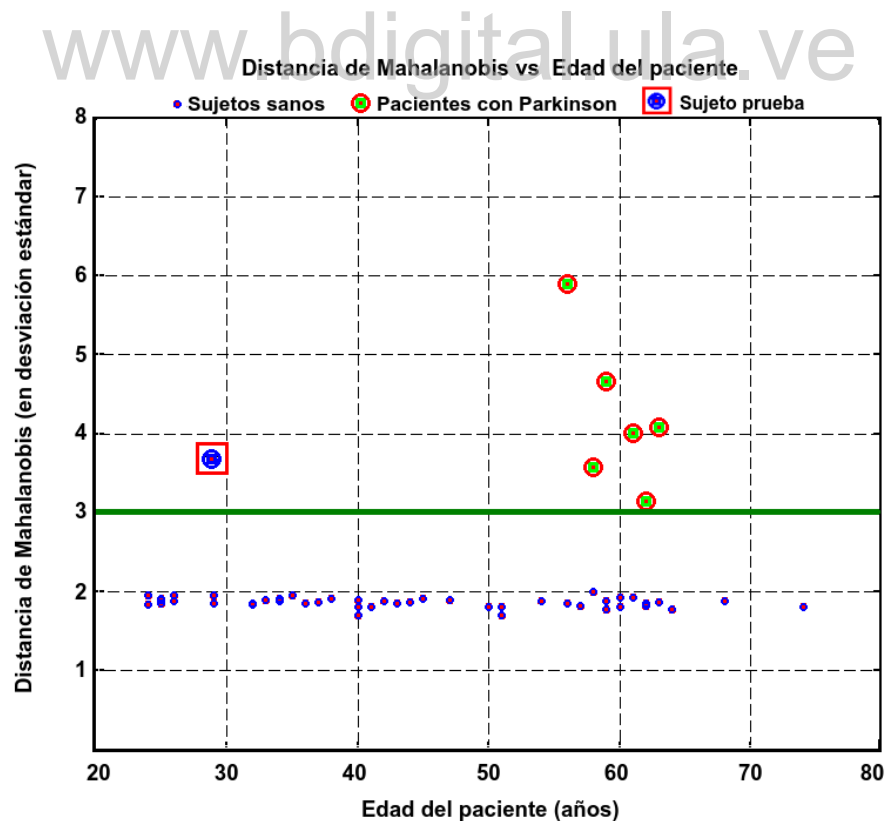


Figura 4.66: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 29 años.

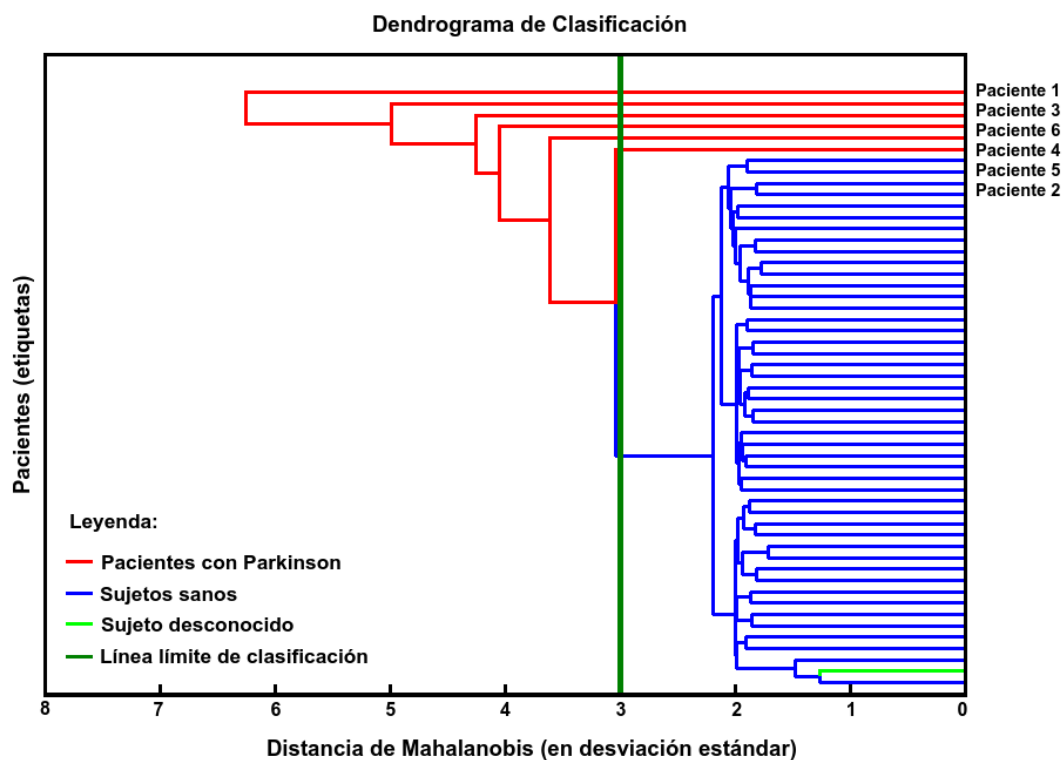


Figura 4.67: Dendrograma de clasificación para un sujeto masculino de 29 años.

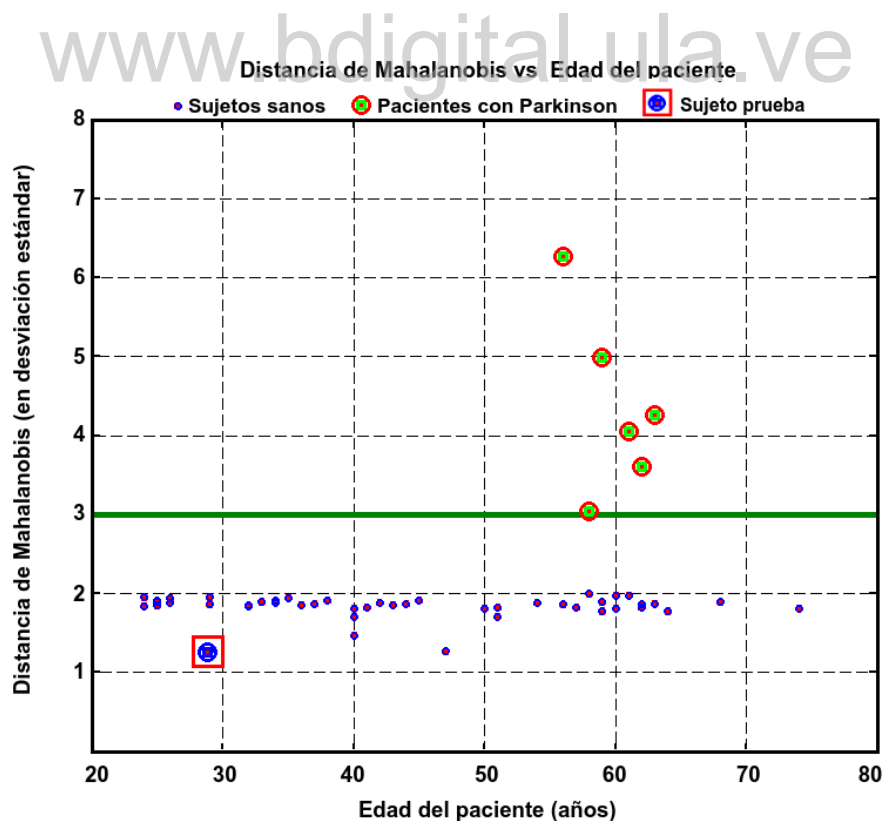


Figura 4.68: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 29 años.

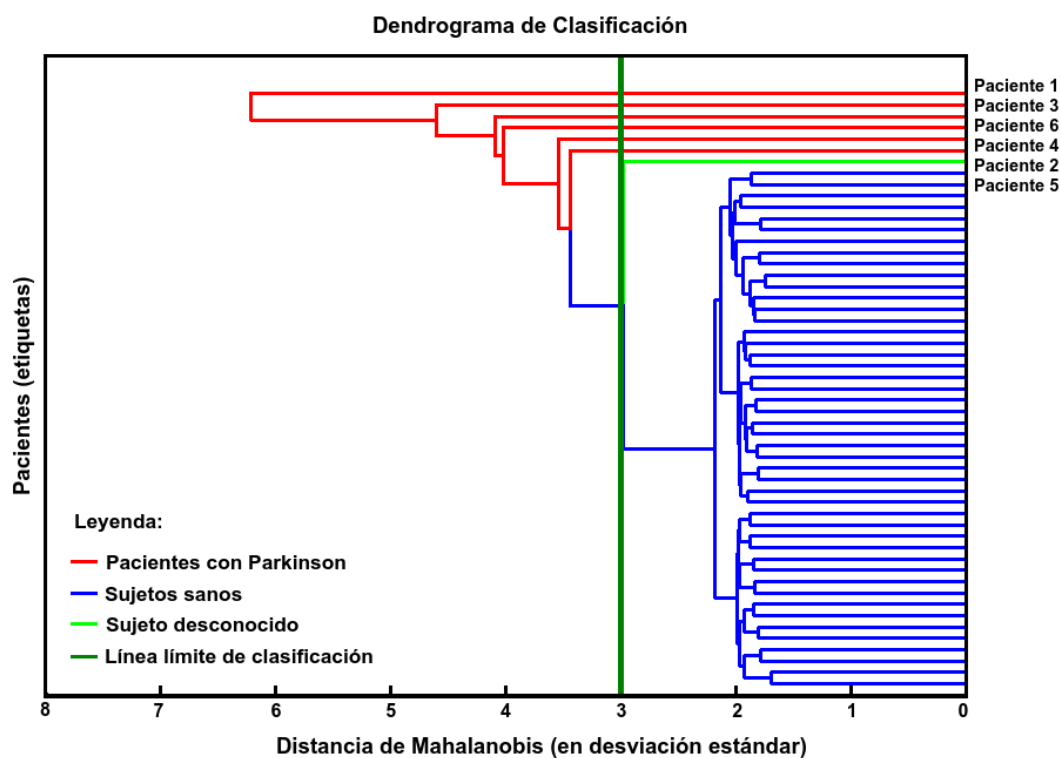


Figura 4.69: Dendrograma de clasificación para un sujeto masculino de 30 años.

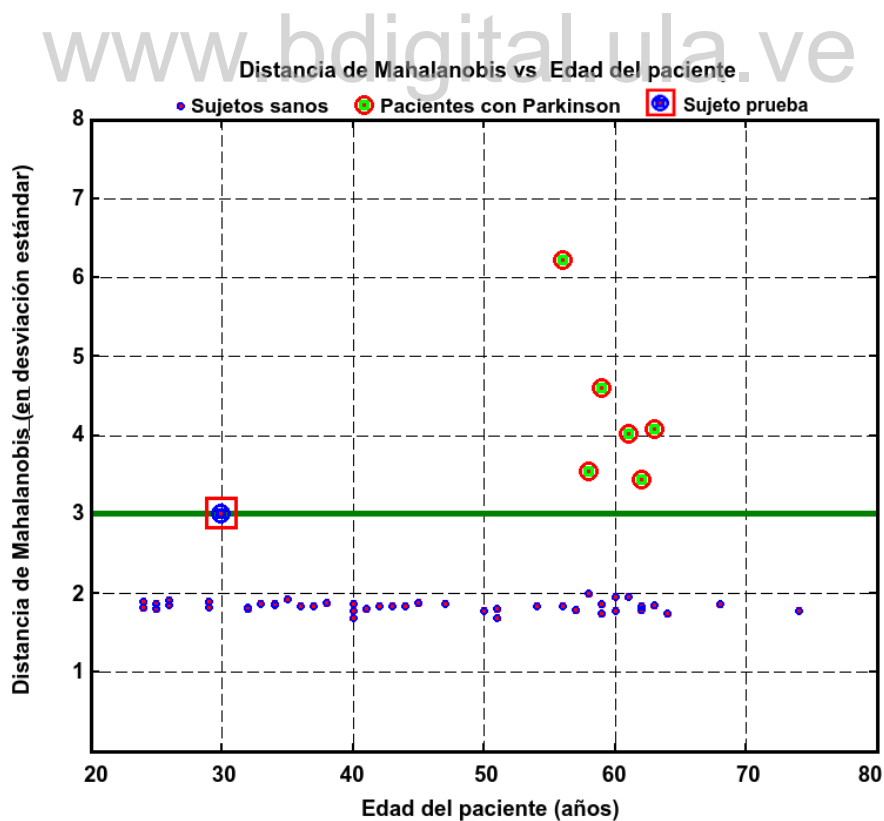


Figura 4.70: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 30 años.

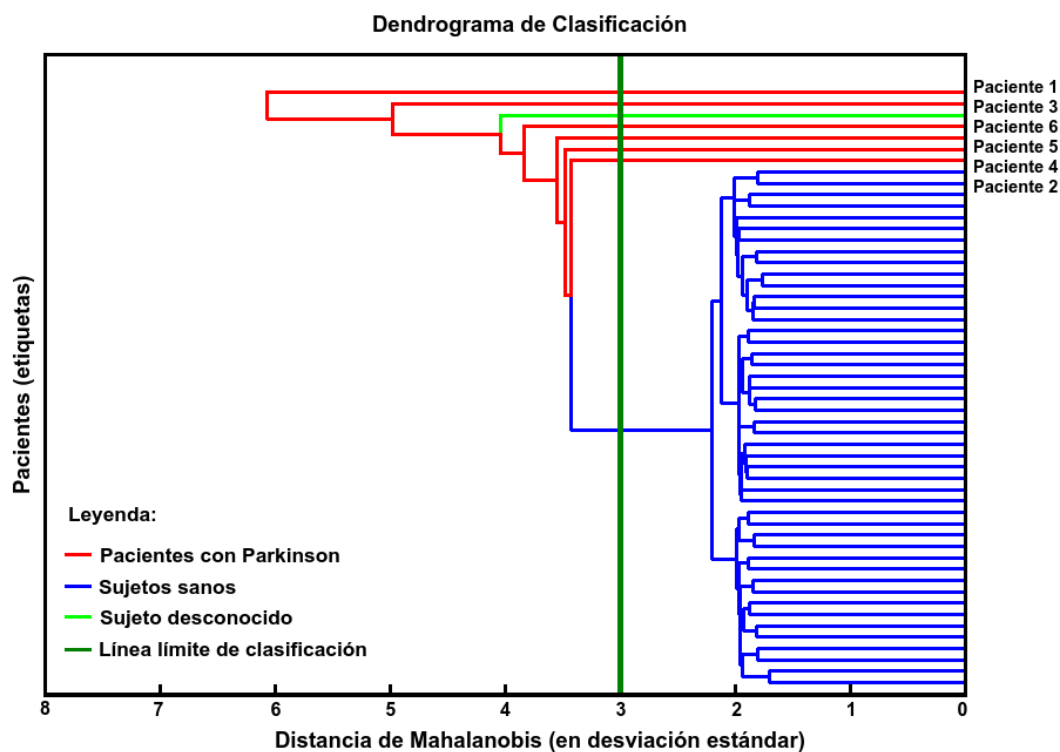


Figura 4.71: Dendrograma de clasificación para un sujeto masculino de 31 años.

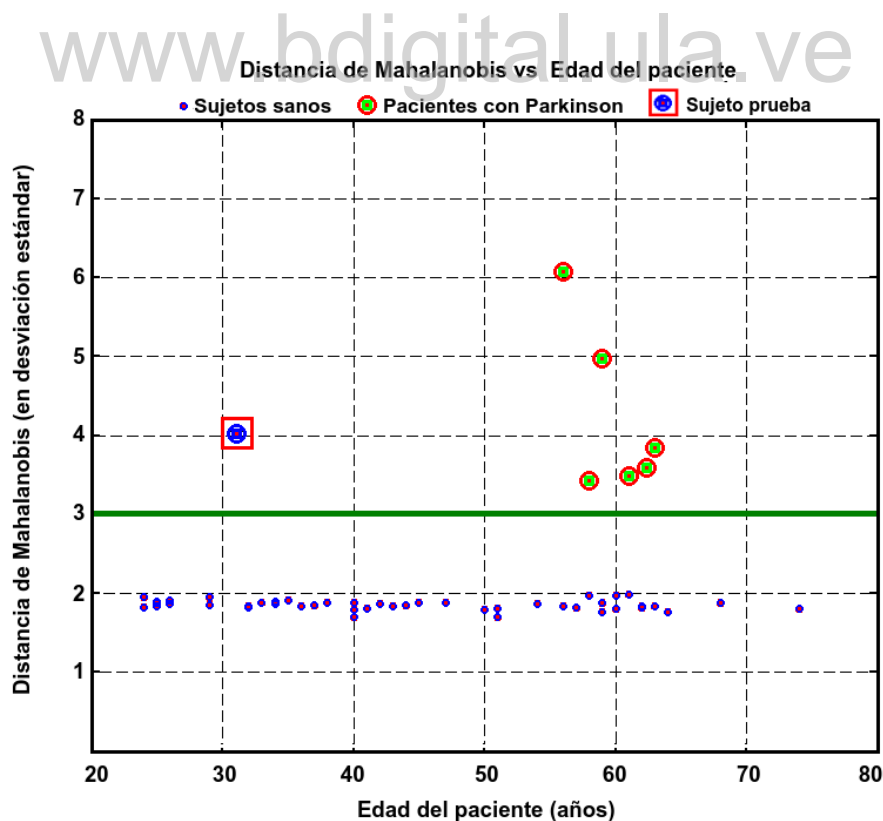


Figura 4.72: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 31 años.

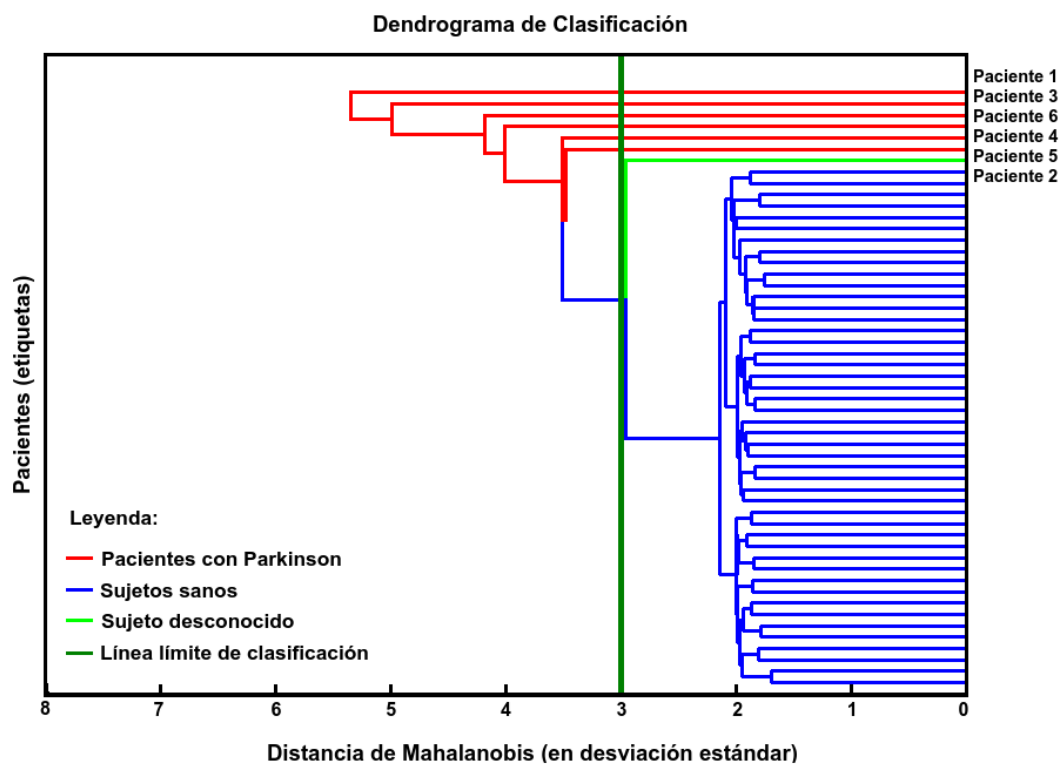


Figura 4.73: Dendrograma de clasificación para un sujeto femenino de 34 años.

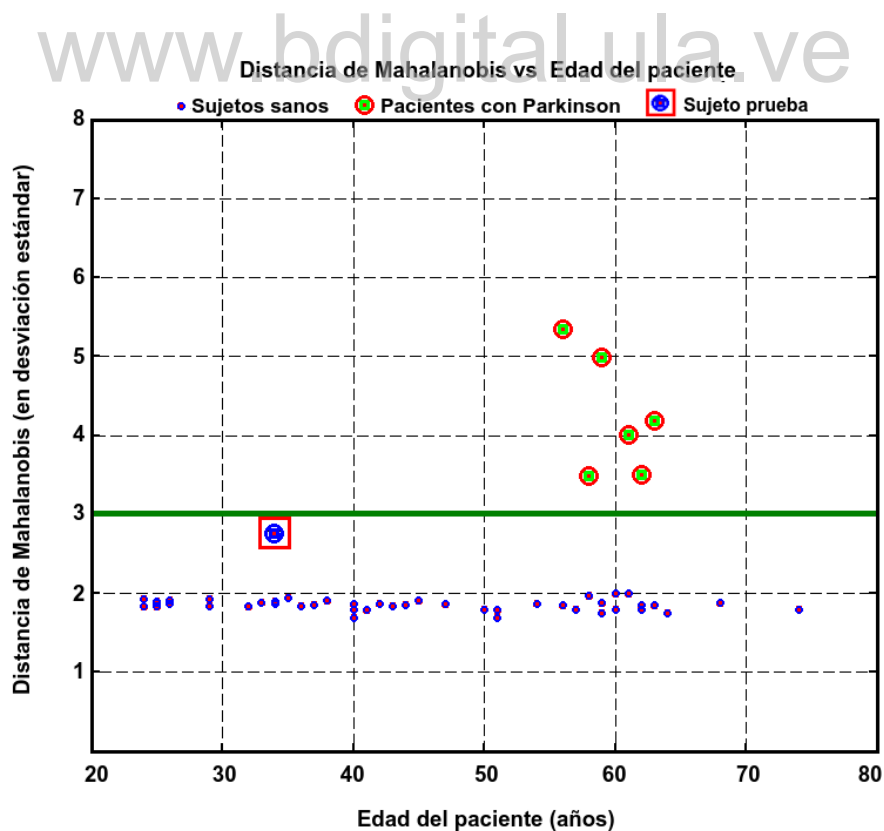


Figura 4.74: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 34 años.

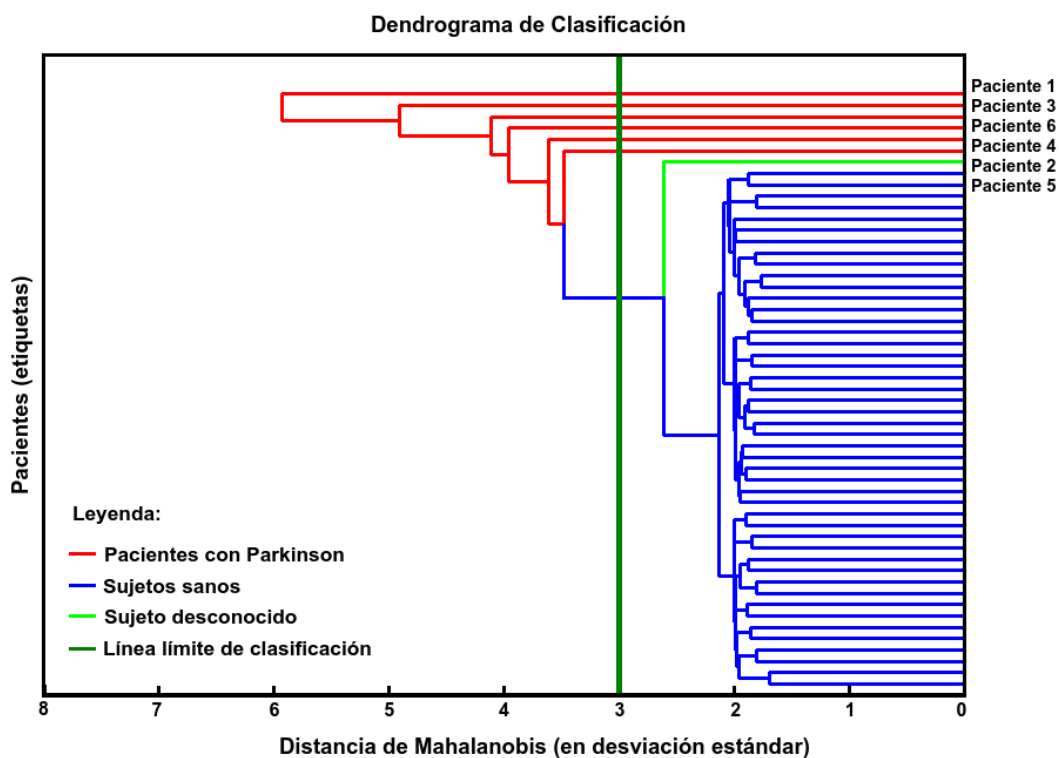


Figura 4.75: Dendrograma de clasificación para un sujeto masculino de 39 años.

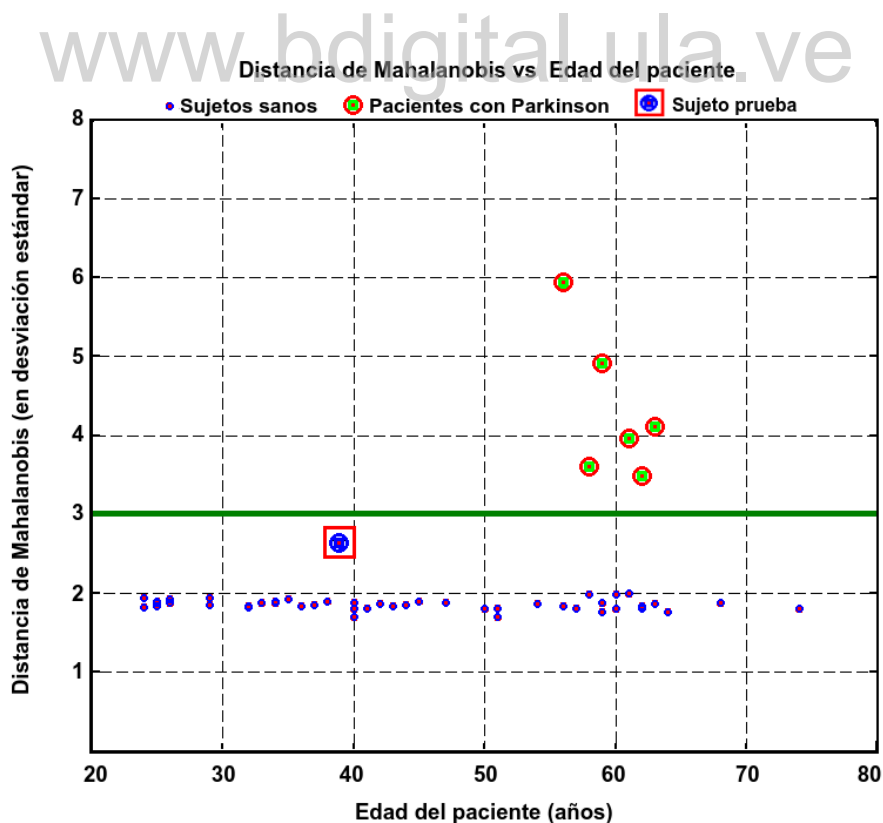


Figura 4.76: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 39 años.

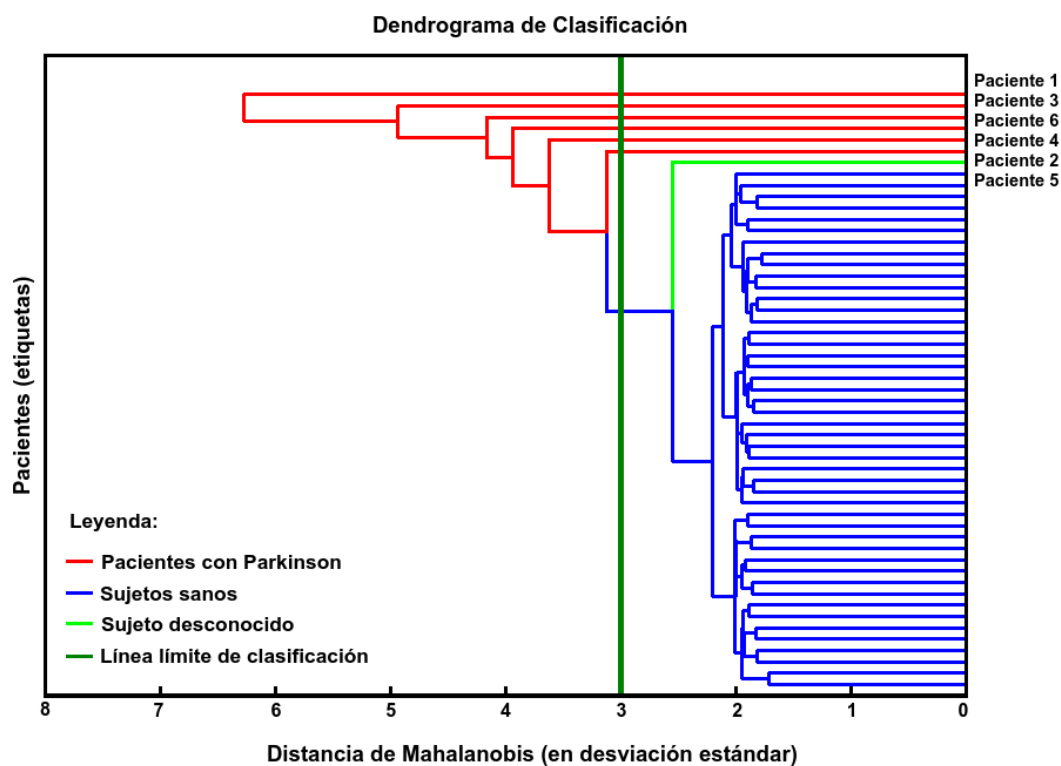


Figura 4.77: Dendrograma de clasificación para un sujeto femenino de 42 años.

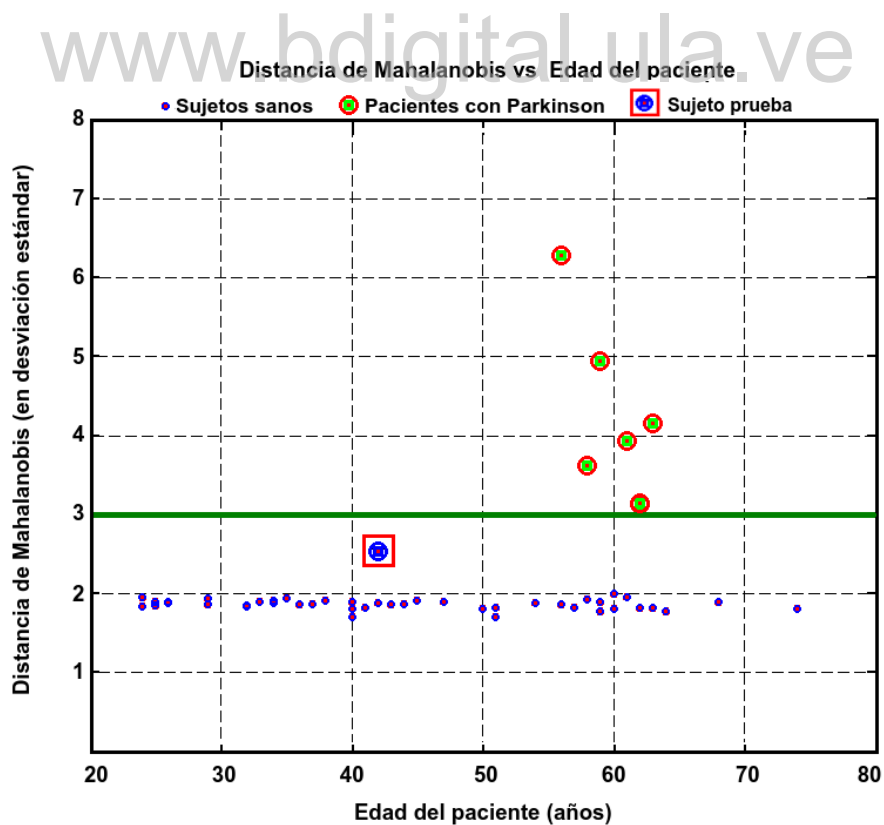


Figura 4.78: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 42 años.

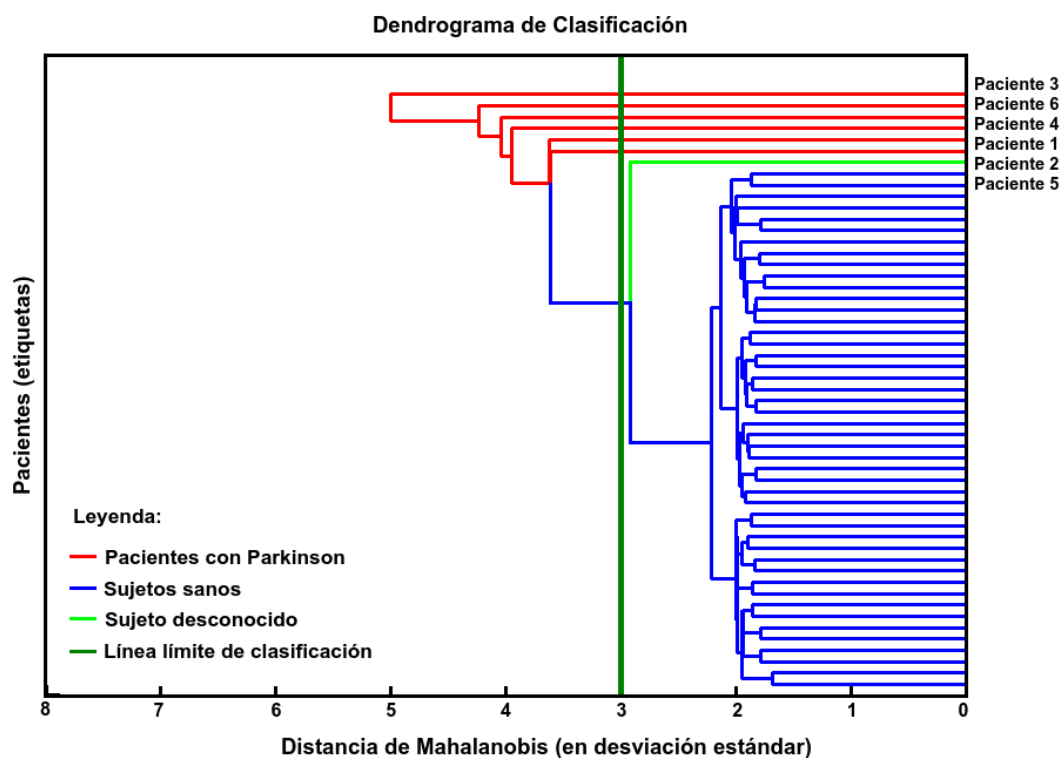


Figura 4.79: Dendrograma de clasificación para un sujeto femenino de 48 años.

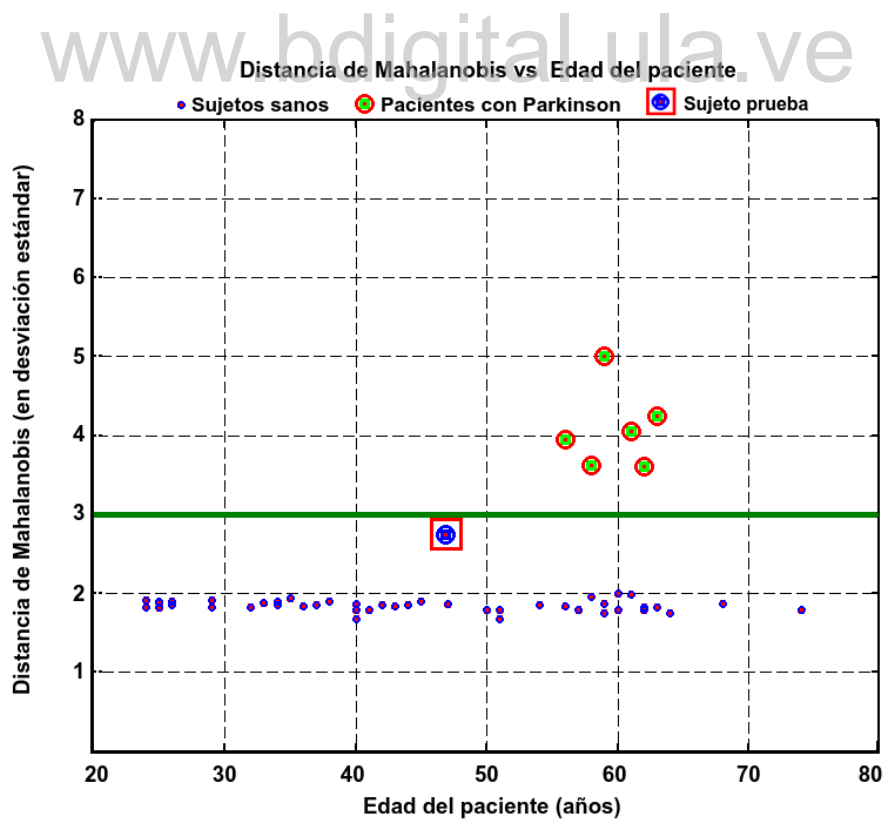


Figura 4.80: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 48 años.

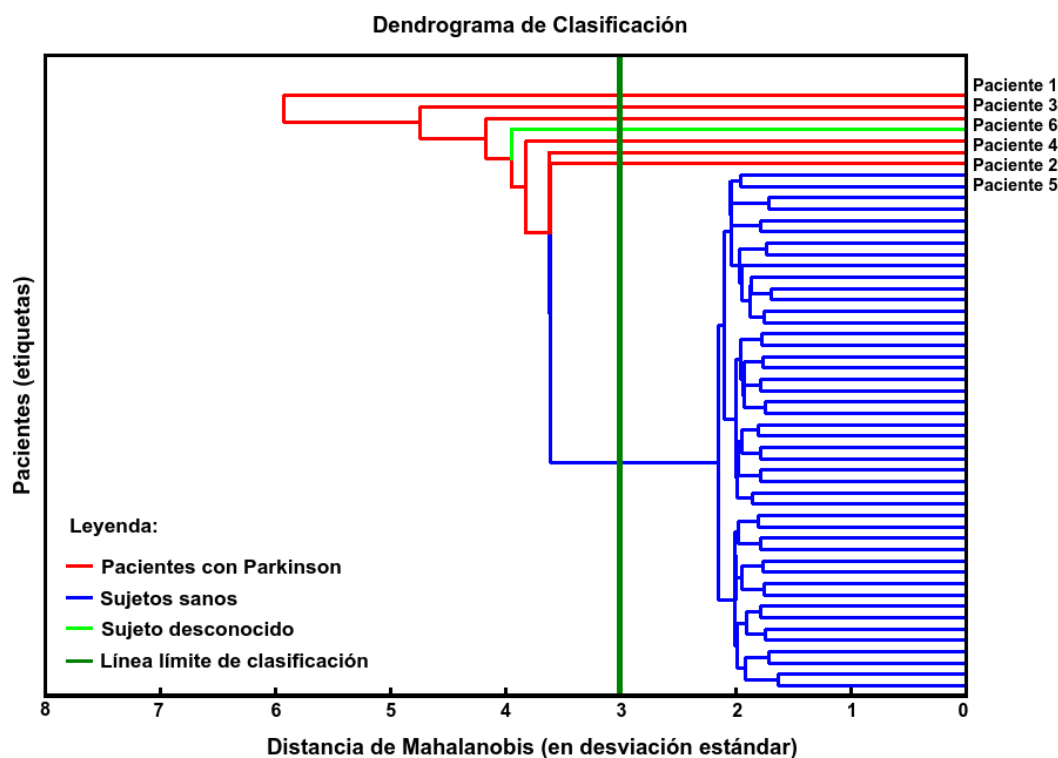


Figura 4.81: Dendrograma de clasificación para un sujeto masculino de 48 años.

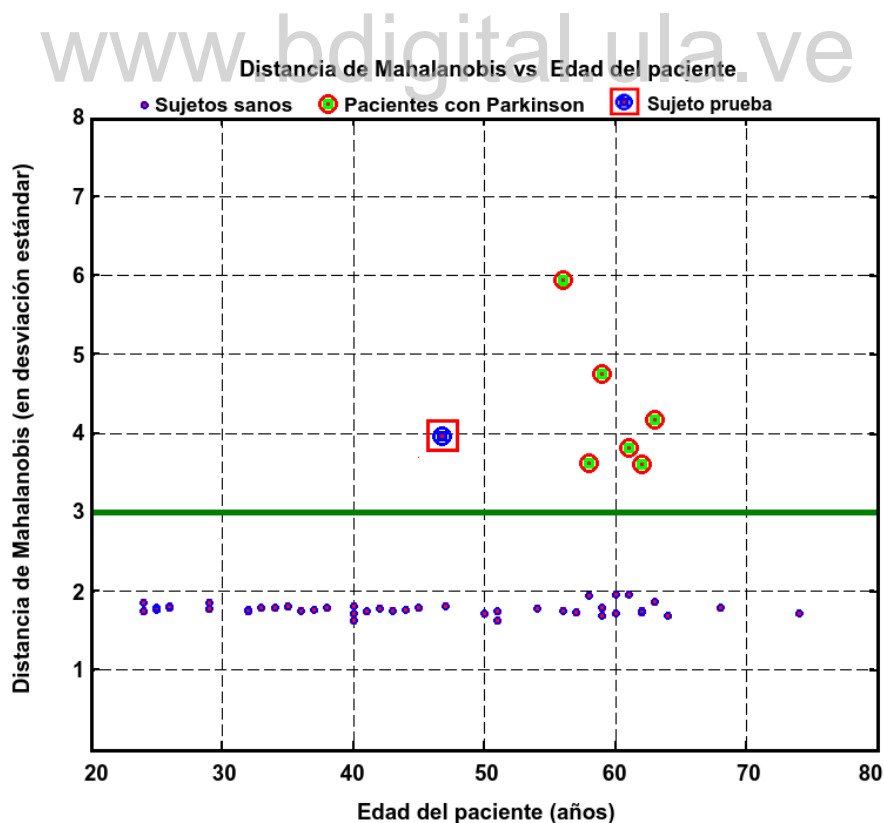


Figura 4.82: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 48 años.

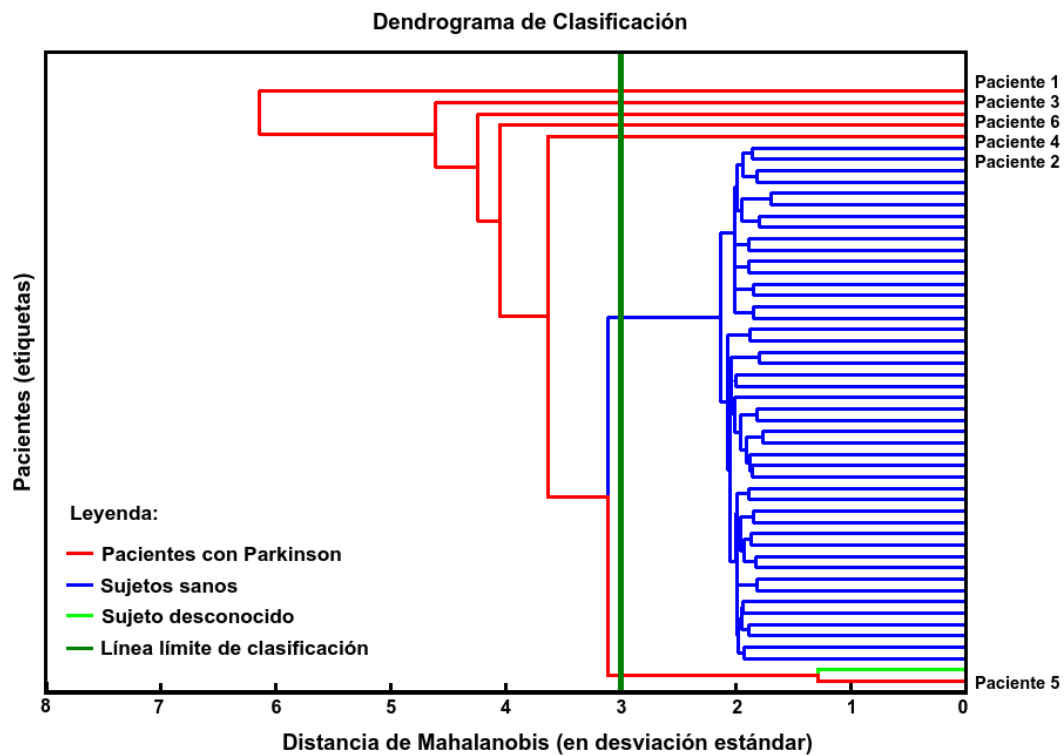


Figura 4.83: Dendrograma de clasificación para un sujeto masculino de 51 años.

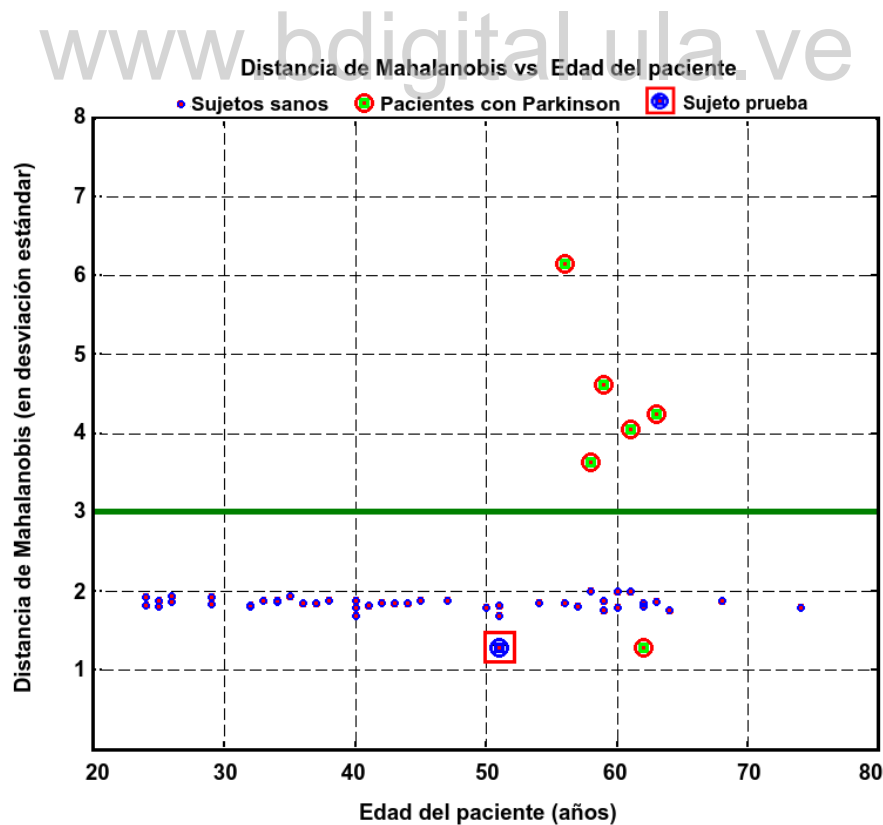


Figura 4.84: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 51 años.

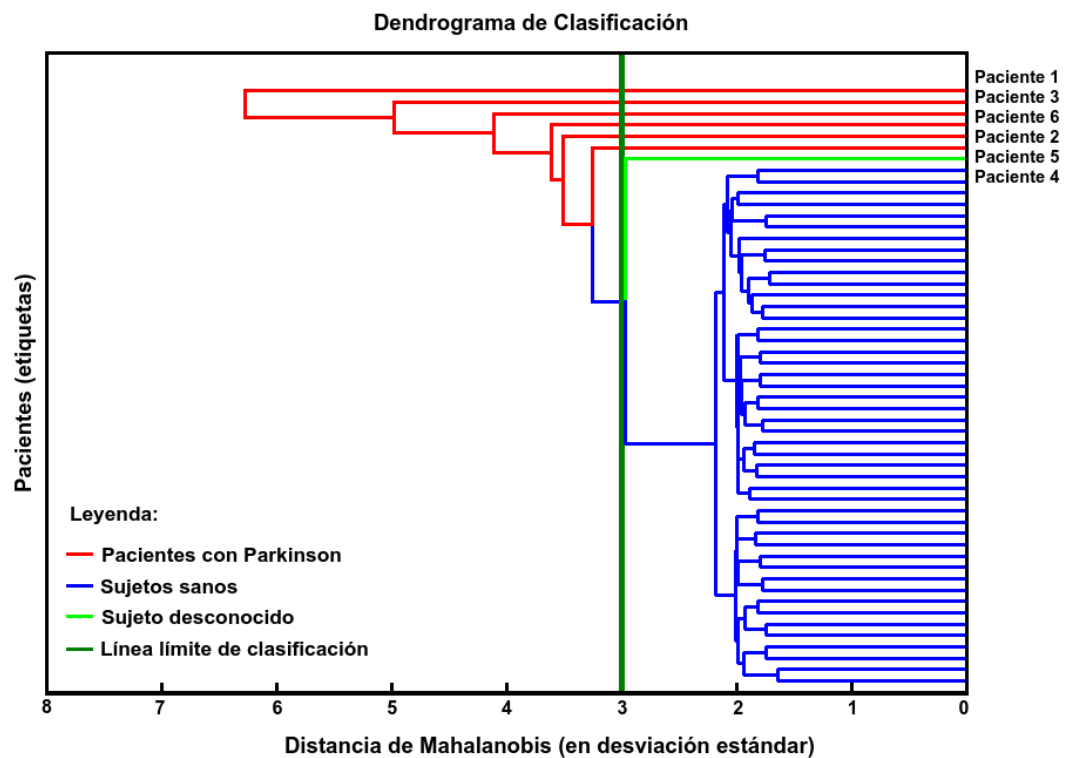


Figura 4.85: Dendrograma de clasificación para un sujeto masculino de 53 años.

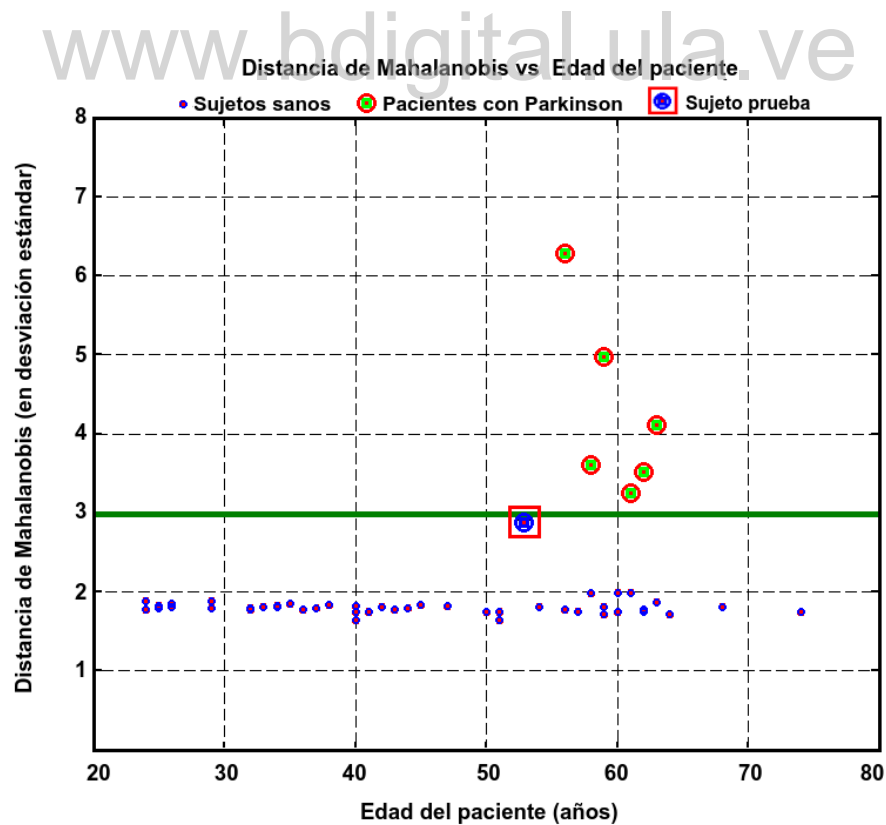


Figura 4.86: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto masculino de 53 años.

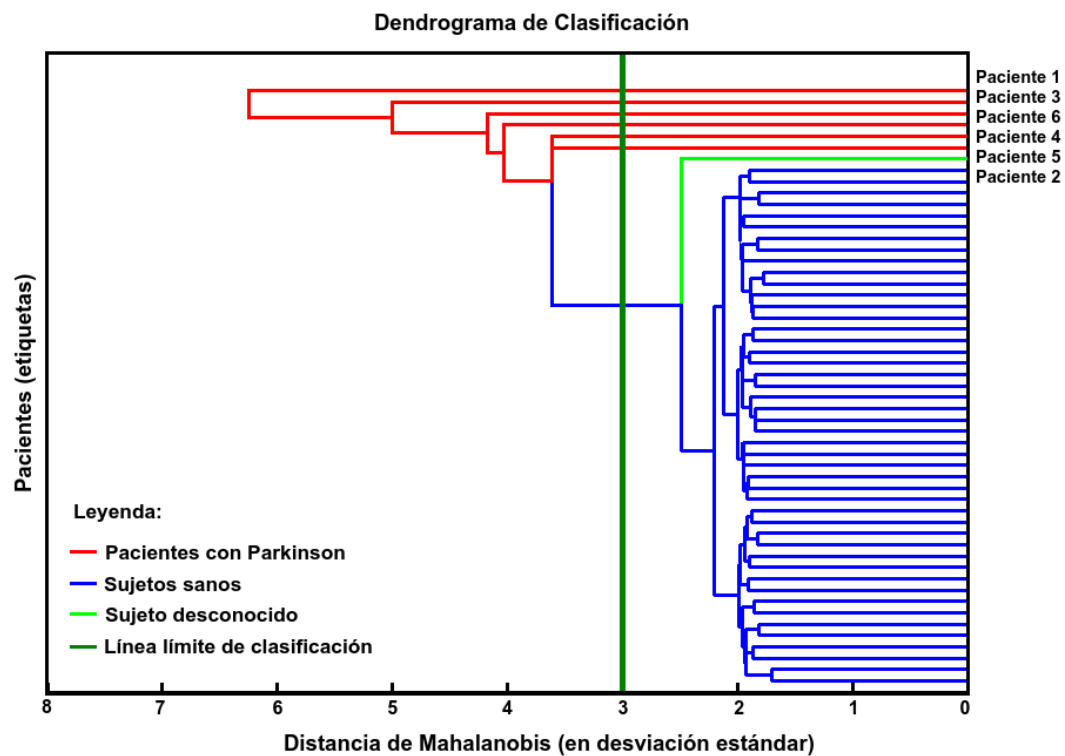


Figura 4.87: Dendrograma de clasificación para un sujeto femenino de 62 años.

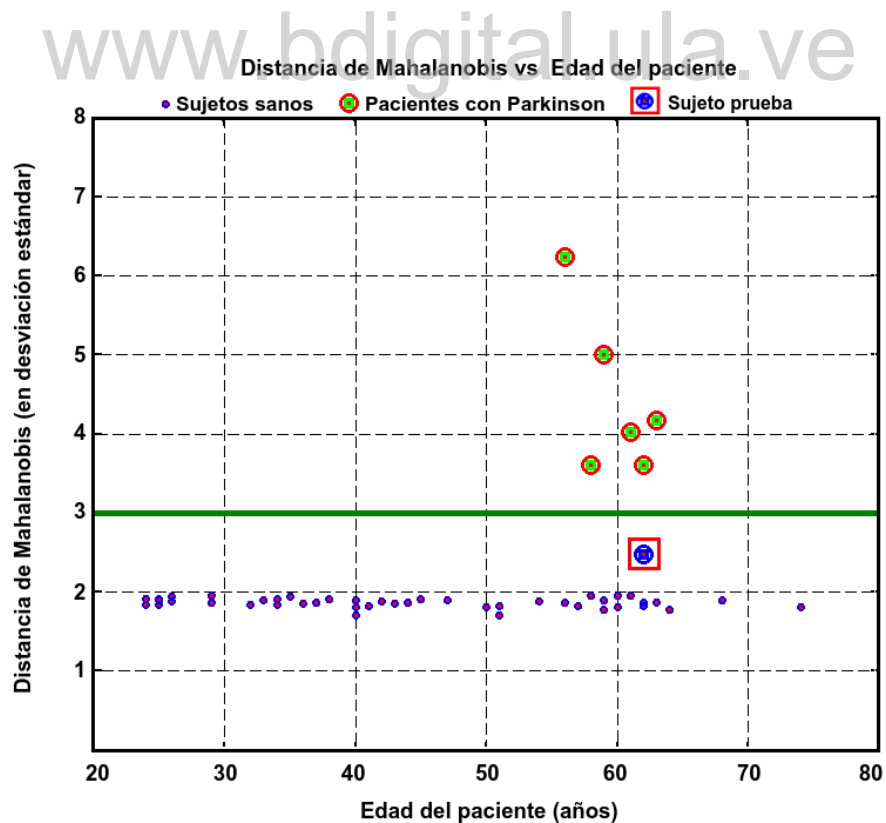


Figura 4.88: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 62 años.

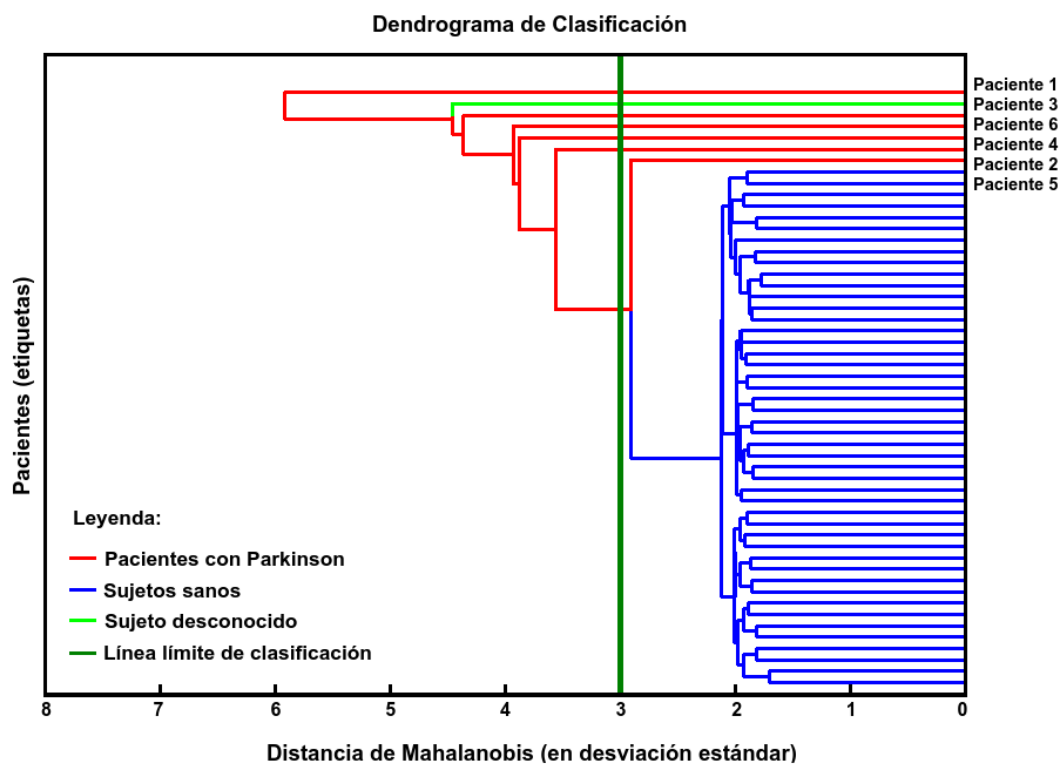


Figura 4.89: Dendrograma de clasificación para un sujeto femenino de 67 años.

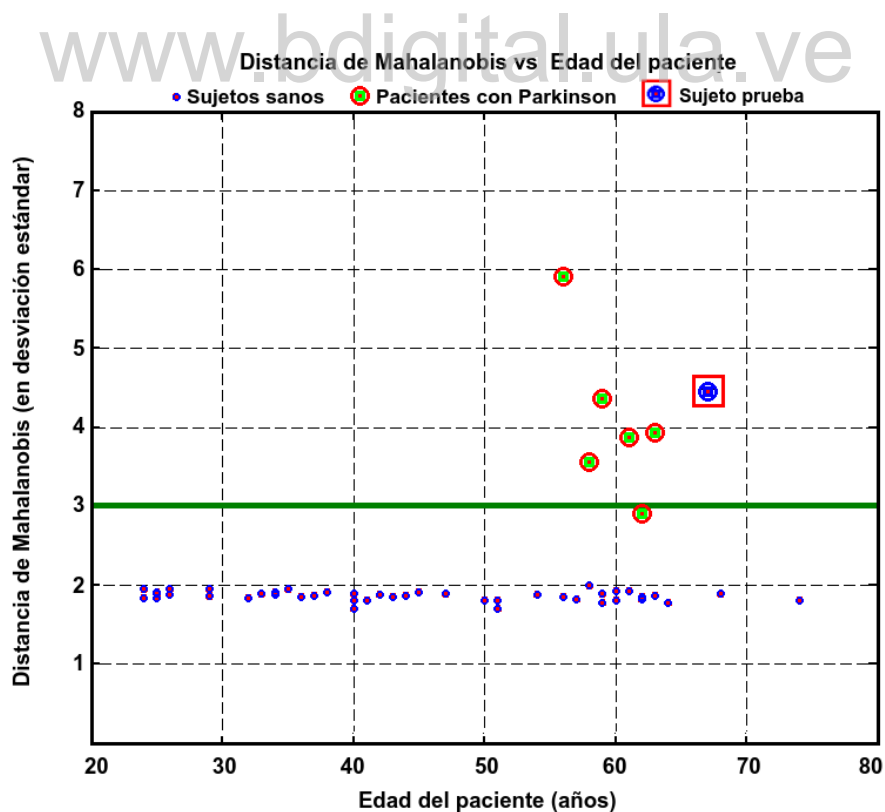


Figura 4.90: Distancia de Mahalanobis en función de la edad de los pacientes con el fin de visualizar la ubicación y separación entre ellos para todas las tareas para un sujeto femenino de 6 años.

Capítulo 5

Discusiones y conclusiones.

5.1. Discusiones.

Se ha propuesto un sistema no invasivo para el diagnóstico temprano de la enfermedad de Parkinson, basado en la dinámica corporal y en redes neuronales, específicamente en el algoritmo de retropropagación (BP). Se construyó una base de datos constituida por 60 sujetos sanos con edades comprendidas entre los 25 y 74 años y 6 pacientes con Parkinson en los estadios I y II de la enfermedad. Los datos fueron adquiridos a través de un sensor GY-80 compuesto por un acelerómetro, un giróscopo y un magnetómetro, además de un pulsómetro todos conectados a una tarjeta Arduino Due. Se calibraron los sensores y se adquirieron los datos con el programa Aqevent.

Inicialmente se realizó el análisis dinámico por cada tarea, este concluyó:

- **Frecuencia central de los parámetros por todas las tareas por pacientes:** Existe una regularidad en la frecuencia central para los dos grupos, entre 0.5 Hz y 1 Hz, por lo que es un indicador de que no hay alguna pre-clasificación de los grupos empleando las tareas seleccionadas.
- **Frecuencia central de los parámetros por tarea y todos los sujetos:** En la tabla 5.1 se muestra el resumen de los valores frecuenciales promedios para la aceleración, campo magnético, velocidad angular y pulso cardíaco de los grupos sanos, patológicos y prueba, por tarea. En general los valores frecuenciales son muy similares por parámetro y por tarea, lo que indica que no hay diferencia entre el grupo de sujetos sanos y el grupo de pacientes con Parkinson, por lo tanto no existe una clasificación en términos de frecuencia para ambos grupos.
- **Valores medios de los parámetros para todos los pacientes por tarea:** En la tabla 5.2 se muestran los valores promedios de la aceleración, campo magnético, velocidad angular y pulso cardíaco de los grupos sanos y patológicos, por cada tarea. Los valores de la aceleración, velocidad angular y pulso cardíaco son muy similares entre ellos en cada tarea, por lo que no son suficientes por sí solos para establecer una clasificación entre los sujetos por parámetro. Los valores del campo magnético varían entre las tareas de círculos (1,2,5,6) y líneas (3,4,7,8) esto es debido al punto de asimetría creado por el imán dispuesto debajo de la mesa, se puede establecer una clasificación inter-tarea pero no entre los sujetos sanos y patológicos.

Tareas	Aceleración (Hz)	Campo magnético (Hz)	Velocidad angular (Hz)	Pulso cardíaco (Hz)
Tarea 1	(0.66 ± 0.02)	(0.67 ± 0.02)	(0.66 ± 0.02)	(1.39 ± 0.08)
Tarea 2	(0.68 ± 0.03)	(0.67 ± 0.03)	(0.69 ± 0.03)	(1.44 ± 0.07)
Tarea 3	(0.68 ± 0.05)	(0.65 ± 0.02)	(0.71 ± 0.03)	(1.44 ± 0.05)
Tarea 4	(0.69 ± 0.04)	(0.64 ± 0.02)	(0.71 ± 0.03)	(1.42 ± 0.10)
Tarea 5	(0.66 ± 0.04)	(0.65 ± 0.03)	(0.66 ± 0.04)	(1.39 ± 0.11)
Tarea 6	(0.67 ± 0.02)	(0.65 ± 0.03)	(0.66 ± 0.02)	(1.35 ± 0.13)
Tarea 7	(0.69 ± 0.03)	(0.67 ± 0.03)	(0.72 ± 0.02)	(1.42 ± 0.07)
Tarea 8	(0.68 ± 0.03)	(0.65 ± 0.06)	(0.70 ± 0.02)	(1.42 ± 0.08)

Tabla 5.1: Resumen de los valores frecuenciales promedios de los grupos: sanos, patológicos y prueba, por tarea.

Tareas	Aceleración (g)	Campo magnético (μ T)	Velocidad angular ($^{\circ}$ /s)	Pulso cardíaco (Hz)
Tarea 1	(1.08 ± 0.03)	(60 ± 3)	(62 ± 5)	(1.40 ± 0.08)
Tarea 2	(1.09 ± 0.02)	(63 ± 5)	(67 ± 7)	(1.43 ± 0.07)
Tarea 3	(1.08 ± 0.01)	(119 ± 7)	(31 ± 3)	(1.44 ± 0.06)
Tarea 4	(1.10 ± 0.03)	(113 ± 7)	(40 ± 4)	(1.42 ± 0.11)
Tarea 5	(1.12 ± 0.02)	(58 ± 4)	(64 ± 5)	(1.38 ± 0.11)
Tarea 6	(1.10 ± 0.01)	(58 ± 4)	(59 ± 8)	(1.36 ± 0.13)
Tarea 7	(1.10 ± 0.02)	(120 ± 4)	(28 ± 5)	(1.43 ± 0.08)
Tarea 8	(1.08 ± 0.03)	(114 ± 4)	(37 ± 5)	(1.43 ± 0.08)

Tabla 5.2: Resumen de los valores promedios de la aceleración, campo magnético, velocidad angular y pulso cardíaco de los grupos sanos y patológicos, por cada tarea.

- **Sincronización del movimiento:** Se observó que la trayectoria del movimiento pudo ser recuperada gracias al magnetómetro. La sincronización para cada sujeto estudiado confirma que la forma de realizar el movimiento es diferente para cada uno, incluso si se sigue un patrón establecido. Ilustró la complejidad del tema estudiado y la necesidad de incluir otras herramientas de análisis más avanzadas para la diferenciación entre los sujetos sanos y los pacientes con enfermedad de Parkinson.
- **Espectro de Fourier por parámetro y por tarea:** Se aplicó sobre el módulo de cada parámetro (aceleración, campo magnético, velocidad angular y pulso cardíaco) para ver si existía una composición frecuencial que marcara diferencias entre los sujetos. Para todos los casos estudiados se encontró un solapamiento entre los dos grupos, por lo que se confirma la necesidad de otros métodos para la separación de clases.

Uno de los problemas que presentan las bases de datos para usos médicos es la alta dimensionalidad, por lo tanto se optó a la aplicación de análisis de componentes principales (PCA) para la reducción de la misma. Esta técnica mejora la complejidad computacional aportando más rapidez en el proceso. Se estandarizaron los datos, se redimensionó la longitud de la data y se estableció 150 autovalores. A partir de la correlación de los datos se obtuvo una matriz de confusión con una eficiencia de clasificación del 72 % para 45 sujetos sanos, la base de conocimiento obtenida,

previamente etiquetada, se empleó como entrada al algoritmo clasificador basado en redes neuronales.

Se seleccionó una red neuronal basada en el algoritmo de propagación hacia atrás. La arquitectura de la red estuvo compuesta de 4 capas, una tasa de aprendizaje de 0.0225 y 500 *epoches*. Se tomó un porcentaje del 70 % de la base de conocimiento para el entrenamiento y una vez culminado se construyó la matriz de confusión con una eficiencia de clasificación del 100 %. Se obtuvo la función de la red y se guardaron los parámetros de la misma (pesos y *bias*) para la proyección del grupo de los pacientes con la enfermedad de Parkinson.

Una vez proyectados se utilizaron dendrogramas como una técnica de visualización para la separación de las clases. El proceso de entrenamiento de la red y proyección de los pacientes fue repetido cada vez para una tarea diferente y algunas combinaciones entre ellas. Los porcentajes de eficiencia por tarea fueron mostrados en la tabla 4.15, donde la mejor clasificación (100 %) se obtuvo para todas las tareas y todos los parámetros, como se mostró en las figuras 4.57 y 4.58 donde se evidencian los dos *clusters*. Además, haciendo uso del magnetómetro también se logró una clasificación de los dos grupos, lo que puede establecer otro camino de investigación.

Por último, se proyectó el grupo prueba sobre los parámetros de la red guardados, de igual manera a como se hizo para los pacientes con Parkinson. Este grupo compuesto por 15 sujetos sanos entre 25 y 67 años, se graficó el dendrograma de clasificación para cada uno y se obtuvo que 11 sujetos (73.3 %) quedaron por debajo de la línea de clasificación, lo que indica que sus características son muy similares al grupo prueba y los 4 restantes (26.6 %) que quedaron por encima de la línea de clasificación poseen características dinámicas similares al grupo patológico, por lo que estos sujetos que actualmente no presentan algún signo o síntoma correspondiente a la enfermedad de Parkinson son potenciales a desarrollarla en el futuro.

5.2. Comparativa con otros trabajos.

Todos los trabajos mostrados a continuación tuvieron como objetivo el diagnóstico de la enfermedad de Parkinson mediante el empleo de redes neuronales y técnicas de procesamiento de datos:

[Belen A., (2019)]

- Atributo estudiado: Registro de la voz.
- Red neuronal que emplea la librería Keras.
- Eficiencia para la base de datos procesados: 71.1 % datos aleatorios y 40.5 % datos continuos.
- Eficiencia para la base de datos en crudo: 62.3 % datos aleatorios y 57.3 % datos continuos.

[Sivaranjini S. y Suatha C., (2019)]

- Atributo estudiado: Imágenes de resonancia magnética.
- Red neuronal *AlexNet*.
- Eficiencia obtenida: 88.9 %

[Caliskan et al., (2017)]

- Atributo estudiado: Registro de voz.
- Red neuronal que combina *Autoencoder (AE)* y *Softmax Classifier*.
- Pre-procesamiento, extracción de características y clasificación de datos.
- Eficiencia obtenida: 93.7 %

[Rustempasic I. y Can M., (2013)]

- Atributo estudiado: Registro de voz.
- Red neuronal: Perceptrón multicapa de propagación hacia atrás.
- Técnicas de filtrado y análisis de componentes principales.
- Eficiencia obtenida: 92 %

[Soleimanian F. y Mohammaid P., (2013)]

- Atributo estudiado: Registro de voz.
- Dos redes neuronales: Perceptrón multicapa de propagación hacia atrás (MLP) y redes neuronales con función de base radial (RBF).
- Eficiencia obtenida: MLP: 93.2 % y RBF: 86.4 %

[Geman et al., (2013)]

- Atributo estudiado: Señales de la marcha y el temblor.
- Análisis lineal y no lineal de la data, análisis de Fourier.
- Algoritmo K-means.
- Eficiencia obtenida: 98 %

[Khemphila A, (2012)]

- Atributo estudiado: Registro de voz.
- Perceptrón multicapa de propagación hacia atrás (MLP).
- Eficiencia obtenida antes y después de la reducción de las características: 80.7 % y 83.3 %

Los sistemas expertos de ayuda constituidos por la información del paciente (base de datos) y los algoritmos de inteligencia artificial, han logrado importantes contribuciones en el diagnóstico y seguimiento de enfermedades. Los trabajos presentados en relación con el diagnóstico de la enfermedad de Parkinson empleando algoritmos de inteligencia artificial y análisis de datos, mostraron eficiencias altas. El sistema propuesto tuvo una eficiencia del 100 % de clasificación cuando se emplearon todas las tareas y parámetros, aun cuando no hubo un trabajo que se basara en el estudio de la actividad motora de las extremidades superiores, la metodología seguida puede compararse con la mayoría.

5.3. Conclusiones.

Los métodos actuales que permiten el estudio de la Enfermedad de Parkinson están orientados a estudiar su evolución una vez ésta ha sido diagnosticada, si bien este es un hecho muy importante, pues permite valorar la verificación positiva o no de los tratamientos, sin embargo no se adelanta en la aparición temprana usando parámetros medibles como por ejemplo el comportamiento de la actividad motora, usando tareas preestablecidas. Existen pocas investigaciones basadas en la motricidad que permitan identificar algún indicio de la enfermedad en etapas tempranas.

En este estudio preliminar se establece una metodología en la adquisición, pre y post-procesamiento, visualización, análisis y clasificación de los datos. Por otra parte, se establecen criterios de extracción de ruido y de data correlacionada, basados en el uso de PCA, y se potencia la clasificación introduciendo el uso de RNA, así también se evidencia la separabilidad de clases usando la distancia de Mahalanobis, que le adjudica robustez al método al hacerlo invariante a transformaciones afines tales como, traslaciones, escalado y a ruido en los datos, permitiéndole por tanto alta confiabilidad en cuanto a las membresías de las clases obtenidas.

Las enfermedades complejas están determinadas por la interacción de múltiples factores como los genéticos, ambientales e incluso el estilo de vida. Con la incorporación de los avances en tecnologías de análisis de datos, en especial el uso de algoritmos de inteligencia artificial, se ha logrado una mejora en el proceso de diagnóstico y tratamiento de varias enfermedades. En este estudio se propuso un sistema no invasivo para el diagnóstico en las primeras etapas de la enfermedad de Parkinson a través de la dinámica corporal del paciente.

Este sistema constituido por un acelerómetro, un giróscopo, un magnetómetro y un pulsómetro, todos conectados a una tarjeta Arduino Due permitió la adquisición de datos del paciente cuando realizó diversas tareas siguiendo patrones circulares y lineales. Se recogieron los datos de 60 sujetos sanos y se construyó la base de conocimiento una vez realizado el tratamiento de la data, desde la estandarización de los datos hasta el análisis de componentes principales. Con esta base fue entrenada una red neuronal basada en un algoritmo de propagación hacia atrás, con una arquitectura de cuatro capas y a partir de los parámetros de la red, pesos y *bias*, se hizo la proyección del grupo de los 6 pacientes con la enfermedad de Parkinson en sus primeros estadios.

A través de herramientas de visualización, en este caso los dendrogramas, se presentó la clasificación del grupo patológico por cada tarea y algunas combinaciones entre ellas, se obtuvo para todas las tareas y todos los parámetros una clasificación del 100 %. Por otra parte, siguiendo la metodología del análisis de la dinámica corporal por medios clásicos, se encontró que no hubo una clasificación entre los grupos en términos de frecuencias y de la media de los parámetros estudiados. Sin embargo, mostró que el magnetómetro puede emplearse individualmente para la clasificación, esto fue confirmado posteriormente empleando el algoritmo de clasificación .

Finalmente, a partir de los parámetros de la red se hizo la proyección de sujetos desconocidos (que no fueron presentados a la red) y continuando la metodología propuesta, se observó que el algoritmo introduce una clasificación del 73.3 % como sujetos que poseen características similares al grupo sano, mientras que el restante los ubica en la zona de sujetos potenciales, teniendo estos características similares al grupo patológico. A pesar de que en la literatura revisada no se

hallaron trabajos relacionados directamente con el tema de investigación, los resultados de este estudio permiten evidenciar la separabilidad de clases en sujetos normales de los que padecen la enfermedad, estamos lejos aún de constatar que los sujetos potenciales puedan padecer Parkinson en algún momento de su vida, dado que habría que hacer un seguimiento durante mucho tiempo, lo que si se pudo constatar es que existen diferencias medibles importantes entre sujetos sanos y sujetos enfermos, extraíbles de su motricidad controlada por las tareas preestablecidas, quedará por valorar su evolución futura como una herramienta cuantitativa de ayuda en la práctica médica.

www.bdigital.ula.ve

5.4. Recomendaciones.

Se recomiendan los siguientes aspectos:

- Ampliar la base de datos de los sujetos sanos para lograr una mayor extracción de las características dinámicas de este grupo y así tener un grupo de entrenamiento más variado.
- Realizar el test a más pacientes diagnosticados con la enfermedad de Parkinson en los estadios I y II, de esta manera se lograría tener un algoritmo más robusto y con mayor eficiencia.
- Instruir al paciente antes de realizar las evaluaciones de la mejor manera posible, explicar las tareas que debe realizar, la postura que debe mantener; para asegurar que los datos sean adquiridos de forma correcta.
- Incluir parámetros de la voz que puedan extraerse de una breve grabación del sujeto diciendo una frase corta preestablecida o las vocales sostenidas durante un tiempo.

5.5. Futuras líneas de investigación.

Como continuación de este trabajo se proponen lo siguiente:

- El estudio de la dinámica corporal basado en parámetros dinámicos no lineales. Se sugiere el uso de herramientas como TISEAN (Nonlinear Time Series Analysis) o CDA (*Chaos Data Analyzer*), con estos software se puede analizar: la probabilidad de distribución, diagramas de bifurcación, frecuencias dominantes, exponente de Lyapunov, dimensión fractal y secciones de Poincaré. De esta manera se podrían encontrar indicios de diferencias o similitud entre los grupos sanos y patológicos que puedan apoyar en el diagnóstico de la enfermedad de Parkinson.
- El diseño de una aplicación que contenga las tareas y pueda registrar los movimientos realizados con el *mouse*, de modo que cualquiera con acceso a una computadora e internet pueda realizar el test y de esta manera se lograría una base de datos muy amplia y un algoritmo más robusto para la detección temprana de la enfermedad.
- Con la metodología propuesta se puede aplicar a la detección de otras enfermedades que involucren una disminución en las capacidades motoras, siguiendo patrones de locomoción o movilidad.

Apéndice A

Programas Matlab.

A.1. Concatena los archivos txt.

```
% CONCATENA ARCHIVOS TXT
% =====
% DESCRIPCIÓN: Este programa sirve para concatenar los archivos en formato .txt que
% son generados cuando se inician los sensores: acelerómetro, magnetómetro, giroscopo
% y pulsómetro a través de la aplicación "aqevent"
% ENTRADAS:
% Se leerán todos los archivos .txt del directorio que se quiere analizar, para ello
% es necesario tener todos los archivos en una carpeta. Los archivos serán ordenados
% por la fecha de creación desde el más antiguo hasta el más reciente y estos serán
% concatenados en uno solo.
% SALIDA:
% Crea y guarda un archivo que contiene la variable DATA
% -DATA: Representa la unión de todos los archivos, es una matriz de n x 10 elementos
% =====
% INICIO
% =====
% DIRECTORIO
% -----
cd /home/porthos/Escritorio/Stephanie/Aqevent
Paciente = '1';
d = dir ('*.txt'); % Lee todos los archivos .txt existentes en la carpeta.
t = [];
for i = 1:size(d,1)
    t = [t;d(i).date]; % Da el tiempo en que se creó el archivo.
end
N = {};
T = datenum(t); % Ordena de menor a mayor la fecha de creación.
[To,jo] = sort(T); % Ordena el arreglo anterior.
Tf = datestr(To); % Visualiza el arreglo ordenado en el formato DMYHMS
for i=1:size(d,1)
    N(i).N = d(jo(i)).name; % Carga los archivos ya ordenados
end
DATA = [];
for i=1:size(d,1)
    data = [];
    if i == 1
        data = textread(d(i).name, '%s', 'delimiter', ',', 'headerlines', 2);
        % Elimina la cabecera del primer archivo
```

```

else
    data = textread(d(i).name,'%s','delimiter',' ');
end
    % Elimina la primera y última fila de datos ya que no siempre se cargan.
    % Si no se hace hay puede haber incompatibilidad en los datos.
    n1 = 2; n2 = 1;
    data = data(n1:end-n2,:);
    data = str2num(char(data));
    DATA = [DATA;data]; % Archivos concatenados finalmente.
end
ho = 1;
ax = DATA(:,1); ay = DATA(:,2); az = DATA(:,3); % Acelerómetro.
mx = DATA(:,4); my = DATA(:,5); mz = DATA(:,6); % Magnetómetro.
gx = DATA(:,7); gy = DATA(:,8); gz = DATA(:,9); % Giroscopo.
t = 1:length(ax);
dt = (datenum(Tf(end,:)) - datenum(Tf(1,:)))*24*3600;
Fs = length(ax)/dt;
t = linspace(0,dt,length(DATA)); % Tiempo en segundos.

% VISUALIZACIÓN
% =====
subplot(5,1,1)
plot(t,smooth(ax-0*mean(ax),ho),'r',t,smooth(ay-0*mean(ay),ho),'g',t,...
    ...smooth(az-0*mean(az),ho),'b'),grid
title('Acelerometro:ax:rojo,ay:verde,az:azul','fontsize',[12],'fontweight','bold')
ylabel('a (cm/s2)','fontsize',[12],'fontweight','bold')
axis([t(1) t(end) -1.5 2])
subplot(5,1,2)
plot(t,smooth(mx-0*mean(mx),ho),'r',t,smooth(my-0*mean(my),ho),'g',t,...
    %xlabel('Time (Sec)','fontsize',[12],'fontweight','bold')
    ...smooth(mz-0*mean(mz),ho),'b'),grid
title('Magnetometro:mx:rojo,my:verde,mz:azul','fontsize',[12],'fontweight','bold')
ylabel('campo (gauss)','fontsize',[12],'fontweight','bold')
axis([t(1) t(end) -250 250])
subplot(5,1,3)
plot(t,smooth(gx-0*mean(gx),ho),'r',t,smooth(gy-0*mean(gy),ho),'g',...
    ...t,smooth(gz-0*mean(gz),ho),'b'),grid
title('Giroscopo: gx:rojo, gy:verde, gz:azul','fontsize',[12],'fontweight','bold')
ylabel('velocidad (cm/s)','fontsize',[12],'fontweight','bold')
axis([t(1) t(end) -200 350])
subplot(5,1,4)
PULSO=DATA(:,10);
plot(t,PULSO)
axis([t(1) t(end) 400 1000])
subplot(5,1,5)
[P,f,Y,Fmax] = Espectro_de_Fourier_De_Senal_Arbitraria_Modificado(Fs,PULSO,1);
cd /home/porthos/Escritorio/Stephanie
cadena = [ 'save ' ' Data_Paciente_Prueba_' num2str(Paciente) ' DATA'];
eval([cadena]);
% FIN
% =====

```

A.2. Análisis Componentes Principales.

```
% ANÁLISIS DE COMPONENTES PRINCIPALES
```

```

% =====
% DESCRIPCIÓN: Este programa hace un análisis de componentes principales de la data
% que se le presenta. Esta data se estandariza y se divide en dos: data de entrena-
% miento y data test. Finalmente se presenta la matriz de confusión y la data con
% sus respectivos patrones.
% ENTRADAS:
% ind_opt: Se toma de la optimización.
% Nt: Número de tareas.
% Fs: Frecuencia de muestreo.
% Dt: Tiempo en segundos el que se divide la ventana total de datos.
% pe: Porcentaje de entrenamiento.
% Na: Número de autovalores a calcular.
% sigma: Umbral de depuración de los datos .
% band: 0, para visualizar la salida.
% SALIDAS:
% Base_Paciente: Una estructura que contiene lo siguiente:
% - Pcf: La data con sus respectivos patrones.
% - Pcf_ent: Data de entrenamiento con sus respectivos patrones.
% - Pcf_test: Data de reconocimiento con sus respectivos patrones.
% - ao,af: Posición de los atributos iniciales y finales.
% - eo,ef: Posición de las etiquetas iniciales y finales.
% - Paciente: Cadena de los pacientes utilizados.
% Proj : Contiene la proyección de los pacientes durante la etapa de entrenamiento.
% V: Autovectores.
% D: Autovalores.
% Na: Número de los autovalores utilizados.
% I_test: Arreglo de etiquetas dadas y encontradas durante el reconocimiento.
% EJEMPLO
% [Base_Paciente,Paciente,Proj,V,D,Na,I_test] =
%     = Analisis_Componentes_Principales(15,[1:8],60,10,70,150,300,1);
% =====
% INICIO
% =====
function [Base_Paciente,Paciente,Proj,V,D,Na,I_test] =
    = Analisis_Componentes_Principales(ind_opt,Nt,Fs,Dt,pe,Na,sigma,band);

test = 0; if test == 0

% DATA SIMULADA
% =====
% [NP] = Simula_Data_Parkinson_Modificado(Np,Nt,Fs,dt);
% [Np] = Simula_Data_Parkinson_Modificado(3,8,60,100);
% P = [];
% Paciente = [];
% for i = 1:Np
%     P = {num2str(i)};
%     num2cell(P);
%     Paciente = [Paciente P];
% end
% =====
Paciente = { 'Gustavo' 'Sandra' 'Ignacio' 'Maria' 'Neptali' 'Jesus' 'Diego' 'Oswaldo' };
Np = length(Paciente); % Número de pacientes

% Se redimensionan los datos a una misma longitud.
% -----
[Dr,Nv,dt,Num_par] = Redimensiona_los_datos(Paciente,ind_opt,Nt,Fs,Dt,sigma,1);

```

```

% -----
% Se crea la matriz con todos los ejemplos y se divide la data de entrenamiento y
% de reconocimiento
% -----
[A,E,Ae,At,Pe,Pt,Im_P,Var_P,I_P,Im_Pac] = Formato_de_Datos_Renombrado(Np,Nt,...
...Nv,Num_par,Dr,pe,1);
% -----
% SE ESTANDARIZA LA DATA DE ENTRENAMIENTO PARA REALIZAR PCA
% -----
    G = Ae'; % Matriz de entrenamiento
    nt = size(G,1);
    M = mean(G); Mm = repmat(M,nt,1);
    S = std(G); Ms = repmat(S,nt,1);
    G = (G - Mm)./Ms;
% SE CALCULA LA MATRIZ DE COVARIANZA
% -----
    C = cov(G);
    S = diag(C);
% SE CALCULAN LOS AUTOVALORES Y AUTOVECTORES DE LA MATRIZ DE COVARIANZA
% -----
    [V,D] = eigs(C,Na,'lm');
    Parametros = struct('V',V,'D',D,'Na',Na);
    save Parametros
% SE ORDENAN LOS AUTOVALORES
% -----
    Lamda = sort(diag(D),'descend');
    Lamda_1 = (Lamda);
% SE VISUALIZAN LOS AUTOVALORES Y LA DESVIACIÓN ESTÁNDAR DE LOS DATOS
% -----
    t = linspace(1,size(G,1),size(G,1))*dt;
if band == 1
    xa = 1:Na;
    xe = randperm(size(G,2));
    xs = 1:length(S);
    subplot(2,1,1)
        plot(xs,S,'-r',xs,S,'.b','linewidth',[1],'markersize',[5]),grid
        xlabel('Parámetro dinámico'),ylabel('Magnitud de su desviación')
        title('Desviación estándar')
        axis([0 xs(end) 0 3])
    subplot(2,1,2)
        plot(xa,Lamda_1,'-r',xa,(Lamda_1),'b','linewidth',[2],'markersize',[30]),grid
        axis([0 Na 0 1.2*max(real(Lamda_1)) ])
        xlabel('Número del autovalor'),ylabel('Magnitud del autovalor')
end
% FINALIZACIÓN DEL CÁLCULO DE LAS COMPONENTES PRINCIPALES (PCA)
% =====
% PROYECCIÓN DE LOS PACIENTES PROMEDIOS EN EL NUEVO ESPACIO
% -----
    [m,n,k] = size(Im_P(:,:,1)');
if test == 0
    Im = [];
    for i = 1:length(Paciente)
        Imedia = Im_P(:,:,i);
        Im(:,:,i) = Imedia;
    end
% VISUALIZACIÓN

```

```

% -----
Proj = []; Xp = []; Yp = [];
for i = 1:Np
    I = reshape(Im(:,:,i),m*n,1);
    p = I'*V;           % Proyección.
    Proj(:,:,i) = p;
    px = I'*V(:,1:2);
    Xp = [Xp;px(1) i];
    Yp = [Yp;px(2) i];
    if band == 1
        subplot(5,round((Np/5)+0.3),i)
        bar(p),title(['Proj: ' Paciente{i}]),grid
        axis([0 Na-100 -4 4])
    end end
    save Proyeccion Proj % Guarda la proyección de los pacientes.
end end

test = 1; if test == 1
Etiqueta = Etiquetas;
% TEST CON PACIENTES NUEVOS
% -----
if test == 1
    I_test = [];
% MATRIZ DE DATOS PARA EL RECONOCIMIENTO
% -----
I_P = I_P; % Ejemplos de cada paciente para el reconocimiento.
w = 5; % Número total de ejemplos por paciente.
% SELECCIÓN DE UNA IMAGEN DE CADA PACIENTE
% -----
Im_Pac = mean(I_P);
I_PP = repmat(Im_Pac,w,1);
PROJ = [];
for k = 1:w
    Imp = [];
    for i = 1:length(Paciente)
        Imedia = I_PP(k,:,i);
        Imp(:, :, i) = Imedia;
    end
% PROYECCIÓN EN EL ESPACIO REAL (RECONOCIMIENTO)
% -----
X = []; Y = [];
for i = 1:length(Paciente)
    Ix = reshape(Imp(:, :, i),1,m*n);
    proj_x = Ix*V; % Proyección del paciente desconocido.
    x = Ix*V(:,1:2);
    proj_xr = repmat(proj_x,size(Proj,3),1);
    ac = squeeze(Proj);
    bc = proj_xr';
% Distancias
% -----
% Coeficiente de correlación
    e = [];for ii=1:size(ac,2);co = corrcoef(ac(:,ii),bc(:,1));e(ii)=co(1,2); end
% Distancia Euclídea
% e = sqrt(sum((ac - bc).^2))/Na;
% Momento
% e = moment(ac-bc,5);

```

```

% Distancia de Mahalanobis
% e = [];for ii=1:size(ac,2);co = sum(mahal(ac(:,ii),bc(:,1)).^2);e(ii)=co(1); end
% e = e/Na;

[im,jm] = max(abs(e));
X = [X;x(1) i jm];
Y = [Y;x(2) i jm];
if band == 1
    subplot(5,round((Np/5)+0.3),i)
    bar(e),title([ Paciente{i} '<=>' Paciente{jm}], 'color',[1 0 0]),grid,end
    axis([0 Np+1 -0.3 0.3])
    I_test = [I_test; i jm]; % Se guardan los índices
    Etiqueta(i,:);
    PROJ = [PROJ;proj_x Etiqueta(i,:)];
end end end

% CONFIABILIDAD DEL TEST (ROBUSTEZ)
% =====
%MATRIZ DE CONFUSIÓN:Determina la coincidencia de las etiquetas dadas vs encontradas.
% -----
if test == 1
    Ind = []; Conf_Ind = [];
    v = length(Paciente);
    for i = 1:v
        ind_a = [];
        for j = 1:v
            ind = find(I_test(:,1)==i & I_test(:,2)==j);
            Ind = [Ind; length(ind) i j];
            ind_a(j) = length(ind);
        end
        Conf_Ind = [Conf_Ind;ind_a];
    end

% COEFICIENTES DEL POLINOMIO
% -----
[coef,st] = polyfit(I_test(:,1),I_test(:,2),1);
if (band == 0 | band == 1) & test == 1
    xlabel('Etiquetas dadas','fontsize',[11],'fontweight','bold','color','k')
    ylabel('Etiquetas encontradas','fontsize',[11],'fontweight','bold','color','k')

% Representación de las etiquetas.
K = length(Paciente);
L = K+1;
f = Paciente;
g = get(figure(5));
go = g.Children;
x = linspace(1,K,length(f));
set(go(1),'XTick',x,'XTickLabel',f)
set(go(1),'YTick',x,'YTickLabel',f)
axis([0 L 0 L])
Ax = []; Ay = [];
for i = 1:length(Ind)
    a = Ind(i,:,:);
    ax = a(2);
    ay = a(3);
    text(ax+0.25,ay,num2str(a(1)),'fontsize',[11],'fontweight','bold','color','b');

```

```

end
line([1 K],[1 K], 'linewidth',[2], 'color','k');
xp = linspace(0,L,L);
yp = polyval(coef,xp);
hold on,plot(xp,yp, '--m', 'linewidth',[4]),hold off
end end
pause(1), warning off

% Se normalizan los datos y se guardan las proyecciones para utilizarse en el
% entrenamiento de redes
% -----
% Posición de los atributos.
    ao = 1; af = Na;
% Posición de las etiquetas.
    eo = Na + 1; ef = eo + 4;
    Base_Paciente = {};
    if test == 1
% Normalización
        Po = PROJ(:,1:Na);
        Max_Po = max(abs(Po(:)));
        Pn = Po/Max_Po;
% Se agregan las etiquetas a cada proyección.
% -----
        Pcf = [Pn PROJ(:,eo:ef)]; % Patrones
% Se divide la data de entrenamiento y test.
% -----
        por_ent = pe;
        [Pcf_ent,Pcf_test,Ind] = Data_De_Entrenamiento_y_Test(Pcf,eo,ef,Paciente, por_ent,1);
% SALIDAS
% =====
        Base_Paciente = [];
        Base_Paciente = struct('Pcf',Pcf,'Pcf_ent',Pcf_ent,'Pcf_test',Pcf_test,'ao',ao,...
                                ... 'af',af,'eo',eo,'ef',ef,'Paciente',Paciente);
        save Base_De_Datos_Paciente Base_Paciente
    end end
% FIN
% =====

```

A.3. Redimensiona los datos.

```

% REDIMENSIONA LOS DATOS
% =====
% DESCRIPCIÓN: Función que permite redimensionar todos los datos registrados a
% una misma longitud.
% ENTRADAS:
%     Paciente: Define el paciente a estudiar (ej: 1, 2, 3,...).
%     ind_opt: Índice óptimo (se toma de la optimización).
%     Nt: Numero de tareas (movimiento de manos, circulos, rectas diagonales).
%     Fs: Frecuencia de muestreo.
%     Dt: Tiempo en segundos el que se divide la ventana total de datos.
%     sigma: Umbral de depuración de los datos
%     band: 1, para visualizar la salida.
% SALIDAS:
%     Dr: Datos finalmente redimensionados y normalizados.
%     Nv: Número de ventanas en que se divide la data.

```

```

%      dt: Tiempo en segundos.
%      Num_par: Número de parámetros elegidos.
% EJEMPLO:
% =====
%[Dr,Nv,dt,Num_par] = Redimensiona_Datos_4(Paciente,ind_opt,Nt,Fs,Dt,sigma,band);
% =====
% INICIO
% =====
function
[Dr,Nv,dt,Num_par] = Redimensiona_Datos_4(Paciente,ind_opt,Nt,Fs,Dt,sigma,band);

% Solo deja los parámetros seleccionados
% -----
[IND,Po,po] = Indices;
    IND = IND(ind_opt,:);
    P = {'ax','ay','az','bx','by','bz','wx','wy','wz','p'};
    Ind = find(IND == 0);
    P(Ind) = [];
    Num_par = size(P,1);

% Se determina la dimensión mínima de los datos para todos.
% -----
cd /home/porthos/Escritorio/Programas_Utilizados_General
ni = [];
for j = 1:length(Paciente)
    DATA = []; nx = [];
    for i = 1:length(Nt)
        eval(['load Data_Paciente_' cell2mat(Paciente(j)) '_' num2str(i)]);
        DATA(:,Ind) = []; nx(i) = size(DATA,1);
    end
    ni(j) = min(nx);
end
m = min(ni); % Se toma el de la mínima dimensión.
% -----
% Se normaliza a media cero y desviación estándar uno
% -----
Dp3 = []; Dp2_2 = [];
for j = 1:length(Paciente)
    DATA = []; d = []; Dp1 = []; Dp1_1 = [];
    for i=1:length(Nt)
        eval(['load Data_Paciente_' cell2mat(Paciente(j)) '_' num2str(i)]);
        DATA(:,Ind) = [];
        d = [d;DATA(1:m,:)]; Dp1(:, :, i) = DATA(1:m,:); [p,q] = size(Dp1(:, :, i));
        M = mean(Dp1(:, :, i)); Mm = repmat(M,p,1);
        S = std(Dp1(:, :, i)); Ms = repmat(S,p,1);
        Dp1_1(:, :, i) = (Dp1(:, :, i) - Mm)./Ms; end
    Dp3(:, :, :, j) = Dp1;
    Dp2_2(:, :, :, j) = Dp1_1; % Matriz normalizada escrita en 4 dimensiones.
end
H = Dp2_2;

% Se depuran los datos
% -----
    Depura_los_datos(H,ind_opt,sigma,1);
% Se determina la dimensión mínima de los datos para todos los pacientes y para
% todas las tareas de la data ya depurada

```

```

% -----
ni = [];
for j = 1:length(Paciente)
    DATA = []; n = [];
    for i=1:length(Nt)
        eval(['load Data_Depurada_Paciente_' num2str(j) '_Tarea_' num2str(i)]);
        n(i) = size(DATA,1); end
    ni(j) = min(n); end
m = min(ni);

% Se normaliza a media cero cero y desviación estándar uno.
% -----
Dp3 = []; Dp2_2 = [];
for j=1:length(Paciente)
    DATA = []; d = []; Dp1 = []; Dp1_1 = [];
    for i=1:length(Nt)
        eval(['load Data_Depurada_Paciente_' num2str(j) '_Tarea_' num2str(i)]);
        d = [d;DATA(1:m,:)]; Dp1(:, :, i) = DATA(1:m,:); [p,q] = size(Dp1(:, :, i));
        M = mean(Dp1(:, :, i)); Mm = repmat(M,p,1);
        S = std(Dp1(:, :, i)); Ms = repmat(S,p,1);
        Dp1_1(:, :, i) = (Dp1(:, :, i) - Mm)./Ms; end
    Dp3(:, :, :, j) = Dp1;
    Dp2_2(:, :, :, j) = Dp1_1; % Matriz normalizada escrita en 4 dimensiones
end

% Matriz normalizada por paciente y por tarea.
% -----
G = Dp2_2;
nt = length(G);
% Se determina el número de ventanas a partir del tiempo elegido para cada una.
% -----
dt = 1/Fs;
Tt = m*dt; % m : longitud de la data.
Nv = round(Tt/Dt);
if Nv == 0, Nv = 1; end
% Se verifica que la longitud de los datos sea múltiplo del número de ventanas.
% -----
for j = 1:100000
    if rem(nt-j,Nv)== 0, ns=j; break, end
end
Ds = G(1:end - ns, :, :, :);
[rs,qs,ss,ts] = size(Ds);
% Se redimensionan los datos
% -----
Dr = reshape(Ds,rs/Nv,qs*Nv,ss,ts);

% FIN
% =====

```

A.4. Combinación entre parámetros.

```

% COMBINACIÓN ENTRE PARÁMETROS
% =====
% DESCRIPCIÓN: Establece un arreglo entre las n posibles combinaciones de los
% parámetros utilizados.

```

```

% SALIDAS:
%   IND: Arreglo de las combinaciones entre los parámetros utilizados.
%   Po: Nombre de las combinaciones entre los parámetros utilizados.
%   po: Posición de las combinaciones entre los parámetros utilizados.
% EJEMPLO:
% =====
%   [IND,Po,po] = Indices;
% =====
% INICIO
% =====
function [IND,Po,po] = Indices;
% Combinaciones entre parámetros
% -----
IND = [1 1 1 0 0 0 0 0 0 0; % ax ay az
       0 0 0 1 1 1 0 0 0 0; % bx by bz
       0 0 0 0 0 0 1 1 1 0; % wx wy wz
       0 0 0 0 0 0 0 0 0 1; % p
       1 1 1 1 1 1 0 0 0 0; % ax ay az bx by bz
       1 1 1 0 0 0 1 1 1 0; % ax ay az wx wy wz
       1 1 1 1 1 1 0 0 0 1; % ax ay az bx by bz p
       1 1 1 0 0 0 1 1 1 1; % ax ay az wx wy wz p
       0 0 0 1 1 1 1 1 1 0; % bx by bz wx wy wz
       1 1 1 0 0 0 0 0 0 1; % ax ay az p
       0 0 0 1 1 1 1 1 1 1; % bx by bz wx wy wz p
       0 0 0 1 1 1 0 0 0 1; % bx by bz p
       0 0 0 0 0 0 1 1 1 1; % wx wy wz p
       1 1 1 1 1 1 1 1 1 0; % ax ay az bx by bz wx wy wz
       1 1 1 1 1 1 1 1 1 1]; % ax ay az bx by bz wx wy wz p % todas

Po = {'Aceleración','Campo magnético','Velocidad angular','Pulso';...
      'Aceleración y Campo magnético','Aceleración y Velocidad angular';...
      'Aceleración, Campo magnético y pulso','Aceleración, Velocidad angular y Pulso';...
      'Campo magnético y Velocidad angular','Aceleración y Pulso';...
      'Campo Magnético, Velocidad angular y Pulso';...
      'Campo magnético y Pulso','Velocidad angular y Pulso';...
      'Aceleración, Campo Magnético y Velocidad angular','Todos los parámetros'};

po = {'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15'};
% FIN
% =====

```

A.5. Depuración de datos.

```

% DEPURACIÓN DE DATOS
% =====
% DESCRIPCIÓN: Depura los datos según un valor de sigma dado a partir de la data ya
% normalizada por paciente y por tarea.
% ENTRADAS:
%   G: Data original normalizada.
%   ind_opt: Índice óptimo (se toma de la optimización).
%   sigma: Umbral de depuración de los datos
%   band: 1, para visualizar la salida.
% SALIDAS:
%   Crea y guarda un archivo que contiene la data ya depurada por paciente
%   y por tarea.

```

```

% EJEMPLO:
%   Depura_Los_Datos_Modificado(G,ind_opt,sigma,band);
%   Depura_Los_Datos_Modificado(G,5,3,1);
% =====
% INICIO
% =====
function Depura_Los_Datos_Modificado(G,ind_opt,sigma,band);
[p,l,m,n] = size(G);
if ind_opt == 15
    for i =1:n
        for j = 1:m
            g = G(:,1:3,j,i); ax = g(:,1); ay = g(:,2); az = g(:,3);
            h = G(:,4:6,j,i); bx = h(:,1); by = h(:,2); bz = h(:,3);
            l = G(:,7:9,j,i); wx = l(:,1); wy = l(:,2); wz = l(:,3);
            q = G(:,10,j,i); p = q(:,1);
            % Eliminación de ruido
            for v = 1:4
                % Módulos
                a = sqrt(ax.^2 + ay.^2 + az.^2);
                b = sqrt(bx.^2 + by.^2 + bz.^2);
                w = sqrt(wx.^2 + wy.^2 + wz.^2);
                % Estandarización de los módulos
                a = (a-mean(a))/std(a);
                b = (b-mean(b))/std(b);
                w = (w-mean(w))/std(w);
                p = (p-mean(p))/std(p);
                K = [a b w p ];
                % Umbral
                ao = std(a)*sigma;
                bo = std(b)*sigma;
                wo = std(w)*sigma;
                po = std(p)*sigma;
                Ko = [ao bo wo po];
                % Búsqueda de los índices
                Ind = find(abs(K(:,v)) >= Ko(v));
                lx = 1:length(K); ly = Ko(v)*ones(1,length(K));
                plot(lx,K(:,v))
                if band == 0
                    hold on, plot(lx(Ind),K(Ind,v),'*r'),plot(lx,ly,lx,-ly),
                    title(['Np: ' num2str(i) ' Nt: ' num2str(j) ' Par: ' num2str(v) ]);
                    hold off,pause ,end
                ax(Ind)=[]; ay(Ind)=[]; az(Ind)=[]; bx(Ind)=[]; by(Ind)=[]; bz(Ind)=[];
                wx(Ind)=[]; wy(Ind)=[]; wz(Ind)=[]; a(Ind)=[]; b(Ind)=[]; w(Ind)=[]; p(Ind)=[];
                end
                DATA = [ax ay az bx by bz wx wy wz p];
            cadena = [ 'save ' ' Data_Depurada_Paciente_' num2str(i) '_Tarea_' num2str(j) 'DATA'];
            eval([cadena]); end
        end
    end
% FIN
% =====

```

A.6. Formato de los datos.

```

% FORMATO DE LOS DATOS
% =====

```

```

% DESCRIPCIÓN: Función que genera la base de conocimiento de los datos para el
% análisis de componentes principales en pacientes con Parkinson.
% ENTRADAS:
%     Np: Número de pacientes en estudio.
%     Nt: Número de tareas (movimiento de manos, círculos, rectas).
%     Nv: Número de ventanas en que se ha dividido la data.
%     Num_par: Número de parámetros elegidos.
%     Dr: Data redimensionada y normalizada que contiene:
%     pe: Porcentaje de entrenamiento.
%     band: 0 para visualizar.
% SALIDAS:
%     A: Matriz m*n de datos que contiene:
%         m: Longitud del vector de datos.
%         n: número de parámetros * número de ejemplos.
%     E: Matriz i*j*k de datos que contiene:
%         i: paciente, j: tarea, k: ejemplo.
%     Ae: Data de entrenamiento.
%     At: Data de test.
%     Pe: Patrón promedio para entrenamiento.
%     Pt: Patrón promedio para el test.
%     Im_P: Imagen media de cada paciente.
%     Var_P: Varianza de cada paciente.
%     I_P: Conjunto de imágenes de cada paciente.
%     Im_Pac: Imagen media de cada paciente.
% EJEMPLO:
% [A,E,Ae,At,Pe,Pt,Im_P,Var_P,I_P,Im_Pac]=Formato_de_Datos(10,15,10,15,Dr,50,1);
% =====
% INICIO
% =====
function
[A,E,Ae,At,Pe,Pt,Im_P,Var_P,I_P,Im_Pac]=Formato_de_Datos(Np,Nt,Nv,Num_par,Dr,pe,band);

alea = 2; % (1) si es aleatorio en la elección de los ejemplos.
% Genera matriz general de datos
% -----
A = []; E = []; Ae = []; At = []; Pe = []; Pt = []; I1 = []; I2 = [];
m = size(Dr,2)/Num_par;
k1 = 1:m;
for i=1:Np % Pacientes
for j=1:length(Nt) % Tareas
if alea == 1
k1 = randperm(length(k1)); % El bus de datos es tomado aleatoriamente
end
for k = 1:m % Ejemplos
V = k1(k):Nv:size(Dr,2);
a = (Dr(:,[V],j,i));
A = [A;a(:)'];
E = [E; i j k]; end end end
B1 = A';

% Filtrare de los datos originales
% -----
B = []; ho = 3;
for i=1:size(B1,2),B(:,i) = smooth(B1(:,i),ho);end
% Separa data test y entrenamiento
% -----

```

```

xp = 100 - pe; % Porcentaje de entrenamiento
p3 = [];
for i = 1:Np
    p1 = []; p2 = [];
    for j = 1:length(Nt)
        ind = find(E(:,1)== i & E(:,2) == j);
        lp = round(xp*length(ind)/100);
        if length(ind) == lp,clc
            disp(['OJO: Aumente el porcentaje de entrenamiento,...']),return,end
        n1 = length(ind) - lp; n2 = n1 + 1;
        Ind1 = ind(1:n1);          % Data de entrenamiento.
        Ind2 = ind(n2:end);        % Data de test.
        I1 = [I1;Ind1]; I2 = [I2;Ind2];
        p1 = [p1;B(:,Ind1)']; p2 = [p2;B(:,Ind2)'];
    end
    % Genera pacientes promedio
    % -----
    pm1 = mean(p1); pm2 = mean(p2);
    Pe = [Pe;pm1]; % Paciente promedio entrenamiento.
    Pt = [Pt;pm2]; % Paciente promedio test.
end
% Salida data test y entrenamiento
% -----
Ae = B(:,I1);
At = B(:,I2);
Pe = Pe'; % Promedio entrenamiento.
Pt = Pt'; % Promedio test.
% Data de reconocimiento (media y varianza) agrupada por paciente
% -----
l = size(Ae,2)/Np;
V0 = 1:l:size(Ae,2); V1 = 0:l:size(Ae,2); V1(1)=[];
C = [];
for i = 1:Np
    C(:, :, i) = (Ae(:, [V0(i):V1(i)]))';
end
% Imagen media y varianza de cada paciente
% -----
Im_P = []; Im_Paciente = []; Var_P = []; Varianza = [];
for j = 1:Np
    Im_Paciente = mean(C(:, :, j)); Im_P(:, :, j) = Im_Paciente;
    Varianza = var(C(:, :, j)); Var_P(:, :, j) = Varianza;
end
% Data de reconocimiento agrupada por paciente
% -----
l = size(At,2)/Np; V0 = 1:l:size(At,2); V1 = 0:l:size(At,2); V1(1)=[];
U = [];
for i = 1:Np
    U(:, :, i) = (At(:, [V0(i):V1(i)]))';
end
% Conjunto de imágenes de cada paciente
% -----
I_P = []; Imag_P = []; Im_Pacient = []; Im_Pac = [];
for j=1:Np
    Imag_P = U(:, :, j); I_P(:, :, j) = Imag_P;
    Im_Pacient = mean(I_P(:, :, j)); Im_Pac(:, :, j) = Im_Pacient;
end

```

```
% FIN
% =====
```

A.7. Data de entrenamiento y test.

```
% DATA DE ENTRENAMIENTO Y TEST
% =====
% DESCRIPCIÓN: Asigna el porcentaje de patrones usados para el entrenamiento y test.
% ENTRADAS:
%     Pcf: Patrones
%     eo,ef: Posición de las etiquetas iniciales y finales.
% Paciente: Pacientes utilizados.
%     por_ent: Porcentaje de pacientes a usar para entrenamiento y test.
% SALIDAS:
%     P_ent: Patrones de entrenamiento.
%     P_test: Patrones de test.
%     Ind: Índices de los patrones de entrenamiento y test.
% EJEMPLO:
% =====
% [P_ent,P_test,Ind]=Data_De_Entrenamiento_y_Test(Pcf,eo,ef,Paciente,por_ent,band);
% =====
% INICIO
% =====
function
[P_ent,P_test,Ind] = Data_De_Entrenamiento_y_Test(Pcf,eo,ef,Paciente,por_ent,band);
N_paciente = length(Paciente);
[m,n] = size(Pcf);
nc = m/N_paciente;
np = por_ent*nc/100;
N = bin2dec(num2str(Pcf(:,eo:ef)));
k = 0;
P_ent = [];
P_test = [];
while k < length(Paciente)
    k = k + 1;
    a = find(N==k);
    P_ent = [P_ent;Pcf(a(1:np),:)]; % Patrones de entrenamiento.
    P_test = [P_test;Pcf(a(np+1:end),:)]; % Patrones de test.
end

% Índices de los patrones de entrenamiento y test.
Ne = bin2dec(num2str(P_ent(:,eo:ef)));
Nt = bin2dec(num2str(P_test(:,eo:ef)));
Ind = [];
k = 0;
while k < length(Paciente)
    k = k + 1;
    a = find(Ne==k);
    b = find(Nt==k);
    Ind = [Ind;length(a) length(b) k length(a)+length(b)];
end
% FIN
% =====
```

A.8. Etiquetas.

```
% ETIQUETAS
% =====
% DESCRIPCIÓN: Función que arroja las etiquetas para cada paciente.
% SALIDA: Etiqueta: Arreglo en números binarios del 1 al 31.
% EJEMPLO: Etiqueta = Etiquetas;
% =====
% INICIO
% =====
function Etiqueta = Etiquetas

Etiqueta = [0 0 0 0 1; 0 0 0 1 0; 0 0 0 1 1; 0 0 1 0 0; 0 0 1 0 1;...
            0 0 1 1 0; 0 0 1 1 1; 0 1 0 0 0; 0 1 0 0 1; 0 1 0 1 0;...
            0 1 0 1 1; 0 1 1 0 0; 0 1 1 0 1; 0 1 1 1 0; 0 1 1 1 1;...
            1 0 0 0 0; 1 0 0 0 1; 1 0 0 1 0; 1 0 0 1 1; 1 0 1 0 0;...
            1 0 1 0 1; 1 0 1 1 0; 1 0 1 1 1; 1 1 0 0 0; 1 1 0 0 1;...
            1 1 0 1 0; 1 1 0 1 1; 1 1 1 0 0; 1 1 1 0 1; 1 1 1 1 0;...
            1 1 1 1 1];

% FIN
% =====
```

A.9. Red Neuronal Backpropagation.

```
% RED NEURAL BACKPROPAGATION
% =====
% DESCRIPCIÓN:
%
% ENTRADAS:
% ind_opt: Se toma de la optimización.
% Nt: Número de tareas.
% Fs: Frecuencia de muestreo.
% Dt: Tiempo en segundos el que se divide la ventana total de datos.
% pe: Porcentaje de entrenamiento.
% Na: Número de autovalores a calcular.
% sigma: Umbral de depuración de los datos .
% N1: Neuronas en la primera capa.
% N2: Neuronas en la segunda capa.
% N3: Neuronas en la tercera capa.
% N4: Neuronas en la cuarta capa.
% lr: Rata de aprendizaje.
% err_goal: Criterio de error para detener el entrenamiento.
% epoch_uo: Epoch máximo para el entrenamiento.
% por_ent: Porcentaje de datos usados para el entrenamiento.
% band: Para visualizar la evolución del entrenamiento (1=si, 0=no).
% SALIDA:

% EJEMPLO:
% =====
% Red_Neuronal_Backpropagation(15,[1:8],60,10,70,150,300,10,9,8,7,0.0225,1E-4,1000,1);
% =====

% INICIO
```

```

% =====
function
Red_Neuronal_Backpropagation(ind_opt,Nt,Fs,Dt,pe,Na,sigma,N1,N2,N3,N4,...
                               ...lr,err_goal,epoch_uo,band);

warning off
clc

% I FASE: APLICACIÓN DEL ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)
% =====
% En esta fase se elige un conjunto reducido de autovectores que representan los
% nuevos pacientes denominados eigen-pacientes.
% -----
[Base_Paciente,Paciente,Proj,V,D,Na,I_test] =
    = Analisis_Componentes_Principales(ind_opt,Nt,Fs,Dt,pe,Na,sigma,0);

% II FASE: CREACIÓN Y ENTRENAMIENTO DE LA RED NEURONAL
% =====
% ENTRADAS
% -----
    ao = Base_Paciente(1).ao;      % Posición del primer atributo (autovector).
    af = Base_Paciente(1).af;      % Posición del último atributo (autovector).
    eo = Base_Paciente(1).eo;      % Posición inicial de las etiquetas.
    ef = Base_Paciente(1).ef;      % Posición final de las etiquetas.
    Pcf = Base_Paciente(1).Pcf;    % Patrones usados para entrenar y test
    Pcf_ent = Base_Paciente(1).Pcf_ent; % Patrones usados para entrenar.
    Pcf_test = Base_Paciente(1).Pcf_test; % Patrones usados para test.
    nc = length(Base_Paciente);    % Número de personas usadas en el entrenamiento.
% -----
Parametros_Red = Ejemplo_Red_Neuronal_Perceptronica_BP(N1,N2,N3,N4,...
    ...lr,err_goal,epoch_uo,ao,af,eo,ef,Pcf,Pcf_ent,Pcf_test,band);

% III FASE: TEST Y PRODUCCIÓN DE LA RED (GENERACIÓN DE LA MATRIZ DE CONFUSIÓN)
% =====
    y = Fase_De_Test_Y_Produccion(Parametros_Red,Pcf,ao,af,eo,ef);

% CONSTRUYE LA MATRIZ DE CONFUSIÓN
% -----
[Conf_Ind,Ef,I_test] = Matriz_De_Confucion(y,Pcf,eo,ef,Paciente,band);
save Parametros_Red_1 Parametros_Red
pause(2)

% IV FASE: CLASIFICACIÓN
% =====
% PACIENTES DESCONOCIDOS
% -----
[PCF,Paciente_test,Base_Paciente_Test] = Proyeccion_Paciente_Desconocido(ind_opt,...
    ...Nt,Fs,Dt,pe,Na,sigma,V,Proj,Paciente,band);

load Parametros_Red_1
ao_t = Base_Paciente_Test(1).ao_t; % Posición del primer atributo (autovector).
af_t = Base_Paciente_Test(1).af_t; % Posición del último atributo (autovector).
eo_t = Base_Paciente_Test(1).eo_t; % Posición inicial de las etiquetas.
ef_t = Base_Paciente_Test(1).ef_t; % Posición final de las etiquetas.

% FASE DE TEST Y PRODUCCIÓN TOMANDO EN CUENTA LA RED YA ENTRENADA
% -----

```

```

[Y,Np] = Fase_De_Test_Y_Produccion_2(Parametros_Red,PCF,ao_t,af_t,eo_t,...
                                     ...ef_t,Paciente_test);

% CLASIFICACIÓN
% -----
p = length(Paciente);
q = length(Paciente_test)+p+1;
[Conf_Ind,Ef,I_test] = Matriz_De_Confucion(y,Pcf,eo,ef,Paciente,band);
hold on
[Efica,I_TEST,I_test_1,EFIC]=Matriz_De_Confucion_TEST(Y,PCF,eo_t,ef_t,Np,Paciente,band);
line([p p], [0 q], 'linewidth',[3], 'color','k');
line([0 q], [p p], 'linewidth',[3], 'color','k');
hold off
% FIN
% =====

```

A.10. Red Neuronal Perceptronica BP.

```

% EJEMPLO DE RED NEURONAL PERCEPTRONICA BP
% =====
% DESCRIPCIÓN: Programa que permite entrenar una red neuronal perceptronica multicapa
% con el algoritmo de retropropagación.
% ENTRADAS:
% (N1,N2,N3,N4): Número de neuronas en las capas (s1,s2,s3,s4).
% lr: Rata de aprendizaje.
% err_goal: Criterio de error para detener el entrenamiento
% epoch_uo: Epoch máximo para el entrenamiento.
% (ao,af): Posición inicial y final de los atributos.
% (eo,ef): Posición inicial y final de las etiquetas.
% Pcf_ent: Patrones usados para entrenar.
% Pcf_test: Patrones usados para tes.
% band: Para visualizar (1=si, 0=no).
% SALIDAS:
% Parametros_Red: {'W1',W1,'W2',W2,'W3',W3,'W4',W4,'W5',W5, % Pesos por capa
%                  'B1',B1,'B2',B2,'B3',B3,'B4',B4,'B5',B5}; % Bias por capa
% EJEMPLO
% =====
% Parametros_Red = Ejemplo_Red_Neuronal_Perceptronica_BP(10,15,8,4,1e-3,100,1,...
%                  ...10,11,14,Pcf_ent,Pcf_test,1);
% =====
% INICIO
% =====
function
Parametros_Red = Ejemplo_Red_Neuronal_Perceptronica_BP(N1,N2,N3,N4,lr,err_goal,...
    ...epoch_uo,ao,af,eo,ef,Pcf,Pcf_ent,Pcf_test,band);

nntwarn off
% RATA DE APRENDIZAJE
% =====
% lr = 0.01250; % Se asigna manualmente
% =====
Po_ent = Pcf_ent'; % Orden original de los patrones.
SE = 1; % Inicializa el error cuadrático.
% INICIALIZA LAS MATRICES
% =====
Error_ent = []; Error_test = []; Dif = [];
sse = []; % Error cuadrático.

```

```

errors_ent = []; % Matriz de errores de entrenamiento.
errors_test = []; % Matriz de errores de test.
epoch = 0; % Contador de fase de presentación de entradas.
bias = []; % Bias.
A = []; % Salida actual.
count = 0; % Contador de presentación de patrones.
[nc,np] = size(Pcf_ent); % Número total de patrones.
% INICIO DE LA RED
% =====
% CONDICION PARA DETENER LA RED
% -----
while (SSE >= err_goal) & (epoch < epoch_uo)
    errors_ent = [];
    errors_test = [];
    epoch = epoch + 1; % Contador de epoch's.
% ENTRENAMIENTO
% =====
a = randperm(size(Po_ent,2));
Pcf = Po_ent(:,a); % Reasignación aleatoria de los patrones.

for k = 1:nc % Ciclo de presentación de los patrones.
    count = count + 1;
    P = Pcf(ao:af,k); % Entrada actual.
    T = Pcf(eo:ef,k); % Salida objetivo actual.
% TOPOLOGÍA Y PARÁMETROS INICIALES
% =====
if count == 1
    [R,Q] = size(P); % Definición de la matriz de entrada:
    % R número de atributos Y Q vector de entrada.
    S1 = N1; % Número de neuronas en la primera capa.
    S2 = N2; % Número de neuronas en la segunda capa.
    S3 = N3; % Número de neuronas en la tercera capa.
    S4 = N4; % Número de neuronas en la cuarta capa.
    [S5,Q] = size(T); % Definición de la matriz de salida: (5ta capa).
% ASIGNACIÓN DE PESOS Y BIAS (AL AZAR EN EL PRIMER EPOCH)
% =====
    [W1,B1] = randn(S1,R); % Pesos y Bias 1 capa.
    [W2,B2] = randn(S2,S1); % Pesos y Bias 2 capa.
    [W3,B3] = randn(S3,S2); % Pesos y Bias 3 capa.
    [W4,B4] = randn(S4,S3); % Pesos y Bias 4 capa.
    [W5,B5] = randn(S5,S4); % Pesos y Bias 5 capa.
end
% FASE DE PRESENTACIÓN DE LOS PATRONES (Epoch) (ENTRENAMIENTO)
% =====
    A1 = tansig(W1*P,B1); % Salida de la primera capa.
    A2 = tansig(W2*A1,B2); % Salida de la segunda capa.
    A3 = tansig(W3*A2,B3); % Salida de la tercera capa.
    A4 = tansig(W4*A3,B4); % Salida de la cuarta capa.
    A5 = tansig(W5*A4,B5); % Salida de la quinta capa.
    E = T-A5; % Error entre objetivo y salida actual.
    SSE = sumsq(E); % Suma cuadrática del error.
% FASE DE APRENDIZAJE:
% CÁLCULO DEL GRADIENTE DESCENDIENTE EN RETROPROPAGACIÓN PARA CADA CAPA.
% =====
    D5 = deltatan(A5,E); % Determinación del Delta para la capa 5.
    D4 = deltatan(A4,D5,W5); % Determinación del Delta para la capa 4.

```

```

D3 = deltatan(A3,D4,W4); % Determinación del Delta para la capa 3.
D2 = deltatan(A2,D3,W3); % Determinación del Delta para la capa 2.
D1 = deltatan(A1,D2,W2); % Determinación del Delta para la capa 1.
% MODIFICACIÓN DE LOS PESOS Y BIAS EN DIRECCIÓN DEL GRADIENTE MÍNIMO
% =====
[dW1,dB1] = learnbp(P,D1,lr); % Modificación para la capa 1.
[dW2,dB2] = learnbp(A1,D2,lr); % Modificación para la capa 2.
[dW3,dB3] = learnbp(A2,D3,lr); % Modificación para la capa 3.
[dW4,dB4] = learnbp(A3,D4,lr); % Modificación para la capa 4.
[dW5,dB5] = learnbp(A4,D5,lr); % Modificación para la capa 5.
% REASIGNACIÓN DE LOS PESOS Y BIAS EN LA DIRECCIÓN DEL GRADIENTE MÍNIMO
% =====
W1 = W1 + dW1; B1 = B1 + dB1; % Modificación para la capa 1.
W2 = W2 + dW2; B2 = B2 + dB2; % Modificación para la capa 2.
W3 = W3 + dW3; B3 = B3 + dB3; % Modificación para la capa 3.
W4 = W4 + dW4; B4 = B4 + dB4; % Modificación para la capa 4.
W5 = W5 + dW5; B5 = B5 + dB5; % Modificación para la capa 5.
% EVOLUCION DEL ERROR
% =====
errors_ent = [errors_ent SSE]; % Vector de errores (entrenamiento).
end
% ERROR EN EL ENTRENAMIENTO
% =====
Error_ent = [Error_ent;sumsqr(errors_ent)];
% ERROR EN EL TEST
% =====
T_test = Pcf_test(:,eo:ef);
Out_test = tansig(W5*tansig(W4*tansig(W3*(tansig(W2*tansig...
... (W1*Pcf_test(:,ao:af)',B1),B2)),B3),B4),B5)'; % salida de la red.
E_test = T_test - Out_test;
SSE_test = sumsqr(E_test); % Suma cuadrática del error.
errors_test = [errors_test SSE_test]; % Vector de errores (test).
Error_test = [Error_test;sumsqr(errors_test)];
Ee = bin2dec(num2str(round(abs(A5))))'; Ed = bin2dec(num2str(T'));
dif = Ee - Ed; Dif = [Dif;dif];
% VISUALIZACIÓN
% =====
if band == 1
    n_ent = num2str(size(Pcf_ent,1)); n_test = num2str(size(Pcf_test,1));
    plot(1:epoch,log10(Error_ent),'-b',1:epoch,log10(Error_test),'-r',1:epoch,...
        ...log10(abs(Dif)+1),'-g','linewidth',[2])
    xlabel('Epoches'),ylabel('log10(SSE)'), pause(.01) end end
% SALIDAS
% =====
Parametros_Red = struct('W1',W1,'W2',W2,'W3',W3,'W4',W4,'W5',W5,...
    'B1',B1,'B2',B2,'B3',B3,'B4',B4,'B5',B5);
save Parametros_De_Entrenamiento Parametros_Red
% FIN
% =====

```

A.11. Fase de test y producción.

```

% FASE DE TEST Y PRODUCCIÓN
% =====
% DESCRIPCIÓN: Representa la función general de la red donde se presentan

```

```

% todos los pacientes en forma silmutánea.
% ENTRADAS:
% Parametros_Red: {'W1',W1,'W2',W2,'W3',W3,'W4',W4,'W5',W5, Pesos por capa
%                  'B1',B1,'B2',B2,'B3',B3,'B4',B4,'B5',B5} Bias por capa
% Pcf: Patrones usados para entrenar y test.
% ao: Posición del primer atributo (autovector)
% af: Posición del último atributo (autovector).
% eo: Posición inicial de las etiquetas.
% ef: Posición final de las etiquetas.
% SALIDAS:
% y: Función general de la red.
% INICIO
% =====
function y = Fase_De_Test_Y_Produccion(Parametros_Red,Pcf,ao,af,eo,ef);
% PESOS
% =====
W1 = Parametros_Red.W1;
W2 = Parametros_Red.W2;
W3 = Parametros_Red.W3;
W4 = Parametros_Red.W4;
W5 = Parametros_Red.W5;
% BIAS
% =====
B1 = Parametros_Red.B1;
B2 = Parametros_Red.B2;
B3 = Parametros_Red.B3;
B4 = Parametros_Red.B4;
B5 = Parametros_Red.B5;
% PRESENTACIÓN DE TODOS LOS PACIENTES EN FORMA SIMULTANEA
% =====
Ao = Pcf(:,ao:af)'; % Patrones de entrada (primera capa).
A1 = tansig(W1*Ao,B1); % Salida de la primera capa.
A2 = tansig(W2*A1,B2); % Salida de la segunda capa.
A3 = tansig(W3*A2,B3); % Salida de la tercera capa.
A4 = tansig(W4*A3,B4); % Salida de la cuarta capa.
A5 = tansig(W5*A4,B5); % Salida de la quinta capa.
% FUNCIÓN GENERAL DE LA RED
% =====
y = A5;
% FIN
% =====

```

A.12. Matriz de Confusión.

```

% MATRIZ DE CONFUSIÓN
% =====
% DESCRIPCIÓN: Determina la coincidencia entre las etiquetas dadas versus las
% encontradas.
% ENTRADAS:
% y: Función general de la red (salida de la red).
% Pcf: Patrones usados para entrenar y test.
% eo: Posición inicial de las etiquetas.
% ef: Posición final de las etiquetas.
% Paciente: Cadena de los pacientes utilizados.
% band: 1, para visualizar la salida.

```

```

% SALIDAS:
% Conf_Ind: Índices de la matriz de confusión.
% Ef: Eficiencia porcentual del algoritmo.
% I_test: Salida actual (después de entrenada la red) y la salida deseada.
% EJEMPLO:
% =====
% [Conf_Ind,Ef,I_test] = Matriz_De_Confucion(y,Pcf,eo,ef,Paciente,1);
% =====
% INICIO
% =====
function [Conf_Ind,Ef,I_test] = Matriz_De_Confucion(y,Pcf,eo,ef,Paciente,band);

    Np = length(Paciente);
    P = []; Paciente = [];
    for i = 1:Np
        P = {num2str(i)};
        num2cell(P);
        Paciente = [Paciente P];
    end
    [m,n] = size(Paciente);
    nv = (Pcf(:,eo:ef));
    Out = bin2dec(num2str((abs(round(y))))); % Salida actual.
    Input = bin2dec(num2str((abs(round(nv))))); % Salida deseada.
    I_test = [Input Out]; % [Salida actual Salida deseada].

% ÍNDICES DE LA MATRIZ DE CONFUSIÓN.
% -----
    Ind = []; Conf_Ind = [];
    for i=1:n
        ind_a = [];
        for j=1:n
            ind = find(I_test(:,1)==i & I_test(:,2)==j);
            Ind = [Ind; length(ind) i j];
            ind_a(j) = length(ind);
        end
        Conf_Ind = [Conf_Ind;ind_a];
    end

% EFICIENCIA DE LA RED
% -----
    N = size(Pcf,1);
    ind1 = find(I_test(:,1)==I_test(:,2)); aciertos = length(ind1);
    ind2 = find(I_test(:,1)~=I_test(:,2)); fallos = length(ind2);
    Ef = (aciertos-fallos)/(aciertos+fallos)*100;
% -----

% COEFICIENTES DEL POLINOMIO (E-dada Vs E-encontrada)
% -----
    [coef,st] = polyfit(I_test(:,1),I_test(:,2),1);
    if band == 1
        plot(I_test(:,1),I_test(:,2),'.g',I_test(:,1),I_test(:,2),'or','markersize',[10])
        xlabel('Etiquetas dadas','fontsize',[12],'fontweight','bold','color','k')
        ylabel('Etiquetas encontradas','fontsize',[12],'fontweight','bold','color','k')
    end

% REPRESENTACIÓN DE LAS ETIQUETAS
% -----
    K = length(Paciente);

```

```

L = K+1;
f = Paciente;
g = get(figure(1));
go = g.Children;
x = linspace(1,K,length(f));
set(go(1),'XTick',x,'XTickLabel',f)
set(go(1),'YTick',x,'YTickLabel',f)
axis([0 L 0 L])
Ax = []; Ay = [];
for i = 1:length(Ind)
    a = Ind(i,:,:)
    ax = a(2);
    ay = a(3);
    text(ax+0,ay,num2str(a(1)),'fontsize',[11],'fontweight','bold','color','b');
end
line([1 K],[1 K],'linewidth',[2],'color','k');
xp = linspace(0,L,L);
yp = polyval(coef,xp);
hold on,plot(xp,yp,'--m','linewidth',[4]),hold off
title(['[E dadas Vs E encontradas] [Ef = ' num2str(round(Ef)) '% ]' ],...
... 'fontsize',[12],'fontweight','bold','color','k')
end end
% FIN
% =====

```

A.13. Proyección del paciente desconocido.

```

% PROYECCIÓN DEL PACIENTE DESCONOCIDO
% =====
% DESCRIPCIÓN: Este programa se utiliza para escribir la data del paciente
% nuevo en función del tamaño de la data de entrenamiento del algoritmo.
% Además los nuevos pacientes se proyectan en el espacio de los autovectores
% de los pacientes de entrenamiento.
% ENTRADAS:
% ind_opt: Se toma de la optimización.
% Nt: Número de tareas.
% Fs: Frecuencia de muestreo.
% Dt: Tiempo en segundos el que se divide la ventana total de datos.
% pe: Porcentaje de entrenamiento.
% Na: Número de autovalores a calcular.
% sigma: Umbral de depuración de los datos .
% V: Autovectores calculados durante el PCA.
% Proj: Contiene la proyección de los pacientes durante la etapa de entrenamiento.
% Paciente: Contiene el número de pacientes durante la etapa de entrenamiento
% band: 1, para visualizar la salida.
% SALIDAS:
% PCF: La data de reconocimiento con sus respectivos patrones.
% Base_Paciente_Test: Una estructura que contiene lo siguiente:
% - PCF: La data de reconocimiento con sus respectivos patrones.
% - ao_t,af_t: Posición de los atributos iniciales y finales.
% - eo_t,ef_t: Posición de las etiquetas iniciales y finales.
% Paciente_test: Cadena de los pacientes utilizados para el reconocimiento.
% EJEMPLO:
% =====
% [PCF,Paciente_test,Base_Paciente_Test]=Proyeccion_Paciente_Desconocido(15,...

```

```

... [1:16],60,10,50,100,300,V,Proj,Paciente,1);
% =====
% INICIO
% =====
function
[PCF,Paciente_test,Base_Paciente_Test]=Proyeccion_Paciente_Desconocido(ind_opt,...
    ...Nt,Fs,Dt,pe,Na,sigma,V,Proj,Paciente,band);

Paciente_test = {'Maria' 'Adolfo' 'Neptali' 'Sandra' 'Stephanie' 'Jesus'};
Np = length(Paciente_test); % Número de pacientes

% SE REDIMENSIONAN LOS DATOS A UNA MISMA LONGITUD
% -----
[Dr,Nv,dt,Num_par]=Redimensiona_Datos_5(Paciente_test,ind_opt,Nt,Fs,Dt,sigma,1);
% -----
% SE CREA LA MATRIZ CON TODOS LOS EJEMPLOS
% -----
[I_P,Im_Pacient] = Formato_de_Datos_Renombrado_3(Np,Nt,Nv,Num_par,Dr,pe,1);

% SE ESCRIBE LA MATRIZ ANTES CREADA EN FUNCIÓN DE LA LONGITUD DE LOS AUTOVECTORES
% UTILIZADOS PARA EL ENTRENAMIENTO
% -----
    f = size(V,1);
    g = size(I_P,2);
    c1 = min(f,g);
    I_P = I_P(:,1:c1,:); % Ejemplos de cada paciente para el reconocimiento.
    V = V(1:c1,:);
    Im_Pacient = Im_Pacient(:,1:c1);
    [m,n,k] = size(Im_Pacient(:, :, 1));

test = 1; if test == 1
Etiqueta = Etiquetas;

% TEST CON PACIENTES NUEVOS (RECONOCIMIENTO)
% -----
    if test == 1
        I_test = [];
% MATRIZ DE DATOS PARA EL RECONOCIMIENTO
% -----
        w = 4; % Número total de ejemplos por paciente.
% SELECCIÓN DE UNA IMAGEN DE CADA PACIENTE
% -----
        Im_Pac = mean(I_P);
        I_PP = repmat(Im_Pac,w,1);
        PROJ = [];
        for k = 1:w
            Imp = [];
            for i = 1:length(Paciente_test)
                Imedia = I_PP(k,:,i);
                Imp(:, :, i) = Imedia;
            end

% PROYECCIÓN EN EL ESPACIO REAL (RECONOCIMIENTO)
% -----
        X = []; Y = [];
        for i=1:Np

```

```

Ix = reshape(Imp(:, :, i), 1, m*n);
proj_x = Ix*V; % Proyección del paciente desconocido
x = Ix*V(:, 1:2);
proj_xr = repmat(proj_x, size(Proj, 3), 1);
ac = squeeze(Proj);
bc = proj_xr';
% Distancia
% Coeficiente de correlación
% -----
e = []; for ii=1:size(ac, 2); co = corrcoef(ac(:, ii), bc(:, 1)); e(ii)=co(1, 2); end
[im, jm] = max(abs(e));
X = [X; x(1) i jm];
Y = [Y; x(2) i jm];
if band == 1
if length(Paciente_test) <= 4
subplot(Np, 1, i)
bar(e), title([ Paciente_test{i} '<=>' Paciente{jm}], 'color', [1 0 0]), grid
else
subplot(5, round((Np/5)+0.3), i)
bar(e), title([ Paciente_test{i} '<=>' Paciente{jm}], 'color', [1 0 0]), grid
end end
I_test = [I_test; i jm]; % Se guardan los índices
Etiqueta(i, :);
PROJ = [PROJ; proj_x Etiqueta(i, :)];
end end end

% SE NORMALIZAN LOS DATOS Y SE GUARDA LAS PROYECCIONES PARA SER
% USADAS EN EL ENTRENAMIENTOS CON REDES
% -----
% Posición de los atributos
ao_t = 1; af_t = Na;
% Posición de las etiquetas
eo_t = Na + 1; ef_t = eo_t + 4;
% -----
Base_Paciente_Test = {};
if test == 1
% Normalización
Po = PROJ(:, 1:Na);
Max_Po = max(abs(Po(:)));
Pn = Po/Max_Po;

% Se agregan las etiquetas a cada proyección
% -----
PCF = [Pn PROJ(:, eo_t:ef_t)]; % Patrones

% SALIDAS
% =====
Base_Paciente_Test = [];
Base_Paciente_Test = struct('PCF', PCF, 'ao_t', ao_t, 'af_t', af_t, 'eo_t', ...
...eo_t, 'ef_t', ef_t);
save Base_De_Datos_Paciente_Test Base_Paciente_Test
end end

% FIN
% =====

```

A.14. Dendrograma Parkinson.

```
% DENDROGRAMA PARKINSON
% =====
% DESCRIPCIÓN: Este programa muestra los dendrogramas para cada caso. Se construye a partir de
% la función de la red.
% ENTRADAS:
%     nc: Número de clases.
%     np: Número de pacientes.
%     Y : Función de la red.
%     band: 1, para visualizar la salida.
% SALIDAS:
%     Gráfico del dendrograma con todos los sujetos y el gráfico de clases.
%     Z: Matriz que codifica un árbol que contiene grupos jerárquicos de la fila de la matriz de
%     datos de entrada.
%     T: Vector que contiene el número de nodos para cada objeto en el conjunto de datos original.
% EJEMPLO:
% =====
% [Z,T] = Dendrograma_Parkinson(nc, np, Y, band);
% [Z,T] = Dendrograma_Parkinson(5, 45, Y, 1);
% =====

% INICIO
% =====
function [Z,T] = Dendrograma_Parkinson(nc, np, Y, band);
    p = Y(:,1:np); % Función de la red
    p = [p';0*p(:,1)'];

% Gráfico clases
% =====
set(figure(1),'Position',[1032 54 863 940],'color','w')
    Yd = pdist(p,'Mahalanobis');
    Z = linkage(Yd,'single');
[H, T] = dendrogram(Z,5);
ylim([0 10])
ylabel('Distancia Mahalanobis'),xlabel('Clase')
title('Distancia Mahalanobis -150 autovalores- *Todas-Las-Tareas-Todos-Los-Parametros')

% Dendrograma
% =====
set(figure(2),'Position',[1032 54 863 940],'color','w')
figure(2)cd /home/steph/Escritorio/PCF/1
H = dendrogram(Z,0,'Orientation','left','ColorThreshold','default');
xlim([0 10])
xlabel('Distancia Mahalanobis'),ylabel('Paciente')
title('Distancia Mahalanobis -150 autovalores- *Todas-Las-Tareas-Todos-Los-Parametros')
% FIN
% =====
```

A.15. Clasificación Parkinson.

```
% CLASIFICACIÓN PARKINSON
% =====
% DESCRIPCIÓN: Programa que visualiza la distancia de Mahalanobis en función
```

```
% de la edad, a partir de la matriz de distancias.
% ENTRADAS:
%      nc: Número de clases.
%      Z:
%      T:
%      band: 1, para visualizar.
% SALIDAS:
%      Gráfico de Distancia de Mahalanobis vs Edad
% EJEMPLO
% =====
%   Clasificacion_Parkinson(nc,T,Z,band);
% =====

INICIO
=====
function Clasificacion_Parkinson(nc,T,Z,band);
A = [];
for i=1:nc %número de clases
    ind = find(T==i);
    ind_x = find(ind>np);
    if length(ind_x)==1
        ind(ind_x) = [];
    end
    A = [A;[ind Z(ind,3)]]; %Matriz de distancias
end
[D,io] = sort(A(:,2)); %Orden de la matriz

% =====
% Vector de edades
% =====
Edad = {'24' '24' '25' '25' '25' '26' '26' '29' '29' '32' '32' '33' ...
        '34' '34' '35' '36' '37' '38' '40' '40' '40' '41' '42' '43' '44' '45' ...
        '47' '50' '51' '51' '54' '56' '57' '58' '59' '59' '60' '60' '61' '62' ...
        '62' '63' '64' '68' '74' '56' '58' '59' '61' '62' '63'};

set(figure(1),'Position',[1032 54 863 940],'color','w')
E = str2num(cell2mat(Edad)); %Vector edad
Ea = E(io); %Vector edad con su distancia correspondiente
plot(Ea,D,'*',Ea,D,'o'),grid
hold on
plot(Ea(46:51,:),D(46:51,:),'*r',Ea(46:51,:),D(46:51,:),'or')
hold off
%FIN
% =====
```

A.16. Dinámica de los Pacientes.

```
% DINÁMICA PACIENTES
% =====
% DESCRIPCIÓN: Programa donde se calculan varios parámetros para el estudio de la dinámica
% de los sujetos estudiados, entre estos se encuentran: Medias, frecuencias y módulos. Y se
% muestra la comparación entre las tareas a través de gráficos de barras.
% ENTRADAS:
%      Nt: Número de tareas
```

```

% Tareas: Vector que contiene las tareas.
% SALIDAS:
% Medias
% Frecuencia
% MODULOS
% EJEMPLO
% =====
% [Medias, Frecuencia, MODULOS] = Dinámica_Parkinson(nt,Tareas)
% =====

% INICIO
% =====
function [Medias, Frecuencia, MODULOS] = Dinámica_Parkinson(nt,Tareas)

% Pacientes
% =====
    Paciente = {'Maria' 'Ignacio' 'Antonio' 'Moises' 'Diego' 'Jesus' 'Stephanie' 'Neptali' };
    Medias = []; Frecuencia = []; []; MODULOS = {}; Fmax = []; N = [];
for j=1:length(Paciente)
    Modulos = {};
for i=1:length(Nt)
    eval(['load Data_Paciente_' cell2mat(Paciente(j)) '_' num2str(i)]);

% Componentes
% =====
    ax = DATA(:,1); ay = DATA(:,2); az = DATA(:,3); bx = DATA(:,4); by = DATA(:,5);
    bz = DATA(:,6); wx = DATA(:,7); wy = DATA(:,8); wz = DATA(:,9); pulso = DATA(:,10);

% Módulos y medias
% =====
    a = sqrt(ax.^2 + ay.^2 + az.^2);
    b = sqrt(bx.^2 + by.^2 + bz.^2);
    w = sqrt(wx.^2 + wy.^2 + wz.^2);
    am = mean(a); bm = mean(b); wm = mean(w); pm = mean(pulso);

% Frecuencias
% =====
    [m,n,mo,no,fmax] = Analisis_Dinamico_Por_Tarea_Por_paciente(DATA,ho,0);
    n1= fmax(10);
    n = mean(n');
    m = mean(m');
    N = [N;n n1 i j]; % Frecuencias

% Resultados por tareas y por pacientes
% =====
    Medias = [Medias; am bm wm pm i j];
Frecuencia = [Frecuencia; n n1 i j];
    Modulos(i).Modulos = [a b w pulso];
end
    MODULOS(j).MODULOS = Modulos;
end

% Cada tarea por todos los pacientes.
% =====
set(figure(1),'Position',[403 336 621 639],'Color','W')
for i=1:8

```

```

ind = find(N(:,5)==i);
plot(N(ind,[1:4]),'-','LineWidth',[2])
title([Tarea(i),'aceleración(azul), campo (rojo), velocidad a. (amarillo),p (m)'])
xlabel('Pacientes'); ylabel('Frecuencia central (Fc)')
end

% Todas las tareas por paciente.
% =====
set(figure(2),'Position',[403 336 621 639],'Color','W')
for i=1:length(Paciente)
    ind = find(N(:,6)==i);
    plot(N(ind,[1:4]),'-','LineWidth',[2]) % 1:4 incluye el pulso.
    title([Paciente(i),'aceleración(azul), campo (rojo), velocidad a. (amarillo),p (morado)'])
    xlabel('Tareas'); ylabel('Frecuencia central (Fc)')
end

% Frecuencia central de cada paciente por tarea y por parámetro.
% =====
X = []; Y = []; Z = []; W = [];
for i=1:length(Paciente)
    ind = find(N(:,6)==i);
    a = N(ind,1); % frecuencia en aceleración
    b = N(ind,2); % frecuencia en campo
    c = N(ind,3); % frecuencia en velocidad angular
    d = N(ind,4); % frecuencia en pulso
    X = [X a]; Y = [Y b]; Z = [Z c]; W = [W d];
end

set(figure(3),'Position',[403 336 1000 1000],'Color','W')
subplot(4,1,1), bar(X), xlabel('Tareas'), ylabel('Fc: aceleracion')
subplot(4,1,2), bar(Y), xlabel('Tareas'), ylabel('Fc: campo')
subplot(4,1,3), bar(Z), xlabel('Tareas'), ylabel('Fc: velocidad angular')
subplot(4,1,4), bar(W), xlabel('Tareas'), ylabel('Fc: pulso')

% Medias de cada paciente por tarea y por parámetro.
% =====
N = Medias;
X = []; Y = []; Z = []; W = [];
for i=1:length(Paciente)
    ind = find(N(:,6)==i);
    a = N(ind,1); % frecuencia en aceleración
    b = N(ind,2); % frecuencia en campo
    c = N(ind,3); % frecuencia en velocidad angular
    d = N(ind,4); % frecuencia en pulso
    X = [X a]; Y = [Y b]; Z = [Z c]; W = [W d];
end

set(figure(5),'Position',[403 336 1000 1000],'Color','W')
subplot(4,1,1), bar(X), xlabel('Tareas'), ylabel('Aceleracion media')
subplot(4,1,2), bar(Y), xlabel('Tareas'), ylabel('Campo medio')
subplot(4,1,3), bar(Z), xlabel('Tareas'), ylabel('velocidad angular media')
subplot(4,1,4), bar(W), xlabel('Tareas'), ylabel('Pulso medio')
% FIN
% =====

```

Apéndice B

Programas Arduino.

B.1. Todos los sensores.

```
//Sensor GY-80
//Bibliotecas
#include <Wire.h>
#include <I2Cdev.h>
#include <GY80.h>

GY80 sensor = GY80(); // create GY80 instance

//variables
int PulseSensor = A0;
int Buzzer = 10;
int Signal = 0;
int Threshold = 525; // 550 Valor típico usado en otros proyectos del PulseSensor

void setup()
{
    pinMode(10,OUTPUT);
    // initialize serial communication at 57600 bits per second:
    Serial.begin(57600);
    sensor.begin(); //initialize sensors
}
void loop()
{
    GY80_scaled val = sensor.read_scaled(); //valores para todos los sensores
    Signal = analogRead(PulseSensor);
    Serial.println(Signal);
    if(Signal > Threshold){
        digitalWrite(Buzzer,HIGH);
        Serial.println();
    }
    if(Signal <= Threshold){
        digitalWrite(Buzzer,LOW);
        Serial.println();
    }
}
//Imprime los valores del acelerómetro
Serial.print(val.a_x,3); Serial.print( "  " );
Serial.print(val.a_y,3); Serial.print( "  " );
Serial.print(val.a_z,3); Serial.print( "  " );
```

```
//Imprime los valores del magnetómetro
  Serial.print(val.m_x,2); Serial.print( "  " );
  Serial.print(val.m_y,2); Serial.print( "  " );
  Serial.print(val.m_z,2); Serial.print( "  " );
//Imprime los valores del giroscopo
  Serial.print(val.g_x,1); Serial.print( "  " );
  Serial.print(val.g_y,1); Serial.print( "  " );
  Serial.print(val.g_z,1); Serial.print( "  " );
  delay(10); // delay in between reads for stability
}
```

B.2. Calibración acelerómetro ADXL345.

```
#include <Wire.h>

#define ADXL345_DEVICE 0x53    // ADXL345 device address
#define ADXL345_TO_READ 6      // num of bytes we are going to read each time.

/* ----- Register names ----- */
#define ADXL345_DEVID 0x00
#define ADXL345_RESERVED1 0x01
#define ADXL345_THRESH_TAP 0x1d
#define ADXL345_OFSX 0x1e
#define ADXL345_OFSY 0x1f
#define ADXL345_OFSZ 0x20
#define ADXL345_DUR 0x21
#define ADXL345_LATENT 0x22
#define ADXL345_WINDOW 0x23
#define ADXL345_THRESH_ACT 0x24
#define ADXL345_THRESH_INACT 0x25
#define ADXL345_TIME_INACT 0x26
#define ADXL345_ACT_INACT_CTL 0x27
#define ADXL345_THRESH_FF 0x28
#define ADXL345_TIME_FF 0x29
#define ADXL345_TAP_AXES 0x2a
#define ADXL345_ACT_TAP_STATUS 0x2b
#define ADXL345_BW_RATE 0x2c
#define ADXL345_POWER_CTL 0x2d
#define ADXL345_INT_ENABLE 0x2e
#define ADXL345_INT_MAP 0x2f
#define ADXL345_INT_SOURCE 0x30
#define ADXL345_DATA_FORMAT 0x31
#define ADXL345_DATAX0 0x32
#define ADXL345_DATAX1 0x33
#define ADXL345_DATAY0 0x34
#define ADXL345_DATAY1 0x35
#define ADXL345_DATAZ0 0x36
#define ADXL345_DATAZ1 0x37
#define ADXL345_FIFO_CTL 0x38
#define ADXL345_FIFO_STATUS 0x39
#define ADXL345_OK 1 // no error
#define ADXL345_ERROR 0 // indicates error is present
#define ADXL345_NO_ERROR 0 // initial state
#define ADXL345_READ_ERROR 1 // problem reading accel
```

```

#define ADXL345_BAD_ARG      2 // bad method argument
#define NRO_PROMEDIO 10

char incoming[20];
bool estatus;
byte error_code;

union XYZBuffer {
    struct {
        short x,y,z;
    } value;
    byte buff[6];
};

void changeEndian(union XYZBuffer *xyz) {
    for (int i=0;i<6;i+=2) {
        byte t=xyz->buff[i];
        xyz->buff[i]=xyz->buff[i+1];
        xyz->buff[i+1]=t;
    }
}

// Generically useful reading into a union type
void readXYZ(int device,union XYZBuffer *xyz) {
    Wire.requestFrom(device, 6);
    long start=millis();
    while (!Wire.available() && (millis()-start)<100);
    if (millis()-start<100) {
        for (int i=0;i<6;i++)
            xyz->buff[i]=Wire.read();
    }
}

void setupAccel(int device) {
    // Check ID to see if we are communicating
    Wire.beginTransmission(device);
    Wire.write(0x00); // One Reading
    Wire.endTransmission();
    Wire.requestFrom(device,1);
    while (!Wire.available());
    byte ch=Wire.read();
    Serial.print("Accel id is 0x");
    Serial.println(ch,HEX);

    // Should output E5

    // https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf
    // Page 16
    Wire.beginTransmission(device);
    Wire.write(ADXL345_POWER_CTL);
    Wire.write(0x08);
    Wire.endTransmission();
    Wire.beginTransmission(device);
    Wire.write(ADXL345_FIFO_CTL);
    Wire.write(0x84);
    Wire.endTransmission();
    writeTo(ADXL345_DATA_FORMAT, 0x0B); // full Res & 16g
    writeTo(ADXL345_BW_RATE, 0x0C); //rate 400 Hz
}

```

```

void readAccel(int device, union XYZBuffer *xyz) {
    Wire.beginTransaction(device);
    Wire.write(ADXL345_DATA0); // One Reading
    Wire.endTransmission();
    readXYZ(device, xyz);
}

void output(union XYZBuffer xyz) {
    float kg;
    kg = 0.00390625;
    // kg = 0.00130719;
    // pad(6, xyz.value.x);
    Serial.print((xyz.value.x)*kg, 4);
    Serial.print(',');
    // pad(6, xyz.value.y);
    Serial.print((xyz.value.y)*kg, 4);
    Serial.print(',');
    // pad(6, xyz.value.z);
    Serial.print((xyz.value.z)*kg, 4);
}

void readFrom(byte address, int num, byte _buff[]) {
    Wire.beginTransaction(ADXL345_DEVICE); // start transmission to device
    Wire.write(address); // sends address to read from
    Wire.endTransmission(); // end transmission

    Wire.beginTransaction(ADXL345_DEVICE); // start transmission to device
    Wire.requestFrom(ADXL345_DEVICE, num); // request 6 bytes from device

    int i = 0;
    while(Wire.available()) // device may send less than requested (abnormal)
    {
        _buff[i] = Wire.read(); // receive a byte
        i++;
    }
    if(i != num){
        estatus = ADXL345_ERROR;
        error_code = ADXL345_READ_ERROR;
    }
    Wire.endTransmission(); // end transmission
}

// Writes val to address register on device
void writeTo(byte address, byte val) {
    Wire.beginTransaction(ADXL345_DEVICE); // start transmission to device
    Wire.write(address); // send register address
    Wire.write(val); // send value to write
    Wire.endTransmission(); // end transmission
}

// print all register value to the serial output, which requires it to be setup
// this can be used to manually to check the current configuration of the device
void printAllRegister() {
    byte _b;
    Serial.print("0x00: ");
    readFrom(0x00, 1, &_b);
    print_byte(_b);
    Serial.println("");
    int i;

```

```

for (i=29;i<=57;i++){
Serial.print("0x");
Serial.print(i, HEX);
Serial.print(": ");
readFrom(i, 1, &_b);
print_byte(_b);
Serial.println("");
}
}

void print_byte(byte val){
int i;
Serial.print("B");
for(i=7; i>=0; i--){
Serial.print(val >> i & 1, BIN);
}
}

// SELF-TEST
void self_test(){
    union XYZBuffer xyz, test;
    byte tmp_rate, tmp_format;
    short nro_ave = NRO_PROMEDIO;
    int aoff_x = 0, aoff_y = 0, aoff_z = 0;
    int aon_x = 0, aon_y = 0, aon_z = 0;
    int i;

    readFrom(ADXL345_BW_RATE, 1, &tmp_rate);
    writeTo(ADXL345_BW_RATE, 0x0A); // rate 100Hz
    readFrom(ADXL345_DATA_FORMAT, 1, &tmp_format);
    writeTo(ADXL345_DATA_FORMAT, 0x0B); // full Res & 16g
    delay(100);

    // Promedio con self_test OFF
    for (i= 0; i < nro_ave; i++){
        readAccel(ADXL345_DEVICE,&xyz);
        aoff_x += xyz.value.x;
        aoff_y += xyz.value.y;
        aoff_z += xyz.value.z;
    }
    aoff_x /= nro_ave;
    aoff_y /= nro_ave;
    aoff_z /= nro_ave;

    // Self-Test en ON
    writeTo(ADXL345_DATA_FORMAT, 0x8B); // full Res & 16g
    delay(100);

    // Promedio con self_test ON
    for (i= 0; i < nro_ave; i++){
        readAccel(ADXL345_DEVICE,&xyz);
        aon_x += xyz.value.x;
        aon_y += xyz.value.y;
        aon_z += xyz.value.z;
    }
    aon_x /= nro_ave;
    aon_y /= nro_ave;
    aon_z /= nro_ave;

```

```

// Resultado del self_test
test.value.x = (short)(aon_x - aoff_x);
test.value.y = (short)(aon_y - aoff_y);
test.value.z = (short)(aon_z - aoff_z);
Serial.println("Resultados Self-Test");
output(test);
Serial.println("");

// Restaurar los valores iniciales
writeTo(ADXL345_BW_RATE, tmp_rate); // rate 100Hz
writeTo(ADXL345_DATA_FORMAT, tmp_format); // full Res & 16g
delay(100);
}
// OFFSET-CALIBRATION

void offset_CAL(){
    union XYZBuffer xyz;
    byte tmp_rate; //, tmp_format;
    short nro_ave = NRO_PROMEDIO;
    int aoff_x = 0, aoff_y = 0, aoff_z = 0;
    int i;

    readFrom(ADXL345_BW_RATE, 1, &tmp_rate);
    writeTo(ADXL345_BW_RATE, 0x0A); // rate 100Hz
    // readFrom(ADXL345_DATA_FORMAT, 1, &tmp_format);
    // writeTo(ADXL345_DATA_FORMAT, 0x0B); // full Res & 16g

    // Inicializando el offset
    writeTo(ADXL345_OFSX, 0);
    writeTo(ADXL345_OFSY, 0);
    writeTo(ADXL345_OFSZ, 0);
    delay(100);

    // Promedio para obtener offset de cada eje
    for (i= 0; i < nro_ave; i++){
        readAccel(ADXL345_DEVICE,&xyz);
        aoff_x += xyz.value.x;
        aoff_y += xyz.value.y;
        aoff_z += xyz.value.z;
    }
    aoff_x /= nro_ave;
    aoff_y /= nro_ave;
    aoff_z = (aoff_z / nro_ave) - 256 ;

    aoff_x = -round(aoff_x/4);
    aoff_y = -round(aoff_y/4);
    aoff_z = -round(aoff_z/4);
    writeTo(ADXL345_OFSX, aoff_x);
    writeTo(ADXL345_OFSY, aoff_y);
    writeTo(ADXL345_OFSZ, aoff_z);

    Serial.println("Resultados Offset-Calibration");
    Serial.print(aoff_x);
    Serial.print(", ");
    Serial.print(aoff_y);

```

```

    Serial.print(", ");
    Serial.println(aoff_z);

// Restaurar los valores iniciales
    writeTo(ADXL345_BW_RATE, tmp_rate); // rate 100Hz
//    writeTo(ADXL345_DATA_FORMAT, tmp_format); // full Res & 16g
    delay(100);
}
void setup()
{
    Serial.begin(57600); // start serial for output
    Wire.begin(); // join i2c bus (address optional for master)
    setupAccel(ADXL345_DEVICE);
}
void loop()
{
    union XYZBuffer accel;
    readAccel(ADXL345_DEVICE,&accel);
    output(accel);
    Serial.println("");

//Verifico el teclado
    if(Serial.available() > 0){
        incoming[0] = Serial.read();
        switch(incoming[0]) {
            case 'a':
                printAllRegister();
                delay(5000);
                break;
            case 't':
                self_test();
                delay(5000);
                break;
            case 'q':
                break;
            case 'b':
                Serial.write(bufChar, sizeof(struct eventos));
                break;
            case 'c':
                offset_CAL();
                delay(5000);
                break;
            case 's':
                break;
            default:
                Serial.println(incoming[0]);
                break;
        }
    }
}

```

Apéndice C

Abreviaturas.

AMR: *Anisotropic magnetoresistive*. Anisotrópico magnetorresistivo.

BP: *Back propagation*. Propagación hacia atrás.

CMOS: *Complementary metal-oxide-semiconductor*. Semiconductor complementario de óxido metálico.

DPS: *Degrees per second*. Grados por segundo.

EMC: *Electromagnetic compatibility*. Compatibilidad electromagnética.

EP: Enfermedad de Parkinson.

FIFO: *First-in, First-out*. Primero en entrar, primero en salir.

GND: *Ground*. Controlador de tierra.

GUI: *Graphical user interface*. Constructor de interfaz gráfica.

IDE: *Integrated Development Environment*. Entorno de desarrollo integrado.

IRM: Imágenes por resonancia magnética.

I²C: *inter integrated circuits*. Circuito inter-intergrado.

JTAG: *Join Test Action Group*. Verificador de circuitos impresos.

K-NN: *K-nearest neighbors*. K- vecinos más cercanos.

LSB: *Least Significant Bit*. El bit menos significativo.

MEMS: *microelectromechanical systems*. Sistema micro-electromecánico.

MLP: *Multilayer perceptron*. Perceptrón multicapa.

OMS: Organización Mundial de la Salud.

PCA: *Principal Component Analysis* Análisis de componentes principales.

PET: *Positron Emission Tomography*. Tomografía por emisión de positrones.

RBF: *Base Data File*. Archivos de base de datos.

RM: Resonancia magnética.

RNA: Redes neuronales artificiales.

SCL: Línea de reloj.

SDA: Línea de datos.

SPI: *Serial Peripheral Interface*. Interfaz de periféricos de serie.

SRAM: *Static Random Access Memory*. Memoria estática de acceso directo.

TWI: *Two-Wire Interface*. Interfaz de dos cables.

UPDRS: *Unified Parkinson's Disease Rating Scale*. Escala unificada para la evaluación de la EP.

Apéndice D

Glosario.

- **Astenia:** Debilidad o fatiga general que dificulta o impide a una persona realizar tareas que en condiciones normales hace fácilmente.
- **Bias:** Es un término asociado al sesgo, ya que controla qué tan predispuesta está la neurona a disparar un 1 o un 0 independientemente de los pesos. Un sesgo alto hace que la neurona requiera una entrada más alta para generar una salida de 1.
- **Catecolamina:** Son hormonas que se vierten en el torrente sanguíneo. Son un grupo de sustancias que incluyen la adrenalina, la noradrenalina y la dopamina. Contienen un grupo catecol y un grupo amino.
- **Carbidopa:** Es un medicamento que se administra a las personas con enfermedad de Parkinson con el fin de inhibir el metabolismo periférico de levodopa. Se usa conjuntamente con levodopa, incrementando la vida media de esta en el plasma y la concentración de dopamina en el cerebro.
- **Colinérgico:** Se refiere típicamente a circuitos neuronales, medicamentos, moléculas y proteínas que hacen uso, transportan o modifican la actividad del neurotransmisor.
- **Covarianza:** Es el valor que refleja en qué cantidad dos variables aleatorias varían de forma conjunta respecto a sus medias.
- **Dopamina:** Es un neurotransmisor con múltiples e importantes funciones dentro del sistema nervioso. Es frecuentemente mencionada como la causante de las sensaciones placenteras y la sensación de relajación. Fue sintetizada en 1910 por los biólogos ingleses George Barger y James Ewens.
- **Epoques:** Se refiere a un largo período de tiempo, especialmente uno en el que hay nuevos desarrollos y grandes cambios.
- **Levodopa:** También conocido como L-DOPA, el precursor metabólico de la dopamina, es el medicamento aislado más eficaz en el tratamiento de la EP; los efectos terapéuticos y adversos de la levodopa resultan de su descarboxilación en dopamina. Se administra en lugar de la dopamina, porque esta última no puede atravesar la barrera hematoencefálica.
- **Outlier:** Es una observación anormal y extrema en una muestra estadística o serie temporal de datos que puede afectar potencialmente a la estimación de los parámetros del mismo.

- **Palidotomía:** Es un procedimiento neuroquirúrgico mediante el cual se coloca una pequeña sonda eléctrica en el globo pálido, que luego se calienta a 80 °C durante 60 segundos, para destruir una pequeña área de células cerebrales.
- **Rouché-Frobenius:** Permite calcular el número de soluciones de un sistema de ecuaciones lineales en función del rango de la matriz de coeficientes y del rango de la matriz ampliada asociadas al sistema.
- **Signo de Babinski:** Es la extensión dorsal del dedo gordo del pie y generalmente acompañado de la apertura en abanico de los demás dedos en respuesta a la estimulación plantar del pie, signo característico de lesión del tracto piramidal o fascículo corticoespinal llamado síndrome de neurona motora superior.
- **Síndrome serotoninérgico:** Es un conjunto de síntomas causados por un exceso de serotonina, un neurotransmisor importante para el buen humor, sueño, alimentación, percepción del dolor, placer y deseo sexual.
- **Varianza:** Es una medida de la dispersión que representa la variabilidad de una serie de datos respecto a su media. Formalmente se calcula como la suma de los residuos al cuadrado divididos entre el total de observaciones. También se puede calcular como la desviación típica al cuadrado.

www.bdigital.ula.ve

Apéndice E

Fichas técnicas (Datasheets).

Se muestra un resumen del funcionamiento y características de los componentes del sensor GY-80 que fueron descritos en la sección **3.2.1**. Los *datasheets* completos se encuentran en las referencias: [45], [44], [46].

www.bdigital.ula.ve

FEATURES

- Ultralow power:** as low as 23 μA in measurement mode and 0.1 μA in standby mode at $V_S = 2.5\text{ V}$ (typical)
- Power consumption scales automatically with bandwidth**
- User-selectable resolution**
 - Fixed 10-bit resolution
 - Full resolution, where resolution increases with g range, up to 13-bit resolution at $\pm 16\text{ g}$ (maintaining 4 mg/LSB scale factor in all g ranges)
- Patent pending, embedded memory management system with FIFO technology minimizes host processor load**
- Single tap/double tap detection**
- Activity/inactivity monitoring**
- Free-fall detection**
- Supply voltage range:** 2.0 V to 3.6 V
- I/O voltage range:** 1.7 V to V_S
- SPI (3- and 4-wire) and I²C digital interfaces**
- Flexible interrupt modes mappable to either interrupt pin**
- Measurement ranges selectable via serial command**
- Bandwidth selectable via serial command**
- Wide temperature range** (-40°C to $+85^\circ\text{C}$)
- 10,000 g shock survival**
- Pb free/RoHS compliant**
- Small and thin:** 3 mm $\times$ 5 mm $\times$ 1 mm LGA package

APPLICATIONS

- Handsets
- Medical instrumentation
- Gaming and pointing devices
- Industrial instrumentation
- Personal navigation devices
- Hard disk drive (HDD) protection

GENERAL DESCRIPTION

The ADXL345 is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to $\pm 16\text{ g}$. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0° .

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Free-fall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated, patent pending memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm $\times$ 5 mm $\times$ 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

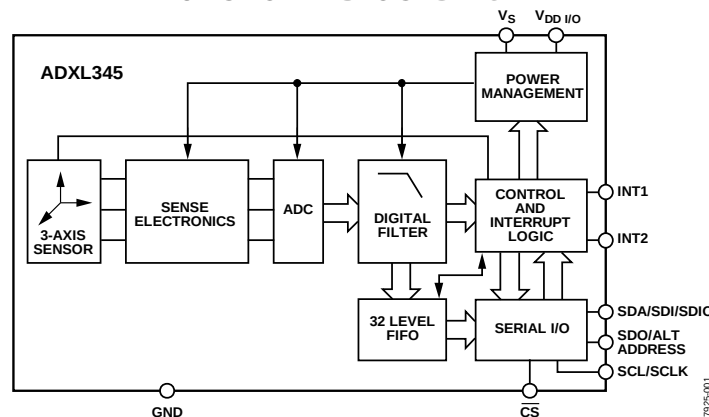


Figure 1.

Rev. D

[Document Feedback](#)

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 ©2009–2013 Analog Devices, Inc. All rights reserved.
[Technical Support](#) www.analog.com

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 2.5\text{ V}$, $V_{DD\text{ I/O}} = 1.8\text{ V}$, acceleration = 0 g, $C_S = 10\text{ }\mu\text{F}$ tantalum, $C_{I/O} = 0.1\text{ }\mu\text{F}$, output data rate (ODR) = 800 Hz, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Test Conditions	Min	Typ ¹	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range	User selectable		$\pm 2, \pm 4, \pm 8, \pm 16$		g
Nonlinearity	Percentage of full scale		± 0.5		%
Inter-Axis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ²			± 1		%
OUTPUT RESOLUTION	Each axis				
All g Ranges	10-bit resolution		10		Bits
$\pm 2\text{ g}$ Range	Full resolution		10		Bits
$\pm 4\text{ g}$ Range	Full resolution		11		Bits
$\pm 8\text{ g}$ Range	Full resolution		12		Bits
$\pm 16\text{ g}$ Range	Full resolution		13		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	All g-ranges, full resolution	230	256	282	LSB/g
	$\pm 2\text{ g}$, 10-bit resolution	230	256	282	LSB/g
	$\pm 4\text{ g}$, 10-bit resolution	115	128	141	LSB/g
	$\pm 8\text{ g}$, 10-bit resolution	57	64	71	LSB/g
	$\pm 16\text{ g}$, 10-bit resolution	29	32	35	LSB/g
Sensitivity Deviation from Ideal	All g-ranges		± 1.0		%
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT}	All g-ranges, full resolution	3.5	3.9	4.3	mg/LSB
	$\pm 2\text{ g}$, 10-bit resolution	3.5	3.9	4.3	mg/LSB
	$\pm 4\text{ g}$, 10-bit resolution	7.1	7.8	8.7	mg/LSB
	$\pm 8\text{ g}$, 10-bit resolution	14.1	15.6	17.5	mg/LSB
	$\pm 16\text{ g}$, 10-bit resolution	28.6	31.2	34.5	mg/LSB
Sensitivity Change Due to Temperature			± 0.01		%/ $^\circ\text{C}$
0 g OFFSET	Each axis				
0 g Output for X_{OUT} , Y_{OUT}		-150	0	+150	mg
0 g Output for Z_{OUT}		-250	0	+250	mg
0 g Output Deviation from Ideal, X_{OUT} , Y_{OUT}			± 35		mg
0 g Output Deviation from Ideal, Z_{OUT}			± 40		mg
0 g Offset vs. Temperature for X-, Y-Axes			± 0.4		mg/ $^\circ\text{C}$
0 g Offset vs. Temperature for Z-Axis			± 1.2		mg/ $^\circ\text{C}$
NOISE					
X-, Y-Axes	ODR = 100 Hz for $\pm 2\text{ g}$, 10-bit resolution or all g-ranges, full resolution		0.75		LSB rms
Z-Axis	ODR = 100 Hz for $\pm 2\text{ g}$, 10-bit resolution or all g-ranges, full resolution		1.1		LSB rms
OUTPUT DATA RATE AND BANDWIDTH	User selectable				
Output Data Rate (ODR) ^{3, 4, 5}		0.1		3200	Hz
SELF-TEST ⁶					
Output Change in X-Axis		0.20		2.10	g
Output Change in Y-Axis		-2.10		-0.20	g
Output Change in Z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD\text{ I/O}}$)		1.7	1.8	V_S	V
Supply Current	ODR $\geq 100\text{ Hz}$		140		μA
	ODR $< 10\text{ Hz}$		30		μA
Standby Mode Leakage Current			0.1		μA
Turn-On and Wake-Up Time ⁷	ODR = 3200 Hz		1.4		ms



L3G4200D

MEMS motion sensor: ultra-stable three-axis digital output gyroscope

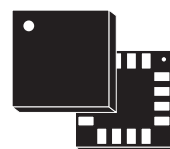
Preliminary data

Features

- Three selectable full scales (250/500/2000 dps)
- I²C/SPI digital output interface
- 16 bit-rate value data output
- 8-bit temperature data output
- Two digital output lines (interrupt and data ready)
- Integrated low- and high-pass filters with user-selectable bandwidth
- Ultra-stable over temperature and time
- Wide supply voltage: 2.4 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Embedded power-down and sleep mode
- Embedded temperature sensor
- Embedded FIFO
- High shock survivability
- Extended operating temperature range (-40 °C to +85 °C)
- ECOPACK® RoHS and "Green" compliant

Applications

- Gaming and virtual reality input devices
- Motion control with MMI (man-machine interface)
- GPS navigation systems
- Appliances and robotics



LGA-16 (4x4x1.1 mm)

Description

The L3G4200D is a low-power three-axis angular rate sensor able to provide unprecedented stability of zero rate level and sensitivity over temperature and time. It includes a sensing element and an IC interface capable of providing the measured angular rate to the external world through a digital interface (I²C/SPI).

The sensing element is manufactured using a dedicated micro-machining process developed by STMicroelectronics to produce inertial sensors and actuators on silicon wafers.

The IC interface is manufactured using a CMOS process that allows a high level of integration to design a dedicated circuit which is trimmed to better match the sensing element characteristics.

The L3G4200D has a full scale of $\pm 250/\pm 500/\pm 2000$ dps and is capable of measuring rates with a user-selectable bandwidth.

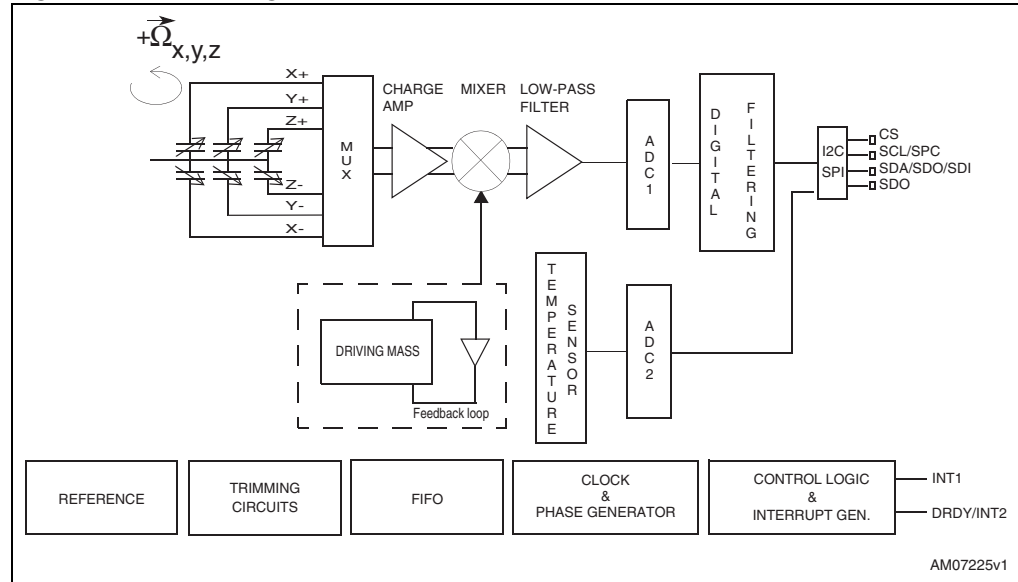
The L3G4200D is available in a plastic land grid array (LGA) package and can operate within a temperature range of -40 °C to +85 °C.

Table 1. Device summary

Order code	Temperature range (°C)	Package	Packing
L3G4200D	-40 to +85	LGA-16 (4x4x1.1 mm)	Tray
L3G4200DTR	-40 to +85	LGA-16 (4x4x1.1 mm)	Tape and reel

1 Block diagram and pin description

Figure 1. Block diagram

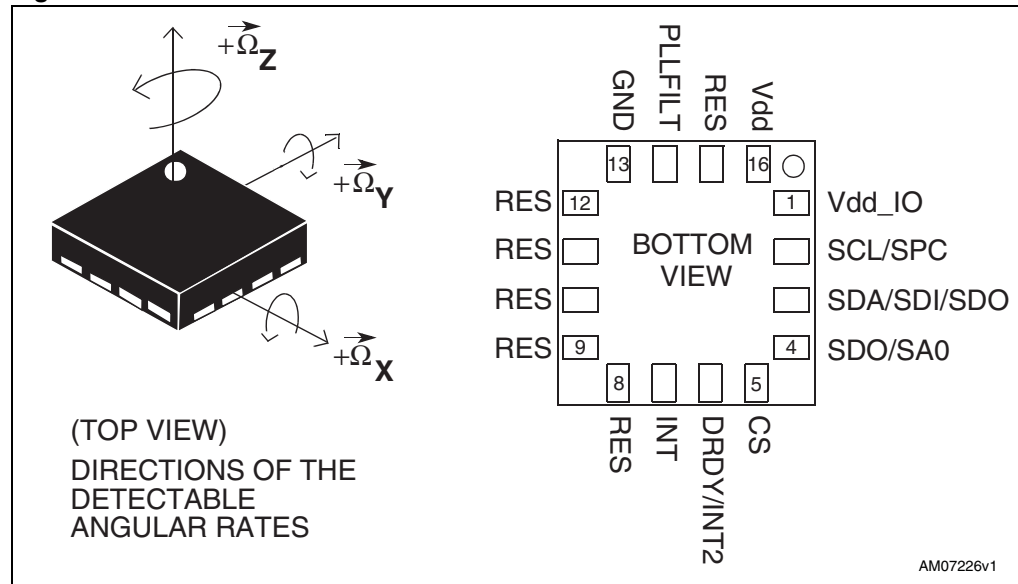


The vibration of the structure is maintained by drive circuitry in a feedback loop. The sensing signal is filtered and appears as a digital signal at the output.

1.1

Pin description

Figure 2. Pin connection



2 Mechanical and electrical characteristics

2.1 Mechanical characteristics

Table 4. Mechanical characteristics @ Vdd = 3.0 V, T = 25 °C, unless otherwise noted⁽¹⁾

Symbol	Parameter	Test condition	Min.	Typ. ⁽²⁾	Max.	Unit
FS	Measurement range	User-selectable		±250		dps
				±500		
				±2000		
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		
SoDr	Sensitivity change vs. temperature	From -40 °C to +85 °C		±2		%
DVoff	Digital zero-rate level	FS = 250 dps		±10		dps
		FS = 500 dps		±15		
		FS = 2000 dps		±75		
OffDr	Zero-rate level change vs. temperature ⁽³⁾	FS = 250 dps		±0.03		dps/°C
		FS = 2000 dps		±0.04		dps/°C
NL	Non linearity ⁽⁴⁾	Best fit straight line		0.2		% FS
DST	Self-test output change	FS = 250 dps		130		dps
		FS = 500 dps		200		
		FS = 2000 dps		530		
Rn	Rate noise density	BW = 50 Hz		0.03		dps/ sqrt(Hz)
ODR	Digital output data rate			100/200/ 400/800		Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V. The operational power supply range is specified in [Table 5](#).

2. Typical specifications are not guaranteed.

3. Min/max values have been estimated based on the measurements of the current gyros in production.

4. Guaranteed by design.

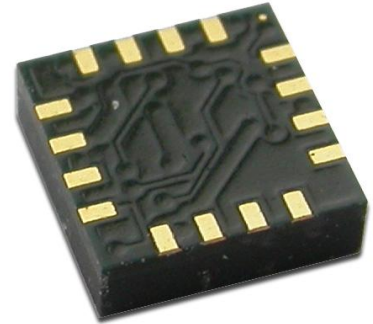
3-Axis Digital Compass IC

HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I²C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES

BENEFITS

- | | |
|--|--|
| ▶ 3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package | ▶ Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly |
| ▶ 12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields | ▶ Enables 1° to 2° Degree Compass Heading Accuracy |
| ▶ Built-In Self Test | ▶ Enables Low-Cost Functionality Test after Assembly in Production |
| ▶ Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA) | ▶ Compatible for Battery Powered Applications |
| ▶ Built-In Strap Drive Circuits | ▶ Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation |
| ▶ I ² C Digital Interface | ▶ Popular Two-Wire Serial Data Interface for Consumer Electronics |
| ▶ Lead Free Package Construction | ▶ RoHS Compliance |
| ▶ Wide Magnetic Field Range (+/-8 Oe) | ▶ Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy |
| ▶ Software and Algorithm Support Available | ▶ Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available |
| ▶ Fast 160 Hz Maximum Output Rate | ▶ Enables Pedestrian Navigation and LBS Applications |

HMC5883L

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
-----------------	-------------	-----	-----	-----	-------

Power Supply

Supply Voltage	VDD Referenced to AGND VDDIO Referenced to DGND	2.16 1.71	2.5 1.8	3.6 VDD+0.1	Volts Volts
Average Current Draw	Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply)	- -	2 100	- -	μA μA

Performance

Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Applied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurement Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I2C commands Analog Circuit Ready for Measurements		200 50		μs ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I ² C Address	8-bit read address 8-bit write address		0x3D 0x3C		hex hex
I ² C Rate	Controlled by I ² C Master			400	kHz
I ² C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		Volts Volts
Self Test	X & Y Axes Z Axis		±1.16 ±1.08		gauss
	X & Y & Z Axes (GN=5) Positive Bias X & Y & Z Axes (GN=5) Negative Bias	243 -575		575 -243	LSb
Sensitivity Tempco	T _A = -40 to 125°C, Uncompensated Output		-0.3		%/°C

General

ESD Voltage	Human Body Model (all pins) Charged Device Model (all pins)			2000 750	Volts
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C

HMC5883L

PCB Pad Definition and Traces

The HMC5883L is a fine pitch LCC package. Refer to previous figure for recommended PCB footprint for proper package centering. Size the traces between the HMC5883L and the external capacitors (C1 and C2) to handle the 1 ampere peak current pulses with low voltage drop on the traces.

Stencil Design and Solder Paste

A 4 mil stencil and 100% paste coverage is recommended for the electrical contact pads.

Reflow Assembly

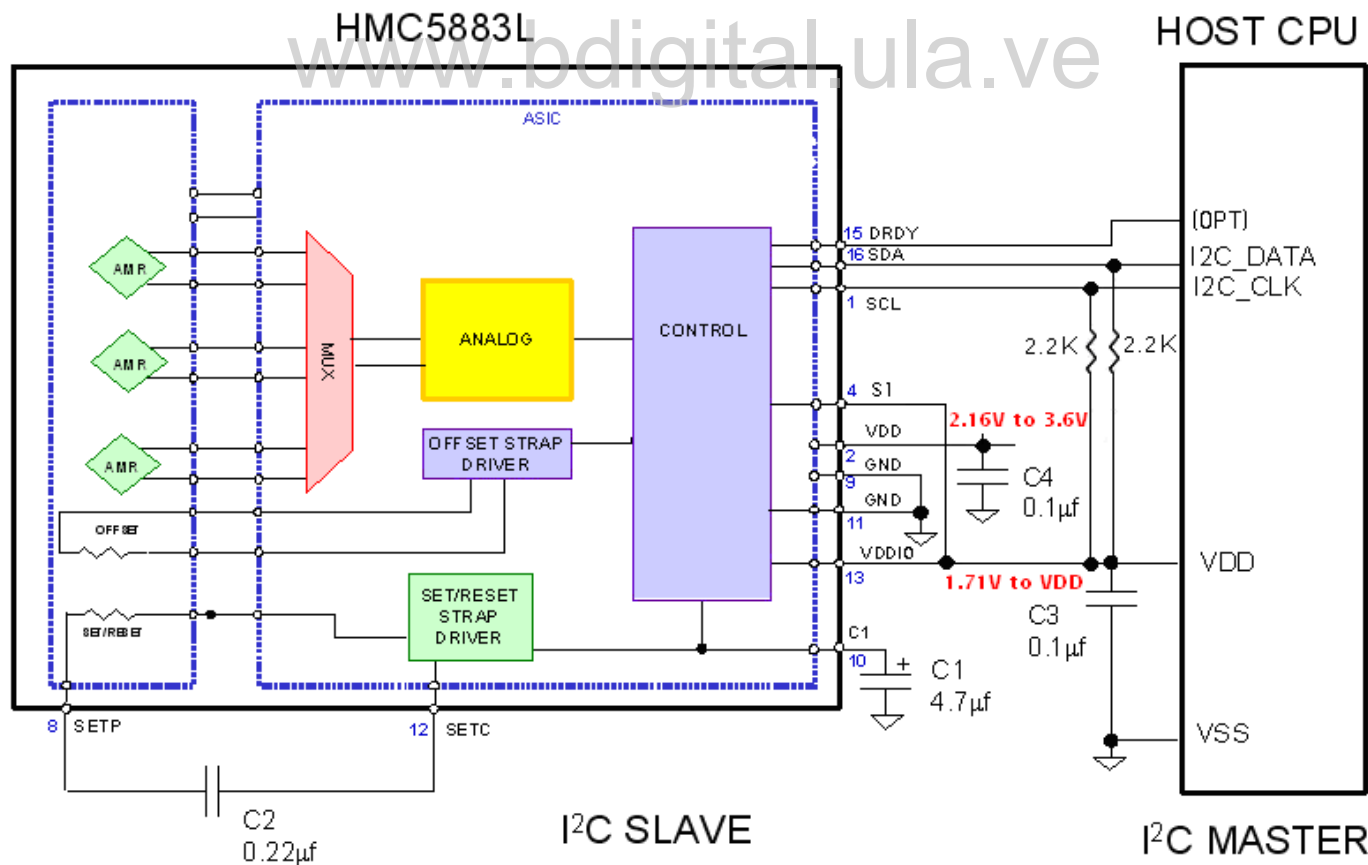
This device is classified as MSL 3 with 260°C peak reflow temperature. A baking process (125°C, 24 hrs) is required if device is not kept continuously in a dry (< 10% RH) environment before assembly. No special reflow profile is required for HMC5883L, which is compatible with lead eutectic and lead-free solder paste reflow profiles. Honeywell recommends adherence to solder paste manufacturer's guidelines. Hand soldering is not recommended. Built-in self test can be used to verify device functionalities after assembly.

External Capacitors

The two external capacitors should be ceramic type construction with low ESR characteristics. The exact ESR values are not critical but values less than 200 milli-ohms are recommended. Reservoir capacitor C1 is nominally 4.7 μF in capacitance, with the set/reset capacitor C2 nominally 0.22 μF in capacitance. Low ESR characteristics may not be in many small SMT ceramic capacitors (0402), so be prepared to up-size the capacitors to gain Low ESR characteristics.

INTERNAL SCHEMATIC DIAGRAM

HMC5883L



Apéndice F

Preguntas y observaciones del jurado.

La defensa de esta tesis se realizó en medio de la pandemia mundial originada por el COVID-19, el jurado no estuvo expuesto al riesgo de contagio, por esta razón la defensa no se dio en forma presencial, así que las observaciones y respuestas a las preguntas formuladas por el jurado también se hacen públicas en este documento.

Observaciones del jurado Dr. John D. Ferreira Rodríguez

“Tomando en cuenta lo complicado que es el estudio y diagnóstico de la enfermedad de Parkinson, la contribución que puede realizar el uso de la instrumentación, el análisis de datos y redes neuronales para un gran número de variables será una poderosa herramienta que ayudará a los médicos en su diagnóstico. Considero que la tesis es muy buena, bastante extensa y también compleja. Ciertamente, se puede considerar una tesis en ciencias aplicadas”.

Preguntas sobre diferentes tópicos relacionadas con la Tesis.

1) En trabajos previos, se observó que para estudiar la enfermedad de Parkinson se emplearon el análisis PCA y redes neuronales. En el trabajo desarrollado en la presente tesis se emplearon estas 2 técnicas. Teniendo presente lo anterior, mi pregunta es la siguiente: ¿Qué aportes originales realizó usted en el actual trabajo, a fin de desarrollar un diagnóstico temprano de la enfermedad?

- Se propuso un sistema de detección no invasivo basado en la dinámica corporal en especial de las extremidades superiores del paciente.
- A nivel local, nacional e incluso internacional, no hay trabajos previos conocidos en el área de detección temprana de la enfermedad de Parkinson a través de redes neuronales, con esta tesis se puede motivar el estudio de nuevas técnicas para la detección o incluso para el desarrollo de terapias no invasivas que ayuden a mitigar los síntomas de la enfermedad.
- La propuesta de estudiar la enfermedad a partir de la dinámica corporal del paciente, en especial del estudio del movimiento de las extremidades superiores y la creación de una base de datos de 60 sujetos sanos y 6 pacientes con la enfermedad en sus primeros estadios.

- La creación de una llave de identificación única de cada sujeto estudiado a partir de la dinámica corporal.
- La incorporación de dendrogramas de clasificación como una técnica de visualización, ya que en ningún trabajo se encontró el uso de esta herramienta.

2) ¿Cuál es la función del protocolo I2C?

El protocolo I2C se trata de un bus bidireccional que utiliza dos líneas, una de datos serie (SDA) y otra de reloj serie (SCL). SCL es la línea de reloj, se utiliza para sincronizar todos los datos SDA de las transferencias. SDA es la línea de datos. Gracias a este protocolo se puede controlar al mismo tiempo un importante número de dispositivos con el Arduino.

Los dispositivos conectados al I2C tienen una dirección única para cada uno y cada mensaje y orden que se transmite al bus, lleva anexa esta dirección, indicando cuál de los muchos posibles, es el receptor del mensaje. También pueden ser maestros o esclavos. El dispositivo maestro inicia la transferencia de datos y además genera la señal de reloj.

Para usar el I2C en Arduino, el entorno de desarrollo proporciona la librería “Wire.h”, que contiene las funciones necesarias para controlar el hardware integrado.

3) ¿Considera usted suficiente el tiempo de 120 segundos para la toma de datos por tarea?

Si, esto se determinó después de realizar varias pruebas donde el tiempo de toma de datos fue superior. Se pudo observar que con tiempos mayores a los 3 minutos, las personas se fatigaban y también había que considerar que eran 8 tareas, no se podía someter a las personas a un examen muy largo.

Por otra parte, con un bus de datos más grande por tarea, la matriz de covarianzas tenía dimensiones muy altas por lo que el proceso de cómputo no se podía realizar ya que la computadora se quedaba sin memoria y solo se podía incluir una cantidad pequeña de sujetos a entrenar y para el propósito del estudio se requería tener la mayor cantidad posible de sujetos para el entrenamiento.

4) El ruido se puede clasificar en varias categorías ¿Cuál considera usted es el tipo de ruido predominante en los transductores y circuitos del sistema de medición en la presente investigación?

El ruido interno que es el producido en el interior de los dispositivos electrónicos como consecuencia de su naturaleza física. Entre ellos los siguientes:

- Ruido térmico: Es el movimiento de los electrones dentro de un conductor causado por la agitación térmica.
- Ruido de disparo: Causado por las fluctuaciones aleatorias de la corriente eléctrica.
- Ruido de tiempo de disparo: Es el producido en los transistores por su naturaleza física.

El ruido externo que se produce en menor o mayor medida por algunos equipos que pueden afectar a la señal, como la computadora, celulares, cables e incluso la fuente de alimentación de Arduino. Son interferencias producidas por acoplamiento eléctrico y/o magnético.

Otro ruido predominante se genera por la fricción entre el vidrio y la caja del sensor GY-80, ya que éste es muy sensible a cualquier perturbación externa. Sin embargo, todos esos valores atípicos se eliminan cuando se hace la depuración de los datos.

5) Respecto a los programas empleados para la toma de datos de los diferentes sensores presentes en la tarjeta Arduino ¿Fueron los mismos elaborados completamente para la presente investigación o fueron adaptados de programas previamente elaborados en otros trabajos?

El programa “Calibración acelerómetro ADXL345” se descargó directamente para su uso, solo se modificaron algunos valores.

Los programas “Aqevent” y “Todos los sensores” si fueron elaborados para la investigación.

Por otra parte, con respecto a los programas de MATLAB, se adaptaron algunos que fueron elaborados en la cátedra Inteligencia Artificial previamente cursada y la mayor parte fueron elaborados para la investigación, como los del análisis de la dinámica, algunas funciones para el programa de análisis de componentes principales y los programas usados para la clasificación.

6) En el análisis de datos se empleó el modelo de red neuronal denominado perceptrón multicapa ¿Consideró usted el uso de otros tipos de redes neuronales? ¿Qué criterios la llevaron a seleccionar este modelo?

No consideré el uso de otras redes neuronales ya que los resultados vistos de otras investigaciones le daban al perceptrón multicapa confiabilidad para resolver problemas que no son linealmente separables.

En la cátedra Inteligencia Artificial ya lo había utilizado y conocía su eficiencia para el reconocimiento de caras y patrones, por lo que para el propósito de nuestra investigación que era encontrar una clasificación de los pacientes con la enfermedad de Parkinson a partir de la dinámica corporal, era la mejor opción.

7) Considerando que los resultados mostrados en los dendrogramas de clasificación y distancia de Mahalanobis por tarea, existen algunos pacientes con Parkinson que caen dentro del grupo de los pacientes sanos ¿A qué atribuye usted estos resultados? ¿Es necesario mayor entrenamiento de la red neuronal? ¿Hace falta una mayor cantidad de personas realicen las tareas elaboradas para este fin? ¿Se puede incrementar el nivel de eficiencia?

¿A qué atribuye usted estos resultados?

Esto se debe a que el empleo de una sola tarea no es suficiente para la extracción de características que permitan una diferenciación entre los sujetos sanos y los pacientes con la enfermedad, además

algunos pacientes están medicados, es probable que eso influya cuando solo se ejecuta una tarea.

¿Es necesario mayor entrenamiento de la red neuronal?

Se hizo un gran número de entrenamientos de la red neuronal con diferentes arquitecturas, el mostrado fue con el que se logró la mayor eficiencia. Si la red se sobreentrena, el algoritmo puede quedar ajustado a características muy específicas, donde solo ‘verá’ los patrones ya entrenados y no reconocerá a los nuevos presentados.

¿Hace falta una mayor cantidad de personas realicen las tareas elaboradas para este fin?

Sí, es una de las recomendaciones, con un mayor número de personas el algoritmo será más robusto.

¿Se puede incrementar el nivel de eficiencia?

Si, con una base de datos más grande seguramente la eficiencia aumentará.

Observaciones del jurado Dr. Marcos Rodríguez

Respuesta del profesor Dr. Marcos Rodríguez enviada al tutor Dr. Misael D Rosales:

“Mi opinión no puede ser otra que la de felicitar a los dos, a la estudiante y a ti. El trabajo destaca muy por encima de lo que uno está acostumbrado a leer y juzgar, y sería una grosería y una ridiculez de mi parte opinar algo crítico. Por las técnicas que se utilizaron no me ha sido fácil seguir el hilo y he aprendido muchas cosas. Es excelente, impecable y lo apruebo sin observaciones. Puedes redactar el veredicto aprobatorio y, por supuesto, lo califico con un 20 bien gordo y lo juzgo digno de publicación.

De nuevo, felicitaciones a la candidata y a ti. Gracias por la oportunidad, ha sido un placer y me ha subido el ánimo saber que ese tipo de trabajos con aplicación médica casi inmediata y resultados probados son posibles aun en la situación que vivimos”.

Bibliografía

- [1] National Institute of Neurological Disorders and Stroke (NIH). Enfermedad de Parkinson: Esperanza en la investigación. [Online]. Disponible en: <https://espanol.ninds.nih.gov/trastornos/parkinson>, 2016.
- [2] Mizuno Y. Progress in the pathogenesis and genetics of Parkinson's disease. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 363(1500):2215–2227, 2008.
- [3] Torres R., Huerta M., Clotet R., González R., Sagbay G., Erazo M., and Pirrone J. Diagnosis of the corporal movement in Parkinson's disease using kinect sensors. *World Congress on Medical Physics and Biomedical Engineering*, 51(1), June 2017.
- [4] Bermeo A. and Bravo M. *Diseño y desarrollo de un sistema inalámbrico que permita monitorear los temblores en pacientes que padecen la enfermedad de Parkinson utilizando software y hardware libre*. Universidad Politécnica Salesiana, Ecuador, 2016.
- [5] Perumal S. and Sankar R. Gait and tremor assessment for patients with Parkinson's disease using wearable sensors. *ICT Express*, 2:168–174, 2016.
- [6] Monsalve A. *Sistemas de Ayuda a la Decisión Clínica en Enfermedades de Diagnóstico Complejo*. Universidad de los Andes, Venezuela, 2017.
- [7] Chaná P., Albuquerque D., Aránguiz R., Baldwin N., and Benavides O. *Enfermedad de Parkinson*. Universidad de Chile, Facultad de Ciencias Médicas CETRAM- USACH, 2010.
- [8] Federación Española de Parkinson. ¿Qué es el Parkinson? [Online]. Disponible en:, 2017.
- [9] Programa de Parkinson y Movimientos Anormales UBA. El diagnóstico en la enfermedad de Parkinson y los biomarcadores. [Online]. Disponible en: <http://neurologia.hospitaldeclinicas.uba.ar>, 2017.
- [10] Campaña S. and Londoño J. Estudio de redes de sensores y aplicaciones orientadas a la recolección y análisis de señales biomédicas. *Gerenci.Tecnol.Inform*, 12(33):85–99, 2013.
- [11] Belén A. *Diagnóstico de la enfermedad de Parkinson usando deep learning y grabaciones de voz mediante teléfono móvil*. Universidad Politécnica de Madrid, 2019.
- [12] Sivaranjini S. and Sujatha C. Deep learning based diagnosis of Parkinson's disease using convolutional neural network. *Springer US*, pages 1–13, 2019.
- [13] Caliskan A., Badem H., Bastürk A., and Yüksel M. Diagnosis of the Parkinson disease by using deep neural network classifier. *IU-JEEE*, 17(2):3311–3318, 2017.

- [14] Rustempasic I. and Can M. Diagnosis of Parkinson's disease using Principal Component Analysis and Boosting Committee Machines. *SOUTHEAST EUROPE JOURNAL OF SOFT COMPUTING*, 2:102–109, 03 2013.
- [15] Soleimani F. and Mohammadi P. A Case Study of Parkinson's Disease Diagnosis using Artificial Neural Networks. *International Journal of Computer Applications*, 73(19):1–6, 2013.
- [16] Geman O., Pohoata S., and Graur A. Acquisition and processing data for early stage of Parkinson's disease. *Génie biomédicale*, 58(3):324–334, 2013.
- [17] Khemphila A. and Boonjing V. Parkinsons disease classification using neural network and feature selection. *International Scholarly and Scientific Research and Innovation*, 6:377–380, 2012.
- [18] Boston C. La inteligencia artificial ya diagnostica enfermedades tan bien como los médicos. [Online]. Disponible en: <https://www.lavanguardia.com/ciencia/cuerpo-humano/20180223/44950677766.html>, 2018.
- [19] Ortiz A., Munilla J., Górriz J., and Ramírez J. Ensembles of deep learning architectures for the early diagnosis of Alzheimer's disease. *International Journal of Neural Systems*, 26(7), 2016.
- [20] Bostan V. and Pantelimon B. Creating a model based on artificial neural network for liver cirrhosis diagnose. *International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, (9):295–298, 2015.
- [21] Atkov O., Gorokhova S., Sboev A., Generozov E., Muraseyeva E., and Moroshkina S. Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters. *Journal of Cardiology*, 59(2), 2012.
- [22] Uğuz H. A biomedical system based on artificial neural network and principal component analysis for diagnosis of the heart valve diseases. *Journal of Medical Systems*, 36(1):61–72, 2012.
- [23] Abdul-Rahman S., Khairil A., Yusoff M., Mohamed A., and Mutalib S. Dermatology diagnosis with feature selection methods and artificial neural networks. *IEEE-EMBS Conference on Biomedical Engineering and Sciences*, 36(1):371–376, 2012.
- [24] Aefifar V., Mazdarani H., Deregeh F., Hayati M., and Payandeh M. Multilayer perceptron neural network with supervised training method for diagnosis and predicting blood disorder and cancer. *IEEE International Symposium on Industrial Electronics*, pages 2075–2080, 2009.
- [25] Ibarra L. *Comparar los efectos del ejercicio, supervisado o no, en la calidad de vida de los pacientes con la Enfermedad de Parkinson*. Universidad de los Andes, Venezuela, 2009.
- [26] Gottapu R. and Dagli C. Analysis of Parkinson's Disease Data. *Procedia Computer Science*, 140:334–341, 2018.
- [27] García J. *Efecto de la clozapina como terapia coadyuvante en el tratamiento del temblor refractario en pacientes con la enfermedad de Parkinson en el I.A.H.U.L.A.* Universidad de los Andes, Venezuela, 2008.

- [28] Prieto J., Alcalde M., López L., Pérez P., Martín L., and Fenández C. Guía de actuación sobre la enfermedad de Parkinson para profesionales de medicina de atención primaria. *CGCOF, FEP, SEMERGEN y SEN*, 2019.
- [29] Andrade A., Alves A., Soares M., Lopes G., Souza A., Bernardino S., and Valdeci C. Human tremor: Origins, detection and quantification. In *Practical Applications in Biomedical Engineering*, chapter 1. IntechOpen, Rijeka, 2013.
- [30] Sociedad Neurológica Argentina. Escala de evaluación neurológica estadios de Hoehn y Yahr. [Online]. Disponible en: <http://www.sns.org.ar>, 2015.
- [31] Grover S., Bhartia S., Akshama, Yadav A., and Seeja K.R. Predicting severity of Parkinson's disease using deep learning. *Procedia Computer Science*, 132:1788–1794, 2018.
- [32] Lázaro A., Puente E., Lázaro M., Capote J., and Alvear D. Posibilidades de un modelo sustituto de incendios mediante el empleo de redes neuronales. *Métodos numéricos para cálculo y diseño en ingeniería: Revista internacional*, 3:129–134, 2013.
- [33] Marín J. Introducción a las Redes Neuronales Aplicadas. [Online]. Disponible en: <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf>, 2019.
- [34] Mora C. *Uso de sistemas inteligentes para la predicción a corto plazo del clima local en La Hechicera, Mérida-Venezuela*. Universidad de los Andes, Venezuela, 2015.
- [35] Torres G. and Tablada C. Redes Neuronales Artificiales. *Facultad de Matemática, Astronomía y Física (FaMAF) - CIEM(CONICET), Universidad Nacional de Córdoba. Argentina*, pages 22–30, 2005.
- [36] Medina C. Perceptrón multicapa. [Online]. Disponible en: <https://www.scribd.com/doc/268885651/Perceptron-multicapa>, 2017.
- [37] Mejía J. Sistema de detección de intrusos en redes de comunicaciones utilizando redes neuronales. *Escuela de Ingeniería, Universidad de las Américas Puebla*, 2014.
- [38] Pérez J. Herramienta MATLAB para la selección de entradas y predicción neuronal de valores de bolsa. *Escuela superior de ingenieros, Universidad de Sevilla*, 2013.
- [39] Marín J. Tema 3: Análisis de componentes principales. [Online]. Disponible en: <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/AMult/tema3am.pdf>, 2019.
- [40] Jolliffe I.T. *Principal Component Analysis*. Springer-Verlag New York, 2002.
- [41] Joaquin R. Análisis de Componentes Principales (PCA) y t-SNE. [Online]. Disponible en: <https://www.cienciadedatos.net/documentos/35-principa-lcomponentan-analysis>, 2017.
- [42] Martínez R. and Romero M. Uso de sensores inerciales en la medición y evaluación de movimiento humano para aplicaciones en la salud. *Universidad Autónoma del Estado de México, Facultad de Ingeniería.*, 2013.
- [43] Electronilab. Tarjeta GY-80 - IMU 10DOF. [Online]. Disponible en: <https://electronilab.co/tienda/tarjeta-gy-80-imu-10dof/>, 2018.

- [44] Honeywell. L3G4200D MEMS motion sensor: three-axis digital output gyroscope. [Online]. Disponible en: <https://strawberry-linux.com/pub/17116.pdf>, 2010.
- [45] Analog Devices. Datasheet: Digital Accelerometer ADXL345. [Online]. Disponible en: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>, 2015.
- [46] Honeywell. Datasheet: 3-axis Digital Compass IC HMC5883L. [Online]. Disponible en: <http://www.farnell.com/datasheets/1683374.pdf>, 2012.
- [47] Yury and Joel. Open Hardware: PulseSensor. [Online]. Disponible en: <https://pulsesensor.com/pages/open-hardware>, 2018.
- [48] Riveros E. *Uso de Arduino en Programación Electrónica con metodología de aprendizaje en problemas*. Univesidad Católica de Santiago de Guayaquil, Ecuador, 2017.
- [49] Arduino. What is Arduino? [Online]. Disponible en: <https://store.arduino.cc/usa/due>, 2019.
- [50] Cordova L. *Aplicación del programa MATLAB en la resolución de ecuaciones diferenciales aplicado a la materia de cálculo tres*. Universidad Católica de Santiago de Guayaquil, Ecuador, 2012.
- [51] Crespo J. Aprendiendo Arduino: Entorno de Programación de Arduino (IDE). [Online]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/03/29/entorno-de-programacion-de-arduino-ide/>, 2016.
- [52] Castro A. *Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM*. E.U.I.T. Telecomunicación, 2013.
- [53] Isaac. Giroscópio:todo lo que necesitas saber. [Online]. Disponible en: hwlibre.com/giroscopio/Como-funciona-un-giroscopio, 2019.
- [54] 5Hertz Electrónica. Sensores magnéticos. [Online]. Disponible en: 5hertz.com/index.php?route=tutoriales/tutorialtutorial-id=1, 2020.
- [55] Llorens D. *Descripción de Interferencias Magnéticas y Procedimiento de Calibración para Magnetómetro HMC5883L*. Instituto Universitario Aeronautico, Córdoba, 2016.
- [56] MacGill M. What should my heart rate be? *Medical News Today*, pages 1–5, 2017.
- [57] Rosales M., Mora C., and Guada C. Redes neuronales en la predicción de micro-clima, zona de estudio La Hechicera-Mérida-Venezuela. *Publicaciones en Ciencias y Tecnología*, 11(2):47–61, 2017.
- [58] Martínez W. and Martínez A. *Exploratory Data Analysis with MATLAB*. Computer Science and Data Analysis Series, 2005.
- [59] Escobedo M. and Plata J. P.Ch.Mahalanobis y las aplicaciones de su distancia estadística. *CULCyT*, 5(27):13–20, 2008.
- [60] Joaquin R. Correlación lineal y regresión simple. [Online]. Disponible en: <https://www.cienciadedatos.net/documentos/24-correlacion-y-regresion-lineal>, 2016.

- [61] Explorable.com. La correlación estadística. [Online]. Disponible en: <https://explorable.com/es/la-correlacion-estadistica>, 2020.
- [62] Moreno A. Base de datos y conocimiento. [Online]. Disponible en: <https://elies.rediris.es/elies9/4-1.htm>, 2000.
- [63] Bock T. What is a dendrogram? [Online]. Disponible en: displayr.com/what-is-dendrogram/, 2018.

www.bdigital.ula.ve