

**PONTIFICIA UNIVERSIDAD  
CATÓLICA DEL PERÚ**

**Escuela de Posgrado**



Diseño de dispositivo bastón electrónico y procedimiento  
clínico avanzado para monitoreo continuo de la rehabilitación  
de marcha con mayor efectividad clínica

Tesis para obtener el grado académico Maestro en  
Ingeniería Biomédica que presenta:

***Javier Augusto Barriga Hoyle***

Asesor:

***Luis Alberto Vilcahuamán Cajacuri***

Lima, 2024

### Informe de Similitud

Yo, Luis Alberto Vilcahuamán Cajacuri, docente de la Escuela de posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada **Diseño de dispositivo bastón electrónico y procedimiento clínico avanzado para monitoreo continuo de la rehabilitación de marcha con mayor efectividad clínica**, del autor **Javier Augusto Barriga Hoyle**, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 23 %. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 06/12/2024. Cabe aclarar que ninguna fuente, incluidas las de Internet, es superior a 1%. Asimismo, se observa mayores casos de similitud en el estudio de estado del arte y en el capítulo de fundamentos, y no en los capítulos de desarrollo de la tesis cuales son los capítulos 3 y 4.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lima, 06 de diciembre del 2024

Apellidos y nombres del asesor: <u>Vilcahuamán Cajacuri, Luis Alberto</u>	
DNI: 08720728	Firma 
ORCID: 0000-0002-7291-101X	



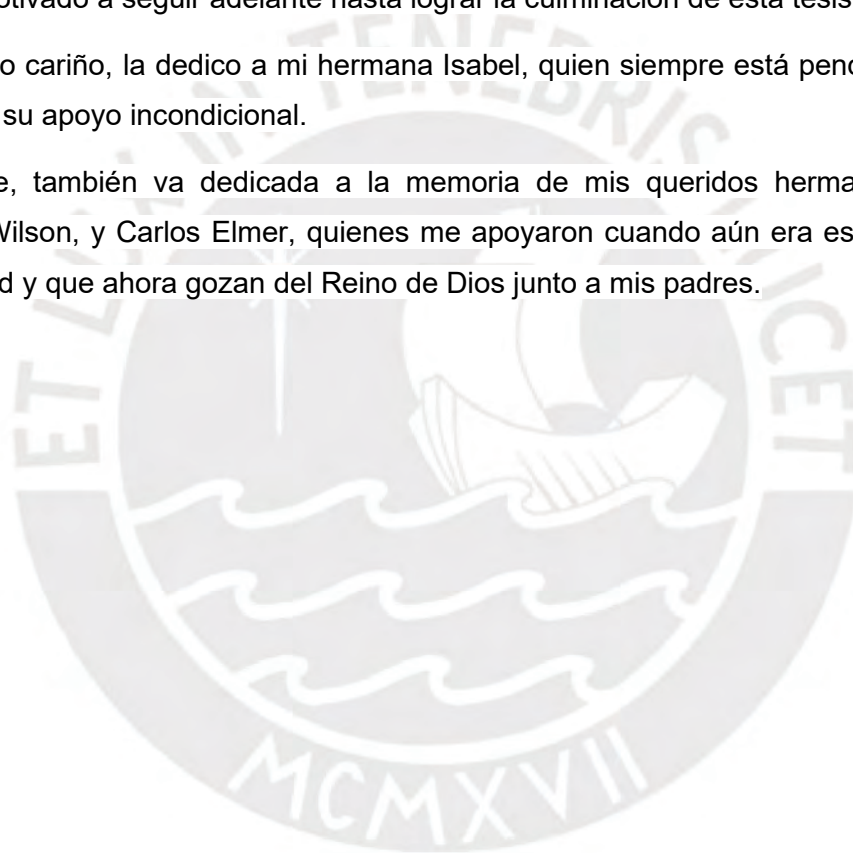
## Dedicatoria

La culminación de esta tesis de maestría está dedicada a la memoria de mis queridos padres, Carlos y Raquel, quienes fueron un eje fundamental en mi vida. Sus sabios consejos y valores, que en su momento me brindaron, hoy los valoro profundamente como profesional y como padre de familia, y siempre trato de aplicarlos en mi trabajo y en mi hogar.

Dedico también esta tesis con todo mi amor y gratitud a mi esposa Carmen, y a mis hijos Carlos Javier y Marissa Cecilia, quienes, con su apoyo, comprensión y amor incondicional, me han motivado a seguir adelante hasta lograr la culminación de esta tesis.

Con mucho cariño, la dedico a mi hermana Isabel, quien siempre está pendiente de mí y me brinda su apoyo incondicional.

Finalmente, también va dedicada a la memoria de mis queridos hermanos Wilfredo, Ernesto, Wilson, y Carlos Elmer, quienes me apoyaron cuando aún era estudiante en la universidad y que ahora gozan del Reino de Dios junto a mis padres.



## Agradecimiento

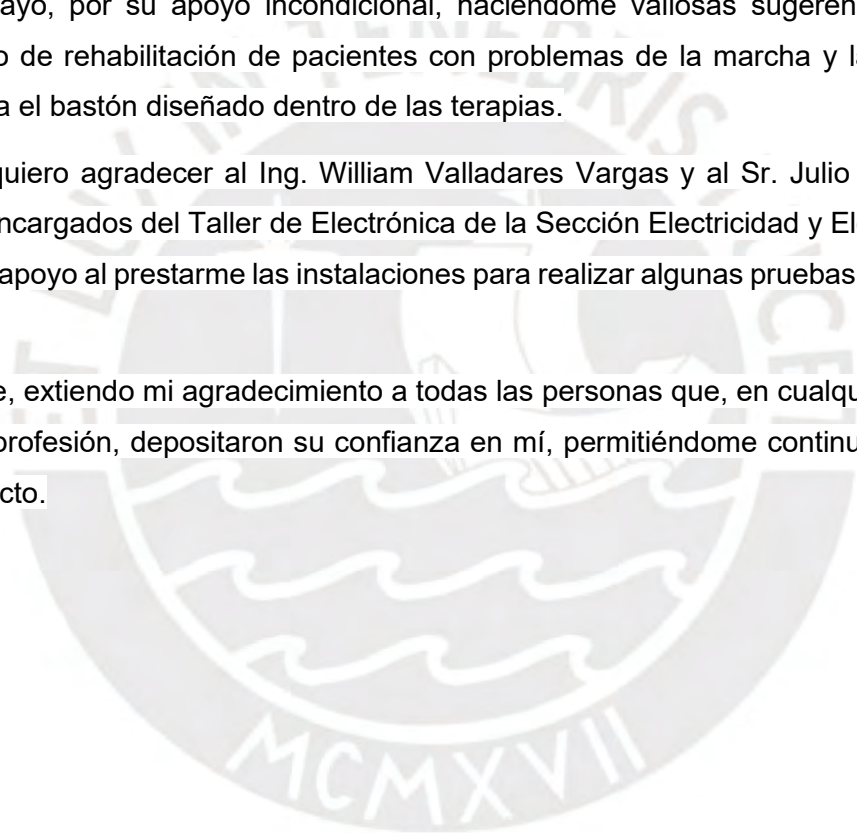
Quiero expresar mi profundo agradecimiento a mi asesor de tesis de maestría, Dr. Luis Vilcahuamán Cajacuri, por su paciencia, dedicación, esclarecedoras explicaciones y gran apoyo, lo cual fue fundamental para la culminación de este trabajo, desde que iniciamos este proyecto hace unos años.

Mi gratitud también va al investigador Marko Chu Válcarcel, por su invaluable asesoría en el desarrollo del software del aplicativo móvil y en las impresiones 3D para el bastón.

Agradezco a la Licenciada María Teresa Arista Rivera, terapeuta ocupacional del Hospital Dos de Mayo, por su apoyo incondicional, haciéndome valiosas sugerencias sobre el tratamiento de rehabilitación de pacientes con problemas de la marcha y la importancia que tendría el bastón diseñado dentro de las terapias.

También quiero agradecer al Ing. William Valladares Vargas y al Sr. Julio Pachamango Mantilla, encargados del Taller de Electrónica de la Sección Electricidad y Electrónica, por su valioso apoyo al prestarme las instalaciones para realizar algunas pruebas con el bastón diseñado.

Finalmente, extiendo mi agradecimiento a todas las personas que, en cualquier ámbito de mi vida y profesión, depositaron su confianza en mí, permitiéndome continuar y culminar este proyecto.



## RESUMEN

A lo largo de la historia, la capacidad de andar ha sido fundamental en la evolución humana, permitiendo la exploración y la adaptación a diversos entornos. Sin embargo, ciertos trastornos de la marcha, derivados de lesiones como caídas, parálisis cerebral, accidentes cerebrovasculares o lesiones musculares, afectan la habilidad de los pacientes para sostener el peso corporal y mantener el equilibrio al caminar o estar de pie. Esto subraya la necesidad urgente de tratamientos de rehabilitación que a menudo implican el uso de bastones, andaderas, trípodes o muletas.

El presente proyecto de tesis propone el diseño de un dispositivo electrónico integrado en un bastón físico. Este dispositivo tiene como objetivo mejorar la rehabilitación motora de la marcha al recopilar datos cruciales, como la fuerza aplicada al bastón, el ángulo de inclinación, la fecha y la hora. Además, permite la transferencia inalámbrica de estos datos a un teléfono móvil o computadora a través de Bluetooth. Este enfoque facilita el análisis por parte del médico o terapeuta ocupacional, proporcionando información valiosa para tomar decisiones informadas sobre la recuperación de la fuerza muscular en las extremidades inferiores, aumentar la estabilidad funcional y mejorar el equilibrio necesario para el proceso de marcha del paciente.

Es importante destacar que este bastón electrónico está diseñado específicamente para pacientes en rehabilitación con la posibilidad de recuperar la capacidad de caminar. Se excluyen a aquellos con daños neurológicos, como ACV, Alzheimer o Parkinson, así como aquellos con enfermedades cardiovasculares descompensadas, discapacidad visual, entre otras condiciones.

En cuanto al desarrollo técnico del bastón electrónico, se emplean dispositivos y tecnologías contemporáneas, como el microcontrolador MK20DX256VLH7 de Freescale integrado en la tarjeta Teensy 3.2, un sensor de aceleración MEMS (ADXL345), un reloj en tiempo real DS3231, un sensor de presión FlexiForce de Tekscan, un Opamp MCP6002 y un módulo Bluetooth HC-06 para la comunicación inalámbrica del microcontrolador y la transferencia de datos almacenados a dispositivos móviles o computadoras.

Este proyecto busca contribuir significativamente al proceso de rehabilitación de pacientes, ofreciendo una herramienta innovadora y tecnológica que permita una monitorización más precisa y personalizada de la marcha, facilitando así una recuperación más efectiva y mejorando la calidad de vida de quienes enfrentan trastornos de la marcha.

La presente tesis se organiza en cuatro capítulos. El capítulo 1, describe básicamente la problemática y los procedimientos clínicos para rehabilitación de marcha. El capítulo 2, reseña procedimientos para rehabilitación de la marcha y tecnología empleada en bastones inteligentes, así como la fisiología y biomecánica de la marcha. En el capítulo 3 se detalla el diseño del sistema de rehabilitación de la marcha basado en bastón electrónico. Finalmente, en el capítulo 4, se presentan las pruebas del bastón en personas voluntarias, del software desarrollado y se analizan los resultados obtenidos.



## ABSTRACT

Throughout history, the ability to walk has been fundamental in human evolution, enabling exploration and adaptation to various environments. However, certain gait disorders resulting from injuries such as falls, cerebral palsy, strokes, or muscle injuries affect patients' ability to bear body weight and maintain balance while walking or standing. This underscores the urgent need for rehabilitation treatments that often involve the use of canes, walkers, tripods, or crutches.

The current thesis project proposes the design of an electronic device integrated into a physical cane. This device aims to enhance motor rehabilitation of gait by collecting crucial data, such as the force applied to the cane, the angle of inclination, date, and time. Additionally, it allows wireless transfer of this data to a mobile phone or computer via Bluetooth. This approach facilitates analysis by the physician or occupational therapist, providing valuable information for making informed decisions regarding the recovery of muscle strength in the lower limbs, increasing functional stability, and improving the balance necessary for the patient's walking process.

It is important to note that this electronic cane is specifically designed for patients in rehabilitation with the potential to regain walking ability. Those with neurological damage, such as stroke, Alzheimer's, or Parkinson's, as well as those with decompensated cardiovascular diseases, visual impairment, among other conditions, are excluded.

Regarding the technical development of the electronic cane, contemporary devices and technologies are employed, such as the Freescale MK20DX256VLH7 microcontroller integrated into the Teensy 3.2 board, a MEMS acceleration sensor (ADXL345), a DS3231 real-time clock, a FlexiForce pressure sensor from Tekscan, an MCP6002 Opamp, and an HC-06 Bluetooth module for wireless communication of the microcontroller and transfer of stored data to mobile devices or computers.

This project aims to significantly contribute to the rehabilitation process for patients by offering an innovative and technological tool that allows more precise and personalized monitoring of gait, facilitating more effective recovery and improving the quality of life for those facing gait disorders.

The thesis is organized into four chapters. Chapter 1 describes the issue and clinical procedures for gait rehabilitation. Chapter 2 reviews procedures for gait rehabilitation, technology used in smart canes, as well as the physiology and biomechanics of walking. Chapter 3 details the design of the gait rehabilitation system based on the electronic cane.

Finally, in Chapter 4, tests of the cane on voluntary participants, the developed software, and the obtained results are presented and analyzed.



# ÍNDICE

1.	CAPITULO I.....	16
1.1.	Problemática de la rehabilitación de la marcha.....	17
1.2.	Estado del arte de bastones para rehabilitación de la marcha.....	21
1.2.1.	Bastón electrónico como apoyo para la marcha.....	21
1.2.2.	Bastones inteligentes para personas con párkinson e invidentes.....	25
1.3.	Justificación para realizar el diseño e impactos.....	27
1.4.	Objetivos generales y específicos .....	29
1.4.1.	Objetivos generales.....	29
1.4.2.	Objetivos específicos.....	29
2.	CAPITULO II.....	30
2.1.	Fisiología y biomecánica de la marcha.....	30
2.1.1.	Ciclo de la marcha .....	31
2.2.	Procedimientos de rehabilitación de la marcha .....	32
2.2.1.	Uso del bastón como apoyo para la marcha .....	32
2.2.2.	Información básica del bastón - uso.....	32
2.2.3.	Objetivos de los procedimientos clínicos de rehabilitación .....	33
2.2.4.	Principios de la rehabilitación .....	34
2.2.5.	Procedimiento clínico general para rehabilitación de la marcha .....	34
2.3.	Dispositivos de asistencia tecnológica: Bastón .....	37
2.4.	Sensores de movimiento y fuerza usados en rehabilitación.....	37
2.4.1.	Acelerómetro MENS.....	39
2.4.2.	Sensores de fuerza .....	40
2.5.	Tecnología inalámbrica .....	42
2.5.1.	Módulo Bluetooth .....	43
2.6.	Microcontroladores ARM.....	49
2.6.1.	Módulo Teensy 3.2 .....	51
2.6.2.	Canal ADC.....	52
2.6.3.	Bus I2C (Inter-Integrated Circuit).....	53
2.6.4.	Bus SPI (Serial Peripheral Interface).....	58
3.	CAPITULO III.....	60
3.1.	Diseño del sistema asociado al bastón electrónico .....	60
3.1.1.	Requisitos principales del diseño propuesto .....	61
3.1.2.	Diagrama de bloques específico del diseño propuesto .....	62

3.1.3.	Selección del Microcontrolador o CPU.....	63
3.2.	Diseño del hardware de adquisición de datos del sensor de fuerza .....	65
3.2.1.	Sensor de fuerza FlexiForce A201 .....	67
3.3.	Diseño de la interfaz de comunicación alámbrica e inalámbrica (I2C, SPI, BT) .....	73
3.3.1.	Interfaz alámbrica con Acelerómetro – I2C.....	73
3.3.2.	Interfaz alámbrica con Reloj en tiempo real DS3231 – I2C.....	77
3.3.3.	Interfaz alámbrica con módulo MicroSD – SPI.....	78
3.3.4.	Interfaz inalámbrica con módulo Bluetooth .....	80
3.4.	Diseño del sistema de alimentación con batería del bastón .....	81
3.5.	Consumo de energía de la tarjeta electrónica del bastón .....	83
3.6.	Acondicionamiento del sensor de fuerza en el bastón de rehabilitación.....	84
3.7.	Diseño del sistema procesador y software .....	86
3.7.1.	Diseño del software del microcontrolador.....	87
3.7.2.	Diseño del software del Aplicativo Móvil.....	91
4.	CAPITULO IV .....	102
4.1.	Prueba del sistema de instrumentación .....	102
4.1.1.	Sensor de fuerza Flexiforce A201.....	102
4.1.2.	Acelerómetro ADXL345 – RTC DS3231.....	111
4.1.3.	Memoria MicroSD .....	114
4.2.	Prueba del sistema de comunicación y software.....	114
4.3.	Pruebas funcionales en el bastón electrónico en modo normal o diario.....	118
4.3.1.	Prueba cuando Peso > 0.....	120
4.3.2.	Prueba cuando Peso >= 0.....	123
4.4.	Pruebas funcionales en el bastón electrónico en modo tiempo real .....	125
4.4.1.	Persona usando el bastón electrónico en tiempo real.....	128
4.5.	Análisis de costos del diseño del bastón inteligente para rehabilitación .....	134
5.	CONCLUSIONES .....	135
5.1.	Limitaciones del Proyecto .....	137
6.	RECOMENDACIONES .....	138
7.	Bibliografía.....	140
8.	ANEXOS .....	144
8.1.	Programa en el microcontrolador ARM con Lenguaje C.....	144
8.2.	Programa del Aplicativo Móvil con Android Studio .....	150
8.2.1.	Archivo: com.jbarriga.bastonint.bo.BluetoothMng.java.....	151
8.2.2.	Archivo: com.jbarriga.bastonint.dao.BastonBD.java.....	158
8.2.3.	Archivo: com.jbarriga.bastonint.domain.DatosPasoVO.java .....	160

8.2.4.	Archivo: com.jbarriga.bastonint.domain.PacienteVO .....	161
8.2.5.	Archivo: com.jbarriga.bastonint.gui.MainActivity .....	162
8.2.6.	Archivo com.jbarriga.bastonint.gui.FrmMenuPrincipal .....	164
8.2.7.	Archivo com.jbarriga.bastonint.gui.FrmMonitorTiempoReal .....	166
8.2.8.	Archivo com.jbarriga.bastonint.gui.MonitorImageView .....	169
8.2.9.	Archivo com.jbarriga.bastonint.gui.FrmHistorialDatos .....	172
8.2.10.	Archivo com.jbarriga.bastonint.gui.MonitorImageView .....	177
8.2.11.	Archivo com.jbarriga.bastonint.gui.FrmNuevoPaciente .....	180
8.2.12.	Archivo com.jbarriga.bastonint.BastonConstant.java .....	182



## INDICE DE FIGURAS

Figura 1-1. Personas con discapacidad según número de limitaciones [1].....	17
Figura 1-2. Personas con discapacidad por el tipo de limitación [1].....	18
Figura 1-3. Discapacidad según tratamientos y/o terapias de rehabilitación [1].....	18
Figura 1-4. Bastón inteligente desarrollado por UMA [2]. .....	19
Figura 1-5. Descripción general del diseño mecánico del bastón UMA [2].....	22
Figura 1-6. The Smart Cane platform [5].....	22
Figura 1-7. Esquema del sistema de emisión (A) – recepción (B) (adaptación propia).....	23
Figura 1-8. Esquema del circuito demodulador (adaptación propia).....	23
Figura 1-9. Esquema del circuito de adaptación (adaptación propia).....	24
Figura 1-10. Bastón inteligente Pauto. (fuente: <a href="https://www.i4life.es/portfolio/pauto/">https://www.i4life.es/portfolio/pauto/</a> ) .....	25
Figura 1-11. Bastón Egara adaptable sobre un bastón blanco <sup>5</sup> .....	26
Figura 1-12. Ejemplo de emisión de ultrasonido del Bastón Egara <sup>5</sup> .....	26
Figura 1-13. Bastón XploR con reconocimiento facial. ....	27
Figura 2-1. Fases de apoyo doble y apoyo simple en el ciclo de la marcha [18].....	30
Figura 2-2. Modelo de bastones: 1 apoyo y 4 apoyos [20]. .....	32
Figura 2-3. Caminando con el bastón sobre pisos planos [20].....	33
Figura 2-4. Ejercicio de fisioterapia.....	35
Figura 2-5. Entrenamiento para usar bastón en escalera.....	36
Figura 2-6. Fortaleciendo músculos y estabilidad. ....	37
Figura 2-7. Diagrama de bloques funcionales y pines del acelerómetro MEMS ADXL345. ....	39
Figura 2-8. Sensor de fuerza piezoresistivo A201 – FlexiForce [29]. ....	41
Figura 2-9. Leading communication achievements for several generations (1G to 5G) [31].....	42
Figura 2-10. Diagrama de bloques de un sistema inalámbrico básico [32]. Adaptación propia. ....	42
Figura 2-11. Clasificación de las tecnologías inalámbricas [34]. ....	43
Figura 2-12. Periféricos conectados por Bluetooth.....	43
Figura 2-13. Topología de una red Bluetooth.....	46
Figura 2-14. Arquitectura del Hardware Bluetooth (Adaptación propia).....	47
Figura 2-15. Espectro de frecuencias en Bluetooth clásico.....	48
Figura 2-16. Espectro de frecuencias en Bluetooth BLE [40] .....	49
Figura 2-17. Clasificación de los procesadores ARM.....	49
Figura 2-18. Arquitectura interna de un ARM Cortex-M4.....	50
Figura 2-19. Módulo ARM Teensy 3.2. ....	51
Figura 2-20. Distribución de pines del Módulo Teensy 3.2. ....	51
Figura 2-21. Diagrama de bloques del ADC [45]. ....	53
Figura 2-22. Ejemplo de bus I2C [46]. ....	53
Figura 2-23. Diagrama de bloques funcional del I2C [45]. ....	55
Figura 2-24. Señales de transmisión en el bus I2C [45].....	56
Figura 2-25. Estándar de comunicación SPI .....	58
Figura 2-26. Red de comunicación SPI con un maestro y 3 esclavos. ....	59
Figura 3-1. Diagrama de bloques general del sistema propuesto del Hardware y Software de App60	
Figura 3-2. Diagrama detallado del diseño propuesto (Fuente: elaboración propia). ....	62
Figura 3-3. Control de Offset del sensor de fuerza (Fuente: elaboración propia).....	65
Figura 3-4. Circuito para obtener el peso en Vout [29] (Fuente: adaptación propia). ....	67
Figura 3-5. Curvas de transferencia entre fuerza, voltaje con Rfeedback a 50KΩ [29]. ....	67
Figura 3-6. Curvas de transferencia entre fuerza, voltaje con Vref a 1V [29].....	68

Figura 3-7. Circuito de adquisición de datos del sensor de fuerza FlexiForce (Fuente: elaboración propia).....	68
Figura 3-8. Voltaje de referencia inicial del sensor de fuerza (Fuente: elaboración propia).....	68
Figura 3-9. Tarjeta principal del bastón con el sensor de fuerza.....	69
Figura 3-10. Fotos de los pesos utilizadas para la obtención de la curva del sensor de fuerza A201. ....	69
Figura 3-11. Combinación de pesas para hallar la curva de transferencia. (Fuente: Elaboración propia).....	70
Figura 3-12. Estructura de aluminio con las pesas. (Fuente: Elaboración propia) .....	70
Figura 3-13. Curva característica del sensor, basado en la tabla 3-4 (Fuente: elaboración propia). 72	
Figura 3-14. Movimientos Pitch y Roll del acelerómetro ADXL345 [26]. .....	76
Figura 3-15. Ejes de sensibilidad de aceleración [26].....	76
Figura 3-16. Respuesta de salida vs. orientación de la gravedad [26]. .....	76
Figura 3-17. Ángulos para la detección de inclinación independiente [26]. .....	77
Figura 3-18. Módulos (a) DS1307, (b) DS3231 y (c) PCF8563.....	77
Figura 3-19. Modelos de módulos que contienen la tarjeta microSD.....	79
Figura 3-20. Modelos de tarjetas SD, Mini SD y Micro SD.....	79
Figura 3-21. Esquemático del circuito de comunicación CPU, ADXL345, DS3231, microSD y BT.....	81
Figura 3-22. Circuito de control de la alimentación al bastón electrónico (Fuente: elaboración propia).....	82
Figura 3-23. Batería 103450 de 3.7 V y 2500 mAh.....	84
Figura 3-24. Software de diseño en 3D. ....	84
Figura 3-25. Imágenes 3D de los soportes de sensor y tarjetas electrónicas (Fuente: elaboración propia).....	85
Figura 3-26. Muestra la forma que se ejerce presión sobre el sensor (Fuente: elaboración propia). ....	85
Figura 3-27. Imágenes impresas en 3D de la figura 3-23 para el bastón de rehabilitación .....	86
Figura 3-28. Circuito principal: CPU, Periféricos y Alimentación con batería (Fuente: elaboración propia).....	86
Figura 3-29. Circuitos impresos del CPU más periféricos y alimentación con batería (Fuente: elaboración propia). ....	87
Figura 3-30. Estrategia de control del software del bastón electrónico para rehabilitación (Fuente: elaboración propia). ....	88
Figura 3-31. Diagrama de flujo de adquisición de datos del ADC (Fuente: elaboración propia). ....	90
Figura 3-32. Arquitectura del software de control del AppMóvil (Fuente: elaboración propia).....	91
Figura 3-33 Pantallas de inicio (a) y Menú Principal (b) (Fuente:elaboración propia). ....	92
Figura 3-34. Pantalla de Paciente Nuevo (Fuente:elaboración propia).....	92
Figura 3-35. Datos almacenados del día 10/04/2024 hasta 14/04/2024 (Fuente: elaboración propia). ....	93
Figura 3-36. Pantalla emergente del calendario (Fuente: elaboración propia).....	95
Figura 3-37. Muestra datos almacenados del día 10.04.2024 (Fuente: elaboración propia). ....	95
Figura 3-38. Pantalla del Monitor en Tiempo Real (Fuente: elaboración propia). ....	97
Figura 3-39. Detalle de los colores del semáforo del peso (Fuente: elaboración propia).....	97
Figura 3-40. Semáforo del peso para distintas fuerzas ejercidas sobre el bastón .....	98
Figura 3-41. Gráfico que muestra las etiquetas del Monitor en Tiempo Real (Fuente: elaboración propia).....	101
Figura 4-1. Características del sensor de fuerza A201 de FlexiForce. (Fuente: <a href="https://www.tekscan.com/sites/default/files/resources/FLX-Datasheet-A201-Rev1.pdf">https://www.tekscan.com/sites/default/files/resources/FLX-Datasheet-A201-Rev1.pdf</a> ) .....	102

Figura 4-2. Pesas para validación de la respuesta del sensor de fuerza en el bastón.....	103
Figura 4-3. Validación de los pesos sobre el bastón (Fuente: elaboración propia).....	104
Figura 4-4 . Imagen del AppMóvil "Serial Bluetooth" cuando está conectado (Fuente: adaptación propia).....	105
Figura 4-5. Comprobación del bastón con un peso de 5.49 Kg (Fuente: elaboración propia). ....	105
Figura 4-6. Comprobación del bastón con un peso de 10.490 Kg (Fuente: elaboración propia)...	107
Figura 4-7. Comprobación del bastón con un peso de 15.490 Kg (Fuente: elaboración propia)...	108
Figura 4-8. Comprobación del bastón con un peso de 19.840 Kg (Fuente: elaboración propia)...	109
Figura 4-9. Gráfico de Fuerza aplicada sobre el bastón vs Error absoluto (Fuente: elaboración propia).....	110
Figura 4-10. Posición del bastón con respecto al ángulo Pitch (Fuente: elaboración propia) .....	111
Figura 4-11. Posición del bastón con respecto al ángulo Roll (Fuente: elaboración propia) .....	111
Figura 4-12. Resultados del Acelerómetro ADXL345 y RTC DS3231 (Fuente: elaboración propia).113	
Figura 4-13. Datos grabados en la memoria MicroSD sobre uso del bastón inteligente .....	114
Figura 4-14. Prueba de comunicación por Bluetooth entre Bastón inteligente y Celular. (Fuente: elaboración propia) .....	114
Figura 4-15. Imagen de Baston2 vinculado al teléfono celular (Fuente: elaboración propia). ....	115
Figura 4-16. Icono del Aplicativo Móvil instalado en el celular (Fuente: elaboración propia). ....	115
Figura 4-17. Pantalla inicial o de enlace. ....	115
Figura 4-18. Pantalla Menú Principal del AppMóvil (Fuente: elaboración propia). ....	116
Figura 4-19. Registro de nuevo paciente con datos completos. ....	116
Figura 4-20. Posibles alternativas de compartir información del bastón.....	117
Figura 4-21. Nombre del archivo transferido en formato .TXT (Fuente: elaboración propia).....	117
Figura 4-22. Pantalla en Excel con los datos precargados. ....	117
Figura 4-23. Datos importados (.TXT) y convertidos en forma de tabla de Excel.....	118
Figura 4-24. Diagrama de bloques de la toma de muestras del bastón en modo normal. ....	118
Figura 4-25. Imagen del analizador lógico (Fuente: <a href="https://hifisac.com/en">https://hifisac.com/en</a> ) .....	119
Figura 4-26. Diagrama de tiempos generados durante el muestreo de datos (Fuente: elaboración propia).....	119
Figura 4-27. Gráfico del Peso, Pitch y Roll con peso mayor que CERO (Fuente: elaboración propia) .....	121
Figura 4-28. Interpretación de las señales generadas por el bastón durante la marcha (Fuente: elaboración propia) .....	122
Figura 4-29. Gráfico del Peso, Pitch y Roll con peso $\geq$ que CERO (Fuente: elaboración propia) ..	124
Figura 4-30. Diagrama de bloques de la toma de muestras del bastón en modo tiempo real.....	125
Figura 4-31. Diagrama de tiempos del proceso en tiempo real (Fuente: elaboración propia) .....	126
Figura 4-32. Gráfico de la tabla 4.11 (Fuente: elaboración propia).....	127
Figura 4-33. Prueba funcional bastón en tiempo real (Fuente: elaboración propia).....	129
Figura 4-34. Gráfico de 156 datos del Peso, °Pitch, °Roll y Hora. (Fuente: elaboración propia) ....	131
Figura 4-35. Gráfico del Peso, °Pitch y °Roll durante 16 seg de la prueba funcional en tiempo real. ....	131
Figura 4-36. Gráfico del Peso vs. Hora durante la prueba funcional en tiempo real. (Fuente: elaboración propia) .....	131
Figura 4-37. Gráfico del °Pitch vs. Hora durante la prueba funcional en tiempo real.....	132
Figura 4-38. Gráfico del °Roll vs. Hora durante la prueba funcional en tiempo real.....	132
Figura 4-39. Contralateral user walking: (a) User walking in the support cane period; and (b) Input signal from force sensor during a 10 s interval in user 8 (range is 0–1023). Maximum peaks (corresponding to cane vertical position) are marked in. ....	133

## INDICE DE TABLAS

Tabla 1-1 Resumen comparativo de 2 bastones similares al propuesto en la tesis. ....	24
Tabla 2-1. Fases del ciclo de la marcha [18] . ....	31
Tabla 2-2. Características de los distintos fabricantes de acelerómetros.....	38
Tabla 2-3 Sensor FlexiForce estándar (modelo A201) [29].....	41
Tabla 2-4. Diferencias entre las versiones de Bluetooth. ....	44
Tabla 2-5. Niveles de potencia y alcance de Bluetooth (Fuente: Bluetooth SIG Group). ....	45
Tabla 2-6. Bandas de frecuencias operativas [41]. ....	48
Tabla 2-7. Bandas de frecuencias BLE operativas [41]. ....	49
Tabla 3-1. Cuadro comparativo de principales características de microcontroladores.....	64
Tabla 3-2. Cuadro comparativo de principales características de sensores de fuerza. ....	66
Tabla 3-3. Valores obtenidos de voltajes por cada medida realizada. ....	71
Tabla 3-4. Valores de peso (lb) vs. voltaje (voltios). ....	71
Tabla 3-5. Cuadro comparativo de principales características de acelerómetros.....	73
Tabla 3-6. Consumo de Corriente vs. Velocidad de datos [26].....	74
Tabla 3-7. Consumo de Corriente vs. Velocidad de datos en bajo consumo [26]. ....	74
Tabla 3-8. Cuadro comparativo de principales características de relojes en tiempo real. ....	78
Tabla 3-9. Cuadro comparativo de principales características de tarjetas microSD.....	79
Tabla 3-10. Cuadro comparativo de principales características de módulos Bluetooth. ....	80
Tabla 3-11. Consumo de corriente por el circuito del bastón. ....	83
Tabla 4-1 Valores registrados por la balanza y teléfono celular para un peso de prueba 5.49 Kg. ....	106
Tabla 4-2. Valores registrados por la balanza y bastón para un peso de prueba 10.490 Kg. ....	107
Tabla 4-3. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 10.490 Kg.....	107
Tabla 4-4. Valores registrados por la balanza y teléfono celular para un peso de prueba 15.49 Kg. ....	108
Tabla 4-5. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 15.490 Kg.....	108
Tabla 4-6. Valores registrados por la balanza y teléfono celular para un peso de prueba 19.840 Kg. ....	109
Tabla 4-7. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 19.840 Kg.....	109
Tabla 4-8 Resultados del sensor de fuerza del bastón para diferentes pesos.....	110
Tabla 4-9. Error de medición del ángulo de inclinación Pitch. ....	113
Tabla 4-10. Datos importados del celular cuando el peso ejercido sobre el bastón es mayor que cero.....	120
Tabla 4-11 Datos importados del celular cuando el peso ejercido sobre el bastón es $\geq$ cero. ....	123
Tabla 4-12. Valores capturados en tiempo real.....	127
Tabla 4-13. Datos captados de la aplicación en tiempo real durante 49 seg (Fuente: elaboración propia) .....	130
Tabla 4-14. Datos captados de la aplicación en tiempo real durante 16 seg (Fuente: elaboración propia) .....	131
Tabla 4-15. Costo total del proyecto Bastón inteligente. ....	134

# CAPITULO I

## INTRODUCCIÓN

Según la OMS<sup>1</sup>, “se calcula que hay más de mil millones de personas en el mundo con algún problema de discapacidad que necesitan tecnología de apoyo, como productos físicos (sillas de ruedas, rampas portátiles, pasamanos, prótesis o lentes), soporte lógico y aplicaciones digitales, pero no tienen acceso a ella. Estas tecnologías de apoyo cambian la vida, mejoran la salud y el bienestar de sus usuarios en la parte socioeconómica, puesto que reducen el gasto de los servicios sanitarios y sociales (por ejemplo, los costos de las sucesivas hospitalizaciones, subsidios y prestaciones sociales) y mejoran su productividad, lo cual, indirectamente, estimula el crecimiento económico”.

Este bastón a diseñarse es un sistema de apoyo con tecnología electrónica que va orientado principalmente a pacientes en rehabilitación con posibilidades de volver a caminar, por ello, los datos que serán registrados en la memoria del bastón, será de vital importancia para el médico tratante o el terapeuta ocupacional para saber si el paciente usó adecuadamente el bastón. En una entrevista a la especialista en terapia ocupacional Lic. María Teresa Arista del Hospital Dos de Mayo, se obtuvo la siguiente apreciación: “Muchas veces, los adultos mayores y jóvenes no cumplen las indicaciones de cómo deben usar el bastón durante su tratamiento y cuando regresan a su próxima cita, el médico observa que la mejoría es muy poca (o casi nula) y al no tener una herramienta o sistema que le indique si el paciente cumplió con todas las indicaciones, lo más probable es que le vuelva a indicar que continúe un tiempo más con el mismo tratamiento”.

Este bastón resolverá el problema mencionado en el párrafo anterior, de la no realimentación entre paciente y terapeuta, porque va monitorear en tiempo real y a su vez almacenar en una memoria microSD los parámetros necesarios para poder evaluar si un paciente en rehabilitación cumplió con su tratamiento, como, por ejemplo, medir la fuerza aplicada sobre el bastón, la forma que utilizó el bastón en cada pisada (ángulos de inclinación del bastón con respecto a la horizontal y vertical) y un reloj en tiempo real para registrar la fecha y hora de cada pisada. Adicionalmente, un familiar o terapeuta podrá realizar un monitoreo al paciente en tiempo real a través de un teléfono móvil smart con S.O. Android, observando la fuerza aplicada y el ángulo de apoyo en cada pisada, de tal manera que pueda instruir al paciente a usar mejor el bastón.

---

<sup>1</sup> <https://www.who.int/es/news/item/16-05-2022-almost-one-billion-children-and-adults-with-disabilities-and-older-persons-in-need-of-assistive-technology-denied-access--according-to-new-report>.

### 1.1. Problemática de la rehabilitación de la marcha

De acuerdo con las estadísticas realizadas por el INEI (Instituto Nacional de Estadística e Informática) en el año 2012, se puede observar que el problema para moverse o marcha representa casi el 60% de las personas con discapacidad, y de este porcentaje casi el 50% corresponden a personas de avanzada edad o adultos mayores [1], siendo las posibles causas una parálisis cerebral, accidentes cerebrovasculares, caídas o lesiones musculares. Los jóvenes no están exentos de sufrir accidentes, que conlleven a la necesidad de recibir un tratamiento de rehabilitación para poder caminar, pero en menor proporción.

La Resolución WHA74.8 de la Asamblea Mundial de la Salud<sup>2</sup> insta a los “Estados Miembros a que garanticen que las personas con discapacidad reciban servicios de salud eficaces como parte de la cobertura sanitaria universal; igual protección ante las emergencias; y acceso igualitario a las intervenciones de salud pública intersectoriales”. De manera similar, se tiene la Ley N° 27050<sup>3</sup> “Ley General de la Persona con Discapacidad”: “Establece el régimen legal de protección, de atención de salud, trabajo, educación, rehabilitación, seguridad social y prevención, para que la persona con discapacidad alcance su desarrollo e integración social, económica y cultural. Establece sus derechos, así como el papel del Estado y la familia”.

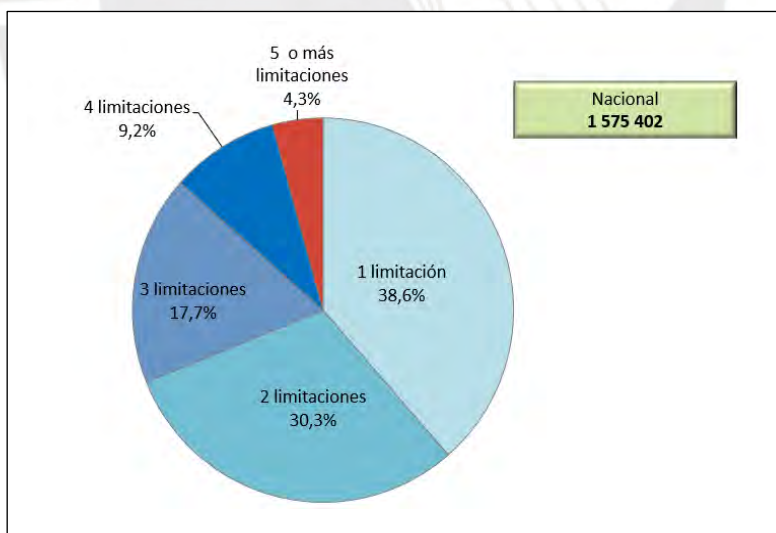


Figura 1-1. Personas con discapacidad según número de limitaciones [1].

<sup>2</sup><https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>

<sup>3</sup> <https://bvs.minsa.gob.pe/local/MINSA/6781.pdf>

La figura 1-1, nos muestra los porcentajes de las personas con discapacidad que tienen una o más limitaciones que las afecta y donde el estado Peruano tiene que intervenir para que la rehabilitación sea oportuna y efectiva.

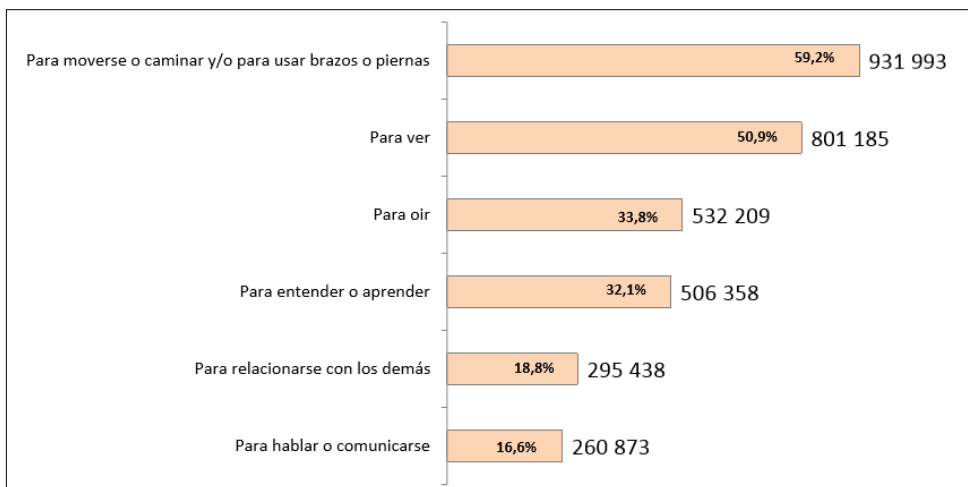


Figura 1-2. Personas con discapacidad por el tipo de limitación [1].

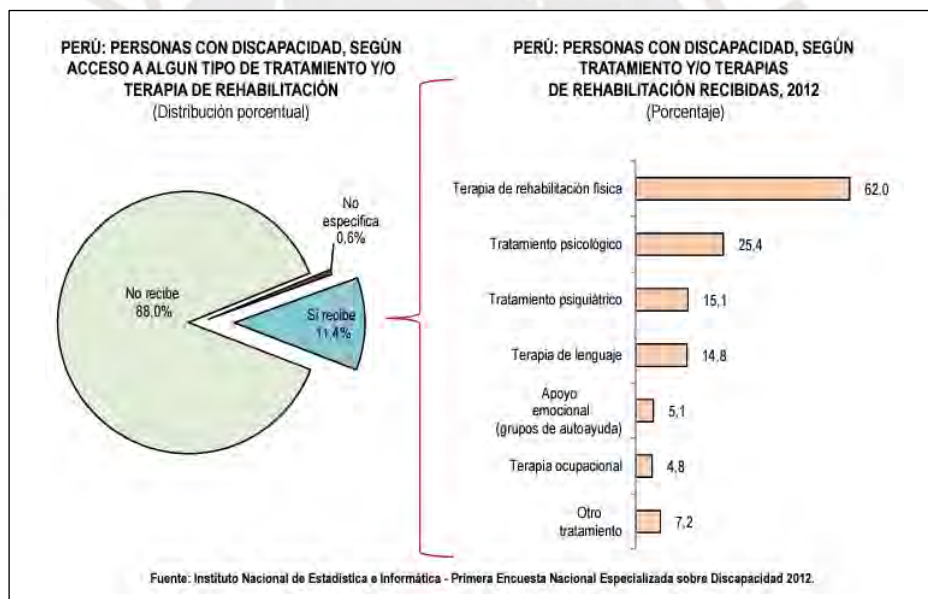


Figura 1-3. Discapacidad según tratamientos y/o terapias de rehabilitación [1].

Según la Figura 1-2, se muestra que el 59.2% de la población con discapacidad tiene dificultades para moverse o caminar, y/o para usar brazos o piernas. En la parte izquierda de la Figura 1-3, se observa que el 88,0% no recibe tratamiento y/o terapia de rehabilitación, mientras que el 11.4% si está recibiendo algún tipo de rehabilitación. En la parte derecha de la Figura 1-3, se muestra que el 62% de ese 11.4% recibe terapia de rehabilitación física. Con estos resultados estadísticos, se puede decir que el Estado no

está cumpliendo con la Ley N° 27050 al tener un alto porcentaje (88%) que no están recibiendo algún tratamiento.

Según la OMS<sup>4</sup>, “La rehabilitación es una inversión rentable en lo que atañe al costo, tanto para quien la precisa como para la sociedad en su conjunto. Puede ayudar a evitar hospitalizaciones costosas, reducir la duración de estas y evitar los reingresos. Dado que hace posible que quienes la reciben puedan empezar a trabajar, volver al empleo o llevar una vida independiente en su hogar, la rehabilitación minimiza la necesidad de recibir apoyo financiero y reduce la dependencia de cuidadores.”

Este proyecto de tesis se enfoca en pacientes en rehabilitación que utilizan bastones con el objetivo de recuperar la capacidad de caminar. Durante la investigación en bases de datos especializadas como PubMed, IEEE Xplore, ScienceDirect, Journals y otras, se identificó que la mayoría de las innovaciones en bastones inteligentes se aplican en el área de discapacidad visual. Estas innovaciones han mejorado el tradicional bastón blanco o bastón de Hoover al incorporar tecnologías como sensores de proximidad, cámaras para reconocimiento facial (que permiten identificar personas, como familiares), GPS, WiFi, entre otros.

Adicionalmente, en la base de datos PubMed se encontró información sobre un bastón diseñado para la rehabilitación de la marcha, desarrollado por la Universidad de Málaga, titulado '*Weight-Bearing Estimation for Cane Users by Using Onboard Sensors*'. Este diseño, que emplea sensores de fuerza, está disponible bajo licencia abierta y representa una valiosa referencia para el presente proyecto.

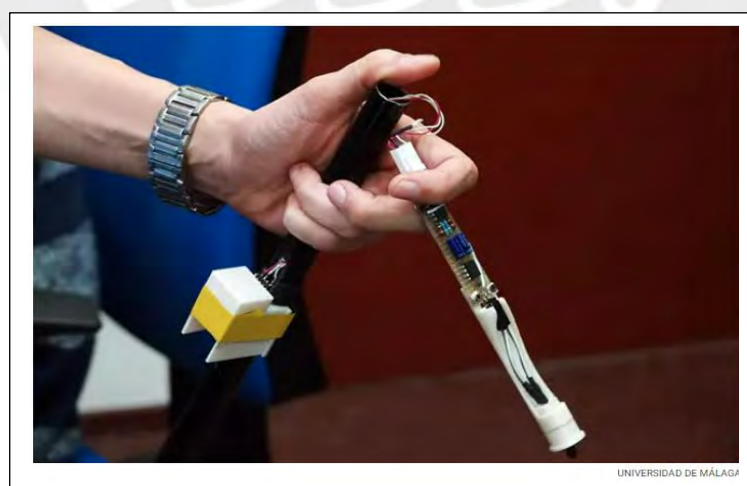


Figura 1-4. Bastón inteligente desarrollado por UMA [2].

<sup>4</sup> <https://www.who.int/es/news-room/fact-sheets/detail/rehabilitation>

*“Este bastón monitoriza el apoyo que la persona realiza mientras camina, facilitando información personalizada sobre su progresión, a partir de la integración de dos sensores de presión colocados a distinta profundidad de la punta del bastón comercial, de forma que no afecten a su ergonomía. Así mismo, para simplificar su uso, dispone de un cargador inalámbrico y los datos se extraen mediante Bluetooth a través de cualquier móvil” [2].*

Este proyecto de tesis desarrollado innova o mejora la ayuda tecnológica existente ([2]), que se ofrece a las personas en proceso de rehabilitación de la marcha, adicionando un reloj en tiempo real y un acelerómetro. Una de las características que se considera más importante del proyecto es que puede registrar en la memoria SD los datos del peso corporal aplicado al bastón, la fecha y los movimientos del bastón cada vez que es utilizado. Además, se ha implementado una plataforma informática para que, en el teléfono móvil, se pueda visualizar en un gráfico de barras la evolución histórica del paciente con respecto al peso corporal ejercido sobre el bastón, y también monitorear en tiempo real los movimientos del bastón al momento de desplazarse. Es decir, los ángulos de inclinación del bastón con respecto a la persona. Esto permitirá orientarlo en el mejor uso del bastón y que su recuperación sea más efectiva.

El bastón inteligente diseñado es no invasivo y, al estar orientado al proceso de rehabilitación de la marcha en pacientes que podrán caminar, se presentan los siguientes escenarios:

1°: Si el tratamiento dado es en el servicio de terapia ocupacional, tratamiento estándar, 3 v/s, por n días y a la evaluación no ha progresado, el protocolo indica que debe modificarse el tratamiento.

2°: Si se le da indicaciones para que lo realice en casa por n días y a la evaluación no ha progresado, se debe evaluar si el paciente ha seguido estrictamente las indicaciones. Si no lo ha seguido continuar con el tratamiento hasta que haga lo que se le indique; Si lo ha seguido, entonces hay que modificar el tratamiento.

3°: Si después del nuevo tratamiento y a la evaluación, no se recupera, entonces tendría una secuela y por consiguiente una discapacidad motora.

## 1.2. Estado del arte de bastones para rehabilitación de la marcha

Para desarrollar este capítulo, se realizó una búsqueda exhaustiva de información en bases de datos especializadas como PubMed, IEEE Xplore, ScienceDirect, entre otras y Journals, enfocada en tecnología electrónica aplicada a bastones para rehabilitación de la marcha. La búsqueda se centró en dispositivos equipados con sensores de fuerza, acelerómetros y conectividad. El objetivo principal fue identificar diferencias y avances tecnológicos en comparación con los bastones convencionales, especialmente aquellos que presentan similitudes con el diseño propuesto en esta tesis.

### 1.2.1. Bastón electrónico como apoyo para la marcha

Existen diversos diseños con tecnología electrónica aplicados a distintas discapacidades, como exoesqueletos robóticos, sillas de ruedas inteligentes y bastones inteligentes, entre otros. Estos dispositivos se utilizan para mejorar la calidad de vida de personas con discapacidades permanentes o problemas de marcha temporales debido a accidentes. A continuación, se describen algunos bastones inteligentes encontrados en las bases de datos mencionadas, los cuales servirán como referencia para el presente proyecto.

#### 1.2.1.1. Bastón monitoriza rehabilitación de los pacientes

"*Weight-Bearing Estimation for Cane Users by Using Onboard Sensors*" es un bastón mecanizado desarrollado por el Grupo de Ingeniería de Sistemas Integrados de la Universidad de Málaga (UMA) [2] [3]. Este dispositivo permite medir el proceso de rehabilitación de un paciente y está diseñado para ser accesible, de bajo costo y adaptable a las necesidades del usuario. Además, los investigadores han puesto a disposición en internet, bajo licencia abierta, los diseños, algoritmos y esquemas electrónicos de sus componentes, facilitando su uso como referencia para futuros desarrollos [4].

Este módulo incluye tres áreas: área de sensores, área de microcontrolador y área de cargador (Figura 1-5). Los sensores se han incrustado dentro del eje, lo más cerca posible de la punta a dos profundidades diferentes: 0.2 mm y 1 mm a través de una pieza de plástico impresa en 3D especialmente diseñada (Figura 1-5b). Por lo tanto, los diseños de la punta y la empuñadura no se modifican, y conservan sus propiedades ergonómicas originales. Colocar los componentes electrónicos dentro del eje es mucho más difícil y dificulta el acceso, por lo que están conectados externamente [5].

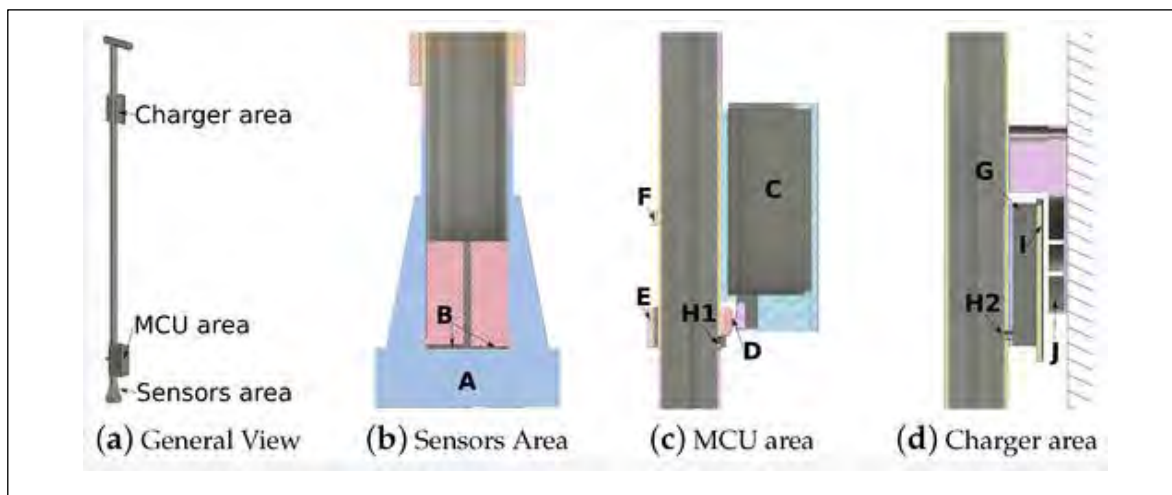


Figura 1-5. Descripción general del diseño mecánico del bastón UMA [2].

(A) punta de goma; (B) ubicaciones de sensores de fuerza; (C) caja de microcontrolador (MCU); (D) conectores entre sensores de carga y MCU; (E) anillo conector; (F) abraçadera para abraçadera para caja MCU; (G) batería y caja del cargador; (H1) orificio a los cables desde el cargador y las áreas de sensores a la caja MCU; (H2) orificio a los cables desde la caja MCU al área del cargador; (I) ubicación del receptor de carga; y (J) ubicación del transmisor de carga.

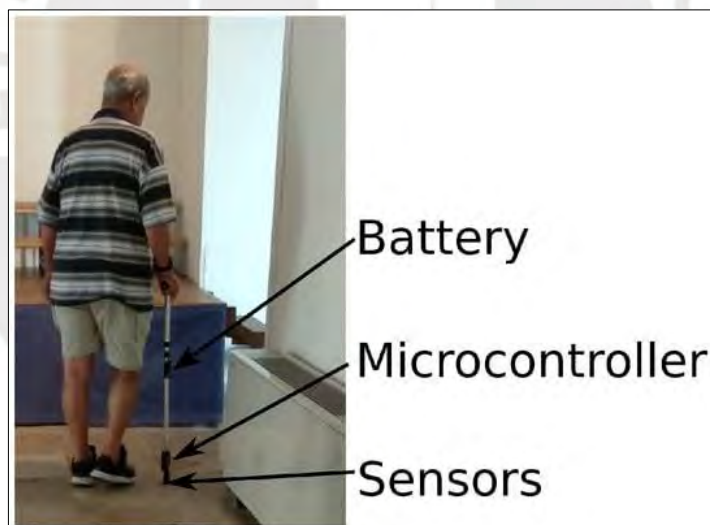


Figura 1-6. The Smart Cane platform [5].

### 1.2.1.2. Análisis de la marcha asistida con bastones mediante un sistema telemétrico

“Analysis of assisted walking with canes using a telemetric system”. En [6], se describe un nuevo sistema de telemetría destinado al análisis de la marcha asistida durante su rehabilitación. Este dispositivo consta de tres receptores-emisores colocados en las dos suelas y en los dos bastones que asisten al paciente durante la deambulaci3n. Para

garantizar un funcionamiento adecuado, se diseñó un circuito ubicado entre el emisor y la antena con el objetivo de eliminar las interferencias producidas por la emisión simultánea de diferentes señales en una misma área.

En la rehabilitación de la marcha asistida mediante el uso de bastones, es fundamental conocer las fuerzas de apoyo tanto del pie como del bastón durante la deambulaci3n. La telemetría se presenta como una técnica ventajosa debido a su estética y ligereza [7].

El sistema telemétrico desarrollado se compone de tres partes diferenciadas: Emisi3n, Recepci3n y La adquisici3n y el tratamiento de las se1ales. Los receptores de fuerza montados en los bastones consisten en un registrador de tensi3n dispuesto en un puente de Wheatstone. Los sensores utilizados, modelo EA-13-062AQ 350S, est1n térmicamente compensados y se colocan en el soporte de tal manera que solo puedan captar el componente vertical de las fuerzas ejercidas. La precisi3n del receptor tiene un rango de 1 N, para fuerzas que oscilan entre 0 y 100 N. Sin embargo, la se1al emitida por el sensor tiene una amplitud débil, no superior a algunos microvoltios, lo que hace necesario amplificarla mediante un amplificador diferencial.

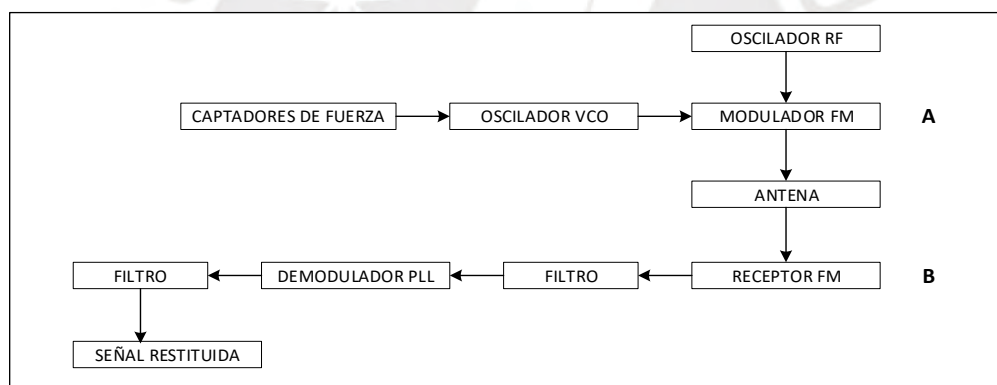


Figura 1-7. Esquema del sistema de emisi3n (A) - recepci3n (B) (adaptaci3n propia).

En la Figura 1-7(A) se representa el esquema del circuito de emisi3n asociado a los captadores de fuerza. En la Figura 1-7(B) se muestra los diferentes pasos por los que la se1al se ve sometida a tratamiento hasta su reconstrucci3n final.

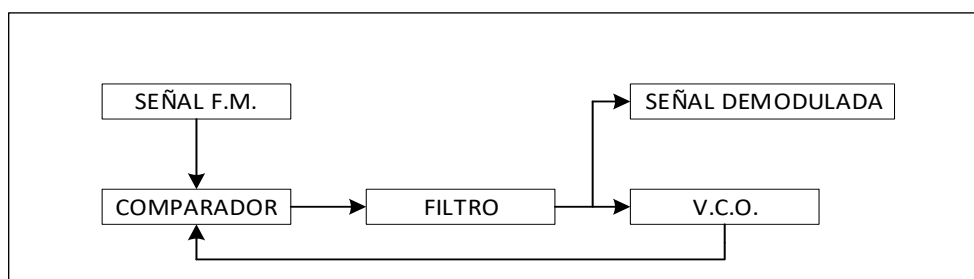


Figura 1-8. Esquema del circuito demodulador (adaptaci3n propia).

Se puede observar en primer lugar, que un filtro pasa bandas de frecuencia, permite el paso de las señales que proceden de los receptores. En segundo lugar, un demodulador transforma de nuevo la señal de los sensores que traducen las fuerzas de apoyo. Este demodulador tiene como componente principal un bucle de cierre en fase (PLL), que se esquematiza en la Figura 1-8, con el que sólo se libera una señal sincrónica de la misma frecuencia y fase que la señal recibida o aplicada a su entrada. Finalmente, un filtro reduce las interferencias.

El objetivo de disminuir la longitud de la antena es para eliminar las interferencias entre las diferentes señales emitidas y para transmitir al máximo la potencia producida. El circuito que permitió reducir el tamaño de la antena se representa esquemáticamente en la Figura 1-9. Finalmente, para la recepción de las señales emitidas se ha utilizado un receptor habitual de telecomunicación y no existe información si este modelo de bastón se está comercializando.

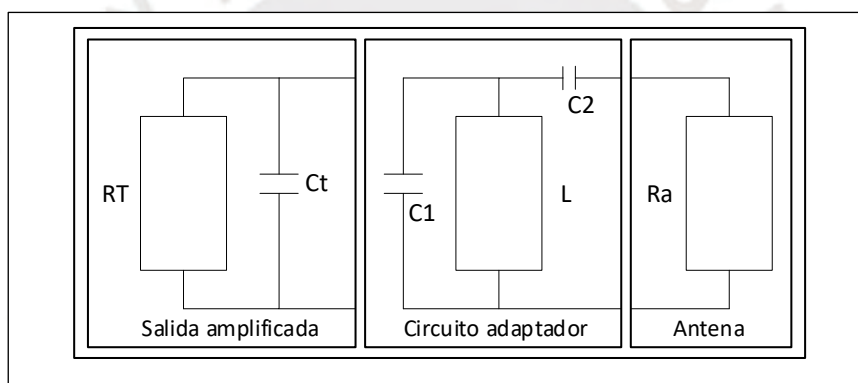


Figura 1-9. Esquema del circuito de adaptación (adaptación propia).

A continuación, se muestra un cuadro resumen comparativo de estos dos bastones con el propuesto en la tesis.

Tabla 1-1 Resumen comparativo de 2 bastones similares al propuesto en la tesis.

Características	Bastón UMA [3]	Bastón Telemetría [6]	Bastón propuesto
Conectividad	Bluetooth	Radio frecuencia	Bluetooth
Sensor de fuerza	FSR402 intetrlink (0 – 100N)	EA-13-062AQ-350 (0 – 100N)	FlexiForce A201 (0 – 111N)
Rango de peso (Kg)	0 – 10.2 Kg	0 Kg – 10.2 Kg	0 – 11.31 Kg
Soporte peso corporal (10%)	102 Kg	102 Kg	113 Kg
Acelerómetro	NO	NO	SI

## 1.2.2. Bastones inteligentes para personas con párkinson e invidentes

A continuación, se describe tres bastones con tecnología electrónica aplicado al tradicional bastón.

### 1.2.2.1. Bastón inteligente Pauto

Este bastón Pauto, se comercializa desde el año 2020, y tiene como objetivo principal recopilar información de quien lo utiliza para detectar síntomas asociados a enfermedades como párkinson o alzhéimer y alertar al cuidador o familiar. Incluye un GPS, acelerómetro, giroscopio, conexión bluetooth y una tarjeta SIM.



Figura 1-10. Bastón inteligente Pauto. (fuente: <https://www.i4life.es/portfolio/pauto/>)

"Bajo la apariencia de un bastón normal tenemos un recurso de captación de datos relativos a síntomas, localización del usuario, detección de caídas y un botón de emergencia. Lo que conseguimos es que esa información vaya a una base de datos accesible al cuidador y cuando haya un evento relevante el sistema lance un mensaje SMS al móvil de los cuidadores, por lo cual el paciente tiene más seguridad al salir a la calle", explica Marián García, responsable de la empresa social asturiana i4life<sup>5</sup>.

Una de las principales aplicaciones de Pauto es prevenir la **congelación de la marcha**, un problema que tienen los pacientes de párkinson, inclusive los más jóvenes pueden sufrir la pérdida temporal de la memoria procedimental que les permite caminar. "De repente se olvidan de caminar, sus pies se quedan pegados al suelo y les es imposible reanudar la marcha". "En este caso, **el bastón produce una línea en el suelo con un láser y da una vibración en la empuñadura, lo que ayuda al cuerpo a volver a caminar**" asegura el diseñador de este producto.

<sup>5</sup> <https://www.i4life.es/portfolio/pauto/>

### 1.2.2.2. Bastón inteligente Egara

La Cátedra de Investigación en Retinosos Pigmentacia Bidons Egara y el Grupo de Neuroingeniería Biomédica de la Universidad Miguel Hernández de Elche (Alicante), junto a otros departamentos de la misma institución, y a la empresa tecnológica Instead Technologies, ha desarrollado en noviembre del 2020 un revolucionario dispositivo tecnológico llamado "Bastón Egara". Se trata de **un mango de asistencia inteligente que se adapta a los bastones guías blancos que utilizan las personas ciegas** y que tiene como objetivo evitar colisiones dolorosas con obstáculos situados por encima de la cintura.

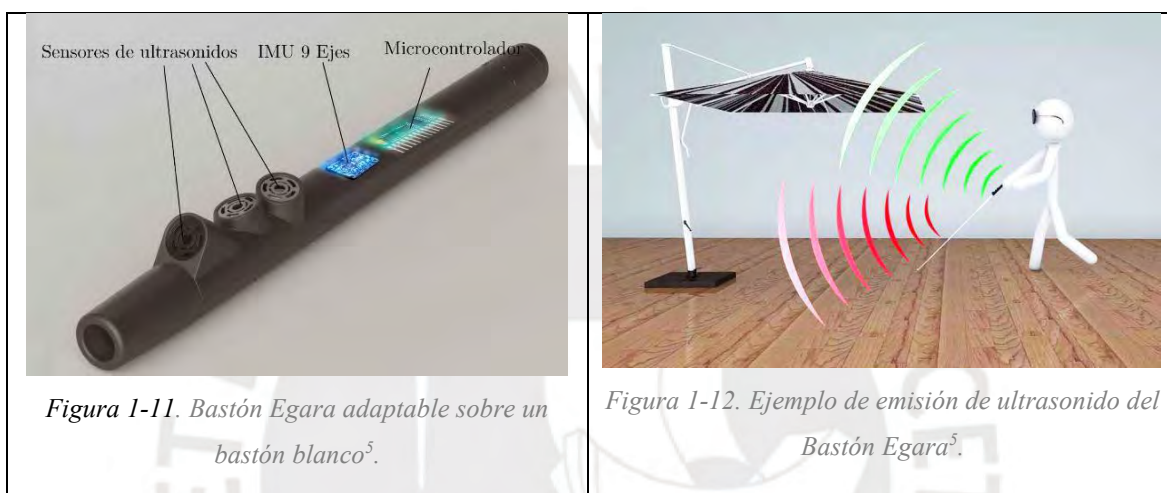


Figura 1-11. Bastón Egara adaptable sobre un bastón blanco<sup>5</sup>.

Figura 1-12. Ejemplo de emisión de ultrasonido del Bastón Egara<sup>5</sup>.

El Bastón Egara permite al usuario detectar aquellos obstáculos aéreos que mediante el rastreo habitual no podría detectar. De este modo el usuario podrá caminar con mayor tranquilidad, inclusive se puede adquirir un mango (Figura 1-11) que se puede incorporar a cualquier bastón blanco tradicionales que ya lo esté usando.

"El bastón solo avisa de aquellos obstáculos que realmente te van a golpear en el pecho, los brazos o la cabeza. Del resto de obstáculos no avisa, porque entendemos que esos se detectan con el propio bastón o van a pasar por un lado sin suponer un peligro. Si la pulsera vibra, entiendes que hay un obstáculo frente a ti, por lo que tan solo tienes que estirar la mano, detectar dónde se encuentra y decidir por dónde esquivarlo".<sup>6</sup>

### 1.2.2.3. Bastón inteligente XPLOR Smartcane que identifica rostros

Un grupo de estudiantes de la Universidad de Birmingham, Reino Unido (año 2015), desarrollaron un bastón inteligente para personas con discapacidad visual que es capaz de identificar los **rostros de** quienes se acerquen a él gracias a una cámara digital

<sup>6</sup> <https://bidonsegara.com/baston-egara/>

y está basado en la tecnología que usan los smartphones para reconocer caras. El 'XploR' puede reconocer rostros familiares a una distancia de hasta 10 metros, vibrando cuando detecta una persona reconocible de un banco de imágenes almacenadas en una tarjeta de memoria SD interna e incluso puede guiar al usuario en base a indicaciones por voz hacia dicha persona gracias a un pequeño auricular que se conecta mediante Bluetooth.

El equipo de creadores presentó en su oportunidad el XploR como toda una revolución a médicos y científicos de Luxemburgo y Francia, pero a la fecha, no hay información si llegó a comercializarse.

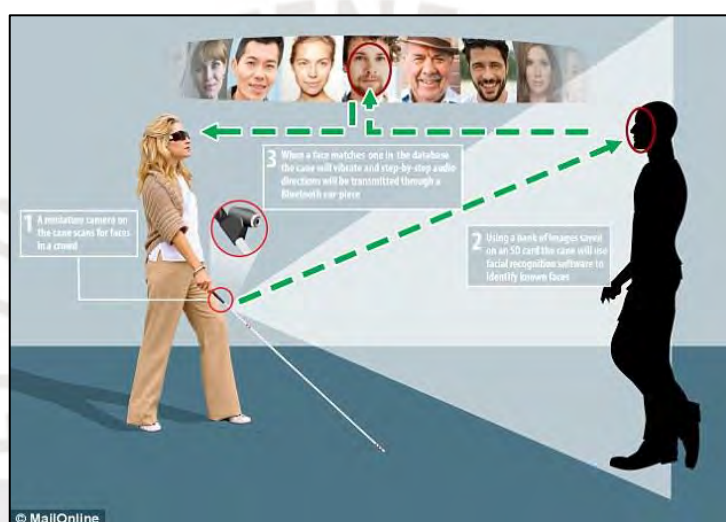


Figura 1-13. Bastón XploR con reconocimiento facial<sup>7</sup>.

### 1.3. Justificación para realizar el diseño e impactos

Según los resultados mostrados por el INEI en las Figura 1-2 y en la Figura 1-3, se observa que hay un bajo porcentaje de pacientes (11.4%) que reciben algún tipo de tratamiento y/o terapia de rehabilitación. Esto representa, indudablemente, un problema social que el Estado no está atendiendo adecuadamente. Las razones pueden ser los altos costos asociados a los recursos humanos capacitados, infraestructura, tratamientos y medicinas. Además, es posible que este grupo de personas viva en zonas alejadas y no cuente con los recursos económicos suficientes para trasladarse hasta los centros de salud para recibir tratamiento de rehabilitación, sumado a que las citas son programadas con intervalos de tiempo largos.

<sup>7</sup> <https://plataformasdt.cl/en/tratando-de-que-los-ciegos-vean-el-caso-del-baston-xplor/>

El sistema propuesto en esta tesis, está orientado a resolver el problema de la nula retroalimentación que existe en la actualidad sobre el tratamiento de rehabilitación con bastón simple en pacientes con problemas de caminar. Es decir, el terapeuta ocupacional o médico tratante da las instrucciones al paciente sobre cómo debe usar el bastón correctamente. Luego, el paciente se retira y regresa para su evaluación después de un determinado número de días. Al ser evaluado, puede ser que se observe una mejoría, si el paciente cumplió con las indicaciones; sin embargo, esto no ocurre con frecuencia, a pesar de que el paciente afirme haber seguido las instrucciones (ver introducción).

El bastón desarrollado con tecnología electrónica contribuirá a establecer una comunicación efectiva entre el paciente, el bastón y el médico, lo que hará que el tratamiento sea más efectivo. Gracias a la información registrada en la memoria del bastón, el paciente podrá analizar si lo está utilizando correctamente y realizar las correcciones necesarias para asegurar el éxito de su rehabilitación en el tiempo indicado.

Este diseño se justifica ante la escasez de recursos económicos disponibles en nuestros sistemas de salud, ya que permitirá reducir los tiempos de tratamiento de rehabilitación de la marcha e insertar a los pacientes al mundo laboral, en línea con las exigencias de la Resolución WHA74.8 de la Asamblea Mundial de la Salud y la Ley N° 27050 del estado peruano.

El diseño propuesto tendrá un impacto significativo en dos aspectos principales. En primer lugar, permitirá que el paciente se convierta en el protagonista de su rehabilitación, ya que podrá monitorear su progreso y verificar si está siguiendo el tratamiento correctamente, sin necesidad de acudir constantemente al hospital. En segundo lugar, contribuirá a descongestionar los centros de rehabilitación, reservando la atención hospitalaria para los casos más graves que requieran la intervención directa del médico.

Es importante mencionar, que existe un estudio de validación de la herramienta *Chamorro Assisted Gait Scale (CHAGS)* [8] [9], la primera escala de valoración observacional de la marcha asistida con bastones de antebrazo, cuyos resultados obtenidos han indicado que la escala CHAGS diseñada, constituye un método válido y fiable para analizar la marcha asistida con bastones de antebrazo de pacientes que padecen esguince de tobillo; y ha **permitido corroborar que el 10% del peso corporal es la descarga máxima** que un usuario puede ejercer con un solo bastón [10]. También Melis EH. et al. [11] en su estudio sobre el bastón de mano, recalca que este bastón se debe utilizar como máximo hasta un 15% de soporte vertical sobre él, porque se vuelve inestable cuando se carga a partir del 20-25% del peso corporal. En consecuencia, podríamos aceptar inicialmente como válido un rango de peso corporal que el paciente puede ejercer sobre el bastón entre 10% al 15%,

hasta que se compruebe en campo por los profesionales de la salud en el área de rehabilitación de la marcha.

#### **1.4. Objetivos generales y específicos**

##### **1.4.1. Objetivos generales**

Diseñar un bastón electrónico e implementar una plataforma informática para dispositivos móviles con el propósito de mejorar el proceso de rehabilitación motora de la marcha, reducir el tiempo de recuperación del paciente, y proporcionar al médico tratante datos precisos sobre el uso del dispositivo, facilitando una retroalimentación efectiva para verificar el cumplimiento de las indicaciones.

##### **1.4.2. Objetivos específicos**

Los objetivos específicos del diseño propuesto son los siguientes:

1. Determinar las necesidades correspondientes a los procedimientos clínicos existentes en rehabilitación de la marcha, para establecer cuáles podrán ser atendidas por el prototipo.
2. Diseñar el hardware y software del bastón electrónico para generar un prototipo funcional.
3. Elaborar un circuito capaz de registrar el peso corporal ejercido sobre el bastón, desde 0 kg hasta 20 kg, y los ángulos de inclinación del bastón: de 60° a 110° en dirección de avance (Pitch) y de -30° a 30° hacia los costados del avance (Roll).
4. Diseñar una plataforma informática basada en sistema operativo Android para el teléfono móvil, amigable para el usuario y que permita reunir los datos necesarios para anexarlos a su historia clínica.
5. Implementar el prototipo del bastón electrónico.
6. Evaluar la funcionalidad del prototipo implementado mediante pruebas funcionales con pacientes hipotéticos en proceso de rehabilitación. Un estudio clínico será realizado en otro proyecto fuera del alcance de esta tesis.

## CAPITULO II

### FUNDAMENTOS DE REHABILITACION DE LA MARCHA Y TECNOLOGIA DE BASTONES INTELIGENTES

#### 2.1. Fisiología y biomecánica de la marcha

La fisiología de la marcha humana implica una compleja interacción de sistemas musculoesqueléticos y neuromotores que trabajan de manera coordinada para permitir el desplazamiento del cuerpo de un lugar a otro [12]. La efectiva interacción de los sistemas que intervienen en la marcha posibilita el desplazamiento del individuo, cuyo peso es cargado de manera alternativa por ambos miembros inferiores [13] [14].

La biomecánica de la marcha se describe como un conjunto de movimientos alternantes y rítmicos de las extremidades inferiores y del tronco, que permite el desplazamiento del cuerpo a través de la acción coordinada de cada uno de los componentes que conforman el sistema locomotor humano [15]. Éste puede verse afectado por un conjunto grande de traumatismos y patologías que modifican notablemente la dinámica natural de la marcha [16]. Cuando la masa corporal es soportada sobre el miembro de apoyo, la extremidad inferior contralateral se balancea preparándose para el siguiente apoyo [17].

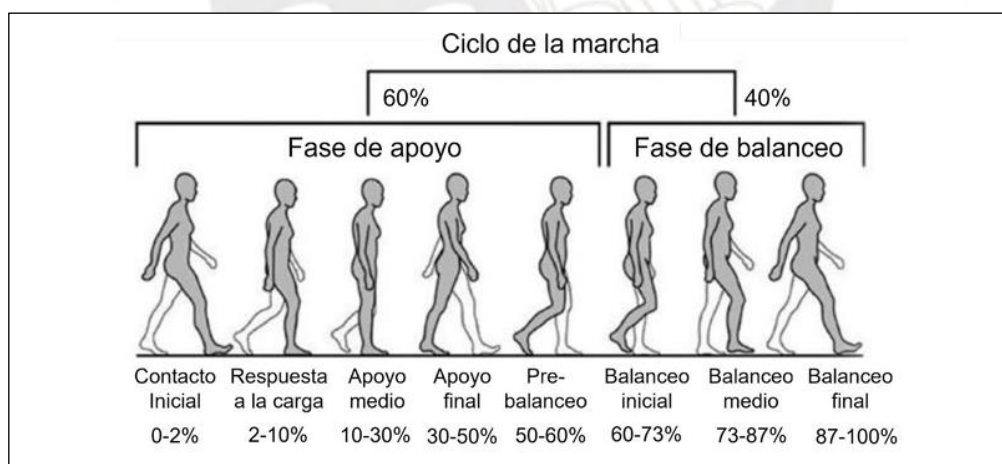


Figura 2-1. Fases de apoyo doble y apoyo simple en el ciclo de la marcha [18].

Los porcentajes correspondientes a estas dos fases de apoyo y oscilación dependen de la velocidad [19] [18].

### 2.1.1. Ciclo de la marcha

Según [18], el ciclo de marcha comienza cuando un pie contacta con el suelo y termina cuando se produce el siguiente contacto del mismo pie. Consta de dos fases:

- **Fase de Apoyo:** período durante el cual el pie está en contacto con el suelo. A una velocidad de la marcha normal, esta fase constituye aproximadamente el 60% del andar.
- **Fase de balanceo:** período durante el cual el pie se separa del suelo y la pierna se mueve hacia adelante. Supone el 40% del ciclo de la marcha.

La **fase de apoyo** está dividida en cinco intervalos:

- Contacto inicial: Instante en que el talón toca el suelo.
- Respuesta de carga: Contacto de la parte anterior del pie con el suelo.
- Apoyo medio: Momento en que el trocánter mayor se encuentra alineado verticalmente con el centro del pie (visto desde el plano sagital).
- Apoyo final: Instante en que el talón se eleva del suelo.
- Prebalanceo: Momento en que los dedos se elevan del suelo.

La **fase de balanceo** está dividida en tres intervalos:

- Balanceo inicial: Se caracteriza por la rápida aceleración del extremo de la pierna inmediatamente después que los dedos dejan el suelo.
- Balanceo medio: La pierna de movimiento rebasa a la pierna de apoyo como un péndulo.
- Balanceo final: La pierna desacelera al acercarse al final del intervalo.

Tabla 2-1. Fases del ciclo de la marcha [18].

PERIODO	FASES	% DEL CICLO
PERIODO DE APOYO	Contacto inicial (CI)	0 – 2 %
	Respuesta de carga (RC)	2 – 10 %
	Apoyo medio (AM)	10 – 30 %
	Apoyo final (AF)	30 – 50 %
	Prebalanceo (PB)	50 – 60 %
PERIODO DE BALANCEO	Balanceo inicial (BI)	60 – 73 %
	Balanceo medio (BM)	73 – 87 %
	Balanceo final (BF)	87 – 100 %

## 2.2. Procedimientos de rehabilitación de la marcha

### 2.2.1. Uso del bastón como apoyo para la marcha

“El uso del bastón particularmente lo asociamos con los adultos mayores, sin embargo, es de suma importancia aprender a utilizarlo para reeducar la marcha después de una cirugía o lesión en la pierna a cualquier edad, ya que la extremidad afectada necesitará apoyo mientras esté sanando. El bastón puede ser una excelente opción si hay desórdenes del equilibrio y la estabilidad, acompañados o no de dolor y debilidad” [20].

En el mercado existen dos tipos de bastones más usados: Bastón de 1 apoyo y Bastón de 4 apoyos (Figura 2-2).



Figura 2-2. Modelo de bastones: 1 apoyo y 4 apoyos [20].

El médico tratante o fisioterapeuta, recomendará el modelo de bastón adecuado para el paciente, acorde a sus necesidades que dependerá de la patología y de cuanto apoyo necesita. Si, por el contrario, el uso de bastón es contraproducente, ya sea que le produzca dolor, problemas del equilibrio o debilidad, el fisioterapeuta le podría indicar un andador (caminador) o muletas pueden ser una mejor opción [20].

### 2.2.2. Información básica del bastón - uso

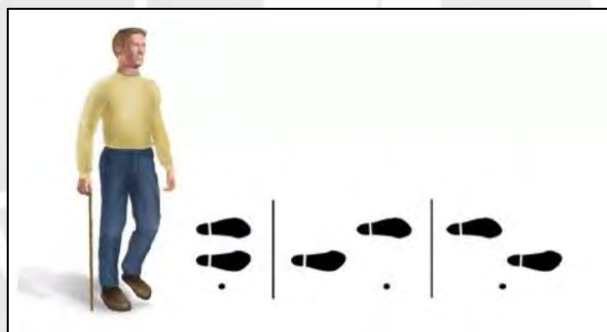
En [20] se explica que el bastón deberá usarse en la mano opuesta a la pierna afectada, ya sea por debilidad o cirugía. Los cuatro apoyos o un apoyo del bastón deberán estar en el suelo antes de dar el paso o cargar su peso en él. Un dato importante es mirar siempre hacia adelante al caminar, no a los pies y, sobre todo, verificar la altura del bastón, tiene que estar ajustado de acuerdo con su estatura:

El codo debe flexionarse ligeramente al sostener el mango.

El mango debe estar a nivel de muñeca y del trocánter mayor (hueso de la cadera).

A continuación, hay una serie de pasos a seguir cuando camine con un bastón:

- a) La postura al estar de pie deberá ser completamente erguido con un agarre firme sobre el bastón.
- b) Al dar un paso hacia adelante con la pierna más débil, simultáneamente, balancee el bastón hacia delante de usted la misma distancia, de modo que la punta del bastón (1 o 4 apoyos) y su pie delantero deben estar igualados.
- c) Ejerza presión sobre el bastón para quitar algo de la presión ejercida sobre su pierna débil.
- d) Al dar el paso con la pierna fuerte, asegúrese de ir más allá del bastón.
- e) Repita los pasos 1, 2 y 3.
- f) Para girar, hágalo sobre su pierna fuerte.
- g) Puede llevarle algo de tiempo acostumbrarse a caminar con un auxiliar de la marcha, como lo es el bastón.



*Figura 2-3. Caminando con el bastón sobre pisos planos [20].*

### **2.2.3. Objetivos de los procedimientos clínicos de rehabilitación**

Según la “Organización Panamericana de la Salud – Oficina Regional de la Organización Mundial de la Salud” define la Rehabilitación, como un “acto de atención de la salud y por lo tanto un componente imprescindible de la promoción, la prevención y la asistencia a las personas en el mantenimiento de su salud y bienestar, el de su familia y la comunidad”. En esta alternativa de tratamiento no farmacológico, los familiares juegan un rol muy importante debido a que serán el apoyo para llevar a cabo el tratamiento de rehabilitación del paciente hasta lograr su autonomía.

El objetivo principal es lograr la Rehabilitación Funcional, y para lograrlo se trazan los siguientes objetivos específicos [21]:

- Readaptación desde el punto de vista psicológico, social y biológico.
- Resocialización para que se mantenga activo en su medio habitual.
- Reincorporación a la familia.

#### 2.2.4. Principios de la rehabilitación

Según la OPS [21], los principios fundamentales son:

- Mantener la máxima movilidad.
- Mantener o restaurar la independencia en actividades básicas de la vida diaria.
- Desde el inicio del tratamiento obtener la confianza y cooperación del paciente.
- Crear una atmósfera razonablemente esperanzadora y optimista.
- Aplicar tratamientos simples e individualizados.
- Efectuar las sesiones terapéuticas por cortos períodos de tiempo.
- Métodos aplicados deberán ser iguales a los aplicados al adulto joven, pero tomando en cuenta la edad, enfermedad de base y capacidad vital.
- Integración de todo el personal de asistencia.

#### 2.2.5. Procedimiento clínico general para rehabilitación de la marcha

El análisis de movimiento, en particular el análisis de la marcha (y la postura), proporciona datos preoperatorios y postoperatorios para la evaluación de resultados. La inclusión de datos de análisis de la marcha en los planes de tratamiento es ampliamente reconocida como una herramienta clínicamente útil para identificar problemas con la movilidad y ha contribuido al desarrollo de aparatos ortopédicos y nuevas técnicas quirúrgicas [22] [23]. El **monitoreo de la marcha** [24], es importante porque puede ayudar a una buena recuperación después de una lesión y/o cirugía, por ello, se suele usar cuando la rehabilitación del paciente así lo amerite, las sillas de ruedas o **bastones inteligentes dotados de diferentes sensores**, que se pueden conectar fácilmente a estas ayudas de movilidad si las personas los requieren para su uso diario.

Considerando que cada paciente es único y puede requerir enfoques adaptados a sus necesidades individuales, es importante trabajar en colaboración con profesionales de la salud, como fisioterapeutas, terapeutas ocupacionales y médicos para diseñar un plan de rehabilitación personalizado. El procedimiento sería<sup>8</sup>:

1. Evaluación inicial:

---

<sup>8</sup> <https://fisiodomicili.com/ejercicios-reeducacion-marcha-adulto-mayor/>

- Realizar una evaluación completa del historial médico del paciente, incluyendo la condición médica subyacente, cirugías previas, y cualquier otra información relevante.
  - Realizar una evaluación física para identificar debilidades musculares, limitaciones de movimiento, desequilibrios y problemas de coordinación que puedan afectar la marcha.
2. Objetivos de rehabilitación:
- Establecer metas realistas y específicas para la rehabilitación de la marcha, teniendo en cuenta las capacidades y limitaciones individuales del paciente.
  - Los objetivos pueden incluir mejorar la fuerza muscular, la flexibilidad, la coordinación, el equilibrio y la resistencia.
3. Diseño de un plan de tratamiento:
- Desarrollar un plan de tratamiento personalizado que aborde las necesidades específicas del paciente.
  - Incluir ejercicios de fortalecimiento muscular, ejercicios de equilibrio y coordinación, técnicas de estiramiento y otros enfoques terapéuticos.
4. Terapia física:
- Implementar sesiones regulares de terapia física dirigidas por un fisioterapeuta.
  - Utilizar ejercicios específicos para mejorar la fuerza y la flexibilidad, así como técnicas de entrenamiento de la marcha.



*Figura 2-4. Ejercicio de fisioterapia<sup>9</sup>*

5. Entrenamiento de la marcha:

---

<sup>9</sup> <https://fisiodomicili.com/ejercicios-reeducacion-marcha-adulto-mayor/>

- Iniciar el entrenamiento de la marcha en una superficie plana y segura (Figura 2-3).
  - Gradualmente progresar hacia terrenos más desafiantes y situaciones más complejas, como subir escaleras o caminar sobre superficies irregulares como se muestra en la Figura 2-5.
6. Uso de ayudas para la marcha:
- Introducir ayudas para la marcha, como bastones o andadores, según sea necesario.
  - Ajustar y modificar las ayudas para la marcha según la progresión del paciente.
7. Educación del paciente:
- Proporcionar educación al paciente y a los cuidadores sobre técnicas de ejercicios, prevención de caídas y estrategias para mantener la funcionalidad a largo plazo.



Figura 2-5. Entrenamiento para usar bastón en escalera<sup>10</sup>.

8. Monitoreo y ajuste:
- Evaluar regularmente el progreso del paciente y ajustar el plan de rehabilitación según sea necesario.
  - Abordar cualquier nueva limitación o desafío que pueda surgir durante el proceso de rehabilitación.
9. Colaboración interdisciplinaria:
- Trabajar en colaboración con otros profesionales de la salud, como terapeutas ocupacionales, ortopedas y médicos especialistas, para abordar de manera integral las necesidades del paciente.
10. Seguimiento a largo plazo:
- Establecer un plan de seguimiento a largo plazo para garantizar que el paciente mantenga y mejore su capacidad de marcha con el tiempo.

---

<sup>10</sup> <https://www.fisioterapia-online.com/articulos/uso-del-baston-como-apoyo-para-la-marcha>

- Realizar ajustes en el tratamiento según sea necesario para abordar cambios en la condición del paciente.



Figura 2-6. Fortaleciendo músculos y estabilidad<sup>11</sup>.

### 2.3. Dispositivos de asistencia tecnológica: Bastón

Con respecto a este punto, existen bastones simples o convencionales que muchos pacientes usan para recuperarse de alguna lesión muscular, fracturas, etc. Sin embargo, el inconveniente de estos bastones es que no proporcionan retroalimentación al médico tratante, lo que impide saber si el paciente ha cumplido con todas las indicaciones dadas. Esto puede ocasionar que la recuperación no sea la esperada y se prolongue por más tiempo.

En tal sentido, el bastón con tecnología electrónica, motivo de la presente tesis, pretende ofrecer al paciente una forma de retroalimentación en su propia recuperación. Este bastón dispone de un sistema que registra en memoria durante el uso diario, el peso corporal aplicado, el ángulo de inclinación del bastón, la hora y la fecha. Además, es posible, con la ayuda de un familiar o amistad, monitorear en tiempo real a través del celular smartphone y mediante comunicación Bluetooth, el peso corporal aplicado en cada pisada y el ángulo de inclinación (pitch y roll). Esta información sería útil para que el familiar ayude a corregir o mejorar la postura conforme a las indicaciones del médico tratante.

### 2.4. Sensores de movimiento y fuerza usados en rehabilitación

Los sensores de movimiento y fuerza usados para rehabilitación existen de diferentes tipos y marcas, cada uno con sus propias características que serán validadas de acuerdo con la aplicación. Los acelerómetros se utilizarán para analizar la marcha, el equilibrio y la fuerza muscular tanto en el hogar como en el entorno clínico [25].

<sup>11</sup> <https://fisiodomicili.com/ejercicios-reeducacion-marcha-adulto-mayor/>

En la siguiente tabla se muestra las principales características de los distintos tipos de acelerómetros y sus aplicaciones:

Tabla 2-2. Características de los distintos fabricantes de acelerómetros

Fuente: <https://biblus.us.es/bibing/proyectos/abreproy/11638/fichero/Capitulo+4.pdf>

Tipo	Margen de Medida	Ancho de Banda	Ventajas e Inconvenientes	Aplicaciones
MENS	1.5g – 250g	0.1 – 1500	-Alta Sensibilidad -Coste Medio -Uso sencillo -Bajas Temperaturas	-Impacto -ABS -Airbag -Uso en automoción
Piezoeléctricos	0g – 2000g	10 – 20000	-Sensibilidad media -Uso complejo -Bajas temperaturas -No funcionan en continua	-Vibración -Impacto -Uso Industrial
Piezo-resistivos	0g – 2000g	0 – 10000	-Respuesta en continua y alterna -prestaciones medias -Bajo coste -Tamaño y peso mínimos -Alta Sensibilidad	-Vibración -impacto -Automoción -Biodinámica -Ensayos en vuelo -Test de túneles de viento
Capacitivos	0g – 1000g	0 – 2000	-Funciona en continua -Bajo ruido -Baja Potencia -Bajo coste	-Uso general -Uso industrial -Sistemas de Alarma -Mediciones Sísmicas
Mecánicos	0g – 200g	0 – 1000	-Alta precisión en continua -Lentos -Alto coste	-Navegación inercial -Guía de misiles -Herramientas -Nivelación

El término "**margen de medida**" de un acelerómetro, conocido como rango de medición o rango dinámico, es el límite de aceleración máximo y mínimo que el sensor puede medir de forma precisa. Este rango se expresa en unidades de aceleración, como metros por segundo cuadrado ( $m/s^2$ ) o en múltiplos de la gravedad terrestre (g). Por ejemplo, un acelerómetro con un margen de  $\pm 2g$  puede registrar aceleraciones que oscilan entre  $-2g$  y  $+2g$ . Si la aceleración excede este límite, el sensor no podrá medirla correctamente, lo que lleva a la saturación o truncamiento de la señal. La elección de un margen adecuado depende de la aplicación: un rango bajo es ideal para detectar vibraciones pequeñas, mientras que un rango mayor es necesario para capturar movimientos bruscos o aceleraciones elevadas.

El término "**ancho de banda**" en las características de un acelerómetro se refiere a la gama de frecuencias de señales (movimientos o vibraciones) que el acelerómetro puede medir de manera precisa. En otras palabras, indica la capacidad del acelerómetro para detectar aceleraciones a diferentes velocidades o frecuencias. Por ejemplo, si un acelerómetro tiene un ancho de banda de 0-1000 Hz, podrá medir aceleraciones de cualquier frecuencia dentro de ese rango. Si el ancho de banda es más pequeño, como 0-100 Hz, solo podrá medir movimientos lentos o suaves.

### 2.4.1. Acelerómetro MENS

Cualquier objeto, sin importar su masa o posición, experimenta una aceleración de  $9.81 \text{ m/s}^2$  hacia el centro de la Tierra, por lo que, un acelerómetro es un dispositivo diseñado para medir esa aceleración, que representa la velocidad de cambio en el movimiento de un objeto. La medición de la aceleración se expresa en metros por segundo al cuadrado ( $\text{m/s}^2$ ) o en unidades de fuerza G (donde  $g = 9.81 \text{ m/s}^2$ ).

Existen diversas tecnologías y fabricantes, pero en este proyecto se usará la tecnología MENS (MicroElectroMechanical Systems) del fabricante ANALOG DEVICES (ADXL345) [26].

*“MEMS es la tecnología de creación de dispositivos diminutos que funcionan mediante electricidad. El tamaño de un dispositivo de este tipo varía entre los 20 micrómetros y un milímetro. A su vez, los componentes que lo forman tienen tamaños que van desde uno hasta 100 micrómetros, haciéndolos ideales para sistemas incrustados al ser más pequeños, rápidos, eficientes y baratos”<sup>12</sup>.*

El ADXL345 es un acelerómetro micromecanizado capacitivo de 3 ejes independientes (3 DOF), que detecta la aceleración en los ejes X, Y y Z. También es posible determinar la orientación del sensor, gracias a la acción de la fuerza de la gravedad, tiene un consumo muy pequeño, cerca de  $45 \mu\text{A}$  en modo de medición y  $0.1 \mu\text{A}$  en standby. Además, dispone de un bloque de memoria FIFO que almacena hasta 32 conjuntos X, Y, Z. La comunicación puede realizarse tanto por bus SPI como por bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de bajo voltaje entre 2.0 a 3.6V.

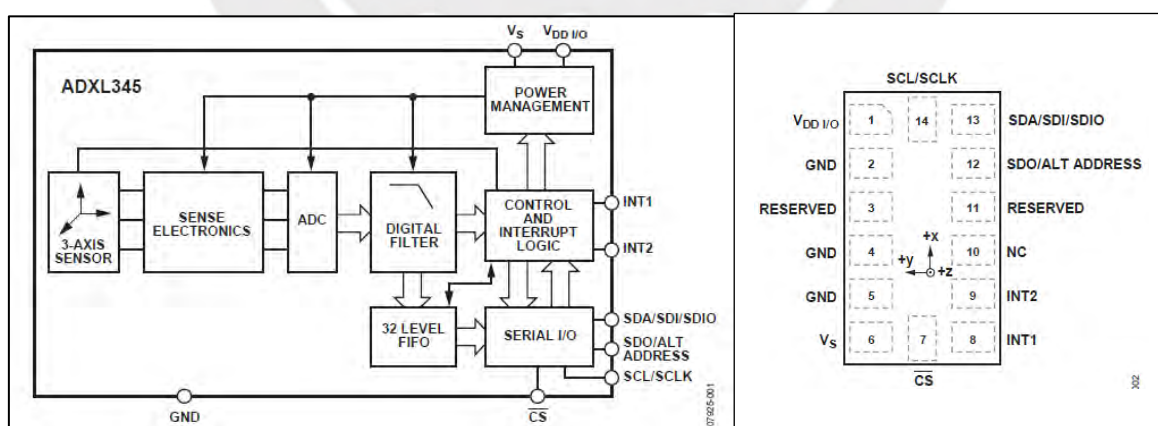


Figura 2-7. Diagrama de bloques funcionales y pines del acelerómetro MEMS ADXL345.

<sup>12</sup> <https://www.etsist.upm.es/estaticos/ingeniatic/index.php/tecnologias/item/516-mems-microelectromechanical-systems.html>

El ADXL345 tiene un rango de medición ajustable entre  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$ , y una alta resolución de hasta 13 bits, con una sensibilidad de 40mg/LSB en todos los rangos, lo que equivale a una precisión superior a un 1°. También dispone de dos pines de interrupciones que podemos configurar para responder a ciertos eventos como detección de movimientos rápidos, golpes y vibraciones, en uno o dos pulsos, y detección de condiciones de caída libre 0-g.

#### **2.4.2. Sensores de fuerza**

Los sensores de fuerza, o células de carga, son dispositivos que permiten obtener una señal eléctrica proporcional a la fuerza que se aplica sobre ellos, es decir, fuerza ejercida en voltaje. Estos transductores se presentan en múltiples formatos dependiendo de la aplicación [27] y requisitos del proyecto a desarrollar, se dividen en dos grandes grupos: dinamómetros y células de carga.

La gama de dinamómetros está concebida para aplicaciones en máquinas de ensayo o instrumentos para calibración. Son dispositivos de gran precisión y estabilidad, que se verifican y clasifican según la norma ISO-736 (clases 00, 05, 1 y 2) [28].

La gama de sensores clasificados como células de carga están pensados para aplicaciones en sistemas de pesaje o instrumentos de verificación en cadenas de producción o sobre el terreno. Su clasificación obedece a los criterios de la OIML-R60 (clases C2, C3 o C4) según la organización internacional de metrología (International Organization of Legal Metrology). Estos dispositivos sólo miden el esfuerzo en una dirección (eje de esfuerzo), bien pueden hacerlo en tracción o compresión. Por consiguiente, deben evitarse los esfuerzos en direcciones diferentes a la indicada ya que, podríamos llegar a dañar el dispositivo.

Para la selección adecuada de estos dispositivos, se debe tener en cuenta el tipo de aplicación del sistema, el rango máximo de fuerza que podemos aplicar y la precisión requerida, además de los condicionantes mecánicos del sistema.

##### **2.4.2.1. Sensor FlexiForce A201**

Según Teckscan [29], el sensor FlexiForce es un circuito impreso ultrafino y flexible que se puede integrar fácilmente en la mayoría de las aplicaciones debido a su construcción delgada como un papel, su flexibilidad y su capacidad de medición de fuerza, puede medir la fuerza entre casi dos superficies cualesquiera y es lo suficientemente duradero como para resistir cualquier entorno.



Figura 2-8. Sensor de fuerza piezoresistivo A201 – FlexiForce [29].

FlexiForce tiene mejores propiedades de detección de fuerza, linealidad, histéresis, deriva y sensibilidad a la temperatura que cualquier otro sensor de fuerza de película delgada. El "área de detección activa" es un círculo de 0,375" de diámetro al final del sensor y para saber cuánta fuerza o presión se está ejerciendo, la señal de salida es un cambio de resistencia, y entre más se presione menor será la resistencia.

Para este proyecto se eligió este sensor principalmente por su linealidad, precisión y costo, aunque había otros sensores, pero este es el que se adaptó mejor al bastón. La siguiente tabla muestra las principales características de estos sensores.

Tabla 2-3 Sensor FlexiForce estándar (modelo A201) [29].

Thickness	0.008 (0.208 mm)
Length	8" (203 mm) 6" (152 mm) 4" (102 mm) 2" (51 mm)
Width	0.55" (14 mm)
Sensing Area	0.375" (9.53 mm) diameter
Connector	3-pin male square pin (center pin is inactive)
Force Ranges	0-1 lb (4.4 N) 0-25 lbs (110 N) 0-100 lbs (440 N)*
Operating Temperature Range	15°F to 140°F (-9°C to 60°C)
Linearity (Error)	+/- 3%
Repeatability	+/- 2.5% of full scale (conditioned sensor, 80% force applied)
Hysteresis	<4.5% of full scale (conditioned sensor, 80% force applied)
Drift	<5% per logarithmic time scale (constant load of 90% sensor rating)
Response Time	<5 microseconds
Output Change/Degree F	Up to 0.2% (~0.36% / °C). Loads <10 lbs, operating temperature can be increased to 165°F (74°C).

## 2.5. Tecnología inalámbrica

Las tecnologías inalámbricas se han incorporado a numerosos productos, servicios y redes en diversas industrias de todo el mundo y han cambiado nuestros estilos de vida en las últimas dos décadas. Estas tecnologías evolucionan y se mejoran continuamente para abordar la creciente demanda de los usuarios en un mundo digital conectado [30] [31].

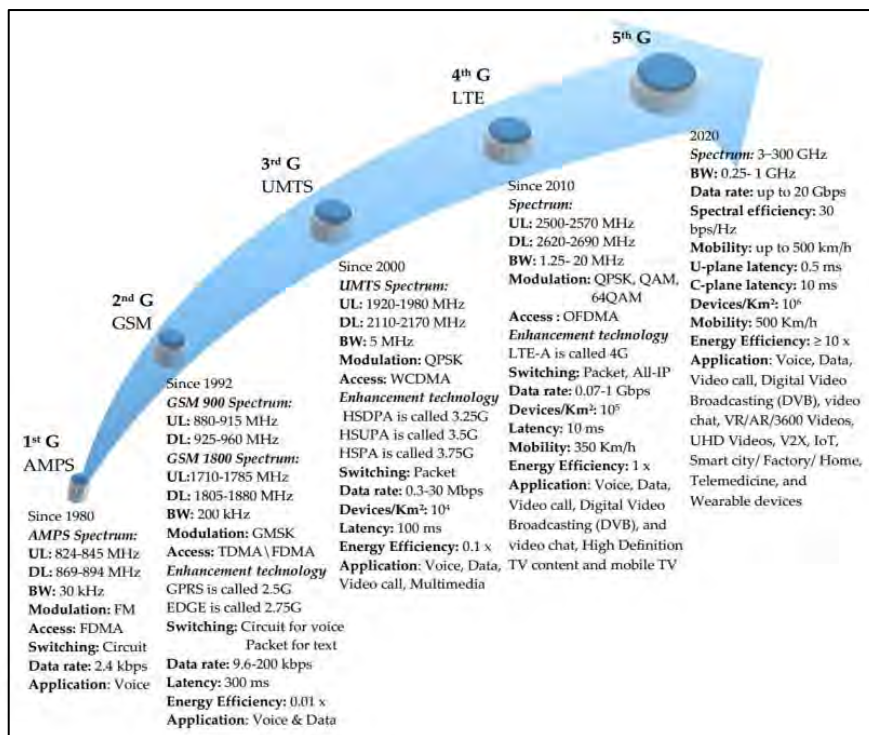


Figura 2-9. Leading communication achievements for several generations (1G to 5G) [31].

El sistema de comunicación inalámbrico más básico consiste en un Transmisor, Receptor y un canal de transmisión, generalmente una radio enlace, como se muestra en la Figura 2-10. La modulación también permite que el canal de radio pueda ser utilizado por más de una señal simplemente utilizando una frecuencia portadora diferente para cada uno. La demodulación es el proceso de modulación inversa, que se realiza en el receptor para recuperar la información original [32].

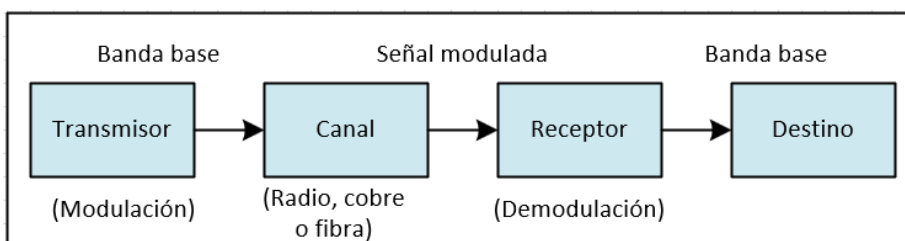


Figura 2-10. Diagrama de bloques de un sistema inalámbrico básico [32]. Adaptación propia.

En el presente, la tecnología inalámbrica está experimentando un rápido desarrollo, especialmente con el incremento significativo de dispositivos inalámbricos como teléfonos celulares, relojes, SmartTV, entre otros. Esta proliferación de dispositivos ha generado una creciente dependencia global para la comunicación e intercambio de información. Como resultado, se han presentado varios desafíos en las redes inalámbricas y su integración con Internet [33].

Según Jordy Salazar [34], las comunicaciones inalámbricas se pueden clasificar en cuatro grupos específicos según el área de aplicación y el alcance de la señal [35] [36] [37]: redes inalámbricas de área personal (Wireless Personal-Area Networks - WPAN), redes inalámbricas de área local (Wireless Local-Area Networks - WLAN), redes inalámbricas de área metropolitana (Wireless Metropolitan-Area Networks - WMAN), y redes inalámbricas de área amplia (Wireless Wide-Area Networks - WWAN). La Figura 2-11 ilustra estas cuatro categorías.

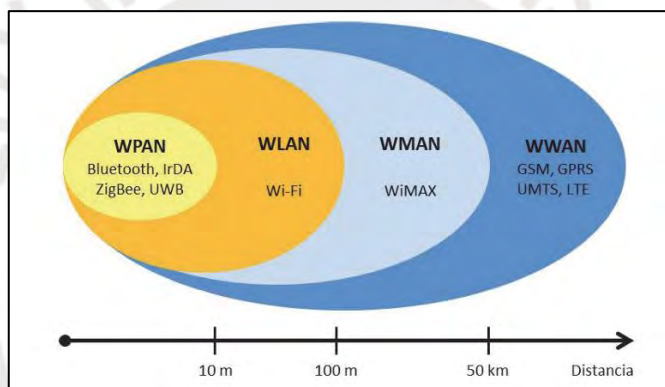


Figura 2-11. Clasificación de las tecnologías inalámbricas [34].

### 2.5.1. Módulo Bluetooth



Figura 2-12. Periféricos conectados por Bluetooth

(fuente: <https://acosta-redaccion-sistemas-2019-i.blogspot.com/2019/04/semana-3.html>)

Esta tecnología se ha convertido en un estándar de comunicaciones inalámbricas basado en radiofrecuencia, de bajo coste y bajo consumo energético [38]. En 1994, la compañía Ericsson Mobile Communications, lo desarrolló como un mecanismo alternativo que permitiese sustituir paulatinamente los enlaces cableados de diversos periféricos (puerto serial RS232, cable USB, cable de teclado y ratón PS2, conector de audio y video RCA, etc.).

No obstante, las características y versatilidad que presenta Bluetooth permiten que dos dispositivos se conecten directamente sin necesidad de infraestructura de red de respaldo, como un enrutador inalámbrico o un punto de acceso. Hoy en día, la tecnología Bluetooth es la más comúnmente usada por las personas de todo el mundo para conectar periféricos como audífonos, teclados, cámaras, y televisores, así como para la conexión inalámbrica muy utilizada para transmitir música entre el celular y el auto radio por medio de Apple CarPlay o Android Auto. Además, se emplea en dispositivos médicos, como los sensores de glucosa, marcapasos y otros.

La tecnología Bluetooth<sup>13</sup> funciona en frecuencias de radio en el rango de 2,4 GHz. Los dos estándares de Bluetooth actualmente en uso son:

**Bluetooth Classic**, que admite dos velocidades de datos distintas, la velocidad básica (BR) y la velocidad de datos mejorada (EDR).

**Bluetooth Low Energy (LE)**, que está optimizado para bajo consumo de energía y se usa principalmente para aplicaciones que están restringidas por la autonomía de la batería. Bluetooth LE no se usa comúnmente para intercambiar grandes cantidades de datos, pero ofrecerá compatibilidad con una mayor calidad de audio y más opciones de escucha que Bluetooth Classic.

En la siguiente tabla se mostrará las principales diferencias entre versiones de Bluetooth:

*Tabla 2-4. Diferencias entre las versiones de Bluetooth.*

	Bluetooth 1.0	Bluetooth 2.0	Bluetooth 3.0	Bluetooth 4.0	Bluetooth 5.0
<b>Rango</b>	Hasta 10m	Hasta 30m	Hasta 30m	Hasta 60m	Hasta 240m
<b>Basic Rate (BR)</b>	√	√	√	√	√
<b>Enhanced Data Rate (EDR)</b>	x	√	√	√	√
<b>High Speed (HS)</b>	x	x	√	√	√
<b>Low Energy (LE)</b>	x	x	x	√	√
<b>Slot Availability Masking (SAM)</b>	x	x	x	√	√

<sup>13</sup> <https://www.intel.la/content/www/xl/es/products/docs/wireless/what-is-bluetooth.html>

**Basic Rate (BR)** – Es un modo en el que la velocidad de transferencia instantánea de datos está limitada a un bit rate de tan solo 1Mbit/s o 125 KBps.

**Enhanced Data Rate (EDR)** – Es un modo que permite la transferencia de datos con velocidades de hasta 3Mbit/s o 375 KBps.

**High Speed (HS)** – Es el primer salto importante como modo de transferencia de datos, utilizando una capa de vinculación 802.11 que permite velocidades teóricas de hasta 24 Mbit/s o 3000 KBps.

**Low Energy (LE)** – Es una tecnología desarrollada que permite reducir drásticamente el consumo energético de dispositivos bluetooth. Su existencia y funcionamiento son de gran relevancia en la actualidad.

**Slot Masking Availability (SAM)** – Esta función tiene la misión de mejorar la coexistencia del Bluetooth con otro tipo de ondas de radio. Básicamente reduce las interferencias, intentando detectar frecuencias cercanas y alejándose de dichos canales.

Bluetooth [39] está integrado en un pequeño transmisor de radiofrecuencia que permite conectar entre sí todo tipo de dispositivos electrónicos (teléfonos, ordenadores, impresoras, faxes, etc.) situados dentro de un radio limitado de 10 metros con posibilidad de ampliarse a 100, aunque con mayor distorsión. Su transmisor está integrado en un pequeño microchip de 9x9 milímetros y opera en una frecuencia de banda global (2,4 GHz, utilizada en muchos países para usos médicos y científicos) que asegura la compatibilidad universal. Lo más resaltante de esta tecnología es que los dispositivos que incorporan Bluetooth se reconocen y se hablan de la misma forma que lo hace una computadora con su impresora. El canal permanece abierto y no requiere la intervención directa y constante del usuario cada vez que se quiere enviar algo.

El transmisor permite enviar voz y datos a una velocidad máxima de 700 Kbps. y consume un 97% menos que un teléfono móvil. Además, es inteligente: cuando el tráfico de datos disminuye el transmisor adopta el modo bajo de consumo de energía.

Tabla 2-5. Niveles de potencia y alcance de Bluetooth (Fuente: Bluetooth SIG Group).

Clases de radio	Potencia máxima (mW)	Potencia máxima (dBm)	Alcance aproximado	Aplicación principal
Clase 1	100	20	100 mt.	Industrial
Clase 2	2.5	4	10 mt.	Dispositivos móviles
Clase 3	1	0	1 mt.	

### 2.5.1.1. Topologías Bluetooth

Bluetooth ha sido diseñado para operar en un ambiente multiusuario. La topología de una red Bluetooth puede ser punto a punto o punto-multipunto [39]. En este tipo de conexiones, el canal se comparte con varias unidades. Dos o más unidades compartiendo el mismo canal forman una piconet y cualquier unidad de una piconet, puede a su vez establecer una conexión a otra piconet para formar un conjunto de conexiones entre diversas piconets denominado Scatternet. En esta red, se tiene cuatro estados:

**M (Master):** puede manejar siete esclavos activos simultáneos y hasta 200 esclavos activos en la piconet.

**S (Slave):** terminales que pueden participar en más de un piconet.

**SB (Stand by):** Espera para unirse a la piconet, significa mantener su dirección MAC.

**P (Parked/hold):** Esperando para adherirse a la piconet más tarde y liberar su dirección MAC.

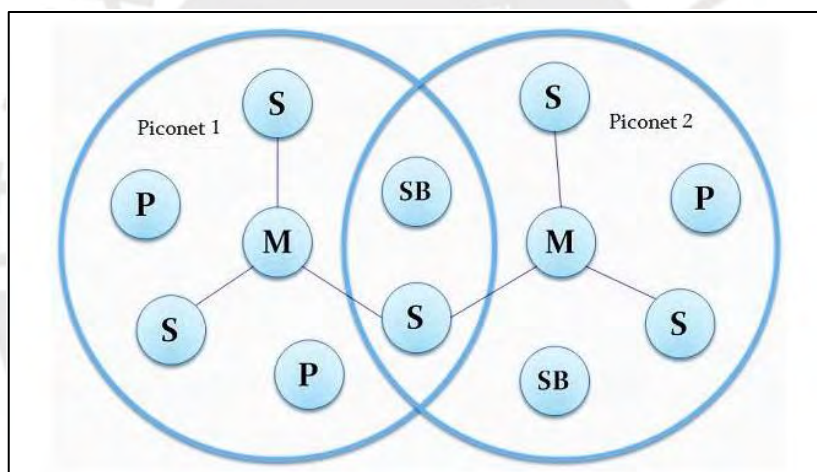


Figura 2-13. Topología de una red Bluetooth.  
(fuente: <https://pc-solucion.es/terminos/bluetooth/>)

La velocidad de datos en full-dúplex dentro de una estructura como la descrita en Figura 2-13, con 10 piconets con carga máxima es de 6 Mb/s, siendo un piconet la conexión de 2 a 8 dispositivos. Todos los dispositivos tienen la misma implementación, pero, al crearse la red una unidad actuará como maestra y el resto como esclavos mientras dure la conexión.

### 2.5.1.2. Hardware Bluetooth

El hardware esencial de un dispositivo Bluetooth consta de dos componentes principales. En primer lugar, tenemos el dispositivo de radio, responsable de modular y transmitir la señal. En segundo lugar, encontramos el controlador digital, que incluye una unidad central

de procesamiento (CPU) y un procesador de señales digitales (DSP - Digital Signal Processor) conocido como Controlador de Enlace (Link Controller), además de las interfaces con el dispositivo anfitrión. Esta estructura básica permite que el dispositivo Bluetooth funcione de manera eficiente al gestionar tanto la transmisión de datos inalámbrica como la comunicación con el dispositivo principal al que está conectado [40].

El LC o Controlador de Enlace está encargado de hacer el procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de capa física. Además, se encarga de las funciones de transferencia (tanto asíncrona como síncrona), codificación de Audio y cifrado de datos.

El CPU del dispositivo se encarga de atender las instrucciones relacionadas con Bluetooth del dispositivo anfitrión, para así simplificar su operación. Para ello, sobre el CPU corre un software denominado Link Manager que tiene la función de comunicarse con otros dispositivos por medio del protocolo LMP. La Figura 2-14 muestra la arquitectura de hardware Bluetooth.

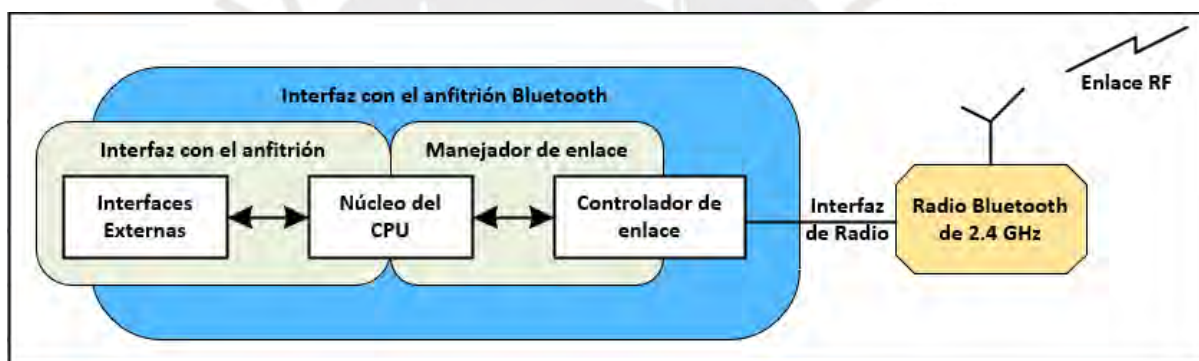


Figura 2-14. Arquitectura del Hardware Bluetooth (Adaptación propia).

Entre las tareas realizadas por el LC y el Link Manager, destacan las siguientes [40]:

- Envío y Recepción de Datos.
- Empaquetado y Peticiones.
- Determinación de Conexiones.
- Autenticación.
- Negociación y determinación de tipos de enlace, por ejemplo, SCO o ACL
- Determinación del tipo de cuerpo de cada paquete.

## Radio Bluetooth (RF)

Según las especificaciones de Bluetooth [41], la frecuencia de radio con la que opera la tecnología Bluetooth se encuentra en la banda no licenciada de 2.4 a 2.48 GHz, caracterizada por su amplio espectro y saltos rápidos de frecuencia. Esta capacidad de salto de frecuencia, con hasta 1600 saltos por segundo, se emplea para evitar interferencias y pérdidas de señal. En el caso del Bluetooth clásico, estos saltos de frecuencia se realizan entre un total de 79 frecuencias con intervalos de 1 MHz, brindando así seguridad y robustez al sistema. En Bluetooth 5.0, el sistema de baja energía (LE) opera en la misma banda, pero utiliza 40 canales de radiofrecuencia (RF) a intervalos de 2 MHz. No obstante, esta adaptación en el número de canales y sus intervalos en Bluetooth 5.0 busca mejorar la eficiencia y el rendimiento de la transmisión de datos inalámbrica en comparación con las versiones anteriores.

## Limitaciones y alcance de la Tecnología Bluetooth

La tecnología Bluetooth es crucialmente afectada por la ubicación y las condiciones circundantes del dispositivo físico para garantizar su óptimo desempeño. Su capacidad de conexión abarca distancias de aproximadamente 10 a 100 metros entre dispositivos. No obstante, este alcance puede variar según factores como el entorno, la presencia de interferencias y la potencia de transmisión. Específicamente, otras tecnologías que operan en el espectro de 2,4 GHz, como PCs, tabletas, teléfonos y redes WiFi más antiguos, junto con dispositivos como microondas y monitores para bebés, pueden interferir con la eficacia de Bluetooth.

Tabla 2-6. Bandas de frecuencias operativas [41].

Ancho de banda	Canales RF (Bluetooth básico)
2.400 – 2.4835 GHz	$f = 2402 + K \text{ MHz}$ , $k=0, 1, \dots, 78$

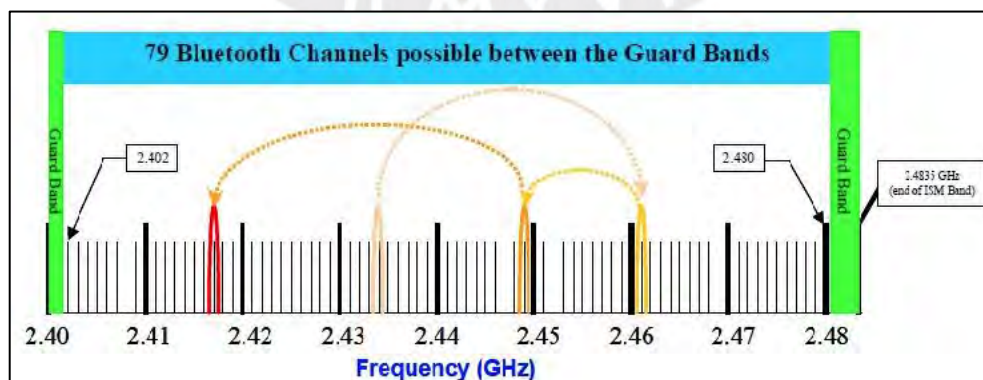


Figura 2-15. Espectro de frecuencias en Bluetooth clásico<sup>14</sup>

<sup>14</sup> <https://www.mokoblue.com/es/complete-guide-on-bluetooth-module/>

Tabla 2-7. Bandas de frecuencias BLE operativas [41].

Ancho de banda	Canales RF (Bluetooth LE)
2.400 – 2.4835 GHz	$f = 2402 + K * 2 \text{ MHz}$ , $k=0, 1, \dots, 39$

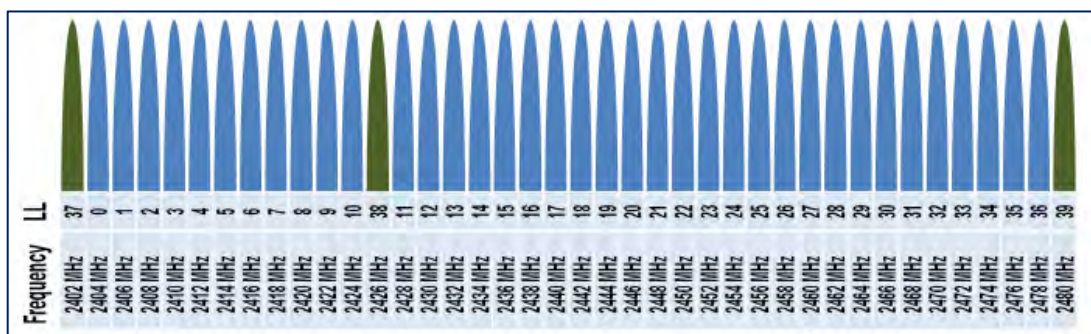


Figura 2-16. Espectro de frecuencias en Bluetooth BLE [40]

## 2.6. Microcontroladores ARM

El microcontrolador ARM significa Advance RISC Machine [42], se utilizan específicamente en dispositivos portátiles como cámaras digitales, teléfonos móviles, redes domésticas, módulos con tecnologías de comunicación inalámbrica y otros sistemas integrados debido a los beneficios, como bajo consumo de energía, rendimiento razonable, etc.

RISC, por ejemplo, fue desarrollado originalmente en la década de 1980, pero no tuvo el impacto deseado en el mercado hasta la llegada de los dispositivos móviles, promovido por **ARM Holdings**, la compañía detrás de esta arquitectura y que desarrolló un formato de instrucciones comprimido. Sin embargo, los procesadores **ARM** pueden llegar a ser todo un desafío ya que **ARM Holdings** no fabrica los procesadores por sí misma, sino que se dedica a **diseñar la tecnología y desarrollar el estándar** para luego licenciarla a otros fabricantes. Este es el motivo por el que hay tantas variantes de este tipo de procesadores.

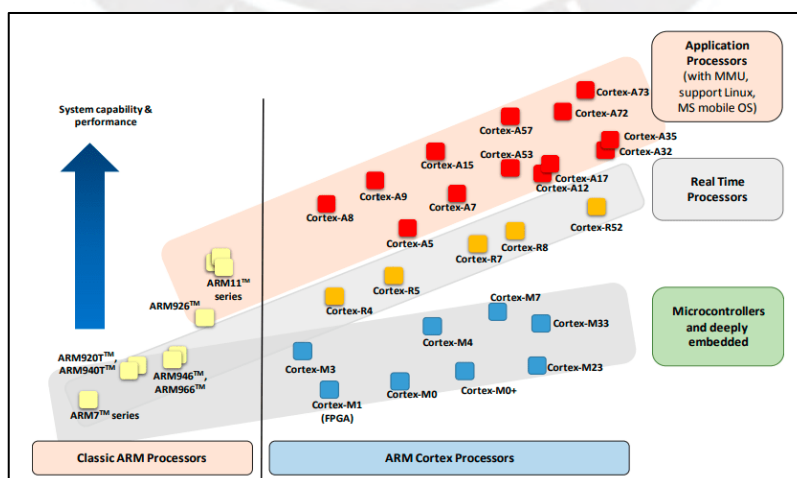


Figura 2-17. Clasificación de los procesadores ARM.

(fuente: <https://blog.330ohms.com/2021/02/17/que-es-arm/>)

Los fabricantes de hardware como ST Microelectronics, Motorola, Freescale, etc, pagan regalías a la compañía que diseña la tecnología central (**ARM Holdings**), pero luego lo adaptan a sus necesidades, requisitos de software y diseños de hardware. Los procesadores ARM tienen una arquitectura RISC de 32 bits y están diseñados para ser lo más eficiente posible, aceptando solo instrucciones que se puedan lograr en un único ciclo de memoria [43] [44].

En la gama ARM hay subfamilias con distinto objetivo, y que son:

**Cortex-A de "Application"**: Para aplicaciones de alto rendimiento con sistemas operativos generales, como Linux, Windows, sistemas operativos para móviles y celulares (Android, Windows mobile, etc.), lo que hace estos procesadores apropiados para usar en computadores, servidores, teléfonos celulares, tablets, smart-TV, etc.

**Cortex-R de "Real-time"**: Para sistemas de tiempo real críticos con buen rendimiento. Por ejemplo, control de automóviles, control de frenos para automóviles, control electrónico de motores, robótica, aviones, cámaras de vigilancia y fotográficas, periféricos de computadores, etc.

**Cortex-M de "Microcontroller"**: Para aplicaciones típicas de microcontroladores. Por ejemplo: lavadoras, microondas, equipos médicos, mandos, nodos inalámbricos, equipos de medición, electrónica de control industrial, juguetes, etc.

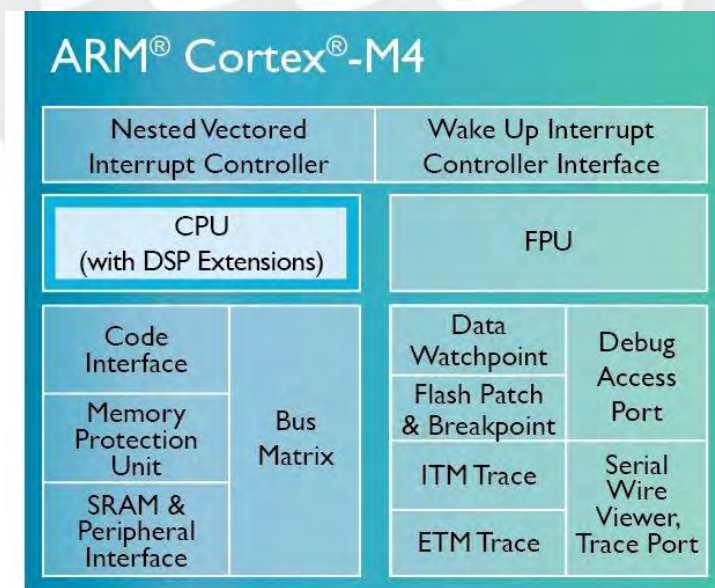


Figura 2-18. Arquitectura interna de un ARM Cortex-M4.

(fuente: <https://microcontrollerslab.com/arm-cortex-m4-architecture/>)

### 2.6.1. Módulo Teensy 3.2

En un principio del desarrollo de la Tesis, se usaba el Arduino Nano por su tamaño pequeño del PCB y tenía la interfaz con la computadora para descargar el programa. Sin embargo, a medida que avanzaba en el desarrollo del software, su memoria de programa de 32 KB resultó ser insuficiente. Por esta razón, fue cambiado por este microcontrolador ARM CORTEX-M4 MK20DX256VLH7 con memoria de 256KB, que viene incorporado en el módulo Teensy 3.2 con un tamaño de PCB de 35x18mm<sup>2</sup>.



Figura 2-19. Módulo ARM Teensy 3.2.

(Fuente: <https://naylampmechatronics.com/ardusystem-tarjetas/319-teensy-32.html>)

Este módulo es compatible con Arduino UNO en lo que se refiere al entorno de programación y distribución de pines.

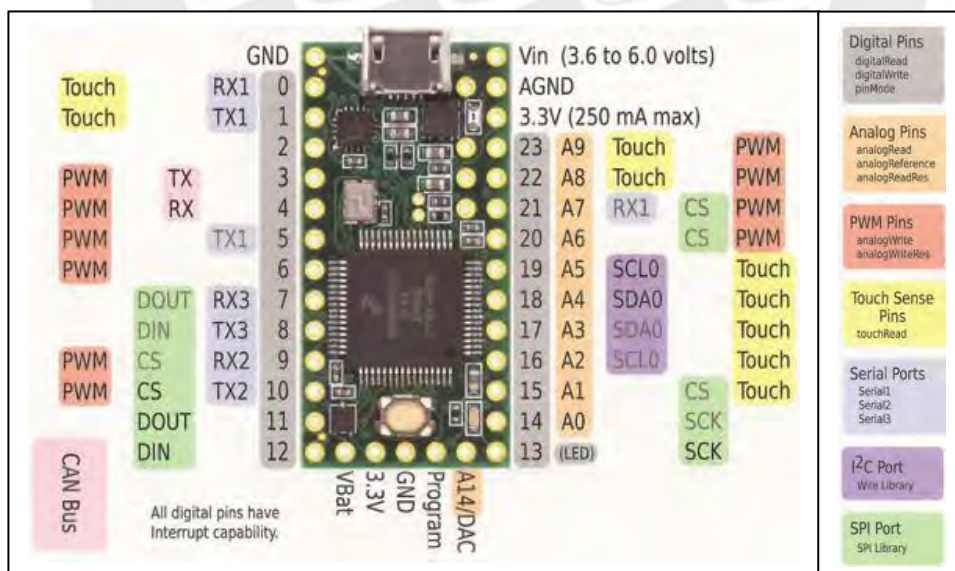


Figura 2-20. Distribución de pines del Módulo Teensy 3.2.

## ESPECIFICACIONES TÉCNICAS DEL MICROCONTROLADOR

Voltaje de Operación: 3.3V DC	Puertos UART: 3
Voltaje de alimentación: 3.6 - 6V	Puertos I2C: 2
Frecuencia de reloj: 72 MHz	1 puerto SPI
Memoria FLASH: 256 KB	Puerto I2S para audio de alta calidad
Memoria SRAM: 64 KB	Modulador IR
Memoria EEPROM: 2 KB	1 puerto BUS CAN
Pines digitales E/S: 34 (12 salidas PWM)	Reloj de Tiempo Real
Entradas analógicas (ADC): 21 (13 bit)	16 canales DMA
Salidas analógicas (DAC): 1 (12 bit)	Entradas para sensores táctiles capacitivos
7 Timers por hardware	Conector Micro-USB
USB con DMA dedicado	Dimensiones: 35x18mm

Para el desarrollo de la presente tesis se está empleando un canal ADC, el bus I2C y SPI principalmente, por lo que procederé a detallarlos:

### 2.6.2. Canal ADC

Las principales características del módulo ADC [45] del microcontrolador incluyen:

- Algoritmo de aproximación lineal sucesiva con resolución de hasta 16 bits.
- Hasta cuatro pares de entradas analógicas externas diferenciales y 24 de un solo extremo.
- Modos de salida:
  - Modos diferenciales de 16 bits, 13 bits, 11 bits y 9 bits.
  - Modos de extremo único de 16 bits, 12 bits, 10 bits y 8 bits.
- Tiempo de muestreo configurable y velocidad/potencia de conversión.
- Indicador e interrupción de conversión completada/promedio de hardware completo.
- Reloj de entrada seleccionable entre hasta cuatro fuentes.
- Funcionamiento en modos de bajo consumo para reducir el ruido.
- Referencia de voltaje seleccionable: externa o alterna.
- Modo de auto calibración.

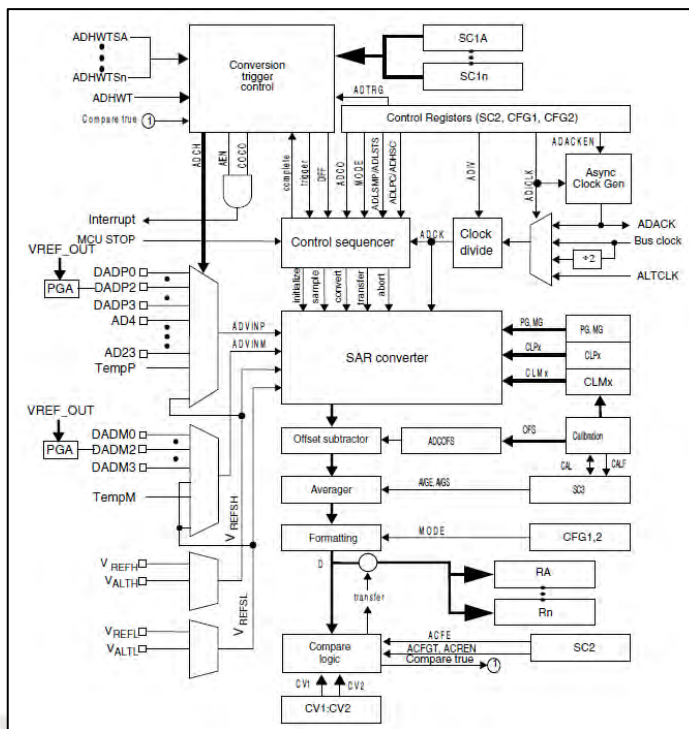


Figura 2-21. Diagrama de bloques del ADC [45].

### 2.6.3. Bus I2C (Inter-Integrated Circuit)

El módulo de circuito integrado (I2C, I2C o IIC) proporciona un método de comunicación entre varios dispositivos. La interfaz está diseñada para funcionar hasta 100 kbit/s con carga y sincronización máximas del bus, así como operar a velocidades de baudios más altas, hasta un máximo de reloj/20 con carga de bus reducida. La longitud máxima de comunicación y el número de dispositivos que se pueden conectar es limitado por una capacitancia de bus máxima de 400 pF. El módulo I2C también cumple con la especificación System Management Bus (SMBus), versión 2 [45] [46].

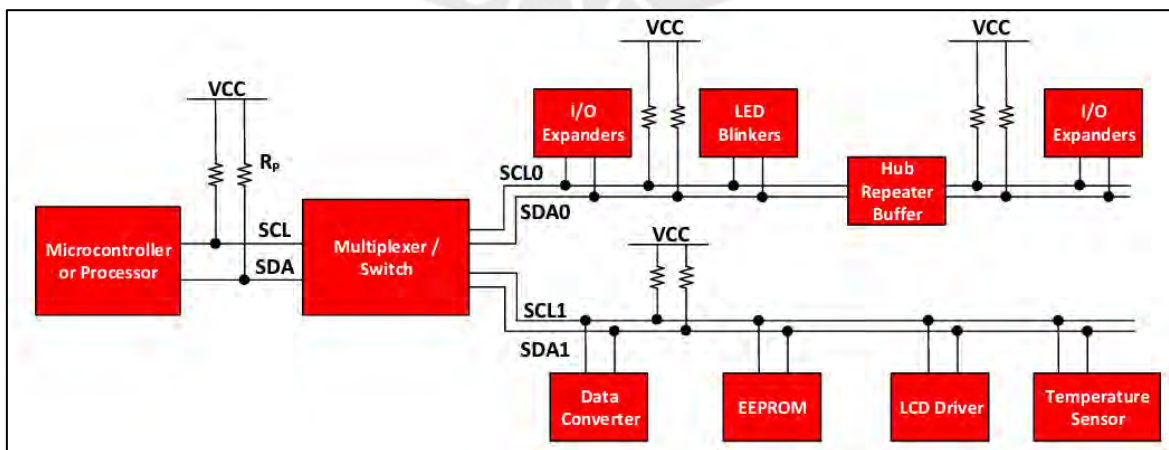


Figura 2-22. Ejemplo de bus I2C [46].

## Especificaciones del bus I2C

El módulo I2C tiene las siguientes características principales:

- Compatible con la especificación Bus- I2C.
- Operación multimaestro.
- Bit de reconocimiento seleccionable por software.
- Transferencia de datos byte a byte controlada por interrupción.
- Interrupción de identificación de dirección de llamada.
- Generación y detección de señales de INICIO y PARADA.
- Generación y detección de señales de INICIO repetidas.
- Extensión de dirección de 10 bits
- Compatibilidad con la especificación System Management Bus (SMBus), versión 2
- Activación en modo de bajo consumo al coincidir la dirección esclava.
- Compatibilidad con el rango de direcciones esclavas.
- Soporte DMA.

## Modos de operación

El funcionamiento del módulo I2C en varios modos de bajo consumo es el siguiente:

- Modo de ejecución: Este es el modo básico de operación. Para conservar energía en este modo, se debe desactivar el módulo.
- Modo de espera: El módulo continúa funcionando cuando el núcleo está en modo de espera y puede proporcionar una interrupción de activación.
- Modo de parada: El módulo está inactivo en el modo de stop para reducir el consumo de energía, excepto que la coincidencia de direcciones esté habilitada en el modo stop. La instrucción stop no afecta los estados del registro del módulo I2C.

## Diagrama de bloques

La siguiente figura es un diagrama de bloques funcional del módulo I2C.

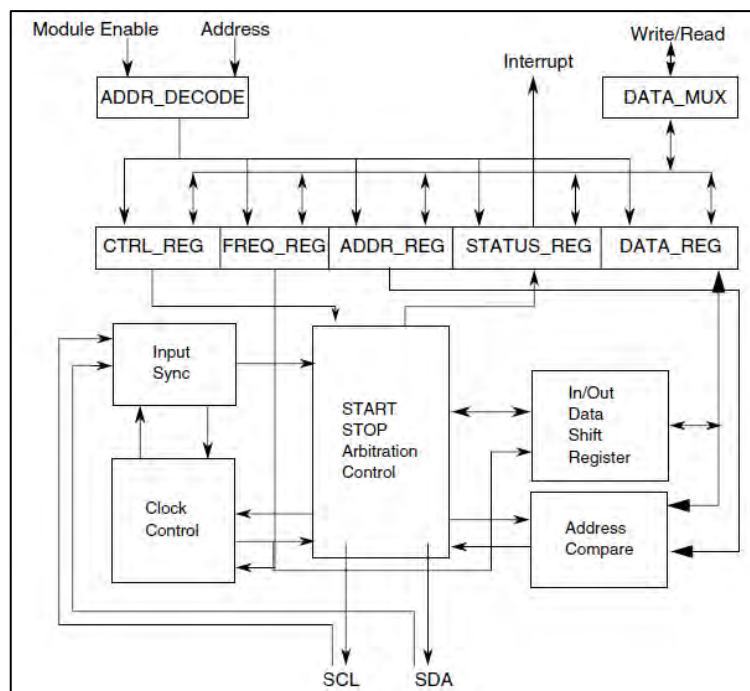


Figura 2-23. Diagrama de bloques funcional del I2C [45].

Donde:

- SCL: Línea de reloj serial bidireccional del sistema I2C.
- SDA: Línea de datos serie bidireccional del sistema I2C.
- Module Enable: Habilita el funcionamiento del módulo I2C.
- Address: Este registro contiene la dirección esclava que utilizará el módulo I2C.
- Interrupt: Flag de interrupción del Módulo I2C.
- Write/Read: Entrada que selecciona la opción de lectura/escritura en el módulo I2C.

### Descripción del protocolo de comunicación i2c

El sistema de bus I2C utiliza una línea de datos serial (SDA) y una línea de reloj serial (SCL) para transferencias de datos [46] [47]. Todos los dispositivos conectados a él deben tener salidas de drenaje abierto o colector abierto. La función lógica AND se ejerce en ambas líneas con resistencias pull-up externas. El valor de estas resistencias depende del sistema y el estándar de comunicación se compone de cuatro partes:

- señal de INICIO o START.
- Transmisión de la dirección del esclavo.
- Transferencia de datos.
- Señal de PARADA o STOP.

Cabe precisar, que la señal STOP no debe confundirse con la instrucción STOP de la CPU. La siguiente figura ilustra la comunicación del sistema de bus I2C.

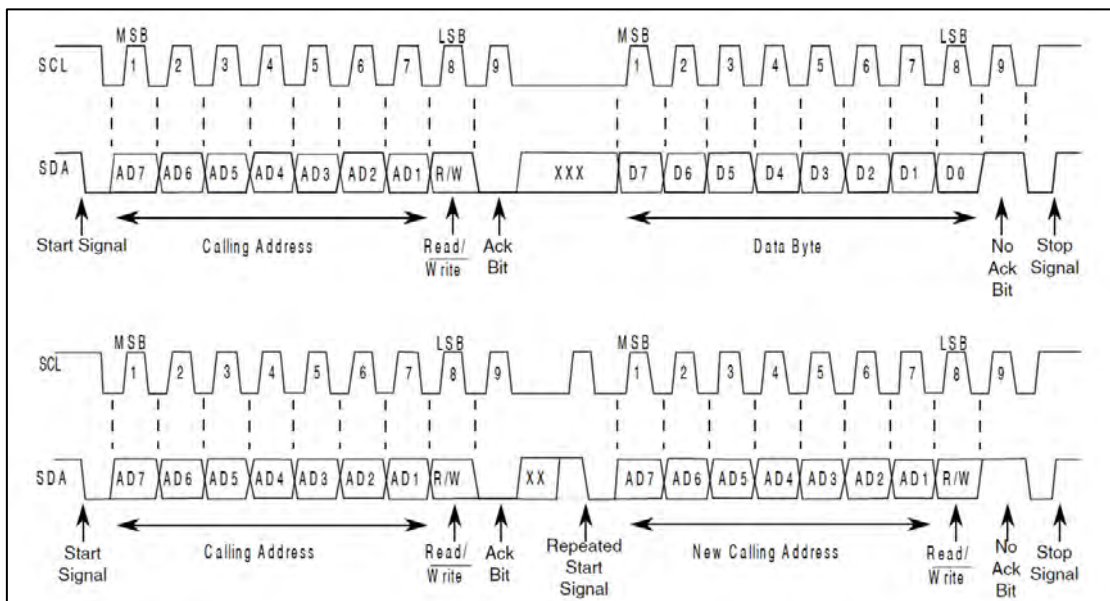


Figura 2-24. Señales de transmisión en el bus I2C [45].

A continuación, se describe las principales señales de control del bus I2C.

### Señal de Start o Inicio

El bus está libre cuando ningún dispositivo maestro está conectado al bus (tanto SCL como SDA están altos). Cuando el bus está libre, un maestro puede iniciar la comunicación enviando una señal de START. Una señal de START se define como una transición de alto a bajo de SDA mientras SCL está alto. Esta señal denota el comienzo de una nueva transferencia de datos; cada transferencia de datos puede contener varios bytes de datos y saca a todos los esclavos de sus estados inactivos.

### Transmisión de dirección de esclavo

Inmediatamente después de la señal START, el primer byte de una transferencia de datos es la dirección del esclavo transmitido por el maestro. Esta dirección consta de 7 bits seguida de un bit R/W. El bit R/W le dice al esclavo el tipo de transferencia de datos.

- 1 = Transferencia de lectura: El esclavo transmite datos al maestro.
- 0 = Transferencia de escritura: El maestro transmite datos al esclavo.

Sólo el esclavo con una dirección de llamada que coincida con la transmitida por el maestro responde enviando un bit de reconocimiento. El esclavo envía el bit de reconocimiento llevando SDA ha estado bajo en el noveno pulso de reloj.

No puede haber dos esclavos en el sistema con la misma dirección. Si el módulo I2C es el maestro, no debe transmitir una dirección que sea igual a su propia dirección esclava. El módulo I2C no puede ser amo y esclavo al mismo tiempo. Sin embargo, si el arbitraje se pierde durante un ciclo de direcciones, el módulo I2C vuelve al modo esclavo y funciona correctamente incluso si está siendo abordado por otro maestro.

### **Transferencias de datos**

Cuando se ha logrado un direccionamiento exitoso del esclavo, la transferencia de datos procede byte a byte, basándose en la dirección especificada por el bit R/W enviado por el maestro. Cada byte de datos tiene una longitud de 8 bits y puede modificarse solo cuando el SCL (reloj serial) está bajo, debiendo permanecer estables mientras el SCL es alto. Para cada bit de datos, hay un pulso de reloj en el SCL, y la transferencia sigue el formato de envío del bit más significativo (MSB) primero.

Después de cada byte de datos, se incluye un noveno bit (confirmación o acknowledge), el cual es enviado desde el dispositivo receptor bajando el SDA (línea de datos serial) en el noveno pulso de reloj. En resumen, una transferencia de datos completa requiere nueve pulsos de reloj.

Si el receptor esclavo no reconoce al maestro en el noveno bit, el esclavo debe colocar el SDA en un estado alto. El maestro interpreta la falta de reconocimiento como un fallo en la transferencia de datos. Por otro lado, si el maestro recibe un no-acknowledge del esclavo después de transmitir un byte de datos, el esclavo interpreta esto como el final de la transferencia de datos y libera la línea SDA.

En el caso de un reconocimiento fallido, tanto por parte del esclavo como del maestro, la transferencia de datos se cancela. En esta situación, el maestro tiene dos opciones: abandonar el bus generando una señal de STOP, o iniciar una nueva llamada generando una señal de INICIO repetida. Estos protocolos aseguran una comunicación efectiva y confiable entre dispositivos en el bus I2C.

### **Señal de STOP o PARADA**

El maestro puede terminar la comunicación generando una señal de STOP para liberar el bus. Una señal de STOP se define como una transición de bajo a alto de SDA mientras SCL se mantiene en alto. El maestro puede generar una señal de STOP incluso si el esclavo ha generado un acknowledge o reconocimiento, momento en el cual el esclavo debe liberar el bus.

## Handshaking

El mecanismo de sincronización del reloj se puede utilizar como protocolo de enlace en las transferencias de datos. El dispositivo esclavo puede mantener bajo la señal SCL después de completar una transferencia de un solo byte (9 bits). En este caso, detiene el reloj del bus y fuerza al reloj maestro a estados de espera hasta que el esclavo lanza SCL.

### 2.6.4. Bus SPI (Serial Peripheral Interface)

El estándar SPI es utilizado para la comunicación serial entre dispositivos [48]. El SPI fue inicialmente creado por Motorola y adoptado posteriormente por diferentes fabricantes. Se trata de un enlace de datos en serie síncrono, y que opera en modo full dúplex, es decir, las señales de datos viajan en ambas direcciones en forma simultánea<sup>15</sup>.

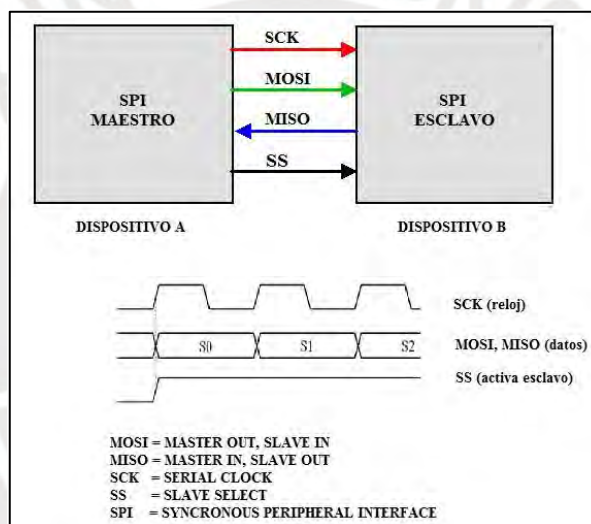


Figura 2-25. Estándar de comunicación SPI

(Fuente: <https://www.puntoflotante.net/COMUNICACION-SPI-TUTORIAL.htm>)

### Principales características del Bus SPI

**Comunicación Síncrona:** Es un protocolo síncrono, porque los dispositivos se sincronizan con una misma señal de reloj para transferir datos.

**Maestro-Esclavo:** El dispositivo que actúa como maestro controla la comunicación, mientras que uno o más dispositivos que actúan como esclavos responden a las solicitudes del maestro.

<sup>15</sup> <https://www.puntoflotante.net/COMUNICACION-SPI-TUTORIAL.htm>

**Múltiples Esclavos:** Un solo maestro puede comunicarse con múltiples dispositivos esclavos en el mismo bus. Cada dispositivo esclavo tiene su propio pin de selección de esclavo (SS) para habilitar o deshabilitar la comunicación con él.

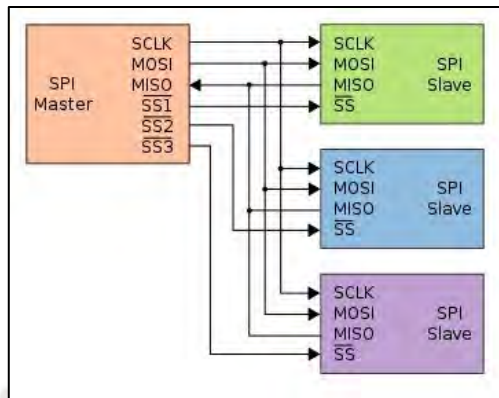


Figura 2-26. Red de comunicación SPI con un maestro y 3 esclavos.

(Fuente: <https://www.puntoflotante.net/COMUNICACION-SPI-TUTORIAL.htm>)

**Interfaz de 4 Hilos:** SPI utiliza una interfaz de 4 hilos para la comunicación:

- MOSI (Master Out Slave In): Utilizado por el maestro para enviar datos a los esclavos.
- MISO (Master In Slave Out): Utilizado por el maestro para recibir datos de los esclavos.
- SCK (Serial Clock): Proporciona el reloj de sincronización compartido entre el maestro y los esclavos.
- SS (Slave Select): Utilizado por el maestro para seleccionar un dispositivo esclavo específico con el que comunicarse.

**Transferencia Full-Duplex:** SPI permite la transferencia de datos full-duplex, lo que significa que los datos pueden ser enviados y recibidos simultáneamente.

**Velocidades de Transferencia Configurables:** Las velocidades de transferencia en SPI son configurables y dependen de la frecuencia del reloj de serie (SCK). Puede variar desde unos pocos kilobits por segundo hasta varios megabits por segundo, dependiendo de la aplicación y los dispositivos involucrados.

**Sin Protocolo de Dirección o Mensajes de Control:** A diferencia de otros protocolos como I2C, SPI no tiene un protocolo de dirección incorporado ni mensajes de control estándar. La comunicación se basa en la sincronización del reloj y la interpretación de los datos por parte del maestro y los esclavos.

## CAPITULO III

### DISEÑO DEL SISTEMA DE REHABILITACIÓN DE LA MARCHA BASADO EN BASTÓN ELECTRÓNICO

#### 3.1. Diseño del sistema asociado al bastón electrónico

El sistema desarrollado para el bastón electrónico de uso médico no invasivo contempla los requerimientos planteados en los objetivos generales y específicos. Su propósito es mejorar el proceso de rehabilitación motora de la marcha y reducir el tiempo de recuperación del paciente, brindando posibilidades de volver a caminar. Además, permite monitorear en tiempo real la rehabilitación a través del teléfono móvil mediante comunicación Bluetooth. Asimismo, la información relevante, como la fecha, hora, peso soportado y ángulos de inclinación del bastón por cada pisada, se almacena en la memoria microSD.

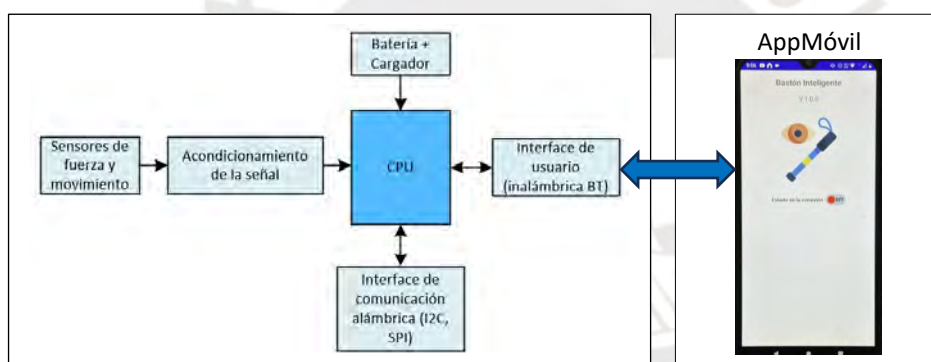


Figura 3-1. Diagrama de bloques general del sistema propuesto del Hardware y Software de App (Fuente: elaboración propia).

Este diagrama de bloques contiene los elementos necesarios para el objetivo propuesto de apoyar en la rehabilitación del paciente. Se describe brevemente cada una de ellas:

**CPU**, se encarga de procesar la información proveniente de los sensores, reloj en tiempo real, y demás componentes según una lógica de control grabada en la memoria de programa. Además, interactúa con el usuario por medio del Bluetooth y un teléfono móvil.

**Sensores de fuerza y movimiento**, capturan las variables físicas como presión ejercida sobre el bastón y acelerómetro, para percibir una modificación de su orientación o de su posición del bastón.

**Acondicionamiento de la señal**, acondiciona los datos analógicos capturados por los sensores.

**Alimentación**, batería y cargador.

**Interfaz de comunicación alámbrica**, se utiliza para el envío/recepción de información desde o hacia el CPU por medio de los buses I2C y SPI.

**Interfaz de comunicación inalámbrica**, se emplea el Bluetooth para comunicarse con la AppMóvil previamente instalada en el celular con Sistema Operativa Android.

### 3.1.1. Requisitos principales del diseño propuesto

Diseñar un bastón electrónico de rehabilitación implica considerar varios aspectos para garantizar su eficacia y seguridad. A continuación, se presentan algunos requisitos principales que se aplican en cualquier diseño electrónico y que podrían ser aplicados al bastón de manera general y para el sensor de fuerza que se describe en el ítem 3.2:

#### 1. Durabilidad y ergonomía:

- Materiales resistentes y ligeros: Para asegurar durabilidad y comodidad.
- Empuñadura ergonómica: Diseño que se adapte cómodamente a la mano del usuario.

#### 2. Ajustabilidad y Personalización:

- Longitud ajustable: Para adaptarse a la altura del usuario.
- Configuración personalizada: Permitir ajustes según las preferencias y necesidades individuales.

#### 3. Batería y Autonomía:

- Batería recargable: Para facilitar el uso continuo.
- Tiempo de autonomía: Garantizar un mínimo de 10 horas la disponibilidad del bastón.

#### 4. Software y Aplicaciones Asociadas:

- Software de análisis de datos: Para realizar un seguimiento del progreso del usuario en la rehabilitación.
- Aplicaciones móviles: Para controlar y personalizar las configuraciones del bastón.

#### 5. Conectividad y Comunicación:

- Bluetooth u otras tecnologías inalámbricas: Para la comunicación con dispositivos externos como smartphones o tabletas, permitiendo la configuración y recopilación de datos.
- Interfaz de usuario intuitiva: Botones, controles táctiles y de fácil interacción.

## 6. Confiabilidad:

- Circuito electrónico: Debe ser confiable en su funcionamiento. Es decir, que cumpla las funciones para lo cual fue diseñado.
- Circuito impreso: Debe tener un buen plano de tierra y el montaje de los componentes deben estar bien soldados para evitar posibles ruidos y así evitar que la captura y transmisión de datos estén errados.

## 7. Facilidad de Mantenimiento:

- Diseño modular: Para facilitar reparaciones y actualizaciones.
- Actualizaciones de firmware: Para mejorar la funcionalidad con el tiempo.

## 8. Peso de bastón y circuito electrónico:

El peso del conjunto (bastón más tarjeta electrónica) debe ser lo mínimo posible, de tal manera que no afecte el uso del bastón y el paciente pueda desplazarse sin problemas.

## 3.1.2. Diagrama de bloques específico del diseño propuesto

La Figura 3-2, muestra con más detalle el diseño propuesto y a partir del cual, se describirá cada uno de los componentes seleccionados para cumplir con los principales requisitos planteados anteriormente.

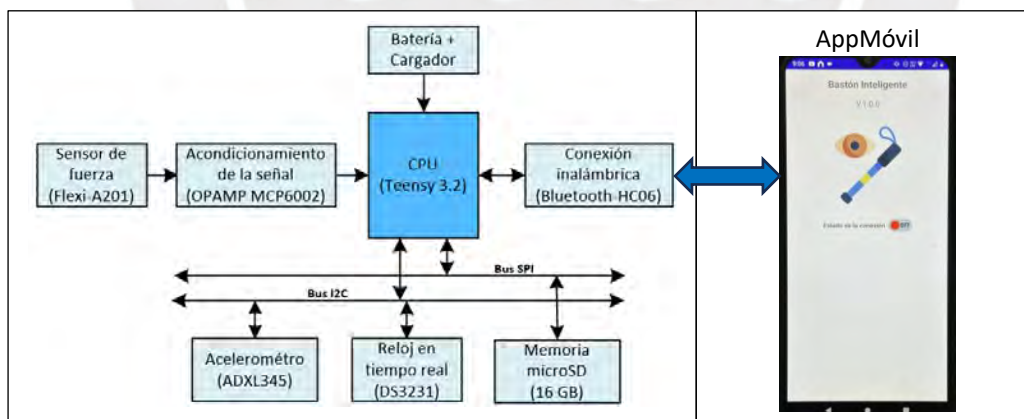


Figura 3-2. Diagrama detallado del diseño propuesto (Fuente: elaboración propia).

Todos los bloques de componentes mostrados en la Figura 3-2 van montados en la caña del bastón, y la comunicación con el AppMóvil del celular smartphone se realiza a través de la comunicación inalámbrica (Bluetooth).

### 3.1.3. Selección del Microcontrolador o CPU

La selección del microcontrolador es un factor importante en el diseño de sistemas de automatización o embebidos [49]. En función a la aplicación que va a tener, los principales criterios a considerar son los siguientes:

1. Arquitectura del Microcontrolador: Puede ser ARM, AVR, PIC, etc. La elección dependerá de la familiaridad, la potencia de procesamiento requerida y la capacidad de la memoria de programa.
2. Requisitos de tiempo real: El proyecto del bastón contempla el procesamiento en tiempo real, por lo tanto, debe tener la velocidad necesaria para cumplir con este requisito.
3. Periféricos Integrados:
  - Puertos de Entrada/Salida: debe tener suficientes pines digitales para señalización.
  - Convertidores Analógico-Digital (ADC): Importante para tomar las muestras de los sensores de fuerza y movimiento.
  - Interfaces de Comunicación: UART, SPI, I2C.
4. Memoria:
  - Flash: Suficiente para almacenar el programa.
  - RAM: Para almacenar datos temporales.
  - EEPROM: Para almacenamiento no volátil.
5. Consumo de Energía: Debe ser de bajo consumo para aplicaciones alimentadas por batería.
6. Facilidad de Desarrollo:
  - Herramientas de Desarrollo: Disponibilidad de compiladores, depuradores, IDE, etc.
  - Soporte de Comunidad: Una comunidad activa puede ser útil para resolver problemas.
7. Costo: El costo del microcontrolador y de su módulo, debe ser lo menor posible para no encarecer el presupuesto del proyecto.
8. Tamaño del Encapsulado y de la tarjeta:
  - Tamaño Físico: El tamaño del microcontrolador debe ser lo más pequeño posible.
  - Tarjeta o módulo: Debe tener dimensiones lo más cercanas a la caña del bastón.
9. Temperatura de Operación: Debe operar dentro del rango de temperatura ambiente promedio (-40°C, 80°C).

Para este proyecto, se considerará a fabricantes de microcontroladores más conocidos como Atmel, Microchip y Freescale, para evaluar dos criterios críticos para la aplicación. En primer lugar, se requiere que la capacidad de la memoria de programa sea superior a los 45 KB, que es el tamaño del código fuente. En otras palabras, se busca un microcontrolador con una capacidad de memoria de programa mínima de 64 KB (45 KB + 20% para posible ampliación). En segundo lugar, se evalúa el voltaje de alimentación. En este caso, se necesita un microcontrolador que opere con 3.3V, ya que se alimentará con una batería.

Se presenta a continuación un cuadro comparativo de microcontroladores de diversos fabricantes, con el objetivo de seleccionar el más adecuado para el proyecto.

Tabla 3-1. Cuadro comparativo de principales características de microcontroladores.

(Fuente: elaboración propia).

Características	Atmel ATMega328P	Atmel Atmega2560	Microchip PIC18F4550	Freescale ARM MK20DX256VLH7
<b>Voltaje de trabajo</b>	1.8 a 5.5V	1.8 a 5.5V	2 a 5.5V	1.71 a 3.6 v
<b>Voltaje de alimentación</b>	4.5 a 5.5 V	4.5 a 5.5 V	4.2 a 5.5V	1.71 a 3.6 v
<b>Frecuencia de reloj</b>	16 MHz	16 MHz	20 MHz	72 MHz
<b>Pines E/S</b>	14	54	35	32
<b>ADC</b>	10 bits	10 bits	10 bits	8 a 13 bits
<b>Memoria Flash</b>	32 KB	256 KB	32 KB	256 KB
<b>Memoria RAM</b>	2 KB	8 KB	2 KB	64 KB
<b>Memoria EEPROM</b>	1 KB	4 KB	256 B	2 KB
<b>Comunicación</b>	EAUSART, I2C, SPI	EAUSART, I2C, SPI	EAUSART, I2C, SPI	EAUSART, I2C, SPI

De la tabla, se tienen dos alternativas que cumplirían con el tamaño de la memoria de programa. El Atmega2560, con un costo<sup>16</sup> de \$20.18 la unidad, y el MK20DX256VLH7, a un costo<sup>17</sup> de \$12.36 la unidad. Sin embargo, en cuanto al voltaje de alimentación, sólo cumple el MK20DX256VLH7. Por esta razón, este será el microcontrolador seleccionado para desarrollar el proyecto cuyas características técnicas se detallan en la sección 2.6.1. Además, es de menor costo y tiene la ventaja de que se puede programar utilizando el entorno de Arduino.

<sup>16</sup> <https://www.microchip.com/en-us/product/atmega2560#purchase-from-store>

<sup>17</sup> <https://www.mouser.com/ProductDetail/NXP-Semiconductors/MK20DX256VLH7?qs=b%252B6ZnhyoAL%2FVf1Ewi0iu9Q%3D%3D>

### 3.2. Diseño del hardware de adquisición de datos del sensor de fuerza

Al elegir un sensor de fuerza para un bastón de rehabilitación, es importante considerar varios criterios para garantizar su adecuación y eficacia en la aplicación. Aquí hay algunos criterios básicos que podrían ser importantes [50]:

1. Rango de medición de fuerza: El sensor debe ser capaz de medir la gama de fuerzas que se esperan durante la rehabilitación con el bastón. Debe poder detectar tanto fuerzas leves como moderadas, dependiendo de las necesidades del usuario y del tipo de rehabilitación.
2. Sensibilidad y precisión: Es crucial que el sensor tenga una sensibilidad suficiente para detectar incluso pequeñas variaciones en la fuerza aplicada al bastón. La precisión en la medición es fundamental para garantizar que la información recopilada en cada pisada sea confiable y útil para los profesionales de la salud que supervisan la rehabilitación. Así mismo, para asegurar que el usuario reciba retroalimentación precisa sobre la carga que está aplicando al bastón.
3. Tamaño y forma: El sensor debe ser lo suficientemente compacto y liviano para integrarse de manera efectiva en el regatón del bastón sin agregar volumen o peso excesivos.
4. Durabilidad y resistencia: Dado que el bastón de rehabilitación será utilizado de manera frecuente y en diferentes condiciones, el sensor debe ser duradero y resistente al desgaste. Debe poder soportar impactos, presión repetida y posibles exposiciones a la humedad o al polvo.
5. Compatibilidad con la interfaz de usuario: El sensor debe ser compatible con la interfaz de usuario que se utilizará para visualizar y analizar los datos de fuerza recopilados durante la rehabilitación. Esto podría incluir interfaces de software en computadoras, aplicaciones móviles u otros dispositivos de visualización.
6. Facilidad de integración: Es deseable que el sensor sea fácil de integrar en el diseño del bastón y de conectar con otros componentes electrónicos que puedan formar parte del sistema de monitoreo y registro de datos.
7. Control de Offset, Zero o null: Es el valor de la salida para entrada cero.

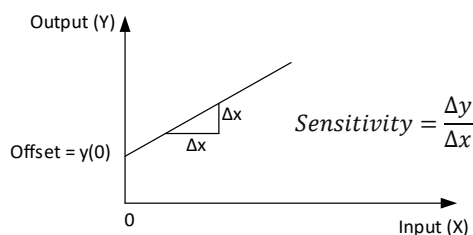


Figura 3-3. Control de Offset del sensor de fuerza (Fuente: elaboración propia)

8. Consumo de energía eficiente: Si el sensor requiere alimentación eléctrica, es importante que tenga un consumo de energía eficiente para prolongar la duración de la batería y minimizar la necesidad de recargas frecuentes.
9. Costo accesible: Para su viabilidad comercial y accesibilidad para los usuarios, el sensor debe ser asequible en términos de costos de fabricación y producción.
10. Facilidad de instalación y mantenimiento: Idealmente, el sensor debería ser fácil de instalar en el regatón del bastón y requerir poco o ningún mantenimiento adicional una vez en su lugar.

Al considerar estos requisitos, se puede seleccionar un sensor de fuerza adecuado que satisfaga las necesidades específicas de un bastón inteligente para rehabilitación.

Existen tres fabricantes de sensores que vamos a comparar:

Tabla 3-2. Cuadro comparativo de principales características de sensores de fuerza.

(Fuente: elaboración propia).

Características	Flexiforce A201	Interlink FSR	Sensitronics Flexiforce
<b>Fabricante</b>	Tekscan	Interlink Electronics	Sensitronics
<b>Rango de fuerza</b>	1 – 100 lbs	1 – 50 lbs	1 – 100 lbs
<b>Principio de funcionamiento</b>	Varía su resistencia eléctrica cuando se aplica presión sobre ella	Varía su resistencia eléctrica cuando se aplica presión sobre ella	Varía su resistencia eléctrica cuando se aplica presión sobre ella
<b>Área sensible (diámetro)</b>	9.53mm	5.08mm (FSR400) 12.7mm (FSR402)	14.53mm
<b>Error de Linealidad</b>	≤ 3%	≥ 10%	≥ 10%
<b>Tamaño</b>	encapsulado pequeño LGA (Land Grid Array).	encapsulado pequeño LGA.	encapsulado pequeño LGA.
<b>Precisión</b>	Típica del 5%	10% al 15% del rango total de medición.	5% al 10% del rango total de medición
<b>Consumo de energía</b>	Ningún fabricante precisa, pero dependerá si se utiliza divisor de voltaje o amplificador operacional. En general, su consumo es bajo.		
<b>Costo aproximado</b>	\$5 y \$15 USD	\$3 y \$10 USD	\$1 a \$5 USD

Analizando los tres fabricantes de sensores, y considerando cómo parámetros muy importantes el tamaño (diámetro) de la caña y regatón del bastón, el error de linealidad y la precisión. El sensor del fabricante Interlink<sup>18</sup> que entraría dentro del regatón al ser de 5.08 mm es pequeño y podría no tener sensibilidad cuando el bastón tocara el piso de manera inclinada y sería necesario colocar uno o dos 2 sensores adicionales, lo cual dificultaba su ubicación. El sensor del fabricante Sensitronics<sup>19</sup> es muy ancho y había que agrandar el diámetro del regatón, lo que hacía imposible optar por esta alternativa. Con

<sup>18</sup> <https://www.interlinkelectronics.com/fsr-402>

<sup>19</sup> [www.sensitronics.com/pdf/Sensitronics\\_FSR\\_101.pdf](http://www.sensitronics.com/pdf/Sensitronics_FSR_101.pdf)

estas consideraciones, se seleccionó al sensor Flexiforce<sup>20</sup> A201 para utilizarlo en el bastón de este proyecto.

### 3.2.1. Sensor de fuerza FlexiForce A201

El circuito de adquisición de datos consta del sensor FlexiForce, un amplificador operacional MCP6002, así como componentes resistivos y capacitivos, como se muestra a continuación:

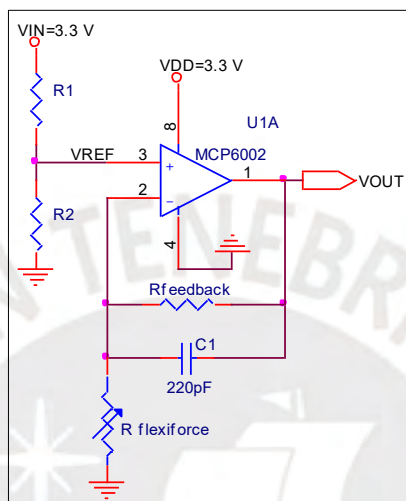


Figura 3-4. Circuito para obtener el peso en Vout [29] (Fuente: adaptación propia).

La ecuación del voltaje de salida Vout es:

$$VOUT = VREF * (1 + (RFEEDBACK / RFLEXIFORCE)) \quad \dots [29]$$

En la sección 2.4.2, se describió las ventajas de este sensor. A continuación, se muestra las curvas de transferencia para diferentes valores de la resistencia Rfeedback y Vref.

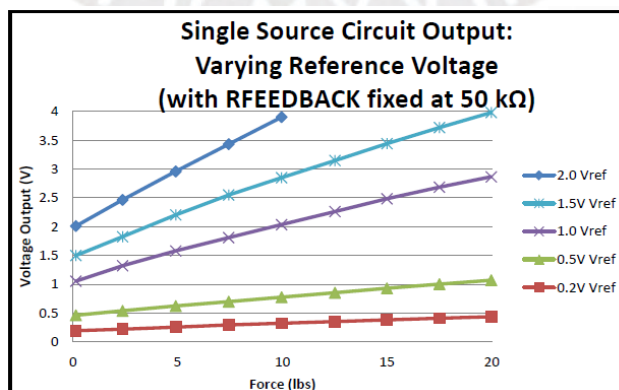


Figura 3-5. Curvas de transferencia entre fuerza, voltaje con Rfeedback a 50KΩ [29].

<sup>20</sup> <https://www.tekscan.com/resources/datasheets-guides/flexiforce-a201-datasheet>

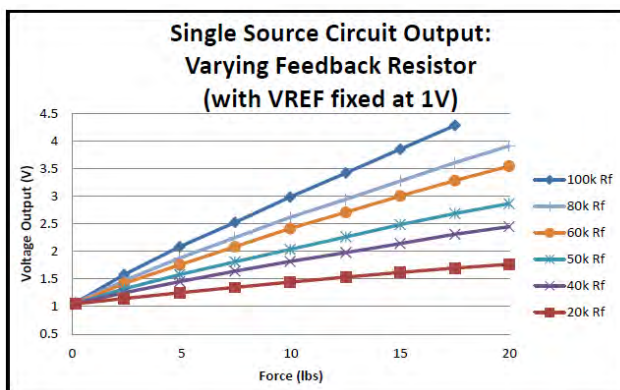


Figura 3-6. Curvas de transferencia entre fuerza, voltaje con Vref a 1V [29].

Tomando como referencia estas curvas de transferencia, se muestra el circuito entre CPU y sensor con los valores siguientes: Rfeedback = 50KΩ y Vref = 1.2 V.

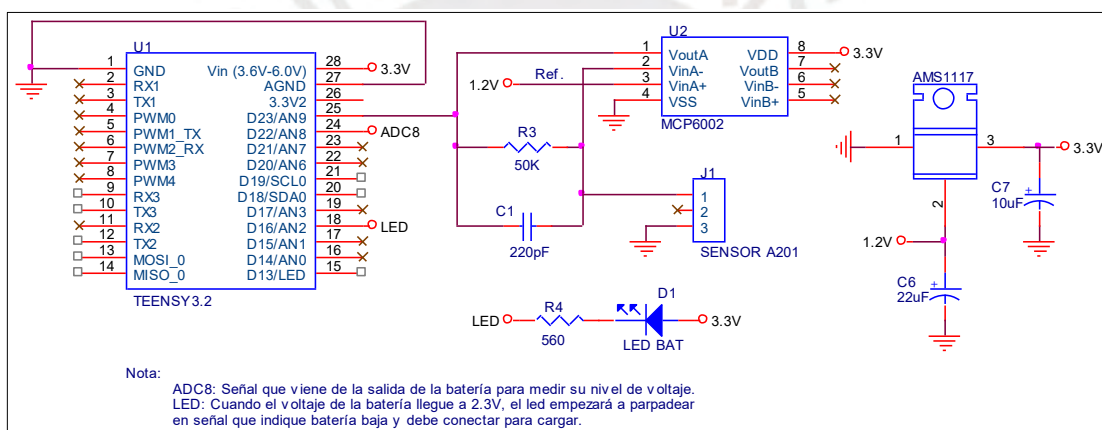


Figura 3-7. Circuito de adquisición de datos del sensor de fuerza FlexiForce (Fuente: elaboración propia).

Inicialmente, en el desarrollo de la presente tesis, se estaba utilizando un divisor de voltaje para obtener 1 voltio como referencia del Sensor de fuerza, pero cuando el voltaje de la batería de litio bajaba, el voltaje de referencia también bajaba y hacía que las lecturas de la fuerza aplicada en el bastón no concuerden con la curva característica del sensor.

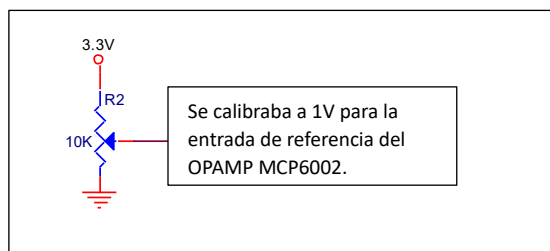


Figura 3-8. Voltaje de referencia inicial del sensor de fuerza (Fuente: elaboración propia).

Ante este problema (ver Figura 3-4), se resuelve usando el C.I. AMS1117<sup>21</sup> (ver Figura 3-7), que es un regulador de voltaje lineal que acepta voltajes de entrada entre [2.0V – 8.0V] y da como salida 1.2V fijo, con una corriente entre [10mA – 1A máx]. Este se utiliza como voltaje de referencia para el OPAMP en reemplazo del divisor de voltaje.

La Figura 3-5 y Figura 3-6 muestran las curvas de transferencia dada por el fabricante. En este proyecto, al fijar como voltaje de referencia 1.2V, era necesario encontrar la nueva curva de transferencia con el circuito de la Figura 3-7. Para lograrlo, se utilizó 3 pesas de 5 Kg c/u, de propiedad del Laboratorio de Mecánica y Electrónica de la PUCP, más un valde plástico (0.360 Kg) y un conjunto de monedas (0.640 Kg) para redondear 1Kg.

La hoja de datos del fabricante menciona que el error de linealidad del sensor FlexiForce A201 de 25 libras, con escala completa (full scale) de 25 libras, es  $\leq \pm 3\%$ . Las pruebas para evaluar las respuestas del sensor dentro del bastón, así como su error, se realizan en el capítulo 4, sección 4.1.1.



Figura 3-9. Tarjeta principal del bastón con el sensor de fuerza.



Figura 3-10. Fotos de los pesos utilizadas para la obtención de la curva del sensor de fuerza A201.

A continuación, se muestran el valde con todas las combinaciones de pesos (6, 11, 16) Kg para obtener la curva de transferencia, así como, la estructura con barras de aluminio, donde en una de ellas se colocó el sensor y en la otra se colgó el valde con los pesos mostrados en la Figura 3-11.

<sup>21</sup> <https://www.alldatasheet.es/datasheet-pdf/pdf/457818/AMS/AMS1117-1.2.html>

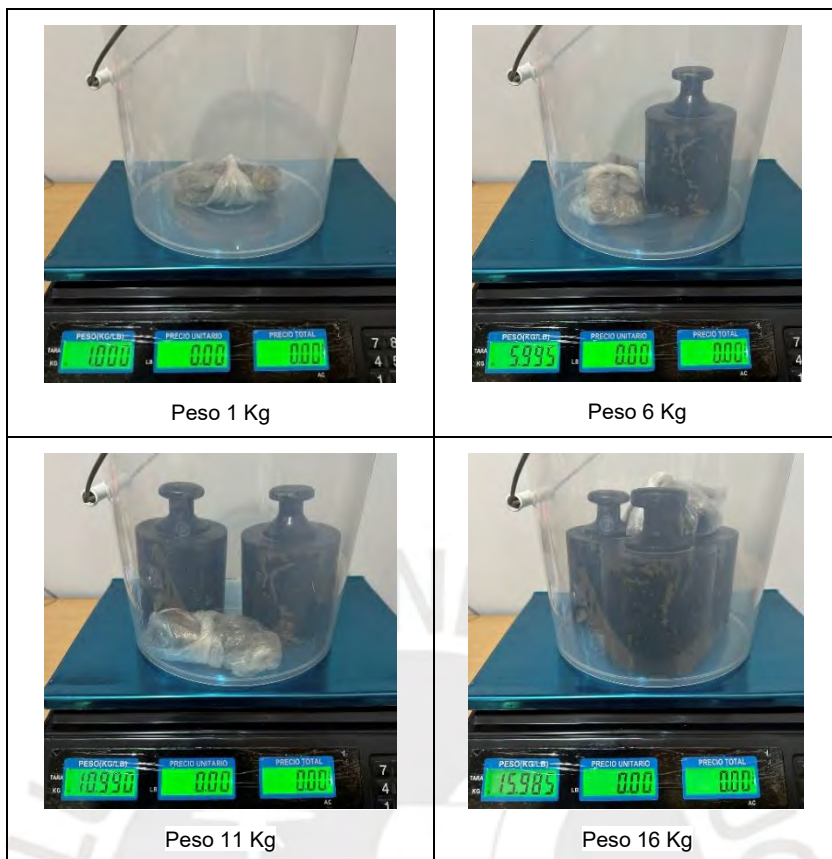


Figura 3-11. Combinación de pesas para hallar la curva de transferencia. (Fuente: Elaboración propia)

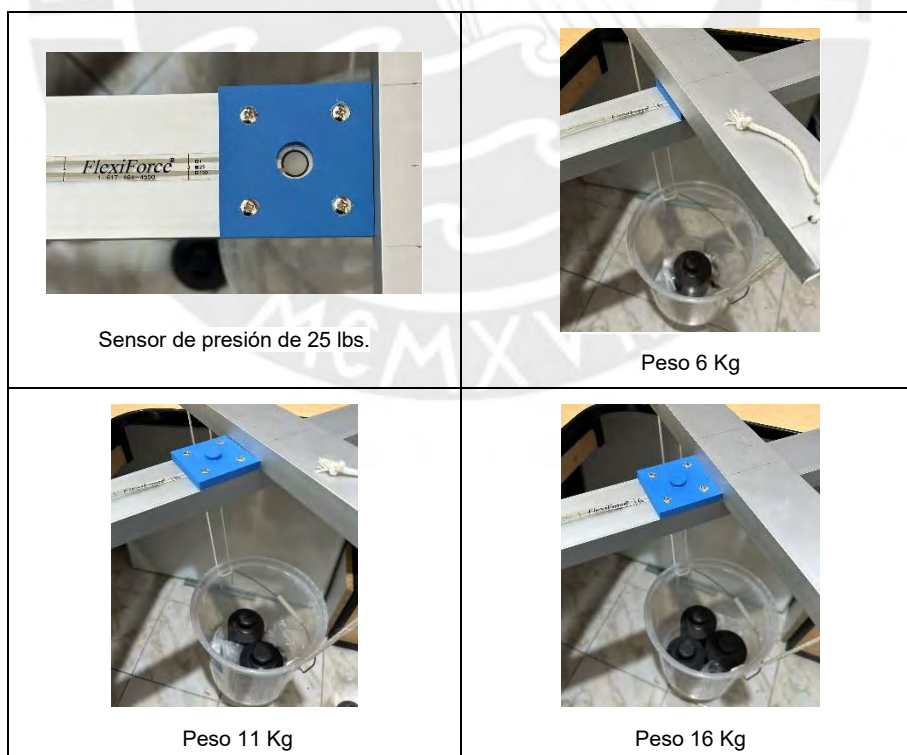


Figura 3-12. Estructura de aluminio con las pesas. (Fuente: Elaboración propia)

El procedimiento consistió en tomar unas 10 lecturas digitales del sensor por medio del ADC para cada medida de peso (0, 6, 11, 16) Kg. a intervalos de 3 segundos. Posteriormente, se calculó el promedio de las muestras de cada medida y se procedió a trazar la nueva curva del sensor. Finalmente, se obtuvo la ecuación respectiva del voltaje vs. peso.

Tabla 3-3. Valores obtenidos de voltajes por cada medida realizada.

(Fuente: elaboración propia).

Peso (Kg)	Peso (Lbs)	Voltaje leído (voltios)	Voltaje promedio (Voltios)	Peso (Kg)	Peso (Lbs)	Voltaje leído (voltios)	Voltaje promedio (Voltios)
0	0.000	1.210	1.213	11	24.251	1.850	1.852
		1.214				1.851	
		1.213				1.850	
		1.212				1.849	
		1.214				1.851	
		1.215				1.853	
		1.214				1.853	
		1.213				1.853	
		1.278				1.854	
		1.213				1.854	
6	13.227	1.625	1.625	16	35.274	1.968	1.968
		1.626				1.972	
		1.623				1.970	
		1.627				1.970	
		1.626				1.970	
		1.625				1.970	
		1.628				1.967	
		1.625				1.967	
		1.628				1.966	
		1.623				1.962	

Tabla 3-4. Valores de peso (lb) vs. voltaje (voltios).

Peso (Lbs)	Voltaje (Voltios)
0	1.213
13.227	1.625
24.251	1.852
35.274	1.968

En la Tabla 3-4 se utiliza el peso en libras para la obtención de la curva porque el sensor está calibrado en libras.

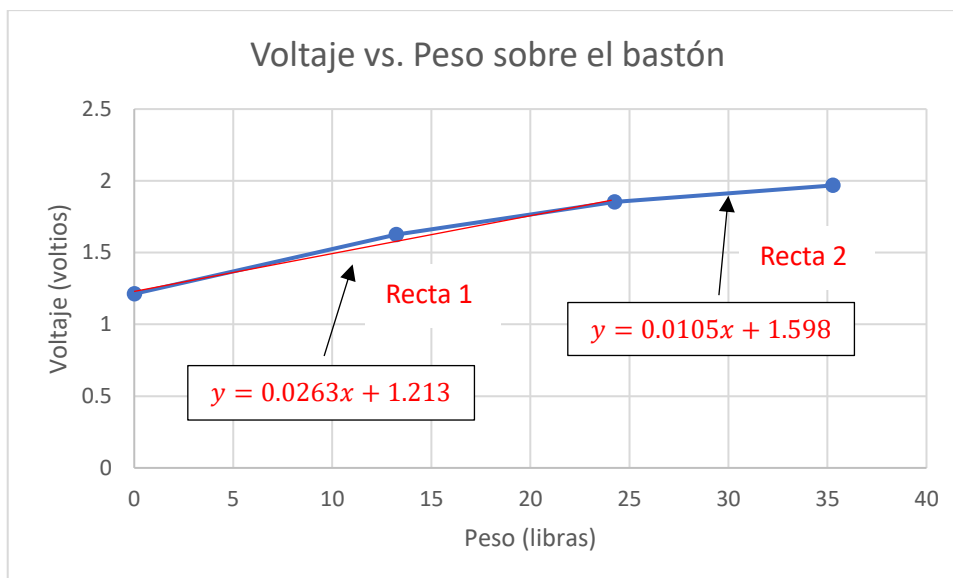


Figura 3-13. Curva característica del sensor, basado en la tabla 3-4 (Fuente: elaboración propia).

De la Figura 3-13 se va a obtener la ecuación para la recta 1 y recta 2, es decir, en los intervalos  $[0, 24.251>$  y  $[24.251, 35.274]$ , ecuaciones que servirán para determinar la fuerza aplicada sobre el bastón al momento de caminar.

Cabe precisar que el sensor utilizado es de 25 libras y la ecuación de la recta 1 responde adecuadamente a la fuerza aplicada en el rango de 0 a 25 libras. Por consiguiente, para fuerzas mayores a 25 libras se utiliza la recta 2.

#### Recta 1:

Se toma los puntos  $(x_0, y_0) = (0, 1.213)$  y  $(x_1, y_1) = (24.251, 1.852)$  para hallar su ecuación:

$$y - y_1 = m. (x - x_1)$$

$$(y - 1.852) = \left[ \frac{1.852 - 1.213}{24.251} \right] (x - 24.251) \implies y = 0.0263x + 1.213$$

$$\rightarrow x = (y - 1.213) / 0.0263 \quad \dots (3.1)$$

#### Recta 2:

Se toma los puntos  $(x_0, y_0) = (24.251, 1.852)$  y  $(x_1, y_1) = (35.274, 1.968)$  para hallar su ecuación:

$$y - y_1 = m. (x - x_1)$$

$$(y - 1.968) = \left[ \frac{1.968 - 1.852}{35.274 - 24.251} \right] (x - 35.274) \implies y = 0.0105x + 1.598$$

$$\rightarrow x = (y - 1.598) / 0.0105 \quad \dots (3.2)$$

Donde:

x = Valor proveniente del sensor de fuerza expresado en libras.

y = Valor proveniente del ADC expresado en voltios.

*Las ecuaciones (3.1) y (3.2) se utilizarán en el software de control del microcontrolador para convertir el voltaje leído del ADC, en la fuerza aplicada al bastón en libras y luego transformada a Kilos, para su posterior proceso.*

### 3.3. Diseño de la interfaz de comunicación alámbrica e inalámbrica (I2C, SPI, BT)

Para el diseño de la interfaz de comunicación alámbrica (I2C, SPI) e inalámbrica (Bluetooth), se debe tener en cuenta lo descrito en el capítulo 2, referente a las tecnologías inalámbrica (sección 2.5.1) y alámbrica (secciones 2.6.3 y 2.6.4). A continuación, se muestra los cuadros comparativos entre posibles dispositivos electrónicos que podrían utilizarse en el desarrollo del hardware del proyecto.

#### 3.3.1. Interfaz alámbrica con Acelerómetro – I2C

La selección del acelerómetro es un factor importante en el diseño del bastón porque será quien brinde la información sobre el movimiento del bastón midiendo los ángulos Pitch y Roll. En función a la aplicación que va a tener, los principales criterios a considerar son los mostrados en la siguiente tabla:

*Tabla 3-5. Cuadro comparativo de principales características de acelerómetros.*

*(Fuente: elaboración propia).*

Características	ADXL345	BMA180	LIS3DH
<b>Fabricante</b>	Analog Devices	Bosch	STMicroelectronics
<b>Rango de medición (seleccionable por el usuario)</b>	±2 g, ±4 g, ±8 g o ±16 g,	±1 g, ±1.5 g, ±2 g, ±4 g, ±8 g o ±16 g,	±2 g, ±4 g, ±8 g o ±16 g,
<b>Interfaz de comunicación</b>	comunicación digital I2C y SPI.	comunicación digital I2C y SPI.	comunicación digital I2C y SPI.
<b>Consumo de energía</b>	relativamente bajo.	moderado.	relativamente bajo.
<b>Resolución</b>	hasta 13 bits.	14 bits	hasta 14 bits.
<b>Tamaño</b>	encapsulado pequeño LGA (Land Grid Array).	encapsulado pequeño LGA.	encapsulado pequeño LGA.
<b>Aplicaciones típicas</b>	dispositivos portátiles, sistemas de navegación, juegos y monitoreo de actividad física.	aplicaciones industriales y de consumo, como la detección de impactos y la monitorización de la vibración.	dispositivos portátiles, teléfonos inteligentes, sistemas de navegación
<b>Costo aproximado</b>	\$5 a \$15 USD	\$3 a \$10 USD	\$1 a \$10 USD
<b>Disponibilidad Perú</b>	SI	NO	NO

De la Tabla 3-5, se tienen dos alternativas que cumplirían con el bajo consumo de energía, el ADXL345 y el LIS3DH, ambos con un costo y resolución similar. Sin embargo, en cuanto a la disponibilidad en el mercado local ante cualquier eventualidad, sólo cumple el ADXL345. Por esta razón, este será el acelerómetro seleccionado para desarrollar el proyecto.

### 3.3.1.1. Funcionamiento del Acelerómetro ADXL345 – I2C

Cómo se describió en la sección 2.6.3, I2C es un protocolo de comunicación que especifica cómo los datos deben ser enviados y recibidos a través del bus. Esto incluye la forma en que se inician y detienen las transmisiones de datos, la dirección de los dispositivos, y otros aspectos detallados de la comunicación. En este contexto, el bus utiliza dos líneas principales, SDA (Serial Data) y SCL (Serial Clock), para facilitar la comunicación entre múltiples dispositivos.

A continuación, se muestra dos tablas del dispositivo ADXL345 relativas al consumo de corriente y velocidad de transmisión de datos o frecuencia.

Tabla 3-6. Consumo de Corriente vs. Velocidad de datos [26]

( $T_A = 25^\circ\text{C}$ ,  $V_S = 2.5\text{ V}$ ,  $V_{DDIO} = 1.8\text{ V}$ )

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	$I_{DD}$ ( $\mu\text{A}$ )
3200	1600	1111	140
1600	800	1110	90
800	400	1101	140
400	200	1100	140
200	100	1011	140
100	50	1010	140
50	25	1001	90
25	12.5	1000	60
12.5	6.25	0111	50
6.25	3.13	0110	45
3.13	1.56	0101	40
1.56	0.78	0100	34
0.78	0.39	0011	23
0.39	0.20	0010	23
0.20	0.10	0001	23
0.10	0.05	0000	23

Tabla 3-7. Consumo de Corriente vs. Velocidad de datos en bajo consumo [26].

Low Power Mode ( $T_A = 25^\circ\text{C}$ ,  $V_S = 2.5\text{ V}$ ,  $V_{DDIO} = 1.8\text{ V}$ )

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	$I_{DD}$ ( $\mu\text{A}$ )
400	200	1100	90
200	100	1011	60
100	50	1010	50
50	25	1001	45
25	12.5	1000	40
12.5	6.25	0111	34

Se observa en la Tabla 3-6, que en el dispositivo ADXL345, su consumo de energía se ajusta automáticamente en proporción a la velocidad de datos de salida, es decir, que la tasa de muestreo seleccionada debe ser adecuada para la aplicación. Para obtener ahorros de energía adicionales, se recomienda un modo de menor consumo de energía, como se

indica en la Tabla 3-7. En este modo, la frecuencia de muestreo interna se reduce, permitiendo ahorros de energía en la velocidad de datos de 400 Hz a 12,5 Hz, aunque esto va acompañado de un ligero aumento en el nivel de ruido.

Para el desarrollo de este proyecto de tesis en cuanto a consumo se refiere, se ha considerado trabajar el acelerómetro en modo normal a la frecuencia de 50 Hz con un consumo promedio de 90uA. Se selecciona esta frecuencia debido a que una persona normal (sin ninguna lesión) puede dar entre 100 a 130 pasos por minuto [51], generando un intervalo de tiempo entre pasos de 0.6 seg (100 pasos) o 0.46 seg (130 pasos).

$$\text{Tiempo entre pasos} = \frac{\text{tiempo caminata}}{\text{cadencia}} = \frac{60 \text{ seg}}{100} = 0.6 \text{ seg}$$

$$\Rightarrow \text{Frec} = 1.66 \text{ Hz}$$

$$\text{Tiempo entre pasos} = \frac{\text{tiempo caminata}}{\text{cadencia}} = \frac{60 \text{ seg}}{130} = 0.46 \text{ seg}$$

$$\Rightarrow \text{Frec} = 2.17 \text{ Hz}$$

Por consiguiente, para una persona con algún problema en una de sus piernas, el tiempo que tardaría con bastón entre paso y paso será mayor que el de una persona que no usa ayudas para caminar, debido a la necesidad de estabilidad y apoyo adicional. Sin embargo, este tiempo puede variar dependiendo de factores individuales como la razón para usar el bastón (por ejemplo, recuperación de una lesión, debilidad general, equilibrio, etc.) y la experiencia del usuario con el bastón. Vamos a suponer para efectos referenciales que una persona con alguna lesión caminando con bastón tiene una cadencia entre 50 y 70 pasos por minuto, generará los siguientes tiempos:

$$\text{Tiempo entre pasos} = \frac{\text{tiempo caminata}}{\text{cadencia}} = \frac{60 \text{ seg}}{50} = 1.2 \text{ seg}$$

$$\Rightarrow \text{Frec} = 0.83 \text{ Hz}$$

$$\text{Tiempo entre pasos} = \frac{\text{tiempo caminata}}{\text{cadencia}} = \frac{60 \text{ seg}}{70} = 0.85 \text{ seg}$$

$$\Rightarrow \text{Frec} = 1.17 \text{ Hz}$$

Con estos 4 resultados obtenidos en la frecuencia [0.83Hz, 2.17Hz], muestrear las aceleraciones en el ADXL345 a la frecuencia de 50 Hz es óptima y no se corre peligro de pérdida de datos como los grados Pitch y Roll.

A continuación, se mostrará algunos gráficos que ilustran el acelerómetro y cómo se lleva a cabo estos movimientos.

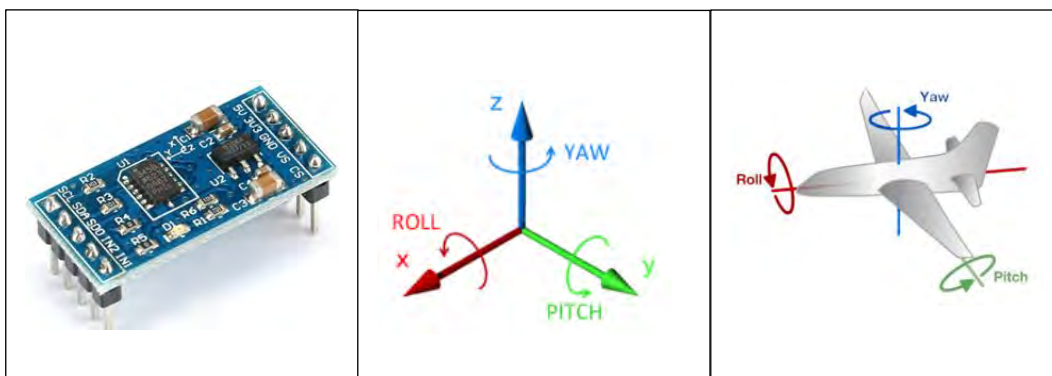


Figura 3-14. Movimientos Pitch y Roll del acelerómetro ADXL345 [26].

El ADXL345 es adecuado para medir la aceleración estática de la gravedad para aplicaciones de detección de inclinación, así como la aceleración dinámica ocasionada por algún movimiento o impacto. Su alta resolución (4 mg/LSB) permite medir cambios de inclinación de menos de  $1.0^\circ$ .

Para calcular los ángulos Pitch y Rolls se debe tener en cuenta los ejes de sensibilidad de la aceleración con respecto a los 3 ejes como se muestra en las Figura 3-15, Figura 3-16 y Figura 3-17.

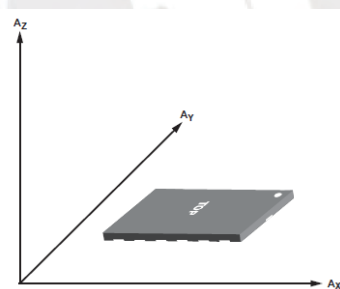


Figura 3-15. Ejes de sensibilidad de aceleración [26].

(El voltaje de salida correspondiente aumenta cuando se acelera a lo largo del eje sensible)

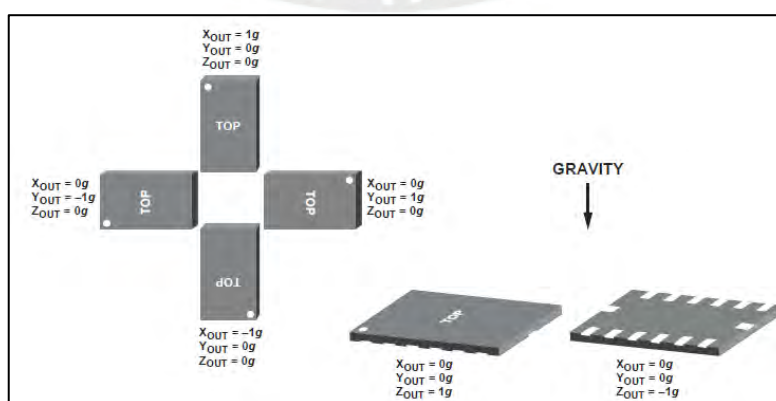


Figura 3-16. Respuesta de salida vs. orientación de la gravedad [26].

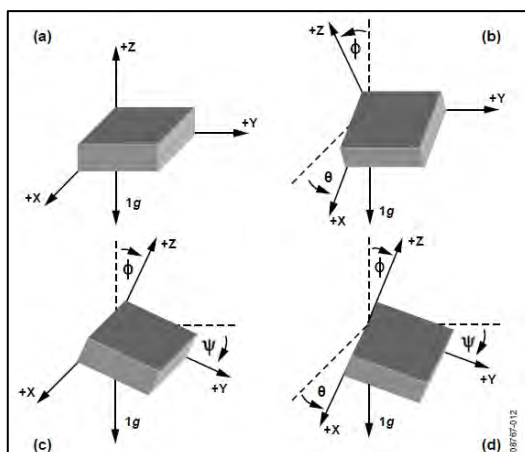


Figura 3-17. Ángulos para la detección de inclinación independiente [26].

Se puede utilizar la trigonometría básica para demostrar que los ángulos de inclinación se pueden calcular utilizando las siguientes ecuaciones [26]:

$$\theta = \tan^{-1} \left[ \frac{A_{X,OUT}}{\sqrt{A_{Y,OUT}^2 + A_{Z,OUT}^2}} \right]$$

$$\psi = \tan^{-1} \left[ \frac{A_{Y,OUT}}{\sqrt{A_{X,OUT}^2 + A_{Z,OUT}^2}} \right]$$

$$\phi = \tan^{-1} \left[ \frac{\sqrt{A_{X,OUT}^2 + A_{Y,OUT}^2}}{A_{Z,OUT}} \right]$$

### 3.3.2. Interfaz alámbrica con Reloj en tiempo real DS3231 – I2C

Para grabar en la memoria microSD la fecha y hora de cada pisada con el bastón, es esencial contar con un módulo de reloj en tiempo real (RTC) en nuestro circuito, estos módulos son comúnmente utilizados para mantener un seguimiento preciso del tiempo en proyectos electrónicos ya que cuentan con una batería que puede mantener la información durante casi 10 años en promedio.

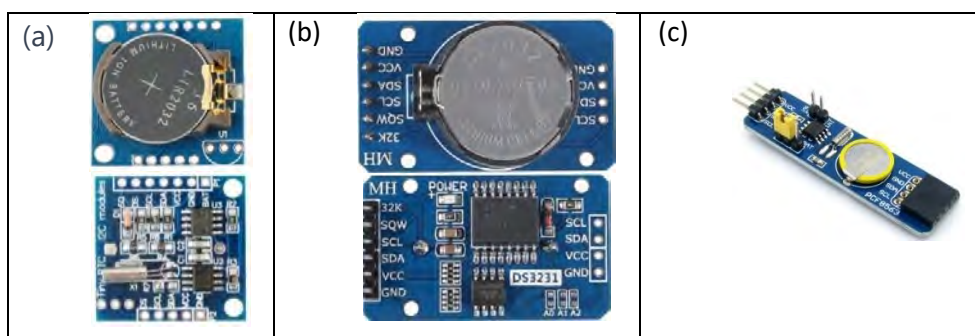


Figura 3-18. Módulos (a) DS1307, (b) DS3231 y (c) PCF8563.

En la Figura 3-18, se muestran las imágenes de los módulos RTC y a continuación, se muestra un cuadro comparativo para determinar cuál es el más apropiado para el proyecto.

Tabla 3-8. Cuadro comparativo de principales características de relojes en tiempo real.

Características	DS1307	DS3231	PCF8563
Fabricante	Maxim Integrated	Maxim Integrated	NXP
Precisión en el tiempo	±20 ppm a ±50 ppm (sin compensación térmica)	±2 ppm desde 0°C a +40°C, ±3.5 ppm desde -40°C a +85°C (con compensación térmica)	±20 ppm a 25°C (sin compensación térmica)
Interfaz de comunicación	I2C	I2C	I2C
Almacenamiento	56 bytes de RAM	236 bytes de RAM	No especifica
Consumo de energía	Normal: 1.5 mA Batería: 0.3 µA	Normal: 200 µA Batería: 2 µA	Normal: 50 µA Batería: 0.9 µA
Temperatura de operación	-40°C a +85°C	-40°C a +85°C	-40°C a +85°C
Alimentación	5V	2.3V a 5V	1.8V a 5.5V
Costo aproximado	\$1 a \$2 USD	\$1.5 a \$4 USD	\$0.75 a \$2 USD
Disponibilidad Perú	SI	SI	NO

Analizando la Tabla 3-8, y basándonos en sus hojas de datos [52], [53], [54] se descarta el DS1307 porque su alimentación es sólo con 5 V, en cuanto al consumo de energía el PCF8563 consume un poco menos respecto al DS3231, sin embargo, el DS3231 es el que tiene mejor precisión y está disponible en el mercado local ante cualquier eventualidad. Por estas razones, se selecciona al RTC DS3231 (Figura 3-18 (b)) para desarrollar el proyecto.

### 3.3.3. Interfaz alámbrica con módulo MicroSD – SPI

Para grabar los datos digitales como la fecha, hora, ángulos Pitch & Roll, y el peso, que se generan al utilizar el bastón de rehabilitación, es necesario e indispensable disponer de un módulo que contenga la tarjeta microSD. Con respecto a la tarjeta microSD, existen diferentes marcas, capacidades expresadas en GB y velocidad de transferencia de datos.

A continuación, se muestra las imágenes de los dos únicos modelos de módulos que existen y para seleccionar uno de ellos, se consideró el tamaño ya que no debía superar los 40mm (ancho de la tarjeta del circuito impreso), ya que ambos módulos podrían trabajar con el mismo voltaje 3.3V. Por ello, se decidió por el módulo con tamaño de 17.9 mm. Así mismo, se muestra los tipos de tarjetas de memoria SD.

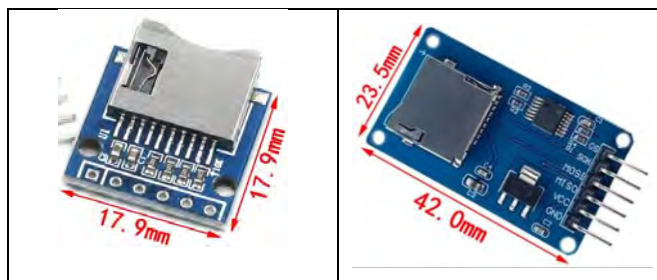


Figura 3-19. Modelos de módulos que contienen la tarjeta microSD.

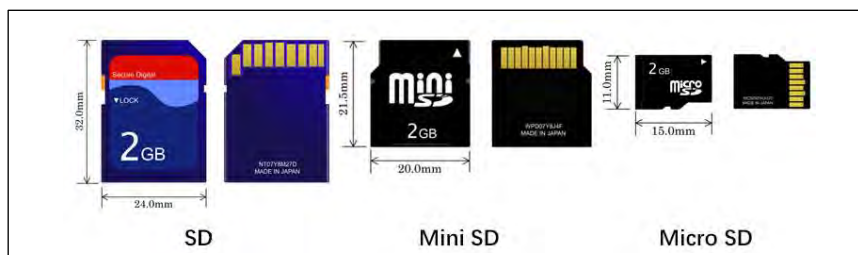


Figura 3-20. Modelos de tarjetas SD, Mini SD y Micro SD.

(Fuente: <https://www.reneelab.es/wp-content/uploads/sites/12/2019/06/SDcar27.jpg>)

Tabla 3-9. Cuadro comparativo de principales características de tarjetas microSD.

(Fuente: <https://www.reneelab.es/que-es-tarjeta-sd.html>)

Características	Kingston	SanDisk	Lexar
<b>Clases</b>	Clase 4 (4): 4 MB/s Clase 6 (6): 6 MB/s Clase 10 (10): 10 MB/s	Clase 4 (4): 4 MB/s Clase 6 (6): 6 MB/s Clase 10 (10): 10 MB/s	Clase 4 (4): 4 MB/s Clase 6 (6): 6 MB/s Clase 10 (10): 10 MB/s
<b>Capacidad</b>	Tarjeta SD: [8MB,2GB] Tarjeta SDHC: [2GB,32GB] Tarjeta SDXC: [32GB, 2TB]	Tarjeta SD: [8MB,2GB] Tarjeta SDHC: [2GB,32GB] Tarjeta SDXC: [32GB, 2TB]	Tarjeta SD: [8MB,2GB] Tarjeta SDHC: [2GB,32GB] Tarjeta SDXC: [32GB, 2TB]
<b>Vida útil de la tarjeta</b>	Garantía de por vida	5 años después de la compra	2 años después de la compra
<b>Costo aproximado</b>	\$ 6 USD	\$ 8 USD	\$ 14 USD
<b>Disponibilidad Perú</b>	SI	SI	SI

Analizando la Tabla 3-9, se observa que todos los fabricantes cumplen con los estándares establecidos para cada categoría. En consecuencia, la selección de uno de los tres fabricantes se basaría en el que ofrece mayor garantía, que en este caso sería Kingston. En cuanto a la capacidad, se va a utilizar una de 16 GB por la siguiente razón:

- 1) Las muestras se toman cada 1 seg aproximadamente.
- 2) Una fila de datos que se almacena en la memoria cada segundo ocupa 41 bytes aproximadamente y comprende lo siguiente:

**Fecha,Hora,Pitch(°),Roll(°),Peso(Kg)** → cabecera, se guarda una sola vez

10/4/2024,11:58:3,76.62,-1.24,5.68 → 41 bytes y se repite por cada muestra

→ En un minuto, se almacena:  $60 \times 41 \text{ B} = 2460 \text{ B}$

→ En una hora de caminata, se almacena:  $60 \times 2460 = 147,600 \text{ B} \cong 145 \text{ KB}$

→ Si usara 10 horas al día, se almacena:  $10 \times 145 \text{ KB} = 1450 \text{ KB} \cong 1.42 \text{ MB}$

→ En un mes de uso, se almacena:  $30 \times 1.42 \text{ MB} = 42.6 \text{ MB}$

→ En un año de uso por paciente (en el extremo), se almacena:

$$12 \times 42.6 \text{ MB} = 511.2 \text{ MB} = 0.5 \text{ GB}$$

Por lo tanto, con una tarjeta de 16 GB, se podría almacenar información durante 32 años teóricamente.

### 3.3.4. Interfaz inalámbrica con módulo Bluetooth

Uno de los objetivos de este proyecto es que toda la información del uso diario del bastón, almacenada en la tarjeta de memoria microSD, sea transferida a un dispositivo móvil para su análisis por parte del médico tratante o cuando se desee hacer un monitoreo en tiempo real con el dispositivo móvil, de manera que el paciente aprenda a usar mejor el bastón. Para ambos casos, es necesario utilizar un módulo Bluetooth para la comunicación inalámbrica entre el bastón y el aplicativo instalado en el dispositivo móvil. En tal sentido, a continuación se muestra un cuadro comparativo con las principales características del módulo Bluetooth versión 2.0 para determinar cuál es el más apropiado para el proyecto.

Cabe precisar, que en la Tabla 2-4 se mencionó las diferencias entre las diferentes versiones de la tecnología BT, de ellas, se selecciona la versión 2.0 porque es la más compatible entre las diferentes marcas y gamas de celulares por su fácil conexión para cualquier edad de usuario.

Tabla 3-10. Cuadro comparativo de principales características de módulos Bluetooth.

Características	HC-05 <sup>22</sup>	HC-06	RN-42 <sup>23</sup>
Fabricante	Jinan Huamao Technology Co., Ltd. Y otras empresas chinas	Jinan Huamao Technology Co., Ltd. Y otras empresas chinas	Microchip Technology
Tipo	Bluetooth Clásico v2.0	Bluetooth Clásico v2.0	Bluetooth Clásico v2.0
Modo	Maestro/Esclavo	Esclavo	Esclavo

<sup>22</sup> <https://docs.google.com/file/d/0B93S-Z6qmW9ER3hyRmNlazd4bVE/edit?resourcekey=0-W53BbEVeBm1WtThyrl1fIA>

<sup>23</sup>

[https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/bluetooth\\_cr\\_UG-v1.0r.pdf](https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/bluetooth_cr_UG-v1.0r.pdf)

<b>Consumo de energía</b>	Modo TX/RX: ~40 mA Modo reposo: ~1 mA	Modo TX/RX: ~40 mA Modo reposo: ~1 mA	Modo TX/RX: ~20 mA Modo reposo: ~0.3 mA
<b>Rango de comunicación</b>	Hasta 10 metros	Hasta 10 metros	Hasta 20 metros
<b>Costo aproximado</b>	\$7 USD	\$7 USD	\$20 USD
<b>Disponibilidad Perú</b>	SI	SI	NO

Analizando la tabla, se descarta el modelo HC-05 porque en el proyecto sólo se usa en modo esclavo. Entre los modelos HC-06 y RN-42, al tener características casi similares, se selecciona el HC-06 por su bajo costo y de fácil disponibilidad en el mercado local.

A continuación, con todos los componentes seleccionados se muestra el circuito esquemático con las conexiones necesarias para tomar los datos del acelerómetro, reloj en tiempo real y tarjeta microSD.

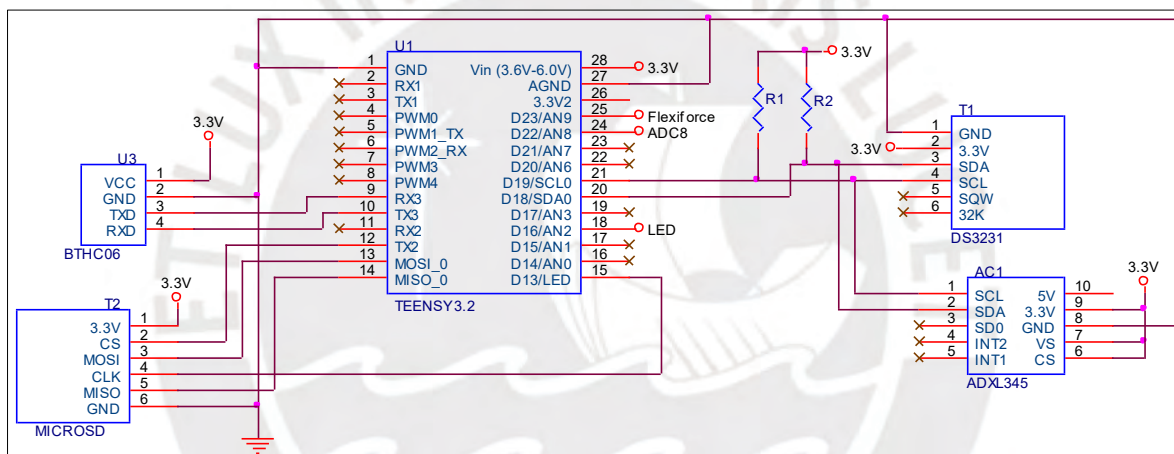


Figura 3-21. Esquemático del circuito de comunicación CPU, ADXL345, DS3231, MicroSD y BT.

(Fuente: elaboración propia)

### 3.4. Diseño del sistema de alimentación con batería del bastón

Para diseñar el sistema de alimentación con batería se utiliza los siguientes componentes básicos:

1. **TP4056:** El TP4056 [55], es un chip de carga de baterías LiPo o Li-ion de una sola celda de 3.7V a 1 Ah, o superior como las 16550 o 18650 que no tienen su propio circuito de protección de una manera segura y eficiente. Sus principales características son:
  - Funciona con un voltaje de entrada de 4.5V a 5.5V, y un voltaje de carga full de 4.2V.

- El TP4056 utiliza el chip DW01 para regular el proceso de carga y evitar daños a la batería. El chip DW01 monitoriza constantemente la tensión de la batería. Cuando detecta que la tensión ha caído por debajo de un umbral seguro (2.4 V), desconecta la carga para evitar la descarga excesiva de la batería. Además, el DW01 puede monitorizar la corriente de carga y desconectarla si detecta corrientes anormalmente altas, asegurando así un proceso de carga seguro.
  - Tiene un led indicador: Rojo (Cargando), Azul o Verde (Cargado completamente).
  - La conexión incorrecta de la polaridad (conectar la batería al revés) puede causar daño instantáneo a la batería y a los dispositivos conectados. El chip de protección detecta esta situación y desconecta la carga para prevenir daños.
2. **TPS63802:** Es un regulador de conmutación step-down (también conocido como regulador buck) de Texas Instruments. Su función principal es convertir una fuente de voltaje de entrada [1.3V – 5V] en una salida de voltaje fija y estable [3.3V]. Este tipo de dispositivo es comúnmente utilizado en aplicaciones donde se requiere una eficiencia energética alta, como en sistemas de alimentación de baterías, dispositivos portátiles, y otros equipos electrónicos.
3. **SW\_BAT:** Sirve para prender/apagar el bastón.
4. **CON3:** Es un terminal que incluye conexiones para 3.3V, GND y una señal destinada al canal 8 del convertidor analógico a digital (ADC) del microcontrolador. Esta conexión estará disponible por si a futuro se necesite para determinar el nivel de voltaje de la batería. La conexión entre este terminal y la tarjeta principal se realiza mediante un cable plano de 3 hilos.

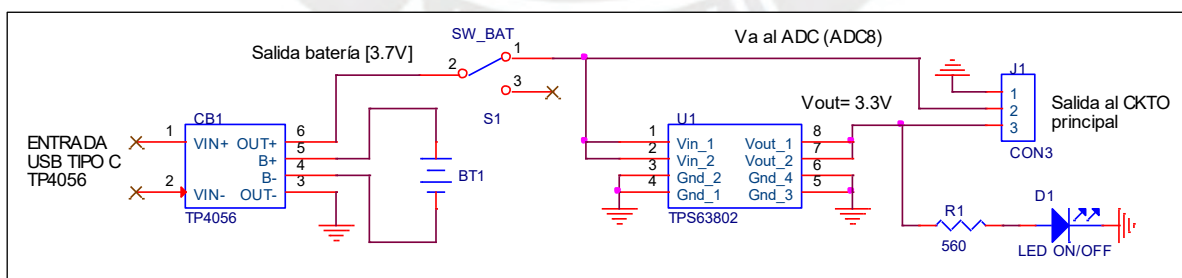


Figura 3-22. Circuito de control de la alimentación al bastón electrónico (Fuente: elaboración propia).

### 3.5. Consumo de energía de la tarjeta electrónica del bastón

Para calcular el consumo total de corriente y la duración de la batería 103450 de 2500 mAh, implica tener información detallada sobre los dispositivos conectados y sus respectivos consumos máximos de corriente. Ver detalle en la tabla adjunta.

Tabla 3-11. Consumo de corriente por el circuito del bastón.

DISPOSITIVO	CONSUMO mA
TEENSY 3.5	45 mA
BLUETOOTH HC06	40 mA (operación) – 1 mA (sleep)
RELOJ DS3231	120 $\mu$ A
ACELEROMETRO ADXL345	45 $\mu$ A - 90 $\mu$ A
OPAMP MCP6002	100 $\mu$ A
<b>Consumo total</b>	$\approx$ 85.31 mA (transmite con el BT) $\approx$ 45.31 mA (sleep con el BT)

Para determinar el tiempo de autonomía de una batería se usa la siguiente fórmula:

$$T = C/I \quad (3.2)$$

Donde:

T: Tiempo de autonomía; C: Capacidad de la batería; I: Corriente consumida por el circuito.

Para realizar el cálculo del tiempo de autonomía de la batería, se tiene dos escenarios:

- (1) **Bastón esté conectado a un celular a través del Bluetooth**, este caso se da cuando se desea crear un nuevo paciente, o se desea revisar el historial de cómo se usó el bastón durante un cierto tiempo (información grabada en la memoria microSD), o va a realizar un monitoreo en tiempo real del uso del bastón, ya sea al inicio del tratamiento para enseñar al paciente a utilizar bien el bastón o cuando el médico crea necesario. En todos estos casos, el bastón estará conectado al BT y su consumo será el máximo.

$$Tiempo = \frac{2500mAh}{85.31mA} = 29.30 \text{ horas} \approx \mathbf{29 \text{ horas} \approx 1 \text{ día } 5h}$$

- (2) **Bastón se usa de manera cotidiana**, en este caso la información de cómo el usuario utiliza el bastón en su desplazamiento, se almacena en la memoria microSD directamente sin emplear la conexión Bluetooth.

$$Tiempo = \frac{2500mAh}{45.31mA} = 55.17 \text{ horas} \approx \mathbf{55 \text{ horas} \approx 2 \text{ días } 7h}$$

De ambos tiempos, se puede concluir que está dentro de un rango de tiempo bastante aceptable, más aún si se considera que en uso combinado, podría llegar a un tiempo promedio de 40 horas.



Figura 3-23. Batería 103450 de 3.7 V y 2500 mAh

### 3.6. Acondicionamiento del sensor de fuerza en el bastón de rehabilitación

En el diseño del prototipo del proyecto de tesis, se tuvo que recurrir a la impresión en 3D usando el software de diseño “FreeCad” para la parte del regatón que sostiene al sensor de fuerza y también para la base de las tarjetas electrónicas. Las siguientes imágenes forman parte de todo el sistema.



Figura 3-24. Software de diseño en 3D.

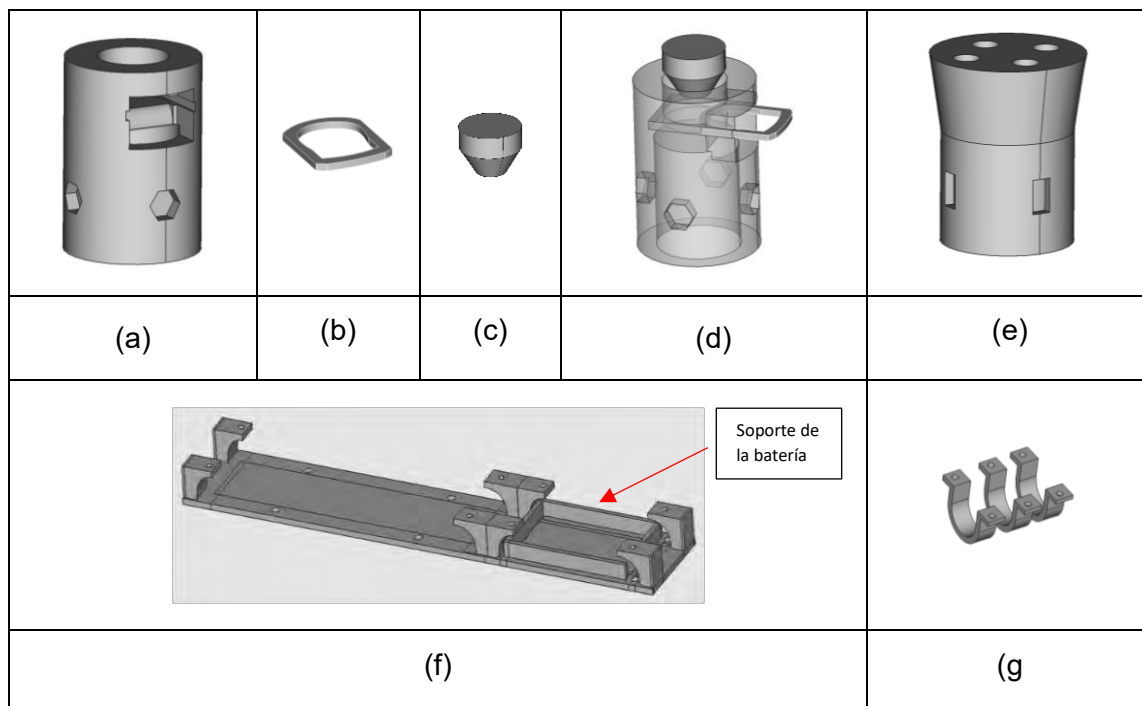


Figura 3-25. Imágenes 3D de los soportes de sensor y tarjetas electrónicas (Fuente: elaboración propia).

Descripción de las imágenes de la Figura 3-25:

- (a) Base principal que soporta el sensor de fuerza A201 y se coloca sobre la caña del bastón.
- (b) Empaquetadura que sirve para fijar el sensor de fuerza en la base (a) de manera plana.
- (c) Pieza que sirve para presionar al área sensible del sensor de fuerza que tiene un diámetro de 9 mm.

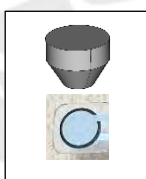


Figura 3-26. Muestra la forma que se ejerce presión sobre el sensor (Fuente: elaboración propia).

- (d) Muestra el montaje de las tres partes (a), (b) y (c).
- (e) Pieza deslizante que está en contacto con el suelo y este a su vez presiona a la pieza (c) para que transmita la presión ejercida sobre el bastón por cada pisada.
- (f) Base de soporte para las tarjetas de adquisición de datos y de alimentación.
- (g) Abrazadera de 2 orejas para fijar la base (f) en la caña del bastón.



Figura 3-27. Imágenes impresas en 3D de la figura 3-23 para el bastón de rehabilitación (Fuente: elaboración propia).

### 3.7. Diseño del sistema procesador y software

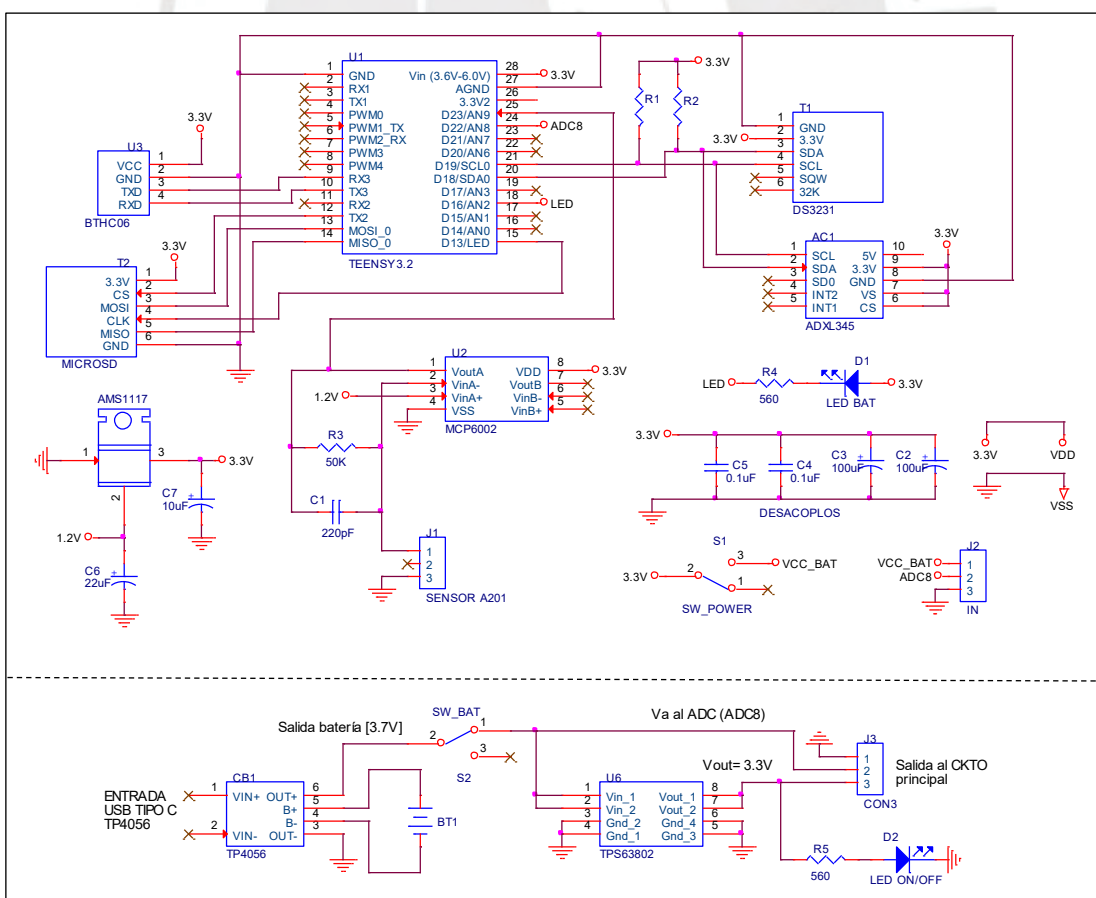


Figura 3-28. Circuito principal: CPU, Periféricos y Alimentación con batería (Fuente: elaboración propia).

En base a la información de la selección de componentes a utilizar en este proyecto del bastón de rehabilitación, la Figura 3-28. Circuito principal: CPU, Periféricos y Alimentación con batería muestra los esquemáticos del diseño del circuito principal y de alimentación de la batería. La Figura 3-29. Circuitos impresos del CPU más periféricos y alimentación con batería, muestra los circuitos electrónicos montados en una tarjeta de circuito impreso.

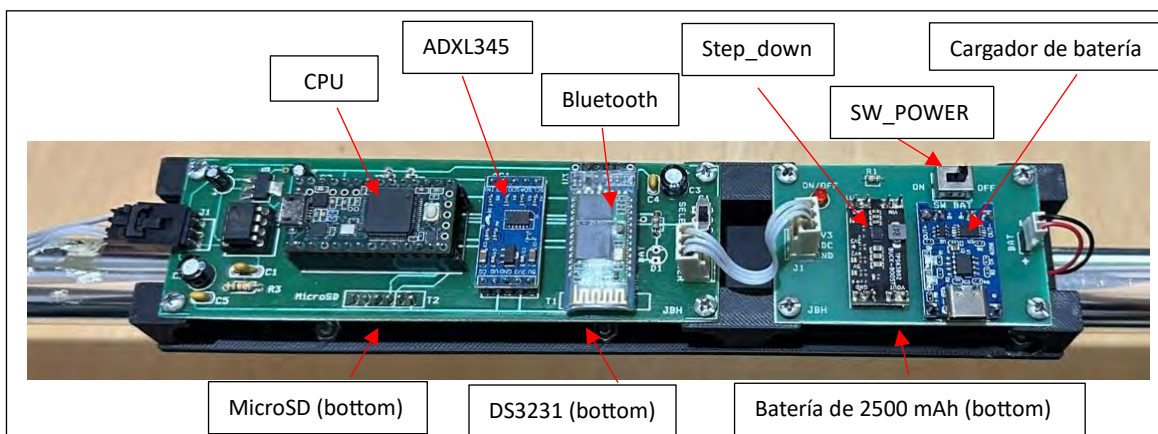


Figura 3-29. Circuitos impresos del CPU más periféricos y alimentación con batería (Fuente: elaboración propia).

### 3.7.1. Diseño del software del microcontrolador

La estrategia de control del software que va instalado en el microcontrolador para el bastón electrónico, se divide en seis bloques que son: Inicio; Setup; Uso bastón de manera cotidiana; Uso del aplicativo celular; Comunicación entre celular y bastón por Bluetooth; y funciones principales del proceso. Para la programación se utilizó el entorno de Arduino Ide para el microcontrolador y Android Studio para el aplicativo del celular.

La Figura 3-30, muestra el diagrama de bloques del software del circuito electrónico del bastón. A continuación, se describe sus partes:

#### 3.7.1.1. Bloque de inicio

Aquí se incluye todas las librerías (\*.h) que serán utilizadas dentro del programa; se declaran las variables y sus tipos, algunas se inicializan como constantes; se definen los registros del ADXL345 con sus respectivas direcciones; y se declaran los prototipos de las funciones.

Por ejemplo:

```
#include <Wire.h>           // Librería I2C
#include <SPI.h>            // Librería SPI
#include <SD.h>             // Librería MicroSD
#include <ADC.h>           // Librería ADC
```

```

#include <RTCLib.h> // Librería RTCLib
//declaro prototipos de las funciones a utilizar por el reloj y acelerómetro
void dateTime(uint16_t* date, uint16_t* time); // Obtiene la fecha y hora del DS3231
void ReadFrom(byte address, int num, byte _buff[]); // Lee bytes del registro de
// direcciones en el buffer ADXL345
void WriteTo(byte address, byte val); // Inicializa ADXL345
void ADXL345_GetBiass(void); // Lee datos de aceleración del ADXL345
void lectura_adxl435(void); // Lee los ángulos pitch y Rolls
void lectura_ADC(void); // Lee la fuerza aplicada sobre el bastón
#define ACEL_X_SCALE 0.0039 // Sensibilidad en cada eje extraído del datasheet
#define ACEL_Y_SCALE 0.0039
#define ACEL_Z_SCALE 0.0039

```

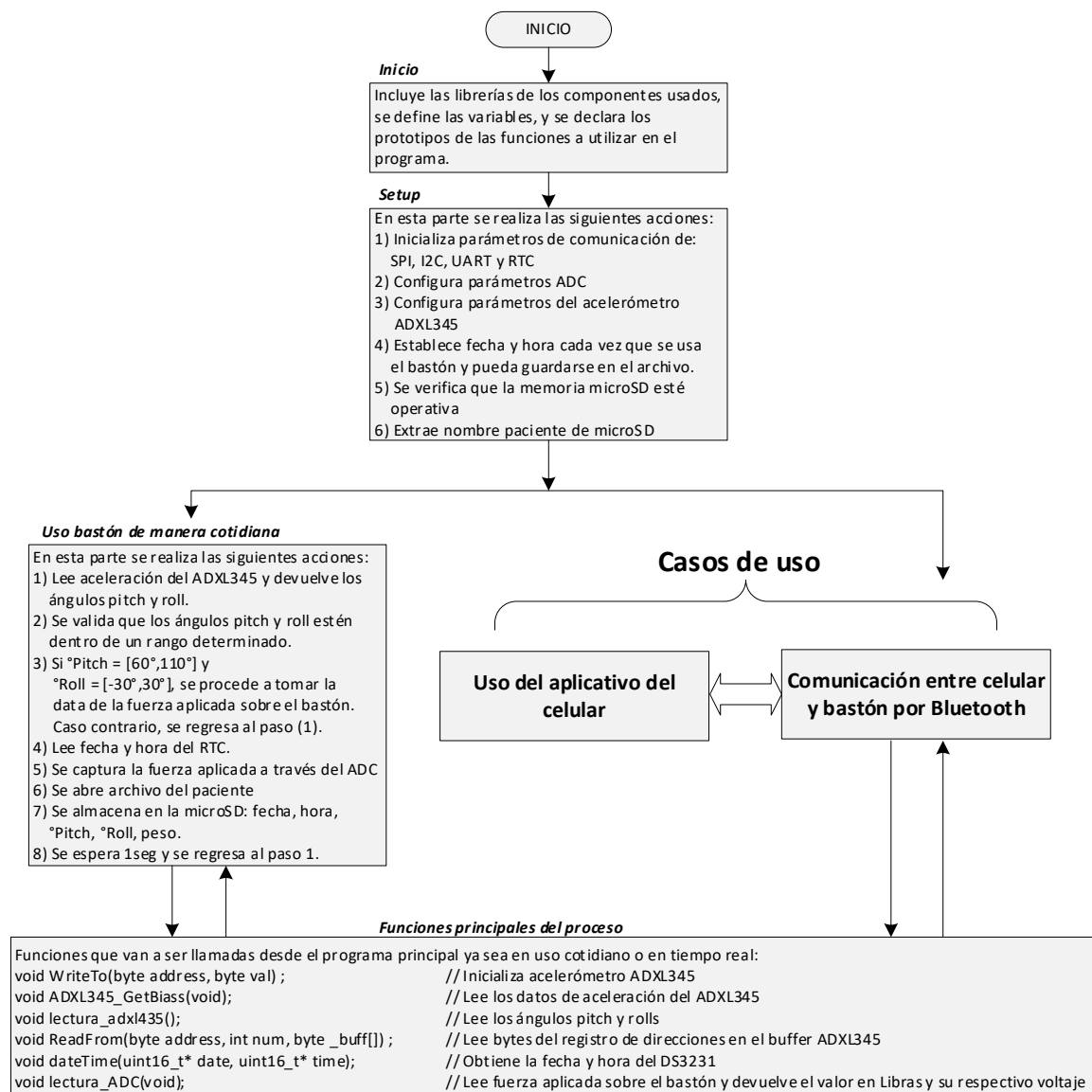


Figura 3-30. Estrategia de control del software del bastón electrónico para rehabilitación (Fuente: elaboración propia).

### 3.7.1.2. Bloque de Setup

Sirve para realizar todas las tareas de inicialización necesarias antes de que el programa entre en su bucle principal de ejecución (loop()). Por ejemplo, iniciar los pines de los puertos a ser usados e iniciar los parámetros de comunicación serial, I2C, SPI, etc.

Por ejemplo:

```
SPI.setMOSI(11);           // Pin 11 ← MOSI - SPI
SPI.setSCK(13);           // Pin 13 ← SCK - SPI
SPI.setMISO(12);          // Pin 12 ← MISO - SPI
Wire.begin();             // Inicializa comunicación I2C
Serial3.begin(115200);     // Establece la comunicación Bluetooth a 115200 Baudios
Serial.begin(9600);        // Establece la comunicación serial a 9600 Baudios
RTC.begin();              // Establece la velocidad de datos del RTC (DS3231)

/***** Configuro ADC del microcontrolador *****/
pinMode(ReadPinADC9, INPUT); // Pin A9 ← Captura la señal analógica
adc->adc0->setAveraging(12); // Establece el núm. promedio de muestras en 12
adc->adc0->setResolution(12); // Resolución a 12 bits
// Establece velocidad de conversión a LOW
adc->adc0->setConversionSpeed(ADC_CONVERSION_SPEED::LOW_SPEED);
// Velocidad de muestreo media
adc->adc0->setSamplingSpeed(ADC_SAMPLING_SPEED::MED_SPEED);
/***** Configuro acelerómetro ADXL345 *****/
WriteTo(POWER_CTL, 0x00); // Seteo ADXL345 en modo de medición escribiendo 0x00
WriteTo(DATA_FORMAT, 0x0B); // Seteo ADXL345 en rango +/- 16G y full precisión
WriteTo(BW_RATE, 0x09); // Opera a 50Hz, consumo 90uA, modo normal
WriteTo(POWER_CTL, 0x08); // Seteo ADXL345 en modo de medición
```

### 3.7.1.3. Bastón en modo uso cotidiano

El programa ejecuta estas instrucciones cuando no se enlaza con el Bluetooth, esta sería la forma más frecuente de uso, aquí cada vez que se utilice el bastón y cumple ciertas condiciones, se irá grabando en la memoria la fecha, hora, ángulos Pitch y Rolls, así como la fuerza aplicada en kilos, para luego ser analizado por el médico tratante a través del aplicativo del celular en su próxima cita y de ser necesario exportar esta información al computador.

#### Rutina de adquisición de datos del ADXL345

La rutina para la adquisición de datos y calcular los ángulos necesarios para el proceso del bastón, son los ángulos Pitch (avance en el sentido de caminar de la persona) y Rolls (desviación del bastón hacia la izquierda o derecha de la persona) y sus rangos válidos de Pitch= [60°, 110°] y Rolls= [-30°, 30°].

#### Rutina principal ADXL345:

```
ReadFrom( X_Axis_Register_DATA_X0, BytesToRead, _buff); //Lee la aceleración de los ejes
accel_X = (((int)_buff[1]) << 8) | _buff[0]; //Se asigna a eje X
accel_Y = (((int)_buff[3]) << 8) | _buff[2]; //Se asigna a eje Y
accel_Z = (((int)_buff[5]) << 8) | _buff[4]; //Se asigna a eje Z
X=((float)accel_X - biass_X) * ACEL_X_SCALE; //Se convierte en valores de gx, gy, gz
Y=((float)accel_Y - biass_Y) * ACEL_Y_SCALE; //El valor de ACEL_Y_SCALE= 0.0039 g/LSB
Z=((float)accel_Z - biass_Z) * ACEL_Z_SCALE;
```

```
pitchAccel = atan2(X,Z) * 360.0 / (2*PI); //calcula el valor del ángulo en °
rollAccel = asin(Y) * 360.0 / (2*PI);
```

Cabe precisar, que el sensor puede detectar aceleraciones en el rango de medición preconfigurado en este proyecto de  $\pm 16g$ , con  $g = 9.8 \text{ m/s}^2$ . Según estos dos parámetros, el rango de medición y la cantidad de bits por dato, determina la sensibilidad, factor de escala o precisión del sensor [26], que es de  $0.0039 \text{ g/LSB} = 3.9 \text{ mg/LSB}$ .

### Rutina de adquisición de datos del ADC

Sólo si se cumple que las dos condiciones del acelerómetro, ángulos Pitch [ $60^\circ$ ,  $110^\circ$ ] y Rolls [ $-30^\circ$ ,  $30^\circ$ ] están dentro de estos rangos, se procede a realizar la captura de la fuerza aplicada por el bastón, caso contrario se descarta y se vuelve a leer el acelerómetro.

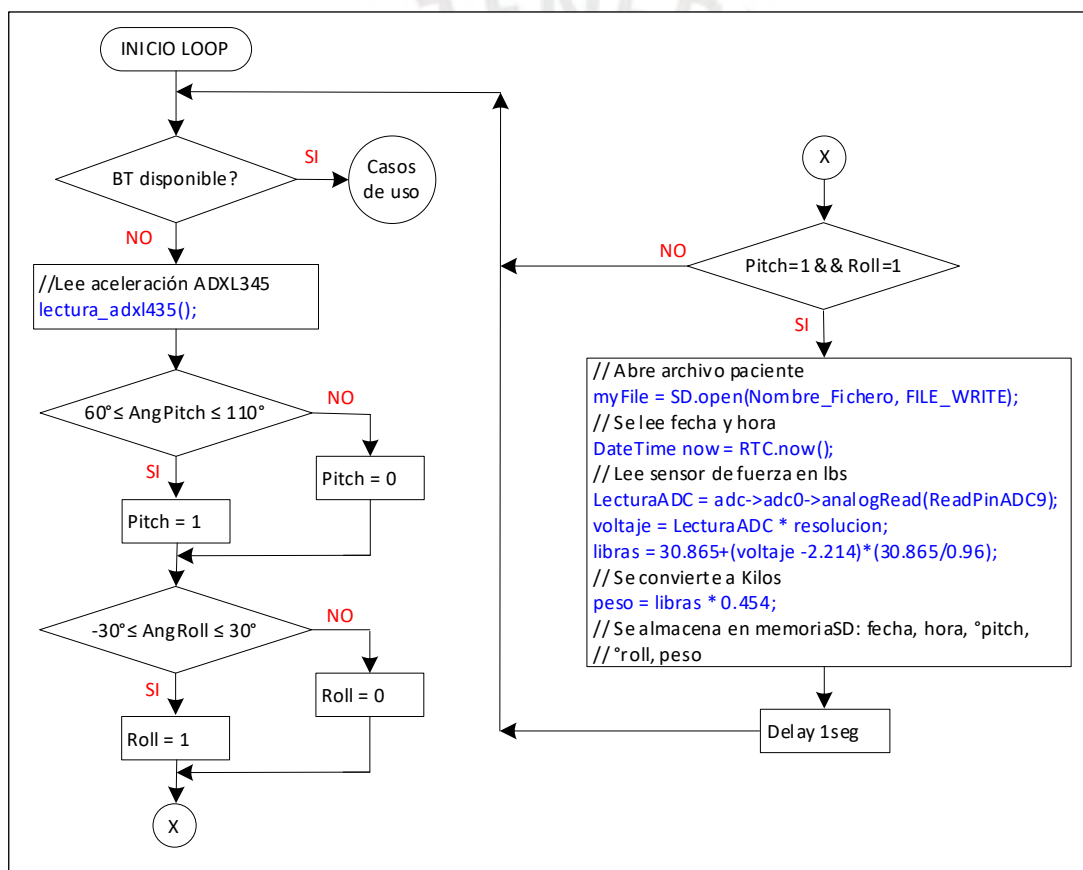


Figura 3-31. Diagrama de flujo de adquisición de datos del ADC (Fuente: elaboración propia).

### 3.7.2. Diseño del software del Aplicativo Móvil

La siguiente Figura 3-32, muestra la arquitectura del software de control del Aplicativo Móvil para el sistema operativo Android [56], la cual se divide principalmente en: Capa de presentación, Capa de negocio y Capa de datos.

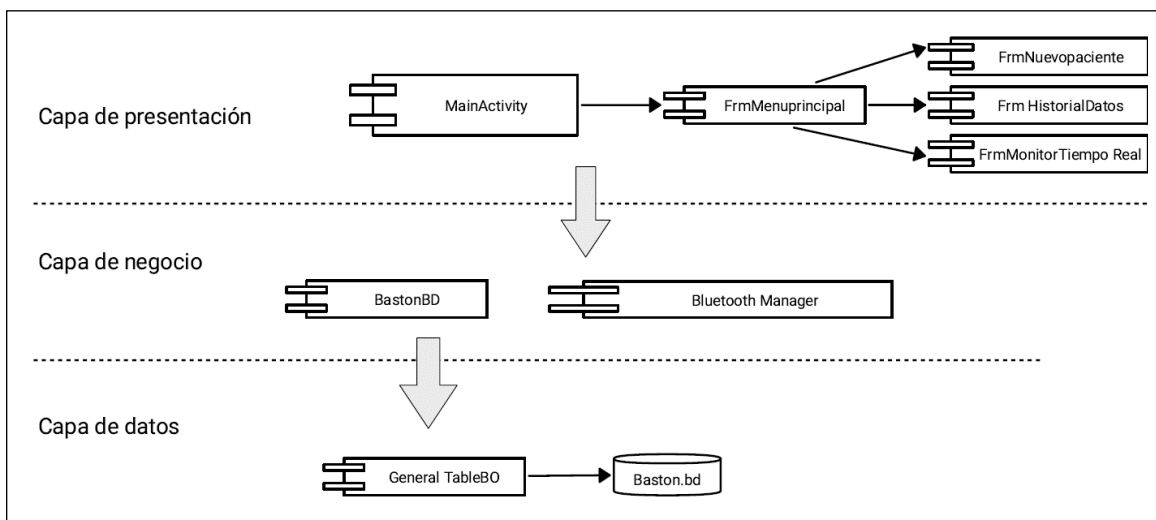


Figura 3-32. Arquitectura del software de control del AppMóvil (Fuente: elaboración propia)

En la **Capa de presentación**, se encuentra la implementación de todas las pantallas del sistema, entre ellas, el Menú Principal del aplicativo y las pantallas de Nuevo paciente, Historial de datos y Monitor de tiempo real. En la **Capa de negocio**, se programa la interacción con el hardware del bastón y la interpretación de los datos que se debe mostrar en la capa de presentación. En la **Capa de datos**, se encuentra la base de datos que almacena los datos del paciente, como Nombre, número de Historia clínica o DNI, Edad y Peso expresado en kilos.

#### 3.7.2.1. Bastón en modo aplicativo celular – bastón (casos de uso)

Tomando como referencia la “Estrategia de control del software del bastón electrónico” (Figura 3-30) y la “Arquitectura del software del aplicativo” (Figura 3-32), se va a detallar la lógica de interacción entre el CPU del bastón de rehabilitación y AppMóvil instalado en el celular con sistema operativo Android y el software Android Studio [57] [58]. Para ingresar al aplicativo se debe dar un clic al icono de la aplicación instalada:



Aparecerá la Figura 3-33(a) Pantalla inicial y luego de dar clic en el botón de OFF, el aplicativo se enlazará con el bastón por medio del Bluetooth, apareciendo la Figura 3-33(b) correspondiente al Menú Principal, mostrando tres opciones:

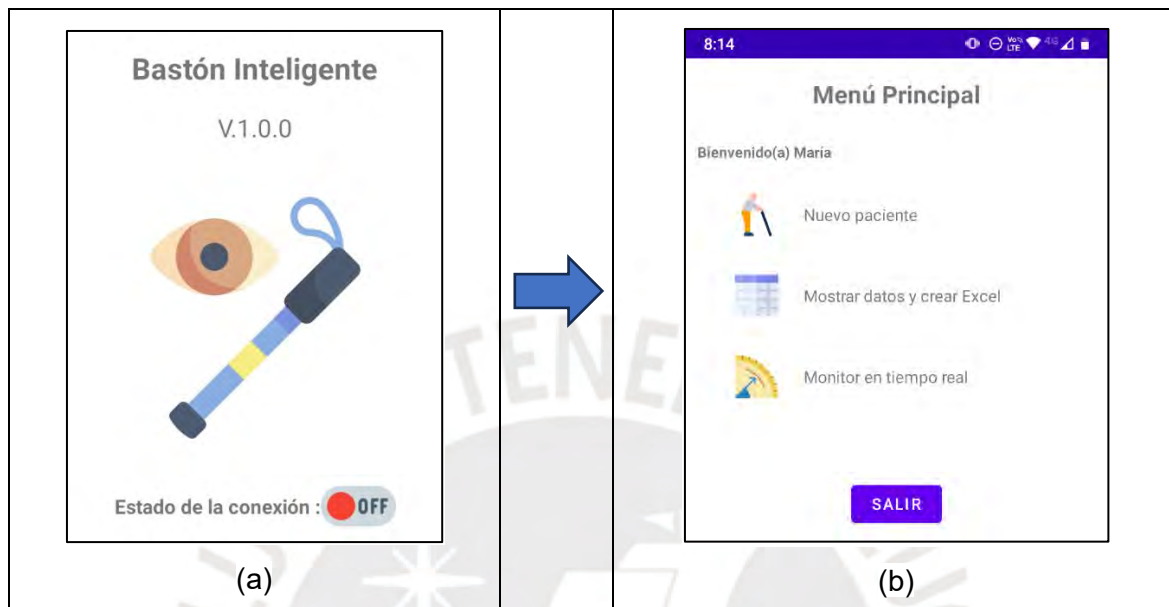


Figura 3-33 Pantallas de inicio (a) y Menú Principal (b) (Fuente:elaboración propia).

### 1) Crear paciente nuevo

#### Descripción:

El usuario selecciona en el menú principal del aplicativo móvil la opción de «Nuevo Paciente», a continuación, se muestra la pantalla con los datos obligatorios que hay que completar.

Figura 3-34. Pantalla de Paciente Nuevo (Fuente:elaboración propia).

Usuario	AppMóvil
	<ul style="list-style-type: none"> <li>● Muestra la pantalla para registrar a un nuevo paciente. (Ver Figura 3-34)</li> </ul>
<ul style="list-style-type: none"> <li>● Ingresa los datos solicitados:               <ul style="list-style-type: none"> <li>● Nombre completo del paciente.</li> <li>● Número de historia clínica.</li> <li>● Peso.</li> <li>● Edad.</li> </ul> </li> <li>● Selecciona el botón «Registrar».</li> </ul>	
	<ul style="list-style-type: none"> <li>● Se valida que el usuario haya ingresado todos los datos, pues son obligatorios.</li> <li>● En caso no se hubiera ingresado uno o varios datos, el sistema mostrará un mensaje indicando que datos faltan y no se registrará al nuevo paciente.</li> <li>● En caso se hubieran ingresado todos los datos:               <ul style="list-style-type: none"> <li>■ Envía vía bluetooth el carácter «1».</li> <li>■ Lee un byte o carácter desde bluetooth.</li> <li>■ Si carácter leído es la letra «c», se registra al usuario en la BD interno del aplicativo y se muestra un mensaje de éxito al usuario.</li> <li>■ Si carácter leído es la letra «e», se registra al usuario en la BD interno del aplicativo y se muestra un mensaje de éxito al usuario.</li> <li>■ En caso contrario se muestra un mensaje de error inesperado.</li> <li>■ En caso de éxito el sistema regresará el menú principal.</li> </ul> </li> </ul>

## 2) Mostrar historial de uso del bastón

### Descripción:

El usuario selecciona en el menú principal del AppMóvil la opción de «Mostrar datos y crear Excel», y se muestra la pantalla del historial con todos los datos almacenados en el bastón.



Figura 3-35. Datos almacenados del día 10/04/2024 hasta 14/04/2024 (Fuente: elaboración propia).

En la Figura 3-35, se visualiza una parte de los datos almacenados en el bastón en forma de grafica de barras y dos íconos de calendario para seleccionar la fecha para analizar los datos. Si se desea ver los demás datos, se debe desplazar hacia la izquierda como scroll.

### Interacción entre Bastón y AppMóvil:

Etapa en la que el AppMóvil se comunica con el bastón para poder mostrar los datos.

Bastón	AppMóvil
<ul style="list-style-type: none"> <li>● Lee todos los datos del microSD.</li> <li>● Crea un string con todos los datos leídos, con formato CSV, con las siguientes columnas:               <ul style="list-style-type: none"> <li>■ Fecha</li> <li>■ Hora</li> <li>■ Pitch</li> <li>■ Roll</li> <li>■ Peso</li> </ul> </li> <li>● Envía vía Bluetooth la letra «k» para confirmar que enviará los datos.</li> <li>● Envía vía Bluetooth el string con todos los datos.</li> </ul>	<ul style="list-style-type: none"> <li>● Envía vía Bluetooth el código «4»</li> </ul>
	<ul style="list-style-type: none"> <li>● Lee string desde Bluetooth.</li> <li>● Procesa el string para generar una lista de objetos con los datos de cada una de las filas.</li> <li>● Calcula el nivel de zoom o la escala para poder mostrar todos los datos, y permitir que el usuario haga desplazamientos horizontales para ver toda la información.</li> <li>● Calcula las fechas mínima y máxima de todos los datos.</li> <li>● Genera un gráfico de barras con todos los datos. (Ver Figura 3-35)</li> </ul>

### El usuario filtra los datos por fecha:

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>● Selecciona el ICONO del calendario que está en la sección «fecha inicial».</li> </ul>	
	<ul style="list-style-type: none"> <li>● Muestra una ventana emergente con un calendario. (Ver Figura 3-36)</li> </ul>
<ul style="list-style-type: none"> <li>● Selecciona una fecha.</li> </ul>	
	<ul style="list-style-type: none"> <li>● Oculta la ventana emergente.</li> <li>● Refresca la gráfica mostrando los datos cuya fecha sea igual o posterior a la «fecha inicial» seleccionada por el usuario.</li> </ul>
<ul style="list-style-type: none"> <li>● Selecciona el ICONO del calendario que está en la sección «fecha final».</li> </ul>	
	<ul style="list-style-type: none"> <li>● Muestra una ventana emergente con un calendario. (Ver Figura 3-36)</li> </ul>
<ul style="list-style-type: none"> <li>● Selecciona una fecha.</li> </ul>	
	<ul style="list-style-type: none"> <li>● Oculta la ventana emergente.</li> <li>● Refresca la gráfica mostrando los datos cuya fecha sea igual o posterior a la «fecha inicial» e igual o anterior a la «fecha final» seleccionada por el usuario.</li> </ul>



Figura 3-36. Pantalla emergente del calendario (Fuente: elaboración propia).

**El usuario solicita ver información del día de hoy:**

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>● Selecciona el BOTON con etiqueta «HOY DIA»</li> </ul>	<ul style="list-style-type: none"> <li>● Muestra la fecha de hoy en la sección «fecha inicial».</li> <li>● Muestra la fecha de hoy en la sección «fecha final».</li> <li>● Refresca la gráfica mostrando los datos cuya fecha, sin considerar los datos de hora, sea igual al día de hoy (Ver Figura 3-37).</li> </ul>



Figura 3-37. Muestra datos almacenados del día 10.04.2024 (Fuente: elaboración propia).


### El usuario solicita ver información de la semana actual:

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>● Selecciona el BOTON con etiqueta «ESTA SEMANA»</li> </ul>	
	<ul style="list-style-type: none"> <li>● Calcula la fecha del lunes.</li> <li>● Calcula la fecha del domingo.</li> <li>● En la sección 'fecha inicial', muestra la fecha del día lunes.</li> <li>● En la sección 'fecha final', muestra la fecha del día domingo.</li> <li>● Refresca la gráfica mostrando los datos cuya fecha sea igual o posterior a la «fecha inicial» e igual o anterior a la «fecha final».</li> </ul>

### El usuario solicita ver toda la información:

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>● Selecciona el BOTON con etiqueta «TODOS»</li> </ul>	
	<ul style="list-style-type: none"> <li>● Muestra la fecha «fecha mínima» calculada al recibir toda la información desde el bastón, en la sección «fecha inicial».</li> <li>● Muestra la fecha «fecha máxima» calculada al recibir toda la información desde el bastón, en la sección «fecha final».</li> <li>● Refresca la gráfica mostrando los datos cuya fecha sea igual o posterior a la «fecha inicial» e igual o anterior a la «fecha final».</li> </ul>

### El usuario comparte o extrae la información:

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>● Selecciona el BOTON «compartir»: </li> </ul>	
	<ul style="list-style-type: none"> <li>● Crea un archivo temporal de texto con formato CSV (datos separados por coma) con el mismo formato empleado por el Bastón: <ul style="list-style-type: none"> <li>■ Fecha</li> <li>■ Hora</li> <li>■ Pitch</li> <li>■ Roll</li> <li>■ Peso</li> </ul> </li> <li>● El nombre del archivo de texto tendrá el siguiente formato: baston_hhhhh_xxxx.csv Donde: <ul style="list-style-type: none"> <li>● hhhhh es el número de historial del paciente.</li> <li>● xxxxx es un número aleatorio que nos asegura que el archivo sea único.</li> </ul> </li> <li>● Solicita al <b>Sistema Android</b> que se cargue las opciones de «compartir» para que cualquier aplicación registrada en el equipo como procesador o visor de archivos de texto, pueda ser el destinatario del archivo creado.</li> </ul>
<ul style="list-style-type: none"> <li>● Selecciona la aplicación a la que desea enviar o con la que se compartirá el archivo.</li> </ul>	
	<ul style="list-style-type: none"> <li>● Cierra u oculta la ventana de compartir.</li> </ul>

### 3) Monitoreo en Tiempo Real

#### Descripción:

El usuario selecciona en el menú principal del aplicativo móvil la opción de «Monitor en Tiempo Real», a continuación, el sistema móvil muestra la pantalla del Monitor.



Figura 3-38. Pantalla del Monitor en Tiempo Real (Fuente: elaboración propia).

El **semáforo del peso** (Figura 3-39) se distribuye en los siguientes porcentajes, considerando la Tesis Doctoral de Fernández, María del Carmen Rídao [10], que indica que sobre el bastón de antebrazo se debe aplicar máximo un 10% del peso según la escala CHAGS (*Chamorro Assisted Gait Scale*), y según Melis EH. et al. [11] en su estudio sobre el bastón de mano, donde recalca que este bastón se debe utilizar como máximo hasta un 15% de soporte vertical sobre él, porque se vuelve inestable cuando se carga a partir del 20-25% del peso corporal. Los demás porcentajes se distribuyeron para darle la forma de semáforo, donde “rojo bajo” puede darse en un paciente que recién está aprendiendo a usar el bastón, “naranja bajo” para un paciente que está mejorando, “verde” que el paciente ya aprendió a usar el bastón y ejerce la fuerza adecuada sobre el bastón.

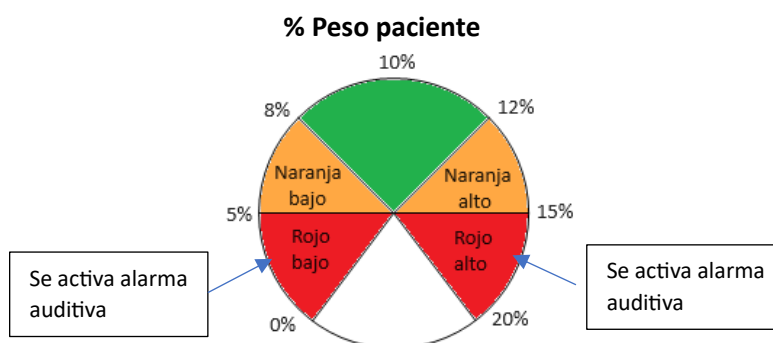


Figura 3-39. Detalle de los colores del semáforo del peso (Fuente: elaboración propia)

Cabe precisar, que en su tesis doctoral, Sánchez, Raquel Sobrino [59], menciona: “Es necesario crear protocolos de actuación que permitan decidir el porcentaje de carga más adecuado en cada fase de la recuperación, pero hay poca información sobre la biomecánica de la marcha con estos dispositivos para guiar la práctica clínica, ..., Controlar la descarga del miembro inferior en estado patológico y conocer las implicaciones biomecánicas que ocasionan en la distorsión del gesto de la marcha nos permitirá realizar tratamientos de fisioterapia más eficientes, con mayor calidad, en menor tiempo y con menos secuelas”.

A continuación, se muestra unas imágenes del semáforo del peso, similar a un tacómetro durante el monitoreo en tiempo real para distintas fuerzas ejercidas sobre el bastón. La aguja es la que muestra en función del peso a que zona cae.

Los valores mostrados son tomados de un paciente con 70 Kg.

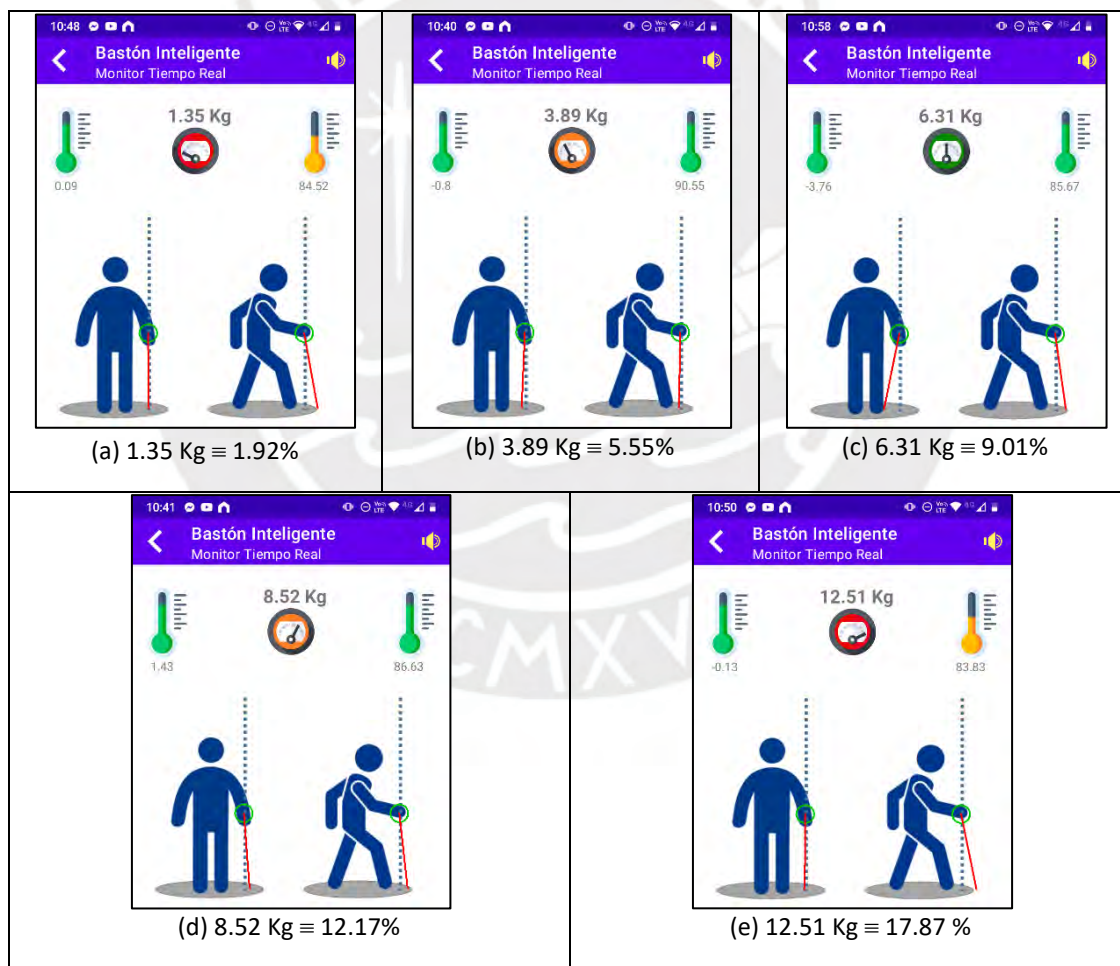


Figura 3-40. Semáforo del peso para distintas fuerzas ejercidas sobre el bastón  
(Fuente: elaboración propia).

## Interacción entre Bastón y App Móvil

Etapa en la que el App Móvil se comunica con el bastón a través del Bluetooth para iniciar el modo de visualización de datos en tiempo real.

Bastón	AppMóvil
	<ul style="list-style-type: none"> <li>● Envía vía Bluetooth el código «5»</li> </ul>
<ul style="list-style-type: none"> <li>● Envía la «0» vía Bluetooth para confirmar que se inicia el envío de datos en tiempo real.</li> </ul>	
	<ul style="list-style-type: none"> <li>● Lee un byte desde el Bluetooth, si este byte es diferente a la letra «0» se muestra un mensaje de error y se termina el proceso.</li> <li>● Muestra la pantalla y las gráficas de los datos en tiempo real (Ver Figura 3-38).</li> </ul>
<p>A continuación, aparece un bucle entre el bastón y al AppMóvil, siempre y cuando el usuario no seleccione la opción de salir.</p>	
<ul style="list-style-type: none"> <li>● Espera a que el usuario mueva el bastón.</li> <li>● Si se detecta que el usuario movió el bastón, se envían vía bluetooth los siguientes datos: <ul style="list-style-type: none"> <li>■ Letra «d».</li> <li>■ Pitch.</li> <li>■ Letra «#».</li> <li>■ Roll.</li> <li>■ Letra «#».</li> <li>■ Peso.</li> <li>■ Letra «#».</li> </ul> </li> </ul>	
	<ul style="list-style-type: none"> <li>● Espera que aparezcan o lleguen datos desde el Bluetooth.</li> <li>● Lee un byte, si es igual a la letra «d», inicia la lectura de los datos enviados por el bastón, descritos en el párrafo previo.</li> <li>● Visualiza las cantidades numéricas que corresponden a cada dato en la pantalla.</li> <li>● Calcula el semáforo del PITCH según las siguientes reglas: <ul style="list-style-type: none"> <li>■ Si <math> pitch-90  &lt; 5</math>, entonces el semáforo es «verde».</li> <li>■ Si <math> pitch-90  &lt; 15</math>, entonces el semáforo es «naranja».</li> <li>■ En otro caso, el semáforo en rojo.</li> </ul> </li> <li>● Calcula el semáforo del ROLL según las siguientes reglas: <ul style="list-style-type: none"> <li>■ Si <math> roll  &lt; 5</math>, entonces el semáforo es verde.</li> <li>■ Si <math> roll  &lt; 10</math>, entonces el semáforo es «naranja».</li> <li>■ En otro caso, el semáforo en rojo.</li> </ul> </li> <li>● Calcula el semáforo del peso según las siguientes reglas: <ul style="list-style-type: none"> <li>■ Si (peso <math>\leq</math> 5% del peso del paciente), entonces se muestra semáforo rojo bajo y se activa una alarma auditiva.</li> <li>■ Si (peso <math>\leq</math> 8% del peso del paciente), entonces se muestra semáforo naranja bajo.</li> <li>■ Si (peso <math>\geq</math> 12% del peso del paciente), entonces se muestra semáforo naranja alto.</li> <li>■ Si (peso <math>\geq</math> al 15% del peso del paciente), entonces se muestra semáforo rojo alto y se activa una alarma.</li> <li>■ De lo contrario se muestra semáforo verde.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>● Para mostrar la gráfica interactiva que representa el movimiento del bastón se hacen los siguientes cálculos:</li> </ul> <ol style="list-style-type: none"> <li>1. La imagen de las personas se escala para que ocupe todo el espacio horizontal de la pantalla del celular, menos el espacio ocupado horizontalmente por los dos termómetros (Ver Figura 3-41):          <math display="block">\text{dimDisponible} = \text{posXBordezquierdoTerPitch} - \text{posXBordezquierdoTerRoll}</math> <math display="block">\text{escala} = \text{dimDisponible} / \text{ancholmgPersonas}</math> </li> <li>2. Previamente se conocen las coordenadas de X de las manos de las personas, así como las coordenadas Y de las manos y del piso de las personas. Estas se obtienen de la imagen original:       <ol style="list-style-type: none"> <li>a) rollPosXMano</li> <li>b) rollPosYMano.</li> <li>c) rollPosYPiso.</li> <li>d) pitchPosXMano.</li> <li>e) pitchPosYMano.</li> <li>f) pitchPosYPiso.</li> </ol> <p>Donde:</p> <math display="block">\text{pitchPosYPiso} = \text{rollPosYPiso}</math> <math display="block">\text{rollPosYMano} = \text{rollPosYMano}</math> </li> <li>3. Se recalculan todas las coordenadas del punto 2 multiplicándolas por la escala calculada en el punto 1.</li> <li>4. Se aplica la siguiente fórmula para calcular la coordenada X del bastón con el ROLL:       <math display="block">\text{rollDeltaX} = \text{rollPosYPiso} * \text{TAN}(\text{roll})</math> <math display="block">\text{rollPosXPiso} = \text{rollPosXMano} + \text{rollDeltaX}</math> </li> <li>5. Se aplica la siguiente fórmula para calcular la coordenada X del bastón con el PITCH:       <math display="block">\text{pithDeltaX} = \text{rollPosYPiso} * \text{TAN}(\text{pitch}-90)</math> <math display="block">\text{pitchPosXPiso} = \text{rollPosXMano} - \text{pithDeltaX}</math> </li> <li>6. A todas las coordenadas X calculadas en el punto 2, se les debe sumar posXBordeDerechoTerRoll.</li> <li>7. A todas las coordenadas Y calculadas en el punto 2, se les debe sumar la coordenada posYBordeInferiorTerPitch.</li> </ol>
--	---

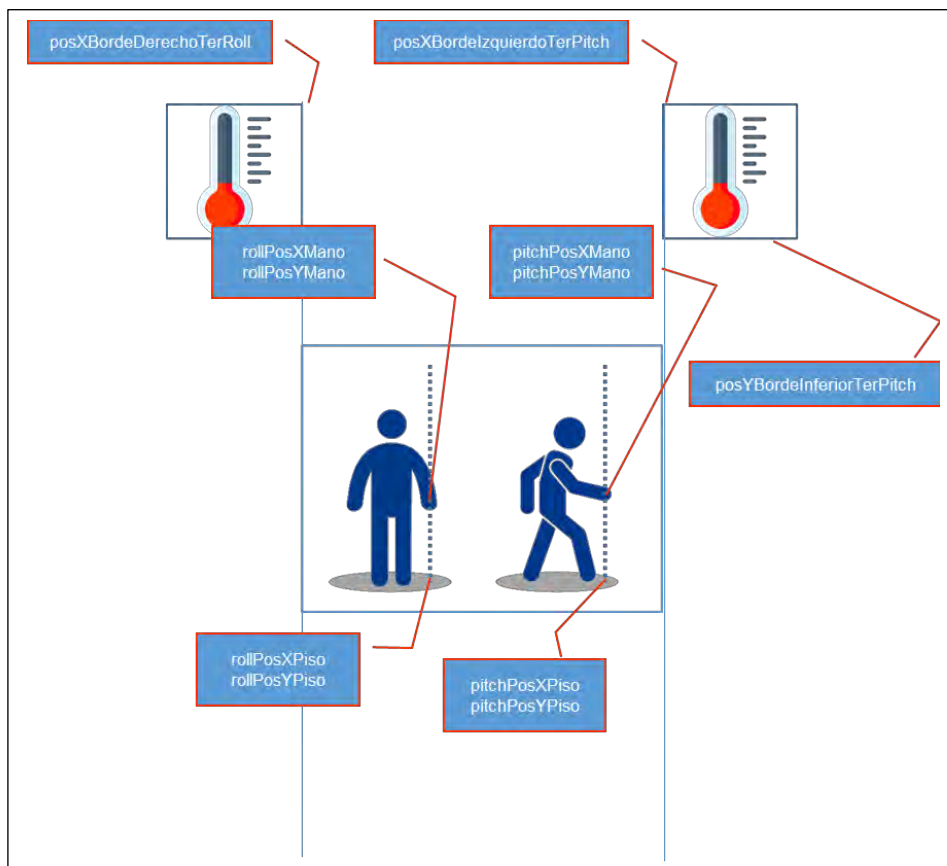





Figura 3-41. Gráfico que muestra las etiquetas del Monitor en Tiempo Real (Fuente: elaboración propia).

### Interacción entre Bastón y App Móvil con el icono de sonido de alarma y salir:

Usuario	AppMóvil
<ul style="list-style-type: none"> <li>Selecciona el ICONO para no emitir alarmas.</li> </ul> 	
<ul style="list-style-type: none"> <li>Selecciona el ICONO para emitir alarmas.</li> </ul> 	<ul style="list-style-type: none"> <li>Deja de emitir sonido cada vez que se recibe data.</li> </ul>
<ul style="list-style-type: none"> <li>Selecciona la opción para salir de la pantalla.</li> </ul> 	<ul style="list-style-type: none"> <li>Envía el carácter «f» vía bluetooth para indicar al bastón que deje de enviar los datos en tiempo real.</li> <li>Regresa al menú principal.</li> </ul>

## CAPITULO IV

### PRUEBA Y RESULTADOS

En esta parte del proyecto se va a realizar en primer lugar, las pruebas de funcionamiento de manera estática de los componentes principales descritos en el capítulo 3 y montados sobre la caña del bastón como se muestra en la Figura 3-29. En segundo lugar, las pruebas se realizarán de manera dinámica utilizando a una persona sana caminando con el bastón donde se registrará en la memoria SD los datos de fecha, hora, pitch, roll y peso, para posteriormente toda la información ser transferida al programa Excel y graficarse para su análisis respectivo.

#### 4.1. Prueba del sistema de instrumentación

Los componentes principales del bastón inteligente para la adquisición y registro de datos se muestran en la Figura 3-2, del capítulo 3:

- Sensor de fuerza (Flexiforce A201)
- Acelerómetro (ADXL345)
- Reloj en tiempo real (DS3231)
- Módulo de memoria microSD

A continuación, se muestra las pruebas de estos componentes.

##### 4.1.1. Sensor de fuerza Flexiforce A201

En la Figura 3-9. Tarjeta principal del bastón con el sensor de fuerza, se muestra la tarjeta de circuito impreso conteniendo los componentes principales del proyecto más el sensor de fuerza Flexiforce de 25 libras (11.35 Kg) que va ubicado en el regatón (Figura 3-27). Cabe mencionar, que estos sensores pueden soportar una sobrecarga del 50% sin que sufran daño permanente.

	Typical Performance	Evaluation Conditions
Linearity (Error)	< ±3% of full scale	Line drawn from 0 to 50% load
Repeatability	< ±2.5%	Conditioned sensor, 80% of full force applied
Hysteresis	< 4.5% of full scale	Conditioned sensor, 80% of full force applied
Drift	< 5% per logarithmic time scale	Constant load of 111 N (25 lb)
Response Time	< 5µsec	Impact load, output recorded on oscilloscope
Operating Temperature	-40°C - 60°C (-40°F - 140°F)	Convection and conduction heat sources
Acceptance Criteria	±40% sensor-to-sensor variation	Output considered at test pressure
Durability	≥ 3 million actuations	Perpendicular load, room temperature, 22 N (5 lb)
Temperature Sensitivity	0.36%/°C (± 0.2%/°F)	Conductive heating

Figura 4-1. Características del sensor de fuerza A201 de FlexiForce. (Fuente: <https://www.tekscan.com/sites/default/files/resources/FLX-Datasheet-A201-RevI.pdf>)

La hoja de datos del fabricante menciona que para el sensor FlexiForce A201 de 25 libras, la escala completa (full scale) es 25 libras. Un error de linealidad de  $\leq \pm 3\%$  de la escala completa implica que en cualquier punto dentro del rango de medición del sensor (0 a 25 libras), la desviación máxima de la respuesta real del sensor de una línea recta ideal no excederá el  $\pm 3\%$  de 25 libras, es decir, de  $\pm 0.75$  libras ( $\pm 0.34$  Kg  $\equiv$  340 gr).

Con el programa cargado en el microcontrolador, se debe realizar las pruebas de funcionamiento con el bastón de manera estática y validar las ecuaciones de la recta1 y recta2 del sensor de fuerza (Figura 3-13). Para ello, se contó con los siguientes componentes o materiales:

- Tres pesas de 5 Kg cada una (prestado por el Laboratorio de Mecánica y Electrónica de la PUCP).
- Una pesa de 4.35 Kg.
- Peso de bastón y tarjeta electrónica con todos sus componentes 0.490 Kg.
- Una balanza digital.

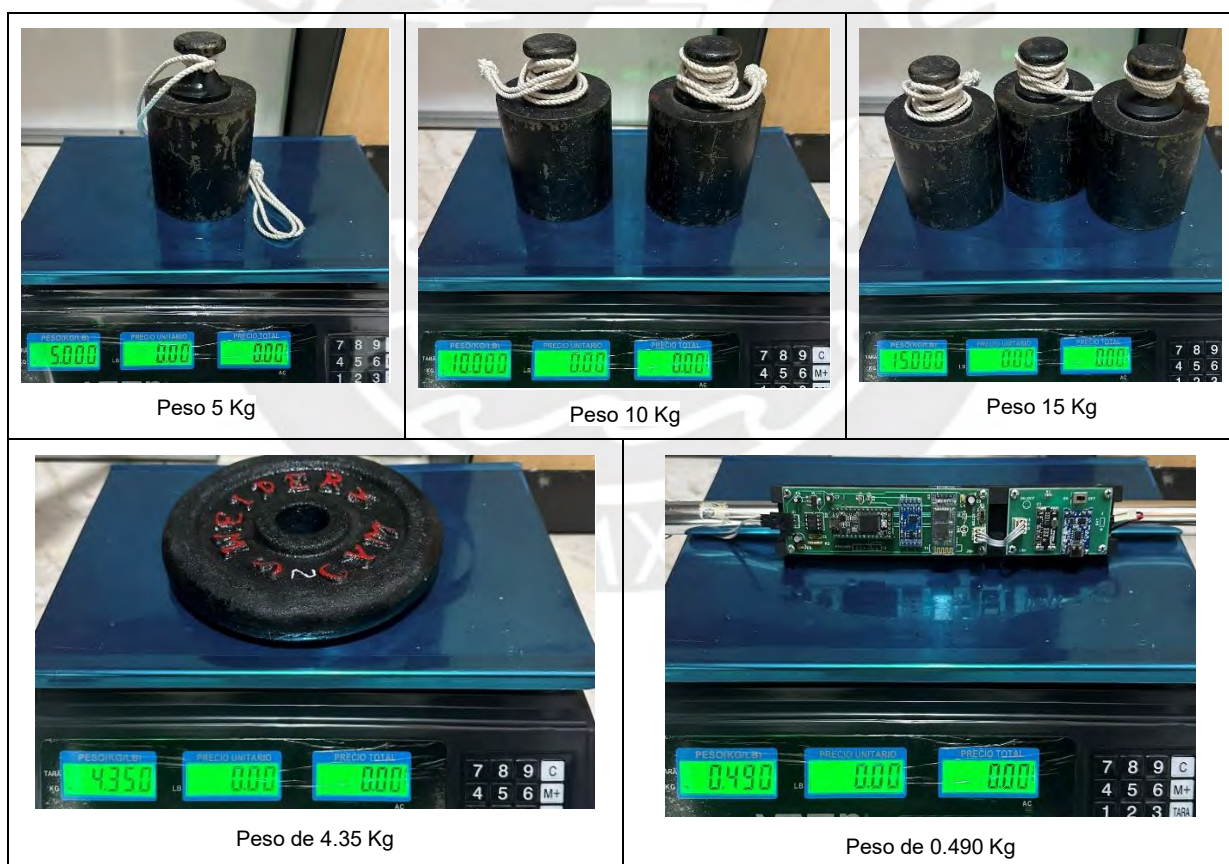


Figura 4-2. Pesas para validación de la respuesta del sensor de fuerza en el bastón.

(Fuente: elaboración propia).

Para realizar las pruebas y evitar que el bastón se caiga con el peso aplicado sobre la empuñadura, se utilizó una prensa y un dispositivo impreso en 3D, similar a un gancho.



Figura 4-3. Validación de los pesos sobre el bastón (Fuente: elaboración propia).

Leyenda:

- (a) Gancho impreso en 3D para soporte del bastón.
- (b) Prensa con el gancho.
- (c) Prensa sosteniendo el bastón.
- (d) Prensa sosteniendo el bastón con un peso de 5 Kg.

- Aplicativo “Serial Bluetooth Terminal” disponible gratuitamente en “Play Store” del teléfono con sistema operativo Android. A continuación, se muestra en la Figura 4-4, el ícono del aplicativo y la pantalla Terminal conectándose con el bastón 2 para realizar las pruebas de validación del peso.

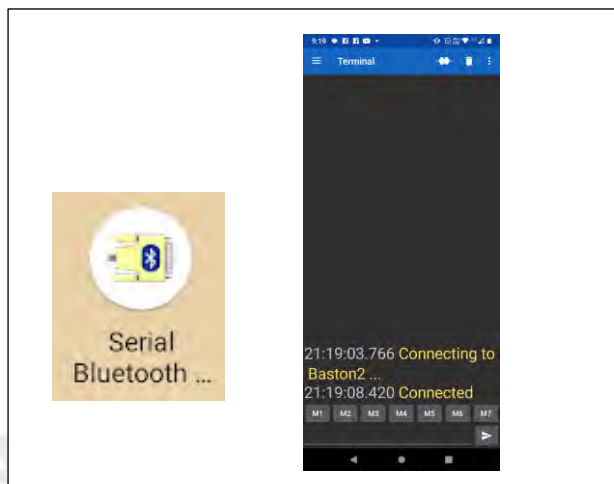


Figura 4-4 . Imagen del AppMóvil "Serial Bluetooth" cuando está conectado (Fuente: adaptación propia).

Cómo siguiente paso, se colocará **sobre la empuñadura del bastón** los pesos con una soga delgada (ver Figura 4-2.), de esta manera se puede corroborar el peso detectado por el sensor de fuerza ubicado en el regatón del bastón y es enviado por Bluetooth al teléfono celular, también en paralelo se visualiza en una balanza digital cuanto marca.

1. Pesa de 5 Kg más peso del bastón 490 gr = peso total 5.49 Kg.

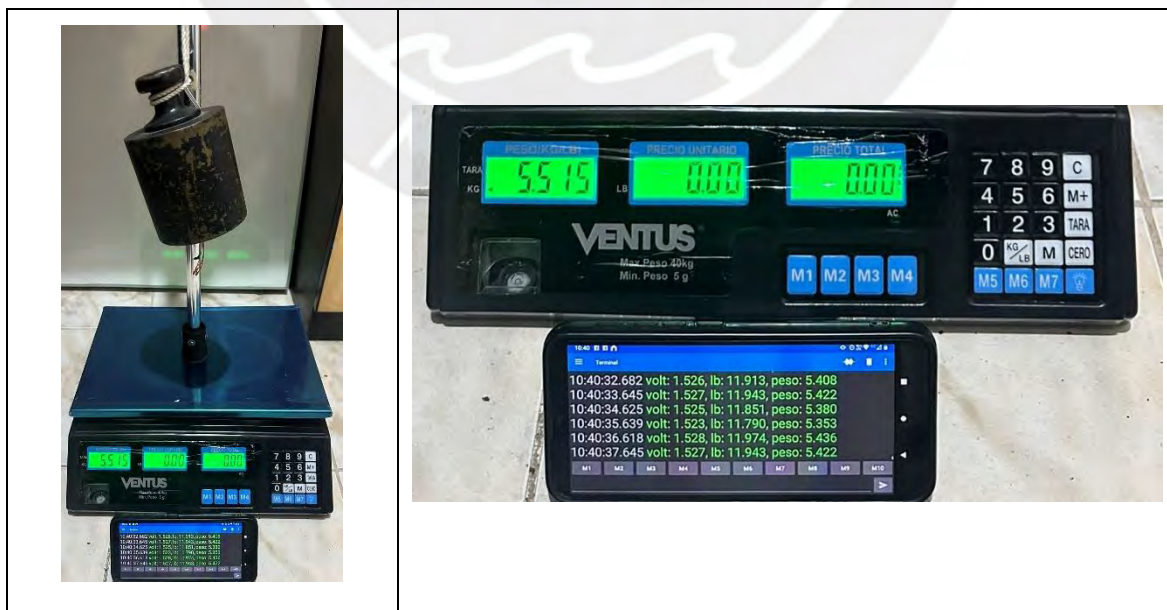


Figura 4-5. Comprobación del bastón con un peso de 5.49 Kg (Fuente: elaboración propia).

Tabla 4-1 Valores registrados por la balanza y teléfono celular para un peso de prueba 5.49 Kg.

Peso por verificar (Kg)	Peso balanza (Kg)	Peso celular (Kg)
5.490	5.515	5.408
		5.422
		5.380
		5.353
		5.436
		5.422
<b>Peso promedio =</b>		<b>5.403</b>

Analizando los valores de la Tabla 4-1, se observa que existe diferencias entre el peso por verificar (5.490 Kg) y las mostradas por la balanza (5.515 Kg) y celular (5.403 Kg) que recibe los datos del CPU del bastón transmitidos por Bluetooth. De ambas lecturas, nos interesa analizar la arrojada por el celular, para ello vamos a calcular sus errores absolutos y porcentuales:

Según el fabricante, las lecturas tienen un error de linealidad de  $\pm 3\%$  de full scale:

$$\rightarrow \pm 3\% \text{ de } 25 \text{ lb} = \pm 0.75 \text{ lb} \equiv \pm 340 \text{ gr} \dots\dots\dots (4.1)$$

- Valor teórico: 5490 gr
- Valor medido: 5403 gr

$$\rightarrow \text{Error absoluto} = |\text{Valor teórico} - \text{Valor medido}| = |5490 - 5403| = 87 \text{ gr} \dots (4.2)$$

$$\rightarrow \text{Error porcentual} = (\text{Error absoluto} / \text{Valor teórico}) * 100$$

$$= (87/5490) * 100 = 1.58\% \dots\dots\dots (4.3)$$

Si bien es cierto, el valor que muestre la balanza es referencial, vamos a calcular su error:

- Valor teórico: 5490 gr
- Valor medido: 5515 gr

$$\rightarrow \text{Error absoluto} = |\text{Valor teórico} - \text{Valor medido}| = |5490 - 5515| = 25 \text{ gr} \dots (4.4)$$

$$\rightarrow \text{Error porcentual} = (\text{Error absoluto} / \text{Valor teórico}) * 100$$

$$= (25/5490) * 100 = 0.45\% \dots\dots\dots (4.5)$$

Comparando los resultados (4.1) y (4.2), se observa que la desviación es de 87 gr, valor que está comprendido dentro del rango  $\pm 340$  gr que da el fabricante como error a full scale. Por lo que se considera aceptable esta medida.

2. Pesa de 10 Kg, más peso bastón 490 gr = peso total 10490 gr.



Figura 4-6. Comprobación del bastón con un peso de 10.490 Kg (Fuente: elaboración propia).

Tabla 4-2. Valores registrados por la balanza y bastón para un peso de prueba 10.490 Kg.

Peso por verificar (Kg)	Peso balanza (Kg)	Peso celular (Kg)
10.490	10.335	10.227
		10.472
		10.500
		10.444
		10.416
		10.486
	<b>Peso promedio =</b>	<b>10.424</b>

A continuación, se mostrará una tabla con los errores absolutos y porcentuales para el bastón y balanza.

Tabla 4-3. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 10.490 Kg. (Fuente: elaboración propia)

	Peso celular (Bastón)	Balanza
<b>Valor teórico (gr)</b>	10490	10490
<b>Valor medido (gr)</b>	10424	10335
<b>Error absoluto (gr)</b>	66	155
<b>Error porcentual (%)</b>	0.63	1.48

Analizando el resultado de la tabla anterior, se observa que el error absoluto en la lectura del bastón es de 66 gr, valor que estaría dentro del rango del sensor de  $\pm 340$  gr.

3. Pesa de 15 Kg, más peso bastón 490 gr = peso total 15490 gr.



Figura 4-7. Comprobación del bastón con un peso de 15.490 Kg (Fuente: elaboración propia).

Tabla 4-4. Valores registrados por la balanza y teléfono celular para un peso de prueba 15.49 Kg.

Peso por verificar (Kg)	Peso balanza (Kg)	Peso celular (Kg)
15.490	15.420	15.332
		15.263
		15.472
		15.472
		15.332
		15.332
	<b>Peso promedio =</b>	<b>15.367</b>

A continuación, se mostrará una tabla con los errores absolutos y porcentuales para el bastón y balanza.

Tabla 4-5. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 15.490 Kg. (Fuente: elaboración propia)

	Peso celular (Bastón)	Balanza
Valor teórico (gr)	15490	15490
Valor medido (gr)	15367	15420
Error absoluto (gr)	123	70
Error porcentual (%)	0.79	0.45

Analizando el resultado de la tabla anterior, se observa que el error absoluto en la lectura del bastón es de 123 gr, valor que estaría dentro del rango del sensor de  $\pm 340$  gr.

4. Pesa de 15 Kg, más pesa de 4.35 Kg, más peso bastón 490 gr = peso total 19840 gr.



Figura 4-8. Comprobación del bastón con un peso de 19.840 Kg (Fuente: elaboración propia).

Tabla 4-6. Valores registrados por la balanza y teléfono celular para un peso de prueba 19.840 Kg.

Peso por verificar (Kg)	Peso balanza (Kg)	Peso celular (Kg)
19.840	19.740	19.339
		19.235
		19.200
		19.444
		19.409
		19.235
	<b>Peso promedio =</b>	<b>19.310</b>

A continuación, se mostrará una tabla con los errores absolutos y porcentuales para el bastón y balanza.

Tabla 4-7. Errores absoluto y porcentual de la balanza y bastón para un peso de prueba de 19.840 Kg. (Fuente: elaboración propia)

	Peso celular (Bastón)	Balanza
Valor teórico (gr)	19840	19840
Valor medido (gr)	19310	19740
Error absoluto (gr)	530	100
Error porcentual (%)	2.67	0.50

En la siguiente tabla, se mostrará los resultados de las pruebas de respuesta del sensor de fuerza de 25 libras colocado en el regatón del bastón.

Tabla 4-8 Resultados del sensor de fuerza del bastón para diferentes pesos.  
(Fuente: elaboración propia)

# Prueba	V.T. (gr)	V.M. (gr)	E.A. (gr)	E.P. (%)	Leyenda:	
1	5490	5403	87	1.58	V.T.	Valor Teórico
2	10490	10424	66	0.63	V.M.	Valor Medido
3	15490	15367	123	0.79	E.A.	Error Absoluto
4	19840	19310	530	2.67	E.P.	Error Porcentual

Analizando el resultado mostrado en la Tabla 4-8, podemos concluir que la respuesta del sensor dentro del rango de las 25 libras (0 a 11 Kg) es bastante aceptable porque está dentro de los  $\pm 340$  gr, se adicionó 10 lb más (35 lb  $\equiv$  16 Kg) y el error fue de 123 gr que también está dentro de los  $\pm 340$  gr, finalmente se adicionó 8 lb más, totalizando un peso de 43.74 lb (19.84 Kg) y su error fue de 530 gr, ósea, 190 gr más con respecto a los 340 gr.

En conclusión, se puede afirmar que el sensor de fuerza responde satisfactoriamente a la fuerza aplicada sobre el bastón hasta las 35 libras.

A continuación, se muestra un gráfico de la fuerza aplicada sobre el bastón vs error absoluto para determinar cuál sería el error para una persona que tiene un peso de 70 Kg. Para ello, se va a considerar el 10% (7 Kg), 15% (10.5 Kg) y 20% (14 Kg) de su peso. Observando el gráfico, se puede deducir que el error para estos porcentajes estaría dentro de los 100 gr, y dentro del margen de error dado por el fabricante de  $\leq \pm 3\%$  de la escala completa de 25 libras, ósea,  $\pm 340$  gr.



Figura 4-9. Gráfico de Fuerza aplicada sobre el bastón vs Error absoluto (Fuente: elaboración propia)

#### 4.1.2. Acelerómetro ADXL345 – RTC DS3231

En este caso, se ha considerado conveniente mostrar el funcionamiento de ambos, ya que en cada movimiento del bastón se va a registrar los ángulos Pitch y Roll, así como la fecha y hora. En la Figura 3-28, se muestra el esquemático del circuito electrónico de comunicación entre el CPU - Acelerómetro – DS3231 y en la Figura 3-29. Circuitos impresos del CPU más periféricos y alimentación con batería, el circuito impreso montado sobre la caña del bastón.

A continuación, se explica cómo se calcula los ángulos Pitch y Roll de manera estática, tomando como referencia el eje X con  $0^\circ$ :

##### Ángulo Pitch:

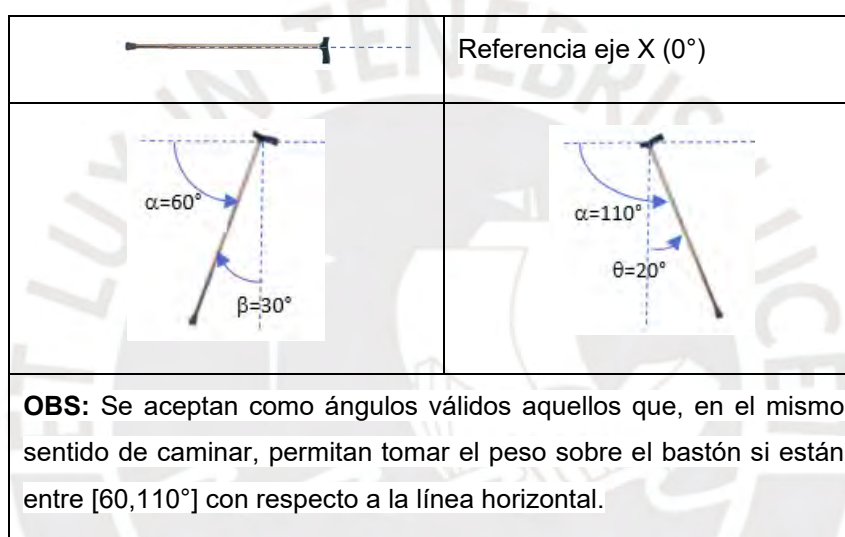


Figura 4-10. Posición del bastón con respecto al ángulo Pitch (Fuente: elaboración propia)

##### Ángulo Roll:

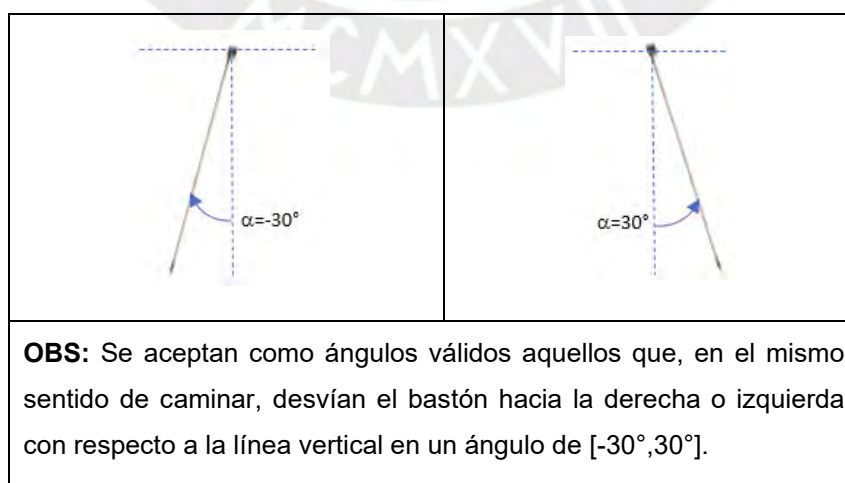






Figura 4-11. Posición del bastón con respecto al ángulo Roll (Fuente: elaboración propia)

Tomando como referencia la Figura 4-10 y Figura 4-11, los siguientes cuadros muestran los valores  $g_x$ ,  $g_y$  y  $g_z$  que arroja el acelerómetro. Con estos valores, se calcularán los ángulos Pitch y Roll que se usan en este proyecto de tesis.

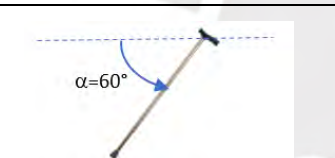

Bastón modo horizontal  $0^\circ$  ( $^\circ$ Pitch), paralelo al eje X:

		$g_x = -0.00$ $g_y = -0.00$ $g_z = 1.00$ $^\circ$ Pitch= -0.02 $^\circ$ Roll= -0.05 $g_x = -0.00$ $g_y = -0.00$ $g_z = 1.00$ $^\circ$ Pitch= -0.02 $^\circ$ Roll= -0.05 $g_x = -0.00$ $g_y = -0.00$ $g_z = 1.00$ $^\circ$ Pitch= -0.02 $^\circ$ Roll= -0.27 $g_x = -0.01$ $g_y = -0.00$ $g_z = 1.00$ $^\circ$ Pitch= -0.47 $^\circ$ Roll= -0.05
---	---	--

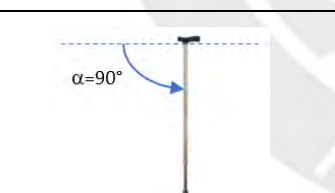

Bastón posicionado a  $30^\circ$  ( $^\circ$ Pitch) con respecto a la horizontal:

		$g_x = 0.51$ $g_y = -0.05$ $g_z = 0.87$ $^\circ$ Pitch= 30.23 $^\circ$ Roll= -2.73 $g_x = 0.51$ $g_y = -0.04$ $g_z = 0.87$ $^\circ$ Pitch= 30.34 $^\circ$ Roll= -2.28 $g_x = 0.51$ $g_y = -0.04$ $g_z = 0.87$ $^\circ$ Pitch= 30.23 $^\circ$ Roll= -2.50 $g_x = 0.51$ $g_y = -0.04$ $g_z = 0.87$ $^\circ$ Pitch= 30.23 $^\circ$ Roll= -2.50
---	---	--

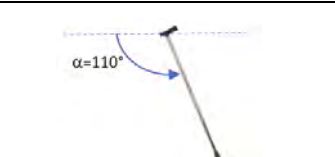

Bastón a  $60^\circ$  ( $^\circ$ Pitch) con respecto a la horizontal:

		$g_x = 0.91$ $g_y = 0.03$ $g_z = 0.52$ $^\circ$ Pitch= 60.19 $^\circ$ Roll= 1.62 $g_x = 0.90$ $g_y = 0.03$ $g_z = 0.51$ $^\circ$ Pitch= 60.65 $^\circ$ Roll= 1.62 $g_x = 0.91$ $g_y = 0.02$ $g_z = 0.52$ $^\circ$ Pitch= 60.40 $^\circ$ Roll= 0.95 $g_x = 0.91$ $g_y = 0.03$ $g_z = 0.52$ $^\circ$ Pitch= 60.01 $^\circ$ Roll= 1.84
---	---	--

Bastón a  $90^\circ$  ( $^\circ$ Pitch) con respecto a la horizontal:

		$g_x = 1.04$ $g_y = -0.01$ $g_z = -0.01$ $^\circ$ Pitch= 90.45 $^\circ$ Roll= -0.29 $g_x = 1.04$ $g_y = -0.01$ $g_z = -0.01$ $^\circ$ Pitch= 90.45 $^\circ$ Roll= -0.73 $g_x = 1.04$ $g_y = -0.01$ $g_z = -0.02$ $^\circ$ Pitch= 90.88 $^\circ$ Roll= -0.29 $g_x = 1.04$ $g_y = -0.01$ $g_z = -0.01$ $^\circ$ Pitch= 90.45 $^\circ$ Roll= -0.51
---	---	--

Bastón a  $110^\circ$  ( $^\circ$ Pitch) con respecto a la horizontal:

		$g_x = 0.97$ $g_y = -0.03$ $g_z = -0.36$ $^\circ$ Pitch= 110.12 $^\circ$ Roll= -1.54 $g_x = 0.96$ $g_y = -0.03$ $g_z = -0.35$ $^\circ$ Pitch= 110.07 $^\circ$ Roll= -1.99 $g_x = 0.97$ $g_y = -0.02$ $g_z = -0.36$ $^\circ$ Pitch= 110.05 $^\circ$ Roll= -1.10 $g_x = 0.97$ $g_y = -0.00$ $g_z = -0.35$ $^\circ$ Pitch= 109.79 $^\circ$ Roll= -0.20
---	---	--

Con los valores obtenidos del acelerómetro para los ángulos de 0°, 30°, 60°, 90° y 110°, se va a calcular el error de medición y constatar con la hoja de datos del fabricante, que indica lo siguiente: “El ADXL345 es muy adecuado para medir la aceleración estática de la gravedad en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento. Su alta resolución (3,9 mg/LSB) permite la medición de inclinación con una precisión angular en hasta ±1°.”

Tabla 4-9. Error de medición del ángulo de inclinación Pitch.

(Fuente: elaboración propia)

Columna	Ángulo= 0°	Ángulo= 30°	Ángulo= 60°	Ángulo= 90°	Ángulo= 110°
1	-0,02	30,23	60,19	90,45	110,12
2	-0,02	30,34	60,65	90,45	110,07
3	-0,02	30,23	60,40	90,88	110,05
4	-0,47	30,23	60,01	90,45	109,79
<b>Lectura Promedio</b>	-0,13	30,26	60,31	90,56	110,01
<b>E.A.</b>	0,13	0,26	0,31	0,56	0,01
<b>E.P.</b>		0,86%	0,51%	0,62%	0,01%
<b>E.A. =  Valor teórico - Valor medido </b> <b>E.P. = (Error absoluto / Valor teórico)*100</b>					

Al analizar la Tabla 4-9., se observa que el error absoluto para los distintos ángulos es menor a 1°, y el error porcentual es inferior al 1%. Por lo tanto, se concluye que los valores obtenidos están dentro del margen de error especificado por el fabricante.

A continuación, se muestra en el Monitor Serie del Arduino el funcionamiento del reloj en tiempo real mostrando la fecha y hora, así como los valores transmitidos por el acelerómetro ante distintos movimientos del bastón, considerando la restricción de Pitch [60°, 110°] y Roll [-30°, 30°], y sin restricciones.

(a)

(b)

Figura 4-12. Resultados del Acelerómetro ADXL345 y RTC DS3231 (Fuente: elaboración propia).

En este cuadro se puede apreciar el acelerómetro con restricciones (a) y sin restricciones (b), de los ángulos para diferentes movimientos del bastón inteligente.

### 4.1.3. Memoria MicroSD

En la Figura 4-13, en el lado izquierdo, se muestra el contenido de todos los archivos grabados en la memoria MicroSD. En el lado derecho, se muestra una parte del historial de los datos grabados sobre cómo se ha venido usando el bastón por el paciente.

Los datos se han registrado en el siguiente orden: Fecha, Hora, Pitch (°), Roll (°) y Peso (Kg), todos separados por comas para poder ser exportado en formato .CSV hacia el computador y luego ser transformado a una hoja Excel.

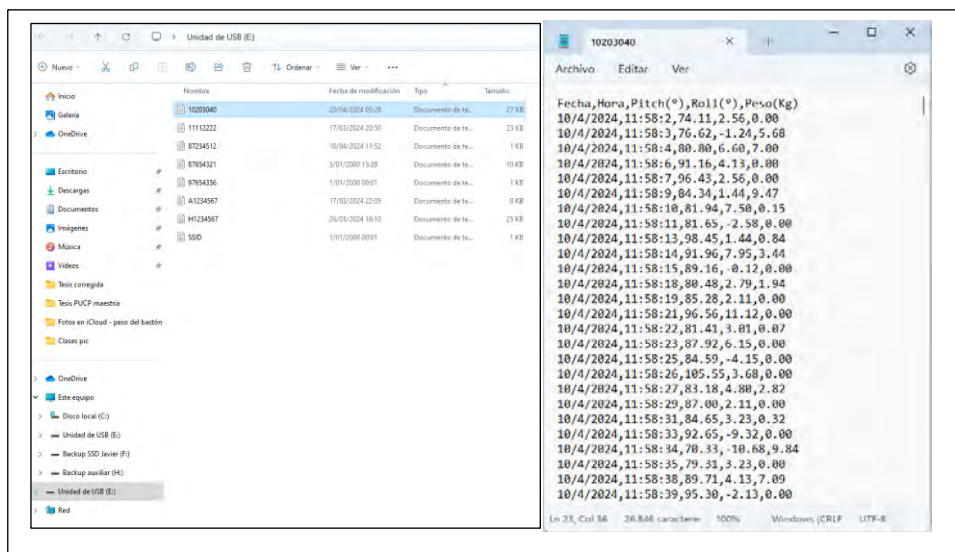


Figura 4-13. Datos grabados en la memoria MicroSD sobre uso del bastón inteligente (Fuente: elaboración propia).

## 4.2. Prueba del sistema de comunicación y software

Para mostrar el funcionamiento del sistema de comunicación por Bluetooth entre el bastón y la AppMóvil, se sigue los siguientes pasos:

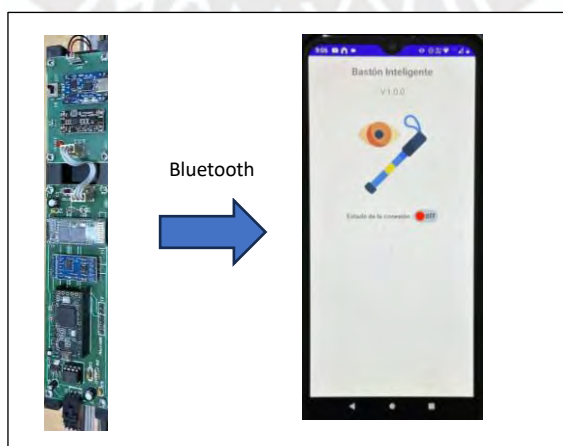


Figura 4-14. Prueba de comunicación por Bluetooth entre Bastón inteligente y Celular. (Fuente: elaboración propia)

1. El médico tratante solicita al paciente su teléfono celular con sistema operativo Android, ingresa a configuración, dispositivos conectados, se debe detectar dispositivos Bluetooth próximos con el nombre "BastonX" (X=1, 2, ..., n), se selecciona el nombre correspondiente con el bastón y se lo vincula o agrega al teléfono para luego poder comunicarse con el aplicativo. La siguiente figura muestra el bastón 2 vinculado al teléfono.

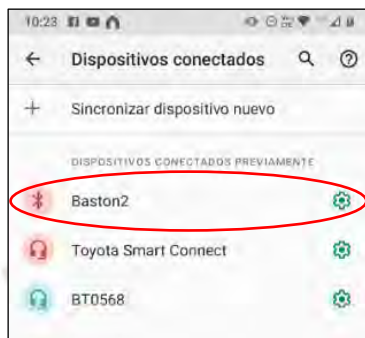


Figura 4-15. Imagen de Baston2 vinculado al teléfono celular (Fuente: elaboración propia).

2. Se instala la aplicación al teléfono celular del paciente. Se debe mostrar el ícono como el encerrado dentro del círculo rojo.

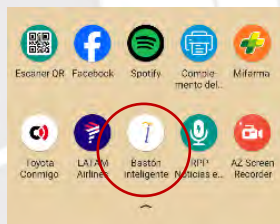


Figura 4-16. Icono del Aplicativo Móvil instalado en el celular (Fuente: elaboración propia).

3. Para comunicarse con el bastón, se debe dar doble clic en el icono del aplicativo, luego debe aparecer la pantalla inicial o de enlace previa a la conexión con el BT.



Figura 4-17. Pantalla inicial o de enlace.

4. Dar clic al botón “OFF” de “Estado de la conexión” de la Figura 4-17 para que se establezca la comunicación entre bastón y celular. Aparece la pantalla de Menú Principal.



Figura 4-18. Pantalla Menú Principal del AppMóvil (Fuente: elaboración propia).

5. Hacer clic en el botón “Nuevo paciente”. Completar toda la información obligatoria solicitada para el registro del nuevo usuario del bastón. Se ha registrado a una persona con un peso de 70 Kg, dato que servirá para analizar los valores almacenados y compartidos en formato .CSV.

Figura 4-19. Registro de nuevo paciente con datos completos.

6. Hacer clic en el botón “Mostrar datos y crear Excel”, para transferir todos los datos almacenados en la memoria MicroSD del bastón y se mostrará como en la Figura 3-35. Luego, si cree conveniente transferir toda esta información al computador y convertirla a formato Excel para ser anexada a su historia clínica o visualizarla en modo gráfico de distintas maneras, dar clic en el siguiente ícono:



A continuación, le solicitará que seleccione cómo desea compartir la información del archivo .TXT:

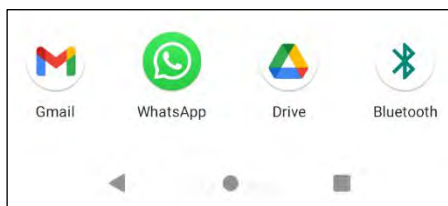


Figura 4-20. Posibles alternativas de compartir información del bastón.

El archivo que se descarga en el computador está compuesto por tres campos: palabra bastón - número de Historia Clínica - dato aleatorio que coloca el sistema Android.

Nombre	Fecha de modificación	Tipo	Tamaño
baston-18022910-2148596146514893918	13/07/2024 10:24	Documento de te...	3 KB

Fecha, Hora, °Pitch, °Roll, Peso (Kg)  
 13/7/2024, 9:32:33, 65.16, 15.32, 2.67  
 13/7/2024, 9:32:34, 81.98, 22.41, 4.81  
 13/7/2024, 9:32:36, 88.55, 8.69, 4.34  
 13/7/2024, 9:32:45, 61.21, 3.75, 7.52

Figura 4-21. Nombre del archivo transferido en formato .TXT (Fuente: elaboración propia)

Con la información descargada en el computador, abra Excel y siga estos pasos:

- Clic en la pestaña de “Datos”.
- Seleccione la opción “De texto/CSV”, ubique el archivo descargado en formato TXT y luego dar clic en importar. Se mostrará como el de la siguiente figura.

Figura 4-22. Pantalla en Excel con los datos precargados.

- De clic en cargar y se mostrarán los datos listos para ser graficados.

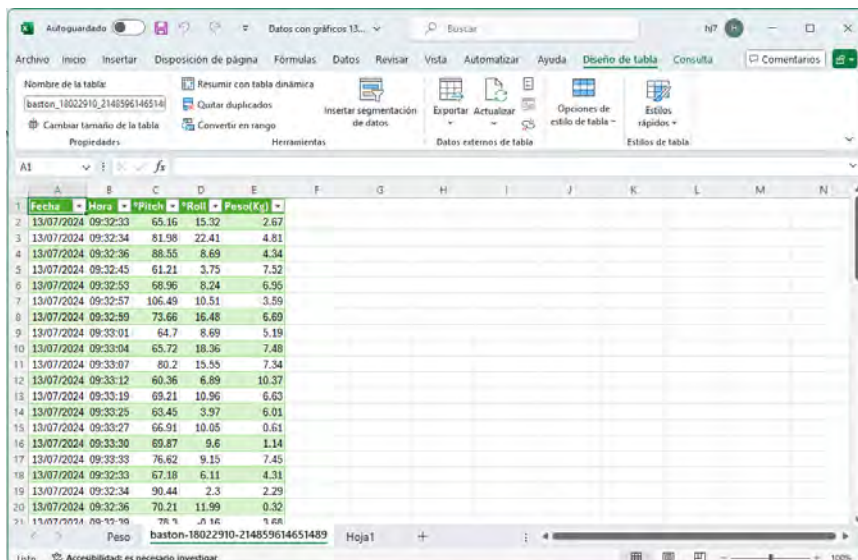


Figura 4-23. Datos importados (.TXT) y convertidos en forma de tabla de Excel.

### 4.3. Pruebas funcionales en el bastón electrónico en modo normal o diario

Para realizar estas pruebas, se ha creído conveniente utilizar un analizador lógico dentro del loop para saber cuáles son los tiempos que se toma en capturar una muestra del peso más sus ángulos pitch y roll, y adicionalmente el tiempo que toma en almacenar toda la información capturada en la memoria SD.

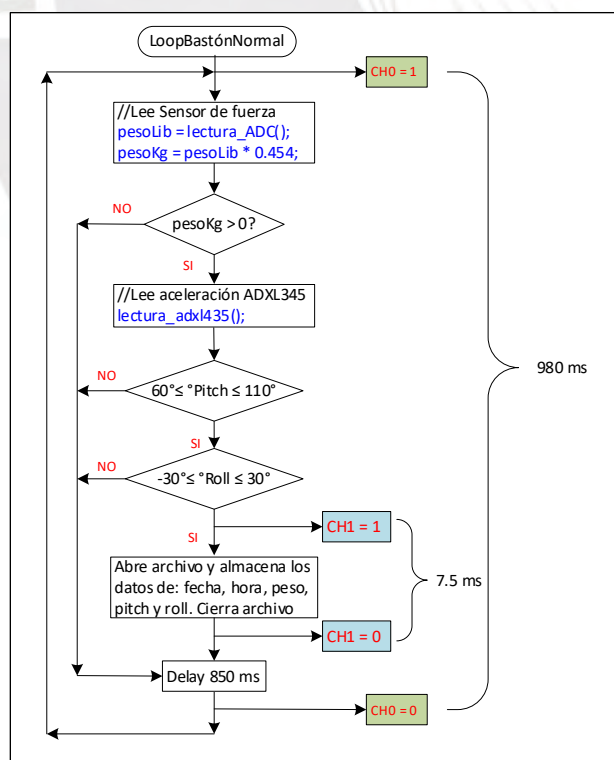


Figura 4-24. Diagrama de bloques de la toma de muestras del bastón en modo normal.

(Fuente: elaboración propia)

Se utilizó el analizador lógico para saber exactamente cuánto tiempo se tarda en procesar la información y calcular el tiempo que se debe dar para completar el periodo de muestreo de 1 segundo aproximadamente. Si el peso es menor o igual que cero, espera 850 ms para volver a tomar otra muestra. El software se obtiene de: <https://www.saleae.com/downloads/>

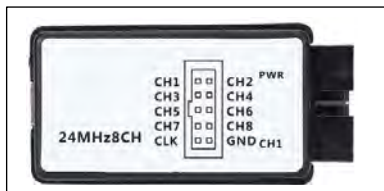


Figura 4-25. Imagen del analizador lógico (Fuente: <https://hifisac.com/en>)

Las señales que se muestran a continuación corresponden al diagrama de la Figura 4-24.

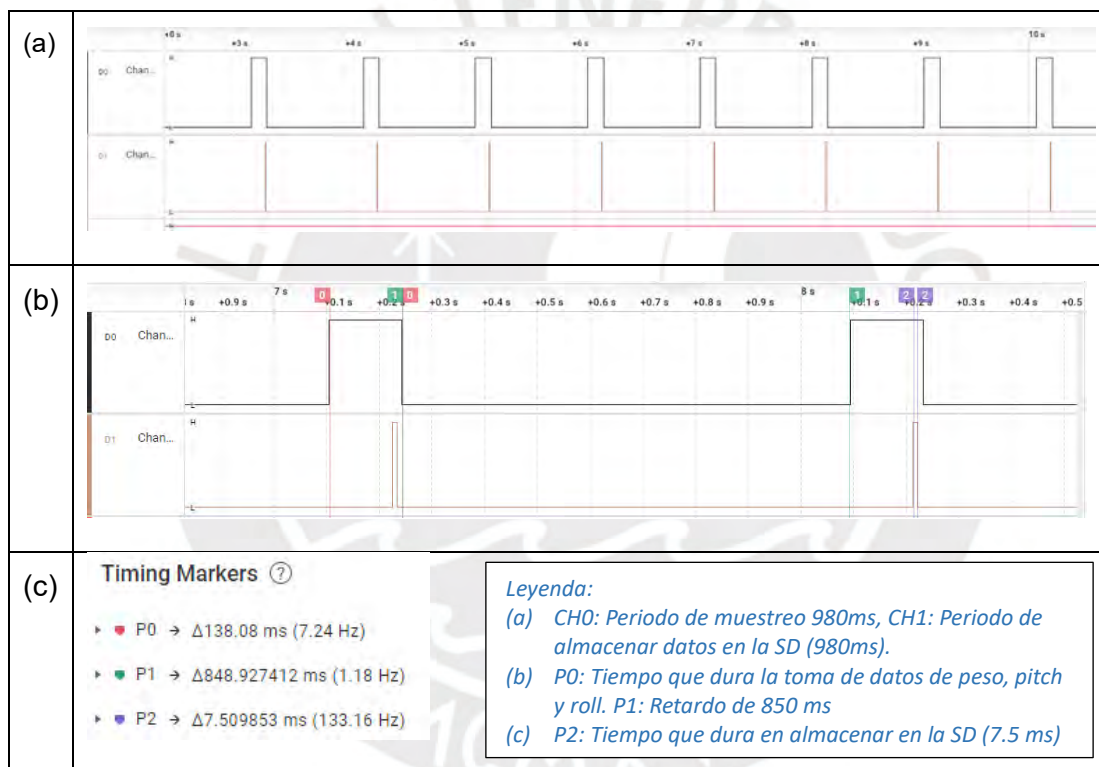
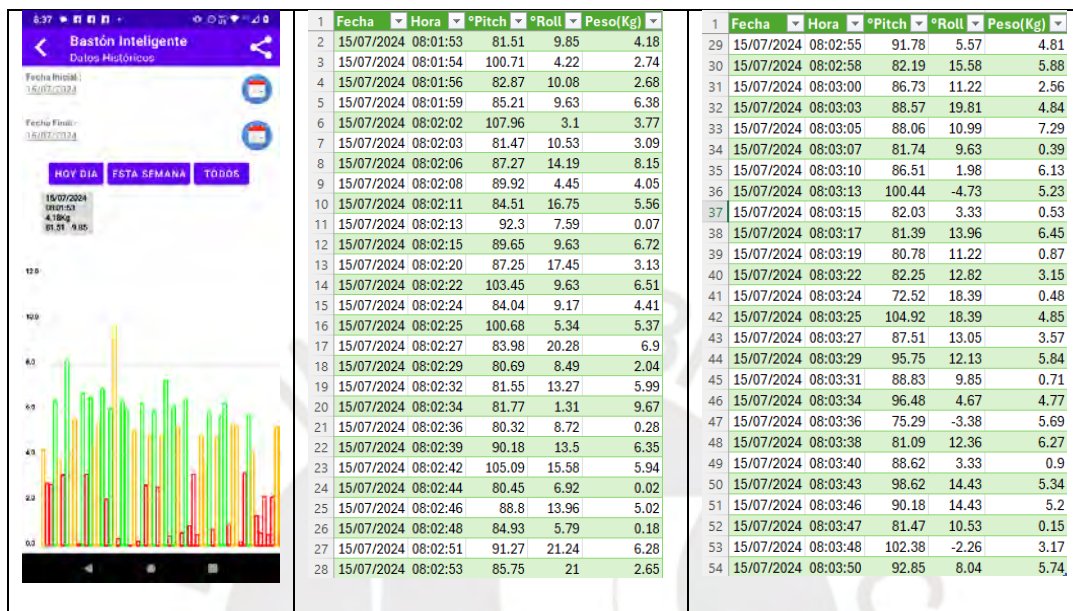


Figura 4-26. Diagrama de tiempos generados durante el muestreo de datos (Fuente: elaboración propia)

### 4.3.1. Prueba cuando Peso > 0

En la siguiente tabla, se muestran una parte de los datos visto en el App del celular y luego exportados del celular al Excel de la computadora para graficar y realizar el análisis respectivo.

Tabla 4-10. Datos importados del celular cuando el peso ejercido sobre el bastón es mayor que cero.



El gráfico de señales Peso, °Pitch, °Roll y Hora mostrado en la Figura 4-27, corresponde a una persona que recién se está entrenando en el uso del bastón. Se puede observar que la fuerza aplicada sobre el bastón varía desde un peso mayor que cero hasta cerca de 9.67 Kg. Si la persona pesa 70 Kg según los datos registrados del paciente en la Figura 4-19, los 9.67 Kg representan el 13.81%, que estaría un 3.81% por encima del rango dado por las investigaciones realizadas en la Tesis Doctoral de Fernández, María del Carmen Ridao [10], que indica una fuerza máxima de 10% del peso del paciente, pero dentro del rango según Melis EH. et al. [11] en su estudio sobre el bastón de mano, donde recalca que este bastón se debe utilizar como máximo hasta un 15% de soporte vertical sobre él, porque se vuelve inestable cuando se carga a partir del 20-25% del peso corporal. Para un porcentaje mayor al 25% de su peso sobre el bastón, se recomienda ser supervisado por su médico o terapeuta, porque una técnica correcta para usar el bastón será importante para asegurar que la distribución del peso y la fuerza aplicada no causen estrés adicional a la pierna afectada o a otras partes del cuerpo.

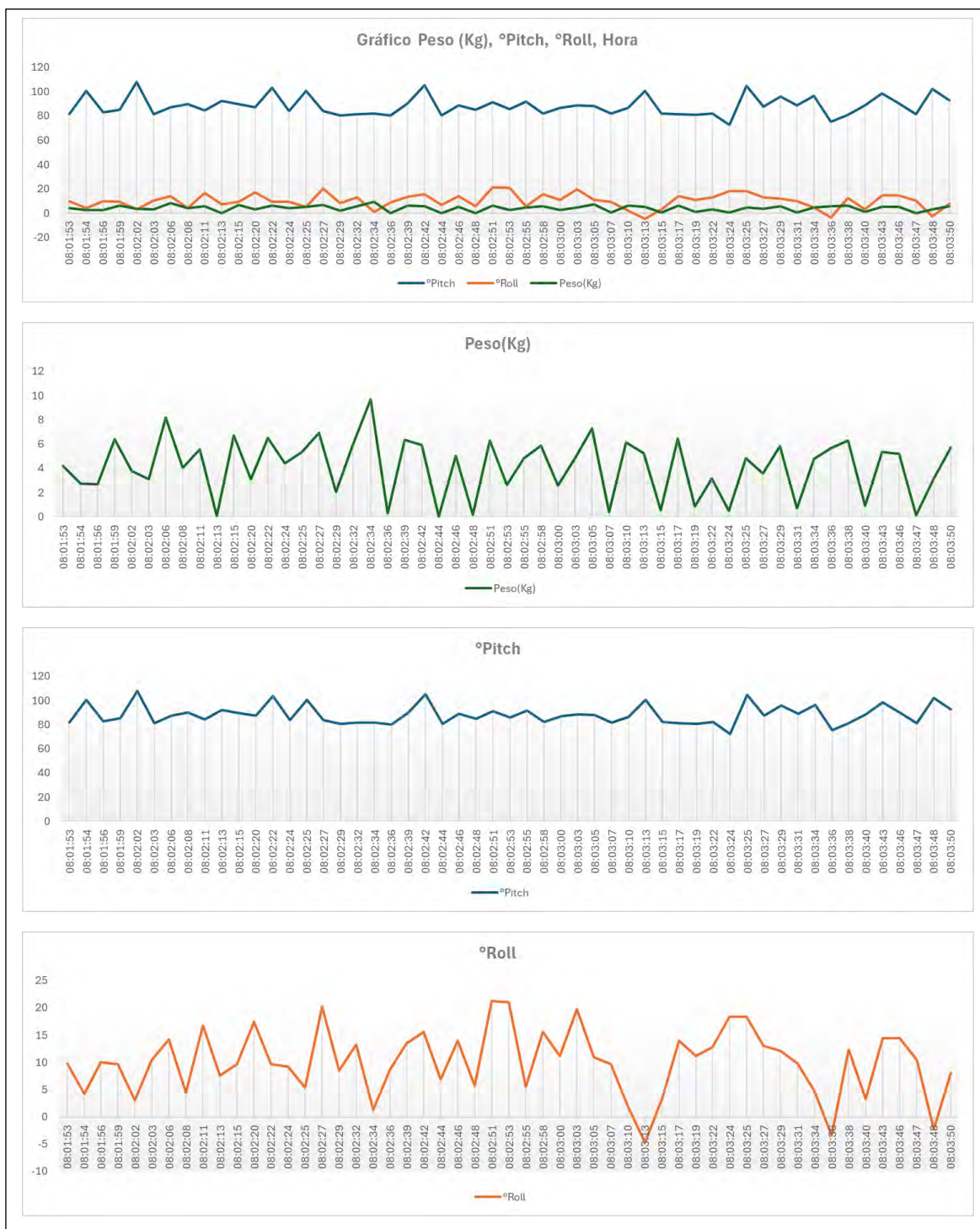


Figura 4-27. Gráfico del Peso, Pitch y Roll con peso mayor que CERO (Fuente: elaboración propia)

A continuación, en la Figura 4-28 se explicarán o interpretarán en detalle los primeros 10 datos de la Tabla 4-10, correspondiente al Peso, °Pitch y °Roll. Esta explicación, servirá para interpretar los próximos gráficos que se presenten.



Figura 4-28. Interpretación de las señales generadas por el bastón durante la marcha (Fuente: elaboración propia)

**Punto 1** (Figura 4\_28(a)): El paciente inicia su marcha asentado el bastón sobre el piso con un peso de 4.18 Kg, con  $81.51^\circ$  hacia adelante y  $9.85^\circ$  hacia su derecha. Sigue caminando hacia el **punto 2**, ejerciendo un peso de 2.74 Kg, con un  $100.71^\circ$  (bastón detrás del paciente en  $10.71^\circ$ ) y  $4.22^\circ$  hacia su derecha.

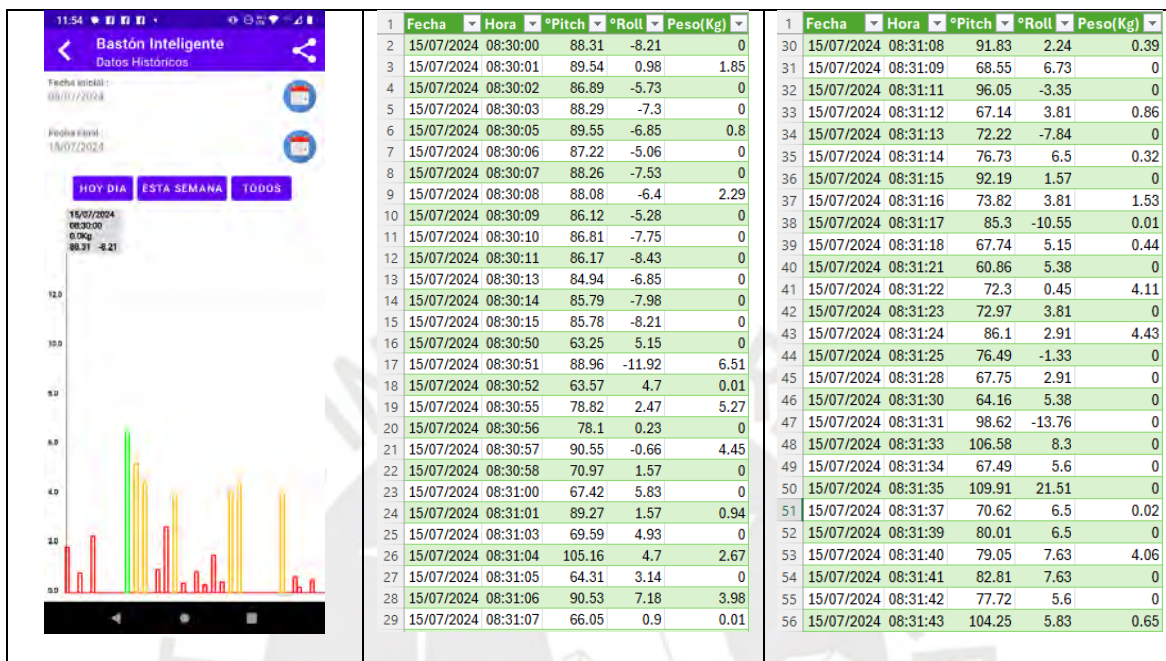
El paciente levanta el bastón hacia el **punto 3**, ejerciendo un peso de 2.68 Kg, con  $82.87^\circ$  hacia adelante y  $10.08^\circ$  hacia su derecha; sigue caminando hacia el **punto 4**, ejerciendo un peso de 6.38 Kg con  $85.21^\circ$  hacia adelante y  $9.63^\circ$  hacia su derecha; sigue caminado hacia el **punto 5**, ejerciendo un peso de 3.77 Kg con  $107.96^\circ$  (bastón detrás del paciente en  $7.96^\circ$ ) y  $3.1^\circ$  hacia su derecha.

El paciente nuevamente levanta el bastón hacia el **punto 6**, ejerciendo un peso de 3.07 Kg, con  $81.47^\circ$  hacia adelante y  $10.53^\circ$  hacia su derecha; sigue caminando hacia el **punto 7**, ejerciendo un peso de 8.15 Kg con  $87.27^\circ$  hacia adelante y  $14.19^\circ$  hacia su derecha; sigue caminado hacia el **punto 8**, ejerciendo un peso de 4.05 Kg con  $89.92^\circ$  (bastón casi en paralelo con el paciente) y  $4.45^\circ$  hacia su derecha. De igual manera, se interpreta para los puntos 9 y 10. Lo descrito en los tres párrafos anteriores, explica cómo el paciente está haciendo uso del bastón y a partir de aquí, su médico tratante o terapeuta puede analizar su evolución e indicar que siga con el tratamiento o variarlo.

### 4.3.2. Prueba cuando Peso >= 0

En la siguiente tabla, se muestran una parte de los datos visto en el App del celular y luego exportados a Excel de la computadora para graficar y realizar el análisis respectivo.

Tabla 4-11 Datos importados del celular cuando el peso ejercido sobre el bastón es >= cero.



El gráfico de señales Peso, °Pitch, °Roll y Hora mostrado en la Figura 4-29, se puede observar que la señal del peso ejercido sobre el bastón es buena y estaría dentro del límite del 10% máximo. Sin embargo, procesar con peso >=0, no es adecuado porque se estaría registrando en la memoria SD intervalos cuando el bastón está suspendido y el objetivo de este proyecto es conocer la fuerza aplicada del paciente sobre el bastón. Con respecto a las formas de las señales de los ángulos Pitch y Roll, se puede observar cómo se movió el bastón durante la marcha, validándose previamente que esté dentro de los ángulos [60, 110] y [-30, 30]. Por lo tanto, el programa quedaría que sólo se registre o grabe en la memoria SD cuando el peso es mayor que cero.

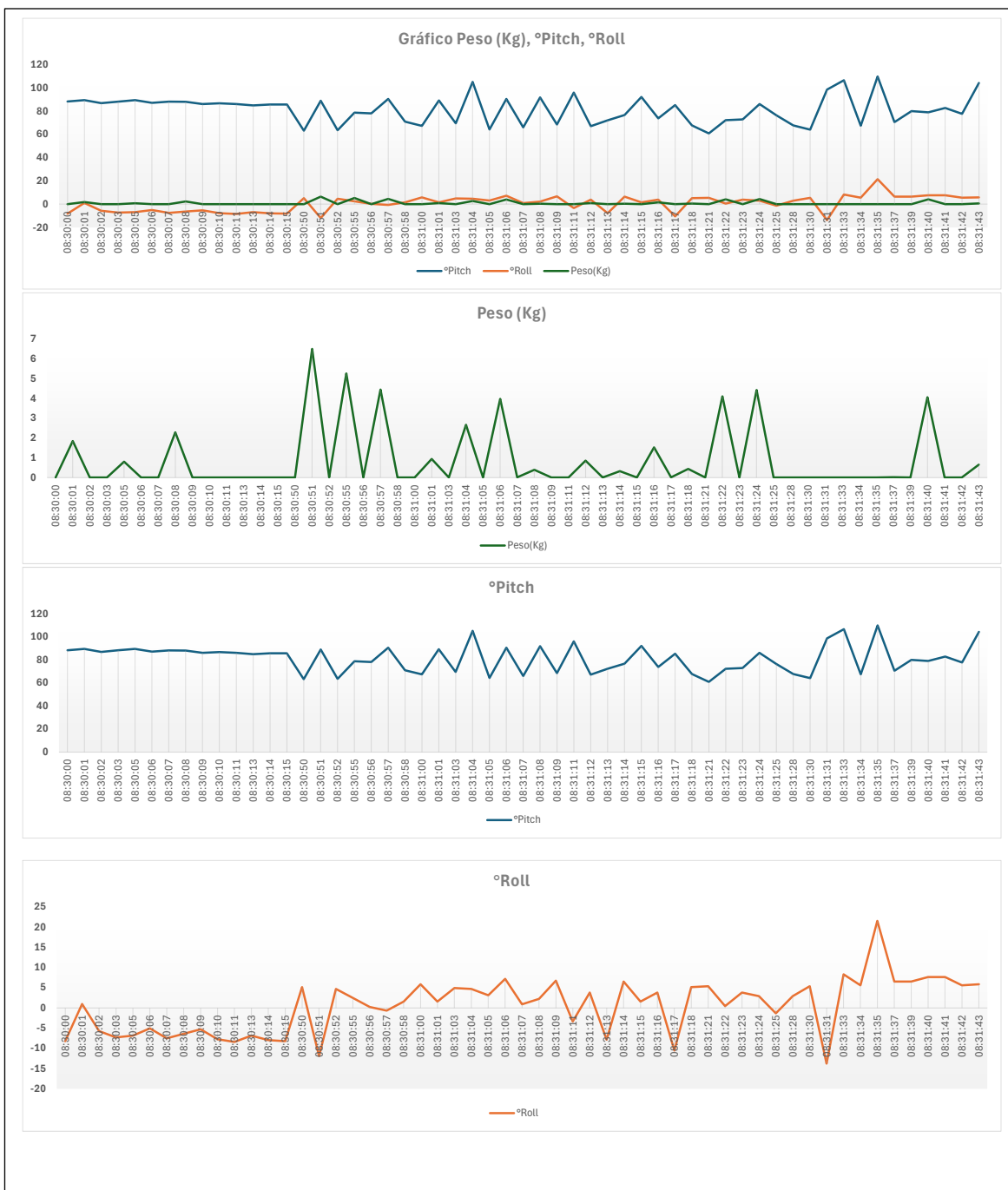


Figura 4-29. Gráfico del Peso, Pitch y Roll con peso  $\geq$  que CERO (Fuente: elaboración propia)

#### 4.4. Pruebas funcionales en el bastón electrónico en modo tiempo real

Para realizar estas pruebas, similar al ítem 4.3, se utilizó un analizador lógico dentro del loop del case ('5') correspondiente a monitor en tiempo real, para saber cuáles son los siguientes tiempos:

- En repetirse el loop dentro del case ('5'),
- Tiempo de conversión del ADC,
- Transmitir por BT del CPU hacia el APP del celular, el peso y los ángulos pitch y roll, para visualizar el movimiento del bastón,
- Almacenar o grabar toda la información capturada en la memoria SD.

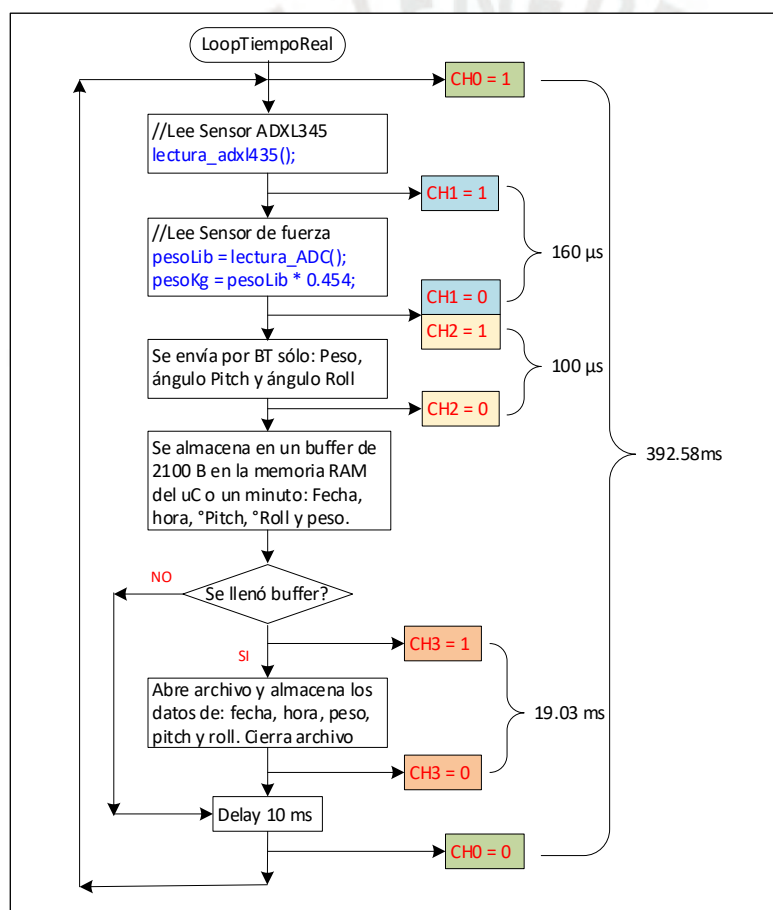


Figura 4-30. Diagrama de bloques de la toma de muestras del bastón en modo tiempo real

(Fuente: elaboración propia).

Cabe precisar, que conocer estos tiempos va a contribuir a mejorar la visualización del movimiento del bastón en la AppMóvil y así eliminar o minimizar los retardos y pequeños saltos cuando se mueve el bastón. El objetivo es que se vea el movimiento del bastón en el App similar a como lo mueve el paciente. A continuación, se muestra las señales obtenidas del analizador lógico ejecutándose en modo de tiempo real.

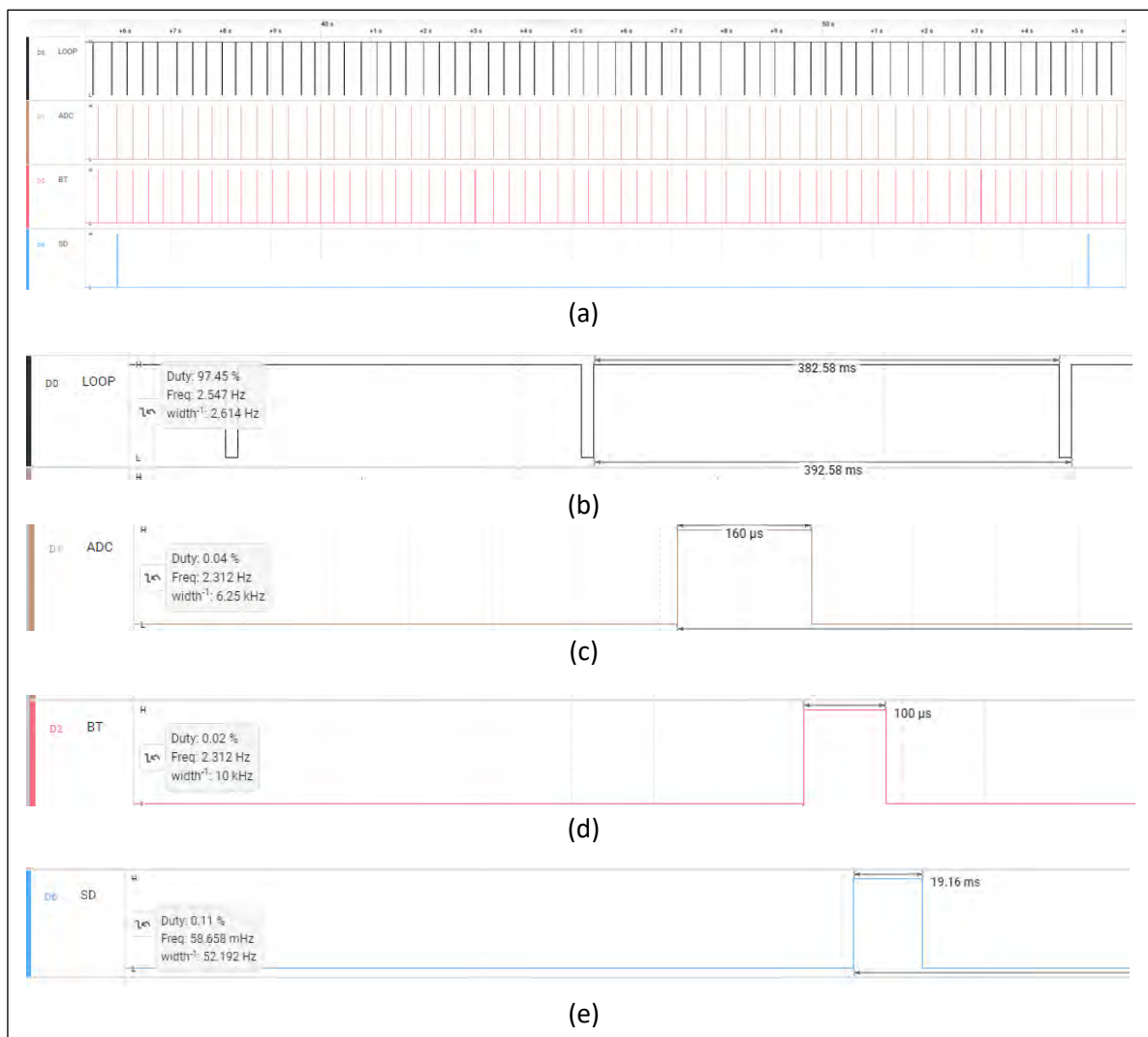


Figura 4-31. Diagrama de tiempos del proceso en tiempo real (Fuente: elaboración propia)

Descripción del diagrama de tiempos de la Figura 4\_31:

- (a) Señales generadas durante el proceso de tiempo real y en concordancia con la Figura 4-30: Loop, Tiempo conversión ADC, tiempo de transmisión por BT y tiempo de escritura en la memoria SD.
- (b) Tiempo del loop igual a 392.58ms. Es decir,  $T_{loop} = 382.58 \text{ ms} + 10 \text{ ms} = 392.58 \text{ ms}$ . Los 10 ms es de un delay introducido al final del bucle.
- (c) Tiempo de conversión del ADC igual a  $160 \mu\text{s}$  (0.16 ms).
- (d) Tiempo de transmisión por BT igual a  $100 \mu\text{s}$ . (0.10 ms).
- (e) Tiempo que se tarda en grabar 50 x 42 Bytes del archivo en memoria SD igual a 19.16 ms.

Con los tiempos encontrados, se realizó la primera prueba en tiempo real para saber cómo respondía el programa y se pudo observar que en el intervalo de tiempo de 1.2 segundos (equivalente a tres repeticiones de loop de 392.58 ms), se toma tres muestras del ADC

(Tiempo<sub>conv</sub> = 160  $\mu$ S). La Tabla 4-12, muestra que cuando el bastón está en contacto con el suelo, se nota las variaciones de la fuerza aplicada dentro del mismo segundo, que estaría bien y, cuando el bastón está suspendido se tiene tres lecturas iguales a cero (Ejemplo: 11:54:38, 11:54:41, 11:54:44, 11:54:46). Estos valores, demuestran que el bastón está registrando los datos adecuadamente y en este modo no se va a descartar cuando el peso sea igual a cero.

Tabla 4-12. Valores capturados en tiempo real.

Fecha	Hora	Peso(k)	Fecha	Hora	Peso(k)
17/07/2024	11:54:34	0.12	17/07/2024	11:54:40	0
17/07/2024	11:54:34	2.67	17/07/2024	11:54:41	0
17/07/2024	11:54:35	4.74	17/07/2024	11:54:41	0
17/07/2024	11:54:35	4.38	17/07/2024	11:54:41	0
17/07/2024	11:54:35	4.31	17/07/2024	11:54:42	0
17/07/2024	11:54:36	1.99	17/07/2024	11:54:42	2.92
17/07/2024	11:54:36	4.98	17/07/2024	11:54:42	8.66
17/07/2024	11:54:36	8.16	17/07/2024	11:54:43	6.23
17/07/2024	11:54:37	8.19	17/07/2024	11:54:43	0
17/07/2024	11:54:37	5.64	17/07/2024	11:54:44	0
17/07/2024	11:54:38	0	17/07/2024	11:54:44	0
17/07/2024	11:54:38	0	17/07/2024	11:54:44	0
17/07/2024	11:54:38	0	17/07/2024	11:54:45	3.78
17/07/2024	11:54:39	0	17/07/2024	11:54:45	7.44
17/07/2024	11:54:39	3.02	17/07/2024	11:54:45	4.21
17/07/2024	11:54:39	8.97	17/07/2024	11:54:46	0
17/07/2024	11:54:40	6.62	17/07/2024	11:54:46	0
			17/07/2024	11:54:46	0

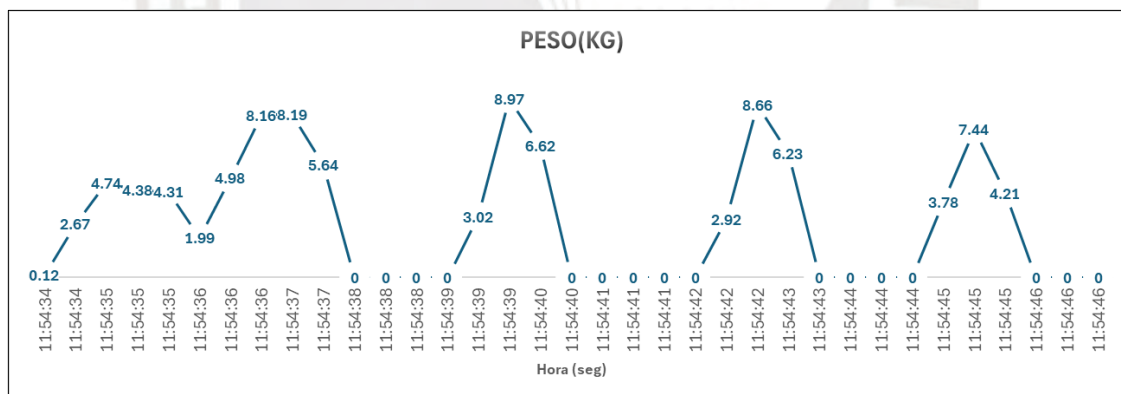


Figura 4-32. Gráfico de la tabla 4.11 (Fuente: elaboración propia)

Por ejemplo, con los tiempos dados en la Figura 4-31, se va a calcular la cantidad de bytes que ocuparán en el archivo durante un tiempo de una hora de entrenamiento, esto a manera de referencia.

$$\text{Número total de bytes} = (42B) * (3600 \times 1000ms / 400ms) = 378000B \approx 369KB$$

Esta cantidad de bytes es relativamente pequeña y no afectaría la capacidad de la memoria SD, inclusive si se elevara a 2 horas el entrenamiento.

#### 4.4.1. Persona usando el bastón electrónico en tiempo real

Para probar el funcionamiento del bastón, se realizaron pruebas caminando sobre una superficie plana en su propio hogar dentro del edificio donde radica. La persona es de sexo masculino con una edad de 63 años y peso de 70 kilos, goza de buen estado de salud físico y mental; puede caminar sin problema alguno, y utilizó el bastón sólo para mostrar su funcionamiento. Cabe precisar, que el objetivo de esta prueba es que se registre la fuerza aplicada sobre el bastón, así como los ángulos Pitch y Roll en cada paso dado y luego analizar las formas de onda generadas.

A continuación, se da algunas recomendaciones para utilizar adecuadamente el bastón<sup>24</sup>:

**Ajuste la altura adecuadamente:** Cuando esté de pie, el mango del bastón debe alcanzar la altura de su muñeca. El codo debe estar ligeramente flexionado cuando sostenga el bastón.

**Aprenda la técnica correcta:** Al caminar con un bastón, colóquelo en el lado opuesto de la pierna afectada o débil. Por ejemplo, si tiene una lesión en la pierna derecha, sostenga el bastón en la mano izquierda y viceversa. Mueva primero el bastón y luego la pierna afectada, luego la otra pierna.

**Use calzado adecuado:** Use zapatos cómodos y bien ajustados para evitar tropezones y caídas.

**Practique en un entorno seguro:** Antes de caminar en terrenos difíciles, practique con el bastón en un área segura y plana, como su casa o un pasillo.

**Mantenga una postura correcta:** Mantenga la espalda recta mientras camina y no se incline demasiado hacia adelante sobre el bastón. Distribuya su peso de manera uniforme entre el bastón y sus piernas.

**Use el bastón como apoyo:** No confíe completamente en el bastón. Úselo como un apoyo adicional para mejorar su estabilidad y equilibrio, pero trate de caminar de manera natural tanto como sea posible.

**Revise regularmente el bastón:** Asegúrese de que el bastón esté en buenas condiciones, sin grietas ni partes sueltas. Si es un bastón ajustable, verifique periódicamente que esté bien ajustado y que los mecanismos de bloqueo estén funcionando correctamente.

**Consulte a un profesional de la salud:** Siempre es recomendable consultar a un fisioterapeuta u otro profesional de la salud para obtener instrucciones personalizadas

---

<sup>24</sup> <https://www.fisioterapia-online.com/articulos/uso-del-baston-como-apoyo-para-la-marcha#:~:text=La%20postura%20al%20estar%20de,pie%20delantero%20deben%20estar%20iguales.>

sobre cómo usar un bastón correctamente, en especial si tiene una lesión específica o una condición médica.

Para la prueba funcional, se utilizó el monitoreo en tiempo real, asumiendo que la pierna izquierda está afectada, por lo que el bastón se coloca en la mano derecha. Esta prueba de marcha se realizó en una sala de 5 metros de longitud y se recorrieron 15 metros en un tiempo de 49 segundos. Se capturaron algunas fotos de la persona caminando, enfocando a su vez la pantalla del celular. Cabe precisar que, en esta prueba, no se han descartado los valores de peso igual a cero.

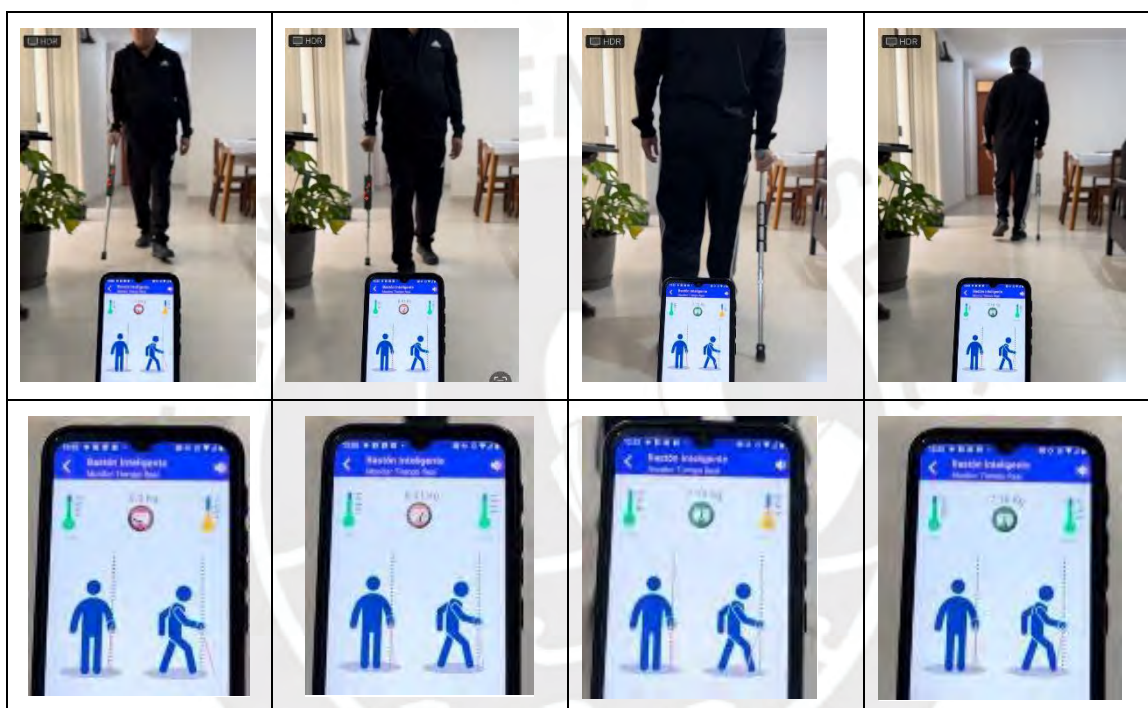
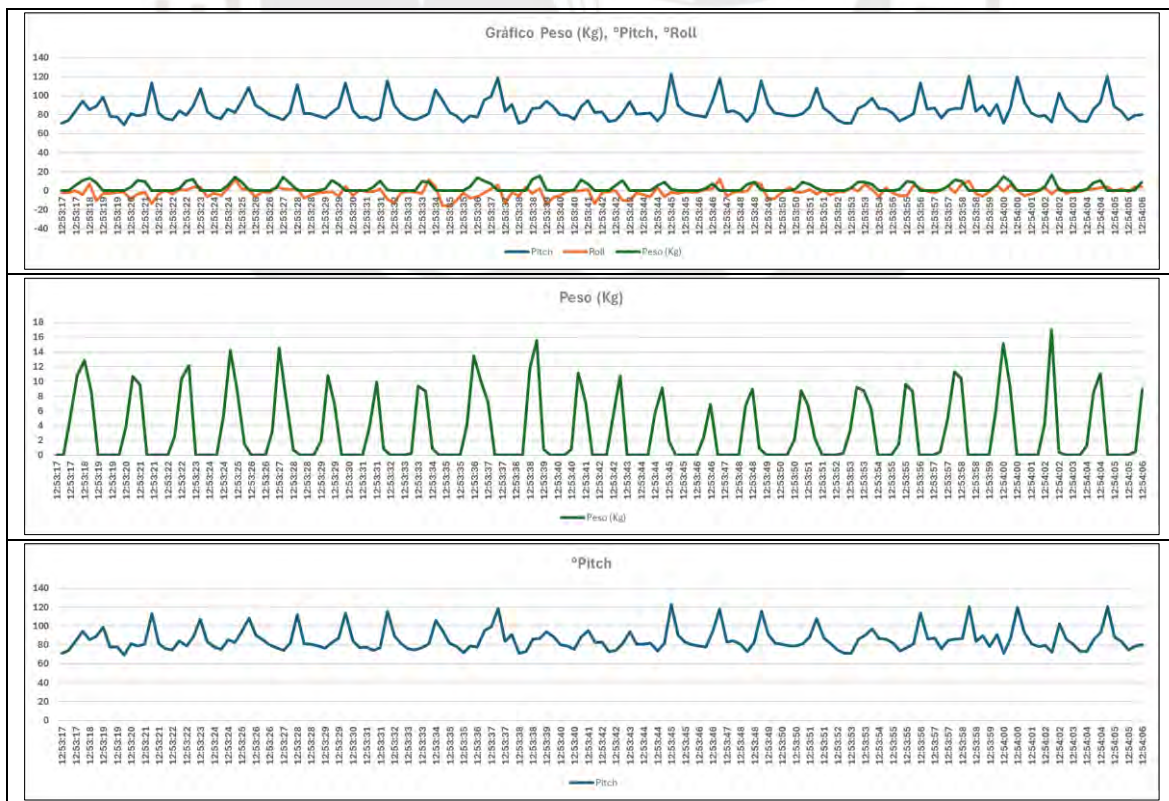


Figura 4-33. Prueba funcional bastón en tiempo real (Fuente: elaboración propia).

Luego de realizadas estas pruebas el día 22 de julio, los 158 datos almacenados en ese lapso de tiempo (49 seg.) fueron exportados a Excel tal como se muestra a continuación.

Tabla 4-13. Datos captados de la aplicación en tiempo real durante 49 seg (Fuente: elaboración propia)

1	Hora	Pitch	Roll	Peso	1	Hora	Pitch	Roll	Peso	1	Hora	Pitch	Roll	Peso	1	Hora	Pitch	Roll	Peso
2	12:53:17	70.87	-2.37	0	41	12:53:29	82.51	-1.03	10.78	80	12:53:42	82.78	-2.14	0	119	12:53:54	97.16	1.21	6.37
3	12:53:17	74.09	-2.37	0	42	12:53:29	87.36	-5.95	6.76	81	12:53:42	73.01	-1.25	0	120	12:53:54	86.7	-6.4	0
4	12:53:17	83.53	0.32	5.13	43	12:53:30	113.51	4.57	0	82	12:53:42	74.17	0.09	5.46	121	12:53:54	85.74	2.77	0
5	12:53:18	94.42	-4.16	10.72	44	12:53:30	84.35	-5.05	0	83	12:53:43	81.72	-10.24	10.72	122	12:53:55	81.55	-2.37	0
6	12:53:18	85.47	7.04	12.86	45	12:53:30	77.02	0.54	0	84	12:53:43	93.74	-10.47	0	123	12:53:55	73.41	-5.5	1.5
7	12:53:18	89.05	-10.7	8.62	46	12:53:31	77.47	-1.03	0	85	12:53:43	80.66	-2.37	0	124	12:53:55	76.75	-5.05	9.6
8	12:53:19	98.51	-2.81	0	47	12:53:31	73.92	-1.25	3.7	86	12:53:44	80.82	-4.16	0	125	12:53:56	81.09	5.24	8.68
9	12:53:19	77.83	-3.04	0	48	12:53:31	77.08	1.66	9.93	87	12:53:44	81.74	-6.63	0	126	12:53:56	113.52	2.55	0
10	12:53:19	77.68	-1.92	0	49	12:53:32	115.43	-8.66	0.86	88	12:53:44	73.44	3.45	5.6	127	12:53:56	85.89	-0.58	0
11	12:53:20	69.14	-1.7	0	50	12:53:32	89.86	-13.21	0	89	12:53:44	81.65	-6.63	9.15	128	12:53:57	87	-2.59	0
12	12:53:20	80.99	-8.88	3.77	51	12:53:32	80.96	-2.59	0	90	12:53:45	122.51	-1.47	1.83	129	12:53:57	76.01	0.76	0.46
13	12:53:20	78.85	-3.49	10.71	52	12:53:33	76.01	-0.36	0	91	12:53:45	90.1	-2.81	0	130	12:53:57	84.66	4.57	4.87
14	12:53:21	80.52	-1.47	9.51	53	12:53:33	74.41	-1.47	0.25	92	12:53:45	83.08	-1.25	0	131	12:53:58	86.24	-2.14	11.33
15	12:53:21	112.96	-13.44	0	54	12:53:33	77.23	-3.04	9.39	93	12:53:46	79.76	-1.7	0	132	12:53:58	86.4	6.59	10.39
16	12:53:21	81.43	-3.71	0	55	12:53:34	81.23	11.34	8.69	94	12:53:46	78.55	-1.03	0	133	12:53:58	120.13	10.2	0
17	12:53:22	75.88	0.09	0	56	12:53:34	105.89	4.34	0.89	95	12:53:46	77.69	1.21	2.38	134	12:53:58	83.45	-3.04	0
18	12:53:22	74.45	-3.71	0	57	12:53:34	95.11	-15.52	0	96	12:53:46	93.92	1.88	6.92	135	12:53:59	89.53	-5.95	0
19	12:53:22	84.03	0.99	2.58	58	12:53:35	82.07	-16.45	0	97	12:53:47	117.7	12.02	0	136	12:53:59	78.5	-1.25	0
20	12:53:22	79.1	0.76	10.36	59	12:53:35	78.45	-10.24	0	98	12:53:47	82.82	-5.73	0	137	12:53:59	90.64	6.14	6.09
21	12:53:23	88.56	3.45	12.16	60	12:53:35	71.84	-2.14	0	99	12:53:47	84.24	-1.7	0	138	12:54:00	71.2	-0.58	15.16
22	12:53:23	107.11	3.45	0	61	12:53:36	78.7	-7.53	4.14	100	12:53:48	80.64	-1.25	0	139	12:54:00	87.13	6.14	9.3
23	12:53:23	83.54	-6.4	0	62	12:53:36	77.53	-6.4	13.49	101	12:53:48	72.67	-0.58	6.74	140	12:54:00	119.99	0.32	0
24	12:53:24	77.32	-2.37	0	63	12:53:36	95.34	-2.59	9.89	102	12:53:48	82.73	8.84	9	141	12:54:01	93.25	-5.5	0
25	12:53:24	75.44	-5.05	0	64	12:53:37	99.17	1.88	7.18	103	12:53:49	115.62	6.81	0.93	142	12:54:01	81.49	-3.71	0
26	12:53:24	85.61	1.66	5.37	65	12:53:37	118.54	5.91	0	104	12:53:49	91	-8.88	0	143	12:54:01	78.18	-0.8	0
27	12:53:25	82.34	12.02	14.22	66	12:53:37	83.53	-13.21	0	105	12:53:49	81.76	-8.2	0	144	12:54:02	79.55	3.22	4.27
28	12:53:25	93.88	1.88	8.73	67	12:53:37	90.65	-1.7	0	106	12:53:50	80.48	-1.92	0	145	12:54:02	71.91	-3.49	17
29	12:53:25	108.27	0.99	1.46	68	12:53:38	71.01	-5.73	0	107	12:53:50	78.95	3.45	0	146	12:54:02	102.42	3.22	0.4
30	12:53:26	90.09	-6.4	0	69	12:53:38	73.09	3.67	0	108	12:53:50	78.7	-1.92	2.06	147	12:54:03	86.29	-3.04	0
31	12:53:26	85.26	-1.25	0	70	12:53:38	86.21	-3.04	11.85	109	12:53:50	81.11	-1.92	8.75	148	12:54:03	80.4	-1.03	0
32	12:53:26	80.08	-2.14	0	71	12:53:39	86.89	2.33	15.51	110	12:53:51	88.26	0.99	6.74	149	12:54:03	73.11	-1.03	0
33	12:53:27	77.3	4.12	3.24	72	12:53:39	93.94	-14.59	0.78	111	12:53:51	107.78	-3.71	2.28	150	12:54:04	72.9	0.99	1.21
34	12:53:27	74.3	1.66	14.57	73	12:53:39	88.46	-6.85	0	112	12:53:51	87.08	0.54	0	151	12:54:04	85.85	1.66	8.51
35	12:53:27	82.25	1.43	7.72	74	12:53:40	79.78	-5.05	0	113	12:53:52	81.5	-4.61	0	152	12:54:04	93.01	3	11.03
36	12:53:28	111.64	1.66	0.68	75	12:53:40	79	-0.8	0	114	12:53:52	74.51	-1.03	0	153	12:54:05	120.49	4.12	0
37	12:53:28	81.29	-7.98	0	76	12:53:40	75.15	-0.36	0.79	115	12:53:52	70.79	-1.03	0.22	154	12:54:05	88.72	-0.36	0
38	12:53:28	80.85	-4.83	0	77	12:53:41	88.25	0.09	11.15	116	12:53:53	70.71	3	3.29	155	12:54:05	83.53	1.88	0
39	12:53:29	78.7	-2.59	0	78	12:53:41	94.93	1.43	7.16	117	12:53:53	86.21	-0.58	9.23	156	12:54:05	74.37	-1.03	0
40	12:53:29	76.42	-1.7	1.96	79	12:53:41	82.33	-13.44	0	118	12:53:53	90.11	6.81	8.73	157	12:54:06	79.11	4.34	0.48
															158	12:54:06	79.74	3.89	8.96



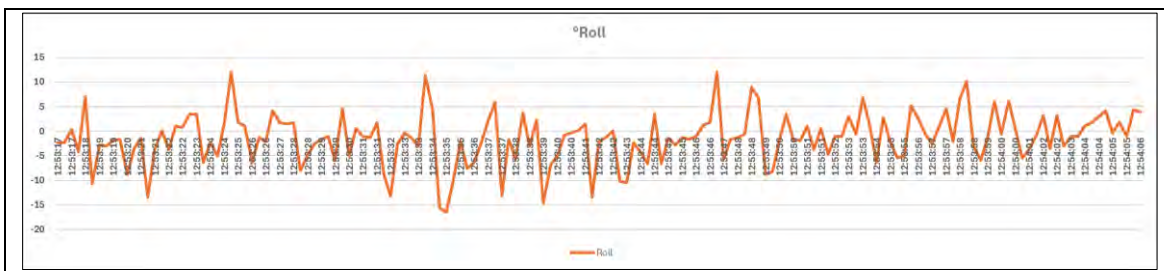


Figura 4-34. Gráfico de 156 datos del Peso, °Pitch, °Roll y Hora. (Fuente: elaboración propia)

Para un mejor análisis del resultado mostrado, se va a restringir a los primeros 54 datos de la Tabla 4-13 como se muestra a continuación.

Tabla 4-14. Datos captados de la aplicación en tiempo real durante 16 seg (Fuente: elaboración propia)

1	Fecha	Hora	Pitch	Roll	Peso (Kg)
2	22/07/2024	12:53:17	70,87	-2,37	0
3	22/07/2024	12:53:17	74,09	-2,37	0
4	22/07/2024	12:53:17	83,53	0,32	5,13
5	22/07/2024	12:53:18	94,42	-4,16	10,72
6	22/07/2024	12:53:18	85,47	7,04	12,86
7	22/07/2024	12:53:18	89,05	-10,7	8,62
8	22/07/2024	12:53:19	98,51	-2,81	0
9	22/07/2024	12:53:19	77,83	-3,04	0
10	22/07/2024	12:53:19	77,68	-1,92	0
11	22/07/2024	12:53:20	69,14	-1,7	0
12	22/07/2024	12:53:20	80,99	-8,88	3,77
13	22/07/2024	12:53:20	78,85	-3,49	10,71
14	22/07/2024	12:53:21	80,52	-1,47	9,51
15	22/07/2024	12:53:21	112,96	-13,44	0
16	22/07/2024	12:53:21	81,43	-3,71	0
17	22/07/2024	12:53:22	75,88	0,09	0
18	22/07/2024	12:53:22	74,45	-3,71	0
19	22/07/2024	12:53:22	84,03	0,99	2,58
20	22/07/2024	12:53:22	79,1	0,76	10,36
21	22/07/2024	12:53:23	88,56	3,45	12,16
22	22/07/2024	12:53:23	107,11	3,45	0
23	22/07/2024	12:53:23	83,54	-6,4	0
24	22/07/2024	12:53:24	77,32	-2,37	0
25	22/07/2024	12:53:24	75,44	-5,05	0
26	22/07/2024	12:53:24	85,61	1,66	5,37
27	22/07/2024	12:53:25	82,34	12,02	14,22
28	22/07/2024	12:53:25	93,88	1,88	8,73
29	22/07/2024	12:53:25	108,27	0,99	1,46
30	22/07/2024	12:53:26	90,09	-6,4	0
31	22/07/2024	12:53:26	85,26	-1,25	0
32	22/07/2024	12:53:26	80,08	-2,14	0
33	22/07/2024	12:53:27	77,3	4,12	3,24
34	22/07/2024	12:53:27	74,3	1,66	14,57
35	22/07/2024	12:53:27	82,25	1,43	7,72
36	22/07/2024	12:53:28	111,64	1,66	0,68
37	22/07/2024	12:53:28	81,29	-7,98	0
38	22/07/2024	12:53:28	80,85	-4,83	0
39	22/07/2024	12:53:29	78,7	-2,59	0
40	22/07/2024	12:53:29	76,42	-1,7	1,96
41	22/07/2024	12:53:29	82,51	-1,03	10,78
42	22/07/2024	12:53:29	87,36	-5,95	6,76
43	22/07/2024	12:53:30	113,51	4,57	0
44	22/07/2024	12:53:30	84,35	-5,05	0
45	22/07/2024	12:53:30	77,02	0,54	0
46	22/07/2024	12:53:31	77,47	-1,03	0
47	22/07/2024	12:53:31	73,92	-1,25	3,7
48	22/07/2024	12:53:31	77,08	1,66	9,93
49	22/07/2024	12:53:32	115,43	-8,66	0,86
50	22/07/2024	12:53:32	89,86	-13,21	0
51	22/07/2024	12:53:32	80,96	-2,59	0
52	22/07/2024	12:53:33	76,01	-0,36	0
53	22/07/2024	12:53:33	74,41	-1,47	0,25
54	22/07/2024	12:53:33	77,23	-3,04	9,39

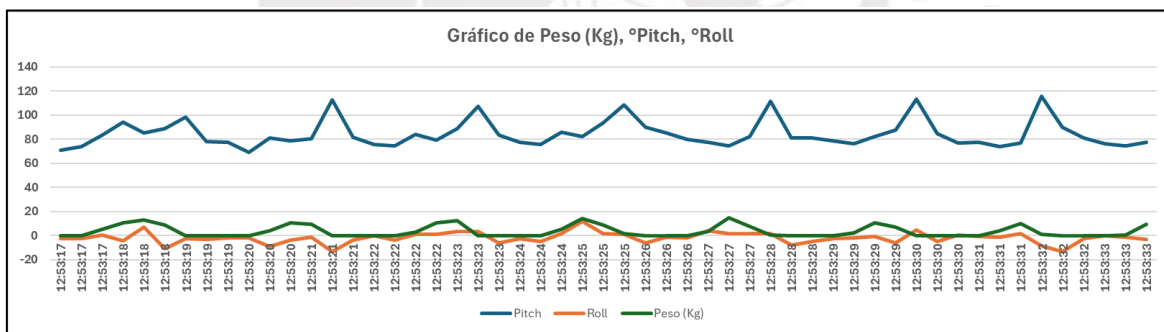


Figura 4-35. Gráfico del Peso, °Pitch y °Roll durante 16 seg de la prueba funcional en tiempo real. (Fuente: elaboración propia)



Figura 4-36. Gráfico del Peso vs. Hora durante la prueba funcional en tiempo real. (Fuente: elaboración propia)



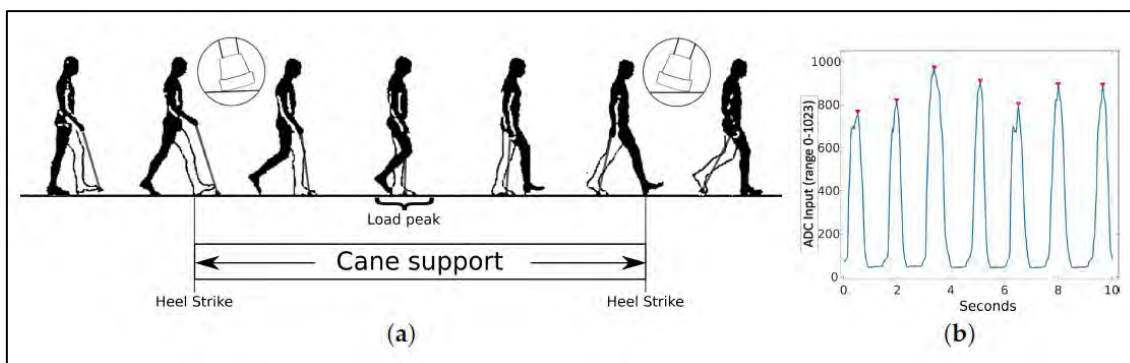


Figura 4-39. Contralateral user walking: (a) User walking in the support cane period; and (b) Input signal from force sensor during a 10 s interval in user 8 (range is 0–1023). Maximum peaks (corresponding to cane vertical position) are marked in.

(Fuente: <https://www.mdpi.com/1424-8220/19/3/509#B4-sensors-19-00509>)

Comparando los resultados del peso corporal aplicado sobre el bastón obtenidos en nuestra prueba (Figura 4-36), con la obtenida en la Figura 4-39 del bastón UMA, se puede deducir que son bastante parecidas, por lo tanto, el proyecto realizado cumplió su objetivo en cuanto al peso. Referente a los ángulos Pitch y Roll, no hay trabajos similares como para hacer la comparación.

Las figuras 4-35 hasta la 4-38, nos muestran cómo se utilizó el bastón durante la prueba en tiempo real. Estas pueden mejorarse con más entrenamiento por parte del médico tratante o terapeuta. Asimismo, se debe hacer hincapié en que el peso corporal aplicado sobre el bastón no debe sobrepasar lo recomendado para no perjudicar el estado clínico del paciente.

Cabe precisar que, cuando se hace monitoreo en tiempo real, no se filtran datos de peso ni ángulos, por ello se registraron pesos de 0 kg, excepto cuando se usa el bastón en modo normal o cotidiano. En este caso, se registran o almacenan en el archivo los pesos > 0 kg y los ángulos de pitch [60°, 110°] y roll [-30°, 30°].

Es preciso mencionar, que los resultados obtenidos en las diferentes pruebas realizadas, sólo se pudo comparar el peso corporal aplicado con otros resultados de investigaciones o tesis anteriores, más no, que monitoreen la forma como se mueve el bastón con algún dispositivo electrónico y que se visualice en un aplicativo del celular, que es fundamental en la parte de entrenamiento y en su posterior uso, por ello en las Tesis Doctorales de Fernández, María del Carmen Ridao [10] y Melis EH. et al. [11] en su estudio sobre el bastón de mano, utilizaron las “plataformas dinamométricas adicionando cámaras de video en el plano frontal y otra en el plano sagital para conseguir un registro de la marcha de los

pacientes, además de facilitar la enseñanza del análisis observacional a otros profesionales sanitarios”.

#### 4.5. Análisis de costos del diseño del bastón inteligente para rehabilitación

El costo del diseño del bastón en lo referente a componentes electrónicos, tarjetas PCB, soportes de sensor (regatón), tarjetas CPU-Alimentación y costo del diseñador por horas dedicadas se muestra en la siguiente tabla.

Tabla 4-15. Costo total del proyecto Bastón inteligente.

<b>Componentes - Tarjeta principal</b>	Valor (S/.)
Tarjeta Teensy 3.2 - Microcontrolador	100.00
Sensor de fuerza Flexiforce A201	90.00
Módulo ADXL345 - Acelerómetro	19.00
Módulo DS3231 - Reloj en tiempo real	16.00
Módulo HC-06 - Bluetooth	25.00
Memoria Micro SD	10.00
C.I. MCP6002 - OPAMP	5.00
C.I. AMS1117 - Regulador de voltaje a 1.2 voltios	3.00
Tarjeta PCB doble cara metalizada para CPU	40.00
<b>Componentes - Tarjeta de alimentación</b>	
Módulo TP4056 - Cargador de batería	5.00
Módulo TPS63802 - Regulador step-down - 3.3V	5.00
Batería de Litio recargable 1000 mAh	10.00
Tarjeta PCB una sola cara	5.00
<b>Componentes en impresión 3D</b>	
Regatón - Soporte sensor de fuerza	6.00
Soporte de Tarjetas CPU - Alimentación	6.00
Bastón metálico	70.00
<b>Costo total componentes</b>	415.00
Costo diseñador: 50 soles/hr x 400 hr	20,000.00
<b>Costo total</b>	<b>20,415.00</b>

El costo relativo a componentes del bastón está cercano a los \$110, que no es caro en comparación con otros bastones inteligentes para invidentes que tienen un precio de venta de \$350<sup>25</sup>. Cabe precisar, que este costo se puede disminuir si los componentes son adquiridos en Ali Express y el procesador se cambia por un ESP32.

<sup>25</sup> <https://www.brailleperu.pe/product-page/baston-inteligente>

## CONCLUSIONES

- a) Este proyecto de tesis se centra en mejorar la implementación de la marcha asistida con bastón, una técnica ampliamente utilizada en la reeducación de la marcha para pacientes con lesiones musculoesqueléticas. Esta técnica no solo alivia el dolor al reducir la carga sobre la extremidad afectada, sino que también mejora significativamente la confianza y seguridad del paciente al caminar, aumentando su independencia. Estas mejoras subrayan la importancia de desarrollar y perfeccionar dispositivos de asistencia como este bastón inteligente, que pueden ofrecer un soporte crucial en la recuperación de los pacientes y optimizar sus resultados de rehabilitación.
- b) El bastón inteligente diseñado proporciona una solución no invasiva, ligera y flexible que aborda necesidades clave en el tratamiento de rehabilitación, proporcionando una herramienta avanzada para el seguimiento integral del proceso del paciente, desde las etapas iniciales hasta las finales. Al almacenar información detallada sobre el uso del bastón en una memoria SD, se facilita un feedback preciso y un mejor control clínico. Esto no solo permite una adaptación personalizada de los tratamientos, sino que también abre la puerta a la creación de nuevos métodos y procedimientos clínicos más eficaces, optimizando el proceso de rehabilitación según las necesidades individuales de cada paciente.
- c) Aunque existen numerosas investigaciones sobre bastones inteligentes para personas invidentes, este proyecto destaca por su enfoque innovador en la rehabilitación de la marcha. El circuito electrónico desarrollado al permitir monitorear y registrar el peso corporal aplicado más los patrones de movimiento del bastón en diferentes etapas de la rehabilitación representa una contribución única y valiosa. Este avance no solo llena un vacío en la investigación existente, sino que también proporciona una herramienta integral para mejorar la calidad y el seguimiento del proceso de rehabilitación.
- d) El capítulo 3 detalla exhaustivamente el proceso de diseño de hardware y software para alcanzar los objetivos del proyecto. La obtención precisa de la curva de respuesta del sensor de fuerza de 25 libras (sección 3.2.1) y el desarrollo del sistema procesador junto con el software de control del CPU y del aplicativo móvil (sección 3.7) destacan como elementos fundamentales para asegurar el éxito y la funcionalidad del sistema completo del bastón para rehabilitación.
- e) El desarrollo del circuito electrónico para el bastón y su integración con una aplicación móvil para dispositivos Android representan un avance significativo en el monitoreo de la rehabilitación. La capacidad de monitorear en tiempo real el peso corporal aplicado

al bastón y los ángulos de pitch y roll proporciona una valiosa herramienta tanto para los pacientes como para los profesionales de la salud. Esto permite un seguimiento preciso y una corrección inmediata, lo cual es crucial para garantizar una rehabilitación segura y efectiva.

- f) La aplicación móvil desarrollada ofrece una interfaz intuitiva y eficaz para el monitoreo del peso corporal aplicado sobre el bastón durante la rehabilitación. La visualización del peso mediante un tacómetro de cinco zonas diferenciadas por colores, junto con las alertas sonoras para las zonas críticas, proporciona al paciente y al profesional de salud una herramienta invaluable para asegurar que el paciente aplique el peso corporal adecuado. Esto contribuye significativamente a la seguridad y eficacia del proceso de rehabilitación, garantizando un uso correcto del bastón (sección 3.7.2.1 ítem 3).
- g) La visualización en tiempo real de los movimientos del bastón a través de los ángulos pitch y roll en la pantalla del celular representa un avance significativo en el monitoreo de la rehabilitación del paciente. Esta funcionalidad permite al médico tratante o al familiar del paciente observar y corregir la postura y el uso del bastón de manera inmediata, lo que facilita un proceso de rehabilitación más seguro y efectivo. (sección 3.7.2.1 ítem 3).
- h) El desarrollo del aplicativo móvil es una herramienta crucial para la rehabilitación de los pacientes, permitiendo la continuidad del entrenamiento en casa bajo la supervisión remota del médico. La funcionalidad de seguimiento de la evolución del paciente, mediante la opción 'Mostrar datos y crear Excel' (sección 3.7.2.1 ítem 2), facilita un monitoreo detallado y efectivo, lo que contribuye a un proceso de rehabilitación más controlado y personalizado.
- i) Los resultados obtenidos, detallados en el capítulo 4, demuestran que el sistema desarrollado cumple satisfactoriamente con los objetivos planteados del proyecto. Las pruebas realizadas mostraron un error porcentual en la medición del peso corporal aplicado sobre el bastón de menos del 3%, lo que indica una alta precisión del dispositivo. Además, los ángulos pitch y roll registrados coincidieron con el movimiento del bastón visualizado en tiempo real en la pantalla del celular, confirmando la fiabilidad del sistema. Estos resultados validan la eficacia del bastón inteligente como una herramienta útil para la reeducación de la marcha, proporcionando a los pacientes y profesionales de la salud datos precisos y valiosos para el seguimiento y ajuste del proceso de rehabilitación.

- j) El sistema propuesto representa un avance significativo en el tratamiento de la marcha con bastón, ya que permite al fisioterapeuta proporcionar al paciente un feedback inmediato. Esto es posible gracias a la información simultánea que se obtiene mientras el paciente camina, lo que facilita la corrección de errores en tiempo real.
- k) Según los resultados obtenidos en el capítulo 4, sección 4.4.1 la ubicación del sensor en el regatón del bastón logra una óptima captación del peso corporal aplicado sobre el bastón. Esta conclusión se corrobora con los resultados de [3] en la Figura 4-39.

### 5.1. Limitaciones del Proyecto

A pesar de los esfuerzos realizados para garantizar la exhaustividad y la precisión de esta investigación, es importante reconocer ciertas limitaciones como las que se detallan a continuación:

1. El laboratorio de Electrónica solo disponía de tres pesas homologadas o certificadas de 5 kg cada una, lo que permitió obtener la curva de transferencia del sensor de 25 libras. Sin embargo, no fue posible obtener la curva del sensor de 100 libras, ya que no se contaba con pesas certificadas de hasta 50 kg. El fabricante del sensor recomienda trabajar con tres puntos principales: 0 kg (inicio), 20 kg (valor intermedio) y 50 kg (20% más de 100 lb).
2. La validación del peso soportado por el bastón se realizó con recursos tecnológicos limitados. Se utilizó una balanza digital colocada debajo del bastón y se añadieron pesas sobre la empuñadura, asegurando el bastón para evitar que se cayera (Figura 4-3 sección 4.1.1.). La falta de acceso a tecnologías avanzadas para realizar estas pruebas no solo prolongó el tiempo de desarrollo, sino que también pudo haber limitado la profundidad del análisis.
3. La prueba de funcionalidad del bastón se realizó con una persona sana y con su mano dominante (diestro). Hubiera sido preferible probarlo con pacientes varones y damas, probando ambas manos ya que esto podría mejorar la generalización de los resultados.

## RECOMENDACIONES

1. Se recomienda incorporar profesionales de ingeniería (mecánica, biomédica, electrónica) para mejorar la estructura física del regatón y la ubicación de los componentes del bastón inteligente diseñado en esta investigación. Dado que se trata de un prototipo innovador tanto en hardware como en software, estas mejoras permitirán obtener un dispositivo con un mejor acabado.
2. Se recomienda realizar pruebas de campo con pacientes en rehabilitación que tengan la posibilidad de recuperar la capacidad de caminar. Para ello, es importante contar con equipos multidisciplinarios que incluyan profesionales de ingeniería y salud, comprometidos en el área de rehabilitación de la marcha. Estos equipos deben establecer protocolos clínicos para el uso del bastón inteligente, con el fin de contribuir a la pronta recuperación de los pacientes.
3. Los sistemas tecnológicos implementados en este bastón inteligente, aunque ofrecen un alto grado de objetividad y proporcionan datos cuantitativos, requieren de una evaluación observacional simultánea en tiempo real entre el paciente y el aplicativo móvil del celular por parte de los médicos especialistas para ser efectivos.
4. Se sugiere que este bastón diseñado se integre con las herramientas de valoración observacional ya existentes para los estudios de rehabilitación de la marcha. De esta manera, se mejorará la calidad de las intervenciones clínicas, contribuyendo a la protocolización de tratamientos, optimizando los tiempos de recuperación y reduciendo el número de recaídas en pacientes que necesitan descargar un miembro inferior afectado debido a una lesión musculoesquelética.
5. Se recomienda realizar pruebas para determinar los porcentajes de peso corporal que los pacientes pueden aplicar sobre este bastón diseñado. Esto permitirá verificar si lo planteado en la tesis doctoral de María del Carmen Rídao [5], que sugiere no sobrepasar el 10% del peso corporal en un bastón de antebrazo, se puede aplicar a nuestro bastón. Así mismo, se sugiere probar con un 20%, 30% hasta 50% del peso corporal con supervisión del fisioterapeuta.
6. Se recomienda que el médico tratante debería prestar especial atención al proceso de enseñanza-aprendizaje de esta tarea cuando el bastón sea portado por la mano no dominante, dada su mayor exigencia coordinativa [10].
7. Este proyecto abre una línea de investigación y desarrollo a futuro, con el objetivo de estudiar la posibilidad de integrar esta tecnología en la red de los hospitales. De este

modo, mediante WiFi, la aplicación móvil se conectaría a una plataforma en la nube, permitiendo que los datos de cada paciente se envíen directamente a su historia clínica de manera remota. Esto facilitaría el acceso del médico tratante a la información sin necesidad de descargarla mediante Bluetooth del bastón de manera presencial. Además, el médico podría proporcionar feedback al paciente sobre el progreso de su tratamiento o, en su defecto, realizar las correcciones pertinentes.



## Bibliografía

- [1] INEI, «Primera Encuesta Nacional Especializada sobre discapacidad,» *Primera Encuesta Nacional Especializada sobre discapacidad - Resumen*, pp. 3-31, 2012.
- [2] J. Ballesteros, C. Urdiales y A. y. T. M. Martinez, «Gait analysis for challenged users based on a rollator equipped with force sensors,» de *In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 28 September - 2 October 2015.
- [3] Joaquin Ballesteros; Alberto Tudela; Juan Rafael Caro-Romero; Cristina Urdiales, «Weight-Bearing Estimation for Cane Users by Using Onboard Sensors,» Malaga, Spain, 2019.
- [4] B. Joaquín, «<https://github.com/joaquinballesteros/Smart-Cane>,» Department of Electronic Technology (Málaga University), 2018. [En línea]. [Último acceso: 14 01 2023].
- [5] I. Ayala, J. Ballesteros, J. R. Caro-Romero, M. Amor y L. Fuentes, «Self-Adaptation of mHealth Devices: The Case of the Smart Cane Platform,» Departamento de Tecnología Electrónica Universidad de Málaga, 29071 Málaga, Spain, 20 11 2019. [En línea]. Available: <https://doi.org/10.3390/proceedings2019031023>. [Último acceso: 20 01 2023].
- [6] A. Benayad, N. Hachenmi, C. Hean y E. y. R. L. P. Yvroud, «Analysis of assisted walking with canes using a telemetric system,» *Motricidad. European Journal of Human Movement*, vol. 5, pp. 133-144, 1999.
- [7] G. Harris, D. Lieutter, B. Bergner, D. Matesi y N. y. R. J. Pelec, «A multichannel PMW telemetry system for kinematic gait analysis.,» *Medical Instrumentation*, nº 21, pp. 304-313, 1987.
- [8] G. Chamorro-Moriana<sup>1</sup>, C. Ridao-Fernández y J. Ojeda, «Reliability and Validity Study of the Chamorro Assisted Gait Scale for People with Sprained Ankles, Walking with Forearm Crutches,» Sevilla - España, 2016.
- [9] S. J. R.-F. C. Chamorro-Moriana G, «A Compact Forearm Crutch Based on Force Sensors for Aided Gait: Reliability and Validity,» *Sensors (Basel)*, vol. 16, nº 6, p. 925, 21 06 2016.
- [10] M. d. C. R. Fernández, *Tesis Doctoral: Valoración de la marcha asistida con bastones de antebrazo*, Sevilla, España: Universidad de Sevilla, 2019.
- [11] Melis EH, Torres-Moreno R, Barbeau H, Lemaire ED, «Analysis of assisted-gait characteristics in persons with incomplete spinal cord injury.,» *International Medical Society of Paraplegia*, vol. 37, nº 6, pp. 430-439, 1999.
- [12] G. y. Hall, *Tratado de Fisiología médica*, 1600 John F. Kennedy Blvd.: Saunders, 2011.  
]
- [13] V. V. A, *Lecciones básicas de biomecánica del aparato locomotor*, Barcelona: Primera ed.  
] Barcelona: Springer, 2000.

- [14 Z. P. Dedieu P, «Effects of gait pattern and arm swing on intergirdle coordination,» *Hum Mov Sci*, 2012.
- [15 Cifuentes C., Martínez F., Romero E., «ANÁLISIS TEÓRICO Y COMPUTACIONAL DE LA MARCHA NORMAL,» vol. 18, nº 2, pp. 182-196, 2010.
- [16 N. R. K. S. Zajac F, «Biomechanics and muscle coordination of human walking: part II: lessons from dynamical simulations and clinical implications,» *Gait Posture*, vol. 17, nº 1, 2003.
- [17 P.-P. J. Sánchez-Lacuesta JJ, «Biomecánica de la marcha humana normal y patológica,» Instituto de Biomecánica de Valencia, Valencia, 1999.
- [18 A. A. Mendoza, T. B. Santamaria, V. G. Urrego, J. P. Ruiz y M. Z. García, «Gait: description, methods, assessment tools and normality parameters reported in the literature,» Colombia, 2013.
- [19 M. G. Jiménez, "Análisis de la marcha: Evaluación de un Exoesqueleto aplicado a la marcha asistida", tesis doctoral, Univ. Politécnica de Madrid, Madrid, 2016.
- [20 S. Gutierr, «Uso del bastón como apoyo para la marcha,» 17 09 2015. [En línea]. Available: <https://www.fisioterapia-online.com/articulos/uso-del-baston-como-apoyo-para-la-marcha>. [Último acceso: 16 08 2023].
- [21 OPS, «Rehabilitación del Adulto Mayor,» *Módulos de Valoración Clínica*, vol. 8, pp. 100-101.
- [22 H. y. S. D. Chambers, «A practical guide to gait analysis. Journal of the American Academy of Orthopaedic Surgeons,» vol. 10, pp. 222-231, 2002.
- [23 J. y. B. J. Perry, *Gait Analysis Normal and Pathological Function* (2ª edición), Thorofare, NJ: SLACK Inc., 2010.
- [24 R. BAKER, A. ESQUENAZI, M. G. BENEDETTI y K. DESLOOVERE, «Gait analysis: clinical facts,» *MEDITERRANEAN JOURNAL OF PHYSICAL AND REHABILITATION MEDICINE*, vol. 52, nº 4, p. 560, 08 2016.
- [25 M. Izquierdo, A. Martínez-Ramírez, J. Larrión, M. Irujo-Espinosa y M. Gómez, «Valoración de la capacidad funcional en el ámbito domiciliario y en la clínica. Nuevas posibilidades de aplicación de la acelerometría para la valoración de la marcha, equilibrio y potencia muscular en personas mayores.,» Agosto 2008. [En línea]. Available: [http://scielo.isciii.es/scielo.php?pid=S1137-66272008000300006&script=sci\\_arttext](http://scielo.isciii.es/scielo.php?pid=S1137-66272008000300006&script=sci_arttext). [Último acceso: 08 06 2023].
- [26 Analog Device, «ADXL345 Digital Accelerometer,» 2017.
- [27 W. X. Y. W. J. e. a. Chen, «Kinematic Analysis of Human Gait Based on Wearable Sensor System for Gait Rehabilitation,» *Journal of Medical and Biological Engineering*, vol. 36, pp. 843-856, 11 11 2016.

- [28 Mapro Ingeniería S.A., «Mapro sensor,» [En línea]. Available:  
] <https://maprosensor.com/dinamometros-sensores-carga>. [Último acceso: 26 09 2023].
- [29 TEKSCAN, «FlexiForce® Sensors - User Manual,» South Boston.  
]
- [30 S. W. a. E. B. Sarah Kate Wilson, *Transmission Techniques for Digital Communications*, Santa Clara, CA, USA: Academic Press, 2016.
- [31 A. A. A. Solyman y K. Yahya, «Evolution of wireless communication networks: from 1G to 6G,» *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, nº 4, pp. 3943-3950, 2022.
- [32 Massachusetts Institute of Technology, «Modulation and Demodulation,» 11 April 2012. [En línea]. Available: <https://web.mit.edu/6.02/www/s2012/handouts/14.pdf>. [Último acceso: 15 10 2023].
- [33 Shahzeb Hussain, Namrata Bhadri, Md. Shaayan Hussain, «Advancements in Wireless Communication,» *SSRG International Journal of Electronics and Communication Engineering*, vol. 7, pp. 1-4, 2020.
- [34 J. Salazar, «Redes Inalámbricas,» Universidad Técnica Checa de Praga Facultad de Ingeniería Eléctrica, República Checa.
- [35 W. Stallings, *Wireless Communications and Networks*, NJ: Pearson Prentice Hall,, 2005.  
]
- [36 B. Ciubotaru, G.M. Muntean, *Advanced Network Programming. Principles and Techniques*, Springer: Verlag London, 2013.
- [37 N. D. K. Sharma, «A study of wireless networks: WLANs, WPANs, WMANs, and WWANs with comparison,» *International Journal of Computer Science and Information Technologies*, vol. 5, nº 6, pp. 7810-7813, 2014.
- [38 Ó. Corredor, L. Pedraza y C. Hernández, «Tecnología Bluetooth: Alternativa para redes celulares de voz y datos,» *Revista Visión Electrónica*, vol. 3, nº 1, pp. 73-84, 11 2009.
- [39 M. Verma, S. Singh y B. Kaur, «An Overview of Bluetooth Technology and its Communication,» *International Journal of Current Engineering and Technology*, vol. 5, nº 3, 10 05 2015.
- [40 Universidad Nacional Autónoma de México, «ESTÁNDAR IEEE 802.15.1 “BLUETOOTH”,» 2015. [En línea]. Available: [www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/229/A5.pdf?sequence=5](http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/229/A5.pdf?sequence=5). [Último acceso: 12 10 2023].
- [41 Bluetooth SIG Proprietary, «BLUETOOTH SPECIFICATION Version 5.0».  
]

- [42 M. Fisher, ARM Cortex M4 Cookbook, Birmingham B3 2PB, UK.: Packt Publishing Ltd., 2016,  
] p. 22.
- [43 W. Hohl y C. Hinds, ARM ASSEMBLY LANGUAGE: Fundamentals and Techniques, Boca Raton,  
] FL 33487-2742: Taylor & Francis Group, 2015.
- [44 A. Elahi y T. Arjeski, ARM Assembly Language with Hardware: Experiments, Connecticut USA:  
] Springer Cham Heidelberg New York Dordrecht London, 2015.
- [45 F. Semiconductor, «Manual de referencia de K20 Sub-Family - MK20DX256VLH7,» 2012.  
]
- [46 J. B. Jonathan Valdez, «Understanding the I2C Bus,» Texas Instrument, 2015.  
]
- [47 J. Martinez y M. Barrón, Prácticas con Microcontroladores de 8 bits, España: McGraw-Hill,  
] 1993, pp. 159-177.
- [48 Sparkfun start something, «Serial Peripheral Interface (SPI),» [En línea]. Available:  
] <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>. [Último acceso: 20 11 2023].
- [49 TEM Electronic Components, «¿Cómo elegir el microcontrolador adecuado para su  
] proyecto?,» 29 03 2024. [En línea]. Available: <https://www.tme.eu/es/news/library-articles/page/58382/como-elegir-el-microcontrolador-adecuado-para-su-proyecto/>. [Último acceso: 05 04 2024].
- [50 Departamento de Electricidad y Electrónica - Universidad de Valladolid, «Sensores,»  
] Valladolid.
- [51 P. G. Miguez, «Análisis Biomecánico de la marcha mediante un modelo de de dinámica  
] inversa en pacientes con alteración del equilibrio sagital,» Universidad Autónoma de Barcelona, Barcelona, 2017.
- [52 M. Integrated, «DS1307 Datasheet,» 2003.  
]
- [53 M. Integrated, «DS3231 Datasheet,» 2010.  
]
- [54 NXP, «PCF8563 datasheet».  
]
- [55 NanJing Top Power ASIC Corp., «TP4056 Standalone Linear Li-Ion Battery Charger with  
] Thermal,» China.
- [56 R. Whitaker, Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android,  
] Derby, UK: Apress Media LLC, 2020.
- [57 S. G. Oliver, Programación Android con Android Studio, Safe Creative, 2016.  
]

- [58 FAQsAndroid, CURSO DE PROGRAMACIÓN EN ANDROID PARA PRINCIPIANTES, Bemoob.  
]
- [59 R. S. Sánchez, *Tesis Doctoral: Influencia de la carga aplicada sobre batones de antebrazo en parámetros cinemáticos durante la marcha asistida en sujetos sanos*, Sevilla España: Universidad de Sevilla, 2013-2017.
- [60 «THE EVOLUTION OF WIRELESS COMMUNICATION,» [En línea]. Available:  
] <https://worldwidesupply.net/blog/blogevolution-wireless-network-technology/>. [Último acceso: 04 10 2023].
- [61 F. Coca, «Mis-notas-sobre-programacion-easy-plug,» [En línea]. Available:  
] <https://fgcoca.github.io/Mis-notas-sobre-programacion-easy-plug/acel/>. [Último acceso: 25 09 2023].
- [62 E. M. Suárez, «ANÁLISIS CINEMÁTICO Y CINÉTICO DE LA MARCHA HUMANA,» Universidad  
] Complutense de Madrid., Madrid, 2003.
- [63 S. G. Oliver, *Manual Programación Android*, 2013.  
]
- [64 J. Ballesteros, A. Tudela, J. Caro-Romero y U. Cristina, «Weight-Bearing Estimation for Cane  
] Users by Using Onboard Sensors,» *Sensors 2019*, vol. 19, nº 3, p. 509, 2019.

## 8 ANEXOS

### 8.1. Programa en el microcontrolador ARM con Lenguaje C

```
#include <Wire.h> // Librería I2C
#include <SPI.h> // Librería SPI
#include <SD.h> // Librería MicroSD
#include <ADC.h> // Librería ADC
#include <RTCLib.h> // Librería RTCLib
RTC_DS3231 RTC; // Crea el objeto RTC
long start = millis();
const int LedPin = 13; // Led del modulo Teensy
// Declaracion de variables para grabar en microSD
String fichero_ssid = "SSID.txt";
String Nombre_temporal = "temporal.txt";
char Nombre_Fichero[15];
char Nombre_leido[15];
//Declaracion de variables del ADC
const int ReadPinADC9 = A9;
volatile double voltaje, LecturaADC, peso;
volatile double kilos, libras, pesoLib, pesoKg;
ADC *adc = new ADC();
#define resolucion (3.3/((pow(2,12))-1));
double lectura_ADC(void);
//Declaracion de REG y VAR del ADXL345
#define ADXLAddress 0x53
#define BW_RATE 0x2C
#define POWER_CTL 0x2D

#define DATA_FORMAT 0x31
#define X_Axis_Register_DATA_X0 0x32
#define X_Axis_Register_DATA_X1 0x33
#define Y_Axis_Register_DATA_Y0 0x34
#define Y_Axis_Register_DATA_Y1 0x35
#define Z_Axis_Register_DATA_Z0 0x36
#define Z_Axis_Register_DATA_Z1 0x37
#define SAMPLES_BIAS 1000
#define ACCEL_X_SCALE 0.0039
#define ACCEL_Y_SCALE 0.0039
#define ACCEL_Z_SCALE 0.0039
byte _buff[6];
uint8_t BytesToRead = 6;
int pitch, roll;
signed short accel_X, accel_Y, accel_Z;
float X,Y,Z;
double biass_X, biass_Y, biass_Z;
double pitchAccel, rollAccel;
//Variables para la escritura de archivo
//desde el bluetooth y memoria SD
String nombrearchivo, auxiliar;
boolean salir, crear;
File myFile;
char ValorRead = 0;
```

```

Sd2Card card;
SdVolume Volume;
SdFile root;
const int chipSelectSD = 10;
int DeletedCount = 0;
int FolderDeleteCount = 0;
int FailCount = 0;
String rootpath = "/";
//Declaro prototipos de las funciones a utilizar
//por el reloj y acelerómetro
void dateTime(uint16_t* date, uint16_t* time);
void ReadFrom(byte address, int num, byte _buff[]);
void WriteTo(byte address, byte val);
void ADXL345_GetBiass(void);
void lectura_adxl435(void);
//Variables del BUFFER en memoria RAM
//fila = 41 bytes, 39 filas en un minuto aprox.
const int bufferSize = 2100;
char dataBuffer[bufferSize];
char bufferLine[42];
int bufferIndex = 0;
unsigned long lastWriteTime = 0;
const unsigned long writeInterval = 60000;
int lenBufferLine;

void setup() {
  pinMode(LedPin, OUTPUT);
  // Inicio la comunicacion SPI, I2C, UART, RTC
  // establezco la fecha y hora para crear el archivo
  SPI.setMOSI(11); // MOSI - SPI
  SPI.setSCK(13); // SCK - SPI
  SPI.setMISO(12); // MISO - SPI
  Wire.begin(); // Inicializa I2C
  Serial3.begin(115200); // Bluetooth a 115200 Baudios
  Serial.begin(9600); // Serial PC a 9600 Baudios
  RTC.begin(); // Inicia RTC
  delay(100);
  //Captura la fecha y hora del sistema
  RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
  // Fija la fecha y hora para crear el archivo
  SdFile::dateTimeCallback(dateTime);

  // Configuro ADC del microcontrolador
  pinMode(ReadPinADC9, INPUT);
  adc->adc0->setReference(ADC_REFERENCE::REF_3V3);
  adc->adc0->setAveraging(12);
  adc->adc0->setResolution(12);
  adc->adc0-
>setConversionSpeed(ADC_CONVERSION_SPEED::LOW_S
PEED);
  adc->adc0-
>setSamplingSpeed(ADC_SAMPLING_SPEED::MED_SPEED
);

  // Configuro acelerometro ADXL345
  WriteTo(POWER_CTL, 0x00);
  WriteTo(DATA_FORMAT, 0x0B);
  WriteTo(BW_RATE, 0x09);
  WriteTo(POWER_CTL, 0x08);
  ADXL345_GetBiass();
  DateTime now = RTC.now();

  // Inicio comunicación con la memoria SD para
  // crear el archivo que guardara la información
  Serial.print("INICIANDO TARJETA SD ...");
  if (!SD.begin(chipSelectSD))
  {
    Serial.println("INICIACION FALLIDA!");
    return;
  }
  Serial.println("INICIADO CORRECTAMENTE.");
  Serial.println(" "); // salto de línea

  // Recupera núm de historia clínica de la memoria SD
  if (!SD.exists(fichero_ssld))
  {
    myFile = SD.open(fichero_ssld, FILE_WRITE);
    delay(50);
    myFile.close();
    //crea "temporal.txt", hasta crear nuevo paciente
    myFile = SD.open(Nombre_temporal, FILE_WRITE);
    delay(50);
    myFile.close();
  }
  else
  {
    myFile = SD.open(fichero_ssld);
    if (myFile)
    {
      myFile.read(Nombre_Fichero,13);
      delay(50);
      Nombre_Fichero[13]=0;
      Serial.println(Nombre_Fichero);
      myFile.close();
      Serial.println("\nLectura del archivo completada");
    }
    else
    {
      Serial.println("Error al abrir el archivo");
    }
  }
  dataBuffer[bufferIndex]=0; // Inicio buffer en cero
  delay(500);
} //fin de setup

void loop()
{
  // Código de comunicación con el Bluetooth
  // recibe la orden de BT para crear archivo "v"
  int contadorSilencio;
  int pruebaBT=Serial3.available();
  if (pruebaBT>0) // BT está disponible
  {
    ValorRead = Serial3.read(); // lee dato via BT
    if (ValorRead !=0)
    {
      switch (ValorRead)
      {
        // Crea archivo en la SD con número HC o DNI
        // cuando en APPMóvil elije "Nuevo paciente
        case('1'):
          salir= true;
          crear= true;
          while (salir)
          {
            if (Serial3.available(> 0) // Recibió dato?

```

```

{
  nombreadarchivo = Serial3.readString();
  nombreadarchivo.trim();
  Serial3.flush();
  auxiliar = nombreadarchivo + ".txt";
  auxiliar.toCharArray(Nombre_Fichero,auxiliar.len
gth()+1);
  salir= false;          // salimos del bucle
  delay(50);
}
}
// Verifica si file existe en memSD antes de crear
if (SD.exists(Nombre_Fichero))
{
  Serial3.print("e"); // envío "e" por BT al celular
  crear=false;        // indica que el archivo existe
}
if(crear==true)      // caso contrario se crea
{
  SD.remove(fichero_ssid); // borra archivo old
  delay(5);
  // borra archivo temporal.txt
  SD.remove(Nombre_temporal);
  delay(5);
  // crea archivo SSID nuevo
  myFile = SD.open(fichero_ssid, FILE_WRITE);
  delay(5);
  // guardamos el número de historia
  myFile.write(Nombre_Fichero,13);
  delay(5);
  myFile.close();      // Cierra el archivo SSID
  delay(5);
  // crea nuevo archivo de paciente en memSD
  myFile = SD.open(Nombre_Fichero, FILE_WRITE);
  delay(5);
  if (myFile)
  {
    myFile.println("Fecha,Hora,Pitch(^),Roll(^),Peso(Kg
)");
    delay(50);
    myFile.close();    // cierra el archivo
    delay(100);
  }
  // Verifica si el archivo existe:
  if (SD.exists(Nombre_Fichero))
  {
    Serial3.print("c"); // Avisa por BT que archivo se
    delay(50);        // creo correctamente
  }
  else
  {
    Serial3.print("n"); // el archivo no se creó.
  }
} // fin crear archivo
break; // fin de case 1

/* Lee contenido de la memoria SD */
/* (no implementado en la App) */
case('2'):
  Serial3.print("h"); // envía conformidad de recibido
  delay(10);
  // usaremos el código de inicialización para
  // probar si la tarjeta SD funciona

if (!card.init(SPI_HALF_SPEED, chipSelectSD))
{
  Serial3.println("inicialización fallada. revisa:");
  Serial3.println("si tarjeta SD está bien insertada?");
  Serial3.println("si la conexión es correcta?");
  return;
}
// Ahora intentaremos abrir el 'volumen'/partición
// debería ser FAT16 o FAT32
if (!Volume.init(card))
{
  Serial3.println("No existe la partición
FAT16/FAT32");
  Serial3.println("Asegúrese de haber formateado la
tarjeta");
  return;
}
Serial3.println("Archivos encontrados en la SD: ");
delay(50);
Serial3.println("nombre, fecha, hora, #Bytes: ");
Serial3.print("=====");
delay(50);
root.openRoot(Volume);
// enumera los archivos en la SD con fecha y tamaño
root.ls(LS_R | LS_DATE | LS_SIZE);
// fin de lectura de memoria microSD
break; // fin de case 2

/* Borra archivos contenidos en la memoria SD */
/* (no implementado en la App) */
case('3'):
  Serial3.print("b"); // envía conformidad de recibido
  delay(5);
  Serial3.println("Borrando archivo de SD ...");
  delay(5);
  if (SD.exists(Nombre_Fichero))
  {
    SD.remove(Nombre_Fichero);
    delay(150);
    Serial3.println("Archivo Eliminado");
  }
  Serial3.println("No existe archivo");
  break; // fin de case 3

/* Envía data almacenada cuando en la APPMóvil */
/* se selecciona "Mostrar datos y Crear Excel */
case('4'):
  Serial3.print("k");
  delay(10);
  if (SD.exists(Nombre_Fichero))
  {
    File dataFile = SD.open(Nombre_Fichero);
    delay(50);
    if (dataFile)
    {
      while (dataFile.available())
      {
        // envía todo el contenido del archivo
        Serial3.write(dataFile.read());
      }
      dataFile.close();
      Serial3.write("#");
    }
  }
}

```

```

}
else // Si el archivo no se abre, hubo error
{
  Serial3.println("error abriendo archivo");
}
break; // fin de case 4

/* Uso del bastón en tiempo real */
case('5'):
  Serial3.print("o"); // envía conformidad de recibido
  contadorSilencio = 0;
  lastWriteTime = millis();
  bufferIndex = 0;
  dataBuffer[0] = 0;
  while(1)
  {
    // Obtiene la fecha y hora del RTC
    DateTime now = RTC.now();
    // Lee datos del acelerómetro y ADC
    // se envía al celular por BT
    lectura_adxl435();
    pesoLib = lectura_ADC();
    if(pesoLib<0) pesoLib=0;
    pesoKg = pesoLib * 0.454;
    Serial3.print("d");
    Serial3.print(pitchAccel);
    Serial3.print("#");
    Serial3.print(rollAccel);
    Serial3.print("#");
    Serial3.print(pesoKg);
    Serial3.print("#");
    //almacena los datos de una línea en la RAM
    bufferLine[0] = 0;
    sprintf(bufferLine, "%d/%d/%d,%d:%d:%d,%d.%2f,%d.%2f
,%d.%2f\n",
    now.day(), now.month(), now.year(),
    now.hour(), now.minute(), now.second(),
    (float)pitchAccel, (float)rollAccel, (float)pesoKg);
    lenBufferLine = strlen(bufferLine);
    // valida si la cantidad de la línea actual agregada
    // al buffer supera el tamaño del buffer y,
    // valida si superó el tiempo máximo permitido
    // para guardar si aún quedan datos en el buffer
    if ( ( bufferIndex + lenBufferLine >= bufferSize ) ||
        ( ( bufferIndex > 0 ) && ( ( millis() - lastWriteTime )
        > writeInterval ) ) )
    {
      myFile = SD.open(Nombre_Fichero, FILE_WRITE);
      if(myFile)
      {
        myFile.print(dataBuffer);
        myFile.close();
      }
      bufferIndex = 0;
      dataBuffer[0] = 0;
      lastWriteTime = millis();
    }
    // Agregar datos al buffer
    for(int i=0;i < lenBufferLine; i++)
    {dataBuffer[bufferIndex++] = bufferLine[i];}
    // pasado un tiempo de inactividad el celular
    // se desconecta del BT
    contadorSilencio++;

    if ( contadorSilencio > 100 ) break;
    while( Serial3.available() == 0 )
    {
      contadorSilencio++;
      delay(10);
      if ( contadorSilencio > 100 ) break;
    }
    if(Serial3.available()>0)
    {
      ValorRead = Serial3.read();
      if( ValorRead == 'f' ) break;
      else
      if (ValorRead == 'r' ) contadorSilencio = 0;
    }
    else
    {
      Serial.println("celular desconectado ");
      break;
    }
    delay(10);
  } //fin de while tiempo real

  //si algo quedo pendiente de datos, se graba
  if (bufferIndex > 0 )
  {
    Serial.println(" grabando lo último en SD ");
    myFile = SD.open(Nombre_Fichero, FILE_WRITE);
    if(myFile)
    {
      myFile.print(dataBuffer);
      myFile.close();
    }
  }
  break; // fin de case 5
} // fin de switch
} // fin de ValorRead
ValorRead = 0;
} // fin de comunicación con BT

// Comunicación con acelerómetro y ADC sin conexión BT
/*****
pesoLib = lectura_ADC(); // Lee ADC y devuelve en libras
if(pesoLib<0) pesoLib=0;
if(pesoLib >= 0)
{
  pesoKg = pesoLib * 0.454; // se convierte a Kg
  // Lee 6 bytes del acelerómetro y devuelve pitch y roll
  lectura_adxl435();
  if(pitchAccel>=60.0 && pitchAccel<=110.0){pitch=1;}
  else{pitch=0;}
  if(rollAccel>=-30.0 && rollAccel<=30.0){roll=1;}
  else{roll=0;}
  if(pitch==1 && roll==1) // Si es true, continuo y
  {
    // capturo la fuerza aplicada al bastón
    if (SD.exists(Nombre_Fichero)) // Si existe, se abre
    {
      myFile= SD.open(Nombre_Fichero, FILE_WRITE);
      delay(10);
    }
  }
  else
  {
    myFile = SD.open("temporal.txt", FILE_WRITE);
    delay(10);
  }
}
}

```

```

}
if (myFile)
{
// Obtiene fecha y hora del RTC y lo registra
DateTime now = RTC.now();
myFile.print(now.day(), DEC);
myFile.print('/');
myFile.print(now.month(), DEC);
myFile.print('/');
myFile.print(now.year(), DEC);
myFile.print(",");
myFile.print(now.hour(), DEC);
myFile.print(':');
myFile.print(now.minute(), DEC);
myFile.print(':');
myFile.print(now.second(), DEC);
myFile.print(",");
myFile.print(pitchAccel); // registra °Pitch
myFile.print(",");
myFile.print(rollAccel); // registra °Roll
myFile.print(",");
myFile.print(pesoKg); // registra peso
myFile.println();
myFile.close(); // cerramos el archivo
}
else
{
// si no se puede abrir, imprime error
Serial.println("Error al abrir el archivo");
}
delay(10);
} // Fin de lectura de datos del bastón
}
delay(850); // tiempo 1s (f_muestreo = 1 Hz)
} // fin de loop

/***** Funciones del programa *****/

// Inicia comunicación I2C
void WriteTo(byte address, byte val)
{
Wire.beginTransmission(ADXLAddress);
Wire.write(address);
Wire.write(val);
Wire.endTransmission();
}

// Lee 6 bytes del buffer en el ADXL345
void ReadFrom(byte address, int num, byte _buff[])
{
Wire.beginTransmission(ADXLAddress);
Wire.write(address);
Wire.endTransmission();
Wire.beginTransmission(ADXLAddress);
Wire.requestFrom(ADXLAddress, num);
int i = 0;
while(Wire.available())
{
_buff[i] = Wire.read();
i++;
}
Wire.endTransmission(); // fin de I2C
}

```

```

// Obtiene el offset para poder restarlo a cada medición
// posterior y obtener una lectura correcta

```

```

void ADXL345_GetBiass(void)
{
// se toma 1000 muestras
for (int i = 0; i < SAMPLESS_BIASS; i += 1)
{
ReadFrom( X_Axis_Register_DATAX0, BytesToRead,
_buff);
accel_X = (((int)_buff[1]) << 8) | _buff[0];
accel_Y = (((int)_buff[3]) << 8) | _buff[2];
accel_Z = (((int)_buff[5]) << 8) | _buff[4];
biass_X += accel_X;
biass_Y += accel_Y;
biass_Z += accel_Z;
delay(1);
}
biass_X /= SAMPLESS_BIASS; //se promedia
biass_Y /= SAMPLESS_BIASS;
biass_Z /= SAMPLESS_BIASS;
biass_Z -= 256;
} // fin de GetBiass

// Lee las aceleraciones para calcular los °pitch y °rolls
void lectura_adxl435()
{
for(int p=0; p<=5; p++) // limpio el buffer
{
_buff[p]=0;
}
ReadFrom(X_Axis_Register_DATAX0, BytesToRead,
_buff);
// cada lectura de eje viene con resolución de 10 bits
accel_X = (((int)_buff[1]) << 8) | _buff[0];
accel_Y = (((int)_buff[3]) << 8) | _buff[2];
accel_Z = (((int)_buff[5]) << 8) | _buff[4];
X=((float)accel_X - biass_X) * ACEL_X_SCALE;
Y=((float)accel_Y - biass_Y) * ACEL_Y_SCALE;
Z=((float)accel_Z - biass_Z) * ACEL_Z_SCALE;
// se mueve en sentido de Y (VERTICAL) y X (horizontal)
pitchAccel = (atan2(X,Z) * 360.0 / (2*PI));
// se mueve en sentido derecha o izquierda del eje Z
rollAccel = asin(Y) * 360.0 / (2*PI);
delay(100);
} // fin de lecturaADXL345

// Lee fecha y hora para grabar en el archivo
void dateTime(uint16_t* date, uint16_t* time)
{
DateTime now = RTC.now();
// retorna fecha, usa macro FAT_DATE para formatear
campos
*date = FAT_DATE(now.year(), now.month(), now.day());
// retorna tiempo, usa macro FAT_DATE para formatear
campos
*time = FAT_TIME(now.hour(), now.minute(),
now.second());
} // fin de dateTime

// Lee ADC, devuelve peso en libras

```

```
double lectura_ADC(void)
{
  LecturaADC = adc->adc0->analogRead(ReadPinADC9);
  voltaje = LecturaADC * resolucion;
  if(voltaje<0) voltaje=0;
  if(voltaje>=0.0 && voltaje<1.852) // recta 1
  {
    libras = (voltaje-1.213)/0.0263;
  }
  if(voltaje>=1.852) // recta 2
  {
    libras = (voltaje-1.598)/0.0105;
  }
  return libras;
} // fin de lecturaADC

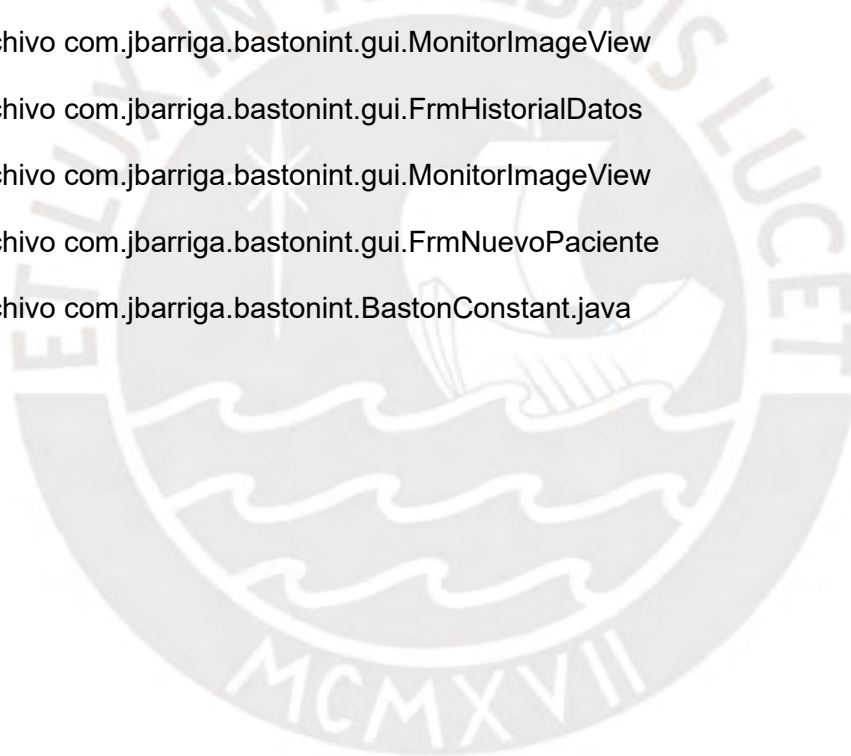
/**** Fin del programa del microcontrolador ****/
```



## 8.2. Programa del Aplicativo Móvil con Android Studio

A continuación se copiará el código fuente del aplicativo móvil dividido en las siguientes partes principales:

- Archivo: com.jbarriga.bastonint.bo.BluetoothMng.java
- Archivo: com.jbarriga.bastonint.dao.BastonBD.java
- Archivo: com.jbarriga.bastonint.domain.DatosPasoVO.java
- Archivo: com.jbarriga.bastonint.domain.PacienteVO
- Archivo: com.jbarriga.bastonint.gui.MainActivity
- Archivo com.jbarriga.bastonint.gui.FrmMenuPrincipal
- Archivo com.jbarriga.bastonint.gui.FrmMonitorTiempoReal
- Archivo com.jbarriga.bastonint.gui.MonitorImageView
- Archivo com.jbarriga.bastonint.gui.FrmHistorialDatos
- Archivo com.jbarriga.bastonint.gui.MonitorImageView
- Archivo com.jbarriga.bastonint.gui.FrmNuevoPaciente
- Archivo com.jbarriga.bastonint.BastonConstant.java



### 8.2.1. Archivo: com.jbarriga.bastonint.bo.BluetoothMng.java

```

package com.jbarriga.bastonint.bo;

import android.annotation.SuppressLint;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.util.Log;
import com.jbarriga.bastonint.BastonConstant;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.gui.FrmMenuPrincipal;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Set;
import java.util.UUID;
import makval.gimak.gui.GActivity;
import makval.gimak.io.GimakLogger;
import makval.gimak.utils.ThreadUtils;

/* Clase con logica de negocio para Bluetooth */
public class BluetoothMng

{
    /* ID del servicio Bluetooth */
    static final UUID remoteControlUUID =
    UUID.fromString("00001101-0000-1000-8000-
    805F9B34FB");

    /* ID de la peticion para que se inicie la interfaz
    bluetooth */
    public static final int REQUEST_ENABLE_BT = 1000;
    /* Maximo numero de reintentos de conexion */

```

```

    private static final int
    MAX_REINTNETOS_CONECCION=5;
    /* PIN para enlazarse con el baston */
    private static final String BLE_PIN = "2910";
    /* Procesa solicitud para emparejar el dispositivo
    bluetooth */
    private BroadcastReceiver broadCastBindReceiver;
    /* Procesa solicitud para detectar dispositivos
    bluetooth */
    private BroadcastReceiver
    broadCastBtDevicesReceiver;
    /* Canal de lectura de datos desde bluetooth */
    private OutputStream out;
    /* Canal de lectura de datos desde bluetooth */
    private InputStream in;
    /* Socket de comunicaciones */
    private BluetoothSocket socketBaston;
    /* Adaptador bluetooth */
    private BluetoothAdapter mBluetoothAdapter;
    /* Actividad con la que se trabaja el tema visual */
    private GActivity actividadActual;
    /* ID del baston */
    private String ble_id = "Baston";
    /* Referencia al baston */
    private BluetoothDevice btDevBaston;
    /* Identificación bluetooth del dispositivo enlazado
    * que representa el baston */
    private String btIdBaston;
    /* Numero de errores de conexion */
    private int numErroresCon = 0;
    /* Indica si se detecto el baston */
    private boolean bastonDetectado = false;
    /* ID del baston con el que se establecio la conexion */
    public static String nombreBastonEnlazado;
    /* Constructor @param activity */
    public BluetoothMng( GActivity activity )

```

```

{
    this.actividadActual = activity;
}
/* Establece la actividad actual @param activity */
public void setActActual( GActivity activity )
{
    this.actividadActual = activity;
}
/* Indica si es un nuevo paciente */
private boolean nuevoPaciente;

/* Inicia una conexion con el baston */
@SuppressLint("MissingPermission")
public synchronized void iniciaConexion( boolean
nuevoPaciente )
{
    this.nuevoPaciente = nuevoPaciente;
    mBluetoothAdapter =
    BluetoothAdapter.getDefaultAdapter();
    if (mBluetoothAdapter == null)
    {
        GimakLogger.d("InicializaControl","No hay soporte
de bluetooth");
        this.actividadActual.hideProgressDialog();
    }
    this.actividadActual.showAcceptAlert(this.actividadActual.
getString(R.string.tituloErrorInesperado),

this.actividadActual.getString(R.string.msgErr_DispNetoTie
neBluetooth),

this.actividadActual.getString(R.string.aceptar), new
DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int
which)

```

```

    {
    }
    });

    return;
}

if (BluetoothAdapter.isEnabled())
{
    Intent enableBluetoothIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

this.actividadActual.startActivityForResult(enableBluetoothIntent, REQUEST_ENABLE_BT);
}
else
{
    if (detectaBaston() == false)
    {
        // Carga la opción de bluetooth para enlazar dispositivos
        // enlazarBaston();
        actividadActual.hideProgressDialog();
        actividadActual.showAlert("Error Inesperado", "Por favor enlace el bastón e intente la conexión de nuevo.");
    }
}

}

/* Crea los streams de comunicacion */
private void creaStreams() throws IOException
{
    if (in == null)
    {
        in = socketBaston.getInputStream();
        out = socketBaston.getOutputStream();
    }
}

```

```

/* Envía un mensaje via bluetooth
 * @param comando comando que se envia
 * @return respuesta recibida desde el baston */
public String enviaMensajeBlueTooth( String comando
) throws IOException
{
    String msgError;
    String rpta;
    byte cmd;
    // BufferedWriter out;
    // BufferedReader in;
    if ( BastonConstant.ACEDER_BASTON == false )
    {
        return BastonConstant.RPTA_CMD_OK;
    }

    msgError="Error al crear canal de lectura de datos bluetooth";
    try
    {
        creaStreams();
        msgError="Error al enviar comando:"+comando;
        Log.e("BluetoothMng", "Enviando comando string:"+comando);
        out.write(comando.getBytes());
        out.flush();
        Thread.sleep(5);
        msgError="Error al recibir respuesta del comando:"+comando;
        // rpta=in.readLine();
        rpta=BastonConstant.RPTA_CMD_OK;
        return rpta;
    }
    catch ( Exception e )
    {
        GimakLogger.e("EnviaMensaeBlueTooth",msgError,e);
        throw new IOException(msgError,e);
    }
}

```

```

}

}

/* Lee un byte desde el bluetooth
 * @return valor leído */
public int leeByteBluetooth() throws IOException
{
    String msgError;
    msgError="Error al crear canal de lectura de datos bluetooth";
    try
    {
        creaStreams();
        return in.read();
    }
    catch ( Exception e )
    {
        GimakLogger.e("LeeMensaeBlueTooth",msgError,e);
        throw new IOException(msgError,e);
    }
}

/**
 * Inicia la deteccion del baston
 * @return true en caso se haya iniciado el proceso de deteccion, false en caso el baston
 * no este emparejado
 */
@SuppressWarnings("MissingPermission")
private synchronized boolean detectaBaston() {
    Set<BluetoothDevice> pairedDevices =
mBluetoothAdapter.getBondedDevices();
    // If there are paired devices
    if (pairedDevices.size() > 0)
    {
        // Loop through paired devices
        for (BluetoothDevice device : pairedDevices)
        {

```

```

GimakLogger.d("FrmControlRemoto",device.getName());
Log.d("FrmControlRemoto",device.getName());

// if ((( nuevoPaciente == true ) && (
device.getName().startsWith(ble_id) )) ||
// (( nuevoPaciente == false ) && (
device.getName().equals(nombreBastonEnlazado) )))
if ( device.getName().startsWith(ble_id) )
{
    nombreBastonEnlazado = device.getName();
    btIdBaston =device.getAddress();
    GimakLogger.d("FrmControlRemoto", "Baston
identificado:" + btIdBaston);
    actividadActual.showFastToast("Intentando
acceder a " + nombreBastonEnlazado);
    // actividadActual.runOnUiThread ( () ->
realizaConexionFinal() );
    try
    {
        realizaConexionFinal();
        return true;
    }
    catch ( Exception e )
    {
        // Se ignora el error, porque hay la
posibilidad que otros dispositivos Bluetooth

        // se conecten
        Log.d("FrmControlRemoto","No se pudo
conectar a :"+ device.getName());
        btIdBaston = null;
    }
}
else
{
    GimakLogger.d("FrmControlRemoto","Se
descarta BLT: "+ device.getName());
    actividadActual.showFastToast("Ignorando el
dispositivo " + nombreBastonEnlazado);

```

```

    }
}
}
}
actividadActual.showFastToast("Se finalizo la
validación de todos los dispositivos enlazado");
btIdBaston = null;

return false;
}

/* Inicia la conexion con el baston */
@SuppressWarnings("MissingPermission")
private void realizaConexionFinal() throws IOException
{
    String msgError,rpta;
    if ( socketBaston != null )
    {
        return;
    }
    msgError="No se pudo obtener la referencia al
baston";
    btDevBaston =
mBluetoothAdapter.getRemoteDevice(btIdBaston);
    msgError="No se pudo abrir el socket";
    socketBaston =
btDevBaston.createInsecureRfcommSocketToServiceReco
rd(remoteControlUUID);
    ThreadUtils.sleep(1000);
    msgError="No se pudo conectar el socket";
    int numError=0;
    while( true )
    {
        try
        {
            socketBaston.connect();
            break;
        }
        catch ( IOException e)

```

```

{
    numError++;
    ThreadUtils.sleep(500);

    if ( numError > 1 )
    {
        socketBaston = null;
        throw e;
    }
}
}
}
actividadActual.hideProgressDialog();
// Carga el menu principal
Intent intent = new Intent(actividadActual,
FrmMenuPrincipal.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
actividadActual.startActivity(intent);
}

/* Retorna la data disponible
 * @return -1 en caso de error, o la cantidad de bytes
disponibles */
public int getDataDisponible()
{
    try
    {
        creaStreams();
        return in.available();
    }
    catch ( Exception e )
    {
        Log.e("BluetoothMng","Error en
getDataDisponible()",e);
        GimakLogger.e("BluetoothMng","Error al obtener
datos disponibles",e);
        return -1;
    }
}
}

```

```

    /* Lee todos los bytes desde el bluetooth hasta que no
    quede ninguno disponible
    * @throws IOException */
    public void limpiarBuffer() throws IOException
    {
        try
        {
            creaStreams();
            while ( in.available() > 0 )
            {
                in.read();
                Thread.sleep(5);
            }
        }
        catch ( Exception e )
        {
            // silenciado al proposito
        }
    }

    /* Envia un mensaje via bluetooth
    * @param comando comando que se envia
    * @return respuesta recibida desde el baston
    */
    public String enviaMensajeBlueTooth( int comando )
    throws IOException
    {
        String msgError;
        String rpta;
        // BufferedWriter out;
        // BufferedReader in;
        msgError="Error al crear canal de lectura de datos
        bluetooth";
        try
        {
            creaStreams();
            msgError="Error al enviar comando:"+comando;
            Log.e("BluetoothMng","Enviando

```

```

comando:"+comando);
        out.write(comando);
        out.flush();
        Thread.sleep(5);
        msgError="Error al recibir respuesta del
comando:"+comando;
        // rpta=in.readLine();
        rpta=BastonConstant.RPTA_CMD_OK;
        return rpta;
    }
    catch ( Exception e )
    {
        GimakLogger.e("BluetoothMng",msgError,e);
        throw new IOException(msgError,e);
    }
}

/* Cierra el socket con el control remoto */
public void closeRemoteSocket()
{
    if ( socketBaston == null )
    {
        return;
    }
    try
    {
        socketBaston.close();
        Log.d("BluetoothMng","Cerrando Socket");
    }
    catch ( Exception e )
    {
        Log.e("BluetoothMng","Error al cerrar el socket
        bluetooth",e);
    }
}

/* Enlaza el celular con el baston */
@SuppressLint("MissingPermission")

```

```

private void enlazarBaston()
{
    if ( broadCastBindReceiver != null )
    {
        return;
    }

    if ( BastonConstant.ACEDER_BASTON == false )
    {
        return;
    }

    // Crea el procesador de mensajes de
    emparejamiento
    broadCastBindReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent
        intent) {
            String action = intent.getAction();
            if(
            BluetoothDevice.ACTION_PAIRING_REQUEST.equals(action))
            {
                BluetoothDevice bluetoothDevice =
                intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVIC
                E);

                bluetoothDevice.setPin(BLE_PIN.getBytes());

                GimakLogger.d("IniciaConexionConbaston","Auto-
                entering pin: " + BLE_PIN);
                bluetoothDevice.createBond();

                actividadActual.unregisterReceiver(broadCastBindReceiv
                er);

                broadCastBtDevicesReceiver=null;

                actividadActual.runOnUiThread() ->
                {

```

```

        try
        {
            Thread.sleep(1000);
            iniciaConexion(nuevoPaciente);
        }
        catch ( Exception e )
        {
        }
    });
}
};

IntentFilter filter = new
IntentFilter(BluetoothDevice.ACTION_PAIRING_REQUEST
);

actividadActual.registerReceiver(broadCastBindReceiver,
filter);

// Crea el procesador de mensajes por dispositivo
bluetooth detectado
broadCastBtDevicesReceiver = new
BroadcastReceiver() {
    public void onReceive(Context context, Intent
intent) {
        String action = intent.getAction();

        //Finding devices
        if
(BluetoothDevice.ACTION_FOUND.equals(action))
        {
            // Get the BluetoothDevice object from the
Intent
            BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVIC
E);

GimakLogger.d("FrmControlRemoto", "Nombre

```

```

Dispositivo:"+device.getName());

GimakLogger.d("FrmControlRemoto", "Direccion
Dispositivo:"+device.getAddress());
        if ( device.getName().startsWith(ble_id) )
        {
            bastonDetectado=true;
            device.createBond();

actividadActual.unregisterReceiver(broadCastBtDevicesR
eceiver);

            broadCastBtDevicesReceiver=null;
            mBluetoothAdapter.cancelDiscovery();
        }
    }
};

filter = new
IntentFilter(BluetoothDevice.ACTION_FOUND);

actividadActual.registerReceiver(broadCastBtDevicesRec
eiver, filter);
bastonDetectado=false;
actividadActual.runOnThread( () ->
validaTimeoutDeteccion() );
mBluetoothAdapter.startDiscovery();
}

/* Procesa un timeout para la deteccion del baston */
@SuppressLint("MissingPermission")
private void validaTimeoutDeteccion()
{
    int numSeg=0;

    while( !bastonDetectado )
    {
        try
        {

```

```

            Thread.sleep(1000);
            numSeg++;
        }
        catch ( Exception e )
        {
        }

        if ( numSeg == 30 )
        {
            mBluetoothAdapter.cancelDiscovery();
            break;
        }
    }

    if ( bastonDetectado == false )
    {
        actividadActual.runOnThread(() ->
finalizaDeteccionError());
    }
}

/* Oculta el proceso de deteccion, muestra un mensaje
de error por no detectar
* el baston, cierra la pantalla y visualiza el menu
principal */
private void finalizaDeteccionError()
{
    if ( broadCastBtDevicesReceiver != null )
    {

actividadActual.unregisterReceiver(broadCastBtDevicesR
eceiver);
        broadCastBtDevicesReceiver=null;
    }
    if ( broadCastBindReceiver != null )
    {

actividadActual.unregisterReceiver(broadCastBindReceiv

```

```

er);
    broadCastBindReceiver=null;
}

    actividadActual.hideProgressDialog();

    actividadActual.showAcceptAlert(actividadActual.getStrin
g(R.string.tituloErrorInesperado),

    actividadActual.getString(R.string.msgErr_NoSeDetectoB
aston),
    actividadActual.getString(R.string.aceptar), new
DialogInterface.OnClickListener()
    {
        @Override
        public void onClick(DialogInterface dialog, int
which)
        {

    actividadActual.startFirstActivity(FrmMenuPrincipal.class
);
        }
    });
}

    /* Lee un string desde el bluetooth
    * @param numDatos cantidad de bytes que se deben
    leer desde el string
    * @return string leído */
    public String readString( int numDatos )
    {
        String rpta = "";
        int dato;
        int numPausas = 0;
        try
        {
            while (true)
            {
                if ( getDataDisponible() == 0 )

```

```

{
    hacerPausa(100);
    numPausas++;
    if ( numPausas > 100 )
    {
        // timeout han pasado 10 segundos y no se
        puede leer el string
        return null;
    }
    continue;
}
numPausas = 0;
dato = leeByteBluetooth();
rpta = rpta + ((char)dato);

numDatos--;
if ( numDatos == 0 )
{
    return rpta;
}
}
}
catch ( Exception e )
{
    return null;
}
}

    /* Lee un string hasta que se reciba un separador
    * @param separador caracter que
    * @return */
    public String readString( String separador )
    {
        String rpta = "",letra;
        int dato;
        int numPausas = 0;
        try
        {
            while (true)

```

```

{
    if ( getDataDisponible() == 0 )
    {
        hacerPausa(100);
        numPausas++;
        if ( numPausas > 100 )
        {
            // timeout han pasado 10 segundos y no se
            puede leer el string
            return null;
        }
        continue;
    }
    numPausas = 0;
    dato = leeByteBluetooth();
    letra = "" + ((char)dato);
    if ( separador.equals(letra) )
    {
        return rpta;
    }

    rpta = rpta + letra;
}
}
catch ( Exception e )
{
    return null;
}
}

    /* Hacer pause de N milisegundos */
    private void hacerPausa( long duracion )
    {
        try
        {
            Thread.sleep(duracion);
        }
        catch ( Exception e )
        {

```

```

    }
}

/* Cierra la conexion con bluetooth
 * @param actividad Actividad desde la que se invoca
 el metodo */
public void close( GActivity actividad )
{
    if ( socketBaston == null )
    {

```

```

        return;
    }
    try
    {
        socketBaston.close();
    }
    catch ( Exception e )
    {
        actividad.showAlert(actividad.getString(R.string.tituloErr

```

```

        orInesperado),
        actividad.getString(R.string.error_no_se_pudo_cerrar_Bl
        uetooth));
    }
    socketBaston = null;
}
}

```



## 8.2.2. Archivo: com.jbarriga.bastonint.dao.BastonBD.java

```
package com.jbarriga.bastonint.dao;
```

```
import android.content.Context;
import com.jbarriga.bastonint.bo.BluetoothMng;
import java.io.IOException;
import makval.gimak.gorm.GDbOption;
import makval.gimak.gorm.GeneralTableBO;
import makval.gimak.gorm.GormConnection;
import makval.gimak.gorm.GormSession;
/**
 * Representa la interfaz para acceder a la base de datos
 de la aplicación
 */
public class BastonBD extends GormConnection
{
    /**
     * Nombre del parametro de la BD en el que se
     almacena
     * el nombre del paciente
     */
    public static final String PARAM_BD_NOMBRE =
"param_nombre";
    /**
     * Nombre del parametro de la BD en el que se
     almacena
     * la historia clinica
     */
    public static final String
PARAM_BD_HISTORIA_CLINICA =
"param_historia_clinica";
    /**
     * Nombre del parametro de la BD en el que se
     almacena
     * el peso del paciente
     */
    public static final String PARAM_BD_PESO =
"param_peso";
```

```
    /**
     * Nombre del parametro de la BD en el que se
     almacena
     * la edad del paciente
     */
    public static final String PARAM_BD_EDAD =
"param_edad";
    /**
     * Nombre del parametro de la BD en el que se
     almacena
     * la edad del paciente
     */
    public static final String PARAM_BD_BASTON =
"param_nombreBaston";
    /**
     * Version de la base de datos
     */
    public static int BD_VERSION = 1;
    /**
     * Nombre de la base de datos
     */
    public static String DBName = "baston.db";
    /**
     * Constructor
     */
    public BastonBD(Context context )
    {
        // Indica el nombre y la version de la base de datos
        super(context,DBName, BD_VERSION);
    }
    /**
     * Registra las clases que se deben mapear a tablas
     */
    @Override
    public void registerClasses()
    {
        addTablaClass(GDbOption.class);
```

```
    }
    /**
     * Guardar los datos del paciente en la BD
     * @param nombre nombre del paciente
     * @param historia numero de la historia clinica
     * @param peso peso en kilos del paciente
     * @param edad edad del paciente
     */
    public void guardarDatosPaciente( String nombre,
String historia, int peso, int edad ) throws IOException
    {
        String msgError;
        GormSession session =
GormSession.getInstance(this);
        GeneralTableBO bo = new GeneralTableBO(session);

        msgError = "Error al guardar el nombre del
paciente";
        try
        {
            bo.saveAppParam(PARAM_BD_NOMBRE,nombre);
            msgError = "Error al guardar la historia clinica";

            bo.saveAppParam(PARAM_BD_HISTORIA_CLINICA,histori
a);

            msgError = "Error al guardar el peso del paciente";
            bo.saveAppParam(PARAM_BD_PESO,"" + peso);
            msgError = "Error al guardar la edad del paciente";
            bo.saveAppParam(PARAM_BD_EDAD,"" + edad);
            msgError = "Error al guardar el nombre del
baston";
            bo.saveAppParam(PARAM_BD_BASTON,"" +
BluetoothMng.nombreBastonEnlazado);
            session.commit();
        }
        catch ( Exception e )
```

```

{
    session.rollback();
    throw new IOException(msgError);
}
finally
{
    session.close();
}
}

/**
 * Retorna un parametro de la BD
 * @param nombreParametro
 * @return

```

```

* @throws IOException
*/
public String getParametro( String nombreParametro )
throws IOException
{
    String msgError;
    GormSession session =
GormSession.getInstance(this);
    GeneralTableBO bo = new GeneralTableBO(session);

    msgError = "Error al guardar el nombre del
paciente";
    try
    {

```

```

        return bo.getAppParam(nombreParametro);
    }
    catch ( Exception e )
    {
        throw new IOException(msgError);
    }
    finally
    {
        session.close();
    }
}
}

```



### 8.2.3. Archivo: com.jbarriga.bastonint.domain.DatosPasoVO.java

```
package com.jbarriga.bastonint.domain;
```

```
import java.util.Date;
```

```
/**
 * Datos de un paso detectado
 */
```

```
public class DatosPasoVO
```

```
{
    /* Fecha del evento */
    private Date fecha;
    /* Pith del baston */
    private float pitch;
    /* ROLL del baston */
    private float roll;
    /* Peso */
    private float peso;
    /* Secuencia del evento en el dia */
    private int secuendiaDia;
    /* Constructor */
    public DatosPasoVO()
    {
```

```
}
```

```
    public Date getFecha() {
        return fecha;
    }
```

```
    public void setFecha(Date fecha) {
        this.fecha = fecha;
    }
```

```
    public float getPitch() {
        return pitch;
    }
```

```
    public void setPitch(float pitch) {
        this.pitch = pitch;
    }
```

```
    public float getRoll() {
        return roll;
    }
```

```
    public void setRoll(float roll) {
        this.roll = roll;
    }
```

```
    public float getPeso() {
        return peso;
    }
```

```
    public void setPeso(float peso) {
        this.peso = peso;
    }
```

```
    public int getSecuendiaDia() {
        return secuendiaDia;
    }
```

```
    public void setSecuendiaDia(int secuendiaDia) {
        this.secuendiaDia = secuendiaDia;
    }
}
```



#### 8.2.4. Archivo: com.jbarriga.bastonint.domain.PacienteVO

```

package com.jbarriga.bastonint.domain;
/**
 * Datos de un paciente
 */
public class PacienteVO
{
    /* Nombre del paciente */
    private String nombre;

    /* Historia clinica */
    private String historiaClinica;

    /* Peso */
    private int peso;

    /* Edad */
    private int edad;

    /* Constructor */
    public PacienteVO()
    {
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {

```

```

        this.nombre = nombre;
    }

    public String getHistoriaClinica() {
        return historiaClinica;
    }

    public void setHistoriaClinica(String historiaClinica) {
        this.historiaClinica = historiaClinica;
    }

    public int getPeso() {
        return peso;
    }

    public void setPeso(int peso) {
        this.peso = peso;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }
}

```

### 8.2.5. Archivo: com.jbarriga.bastonint.gui.MainActivity

```
package com.jbarriga.bastonint.gui;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import com.jbarriga.bastonint.BastonConstant;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.bo.BluetoothMng;
import com.jbarriga.bastonint.dao.BastonBD;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Date;
import java.util.Set;
import java.util.UUID;
import makval.gimak.gui.GActivity;
import makval.gimak.io.GimakLogger;

/**
 * Punto de ingreso a la aplicacion
```

```
 */
public class MainActivity extends GActivity {
    public static BluetoothMng bluetoothMng;

    // private ActivityResultLauncher<String>
    requestPermissionLauncher;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GimakLogger.context = this;
        MainActivity.bluetoothMng = new
        BluetoothMng(this);
        initGui();
        solicitaAutorizaciones();
    }

    /**
     * Inicializa los componentes visuales
     */
    private void initGui() {
        ImageView imageView;

        imageView = (ImageView)
        findViewById(R.id.imgConectar);
        imageView.setOnClickListener((view) ->
        imgConectar_onClick());
        imageView.setVisibility(View.INVISIBLE);
    }

    /**
     * Solicita autorizaciones para poder trabajar
     */
    private void solicitaAutorizaciones() {
        boolean autorizado;

        if (android.os.Build.VERSION.SDK_INT >=
```

```
Build.VERSION_CODES.M) {
        autorizado = !pideAccesos();
    } else {
        autorizado = true;
    }

    if (autorizado == true) {

        findViewById(R.id.imgConectar).setVisibility(View.VISIBLE);
    }
    }

    /**
     * Pide accesos a los diferentes servicios
     * @return true si se solicito acceso, false si no se
    solicito acceso
     */
    @TargetApi(23)
    private boolean pideAccesos() {

        if ( ContextCompat.checkSelfPermission(this,
        Manifest.permission.INTERNET) !=
        PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.INTERNET}, 1000);

            return true;
        }

        if ( ContextCompat.checkSelfPermission(this,
        Manifest.permission.BLUETOOTH) !=
        PackageManager.PERMISSION_GRANTED)
        {
            if (
            ActivityCompat.shouldShowRequestPermissionRationale(
            this,Manifest.permission.BLUETOOTH) )
```

```

    {
        return true;
    }

    ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.BLUETOOTH}, 1002);

    return true;
}

    if ( ContextCompat.checkSelfPermission(this,
Manifest.permission.BLUETOOTH_ADMIN) !=
PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.BLUETOOTH_ADMIN},
1002);

        return true;
    }

    return false;
}

@Override
public void onRequestPermissionsResult(int
requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
    if (grantResults[0] ==

```

```

PackageManager.PERMISSION_GRANTED) {
    solicitaAutorizaciones();
    } else {

showAlert(getString(R.string.tituloErrorValidacion),
getString(R.string.msgErr_AutorizacionesApp));
    }
    }

/**
 * Procesa el evento click sobre el ICONO de conectar /
desconectar
 */
private void imgConectar_onClick() {
    BastonBD bastonBD = new BastonBD(this);

showProgressDialog(getString(R.string.msgUbicandoBast
on));

    try
    {
        BluetoothMng.nombreBastonEnlazado =
bastonBD.getParametro(BastonBD.PARAM_BD_BASTON)
;

        bastonBD.close();
        runOnUiThread()->
MainActivity.bluetoothMng.iniciaConexion(BluetoothMn
g.nombreBastonEnlazado == null) );
    }
    catch( Exception e )

```

```

    {
        hideProgressDialog();
        bastonBD.close();
    }
}

@Override
protected void onActivityResult(int requestCode, int
resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode,
data);

    switch( requestCode )
    {
        case BluetoothMng.REQUEST_ENABLE_BT:
            if ( requestCode == RESULT_OK ) {
                runOnUiThread() -> {
                    try {
                        Thread.sleep(2000);
                        bluetoothMng.iniciaConexion(
BluetoothMng.nombreBastonEnlazado == null) );
                    } catch (Exception e) {

                    }
                }
            }
        }
    }
}
}

```

## 8.2.6. Archivo com.jbarriga.bastonint.gui.FrmMenuPrincipal

```

package com.jbarriga.bastonint.gui;

import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.TextView;
import com.jbarriga.bastonint.BastonConstant;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.bo.BluetoothMng;
import com.jbarriga.bastonint.dao.BastonBD;
import makval.gimak.gui.GActivity;
import makval.gimak.gui.InputDialogListener;

/**
 * Representa el menú principal de la aplicación, una vez
 * iniciada la conexión con BlueTooth
 */
public class FrmMenuPrincipal extends GActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_frm_menu_principal);
        initGui();
    }

    /**
     * Inicializa los componentes visuales
     */
    private void initGui()
    {
        BastonBD bastonBD = new BastonBD(this);
        String txt;
        TextView textView;

```

```

        CheckBox checkBox;
        View view;
        Button boton;

        boton = (Button)findViewById(R.id.btnSalir);
        boton.setOnClickListener( (v) -> btnSalir_onClick() );

        view = findViewById(R.id.pnlMemoriaSD);
        view.setOnClickListener( (v) ->
        pnlMemoriaSD_onClick() );

        view = findViewById(R.id.pnlNuevoPaciente);
        view.setOnClickListener( (v) ->
        pnlNuevoPaciente_onClick() );

        view = findViewById(R.id.pnlMonitor);
        view.setOnClickListener( (v) -> pnlMonitor_onClick()
        );

        view = findViewById(R.id.pnlMostrarDatos);
        view.setOnClickListener( (v) ->
        pnlMostrarDatos_onClick() );

        checkBox = (CheckBox)
        findViewById(R.id.chkUsarDatosSimulados);
        // checkBox.setChecked(true);
        try
        {
            txt =
            bastonBD.getParametro(BastonBD.PARAM_BD_NOMBR
            E);
            if ( txt == null )
            {
                txt =
                getString(R.string.paciente_nuevo).toLowerCase();

                Intent intent = new Intent(this,

```

```

        FrmNuevoPaciente.class);
            startActivity(intent);
        }
        else
        {
            textView =
            (TextView)findViewById(R.id.lblNombrePaciente);
            textView.setText("Bienvenido(a) " +
            txt.substring(0,1).toUpperCase() + txt.substring(1));

            txt =
            bastonBD.getParametro(BastonBD.PARAM_BD_PESO);
            BastonConstant.pesoPaciente =
            Integer.parseInt(txt);
        }
        catch ( Exception e )
        {
            showAlert(getString(R.string.tituloErrorInesperado),getSt
            ring(R.string.error_leer_nombre_paciente));
        }
    }

    /**
     * Procesa la seleccion del menu memoria
     */
    private void pnlMemoriaSD_onClick()
    {
        Intent intent = new Intent(this, FrmMemoria.class);
        startActivity(intent);
    }

    /**
     * Procesa la seleccion del crear un nuevo paciente
     */
    private void pnlNuevoPaciente_onClick()
    {

```

```

showInputDialog(getString(R.string.tituloPasswordDoctor
), "", InputType.TYPE_CLASS_TEXT, new
InputDialogListener() {
    @Override
    public void onInputDialog(String value) {

        if ( "123456".equals(value) )
        {
            cargaVentanaNuevoPaciente();
        }
    }

    @Override
    public void onCancel() {

    }
});
}

/**
 * Carga la ventana para un nuevo paciente
 */
private void cargaVentanaNuevoPaciente()
{
    Intent intent = new Intent(this,
FrmNuevoPaciente.class);
    startActivity(intent);
}

```

```

/**
 * Procesa la seleccion para mostrar el monitor en
tiempo real
 */
private void pnlMonitor_onClick()
{
    Intent intent = new Intent(this,
FrmMonitorTiempoReal.class);
    startActivity(intent);
}

/**
 * Procesa el evento click para cargar la pantalla que
muestra los datos historicos
 */
private void pnlMostrarDatos_onClick()
{
    CheckBox checkBox =
(CheckBox)findViewById(R.id.chkUsarDatosSimulados);
    Intent intent = new Intent(this,
FrmHistorialDatos.class);

    intent.putExtra(FrmHistorialDatos.PARAM_USAR_DATOS
_SIMULADOS, checkBox.isChecked());
    startActivity(intent);
}

```

```

@Override
public void onBackPressed()
{
    MainActivity.bluetoothMng.close(this);
    super.onBackPressed();
}

/**
 * Procesa el click sobre el boton salir
 */
private void btnSalir_onClick()
{

showConfirmES(getString(R.string.tituloConfirmarOperac
ion), getString(R.string.confirmarSalir),
    (p1, p2) -> btnSalir_onAccept(), null );
}

/**
 * Procesa la confirmacion de salir
 */
private void btnSalir_onAccept()
{
    onBackPressed();
}
}

```

### 8.2.7. Archivo com.jbarriga.bastonint.gui.FrmMonitorTiempoReal

```

package com.jbarriga.bastonint.gui;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.bo.BluetoothMng;
import java.io.IOException;
import makval.gimak.gui.GActivity;

/**
 * Pantalla que muestra el estado del bastón en tiempo
 real, para aprender a usar el bastón */
public class FrmMonitorTiempoReal extends GActivity {
    /**
     * Indica si debe o no terminar el bucle de monitoreo
     */
    private boolean finalizarBucle;

    /** Administrador de conectividad con bluetooth */
    private BluetoothMng bluetoothMng;

    /** Indica que se esta esperando a que se actualice la
 pantalla */
    private boolean esperandoGui;

    /**
     * Visor de los datos en tiempo real */
    private MonitorImageView monitorImageView;

    /** Reproductor del beep pitch */
    private MediaPlayer mediaPlayerPith;

```

```

    /** Reproductor del beep roll */
    private MediaPlayer mediaPlayerRoll;

    /** Indica si se reproduce o no el sonido de alarma */
    private boolean sonidoON = true;

    /** Inicia el componente
     * @param savedInstanceState */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_frm_monitor_tiempo_r
eal);

        Toolbar toolbar = (Toolbar)
        findViewById(R.id.toolbar);
        toolbar.setTitle(R.string.app_name);

        toolbar.setSubtitle(R.string.tituloMonitorTiempoReal);
        setGoBackImage(R.id.imgRetroceder);

        bluetoothMng = MainActivity.bluetoothMng;
        mediaPlayerPith = MediaPlayer.create(this,
R.raw.beep07a);
        mediaPlayerRoll = MediaPlayer.create(this,
R.raw.beep08b);

        monitorImageView =
        (MonitorImageView)findViewById(R.id.imgVistaPersonas
);
        monitorImageView.setMonitorTiempoReal(this);
        finalizarBucle = false;
        runOnUiThread(() -> bucleBaston());
        initgui();
    }

```

```

    /** Inicializa los componentes visuales */
    private void initgui()
    {
        ImageView image;
        image = (ImageView)
        findViewById(R.id.imgAudioON);
        image.setOnClickListener( (v) ->
        imgAudioON_onClick());
        image = (ImageView)
        findViewById(R.id.imgAudioOFF);
        image.setOnClickListener( (v) ->
        imgAudioOFF_onClick());
    }

    /** Procesa el CLICK para ativar el sonido */
    void imgAudioOFF_onClick()
    {
        sonidoON = true;
        findViewById(R.id.imgAudioON).setVisibility(View.VISIBL
E);

        findViewById(R.id.imgAudioOFF).setVisibility(View.GONE
);
    }

    /** Procesa el CLICK para ativar el sonido */
    void imgAudioON_onClick()
    {
        sonidoON = false;

        findViewById(R.id.imgAudioON).setVisibility(View.GONE)
;

        findViewById(R.id.imgAudioOFF).setVisibility(View.VISIBL
E);
    }

```

```

/* Reproduce el beep para el pitch */
public void beepPith()
{
    if ( sonidoON)
    {
        mediaPlayerPith.start();
    }
}

/* Reproduce el beep para el Roll */
public void beepRoll()
{
    if ( sonidoON )
    {
        mediaPlayerRoll.start();
    }
}

/** Valida si debe o no terminar de ejecutar el bucle de monitoreo
 * @return
 */
private synchronized boolean getFinalizarBucle()
{
    return this.finalizarBucle;
}

/* Indica que se debe finalizar el bucle */
private synchronized void setFinalizarBucle()
{
    this.finalizarBucle = true;
}

/** Bucle que se ejecuta mientras este conectado con el baston */
private void bucleBaston()
{
    int estado = 0;

```

```

    try
    {
        bluetoothMng.limpiarBuffer();
    }
    catch ( Exception e )
    {
        // silenciado al propocito
    }

    while( getFinalizarBucle() == false )
    {
        if ( estado == 0 )
        {
            if ( solicitarEnvioData() == true )
            {
                estado = 1;
            }
        }
        else
        {
            if ( bluetoothMng.getDataDisponible() > 0 )
            {
                leerData();
            }
        }
        solicitaFinalizarEnvioDatos();
        Log.e("FrmMonitorTiempoReal", "Fin del bucle de lectura de datos");
    }

    /* Solicita al Bluetooth finalizar el envio de datos */
    private void solicitaFinalizarEnvioDatos()
    {
        Log.d("FrmMonitorTiempoReal", "Envia codigo para finalizar bucle tiempo real");
        try
        {

```

```

            bluetoothMng.enviaMensajeBlueTooth('f');
        }
        catch ( Exception e )
        {
            Log.e("FrmMonitorTiempoReal", "Error al solicitar no enviar datos");
        }
    }

    /* Hacer pause de N milisegundos */
    private void hacerPausa( long duracion )
    {
        try
        {
            Thread.sleep(duracion);
        }
        catch ( Exception e )
        {
        }
    }

    /* Solicita al servidor que se */
    private boolean solicitarEnvioData()
    {
        int rpta;
        try
        {
            Log.d("FrmMonitorTiempoReal", "Solicita enviar data tiempo real");
            bluetoothMng.enviaMensajeBlueTooth('5');
            rpta = bluetoothMng.leeByteBlueTooth();

            Log.d("FrmMonitorTiempoReal", "Rpta.Solicitud enviar data tiempo read!" + ((char)rpta));

            if ( rpta == 'o' ) return true;

            return false;
        }
    }

```

```

    catch ( Exception e )
    {
        Log.e("FrmMonitorTiempoReal", "Error al solicitar
enviar data en tiempo real",e);
        return false;
    }
}

/* Lee los datos en tiempo real desde el dispositivo
bluetooth */
private boolean leerData()
{
    String dato;
    int comando;
    TextView textView;
    String valores[] = new String[3];
    valores[2] = "50";
    try
    {
        comando = bluetoothMng.leeByteBluetooth();
        if ( comando != 'd')
        {
            Log.d("FrmMonitorTiempoRal", "Comando no
identificado:" + (char)comando);
            return false;
        }

        dato = bluetoothMng.readString("#");
        if ( dato == null )
        {
            throw new IOException("NO se leyo el PITCH
desde el bluetooth");
        }
        valores[0] = dato;
        dato = bluetoothMng.readString("#");
        if ( dato == null )
        {
            throw new IOException("NO se leyo el ROLL
desde el bluetooth");

```

```

    }
    valores[1] = dato;

    dato = bluetoothMng.readString("#");
    if ( dato == null )
    {
        throw new IOException("NO se leyo el peso
desde el bluetooth");
    }
    valores[2] = dato;

    monitorImageView.setData(Double.parseDouble(valores[
0]), Double.parseDouble(valores[1]),
Double.parseDouble(valores[2]));
    setEsperandoGui(true);
    runOnUiThread( () -> solicitaRefrescarVista() );

    while( getEsperarGui() == true )
    {
        hacerPausa(20);
    }
    // indica al baston que ya se redibujó
    bluetoothMng.enviaMensajeBlueTooth('r');

    return true;
}
catch ( Exception e )
{
    Log.e("FrmMonitorTiempoReal", "Error leyendo
datos en tiempo real:" +valores[0]+ " "+valores[1] + "
"+valores[2],e);
    try
    {
        bluetoothMng.enviaMensajeBlueTooth('r');
    }
    catch ( Exception e1 )
    {
    }
    return false;

```

```

    }
}

/* Retorna el valor del flag para esperar al GUIs */
public synchronized boolean getEsperarGui()
{
    return esperandoGui;
}

/* Establece el valor del Flag de espera del GUI
* @param valor */
public synchronized void setEsperandoGui( boolean
valor )
{
    esperandoGui = valor;
}

@Override
protected void onStop()
{
    super.onStop();
    setFinalizarBucle();
}

@Override
protected void onRestart() {
    super.onRestart();
    finalizarBucle = false;
    runOnUiThread(() -> bucleBaston() );
}

/* Solicita refrescar la vista de datos */
private void solicitaRefrescarVista()
{
    monitorImageView.invalidate();
}
}

```

## 8.2.8. Archivo com.jbarriga.bastonint.gui.MonitorImageView

```
package com.jbarriga.bastonint.gui;
```

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import com.jbarriga.bastonint.BastonConstant;
import com.jbarriga.bastonint.R;
import java.io.InputStream;
```

```
/* Representa el visor del baston, el que muestra los
termómetros y la animación en tiempo real */
```

```
public class MonitorImageView extends
androidx.appcompat.widget.AppCompatImageView
{
```

```
    /* Configuración del pintado */
    private Paint currentPaint;
```

```
    /* Configuración del pintado de línea de ángulo */
    private Paint currentPaintAngulo;
```

```
    /* Componente monitor que contiene al visor */
    private FrmMonitorTiempoReal monitorTiempoReal;
```

```
    /* Imagen original */
    Bitmap bmpPersonasOrig;
```

```
    /* Ángulo de la vista frontal */
    float pithActual = 4;
    /* Ángulo de la vista lateral */
    float rollActual = 45;
```

```
    /* Peso actual */
    float pesoActual = 0;

    /**
     * Constructor
     * @param context
     * @param attrs
     */
    public MonitorImageView(Context context,
        AttributeSet attrs) {
        super(context, attrs);

        currentPaint = new Paint();
        currentPaint.setDither(true);
        currentPaint.setColor(0xFF00CC00); // alpha.r.g.b
        currentPaint.setStyle(Paint.Style.STROKE);
        currentPaint.setStrokeJoin(Paint.Join.ROUND);
        currentPaint.setStrokeCap(Paint.Cap.ROUND);
        currentPaint.setStrokeWidth(4);
        currentPaintAngulo = new Paint();
        currentPaintAngulo.setDither(true);
        currentPaintAngulo.setColor(0xFFFF0000); //
        alpha.r.g.b
        currentPaintAngulo.setStyle(Paint.Style.STROKE);

        currentPaintAngulo.setStrokeJoin(Paint.Join.ROUND);

        currentPaintAngulo.setStrokeCap(Paint.Cap.ROUND);
        currentPaintAngulo.setStrokeWidth(4);

        InputStream is =
        getResources().openRawResource(+R.drawable.persona_
        vista_guia);
        bmpPersonasOrig =
        BitmapFactory.decodeStream(is);
    }
```

```
    /**
     * Establece el monitor en el que se visualiza el
     componente
     * @param monitorTiempoReal
     */
    public void setMonitorTiempoReal(
        FrmMonitorTiempoReal monitorTiempoReal )
    {
        this.monitorTiempoReal = monitorTiempoReal;
    }

    /* Establece la data que se debe visualizar.
     * @param pithParam
     * @param rollParam
     * @param peso
     */
    public synchronized void setData( double pithParam,
        double rollParam, double peso )
    {
        float pith,roll;

        Log.d("MonitorImageView","Pitch Orig:" +
        pithParam + " ROLL Orig:" + rollParam + " Peso: " + peso);

        pith = (float)pithParam;
        roll = (float)rollParam;
        pesoActual = (float) peso;

        if (( pith != pithActual) || ( roll != rollActual))
        {
            pithActual = pith;
            rollActual = roll;
        }
    }
```

```

/* Actualiza la vista en tiempo real
 * @param canvas
 */
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    TextView txtAngulo;

    View termometroLateral =
    rootView.findViewById(R.id.imgTermometroLateral
    );
    View termometroFrontal =
    rootView.findViewById(R.id.imgTermometroFrontal
    );
    View tacometro =
    rootView.findViewById(R.id.imgTacometro);
    View separador =
    rootView.findViewById(R.id.separador);

    txtAngulo =
    rootView.findViewById(R.id.lblAnguloFrontal);
    txtAngulo.setText(""+ rollActual);

    txtAngulo =
    rootView.findViewById(R.id.lblAnguloLateral);
    txtAngulo.setText(""+ pithActual+" ");

    String dato = "" + pesoActual;
    txtAngulo =
    rootView.findViewById(R.id.lblPeso);
    txtAngulo.setText(" "+dato + " Kg");

    Bitmap personasFinal;

    // Establece el color del termometro lateral
    if ( Math.abs(pithActual-90f) < 5 )
    ((ImageView)termometroLateral).setImageResource(R.dr
    awable.termometro_03_verde);
    else
    if ( Math.abs(pithActual-90f) < 15 )

```

```

((ImageView)termometroLateral).setImageResource(R.dr
awable.termometro_02_naranja);
    else
    {

    ((ImageView)termometroLateral).setImageResource(R.dr
awable.termometro_01_rojo);
        // monitorTiempoReal.beepPith();
    }

    // Establece el color del tacometro del peso
    if ( pesoActual <= (0.05 *
    BastonConstant.pesoPaciente) )
    {

    ((ImageView)tacometro).setImageResource(R.drawable.t
achometer_rojo_bajo);
        monitorTiempoReal.beepPith();
    }
    else
    if ( pesoActual >= (0.15 *
    BastonConstant.pesoPaciente) )
    {

    ((ImageView)tacometro).setImageResource(R.drawable.t
achometer_rojo_alto);
        monitorTiempoReal.beepRoll();
    }
    else
    if ( pesoActual >= (0.12 *
    BastonConstant.pesoPaciente))
    {

    ((ImageView)tacometro).setImageResource(R.drawable.t
achometer_ambar_alto);
    }
    else
    if ( pesoActual <= (0.08 *
    BastonConstant.pesoPaciente))

```

```

{

    ((ImageView)tacometro).setImageResource(R.drawable.t
achometer_ambar_bajo);
    }
    else
    {

    ((ImageView)tacometro).setImageResource(R.drawable.t
achometer_verde);
    }

    // Establece el color del termometro frontal
    if ( Math.abs(rollActual) < 5 )
    ((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_03_verde);
    else
    if ( Math.abs(rollActual) < 10 )
    ((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_02_naranja);
    else
    {

    ((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_01_rojo);
        // monitorTiempoReal.beepRoll();
    }

    float left = termometroFrontal.getLeft();
    float top = termometroFrontal.getTop();
    float right = termometroLateral.getRight();
    float bottom = separador.getTop()-100;

    float ancho = right-left;
    float altura = bottom-top;

    float coordFrontalX = 203;
    float coordFrontalY = 377;

```

```

float coordLateralX = 485;
float coordLateralY = 377;

float anchoImg = bmpPersonasOrig.getWidth();
float alturaImg = bmpPersonasOrig.getHeight();
float escala;
float alturaLineaAngulo;
float posYLinea;

if ( anchoImg > alturaImg )
{
    escala = ancho/anchoImg;
    alturaLineaAngulo = (alturaImg*escala*0.6f);
    personasFinal =
Bitmap.createScaledBitmap(bmpPersonasOrig,(int)ancho
,(int)(alturaImg*escala),true);
}
else
{
    escala = altura/alturaImg;
    alturaLineaAngulo = altura*0.6f;
    personasFinal =
Bitmap.createScaledBitmap(bmpPersonasOrig,(int)(anch

```

```

oImg*escala),(int)altura,true);
}

coordFrontalX*= escala;
coordFrontalY*= escala;
coordLateralX*= escala;
coordLateralY*= escala;

float vistaFrontalX = left + coordFrontalX;
float vistaLateralX = left + coordLateralX;
float coordenadaPisoY = top + coordFrontalY;
float coordenadaY = coordenadaPisoY - 0.40f *
alturaImg;

Log.e("Visor", "" + coordenadaY);

posYLinea= coordenadaY - alturaLineaAngulo;

float deltaFrontal = coordenadaPisoY *
((float)Math.tan((rollActual)/180.0f*Math.PI));
// if ( rollActual < 0.0f ) deltaFrontal*=-1.0f;

float deltaLateral = alturaLineaAngulo *

```

```

((float)Math.tan((pitchActual-90f) /180.0f*Math.PI));

canvas.drawBitmap(personasFinal,left,top,currentPaint);
    canvas.drawCircle(vistaFrontalX, coordenadaY,20,
currentPaint);
    canvas.drawCircle(vistaLateralX, coordenadaY,20,
currentPaint);

// Dibuja la linea del ángulo
    canvas.drawLine(vistaFrontalX, coordenadaY,
vistaFrontalX + deltaFrontal, coordenadaPisoY,
currentPaintAngulo );

    canvas.drawLine(vistaLateralX, coordenadaY,
vistaLateralX - deltaLateral, coordenadaPisoY,
currentPaintAngulo);

    monitorTiempoReal.setEsperandoGui(false);
}
}

```

### 8.2.9. Archivo com.jbarriga.bastonint.gui.FrmHistorialDatos

```
package com.jbarriga.bastonint.gui;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.content.FileProvider;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.jbarriga.bastonint.BuildConfig;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.bo.BluetoothMng;
import com.jbarriga.bastonint.dao.BastonBD;
import com.jbarriga.bastonint.domain.DatosPasoVO;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.StringTokenizer;
import makval.gimak.gui.GActivity;
import makval.gimak.utils.DateUtils;

/* Muestra los datos historicos del baston */
public class FrmHistorialDatos extends GActivity {

    /* Nombre del parametro BOOL que indica si se deben
    o no usar datos simulados */
    public static final String
```

```
PARAM_USAR_DATOS_SIMULADOS =
"param_usarDatosSimulados";

    /* Indica si se debe o no usar datos simulados */
    private boolean usarDatosSimulados;

    /* Administrador de conectividad con bluetooth */
    private BluetoothMng bluetoothMng;

    /* Constante que representa el formato de las fechas
    */
    private static final String formatoFecha =
"dd/MM/yyyy";

    /* Procesador de formatos de fecha */
    private SimpleDateFormat sdf = new
SimpleDateFormat(formatoFecha);

    /* Lista de pasos leidos */
    private ArrayList<DatosPasoVO> lstPasos = new
ArrayList<>();

    /* Fecha minima de la lista de pasos */
    private Date fechaMinima;

    /* Fecha maxima de la lista de pasos */
    private Date fechaMaxima;

    /* Contiene los datos del historial, separados por coma
    */
    private String csvContent;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_frm_historial_datos);

        Toolbar toolbar = (Toolbar)
findViewById(R.id.toolbar);
        toolbar.setTitle(R.string.app_name);
        toolbar.setSubtitle(R.string.tituloDatosHistoricos);
        setGoBackImage(R.id.imgRetroceder);

        bluetoothMng = MainActivity.bluetoothMng;
        initgui();
        Intent intent = getIntent();

        usarDatosSimulados =
intent.getBooleanExtra(PARAM_USAR_DATOS_SIMULAD
OS, false);

        fechaMinima = null;
        fechaMaxima = null;

        showProgressDialog(getString(R.string.descargandoData
s));

        if ( usarDatosSimulados )
        {
            runOnUiThread( () -> leeDatosSimulados() );
        }
        else
        {
            runOnUiThread( () -> descargarDatos() );
        }
    }

    /* Inicializa los componentes visuales */
    private void initgui()
    {
        Button button;
        ImageView imageView;
```

```

        setDatePicker((TextView)
findViewById(R.id.lblValorFechaInicial),(ImageView)
findViewById(R.id.imgCalendarioFechaInicial),
formatoFecha, (v) -> calFechaInicial_onClick() );
        setDatePicker((TextView)
findViewById(R.id.lblValorFechaFinal), (ImageView)
findViewById(R.id.imgCalendarioFechaFinal),
formatoFecha, (v) -> calFechaFinal_onClick());

        button = (Button)
findViewById(R.id.btnDatosHoyDia);
        button.setOnClickListener((v) ->
btnDatosHoyDia_onClick() );

        button = (Button)
findViewById(R.id.btnDatosEstaSemana);
        button.setOnClickListener((v) ->
btnDatosEstaSemana_onClick() );

        button = (Button)
findViewById(R.id.btnDatosTodos);
        button.setOnClickListener((v) ->
btnDatosTodos_onClick() );

        imageView = (ImageView)
findViewById(R.id.imgCompartir);
        imageView.setOnClickListener( (v) ->
imgCompartir_onClick());

    }

    /**
     * Procesa el evento click sobre el boton de compartir
     */
    private void imgCompartir_onClick()
    {
        FileOutputStream fos = null;
        File csvFile = null;

```

```

        BastonBD bastonBD = new BastonBD(this);

        try
        {
            // Crear el archivo temporal con los datos en
            formato CSV
            csvFile = File.createTempFile("baston-" +
bastonBD.getParametro(BastonBD.PARAM_BD_HISTORI
A_CLINICA) + "-", ".txt", getExternalFilesDir(null));
            fos = new FileOutputStream(csvFile);
            fos.write(csvContent.getBytes());
            fos.flush();
            fos.close();

            // Ejecuta la opcion para compartir el archivo
            Uri fileUri = FileProvider.getUriForFile(this,
BuildConfig.APPLICATION_ID + ".provider", csvFile);
            Intent shareIntent = new Intent();
            shareIntent.setAction(Intent.ACTION_SEND);
            shareIntent.setType("text/plain");
            shareIntent.putExtra(Intent.EXTRA_STREAM,
fileUri);

            shareIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
            startActivity(Intent.createChooser(shareIntent,
"Compartir DATOS"));
        }
        catch (IOException e)
        {
            e.printStackTrace();

            showAlert(getString(R.string.tituloErrorInesperado), "NO
se pudo compartir los datos: " + e);
        }
    }

    /**
     * Procesa el evento click para visualizar los datos del

```

```

    dia de hoy
    */
    private void btnDatosHoyDia_onClick()
    {
        TextView textView;
        Date fecha = new Date();

        textView = (TextView)
findViewById(R.id.lblValorFechaInicial);
        textView.setText(sdf.format(fecha));

        textView = (TextView)
findViewById(R.id.lblValorFechaFinal);
        textView.setText(sdf.format(fecha));

        filtrarDatos();
    }

    /**
     * Procesa el evento click para visualizar los datos de la
    semana actual
    */
    private void btnDatosEstaSemana_onClick()
    {
        TextView textView;
        Date fechaMin = new Date();
        Date fechaMax;

        int diaSem = DateUtils.getWeekDay(fechaMin)-2;

        DateUtils.addDateValue( fechaMin,0,0,-diaSem);
        fechaMax = new Date(fechaMin.getTime());
        DateUtils.addDateValue(fechaMax,0,0,6);

        textView = (TextView)
findViewById(R.id.lblValorFechaInicial);
        textView.setText(sdf.format(fechaMin));

        textView = (TextView)

```



```

showToast(R.string.msgerr_fechaMaximaMenor);
    }
    filtrarDatos();
}
catch( Exception e )
{
    // Silenciado porque no hay posibilidad de error
}
}

/**
 * Filtra los datos y solicita al visor que solo visualise los
 * datos seleccionados
 */
private void filtrarDatos()
{
    TextView textView;
    ArrayList<DatosPasoVO> lstPasosFiltrados = new
ArrayList<>();
    Date fechaInicial, fechaFinal, fechaPaso;

    try
    {
        textView = (TextView)
findViewById(R.id.lblValorFechaInicial);
        fechaInicial =
sdf.parse(textView.getText().toString());

        textView = (TextView)
findViewById(R.id.lblValorFechaFinal);
        fechaFinal =
sdf.parse(textView.getText().toString());
        DateUtils.addDateValue(fechaFinal,0,0,1);

        for( DatosPasoVO pasoVO: lstPasos )
        {
            fechaPaso = pasoVO.getFecha();
            if ( fechaPaso.after(fechaFinal) )

```

```

        {
            break;
        }

        if ( fechaPaso.compareTo(fechaInicial) >= 0 )
        {
            lstPasosFiltrados.add(pasoVO);
        }
    }

    if ( lstPasosFiltrados.size() == 0 )
    {
        showAlert(getString(R.string.error_data),getString(R.string.
msgErr_noHayDatos));
        return;
    }

    HistorialImageView panel =
(HistorialImageView)findViewById(R.id.pnlHistorial);
    panel.setLstDatos(lstPasosFiltrados);
}
catch ( Exception e )
{
    // Silenciado porque no hay posibilidad de error
}
}

/**
 * Retorna datos simulados
 * @return
 */
public void leeDatosSimulados( )
{
    int resourceId = R.raw.datos_pesos;

    InputStream inputStream =
getResources().openRawResource(resourceId);
    InputStreamReader inputStreamReader = new

```

```

InputStreamReader(inputStream);
    BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);
    StringBuilder stringBuilder = new StringBuilder();
    String linea;

    try
    {
        while ((linea = bufferedReader.readLine()) != null)
        {
            stringBuilder.append(linea);
            stringBuilder.append("\n");
        }
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    finally
    {
        try
        {
            bufferedReader.close();
            inputStreamReader.close();
            inputStream.close();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }

    extraeDatos(stringBuilder.toString());
    hideProgressDialog();
    showToast("Se leyeron " + lstPasos.size() + "
eventos");
}

/* Descargar los datos desde el bluetooth */

```

```

private void descargarDatos()
{
    String msgError = "No se pudo solicitar el envio de
datos";
    int rptaNumerica;
    String txt;

    try
    {
        // Solicita la lista de todos los datos
        bluetoothMng.limpiarBuffer();
        bluetoothMng.enviaMensajeBlueTooth('4');

        // lee respuesta
        rptaNumerica =
        bluetoothMng.leeByteBluetooth();
        if ( rptaNumerica == 'k' )
        {
            // Si tiene datos
            txt = bluetoothMng.readString("\n");
            Log.e("FrmHistorial","Descripcio:" + txt);
            txt = bluetoothMng.readString("#");
            Log.e("FrmHistorial","Datos completos:"+txt);

            extraeDatos(txt);
            showToast("Se leyeron " + lstPasos.size() + "
eventos");
        }

        hideProgressDialog();
    }
    catch ( Exception e )
    {
        hideProgressDialog();

        showAlert(getString(R.string.tituloErrorInesperado),getSt
ring(R.string.error_descargar_datos));
    }
}

```

```

/* Llena el arreglo de pasos */
private void extraeDatos( String datos )
{
    String fila;
    String fecha;
    String txt;
    String msgError;
    SimpleDateFormat sdf = new
SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    StringTokenizer tokenLinea = new
StringTokenizer(datos, "\n");
    StringTokenizer tokenDatos;
    DatosPasoVO pasoVO;

    lstPasos.clear();
    fecha = null;

    csvContent = "Fecha,Hora,Pitch,Roll,Peso (Kg)\r\n" +
datos;

    fila = tokenLinea.nextToken();
    while( tokenLinea.hasMoreElements() )
    {
        fila = tokenLinea.nextToken().trim();
        tokenDatos = new StringTokenizer(fila, ",");
        // if ( tokenDatos.countTokens() != 6 )
        if ( tokenDatos.countTokens() != 5 )
        {
            // Se espera : fecha,hora,pitch,roll,num_pasos
            Log.e("FrmHistorialDatos","No se recibieron 5
valores separados por coma: " + fila);
            continue;
        }

        pasoVO = new DatosPasoVO();
        msgError = "No se pudo leer la fecha de un paso";

        try

```

```

{
    fecha = tokenDatos.nextToken() + " " +
tokenDatos.nextToken();
    pasoVO.setFecha(sdf.parse(fecha));

    msgError = "No se pudo leer el pitch";
    txt = tokenDatos.nextToken();
    pasoVO.setPitch(Float.parseFloat(txt));

    msgError = "No se pudo leer el roll";
    txt = tokenDatos.nextToken();
    pasoVO.setRoll(Float.parseFloat(txt));

    msgError = "No se pudo leer el peso";
    txt = tokenDatos.nextToken();
    pasoVO.setPeso(Float.parseFloat(txt));

    lstPasos.add(pasoVO);
}
catch ( Exception e )
{
    // Se ignora la fila pues la fecha no tiene el
formato adecuado
    Log.e("FrmHistorialDatos",msgError + " : " +
fila);
    continue;
}

Collections.sort(lstPasos, (p1,p2) ->
p1.getFecha().compareTo(p2.getFecha() ) );
calcularFechasExt();

HistorialImageView panel =
(HistorialImageView)findViewById(R.id.pnlHistorial);
panel.setLstDatos(lstPasos);
}
}

```

## 8.2.10. Archivo com.jbarriga.bastonint.gui.MonitorImageView

```
package com.jbarriga.bastonint.gui;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import com.jbarriga.bastonint.BastonConstant;
import com.jbarriga.bastonint.R;
import java.io.InputStream;

/**
 * Representa el visor de barras que acepta eventos
 * tactiles para mostrar la informacion historica
 */
public class MonitorImageView extends
androidx.appcompat.widget.AppCompatImageView
{
    /**
     * Configuracion del pintado
     */
    private Paint currentPaint;

    /**
     * Configuracion del pintado de linea de angulo
     */
    private Paint currentPaintAngulo;

    /**
     * Componente monitor que contiene al visor
     */
    private FrmMonitorTiempoReal monitorTiempoReal;
```

```
/**
 * Imagen original
 */
Bitmap bmpPersonasOrig;

/**
 * Angulo de la vista frontal
 */
float pithActual = 4;

/**
 * Angulo de la vista lateral
 */
float rollActual = 45;

/**
 * Peso actual
 */
float pesoActual = 0;

/**
 * Constructor
 * @param context
 * @param attrs
 */
public MonitorImageView(Context context,
AttributeSet attrs) {
    super(context, attrs);

    currentPaint = new Paint();
    currentPaint.setDither(true);
    currentPaint.setColor(0xFF00CC00); // alpha.r.g.b
    currentPaint.setStyle(Paint.Style.STROKE);
    currentPaint.setStrokeJoin(Paint.Join.ROUND);
```

```
currentPaint.setStrokeCap(Paint.Cap.ROUND);
currentPaint.setStrokeWidth(4);

currentPaintAngulo = new Paint();
currentPaintAngulo.setDither(true);
currentPaintAngulo.setColor(0xFFFF0000); //
alpha.r.g.b
currentPaintAngulo.setStyle(Paint.Style.STROKE);

currentPaintAngulo.setStrokeJoin(Paint.Join.ROUND);

currentPaintAngulo.setStrokeCap(Paint.Cap.ROUND);
currentPaintAngulo.setStrokeWidth(4);

InputStream is =
getResources().openRawResource(+R.drawable.persona_
vista_guia);
    bmpPersonasOrig =
BitmapFactory.decodeStream(is);
}

/**
 * Establece el monitor en el que se visualiza el
 * componente
 * @param monitorTiempoReal
 */
public void setMonitorTiempoReal(
FrmMonitorTiempoReal monitorTiempoReal )
{
    this.monitorTiempoReal = monitorTiempoReal;
}

/**
 * Establece la data que se debe visualizar.
 * @param pithParam
 * @param rollParam
 * @param peso
```

```

*/
public synchronized void setData( double pithParam,
double rollParam, double peso )
{
    float pith,roll;

    Log.d("MonitorImageView","Pitch Orig:" +
pithParam + " ROLL Orig:" + rollParam + " Peso: " + peso);

    pith = (float)pithParam;
    roll = (float)rollParam;
    pesoActual = (float) peso;

    if (( pith != pithActual) || ( roll != rollActual))
    {
        pithActual = pith;
        rollActual = roll;
    }
}

/**
 * Actualiza la vista en tiempo real
 * @param canvas
 */
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    TextView txtAngulo;

    View termometroLateral =
getRootView().findViewById(R.id.imgTermometroLateral
);
    View termometroFrontal =
getRootView().findViewById(R.id.imgTermometroFrontal
);
    View tacometro =
getRootView().findViewById(R.id.imgTacometro);
    View separador =
getRootView().findViewById(R.id.separador);

```

```

    txtAngulo =
getRootView().findViewById(R.id.lblAnguloFrontal);
txtAngulo.setText(""+ rollActual);

    txtAngulo =
getRootView().findViewById(R.id.lblAnguloLateral);
txtAngulo.setText(""+ pithActual+" ");

    String dato = "" + pesoActual;
    txtAngulo =
getRootView().findViewById(R.id.lblPeso);
txtAngulo.setText(" "+dato + " Kg");

    Bitmap personasFinal;

    // Establec el color del termometro lateral
    if ( Math.abs(pithActual-90f) < 5 )
((ImageView)termometroLateral).setImageResource(R.dr
awable.termometro_03_verde);
    else
    if ( Math.abs(pithActual-90f) < 15 )
((ImageView)termometroLateral).setImageResource(R.dr
awable.termometro_02_naranja);
    else
    {
((ImageView)termometroLateral).setImageResource(R.dr
awable.termometro_01_rojo);
        // monitorTiempoReal.beepPith();
    }

    // Establece el color del tacometro del peso
    if ( pesoActual <= (0.05 *
BastonConstant.pesoPaciente) )
    {
((ImageView)tacometro).setImageResource(R.drawable.t
achometer_rojo_bajo);

```

```

        monitorTiempoReal.beepPith();
    }
    else
    if ( pesoActual >= (0.15 *
BastonConstant.pesoPaciente) )
    {
((ImageView)tacometro).setImageResource(R.drawable.t
achometer_rojo_alto);
        monitorTiempoReal.beepRoll();
    }
    else
    if ( pesoActual >= (0.12 *
BastonConstant.pesoPaciente) )
    {
((ImageView)tacometro).setImageResource(R.drawable.t
achometer_ambar_alto);
    }
    else
    if ( pesoActual <= (0.08 *
BastonConstant.pesoPaciente) )
    {
((ImageView)tacometro).setImageResource(R.drawable.t
achometer_ambar_bajo);
    }
    else
    {
((ImageView)tacometro).setImageResource(R.drawable.t
achometer_verde);
    }

    // Establec el color del termometro frontal
    if ( Math.abs(rollActual) < 5 )
((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_03_verde);
    else

```

```

    if ( Math.abs(rollActual) < 10 )
    ((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_02_naranja);
    else
    {

    ((ImageView)termometroFrontal).setImageResource(R.d
rawable.termometro_01_rojo);
        // monitorTiempoReal.beepRoll();
    }

    float left = termometroFrontal.getLeft();
    float top = termometroFrontal.getTop();
    float right = termometroLateral.getRight();
    float bottom = separador.getTop()-100;

    float ancho = right-left;
    float altura = bottom-top;

    float coordFrontalX = 203;
    float coordFrontalY = 377;

    float coordLateralX = 485;
    float coordLateralY = 377;

    float anchoImg = bmpPersonasOrig.getWidth();
    float alturaImg = bmpPersonasOrig.getHeight();
    float escala;
    float alturaLineaAngulo;
    float posYLinea;

```

```

    if ( anchoImg > alturaImg )
    {
        escala = ancho/anchoImg;
        alturaLineaAngulo = (alturaImg*escala*0.6f);
        personasFinal =
        Bitmap.createScaledBitmap(bmpPersonasOrig,(int)ancho
,(int)(alturaImg*escala),true);
    }
    else
    {
        escala = altura/alturaImg;
        alturaLineaAngulo = altura*0.6f;
        personasFinal =
        Bitmap.createScaledBitmap(bmpPersonasOrig,(int)(anch
oImg*escala),(int)altura,true);
    }

    coordFrontalX*= escala;
    coordFrontalY*= escala;
    coordLateralX*= escala;
    coordLateralY*= escala;

    float vistaFrontalX = left + coordFrontalX;
    float vistaLateralX = left + coordLateralX;
    float coordenadaPisoY = top + coordFrontalY;
    float coordenadaY = coordenadaPisoY - 0.40f *
alturaImg;

    Log.e("Visor", "" + coordenadaY);

```

```

    posYLinea= coordenadaY - alturaLineaAngulo;

    float deltaFrontal = coordenadaPisoY *
((float)Math.tan((rollActual)/180.0f*Math.PI));
    // if ( rollActual < 0.0f ) deltaFrontal*= -1.0f;

    float deltaLateral = alturaLineaAngulo *
((float)Math.tan((pitchActual-90f) /180.0f*Math.PI));

    canvas.drawBitmap(personasFinal,left,top,currentPaint);
    canvas.drawCircle(vistaFrontalX, coordenadaY,20,
currentPaint);
    canvas.drawCircle(vistaLateralX, coordenadaY,20,
currentPaint);

    // Dibuja la linea del ángulo
    canvas.drawLine(vistaFrontalX, coordenadaY,
vistaFrontalX + deltaFrontal, coordenadaPisoY,
currentPaintAngulo );

    canvas.drawLine(vistaLateralX, coordenadaY,
vistaLateralX - deltaLateral, coordenadaPisoY,
currentPaintAngulo);

    monitorTiempoReal.setEsperandoGui(false);
}
}

```

### 8.2.11. Archivo com.jbarriga.bastonint.gui.FrmNuevoPaciente

```

package com.jbarriga.bastonint.gui;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.jbarriga.bastonint.R;
import com.jbarriga.bastonint.dao.BastonBD;
import java.io.IOException;
import java.util.HashMap;
import makval.gimak.gui.EditTextUtils;
import makval.gimak.gui.GActivity;

/* Representa la pantalla donde se registran los datos de
un paciente */
public class FrmNuevoPaciente extends GActivity {

    /* Lista de parametros leidos */
    private HashMap<String, String> lstParmas = new
HashMap<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_frm_nuevo_paciente);

        Toolbar toolbar = (Toolbar)
findViewById(R.id.toolbar);
        toolbar.setTitle(R.string.app_name);
        toolbar.setSubtitle(R.string.tituloNuevoPaciente);
        setGoBackImage(R.id.imgRetroceder);

        initGui();
    }

```

```

/**
 * Inicializa los datos
 */
private void initGui()
{
    Button boton = (Button)
findViewById(R.id.btnRegistrar);
    boton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            btnRegistrar_onClick();
        }
    });
}

/**
 * Procesa el registro
 */
private void btnRegistrar_onClick()
{
    if ( validaDatos() == false )
    {
        return;
    }
    EditText editText =
(EditText)findViewById(R.id.txtHistoriaClinica);

    showProgressDialog(getString(R.string.esperar_mientras
_guarda_usuario));
    runOnUiThread( () ->
guardaDatos(editText.getText().toString()) );
}

/**

```

```

 * Guarda los datos
 * @param historia
 */
private void guardaDatos( String historia )
{
    int rptA;
    try
    {
        MainActivity.bluetoothMng.enviaMensajeBlueTooth("1")
;

        MainActivity.bluetoothMng.enviaMensajeBlueTooth(hist
oria);

        rptA =
MainActivity.bluetoothMng.leeByteBluetooth();
        if ( rptA == 'c' )
        {
            guardaDatosBd();
            hideProgressDialog();

            showAlert(getString(R.string.tituloOperacionExitos
a), getString(R.string.msgPacienteRegistrado),
getString(R.string.aceptar),
                (v1,v2) ->
startFirstActivity(FrmMenuPrincipal.class));
        }
        else
        if ( rptA == 'e' )
        {
            guardaDatosBd();
            hideProgressDialog();

            showAlert(getString(R.string.tituloOperacionExitos
a), getString(R.string.msgPacienteYaExiste),
getString(R.string.aceptar),

```

```

        (v1,v2) ->
startFirstActivity(FrmMenuPrincipal.class));
    }
    else
    {
        hideProgressDialog();

showAlert(getString(R.string.tituloOperacionFallida),
getString(R.string.msgPacienteNoRegistrado));
    }
}
catch ( Exception e )
{
    hideProgressDialog();

showAlert(getString(R.string.tituloErrorInesperado),getSt
ring(R.string.msgErr_ErrorCrearNuevoPaciente));
}
}

/**
 * Guarda los datos
 */
private void guardaDatosBd() throws IOException
{
    BastonBD bastonBD = new BastonBD(this);

bastonBD.guardarDatosPaciente(lstParmas.get(BastonB
D.PARAM_BD_NOMBRE),
lstParmas.get(BastonBD.PARAM_BD_HISTORIA_CLINICA)
,
Integer.parseInt(lstParmas.get(BastonBD.PARAM_BD_PE
SO)),
Integer.parseInt(lstParmas.get(BastonBD.PARAM_BD_ED
AD)) );
}

```

```

/**
 * Valida los datos del formulario
 * @return true en caso que todo este bien, false en
 caso de error
 */
private boolean validaDatos() {
    EditText editText;
    String txt;
    String msgError = "";

    editText = (EditText) findViewById(R.id.txtNombre);
    txt= editText.getText().toString();
    if ((txt == null) || (txt.length() == 0))
    {
        msgError+= "," +
getString(R.string.msgErr_ElNombreEsRequerido);
    }
    else
    {
        lstParmas.put(BastonBD.PARAM_BD_NOMBRE,txt);
    }

    editText = (EditText)
findViewById(R.id.txtHistoriaClinica);
    txt= editText.getText().toString();
    if ((txt == null) || (txt.length() == 0))
    {
        msgError+= "," +
getString(R.string.msgErr_ElHistorialEsRequerido);
    }
    else
    {
        lstParmas.put(BastonBD.PARAM_BD_HISTORIA_CLINICA,
txt);
    }
}

```

```

editText = (EditText) findViewById(R.id.txtPeso);
txt= editText.getText().toString();
if ((txt == null) || (txt.length() == 0))
{
    msgError+= "," +
getString(R.string.msgErr_ElPesoEsRequerido);
}
else
{
    lstParmas.put(BastonBD.PARAM_BD_PESO,txt);
}

editText = (EditText) findViewById(R.id.txtEdad);
txt= editText.getText().toString();
if ((txt == null) || (txt.length() == 0))
{
    msgError+= "," +
getString(R.string.msgErr_ElEdadEsRequerido);
}
else
{
    lstParmas.put(BastonBD.PARAM_BD_EDAD,txt);
}

if ( msgError.length() > 0 )
{
showAlert(getString(R.string.tituloErrorValidacion),msgEr
ror.substring(1));
    return false;
}

    return true;
}
}

```

### 8.2.12. Archivo com.jbarriga.bastonint.BastonConstant.java

```
package com.jbarriga.bastonint;

/**
 * Constantes empleadas por la aplicacion
 * Esta es la version 1.1
 */
public class BastonConstant
{
    /**
     * Indica si se debe acceder al mueble via BLUETOOTH.
     * El valor para produccion de esta variable deberá ser true, pero para desarrollo
     * sin el mueble debera ser false
     */
    public static final boolean ACCEDER_BASTON = true;

    /**
     * Respuesta que indica comando exitoso
     */
    public static final String RPTA_CMD_OK = "OK";

    /**
     * Respuesta que indica error
     */
    public static final String RPTA_CMD_ERR = "ER";

    /**
     * Peso del paciente
     */
    public static float pesoPaciente = 40;
}
```