

UNIVERSIDAD DE SANTIAGO DE
COMPOSTELA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

**Desarrollo de un prototipo para un
wearable o smartphone que permita
estudiar la fragilidad física en las
personas mayores aplicando técnicas de
inteligencia artificial**

Autor/a:

Alex Fuentes Gómez

Tutores:

**Sonia Maria Valladares Rodriguez
Jesus Maria Rodriguez Presedo**

Grado en Ingeniería Informática

Julio 2024

Trabajo de Fin de Grado presentado en la Escuela Técnica Superior de
Ingeniería de la Universidad de Santiago de Compostela para la obtención del
Grado en Ingeniería Informática

Agradecimientos

Agradezco en primer lugar a mis padres, ya que sin ellos no habría sido posible nada de lo conseguido en estos últimos años.

Agradezco también a mis hermanos por hacerme reír en los momentos más estresantes a lo largo del curso y por ayudarme a afrontar mis problemas desde otra perspectiva.

Agradezco a Alba por escucharme y entenderme en mis momentos más difíciles, y finalmente a toda la gente que quiero y que ha hecho de estos años una de las experiencias más bonitas de mi vida.

Resumen

En este trabajo de fin de grado se investiga la viabilidad del desarrollo de un prototipo para un dispositivo wearable o smartphone que permita evaluar la fragilidad física en personas mayores. El prototipo calculará medidas como la velocidad de la marcha y otros factores relacionados con la fragilidad, y será capaz de realizar predicciones sobre el estado de fragilidad mediante técnicas de inteligencia artificial, ofreciendo una herramienta innovadora para la identificación temprana y el seguimiento de la fragilidad en la población mayor.

Índice general

1. Introducción	1
2. Estado del conocimiento	5
3. Evaluación Preliminar de Dispositivos y APIs	9
3.1. Experimentación Inicial	9
3.1.1. Selección y Pruebas Preliminares de Dispositivos	9
3.1.2. Prueba de desarrollo en Fitbit Alta HR	10
3.1.3. Exploración de Alternativas Tecnológicas	13
3.2. Decisión Final y Cancelación de Pedidos	13
4. Metodología	15
4.1. Selección de Dispositivos y Plataformas	16
4.1.1. Criterios de selección	16
4.1.2. Proceso de Evaluación	16
4.2. Materiales Utilizados	17
4.2.1. Descripción del dispositivo seleccionado: Nokia 6.1	17
4.3. Procedimientos Experimentales	18
4.3.1. Investigación Inicial	18
4.3.2. Funcionamiento del Acelerómetro	19
4.3.3. Orientación del teléfono	19
4.3.4. Eliminación del ruido del acelerómetro	21
4.3.5. Estimación de la velocidad	26
4.4. Desarrollo del Prototipo	31
4.4.1. Obtención de datos en Kotlin	31
4.4.2. Implementación de Scripts Matlab en el Prototipo	32
4.4.3. Resultado del Prototipo	32
4.5. Métodos de análisis	33
4.5.1. Procedimiento Inicial	33
4.5.2. Análisis del DataSet	34
4.5.3. Análisis correlación	35
4.5.4. Construcción de Modelo y predicción	37
4.5.5. Potencial de la Implementación de Algoritmos de Predicción	37

4.6.	Protocolo Experimental	38
4.6.1.	Procedimiento general	38
4.6.2.	Protocolo de uso de la aplicación	39
5.	Pruebas	41
5.1.	Objetivo	41
5.2.	Plan de Pruebas	41
5.2.1.	Prueba de Funcionalidad de la Aplicación	41
5.2.2.	Prueba de Viabilidad de Predicciones Asíncronas	42
5.3.	Conclusiones de las Pruebas	43
6.	Discusión de los resultados	45
6.1.	Comparativa con el estado del arte	45
6.2.	Mejoras sobre Soluciones Anteriores	45
7.	Conclusiones y Posibles ampliaciones	47
7.0.1.	Principales aportaciones	47
7.0.2.	Limitaciones y problemas encontrados	48
7.0.3.	Lineas Futuras	49
A.	Anexo: Código en Kotlin	51
A.1.	Código en Python	51
A.1.1.	Código conexión Fitbit Simple	51
A.1.2.	Código conexión Fitbit PCKE	51
A.2.	Código en Kotlin	52
A.2.1.	Código Eliminación ruido	52
A.2.2.	Código Filtro paso bajo	52
A.2.3.	Código Invocación Filtro	52
A.2.4.	Código Magnitud Aceleración	53
B.	Anexo: Imágenes Adicionales	55
B.1.	Imágenes Adicionales Fitbit	56
B.1.1.	Creación Aplicación	56
B.1.2.	Datos para la creación de la URL de la app	57
B.1.3.	Formulario de Permisos	57
B.1.4.	Foro Fitbit Aclaración	58
	Bibliografía	59

Índice de figuras

2.1. Formas de onda de aceleración del smartphone y del acelerómetro triaxial.	7
4.1. Direcciones de los ejes cartesianos en el teléfono. [31]	19
4.2. Valores del acelerómetro en reposo.	19
4.3. Ubicaciones del teléfono para las mediciones; cuerpo (A), bolso (B), cinturón (C), mano (D) y bolsillo (E). El borde rectangular indica la orientación y ubicación del teléfono inteligente en cada condición [33]	21
4.4. Remuestreo valores acelerómetro.	24
4.5. Señal tras aplicar filtro de 20 Hz.	25
4.6. Señal tras aplicar filtro de 2 Hz.	25
4.7. Señal tras aplicar filtro de 2 Hz en detalle.	26
4.8. Prueba de Caminata.	28
4.9. Prueba de Caminata Filtro 20 Hz.	28
4.10. Prueba de Caminata Filtro 2 Hz.	29
4.11. Tabla de correlaciones [33].	30
4.12. Identificación de golpes de talón.	31
4.13. Interfaz de la aplicación antes y después de la prueba	33
4.14. Matriz de correlación de la predicción	35
4.15. Distribución de los datos.	36
B.1. Creación aplicación.	56
B.2. Datos para la creación de la URL de la aplicación.	57
B.3. Formulario de permisos de la aplicación.	57
B.4. Aclaración acerca del acceso a la información del acelerómetro en el dispositivo Alta HR	58

Índice de cuadros

4.1. Información Básica de los Sensores del Smartphone	18
4.2. Información Técnica de los Sensores del Smartphone	18
4.3. Resumen estadístico del dataset utilizado para el análisis.	35
5.1. Métricas de evaluación para los diferentes modelos.	43
5.2. Matrices de confusión para los diferentes modelos.	43

Capítulo 1

Introducción

En la actualidad, el uso generalizado de dispositivos wearables y sensores integrados en teléfonos móviles inteligentes ha generado un amplio campo de posibilidades en el ámbito de la salud y el bienestar. Estos dispositivos son cada vez más ubicuos en la vida cotidiana y tienen la capacidad de capturar una amplia gama de datos sobre diversas actividades y funciones fisiológicas. A medida que nuestro sistema de salud enfrenta el desafío de atender a una población cada vez más envejecida, con más del 17 % de la población de nuestro país superando los 64 años y proyecciones de aumento al 32 % para el 2050 [3], la importancia de implementar herramientas eficaces de cribado se hace cada vez más evidente. Entre estas herramientas, destaca la velocidad de la marcha, ya que se trata de un indicador de la fragilidad con un buen nivel de precisión y que además es muy sencilla de realizar [2].

La detección temprana de la fragilidad, además de poder aportar beneficios clínicos, tiene un notable impacto en la seguridad de los pacientes y en la sostenibilidad del sistema. Por ejemplo, el estudio de cohortes ESTHER, que siguió a 2595 adultos de entre 57 y 84 años durante 8 años, utilizó el fenotipo de Fried para definir el síndrome de fragilidad y encontró que los costes sanitarios eran considerablemente más altos para los sujetos frágiles comparados con los no frágiles. Los sujetos con 4 o 5 síntomas de fragilidad incurrieron en un coste medio de 3659 €. en tres meses, en comparación con 642 €. para los individuos no frágiles [17]. Este aumento en el coste está fuertemente asociado con ingresos hospitalarios y gastos farmacéuticos, independientemente de la edad y comorbilidades.

En los Estados Unidos, un análisis de la base de datos de Medicare de 2011 y 2012 mostró que los ancianos frágiles constituían el 46.2 % del 10 % de usuarios con los mayores gastos sanitarios, y este subgrupo presentó el mayor porcentaje de gastos potencialmente evitables con un 43.9 % [6]. Esto resalta la importancia de las intervenciones sanitarias destinadas a prevenir y revertir la fragilidad como medio para reducir estos costes. Sin embargo, en un sistema sanitario cada vez

más saturado, para que la implementación de estrategias de detección pueda ser viable y se implemente en la práctica clínica habitual, deben ser pruebas sencillas y fáciles de realizar por los profesionales sanitarios [2].

La prevalencia de la fragilidad está íntimamente ligada al envejecimiento de la población, aumentando su porcentaje a medida que aumenta la edad. En los sujetos de 65-70 años se encontró una prevalencia del 3-15 %, en los de 70-80 años del 5-17 % y el grupo de mayores de 85 o más años la prevalencia es muy elevada con valores del 26 % hasta el 50-56 % [4]. Actualmente, carecemos de una herramienta de cribado universal, siendo la velocidad de la marcha un factor común en varias herramientas de detección de la fragilidad. Sin embargo, la variabilidad en los parámetros de medición y la falta de protocolos estándar, complican una aplicación eficaz de las mismas [5].

Ya que el objetivo principal de la prueba de la velocidad de la marcha es un cribado y no un diagnóstico, se podría proponer como primera línea de detección (en aquellos sujetos en condiciones de realizarla), con diferentes puntos de corte para clasificar a los individuos en prefrágiles, frágiles o robustos, y posteriormente realizar una valoración más completa para su diagnóstico de clasificación [7]. Además, la detección del paciente con síndrome de fragilidad no puede quedar solo en la realización de un diagnóstico precoz, sino que debe llevar asociado la instauración de una estrategia de manejo integral de aquellas personas detectadas como frágiles.

Por otra parte, tal y como se menciona en [7], se debe promover la equidad en el acceso y garantizar que no exacerba desigualdades existentes. Así como asegurar que se respeta la autonomía y confidencialidad. Sería también necesario evaluar la calidad del proceso, asegurando que los resultados finales a medir sean accesibles y estén acordados de antemano. Para todo esto, sería necesario contar con un sistema de recogida de datos que permita su adecuada evaluación con respecto al impacto en la práctica clínica y sobre la salud de los pacientes detectados.

En este contexto, el desarrollo de la tecnología de los wearables y los smartphones, combinado con el uso de la inteligencia artificial, nos ofrece un gran potencial para superar estas barreras, proporcionando una posible forma sencilla y precisa de recopilar datos fisiológicos, que podría facilitar la implementación de estrategias de detección en la práctica clínica habitual y ayudar a aliviar la saturación del sistema sanitario. Las posibilidades que ofrecen estos dispositivos son amplias y se extienden en diversas áreas, incluyendo la monitorización de la salud hasta la detección de patrones de movimiento. Este avance tecnológico ha despertado un interés significativo en el ámbito de la detección de la fragilidad en personas mayores.

Considerando todo esto, el presente trabajo de investigación se enfoca en la exploración de dispositivos potencialmente beneficiosos para la detección temprana de problemas de fragilidad en personas mayores. Concretamente, se propone probar la viabilidad de uso de wearables o smartphones para medir la velocidad de la marcha mediante un acelerómetro. Donde la hipótesis a testar es que estos dispositivos puedan proporcionar una estimación precisa de la fragilidad en personas mayores, comparable a las evaluaciones clínicas tradicionales. Este enfoque podría ofrecer una solución mas accesible para una valoración rápida y precisa de la fragilidad.

Partiendo de esta premisa, el objetivo es desarrollar un prototipo de aplicación que permita la digitalización del test de valoración de fragilidad física, centrándose especialmente en la medición de la velocidad de la marcha, de acuerdo a los criterios establecidos por Fried. De esta forma, realizaremos un estudio exhaustivo sobre la posible aplicación de wearables o teléfonos móviles con sensores acelerómetros en la detección de fragilidad.

Este prototipo empleará también técnicas de inteligencia artificial para analizar los datos recopilados y realizar una valoración precisa del estado de fragilidad física del individuo. Para alcanzar este propósito, se establecerán algunos objetivos más específicos:

- Desarrollar un prototipo de aplicación específico para tratar de digitalizar el test de valoración de fragilidad física de Fried, enfocandose particularmente en la velocidad de la marcha como un indicador clave en la fragilidad en personas mayores.
- Integrar un dispositivo móvil o wearable que sea capaz de capturar los datos relevantes relacionados con la movilidad y que además aplique técnicas de inteligencia artificial para analizar estos datos y genere una evaluación precisa del estado de fragilidad física de cada persona.
- Evaluar la viabilidad técnica y la eficacia del prototipo desarrollado en la digitalización del test de Fried, en términos de precisión en la predicción de la fragilidad física. Tambien sería interesante comprobar la aceptación de la herramienta entre los profesionales del sector.

Los objetivos del trabajo están enfocados en la aplicación práctica de las tecnologías emergentes para enfrentarnos a desafíos específicos en el campo de la salud. La digitalización del test de Fried representa un paso importante hacia la detección temprana y la intervención preventiva en los casos de fragilidad física, con el potencial de mejorar de manera significativa la calidad de vida de esta población vulnerable.

Capítulo 2

Estado del conocimiento

El envejecimiento de la población es un fenómeno global que plantea desafíos significativos para los sistemas de atención médica y el bienestar de las personas mayores. Entre los aspectos más críticos asociados con el envejecimiento se encuentra la fragilidad, definida por Linda Fried como un síndrome multifactorial, caracterizado por la presencia de **3 de los siguientes 5 criterios**: fatiga crónica autorreferida, debilidad, inactividad, disminución de la velocidad de la marcha y pérdida de peso [1]. O de manera mas concreta como un estado de vulnerabilidad para padecer problemas de salud, como discapacidad, dependencia, caídas, necesidad de cuidados a largo plazo y mortalidad. [1]

Actualmente, existen numerosos métodos de medición de la fragilidad física en personas de la tercera edad. Tal y como se explica en [2], existen varias técnicas de medición de la fragilidad como pueden ser la Prueba Corta de Desempeño Físico o Short Physical Performance Battery (SPPB), la velocidad de la marcha (VM), el Test Get Up and Go (TUG) o la escala FRAIL. Los parametros de detección de estas pruebas se han ajustado a partir de evidencias, para mejorar la deteccion de la fragilidad.

- **SPPB (Short Physical Performance Battery)**: Consiste en realizar tres pruebas: en primer lugar una de equilibrio (en tres posiciones: pies juntos, semitándem y tándem), luego otra de velocidad de la marcha (sobre 2,4 ó 4 metros) y finalmente levantarse y sentarse en una silla cinco veces.
- **Velocidad de la Marcha (VM)**: Esta mide el tiempo que tarda el paciente en caminar una distancia de 4 metros. Una velocidad menor a 0.8 m/s se considera señal de fragilidad.
- **Test Get Up and Go (TUG)**: Evalúa la movilidad y el equilibrio del paciente. Consiste en levantarse de una silla, luego caminar una distancia corta, girar, y volver a la silla para sentarse de nuevo. Un tiempo superior a 12 segundos puede indicar una mayor probabilidad de fragilidad.

- **Escala FRAIL:** Esta escala está compuesta por 5 preguntas que abordan distintos dominios: Fatiga, Resistencia, Deambulación, Comorbilidad y Pérdida de Peso. Cada ítem se valora con 1 punto, y luego se suman los resultados para obtener el nivel de fragilidad. Si sumas de 1 a 2 puntos indica prefragilidad, y 3 o más, fragilidad.

Además, en general, se recomienda el uso de la prueba de SPPB debido a que es mucho mas completa que otras comentadas anteriormente. No obstante en estudios como [13] se indicó que la medición de la velocidad de la marcha puede ser un predictor de riesgo de efectos adversos en población mayor similar a otras herramientas más completas como el SPPB. De los tres componentes mencionados anteriormente sobre esta herramienta (equilibrio, TUG y velocidad de la marcha), diversos estudios encontraron que la velocidad de la marcha, como predictor único, presentaba resultados similares a los de la realización de esta batería completa. Asimismo, si la VM es mas sencilla y rápida de realizar, podría establecerse como opción que reemplace el SPPB facilitando la implementación de una prueba de detección para la fragilidad en la práctica clínica rutinaria [7].

Como vemos, existen numerosas técnicas de medición de la fragilidad, sin embargo, todas ellas dependen de un especialista que las realice manualmente y en consulta, apuntando los tiempos de duración por cada prueba. Una vez registrados los resultados, se asigna la puntuación correspondiente para obtener el grado de fragilidad. A partir de esto surgen varias aplicaciones especializadas en la medición de este fenómeno. Por ejemplo, de nuevo en [7] se menciona el uso de programas informáticos como el GAITRite, que se trata de un software exclusivo que obtiene multiples parámetros acerca de la marcha del paciente. Este software registra los datos capturando la geometría y disposición de cada pisada en función del tiempo y el espacio dentro de una pasarela [14]. Este, sería un software ideal para la medicion de la marcha, no obstante, como ocurre con muchas tecnologías avanzadas, la principal desventaja de este sistema sería el precio, que esta entre los 10.000\$ y los 15.000\$ [15].

Por otro lado, tambien podemos encontrar aplicaciones para smartphone, como la denominada **PowerFrail**, que tal y como se comenta en [16] evalua la potencia muscular y fragilidad en personas mayores mediante dos pruebas principales:

- **Test de potencia STS (Sit-To-Stand):** Que se trata de una prueba muy similar al TUG, sin embargo, en este caso el objetivo de la medición sería la potencia muscular a diferencia del TUG que mide el riesgo de caidas. La aplicación graba este movimiento y utiliza un algoritmo para determinar la potencia muscular del usuario.

- **Test de fragilidad FTS-5:** Evalúa cinco áreas críticas que incluyen índice de masa corporal, equilibrio, fuerza de agarre, velocidad de la marcha y actividad física. Basado en estos resultados, la aplicación nos devuelve el grado de fragilidad del paciente y otras características.

La aplicación también permite guardar y seguir la información de cada paciente para futuras consultas y seguimientos. No obstante, como vemos, esta aplicación utiliza la cámara del smartphone para las mediciones a diferencia de la idea de nuestro prototipo que accedería al sensor acelerómetro de nuestro smartphone recogiendo las medidas necesarias para así poder evaluar la fragilidad de manera mas precisa.

En el estudio de Grouios et al. [27] evaluaron la precisión de los acelerómetros de tres smartphones modernos, comparándolos con el sistema de captura de movimiento Vicon MX, el cual se trata de un estándar de oro en los sistemas de captura de movimiento de mercado. Los resultados demostraron que los acelerómetros de los smartphones son dispositivos válidos y confiables para estimar aceleraciones, sin diferencias significativas en comparación con el sistema Vicon MX. Esto sugiere de manera clara que los acelerómetros de smartphones modernos pueden ser utilizados como una herramienta efectiva para medir el movimiento humano, abriendo de esta forma nuevas posibilidades para la investigación experimental y clínica como la de este caso.

Por otro lado, Nishiguchi et al. [28] evaluaron la fiabilidad y validez del acelerómetro de un smartphone y de un acelerómetro convencional. Los resultados demostraron una alta correlación entre los análisis, sugiriendo que los smartphones pueden detectar los parámetros de la marcha con precisión comparable. Este estudio apoya el uso de smartphones como herramienta efectiva para medir la marcha y evaluar la fragilidad física.

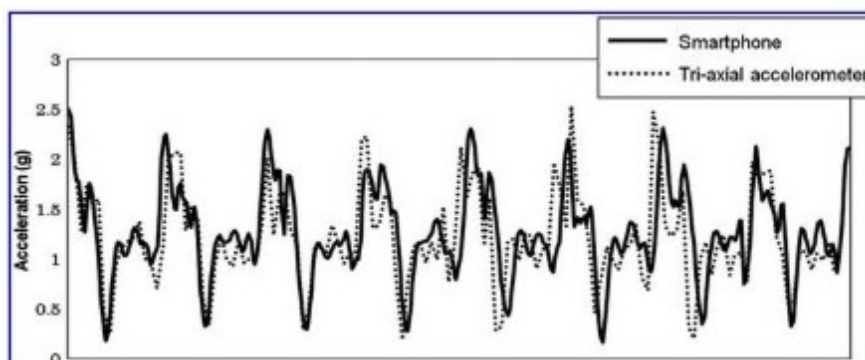


Figura 2.1: Formas de onda de aceleración del smartphone y del acelerómetro triaxial.

Un estudio reciente realizado por Shahar y Agmon (2021) valida el uso de un smartphone con acelerómetros integrados para medir algunos parámetros espaciales y temporales del caminar de una persona, comparando su precisión con un sistema de referencia denominado (APDM Mobility Lab). Este estudio demuestra que los smartphones pueden proporcionar datos precisos sobre la velocidad de la marcha, además de la cadencia y la longitud de los pasos, entre otros muchos parámetros. Aunque, evidentemente la precisión de la medición es menor para algunos parámetros, la consistencia suele ser alta, lo que respalda el uso del smartphone para realizar análisis de la marcha en diversos entornos [12].

Paralelamente, durante los últimos años, el desarrollo de técnicas de inteligencia artificial ha evolucionado la forma en la que se analizan y utilizan los datos de salud, ofreciendo herramientas avanzadas para la predicción de resultados clínicos y la personalización de intervenciones en varios ámbitos. Sin ir más lejos, existen numerosos estudios que realizan mediciones de diferentes características de la marcha utilizando el teléfono [9] [10], así como otros ejemplos que emplean técnicas de deep learning para detectar enfermedades como la esclerosis múltiple [8]. Además, también hay investigaciones sobre la combinación de los wearables con deep learning para realizar un reconocimiento de la actividad [11]. Por todo esto, la aplicación de la inteligencia artificial en la evaluación de la fragilidad en personas mayores presenta una oportunidad de detección temprana.

En relación con mi trabajo de fin de grado, estos enfoques comparten la utilización de dispositivos específicos de sensores portátiles o smartphones para la medición de la fragilidad. Sin embargo, a diferencia de estos enfoques, mi investigación se centra en el desarrollo de un prototipo aprovechando los avances tecnológicos mencionados anteriormente sobre el reconocimiento de la actividad, no obstante, esta vez el objetivo abordado será la identificación de personas frágiles. Los dispositivos investigados presentan la ventaja de su gran ubicuidad, al tratarse de instrumentos comúnmente utilizados por toda la población. Esta característica los convierte en una opción ventajosa para la autoevaluación de la fragilidad física, o incluso para el uso de un dispositivo más económico que permita iniciar un cribado universal sobre este fenómeno.

Capítulo 3

Evaluación Preliminar de Dispositivos y APIs

En el desarrollo de cualquier proyecto de investigación, la selección de las herramientas y tecnologías apropiadas constituye una de las fases más importantes a la hora de determinar el éxito o el fracaso del mismo. En este apartado, se describe la fase inicial de exploración y evaluación de distintos dispositivos y API, que se llevó a cabo para identificar las opciones más adecuadas para el estudio. Además, se detalla el proceso por el cual se probaron varias alternativas para el desarrollo, enfrentándonos de esta forma a distintos desafíos y limitaciones que guiaron las decisiones metodológicas finales del proyecto. Esto proporcionará un entendimiento de porque se seleccionaron finalmente ciertas tecnologías y dispositivos sobre otros inicialmente considerados como válidos.

3.1. Experimentación Inicial

3.1.1. Selección y Pruebas Preliminares de Dispositivos

En la etapa inicial, se llevó a cabo una evaluación exhaustiva de varios dispositivos para determinar cuál sería el más adecuado para cumplir con los objetivos específicos del estudio. Esta selección preliminar se centró en características esenciales para la investigación, tales como la disponibilidad del acelerómetro, la durabilidad de la batería, y algunas otras que comentaremos de manera más específica en secciones posteriores.

Se consideraron durante esta fase diversos modelos de smartwatches de marcas reconocidas como Xiaomi, Huawei, Fitbit, así como dispositivos de Garmin. Cada uno de estos dispositivos ofrecía diferentes ventajas, así como desventajas y rangos de precio. Por ejemplo, algunos modelos destacaban por ofrecer funcionalidades avanzadas de salud, mientras que otros podrían ser preferidos por

su relación calidad-precio, especialmente importante para mantener el proyecto dentro de unos presupuestos asignados.

Durante esta evaluación, para los relojes de Huawei no se encontró ningún artículo que hablase sobre la posibilidad de una exportación de los datos recopilados por el acelerómetro en tiempo real, lo que limitó claramente su viabilidad en el proyecto. Por otro lado, para los dispositivos Xiaomi se identificó una web con documentación acerca de las herramientas de desarrollo [19], no obstante, al consultarla se podían observar varios problemas, como que las últimas contribuciones a dicha documentación fueron en 2021. A pesar de todo esto, se realizaron varios intentos de desarrollo, sin embargo, las indicaciones parecían muy incompletas, las traducciones poco acertadas y los distintos archivos que debíamos descargar para utilizar el IDE mostraban diversos errores a la hora de descomprimirlos, por lo que finalmente se decidió buscar opciones más viables.

Siguiendo un poco con la investigación de dispositivos de esta marca, en GitHub, encontramos un repositorio muy interesante en el que se mostraba un ejemplo de código para poder exportar la información del acelerómetro en tiempo real con un video de ejemplo [20]. Esto hizo que los dispositivos Xiaomi se convirtieran en una opción muy atractiva para tratar de desarrollar el prototipo. Es por esto que fue uno de los elegidos, de forma más concreta, el modelo Xiaomi SmartBand 8 cuyo precio rondaba los 40€. No obstante, mientras se realizaba el pedido y la compra de dicho dispositivo, se decidió continuar con la investigación de otras marcas.

La marca de relojes Garmin tiene una calidad de muy alto nivel, no obstante, sus precios son demasiado elevados y además, rápidamente se descubrió que su API oficial permitía el acceso a los datos del acelerómetro, pero no en tiempo real, descartándolo como opción viable. Finalmente, la opción de Fitbit, disponía de una API oficial, que en una primera impresión, permitía la recogida de datos del acelerómetro en tiempo real. Por otro lado, dado que esta opción parecía bastante atractiva, la tutora del proyecto mencionó que disponía de un modelo Fitbit Alta HR, lo que permitió realizar pruebas preliminares mientras se esperaba por la llegada del dispositivo Xiaomi. En esta primera toma de contacto, el Fitbit proporcionó una oportunidad para explorar sus capacidades en un contexto real y tratar de comenzar con el inicio de un pequeño desarrollo de prueba para testar sus limitaciones.

3.1.2. Prueba de desarrollo en Fitbit Alta HR

Siguiendo con la evaluación inicial de dispositivos, la opción de Fitbit Alta HR parecía una alternativa prometedora. Gracias a la disponibilidad de este modelo, se pudo proceder con pruebas preliminares sin incurrir en costes adicionales sobre

el proyecto. Al comenzar con la exploración de las posibilidades nos encontramos con [21] donde se realiza un tutorial de cómo utilizar la Web API oficial de Fitbit. Además, también encontramos en esta misma guía la siguiente sección [22] donde se muestran ejemplos de código de cómo obtener los datos del acelerómetro en tiempo real. Esto resulta confuso, ya que en este caso, no se comprende del todo si se está hablando sobre la Web API u otra API diferente. No obstante, con toda esta información, parece que el desarrollo sería posible, por lo que comenzamos con ello.

En primer lugar, se pide que registremos una aplicación para poder comenzar a utilizar la Web API. En ella configuramos todas las opciones para poder acceder a la información de los sensores sin problema. Ver Apéndice B.1.1 con imágenes. En la pantalla de configuración podemos observar un **formulario de registro de la aplicación** que más tarde nos permitirá la integración de la API oficial en nuestro desarrollo. Como vemos, se piden campos como:

- **Nombre de la aplicación** o la **descripción**, que son campos sencillos de rellenar.
- **URL del sitio web de la aplicación**, **URL del sitio web de la organización**, o la **URL de los términos de servicios y privacidad**. Al tratarse de un simple prototipo de prueba, rellenaremos los datos con URLs de USC de manera que no nos dé ningún error al registrarla.
- **OAuth 2.0 Application Type**, que determina el tipo de aplicación en relación al uso de OAuth 2.0 para la realización de autenticaciones seguras. En nuestro caso será una aplicación para clientes debido a las características de la misma y que estaría ideada para utilizarse en un entorno de aplicación móvil o escritorio.
- **URL de redirección** que será la dirección a donde se redirijan los usuarios una vez completada la autenticación.
- **Tipo de acceso por defecto** que será *Read and Write* para que la aplicación pueda leer y modificar datos.

Una vez tenemos la aplicación registrada, se nos devolverán los siguientes datos. Ver Apéndice B.1.2 con imágenes. Estos serán necesarios para formar la URL de autorización, necesaria para rellenar el formulario de los datos a los que queremos acceder. En versiones anteriores de la documentación de la API, la creación de esta URL era manual, sin embargo, ahora disponemos de la siguiente sección [23], que nos ayuda a crear dicha URL simplemente rellenando los campos con los datos obtenidos una vez tenemos la aplicación registrada. Al acceder a la URL, se nos muestra la siguiente página mostrada en la imagen de la derecha, donde

tenemos toda la información que podemos obtener en la API. Ver Apéndice B.1.3 con imágenes.

En este momento, al no observar ninguna casilla con algún término relacionado con el acelerómetro, se marcan todas para así poder probar las opciones que nos ofrece esta API. Una vez enviemos el formulario, se nos devolverá un token de acceso con el cual podremos comenzar a utilizar la API. Aquí tenemos un ejemplo de código con el que podremos acceder por medio de una consulta a la información en Python. Ver Apéndice A.1.1 para el código completo.

Como podemos ver, el código para obtener información acerca del dispositivo es muy sencillo, simplemente debemos importar las librerías necesarias y utilizar el token de acceso para realizar una petición del archivo JSON con la información solicitada. Explorando un poco más la documentación, observamos que esta manera de acceder al token resulta muy poco segura. Por esto, se propone el uso del PKCE (Proof Key for Code Exchange) que introduce una capa de seguridad adicional en comparación al acceso normal a través de OAuth 2.0. El uso de esta herramienta implica que la aplicación genere tres valores únicos para cada solicitud de autorización. De esta forma, el código de acceso a la API sería el siguiente. Ver Apéndice A.1.2 para el código completo.

Una vez configurado un acceso seguro a la información, se retomó la investigación acerca del posible acceso a los datos, por medio de la API. No obstante, visitando distintas páginas de documentación donde se explica el uso de la misma [24] [25], no se encontraba nada relacionado con el apartado [22] donde se referenciaba claramente la posibilidad de obtener información del acelerómetro. Al no encontrar ninguna evidencia, decidí enviar una pregunta al foro de desarrolladores de Fitbit. Esta fue respondida de manera clara por el equipo [26]. Ver Apéndice B.1.4 con imágenes.

Como podemos observar en la imagen, uno de los desarrolladores nos indica que la Web API está limitada en el aspecto de exportación de datos del acelerómetro, y además indica que para realizar una aplicación de este estilo, se necesitaría un modelo de smartwatches muy concretos que son compatibles con esta SDK. Sabiendo esto, comenzamos con una exploración de los modelos concretos que disponen de estas cualidades, y encontramos rápidamente sus nombres. Al realizar una consulta de los precios de los mismos observamos cómo se salen de nuestro presupuesto (rozan los 200€), por lo que tendremos que explorar nuevas opciones.

3.1.3. Exploración de Alternativas Tecnológicas

Una vez exploradas las limitaciones con los wearables, se comenzó a explorar alternativas tecnológicas que permitieran una implementación de las funcionalidades necesarias para el estudio. Esta vuelta atrás resultó esencial debido a la información que surgió durante la evaluación de los anteriores smartwatches.

Mientras se esperaba la llegada de la SmartBand de Xiaomi, de la cual se habían encontrado diversos repositorios que demostraban su potencial para el acceso a la información del acelerómetro en tiempo real, se investigaron otras tecnologías. Una de estas nuevas tecnologías exploradas fue la librería de desarrollo Kivy para Python, la cual permitía la creación de aplicaciones multiplataforma con acceso a varias APIs de los dispositivos móviles. Sin embargo, rápidamente se encontraron diversos obstáculos en la utilización de este tipo de bibliotecas debido a que era necesaria una compilación por medio de una herramienta denominada Buildozer, lo cual dificultaba enormemente las iteraciones y pruebas del prototipo durante su desarrollo debido a los altos tiempos de espera a la hora de compilar.

Ante estos desafíos, se continuó con la búsqueda de soluciones más eficientes, llevando finalmente al descubrimiento de Android Studio y Kotlin. Estas herramientas ofrecen un entorno de desarrollo nativo para dispositivos android, además de una compilación de los proyectos mucho más rápida que en las tecnologías anteriores. Con todo esto, se realizaron diversas pruebas de aplicaciones para comprobar su funcionamiento y se descubrió que facilitaba enormemente el trabajo de desarrollo y la realización de pruebas en los dispositivos android, pudiendo llegar a probar cualquier proyecto en diversos dispositivos con tan solo conectar el mismo a nuestro equipo con un cable USB.

3.2. Decisión Final y Cancelación de Pedidos

Finalmente, debido al descubrimiento de Android Studio en combinación con el lenguaje de desarrollo nativo Kotlin, se tomó la decisión de cancelar el pedido de la Smart Band de Xiaomi. Esto se debía a varios motivos; en primer lugar, las capacidades de las tecnologías descubiertas se alineaban mejor con los objetivos del proyecto. Además, los diversos repositorios que demostraban la posibilidad de exportación de datos del acelerómetro en tiempo real se llevaban a cabo con tecnologías no oficiales, lo que podía llevarnos a comprometer la seguridad del dispositivo. La posibilidad de utilizar una MiBand de Xiaomi aún estaba contemplada, sin embargo, esta no ofrecía ninguna API o SDK oficial para realizar una exportación de la información de forma segura. Es por esto que parecía preferi-

ble utilizar una opción que permitiera el desarrollo de esta aplicación mediante métodos oficiales que no tuvieran riesgo de inutilizar el dispositivo.

Capítulo 4

Metodología

En el ámbito sanitario que aborda este TFG, mas concretamente el campo de la geriatría, la detección precoz de la fragilidad mediante el uso de las nuevas tecnologías es crucial para la prevención del deterioro funcional en las personas de la tercera edad. Además, dada la prevalencia de los dispositivos inteligentes ya sean smartphones o wearables, que están equipados con sensores capaces de medir una amplia gama de movimientos y actividades físicas, este proyecto busca explorar como estos dispositivos pueden ser utilizados eficazmente para monitorizar y evaluar la fragilidad en ancianos.

El trabajo se centra en el desarrollo de un prototipo que acceda al acelerómetro integrado en uno de estos dispositivos para medir la velocidad de la marcha. Este parámetro ha sido identificado repetidas veces en la literatura como un predictor fiable de fragilidad, caídas, hospitalización y mortalidad en ancianos. Por ello, proporcionar una herramienta que facilite esta medición de manera sencilla y no invasiva podría presentar un gran avance en la atención preventiva en el ámbito de la geriatría.

Por eso, para asegurar que los datos recogidos son válidos, se adoptará un procedimiento de selección de tecnología para el desarrollo de la aplicación. En este capítulo se describirá en detalle los materiales y equipos utilizados, el diseño experimental y los métodos matemáticos y estadísticos utilizados para el análisis de los datos y la obtención de las velocidades a partir de los distintos parámetros del acelerómetro.

4.1. Selección de Dispositivos y Plataformas

4.1.1. Criterios de selección

La selección de un dispositivo adecuado se basó en una evaluación detallada de las capacidades técnicas requeridas. Algunos de los criterios utilizados serían los siguientes:

- **Disponibilidad de un acelerómetro:** Esencial para capturar movimientos precisos al realizar la prueba para medir la fragilidad.
- **Facilidad de uso para usuarios mayores:** Debe ser fácil de manejar y entender, debido a que este prototipo está destinado a ser usado en personas mayores.
- **Compatibilidad con herramientas de desarrollo accesibles:** Importante para la integración y el desarrollo de la aplicación, de manera que por ejemplo el acceso a la información del acelerómetro en tiempo real sea sencilla.
- **Durabilidad de la batería:** Capacidad de funcionar durante sesiones prolongadas sin recargas frecuentes. De esta forma se facilita el trabajo de los profesionales sanitarios para que puedan realizar un cribado por medio de esta herramienta.
- **Precio del dispositivo:** Tal y como se comenta en la introducción, uno de los puntos más importantes es que el dispositivo sea asequible para todos, y la relación de calidad-precio adecuada.

4.1.2. Proceso de Evaluación

Inicialmente, tal y como se comenta en secciones anteriores, se consideraron varios dispositivos wearables, incluidos modelos de Xiaomi, Huawei y Fitbit, debido a la calidad de sus sensores. No obstante, estos dispositivos presentaron limitaciones a la hora de acceder a los datos del acelerómetro en tiempo real, lo cual es una característica crucial para que la realización de las pruebas conlleve un resultado inmediato sobre el estado de fragilidad del paciente.

Con esta investigación, se llegó a la conclusión de que los smartphones ofrecían opciones más sencillas a la hora de acceder a los distintos sensores necesarios en las mediciones. En concreto, en los teléfonos android, se encontraban lenguajes y plataformas de desarrollo más robustas. Por ejemplo con la combinación del lenguaje nativo de Android denominado 'Kotlin' a través del IDE de 'Android Studio'. Es por esto que la decisión de utilizar un smartphone en lugar de un wearable se basó en diversos factores:

- **Acceso completo a la API del acelerómetro:** Los smartphone proporcionan APIs mas completas que permiten un acceso mas sencillo y completo a los datos del acelerometro.
- **Pantalla más grande y mejor interfaz de usuario:** Teniendo en cuenta que el objetivo de este prototipo es el análisis de la fragilidad en personas mayores, es esencial para una interacción mas cómoda con la aplicación por parte de los usuarios.

4.2. Materiales Utilizados

4.2.1. Descripción del dispositivo seleccionado: Nokia 6.1

En el desarrollo del prototipo de la aplicación, se ha seleccionado el dispositivo Nokia 6.1 como smartphone donde se realizará el proyecto. A continuación, mostraremos algunas características del mismo [18]:

- **Especificaciones Técnicas:** Equipado con un procesador Qualcomm Snapdragon 630, 3 GB de RAM y 32 GB de almacenamiento, adecuados para procesamiento y almacenamiento de datos en tiempo real.
- **Sistema Operativo:** Utiliza Android 9.0, que soporta actualizaciones modernas y facilita la integración de APIs avanzadas.
- **Sensores Integrados:** Incluye varios sensores, entre ellos el acelerometro, sin embargo, no se ha encontrado información detallada que destaque la calidad del acelerómetro del Nokia 6.1 frente a otros dispositivos. Las revisiones disponibles no proporcionan análisis específicos de este componente, indicando que cumple con las funciones estándar de los smartphones modernos.
- **Vida de la batería:** La batería del dispositivo ofrece hasta 76 horas de duración, que permite un uso prolongado sin necesidad de recargarlo frecuentemente.
- **Pantalla y Interfaz de Usuario:** Dispone de una pantalla Full HD de 5.5 pulgadas que ofrece una interfaz clara y accesible.

Además, utilizando la aplicación 'Accelerometer Meter' desarrollada por 'Keuwlsoft' se ha podido extraer la información acerca de todos los sensores de los que dispone el teléfono además de mucha otra información valiosa. En primer lugar, acerca de los distintos sensores de los que dispone nuestro dispositivo y algunas especificaciones técnicas mas interesantes.

Sensor	Nombre	Proveedor	Versión
Acelerómetro	BMI160 Accelerometer	BOSCH	2062705
Acelerómetro Lineal	Linear Acceleration	QTI	2
Sensor de Gravedad	Gravity	QTI	2
Giroscopio	BMI160 Gyroscope	BOSCH	2062705
Magnetómetro	MMC3630KJ Magnetometer	MEMSIC	1
Sensor de Luz	stk3x1x alsprx	Sensortek	1
Sensor de Proximidad	stk3x1x alsprx	Sensortek	2

Cuadro 4.1: Información Básica de los Sensores del Smartphone

Sensor	Resolución (m/s ²)	Rango Máximo (m/s ²)	Potencia (mW)
Acelerómetro	0.0023956299	156.9064	0.18
Acelerómetro Lineal	0.0023956299	156.9064	1.0799866
Sensor de Gravedad	0.0023956299	156.9064	1.0799866
Giroscopio	0.0010681152	34.906586	0.9
Magnetómetro	0.09765625	3000.0	0.32
Sensor de Luz	0.5399933	4096.0	0.001
Sensor de Proximidad	0.05000305	5.0000305	0.9

Cuadro 4.2: Información Técnica de los Sensores del Smartphone

Esta información técnica puede ser muy útil para la investigación que estamos realizando. Esto se debe a que, por ejemplo, en el caso de la resolución, se nos está indicando la menor cantidad de aceleración que el sensor puede detectar. En nuestro caso, tal y como se observa en la tabla, la resolución permite detectar cambios muy pequeños en la aceleración, lo que puede mejorar enormemente la calidad de los datos recogidos. Por otro lado, el rango máximo, tal y como indica el nombre, nos indica la máxima aceleración que el sensor puede medir. Finalmente, la potencia es fundamental para prolongar la duración de las baterías. En este caso, los valores de estas tres características son muy buenos, por lo que podemos proseguir con la investigación sin mayor problema.

4.3. Procedimientos Experimentales

4.3.1. Investigación Inicial

Para llevar a cabo el estudio, se realizó una investigación preliminar sobre los problemas que se detectan en el uso del smartphone como detector de la velocidad. Esta fase ayudó a identificar y abordar los desafíos comunes en la medición de la velocidad a través de sensores de acelerómetro en dispositivos móviles.

4.3.2. Funcionamiento del Acelerómetro

El acelerómetro es el dispositivo fundamental en nuestro estudio. Este sensor mide la aceleración a lo largo de los tres ejes cartesianos, proporcionando datos de aceleración en m/s^2 .

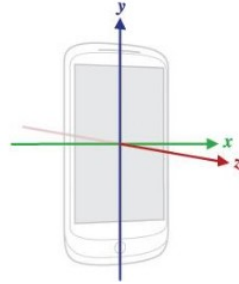


Figura 4.1: Direcciones de los ejes cartesianos en el teléfono. [31]

En nuestro caso, se puede observar gracias a las indicaciones de distintas aplicaciones, los valores de cada uno de los ejes dependiendo de la posición. Por ejemplo, en la siguiente imagen se muestran sus valores, cuando el teléfono se encuentra en reposo, con la pantalla mirando cara arriba. Es importante mencionar que estos valores han sido obtenidos por medio de la aplicación 'Accelerometer Meter' ya mencionada anteriormente.



Figura 4.2: Valores del acelerómetro en reposo.

4.3.3. Orientación del teléfono

En este artículo [30] se aborda uno de los primeros problemas detectados, relacionado con la orientación del teléfono durante las mediciones. Este es un problema importante, ya que, aunque el dispositivo se lleve siempre en el mismo

lugar (por ejemplo, en una mochila o en el bolsillo), su orientación puede cambiar. Esto afecta a la precisión en el reconocimiento de las actividades. Ante este problema, el artículo menciona dos posibles enfoques.

En el primer enfoque, se transformaría el sistema de coordenadas, ajustando las lecturas del acelerómetro de manera que se alineen con un sistema de referencia común. De esta forma, aunque el teléfono se mueva en diferentes direcciones, las lecturas se transformarían para coincidir con una referencia fija, permitiendo así una interpretación consistente de las lecturas.

El segundo enfoque consiste en no utilizar características que dependan de la orientación del teléfono. Esto se puede lograr, por ejemplo, calculando la magnitud total de la aceleración, una característica que siempre representará la misma aceleración total independientemente de la orientación del teléfono. Esta característica se calcula sumando y elevando al cuadrado la señal del acelerómetro en los tres ejes, después de eliminar la componente estática de la gravedad [29].

$$\text{Magnitud aceleración} = \sqrt{X^2 + Y^2 + Z^2} \quad (4.1)$$

Transformar el sistema de coordenadas puede mejorar el rendimiento hasta un 20 %, sin embargo, es una técnica menos práctica en situaciones reales donde la orientación del teléfono puede cambiar continuamente. Por ello, se consideró que utilizar características independientes de la orientación, como la magnitud de aceleración, sería un indicador más robusto. A pesar de todo esto, estos dos enfoques no resultaron muy atractivos ni prometedores, por lo que se investigó la posibilidad de utilizar algún instrumento para mantener el teléfono en una posición concreta durante la realización de las pruebas y que propiciara resultados fiables en sus mediciones.

En el estudio [33] se comparan los resultados de las estimaciones de velocidad y otros parámetros realizadas por la tecnología GAITRite, ya mencionada anteriormente, y un smartphone. El resultado fue que las mediciones de los datos realizadas por los teléfonos móviles tuvieron una excelente confiabilidad ($\text{ICC}_{2,1} \geq 0.90$) para las ubicaciones del cuerpo y el cinturón, y una confiabilidad de buena a excelente ($\text{ICC}_{2,1} \geq 0.60$) para las ubicaciones de bolso, manos y bolsillos. Todo esto, manteniendo el teléfono en una posición estática por medio de un bolso, una funda o una riñonera, lo que resultó ser uno de los enfoques más favorables a la naturaleza de la investigación.

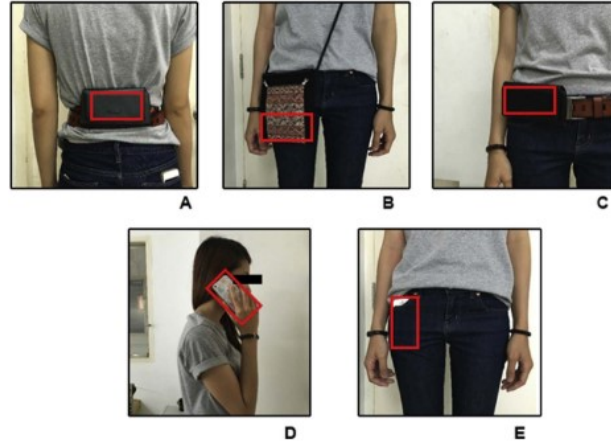


Figura 4.3: Ubicaciones del teléfono para las mediciones; cuerpo (A), bolso (B), cinturón (C), mano (D) y bolsillo (E). El borde rectangular indica la orientación y ubicación del teléfono inteligente en cada condición [33]

Dado que los mejores resultados se obtienen cuando el teléfono se encuentra en el cuerpo o el cinturón, utilizaremos la ubicación en el cuerpo, como se muestra en la imagen anterior para realizar las mediciones. Esta es una parte importante a establecer en el protocolo de medición, ya que la orientación del teléfono puede tener una gran repercusión en los resultados obtenidos. Al establecer la ubicación en el cuerpo, el eje principal de movimiento será el eje Z).

4.3.4. Eliminación del ruido del acelerómetro

En una primera instancia del desarrollo del prototipo en kotlin, se comenzó por imprimir los valores de las mediciones captadas por el acelerómetro de nuestro smartphone. Esto se realizó por medio del siguiente código en kotlin. Ver Apéndice A.2.1 para el código completo.

Al ejecutar el código en nuestro dispositivo dejándolo en reposo con la pantalla hacia arriba en una superficie aparentemente plana, los valores del acelerómetro observados son los siguientes:

- Eje X: 0.12 a 0.14 m/s^2
- Eje Y: -0.30 a -0.29 m/s^2
- Eje Z: 10.04 a 10.06 m/s^2

Estos valores en reposo no son exactamente cero debido a diversas razones, como la presencia de ruido en las mediciones, la orientación del teléfono y la gravedad terrestre. El valor de z alrededor de 10.04 a 10.06 m/s^2 se debe a la

gravedad terrestre. Como ya sabemos, el acelerómetro mide la aceleración en todas las direcciones y cuando el teléfono se encuentra en reposo, uno de los ejes (en este caso el eje Z) está alineado con la gravedad, que es una constante de aproximadamente 9.81 m/s^2 . Las pequeñas fluctuaciones se deben a ruidos en la medición. Por otro lado, los valores de x de entre 0.12 a 0.14 m/s^2 y los de y de -0.30 a -0.29 m/s^2 son valores pequeños pero no cero debido al ruido inherente de los sensores ya mencionado anteriormente para el eje Z . Esto es, el ruido eléctrico y mecánico que causa pequeñas variaciones en las lecturas.

Finalmente, la orientación del teléfono afecta a cómo se mide la aceleración. Si el teléfono no está perfectamente plano, la componente de gravedad se distribuirá entre los diferentes ejes x , y y z . Además, los acelerómetros no son perfectos y pueden tener errores de offset (desviación de ceros), lo que causaría que las lecturas no sean nulas incluso cuando no existe ninguna aceleración. Es por esto que para mejorar la precisión y reducir el ruido en las mediciones debemos aplicar técnicas de filtrado y calibración de estos sensores. En un primer momento, se trató de realizar el filtrado por medio de un 'Filtro de Paso Bajo' que ayudaría a suavizar las lecturas y reducir el ruido.

Este 'Filtro de paso bajo' se menciona comúnmente en estudios con características similares al nuestro, donde se mide en primer lugar las frecuencias en las que se concentra el ruido de nuestro acelerómetro por medio de un análisis espectral, como en el estudio de Shrestha y Won [32], entre 0 y 15 Hz . Por lo tanto, se establece el filtro con una frecuencia de corte de 15 Hz , perfecto para eliminar el ruido sin afectar significativamente a las señales útiles del sensor. No obstante, en nuestro caso el filtro utilizado no será tan preciso, ya que no se realizará un análisis espectral del ruido de nuestro dispositivo. Esto se debe a que se estaba realizando una exploración de los posibles filtros a utilizar. De esta forma, se utilizó la siguiente función que simulaba el efecto deseado mediante una implementación mucho más sencilla.

Este tipo de filtros son simples pero muy eficaces para suavizar los datos y reducir el ruido de las señales. La fórmula implementada sería la siguiente:

$$y[i] = \alpha \cdot x[i] + (1 - \alpha) \cdot y[i - 1] \quad (4.2)$$

- $y[i]$ es la salida filtrada en el tiempo i .
- $x[i]$ es la entrada no filtrada en el tiempo i .
- α es el coeficiente de suavizado ($0 < \alpha < 1$).
- $y[i - 1]$ es la salida filtrada en el tiempo $i - 1$.

El coeficiente de suavizado α ajusta el balance entre la entrada actual $x[i]$ y la salida previa $y[i - 1]$, donde valores cercanos a 1 priorizan la entrada actual y valores cercanos a 0 priorizan la salida previa. Una vez aclarado esto, presentaremos el código de la función Kotlin que sería algo como lo que se observa a continuación. Ver Apéndice A.2.2 para el código completo.

Con esta función, al aplicar los cálculos del filtro sobre las lecturas del acelerómetro, seremos capaces de separar la aceleración lineal de la gravedad de cada eje. En mi caso el valor de alpha será de 0.50, ya que es la que mejor resultados devuelve. Por lo tanto nos quedaría así el código de invocación. Ver Apéndice A.2.3 para el código completo.

Como se observa en el código anterior, una vez calculamos el valor de la gravedad aplicando el filtro de paso bajo en cada eje, se aplicará la diferencia a la medición de cada dirección, de forma que se restará la gravedad de las lecturas y obtendremos la aceleración lineal. Una vez tenemos dichas aceleraciones, podremos calcular la '**Magnitud de aceleración**' 4.1. Ver Apéndice A.2.4 para el código completo.

Con todas estas mejoras, al arrancar la aplicación podemos observar que las lecturas siguen fluctuando ligeramente, sin embargo, estas siempre están en rangos muy cercanos a 0. No obstante, los resultados no parecían suficientemente alentadores por lo que se exploraron nuevas posibilidades. En primer lugar se encontraron los filtros FIR (Filtro de Respuesta finita de impulso). Este permite introducir una frecuencia de corte precisa, lo que elimina el ruido en rangos de frecuencia definidos. Para ello, es importante realizar un estudio preliminar del ruido del sensor en reposo para calcular diversas variables. Explorando un poco más, observamos que estudios como [32] implementan este tipo de filtros.

Por otro lado también tenemos los filtros IIR (Respuesta de Impulso Infinita), que es un filtro que tiene incluso una mayor precisión que los anteriormente mencionados. Estudios como [33] con muy buenos resultados en las mediciones de velocidad, utilizan este tipo de filtros para la eliminación del ruido. Concretamente el filtro Butterworth de cuarto orden, que es ideal para aplicaciones que requieren una alta precisión en la medición de sus datos.

Debido a los buenos resultados mostrados en el estudio anterior, decidimos proceder con la implementación de este tipo de filtros, ya que el objetivo de nuestro TFG requiere de una aplicación que sea capaz de obtener la velocidad de la marcha con la mayor precisión posible. Para ello seguiremos los pasos reflejados en la metodología del artículo, de forma que reproduciremos el experimento tratando de obtener unos resultados similares. Toda este procesamiento de los datos se realizará por medio de funciones en MatLab.

En primer lugar, tal como se explica en la sección de análisis de datos, procederemos a recoger una muestra con nuestro acelerómetro. Este primer experimento se realizará con el teléfono en reposo, recopilando los datos durante aproximadamente 15 o 20 segundos. Esto se hará por medio de la aplicación *Accelerometer Meter*, que nos permitirá tomar una muestra durante el tiempo necesario para luego exportar la información a un fichero CSV. Luego, tal y como se especifica en la investigación, se realizará un remuestreo cambiando la frecuencia a 100 Hz mediante la función *resample* en MATLAB. Esto se hace debido a que la frecuencia de muestreo de los acelerómetros de teléfonos móviles no suele ser constante, lo que nos dejará con el siguiente resultado:

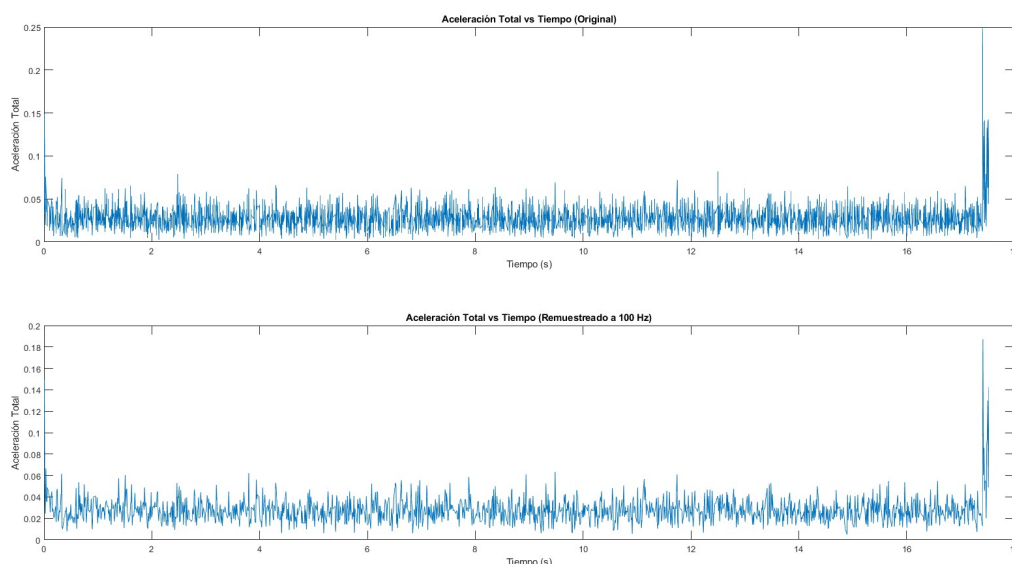


Figura 4.4: Remuestreo valores acelerómetro.

A continuación, siguiendo las indicaciones, deberemos filtrar las señales de los tres ejes del acelerómetro utilizando un filtro Butterworth de 4^o orden y paso bajo con una frecuencia de corte de 20 Hz. Luego, se aplicará nuevamente un filtro Butterworth de 4^o orden y paso bajo, pero esta vez con una frecuencia de 2 Hz. La razón para aplicar estos dos pasos de filtrado es para asegurar una eliminación más gradual y controlada de las componentes de alta frecuencia, lo cual puede mejorar la estabilidad numérica y la precisión del filtrado en sistemas con ruido significativo. Finalmente, se mostrarán los resultados aplicados a la gráfica anterior. La aplicación de estos filtros se realizará de manera muy simple gracias a la función *butter* de MATLAB. Es importante recordar también que estos datos han sido obtenidos del acelerómetro cuando el teléfono móvil se encontraba en reposo, por lo que la señal ideal a conseguir sería 0 en los distintos ejes.

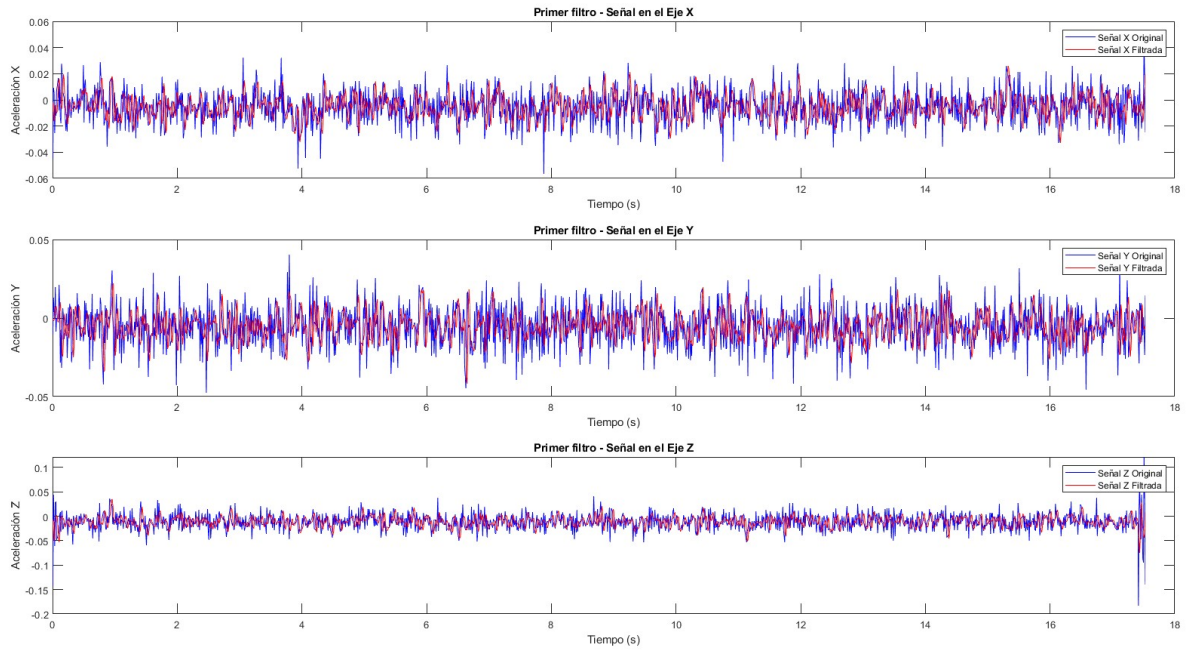


Figura 4.5: Señal tras aplicar filtro de 20 Hz.

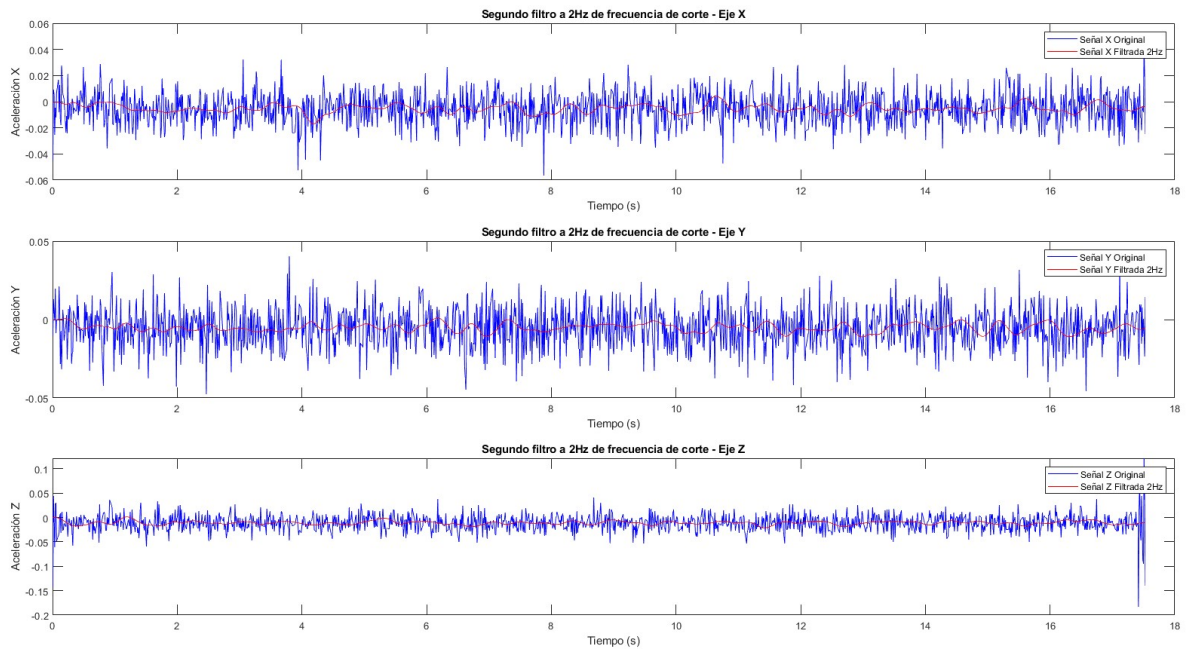


Figura 4.6: Señal tras aplicar filtro de 2 Hz.

Este sería el resultado de la aplicación de los filtros mencionados en el estudio anterior. Es posible que en este caso no se observe del todo claro el enorme cambio que supone la aplicación de dichos filtros en las mediciones de nuestro acelerómetro. Es por esto que se generó a mayores la siguiente imagen, en la cual se podrá observar en detalle, la diferencia que supone.

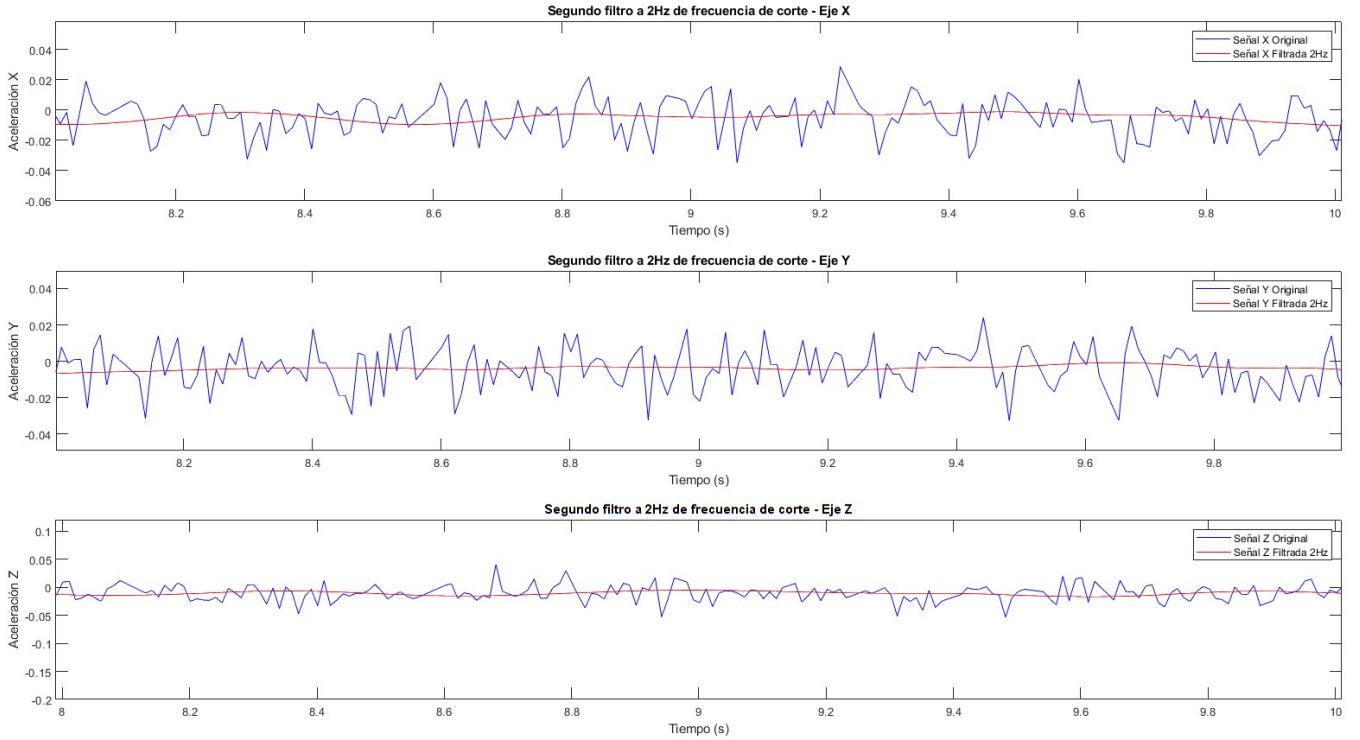


Figura 4.7: Señal tras aplicar filtro de 2 Hz en detalle.

Como podemos observar, al tratarse de unos datos recopilados mientras el teléfono móvil estaba en reposo, la aceleración recogida está entorno al 0 en todo momento. Lo cual es una señal excelente que nos da paso a comenzar con los intentos de estimación de velocidad por medio de nuestro dispositivo.

4.3.5. Estimación de la velocidad

En el estudio [32] se presenta un enfoque sobre cómo realizar una estimación de la velocidad de la marcha utilizando solo los sensores del smartphone. De manera más concreta, utilizando el acelerómetro y el giroscopio. Se emplea para ello una red convolucional profunda (DCNN) para identificar y extraer las

características más efectivas de los datos del acelerómetro y el giroscopio.

El modelo entrenado demostró una precisión comparable a las soluciones basadas en múltiples sensores corporales, con un error cuadrático medio (RMSE) promedio de 0.16 m/s. Buscando información por la red, no se encuentra forma alguna de implementar dicha tecnología en nuestro proyecto, debido a que, al parecer, la base de datos utilizada no es pública. Al ponernos en contacto con los investigadores, no se recibe respuesta, por lo que deberemos encontrar otros métodos de estimación para la velocidad.

Revisando la literatura, encontramos que la idea de calcular la velocidad de la marcha a partir de los datos del acelerómetro implica primero derivar otros parámetros de la marcha. Por ejemplo, en esta investigación [34] se trata de capturar datos del acelerómetro en 'crudo' para después determinar eventos de la marcha como los impactos del talón en el suelo, es decir, identificar los pasos de una persona. A partir de estos eventos, se calculan parámetros espaciotemporales tales como el tiempo tardado en dar cada paso y su longitud. Con estos, se puede derivar finalmente la velocidad de la marcha, en lugar de sacarla directamente de los datos del acelerómetro.

Con toda esta información, procederemos a realizar el script que nos permitirá estimar la velocidad de la marcha en MATLAB. Para ello, en primer lugar, necesitaremos unos datos de prueba. En este caso, realizaré una medición utilizando una riñonera que colocaré en la posición (A) de la imagen 4.3.

De esta forma, activaré las mediciones y colocaré el teléfono en la posición deseada. Una vez hecho esto, me mantendré unos segundos quieto simplemente para poder identificar en la gráfica los segundos interesantes a medir. Luego, comenzaré a caminar y, pasado unos segundos, me detendré de nuevo durante cierto tiempo. Así estará mucho más claro el período de caminata. De esta forma, podemos observar en una primera instancia el resultado de los datos en la siguiente imagen:

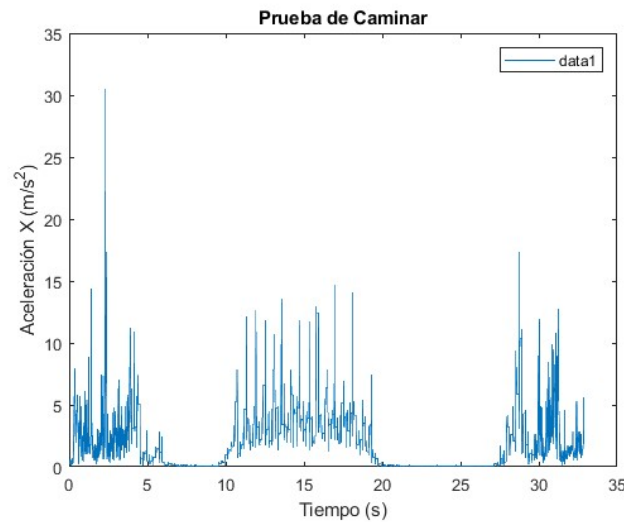


Figura 4.8: Prueba de Caminata.

Como vemos, los datos útiles a analizar serán los tomados entre el segundo 7 hasta el segundo 22, por lo que una vez conocida esta información, podremos comenzar con la eliminación del ruido y la aplicación de los filtros anteriormente comentados en el intervalo útil para así observar el resultado de su aplicación. Con esto, obtendremos los siguientes resultados en los distintos ejes de nuestra muestra.

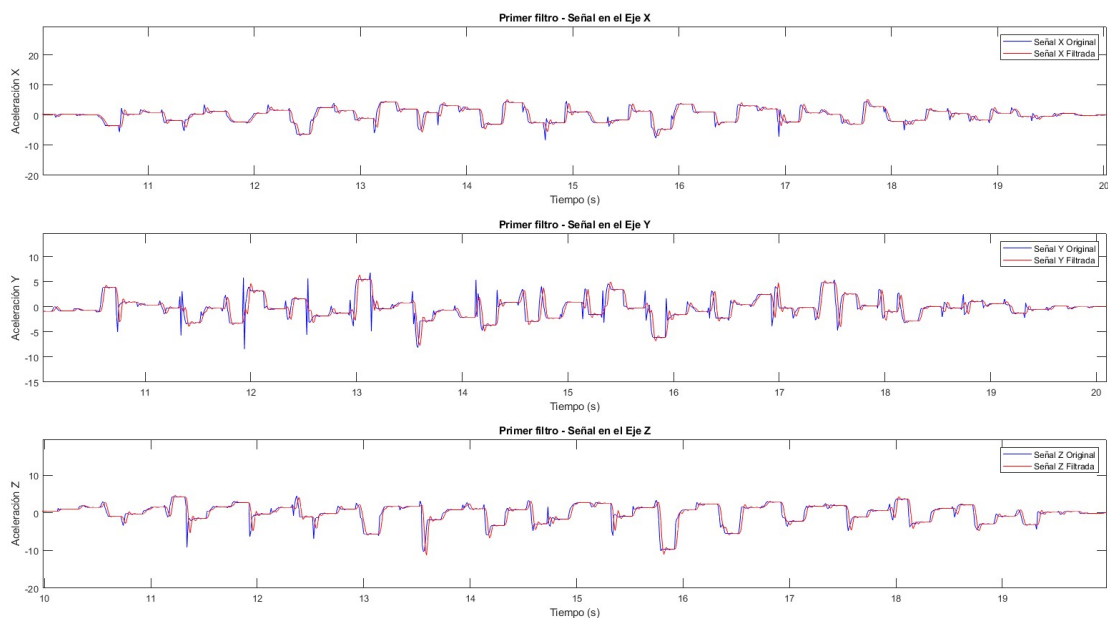


Figura 4.9: Prueba de Caminata Filtro 20 Hz.

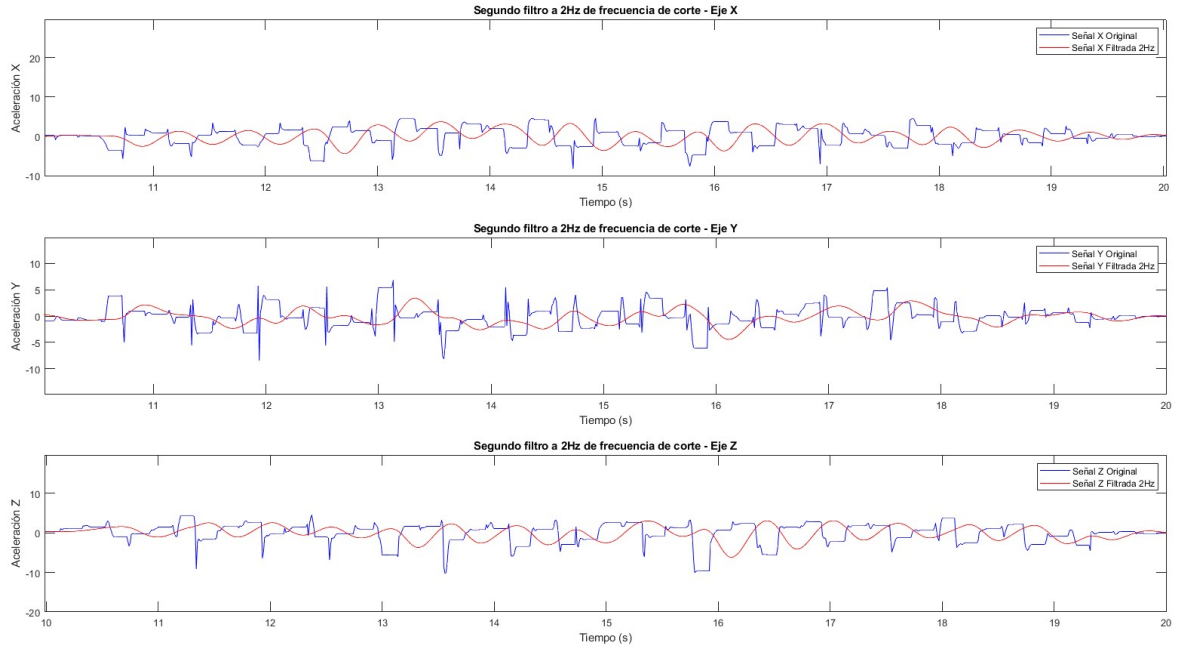


Figura 4.10: Prueba de Caminata Filtro 2 Hz.

Una vez obtenidas las señales filtradas, nos será mucho más fácil identificar picos en los distintos ejes, pudiendo así diferenciar entre golpes de talón en el suelo. De esta forma, siguiendo las indicaciones del estudio anterior, identificaremos los golpes de talón utilizando la función `findpeaks` de Matlab. Luego calcularemos la diferencia de tiempo entre cada uno de los golpes, realizaremos una doble integración de la aceleración vertical añadiéndole al resultado un filtro de paso alto de Butterworth de 4º orden con una frecuencia de corte de 0.1 Hz para eliminar la denominada deriva de la integración. Finalmente, calcularemos la longitud del paso por medio de la siguiente fórmula:

$$SL = 2\sqrt{2hl - h^2} \quad (4.3)$$

donde:

- **SL** es la longitud del paso.
- **h** es el cambio en la posición vertical.
- **l** es la longitud de la pierna del participante.

Por medio de estos datos, es decir, utilizando la longitud del paso y el tiempo entre pasos, seremos capaces de obtener la velocidad de la marcha con una

precisión presuntamente similar a las obtenidas en la siguiente tabla del estudio anterior.

Table 1c
Gait velocity reliability and validity across smartphone location and walking speed.

Location	Speed	GAITrite		Smartphone		Validity			
		Mean \pm SD (m/s)	ICC (2,1)	Mean \pm SD (m/s)	ICC (2,1)	r	Bias	LOA Lower	LOA Upper
Body	Comfortable	1.11 \pm 0.21	0.97	1.06 \pm 0.21	0.93	0.903 *	-0.052	-0.234	0.129
	Slow	0.83 \pm 0.22	0.95	0.76 \pm 0.20	0.89	0.904 *	-0.042	-0.260	0.177
	Fast	1.60 \pm 0.29	0.96	1.47 \pm 0.25	0.94	0.863 *	-0.133	-0.421	0.155
	All Speeds	1.20 \pm 0.40	0.99	1.11 \pm 0.36	0.97	0.958 *	-0.076	-0.323	0.170
Bag	Comfortable	1.10 \pm 0.21	0.97	1.08 \pm 0.24	0.94	0.915 *	-0.008	-0.195	0.178
	Slow	0.83 \pm 0.24	0.96	0.83 \pm 0.19	0.86	0.914 *	0.020	-0.207	0.246
	Fast	1.58 \pm 0.29	0.97	1.50 \pm 0.30	0.92	0.881 *	-0.079	-0.370	0.212
	All Speeds	1.19 \pm 0.40	0.99	1.15 \pm 0.37	0.96	0.953 *	-0.023	-0.274	0.229
Belt	Comfortable	1.10 \pm 0.22	0.97	1.07 \pm 0.18	0.91	0.873 *	-0.327	-0.247	0.182
	Slow	0.83 \pm 0.24	0.98	0.86 \pm 0.18	0.92	0.889 *	0.039	-0.206	0.284
	Fast	1.57 \pm 0.29	0.94	1.34 \pm 0.23	0.86	0.756 *	-0.236	-0.617	0.145
	All Speeds	1.18 \pm 0.40	0.99	1.09 \pm 0.28	0.94	0.914 *	-0.078	-0.447	0.291
Hand	Comfortable	1.11 \pm 0.22	0.98	0.86 \pm 0.17	0.78	0.663 *	-0.255	-0.578	0.068
	Slow	0.81 \pm 0.24	0.93	0.67 \pm 0.19	0.82	0.803 *	-0.097	-0.406	0.212
	Fast	1.60 \pm 0.29	0.96	1.10 \pm 0.29	0.87	0.453 *	-0.495	-1.095	0.106
	All Speeds	1.19 \pm 0.51	0.98	0.89 \pm 0.28	0.91	0.785 *	-0.284	-0.822	0.254
Pocket	Comfortable	1.07 \pm 0.22	0.93	1.11 \pm 0.22	0.90	0.838 *	0.044	-0.204	0.291
	Slow	0.82 \pm 0.22	0.92	0.91 \pm 0.21	0.87	0.783 *	0.101	-0.197	0.400
	Fast	1.57 \pm 0.31	0.95	1.47 \pm 0.26	0.86	0.748 *	-0.101	-0.517	0.315
	All Speeds	1.16 \pm 0.40	0.98	1.17 \pm 0.33	0.94	0.897 *	0.013	-0.355	0.381

Abbreviations: r, Pearson's Correlation; Bias, Mean Difference (Smartphone - GAITrite); LOA, Limits of agreement.
*P < 0.001.

Figura 4.11: Tabla de correlaciones [33].

Tal como se observa en la tabla, la confiabilidad y el ICC son factores clave en la evaluación de la precisión de las mediciones. La confiabilidad se refiere a la consistencia de los resultados obtenidos bajo las mismas condiciones, mientras que el ICC (Coeficiente de Correlación Intraclass) es una medida de la fiabilidad que evalúa la consistencia o reproducibilidad de las mediciones cuantitativas. En este estudio, los resultados muestran que la confiabilidad y el ICC son mayores cuando el teléfono se coloca en el cuerpo o el cinturón, alcanzando valores de ICC superiores a 0.90. Esto indica una excelente consistencia en las mediciones de velocidad de la marcha. Debido a estos altos valores de confiabilidad e ICC, se concluye que estas ubicaciones son las más adecuadas para obtener estimaciones precisas de la velocidad de la marcha.

Es por esto que, en este caso concreto escogeremos la ubicación del cuerpo, no obstante, todas las ubicaciones tienen unos resultados muy buenos cuando se trata de este parámetro. Esta decisión se basa en los resultados del estudio referenciado debido a que para la ubicación del cuerpo, se muestra una muy buena combinación de alta fiabilidad (ICC), baja desviación estándar (SD) y una alta validez (r) con bajo sesgo (Bias).

Finalmente, para la prueba llevada a cabo con el dataset anterior, el resultado de nuestro script sería el siguiente. Es importante tener en cuenta a mayores que

la estimación de esta velocidad depende en gran medida de la longitud de la pierna del paciente (medida desde el maléolo lateral (tobillo) hasta el trocánter mayor (cadera) que sería la prominencia ósea palpable del lateral de la cadera) , por lo que nos interesa mucho recoger este dato de manera correcta. En el caso de mi prueba, se estableció una longitud de pierna de 1 metro.

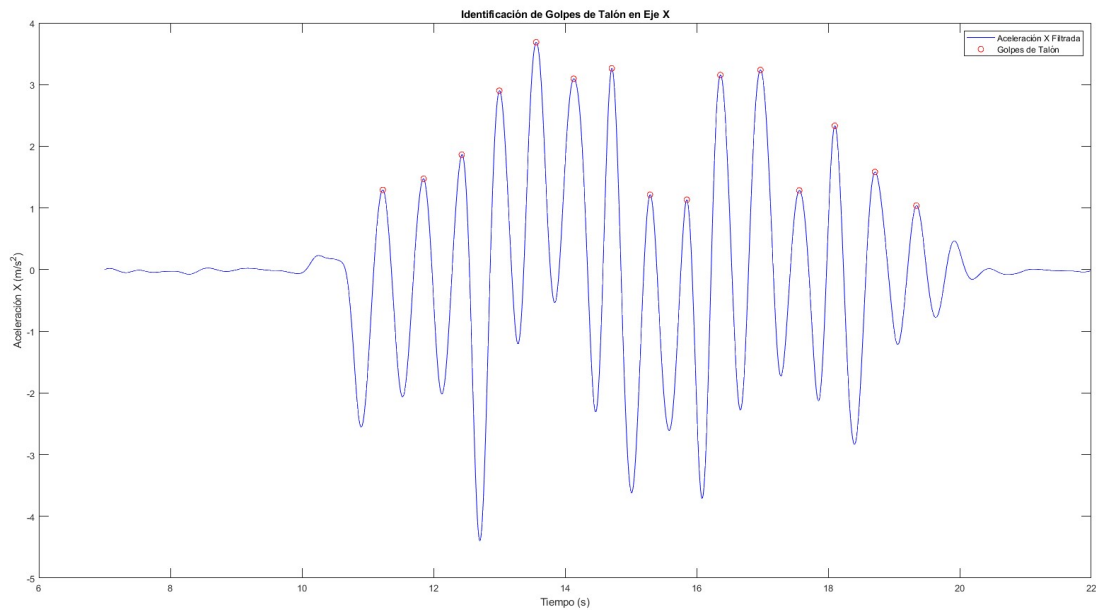


Figura 4.12: Identificación de golpes de talón.

Con esto, y siguiendo todos los pasos comentados anteriormente, se obtiene una velocidad de la marcha promedio de unos 2.11 m/s.

4.4. Desarrollo del Prototipo

4.4.1. Obtención de datos en Kotlin

El código realizado se trata de una aplicación Android desarrollada en Kotlin que registra datos de acelerómetros, guardándolos en un archivo CSV. A continuación, explicaré de manera detallada el funcionamiento de la misma.

La aplicación comienza inicializando los sensores y los elementos de la interfaz de usuario, incluyendo botones y un campo de texto para ingresar la longitud de la pierna del usuario. Los botones permiten iniciar y detener la grabación de datos y calcular la velocidad de la marcha.

Cuando el usuario presiona el botón de inicio, se inicia una cuenta regresiva de 15 segundos. Al finalizar, suena un tono de aviso y comienza la grabación de datos del acelerómetro durante 20 segundos. Estos datos se guardan en un archivo CSV dentro de una carpeta local del dispositivo.

La aplicación utiliza el ‘SensorManager’ para acceder al acelerómetro y registra los datos de aceleración lineal en los ejes X, Y y Z. A diferencia de enfoques anteriores, esta vez se utiliza la aceleración lineal directamente recopilada del sensor en Kotlin, de manera que se desprecia la gravedad en todos los ejes. Estos datos se almacenan junto con el tiempo transcurrido en segundos desde el inicio de la grabación en el archivo CSV creado.

Una vez completada la grabación, surge el problema de tratar de procesar los datos del CSV por medio de un script de MATLAB que nos permitiría aplicar de forma sencilla diversas funciones y finalmente obtener la velocidad de la marcha del paciente. No obstante, los pasos seguidos para realizar esta tarea se comentarán en la siguiente subsección de manera mucho más clara, mencionando así los distintos pasos que se siguieron y las limitaciones que se encontraron a la hora de implementar dichas funcionalidades.

4.4.2. Implementación de Scripts Matlab en el Prototipo

4.4.3. Resultado del Prototipo

Finalmente, después de superar todos los obstáculos técnicos, obtuvimos el siguiente prototipo funcional que permite la recolección y análisis de datos de la marcha. La aplicación desarrollada se encarga de registrar los datos del acelerómetro y guardarlos en un archivo CSV, que posteriormente es procesado por un servidor en Python ejecutando scripts de MATLAB. A continuación, se presentan imágenes de la aplicación antes de iniciar la prueba y al finalizar la misma:

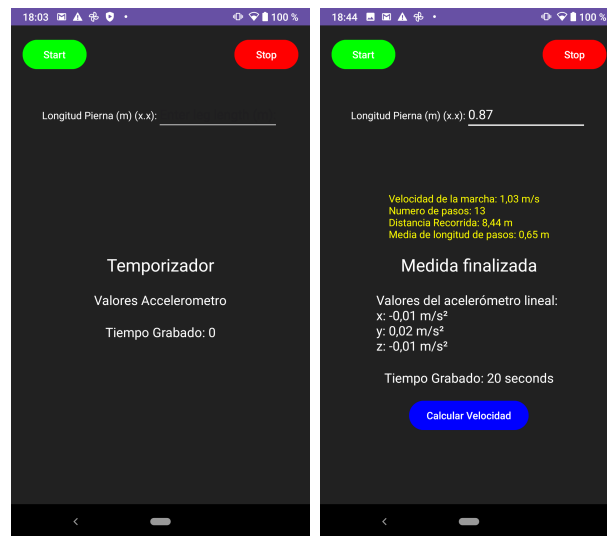


Figura 4.13: Interfaz de la aplicación antes y después de la prueba

La primera imagen muestra la interfaz inicial donde el usuario puede configurar la longitud de la pierna y prepararse para iniciar la grabación. La segunda imagen ilustra la pantalla final, donde se presentan los resultados del análisis, incluyendo la velocidad de la marcha calculada.

Este prototipo demuestra la viabilidad de integrar tecnologías diversas (Kotlin, Python, MATLAB) para desarrollar una solución completa y funcional. La aplicación no solo registra y almacena los datos, sino que también los procesa en tiempo real, proporcionando resultados inmediatos y precisos sobre la velocidad de la marcha del usuario. Este flujo de trabajo integrado resuelve el problema inicial de ejecutar scripts de MATLAB en un dispositivo Android y ofrece una herramienta efectiva para el análisis de la marcha.

4.5. Metodos de análisis

4.5.1. Procedimiento Inicial

En este apartado se explicará el procedimiento mediante el cual se recopilaban los datos para realizar un estudio analítico y comprobar la viabilidad de las predicciones en la clasificación de los datos entre pacientes frágiles y no frágiles. Para ello, en lugar de tomar medidas directamente de distintos pacientes, lo cual llevaría demasiado tiempo, se investigó la existencia de bases de datos que nos pudieran ayudar a realizar un estudio de los datos.

Durante esta exploración, se encontraron diversas bases de datos, como las siguientes [38] [39], que, a pesar de tener varios atributos interesantes, carecían de la etiqueta de fragilidad, por lo que tuvieron que ser descartadas.

Por ello, se continuó con la exploración y finalmente se encontró la base de datos GSTRIDE [37], que contiene 163 muestras de distintos pacientes clasificados entre frágiles y no frágiles, además de muchos otros datos. En nuestro caso, para realizar el análisis de estos datos, seleccionaremos unos pocos atributos de manera que observaremos la correlación entre ellos y determinaremos si existe alguna forma de predecir la fragilidad mediante esta información. Es importante mencionar que, a pesar de ser una base de datos bastante completa, no tenía el atributo de la longitud de la pierna. No obstante, obviaremos dicho atributo y continuaremos con el análisis.

4.5.2. Analisis del DataSet

Una vez tenemos la base de datos seleccionada, escogeremos algunos atributos que serán los que utilizaremos para realizar el estudio. En este caso, se escogieron los atributos de Altura, peso, Edad, Sexo, Índice de masa corporal y velocidad de la marcha para tratar de predecir la clasificación del paciente entre frágil y no frágil.

Es importante mencionar que existen varios grados de fragilidad, y esta se suele dividir en una escala de 5. No obstante, debido al objetivo de este estudio, ya que queremos una herramienta que sea capaz de realizar un cribado inicial sobre pacientes con fragilidad, simplemente clasificaremos los pacientes en frágiles y no frágiles. De esta forma, una vez tenemos el dataset formado con los datos seleccionados comenzaremos a analizarlos en Python.

Para predecir la variable de fragilidad, utilizaremos técnicas de aprendizaje supervisado, ya que contamos con datos etiquetados que nos indican si un paciente es frágil o no. Un problema de aprendizaje supervisado es aquel en el que se dispone de una variable objetivo (en este caso, la fragilidad) que se quiere predecir a partir de una serie de variables predictoras (Edad, Género, Altura, Peso, Índice de Masa Corporal y Velocidad de la Marcha).

A continuación, realizaremos una limpieza y preprocesamiento de los datos, transformando las variables categóricas en numéricas si es necesario y asegurándonos de que todas las variables están en el formato adecuado. Luego, procederemos a dividir el dataset en conjuntos de entrenamiento y prueba, y utilizaremos algoritmos de clasificación para construir modelos que predigan la fragilidad. Evaluaremos estos modelos utilizando métricas de rendimiento adecuadas para asegurar que sean precisos y efectivos en la predicción de la fragilidad en pacientes.

	count	mean	std	min	25 %	50 %	75 %	max
Edad (años)	163.0	82.83	6.44	72.00	77.00	82.00	87.00	97.00
Altura (cm)	163.0	1.57	0.10	1.32	1.50	1.56	1.63	1.84
Peso (Kg)	163.0	64.28	13.11	33.10	55.45	64.00	73.00	105.00
Índice de Masa Corporal	163.0	26.16	5.03	16.46	22.51	25.51	29.33	42.52
Velocidad marcha (m/s)	163.0	0.82	0.33	0.10	0.60	0.80	1.05	2.04

Cuadro 4.3: Resumen estadístico del dataset utilizado para el análisis.

Como se puede observar en la Tabla 4.3, el dataset utilizado es pequeño, con solo 163 muestras. No obstante, como se ha comentado anteriormente, esto será una prueba de concepto para comprobar la viabilidad de implementar este enfoque en futuras mejoras de la aplicación.

4.5.3. Análisis correlación

Una vez analizada la información del DataSet procederemos con el análisis de la correlación de las distintas variables. Para esto, simplemente seleccionaremos las variables cuantitativas, y por medio de nuestro código python obtendremos la siguiente matriz de correlación.

Matriz de correlación de predicción de Fragilidad



Figura 4.14: Matriz de correlación de la predicción

Como podemos observar, ninguna de las correlaciones observadas es demasiado alta, lo que indica que no hay necesidad de eliminar ninguna variable. Esto significa que todas las variables aportan información única al modelo. Podríamos prestar atención a la correlación entre las variables de Peso e Índice de Masa Corporal, sin embargo, esta correlación no supera el umbral de 0.8, por lo que

mantendremos ambas variables en el análisis.

Además, al incluir el atributo de Fragilidad en la matriz de correlación, podemos observar que también tiene una correlación moderada con algunas de las otras variables, lo que sugiere que aporta información relevante al modelo. La inclusión de este atributo es crucial para nuestro análisis, ya que nos permite evaluar directamente cómo las distintas variables están relacionadas con la fragilidad.

Por otro lado, también se realizó un análisis de la distribución de los datos, donde se puede hacer hincapié sobre todo en los datos de la edad que fueron modificados ya que el dataset original tan solo nos daba algunos rangos. Esta modificación fue necesaria para obtener una mejor representación de la variable en nuestro modelo y mejorar la precisión de las predicciones.

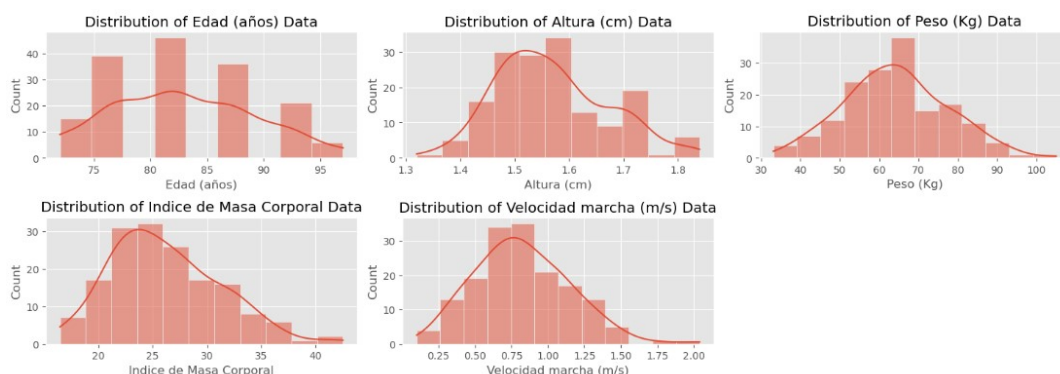


Figura 4.15: Distribución de los datos.

Si bien sería ideal realizar un análisis exhaustivo de los outliers para identificar y manejar datos atípicos que puedan afectar el rendimiento del modelo, en este caso particular hemos decidido no hacerlo debido a la baja cantidad de datos disponibles. El análisis de outliers en conjuntos de datos pequeños puede resultar en la eliminación de puntos de datos críticos, lo que podría llevar a una pérdida significativa de información y afectar negativamente la capacidad del modelo para generalizar y aprender de los datos disponibles. Por lo tanto, mantendremos todos los datos en el conjunto para asegurar que el modelo tenga suficiente información para aprender patrones útiles.

Por otro lado, en el análisis, se verificó también que el conjunto de datos no contiene valores nulos, por lo que no es necesario realizar imputaciones o eliminar filas y columnas debido a valores faltantes. Esto nos permite trabajar directamente con todos los datos disponibles para el análisis y modelado.

Finalmente, se gestionó además la presencia de variables categóricas. En este caso, solo disponíamos de la variable género, que es una variable nominal con dos posibles valores, por lo que se procedió con una conversión a variables binarias, siendo en este caso el género masculino un 1 y el género femenino un 0.

4.5.4. Construcción de Modelo y predicción

En este caso, tal y como se comenta en apartados anteriores, al tratarse de un problema de clasificación, deberemos seleccionar un método para comenzar a realizar las predicciones, una vez tenemos los datos procesados.

En este caso, se escogió el algoritmo KNN (K-Nearest Neighbour), que es un algoritmo de clasificación. En este tipo de algoritmos, es muy importante evitar fenómenos como el underfitting o el overfitting sobre el conjunto de datos con el que se entrena. Para ello, se deben ajustar los hiperparámetros del modelo de manera adecuada y asegurarse de que se utiliza una cantidad suficiente de datos para entrenar el modelo.

Además del KNN, se implementaron otros dos modelos para comparar su rendimiento y determinar cuál es el más adecuado para predecir la fragilidad en los pacientes: el modelo de Árbol de Decisión y el modelo Naive Bayes de Bernoulli. Estos modelos se desarrollaron y evaluaron en el archivo Jupyter disponible en mi repositorio [36].

4.5.5. Potencial de la Implementación de Algoritmos de Predicción

Implementar un algoritmo de predicción en la aplicación de medición de la fragilidad tiene un gran potencial para mejorar la precisión y utilidad de la herramienta. Sin embargo, es importante destacar que actualmente disponemos de una cantidad muy limitada de datos, lo cual restringe la capacidad del modelo para generalizar y hacer predicciones precisas. Además, el análisis de los datos se realizará de forma asincrónica, lo que podría introducir ciertos retrasos en el procesamiento. A pesar de estas limitaciones, si tuviéramos acceso a una base de datos más amplia, la implementación de algoritmos de predicción sería mucho más sencilla y permitiría realizar predicciones más acertadas.

Por ejemplo, la inclusión de datos como la fuerza de prensión, que no se puede medir actualmente debido a la falta de un dinamómetro, podría mejorar significativamente la precisión del modelo. Numerosos estudios han demostrado

que la fuerza de prensión está fuertemente correlacionada con la fragilidad en personas mayores. Incorporar este tipo de mediciones, junto con una base de datos más completa y robusta, permitiría a la aplicación ofrecer evaluaciones de fragilidad mucho más precisas y útiles para el diagnóstico y seguimiento de los pacientes.

4.6. Protocolo Experimental

4.6.1. Procedimiento general

Para medir la velocidad de la marcha, se recomienda seguir un protocolo estandarizado para garantizar la precisión y la fiabilidad de las mediciones [7]. Las precauciones y el procedimiento estándar a seguir son los siguientes:

Precauciones

- Realizar la medición en un entorno bien iluminado, preferiblemente con luz artificial.
- Usar una habitación cerrada y silenciosa, sin interferencias visuales o auditivas.
- Asegurarse de que los participantes lleven ropa cómoda y zapatos abrochados con menos de 3 cm de altura, usados de manera consistente para cada medición.
- Utilizar un cinturón de seguridad para prevenir caídas durante la medida.

Procedimiento estándar

- Medir la velocidad de la marcha en diferentes pasos (lento, normal y rápido).
- Comenzar a caminar 2 metros antes y parar 2 metros después de la medición.
- Mantener la misma dirección de caminata durante la medición.
- Dar instrucciones claras, por ejemplo: 'empiece a caminar a su paso normal'.
- Permitir el uso de ayudas auditivas o visuales según sea necesario, asegurando consistencia en las mediciones (ya sea con o sin prótesis).
- Evaluar la variabilidad de las medidas a lo largo de al menos 6 ciclos de medición (3 trayectos).

4.6.2. Protocolo de uso de la aplicación

Básicamente, la idea consiste en pulsar un botón, que activará un temporizador de unos 10 o 15 segundos, durante los cuales el profesional tendrá tiempo suficiente para colocar el teléfono en la posición establecida. Una vez colocado el dispositivo, y aunque no se haya escuchado todavía el pitido, se podrá comenzar con la marcha. La prueba establecida sería la de medir la velocidad de la marcha para unos 4 metros. No obstante, en este caso, simplemente caminaremos hasta que se escuche de nuevo el pitido final. De esta forma, una vez recogido el dispositivo de la riñonera en la espalda del paciente, pulsaremos el botón de 'Calcular Velocidad', con el cual se enviará una petición al servidor para obtener la estimación de la velocidad de la marcha y poder realizar un diagnóstico.

En este caso, el procedimiento estándar comentado en el artículo [7] menciona que sería interesante comenzar a caminar 2 metros antes de la medición y parar 2 metros después. Esta es una norma interesante a la hora de realizar las mediciones, ya que permite al paciente tener una zona de aceleración y otra de deceleración. Sería interesante, por lo tanto, añadir un temporizador más a nuestra aplicación, de forma que después del primer temporizador de 10 o 15 segundos, se establezca otro de unos 5 segundos para acelerar.

Además, dependiendo del espacio disponible, se podría añadir también un campo de entrada para introducir el tiempo de grabación de los datos, de forma que todo esto fuera personalizado. En este caso, no se implementarán estas mejoras, ya que el objetivo de la investigación consiste en digitalizar la medición de la fragilidad y analizar sus datos. No obstante, como se comentó anteriormente, sería muy interesante incluir parámetros más personalizados para cada centro médico. Aquí tenemos a mayores un video donde se muestra el funcionamiento del prototipo de manera detallada [40].

Capítulo 5

Pruebas

5.1. Objetivo

El objetivo del plan de pruebas es verificar la funcionalidad y corrección global del prototipo de la aplicación que mide la fragilidad y realiza predicciones mediante técnicas de IA. Las pruebas se dividirán en dos fases: la primera para mostrar el correcto funcionamiento de la aplicación y la segunda para demostrar la viabilidad de realizar predicciones asincrónicas mediante el algoritmo KNN.

5.2. Plan de Pruebas

5.2.1. Prueba de Funcionalidad de la Aplicación

En esta primera fase, se realizará una prueba del correcto funcionamiento del prototipo, mostrando cómo se debe realizar una medición y cómo se calcula la velocidad de la marcha conectándonos al servidor en Python.

Primero, conectamos el dispositivo y colocamos la riñonera en la posición (A) de la imagen 4.3, encima de la tercera vértebra, ajustándola perfectamente al cuerpo del paciente. El analizado se coloca en espera con los pies juntos. Luego, le damos a 'Start' y ponemos el teléfono en la riñonera. Cuando suene el primer pitido, el paciente comenzará a caminar. Al completar la distancia disponible, el paciente se queda quieto. Al sonar de nuevo el pitido, puede retirarse la riñonera y el teléfono.

A continuación, medimos la pierna del paciente desde el trocánter mayor hasta el maléolo del tobillo, introducimos la medida en metros y pulsamos el botón de 'calcular velocidad'. Es importante tener en cuenta que al pulsar dicho botón, debe estar iniciado el servidor python, de forma que se procese dicha señal por medio del script de MatLab y nos devuelva el resultado deseado.

5.2.2. Prueba de Viabilidad de Predicciones Asíncronas

En esta fase, se prueba la viabilidad de predicciones asincrónicas usando el algoritmo de clasificación con la base de datos GSSTRIDE de 163 muestras de pacientes con medidas de marcha como velocidad, longitud de paso, y cadencia. Se seleccionan los atributos clave: Edad, Altura, Género, Peso, Índice de masa corporal, Velocidad de la marcha. Los datos están etiquetados como frágiles o no frágiles según umbrales de velocidad de marcha, pero la base tiene pocas muestras, lo que afecta la credibilidad de las conclusiones.

Análisis de Resultados

Para el análisis de los resultados obtenidos con los diferentes modelos, se han utilizado métricas como 'accuracy', 'precision' y 'recall'. Cada una de ellas tiene sus características y nos ofrece una perspectiva diferente del rendimiento del modelo. Por ejemplo, el 'accuracy' calcula la precisión total del modelo, considerando tanto los verdaderos positivos como los verdaderos negativos. Es una medida global que nos dice qué proporción de todas las predicciones realizadas por el modelo son correctas.

La 'precision', por otro lado, se enfoca solo en los verdaderos positivos, mostrándonos la proporción de instancias clasificadas como positivas que son realmente positivas. Es una métrica crucial cuando el costo de los falsos positivos es alto. Finalmente, el 'recall', también conocido como sensibilidad, nos indica qué tan bien predice nuestro modelo los verdaderos positivos, es decir, la proporción de instancias positivas que son correctamente identificadas por el modelo.

Para evaluar el rendimiento de estos modelos, se utilizaron estas métricas de precisión según se explica en detalle en el Jupyter Notebook. Estas métricas permiten determinar la efectividad de los modelos en la clasificación correcta de los pacientes entre frágiles y no frágiles, proporcionando una visión clara de su desempeño y utilidad para esta aplicación.

Además, se empleó también la validación cruzada Leave-One-Out (LOOCV) para maximizar la robustez de la evaluación de los modelos. En la LOOCV, cada instancia del conjunto de datos se utiliza exactamente una vez como conjunto de prueba, mientras que el resto se utiliza para entrenamiento. Esto se realizó 163 veces, una por cada muestra en nuestro dataset, asegurando una evaluación exhaustiva y precisa del rendimiento de cada modelo.

Modelo	Accuracy	Precision	Recall	Precision LOOCV
KNN	0.76	1.00	0.14	0.6810
BernoulliNB	0.71	0.00	0.00	0.7546
Árbol de Decisión	0.61	0.27	0.21	0.6074

Cuadro 5.1: Métricas de evaluación para los diferentes modelos.

Matrices de Confusión		
KNN	BernoulliNB	Árbol de Decisión
$\begin{bmatrix} 35 & 0 \\ 12 & 2 \end{bmatrix}$	$\begin{bmatrix} 35 & 0 \\ 14 & 0 \end{bmatrix}$	$\begin{bmatrix} 27 & 8 \\ 11 & 3 \end{bmatrix}$

Cuadro 5.2: Matrices de confusión para los diferentes modelos.

En el caso del dataset disponible, el modelo de KNN es el que mejores resultados nos ha devuelto debido a que tenía una precisión de 0.75. No obstante, a pesar de estos resultados iniciales favorables para KNN, es importante destacar que la validación cruzada Leave-One-Out (LOOCV) muestra que BernoulliNB tiene una precisión más alta (0.7546) en comparación con KNN (0.6810). Como ya se comentó anteriormente, la LOOCV evalúa cada muestra individualmente como datos de prueba, lo que ayuda a mitigar el sesgo que podría surgir de la partición específica de los datos en conjuntos de entrenamiento y prueba. Por lo tanto, deberíamos elegir la LOOCV para obtener una evaluación más objetiva y fiable de cómo generalizan los modelos a nuevos datos. Por otro lado, al generar la matriz de confusión, observamos que el modelo BNB tiene 35 verdaderos positivos con 14 falsos positivos y ningún falso negativo.

Esto es crucial para la aplicación, ya que garantiza que todos los pacientes que necesitan atención sean identificados adecuadamente. Aunque el número de falsos positivos es elevado, lo que indica que muchos pacientes no frágiles son clasificados incorrectamente como frágiles, el costo de estos es aceptable dado el beneficio de no perder a ningún paciente frágil. A pesar de estos resultados, es evidente que la matriz de confusión no se asemeja a una matriz diagonal, lo que indica que el modelo tiene margen de mejora. Se necesitaría un análisis más profundo y posiblemente ajustes adicionales al modelo o más datos para reducir el número de falsos positivos y mejorar la precisión general.

5.3. Conclusiones de las Pruebas

Las pruebas realizadas demostraron la viabilidad y precisión del prototipo para medir la velocidad de la marcha y detectar patrones de marcha específicos,

como los golpes de talón. Además, el análisis de los modelos de predicción mostró la capacidad de la herramienta para predecir la fragilidad en los participantes con una precisión considerable. Aunque el conjunto de datos utilizado en las pruebas era limitado en tamaño, los resultados obtenidos sugieren que la integración de diversas tecnologías, como Kotlin, Python y MATLAB, proporciona una solución robusta y efectiva para el análisis de la marcha y la predicción de la fragilidad en personas mayores. Estos hallazgos respaldan la continuación del desarrollo y refinamiento del prototipo, con el objetivo de ofrecer una herramienta útil y precisa para la evaluación de la fragilidad y la detección temprana de problemas de movilidad en la población objetivo.

Capítulo 6

Discusión de los resultados

Los resultados obtenidos en este estudio han permitido comprobar la hipótesis inicial de que los dispositivos móviles o wearables pueden proporcionar una estimación precisa de la fragilidad física en personas mayores, comparable a las evaluaciones clínicas tradicionales. Estos hallazgos respaldan la viabilidad del uso de tecnologías accesibles y de bajo costo para la detección temprana de fragilidad, lo que podría facilitar enormemente la implementación en entornos clínicos y comunitarios para la realización de un cribado.

6.1. Comparativa con el estado del arte

En comparación con las soluciones descritas en el estado del conocimiento, como el sistema GAITRite o la aplicación PowerFrail, nuestro prototipo ofrece varias ventajas significativas. El sistema GAITRite, aunque es extremadamente preciso, tiene un costo elevado (entre \$10,000 y \$15,000), lo que limita enormemente su accesibilidad. Además, tal y como se respalda en el estudio [33], el uso del acelerómetro del smartphone aplicando diversos filtros puede resultar en una precisión bastante similar. Por otro lado, la aplicación PowerFrail, que evalúa la fragilidad mediante la grabación de video, depende de la calidad de la cámara del smartphone utilizado y puede no ser tan precisa en condiciones de iluminación variables. Este prototipo, al utilizar directamente los datos del acelerómetro del dispositivo móvil, proporciona una solución más económica y consistente para medir la velocidad de la marcha y detectar la fragilidad.

6.2. Mejoras sobre Soluciones Anteriores

El trabajo presentado en este estudio destaca por su mejora significativa en la predicción de fragilidad en personas mayores, un aspecto que no ha sido abordado de manera exhaustiva en otros estudios. Al emplear sensores integrados en dispositivos móviles comunes, se reduce significativamente el costo y se aumenta la

accesibilidad de la tecnología para la detección temprana de fragilidad. Además, a diferencia de los métodos manuales y los sistemas de grabación de video utilizados en soluciones anteriores, nuestro prototipo es menos invasivo y puede ser utilizado en una gran variedad de entornos sin requerir equipos adicionales costosos. Estas mejoras representan un avance significativo en la detección y prevención de la fragilidad en la población mayor, proporcionando una herramienta efectiva y económica para su evaluación y seguimiento en entornos clínicos y comunitarios.

Capítulo 7

Conclusiones y Posibles ampliaciones

7.0.1. Principales aportaciones

Este trabajo presenta una prueba de concepto sobre la viabilidad de desarrollar un prototipo innovador para la medición de la fragilidad física en personas mayores utilizando dispositivos móviles o wearables junto con técnicas de inteligencia artificial. Las principales aportaciones incluyen la creación de una aplicación que permite monitorizar y analizar en tiempo real la velocidad de la marcha, la cual es un indicador clave de fragilidad.

El prototipo utiliza una metodología de análisis de los parámetros del acelerómetro para calcular la velocidad de la marcha, respaldando su precisión y viabilidad en diversos estudios mencionados a lo largo de la memoria. Estos estudios demuestran el potencial de los acelerómetros de distintos teléfonos móviles.

Además, esta implementación ofrece una solución más sencilla de utilizar para los profesionales en contraposición a los métodos tradicionales, donde simplemente se recorría una distancia midiendo el tiempo y se calculaba la velocidad. En el método tradicional, esto se hacía de forma manual con cronómetros y medidas tomadas en lápiz y papel, lo que introducía errores de factor humano y subjetivos. Este prototipo automatizado mejora todas esas limitaciones, ofreciendo una solución más accesible y económica en comparación con otros métodos, como el ya mencionado GAITRite. Finalmente, su interfaz intuitiva facilita la adopción en diversos entornos clínicos.

Es importante destacar que este prototipo demuestra la viabilidad de integrar tecnologías diversas (Kotlin, Python, MATLAB) para desarrollar una solución completa y funcional. En cuanto a los criterios descritos en la subsección 4.1.1, se ha demostrado que este prototipo los cumple:

- Disponibilidad de un acelerómetro: El prototipo utiliza acelerómetros disponibles en dispositivos móviles comunes.
- Facilidad de uso para usuarios mayores: La interfaz intuitiva y simplificada asegura que los usuarios mayores puedan utilizar el dispositivo sin dificultades.
- Compatibilidad con herramientas de desarrollo accesibles: El desarrollo en Kotlin, Python y MATLAB muestra la compatibilidad con herramientas ampliamente accesibles y usadas.
- Durabilidad de la batería: El uso eficiente del acelerómetro y el procesamiento optimizado garantizan una buena durabilidad de la batería del dispositivo.
- Precio del dispositivo: Al utilizar teléfonos móviles comunes, el costo del dispositivo es significativamente menor en comparación con otras soluciones específicas del mercado.

7.0.2. Limitaciones y problemas encontrados

Durante el desarrollo de este trabajo se han realizado diversas suposiciones. En primer lugar, se supone que la metodología de análisis aplicada a las señales recopiladas por el acelerómetro nos da resultados similares a los obtenidos en el estudio de [33]. Además, existieron diversas limitaciones durante la investigación, como por ejemplo el presupuesto disponible para la obtención de diversos materiales necesarios para realizarlo.

Sin ir más lejos, podemos revisar el apartado donde se realiza la experimentación preliminar con el 'Fitbit Alta HR'. Al finalizar dicha experimentación se concluye que existen modelos superiores de relojes de esta marca que sí son capaces de devolver datos del acelerómetro en tiempo real; sin embargo, al disponer de un presupuesto limitado, no se pudo experimentar con otro tipo de wearables que resultaban mucho más atractivos y prometedores para el desarrollo del prototipo y que además nos facilitaban 'APIs' o 'SDKs' oficiales para acceder a mucha información útil. Es por esto que se tuvo que cambiar el enfoque del trabajo, orientándolo más hacia la utilización de teléfonos móviles para el desarrollo del prototipo.

En cuanto a la implementación de técnicas de inteligencia artificial para la creación de algoritmos que nos permitiesen mejorar las predicciones de fragilidad, una de las principales limitaciones identificadas es la disponibilidad de datos etiquetados. Aunque se han utilizado conjuntos de datos como el GSSTRIDE, que proporciona información relevante sobre la marcha de los pacientes mayores,

la cantidad de datos etiquetados sigue siendo limitada. Esto puede afectar la capacidad del modelo para generalizar y predecir con precisión la fragilidad en diferentes contextos y poblaciones. Además, la falta de datos sobre variables importantes, como la fuerza de prensión, también puede limitar la capacidad del modelo para capturar la complejidad de la fragilidad.

7.0.3. Líneas Futuras

Para futuras investigaciones, se proponen varias vías de mejora. En primer lugar, sería beneficioso realizar pruebas con muestras más grandes y diversas para validar los resultados obtenidos y garantizar su generalización a una población más amplia. Además, la integración de algoritmos de inteligencia artificial más avanzados y el uso de técnicas de aprendizaje profundo podrían mejorar aún más la precisión y la fiabilidad de las mediciones. Esto implicaría la incorporación de un servicio de análisis de datos integrado en la aplicación y que se realice en tiempo real, en lugar de de forma asincrónica como se hizo en este trabajo. Otra vía de mejora incluye la exploración de otros indicadores de fragilidad, además de la velocidad de la marcha, para proporcionar una evaluación más completa de la condición física de las personas mayores.

Además, como se mencionó anteriormente, se recomienda la recopilación de conjuntos de datos más grandes y completos, que incluyan mediciones adicionales como la fuerza de prensión. Esto permitiría entrenar modelos más robustos y mejorar las predicciones de fragilidad. Además, la inclusión de datos de múltiples fuentes, como registros médicos electrónicos y datos de actividad física diaria, podría enriquecer aún más el conjunto de datos y mejorar la capacidad del modelo para capturar la variabilidad en la fragilidad entre los individuos. Es decir, en general, con conjuntos de datos más grandes y completos, y con datos adicionales relevantes, seguramente se mejorarán las predicciones de fragilidad y se aumentará la utilidad clínica del prototipo desarrollado.

Apéndice A

Anexo: Código en Kotlin

A.1. Código en Python

A.1.1. Código conexión Fitbit Simple

```
import requests
import fitbit

access_token = 'XXXXXX'

header = {'Authorization': 'Bearer {}'.format(access_token)}
response = requests.get('https://api.fitbit.com/1/user/-/profile.json', headers=header).json()

print(response['user'])
```

A.1.2. Código conexión Fitbit PCKE

```
from fitbit import Fitbit
import requests

CLIENT_ID = '23RSM9'
CLIENT_SECRET = 'XXXX'
ACCESS_TOKEN = 'XXXX'
REFRESH_TOKEN = 'XXX'

auth_client = Fitbit(CLIENT_ID, CLIENT_SECRET,
    oauth2=True, access_token=ACCESS_TOKEN,
    refresh_token=REFRESH_TOKEN)
```

```
profile_data = auth_client.user_profile_get()

activity_data = auth_client.activities()

print(profile_data)
```

A.2. Código en Kotlin

A.2.1. Código Eliminación ruido

```
override fun onSensorChanged(event: SensorEvent?) {
    if (event?.sensor?.type == Sensor.TYPE_ACCELEROMETER) {
        val accelX = event.values[0]
        val accelY = event.values[1]
        val accelZ = event.values[2]

        velocityTextView.text = "Valores del acelerometro:
            \nx: ${ "%.2f".format(accelX)} m/s\ny: ${ "%.2f".
                format(accelY)} m/s\nz: ${ "%.2f".format(accelZ)}
                m/s"

        speedTextView.text = "Magnitud aceleracion: %.2f m/
            s".format(speed)
    }
}
```

A.2.2. Código Filtro paso bajo

```
fun lowPassFilter(input: FloatArray, output: FloatArray):
    FloatArray {
    val alpha = 0.50f
    for (i in input.indices) {
        output[i] = alpha * input[i] + (1 - alpha) * output
            [i]
    }
    return output
}
```

A.2.3. Código Invocación Filtro


```
gravity[0] = lowPassFilter(event.values[0], gravity[0])
gravity[1] = lowPassFilter(event.values[1], gravity[1])
gravity[2] = lowPassFilter(event.values[2], gravity[2])
```

```
linearAcceleration[0] = event.values[0] - gravity[0]
linearAcceleration[1] = event.values[1] - gravity[1]
linearAcceleration[2] = event.values[2] - gravity[2]
```

A.2.4. Código Magnitud Aceleración

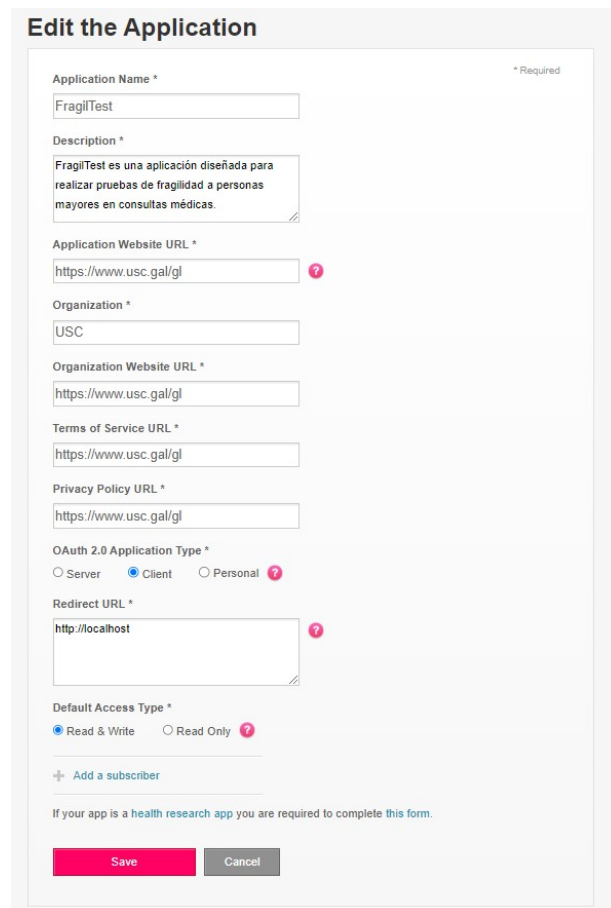
```
val MagAccel = sqrt(linearAcceleration[0] *
    linearAcceleration[0]
    + linearAcceleration[1] * linearAcceleration[1] +
    linearAcceleration[2] * linearAcceleration[2])
```


Apéndice B

Anexo: Imágenes Adicionales

B.1. Imágenes Adicionales Fitbit

B.1.1. Creación Aplicación



The screenshot shows the 'Edit the Application' form in the Google Cloud console. The form is titled 'Edit the Application' and includes a '* Required' label. The fields and their values are as follows:

- Application Name ***: FragilTest
- Description ***: FragilTest es una aplicación diseñada para realizar pruebas de fragilidad a personas mayores en consultas médicas.
- Application Website URL ***: https://www.usc.gal/gl
- Organization ***: USC
- Organization Website URL ***: https://www.usc.gal/gl
- Terms of Service URL ***: https://www.usc.gal/gl
- Privacy Policy URL ***: https://www.usc.gal/gl
- OAuth 2.0 Application Type ***: Client (selected), Server, Personal
- Redirect URL ***: http://localhost
- Default Access Type ***: Read & Write (selected), Read Only

At the bottom of the form, there is a link to 'Add a subscriber' and a note: 'If your app is a health research app you are required to complete this form.' The form has 'Save' and 'Cancel' buttons at the bottom.

Figura B.1: Creación aplicación.

B.1.2. Datos para la creación de la URL de la app

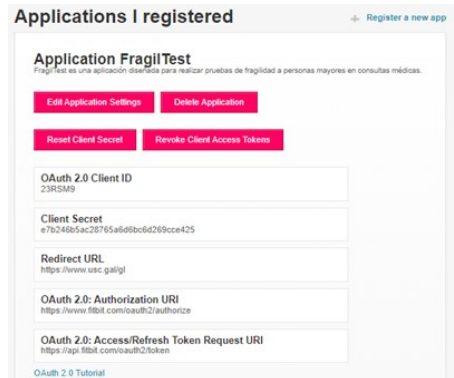


Figura B.2: Datos para la creación de la URL de la aplicación.

B.1.3. Formulario de Permisos

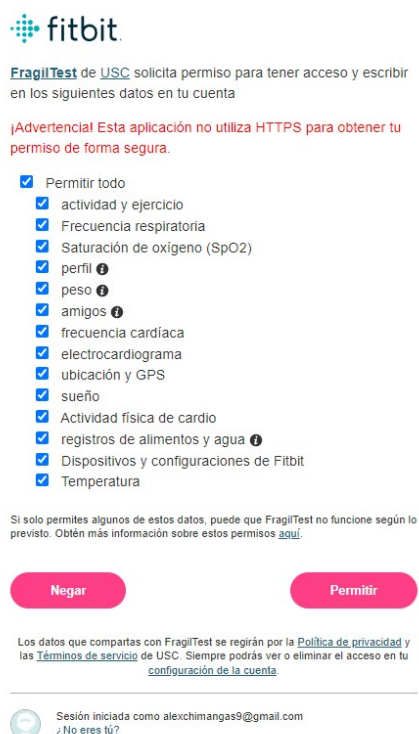


Figura B.3: Formulario de permisos de la aplicación.

B.1.4. Foro Fitbit Aclaración

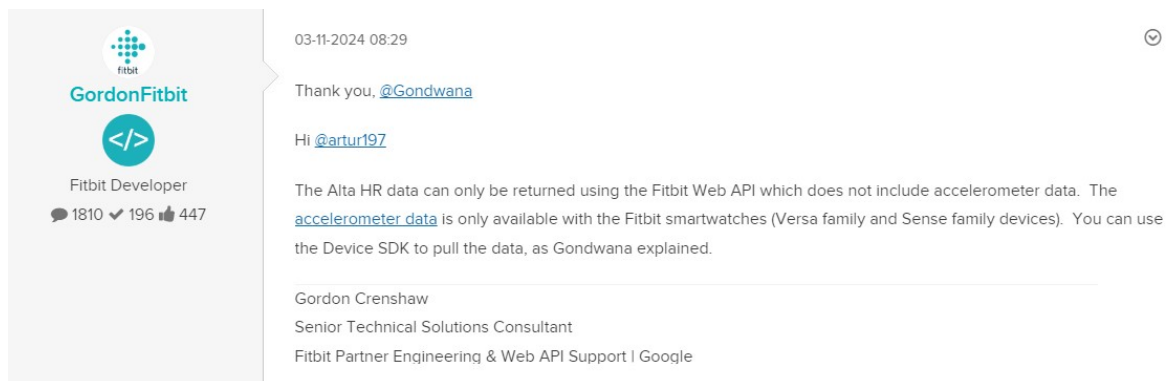


Figura B.4: Aclaración acerca del acceso a la información del acelerómetro en el dispositivo Alta HR

Bibliografía

- [1] Fried, L. P., Ferrucci, L., Darer, J., Williamson, J. D., & Anderson, G. F. (2004). Untangling the Concepts of Disability, Frailty, and Comorbidity: Implications for Improved Targeting and Care. *The Journals Of Gerontology: Series A*, **59**(3), M255-M263. <https://doi.org/10.1093/gerona/59.3.m255>
- [2] Abizanda Soler, P., Espinosa Almendro, J., Juaréz Vela, R., et al. Actualización del documento de consenso sobre prevención de la fragilidad en la persona mayor (2022) Estrategia de Promoción de la Salud y Prevención en el SNS. Madrid: Ministerio de Sanidad, Servicios Sociales e Igualdad. https://www.sanidad.gob.es/areas/promocionPrevencion/envejecimientoSaludable/fragilidadCaidas/docs/ActualizacionDoc_FragilidadCaidas_personamayor.pdf
- [3] Ferrer, A., Formiga, F., Sanz, H., Monserrate, E., & Verges, D. (2014). Envejecimiento satisfactorio e indicadores de fragilidad en los mayores de la comunidad. *Estudio Octabaix. Atención Primaria*, **46**(9), 475-482. <https://doi.org/10.1016/j.aprim.2014.01.004>
- [4] Shamliyan, T. A., Talley, K. M., Ramakrishnan, R., & Kane, R. L. (2013). Association of frailty with survival: A systematic literature review. *Ageing Research Reviews*, **12**(2), 719-736. <https://doi.org/10.1016/j.arr.2012.03.001>
- [5] Peel, N. M., Kuys, S., & Klein, K. (2012). Gait Speed as a Measure in Geriatric Assessment in Clinical Settings: A Systematic Review. *The Journals Of Gerontology: Series A*, **68**(1), 39-46. <https://doi.org/10.1093/gerona/gls174>
- [6] Figueroa, J. F., Maddox, K. E. J., Beaulieu, N., Wild, R. C., & Jha, A. K. (2017). Concentration of Potentially Preventable Spending Among High-Cost Medicare Subpopulations. *Annals of Internal Medicine*, **167**(10), 706. <https://doi.org/10.7326/M17-0767>

- [7] Paz Valiñas, L., Faraldo Vallés, M. J., & Bugarín González, R. (2019). Empleo de la velocidad de la marcha como indicador de fragilidad. Recuperado de <https://hdl.handle.net/20.500.11940/13762>
- [8] J. Gong, M. D. Goldman, and J. Lach, “Deepmotion: a deep convolutional neural network on inertial body sensors for gait assessment in multiple sclerosis.” in *Proceedings of Wireless Health*, 2016.
- [9] M. Gadaleta and M. Rossi, “Idnet: Smartphone-based gait recognition with convolutional neural networks,” *Pattern Recognition*, vol. 74, pp. 25–37, 2018.
- [10] J. Hannink, T. Kautz, C. F. Pasluosta, J. Barth, S. Schulein, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier, “Stride length estimation with deep learning,” *arXiv preprint arXiv:1609.03321*, 2016.
- [11] F. J. Ordoñez and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [12] Shahr, R. T., & Agmon, M. (2021). Gait Analysis Using Accelerometry Data from a Single Smartphone: Agreement and Consistency between a Smartphone Application and Gold-Standard Gait Analysis System. *Sensors*, **21**(21), 7497. <https://doi.org/10.3390/s21227497>
- [13] Van Kan, G. A., Rolland, Y., Andrieu, S., Bauer, J., Beauchet, O., Bonnefoy, M., Cesari, M., Donini, L. M., Gillette-Guyonnet, S., Inzitari, M., Nourhashemi, F., Onder, G., Ritz, P., Salvà, A., De Visser, M., & Vellas, B. (2009). Gait speed at usual pace as a predictor of adverse outcomes in community-dwelling older people an International Academy on Nutrition and Aging (IANA) Task Force. *The Journal of Nutrition, Health & Aging/The Journal of Nutrition, Health and Aging*, 13(10), 881-889. <https://doi.org/10.1007/s12603-009-0246-z>
- [14] GAITRITE — GAIT Analysis Software. (2024, abril 2). *Gaitrite*. Recuperado de <https://www.gaitrite.com/gait-analysis-software>
- [15] O’Sullivan, S. B., & Schmitz, T. J. (2007). *Physical Rehabilitation*. In Chapter 32: Gait Assessment and Training. Philadelphia, PA: F.A. Davis. Información detallada del precio recuperada de <https://www.sciencedirect.com/topics/nursing-and-health-professions/gait-analysis-system>
- [16] Arcuri, G. (2023). *PowerFrail: uso clínico para evaluar índice de fragilidad y potencia muscular en población general en la ciudad de Rosario, Santa Fe* (Bachelor’s thesis). Recuperado de

- <https://rid.ugr.edu.ar/bitstream/handle/20.500.14125/797/Inv.%20D-591%20tesina.pdf?sequence=1&isAllowed=y>
- [17] Bock, J. O., König, H. H., Brenner, H., Haefeli, W. E., Quinzler, R., Matschinger, H., et al. (2016). Associations of frailty with health care costs—results of the ESTHER cohort study. *BMC Health Services Research*, **16**, 128. <https://doi.org/10.1186/s12913-016-1386-x>. PubMed PMID: 27074800. Recuperado de <https://pubmed.ncbi.nlm.nih.gov/27074800/>
- [18] GSMArena. (2024, abril 12). *Nokia 6.1 - Full phone specifications*. Recuperado de https://www.gsmarena.com/nokia_6_1-8972.php
- [19] Xiaomi Watch. (2024, marzo 5). *Development Tools*. Recuperado de <https://xiaomiwatch.70mai.com.cn/en/>
- [20] Sreejith Sasidharan. (2024, marzo 6). *GitHub - 4lhc/MiBand_HRX: Library to work with Xiaomi MiBand HRX Accelerometer Data*. Recuperado de https://github.com/4lhc/MiBand_HRX
- [21] Fitbit Development. (2024, marzo 6). *Getting started*. Recuperado de <https://dev.fitbit.com/build/reference/web-api/developer-guide/getting-started/>
- [22] Fitbit Development. (2024, marzo 7). *Accelerometer Sensor Guide*. Recuperado de <https://dev.fitbit.com/build/guides/sensors/accelerometer/>
- [23] Fitbit Development. (2024, marzo 7). *OAuth 2.0 Tutorial*. Recuperado de <https://dev.fitbit.com/build/reference/web-api/troubleshooting-guide/oauth2-tutorial/>
- [24] Swagger UI. (2024, marzo 8). Recuperado de <https://dev.fitbit.com/build/reference/web-api/explore/>
- [25] Overview — Python-Fitbit 0.3.0 documentation. (2024, marzo 8). Recuperado de <https://python-fitbit.readthedocs.io/en/latest/>
- [26] The Fitbit Community. (2024, marzo 11). Recuperado de <https://community.fitbit.com/t5/Web-API-Development/Real-time-Data-Export-ALTA-HR/td-p/5559559>
- [27] Grouios, G., Ziagkas, E., Loukovitis, A., Chatzinikolaou, K., & Koidou, E. (2023). Accelerometers in Our Pocket: Does Smartphone Accelerometer Technology Provide Accurate Data?. *Sensors*, *23*(192). <https://doi.org/10.3390/s23010192>

- [28] Nishiguchi, S., Yamada, M., Nagai, K., Mori, S., Kajiwar, Y., Sonoda, T., ... & Aoyama, T. (2012). Reliability and Validity of Gait Analysis by Android-Based Smartphone. *Telemedicine and e-Health*, 18(4), 292-296. <https://doi.org/10.1089/tmj.2011.0132>
- [29] P. Siirtola and J. Roning. Recognizing human activities user-independently on smartphones based on accelerometer data. *IJIMAI*, 1(5):38-45, 2012.
- [30] Altini, M., Vullers, R., Van Hoof, C., van Dort, M., & Amft, O. (2014, March). Self-calibration of walking speed estimations using smartphone sensors. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)* (pp. 10-18). IEEE.
- [31] Ustun, I., & Cetin, M. (2019). Speed estimation using smartphone accelerometer data. *Transportation Research Record*, 2673(3), 65-73. Disponible en: http://www.tri-odu.org/uploads/3/9/2/6/39268609/trb_2019_speed_estimation_v5.pdf
- [32] A. Shrestha and M. Won, "DeepWalking: Enabling Smartphone-based Walking Speed Estimation Using Deep Learning," in *2018 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, 2018, pp. 10-18. IEEE.
- [33] Silsupadol, P., Teja, K., & Lugade, V. (2017). Reliability and validity of a smartphone-based assessment of gait parameters across walking speed and smartphone locations: Body, bag, belt, hand, and pocket. *Gait & Posture*, 58, 516-522.
- [34] McCamley, J., Donati, M., Grimpampi, E., & Mazzà, C. (2012). An enhanced estimate of initial contact and final contact instants of time using lower trunk inertial sensor data. *Gait & posture*, 36(2), 316-318.
- [35] MathWorks. (2020, febrero 18). Implementing code from MATLAB into Android Studio? MATLAB Answers - MATLAB Central. Recuperado el 23 de mayo de 2024, de <https://es.mathworks.com/matlabcentral/answers/505826-implementing-code-from-matlab-into-android-studio>
- [36] Alextryfg. GitHub - Alextryfg/CodigoTFGFragilidad. GitHub. Recuperado de <https://github.com/Alextryfg/CodigoTFGFragilidad/tree/master>
- [37] Neira, G. G., Álvarez, M. N., Hoyas, E. H., Ruiz, L. R., García-De-Villa, S., Del-Ama, A. J., Sánchez, M. C. R., & Ruiz, A. J. (2022). GSTRIDE: A database of frailty and functional assessments with inertial gait data from elderly fallers and non-fallers populations [Conjunto de datos]. En Zenodo (CERN European Organization for Nuclear Research). <https://doi.org/10.5281/zenodo.6883292>

- [38] figshare. (2023, 30 mayo). A public data set of overground and treadmill walking kinematics and kinetics of healthy individuals. Figshare. https://figshare.com/articles/dataset/A_public_data_set_of_overground_and_treadmill_walking_kinematics_and_kinetics_of_healthy_individuals/5722711?file=10097319
- [39] GAIT Maturation Database. (31 de mayo 2024). <https://archive.physionet.org/physiobank/database/gait-maturation-db/data/>
- [40] Alex Fuentes Gómez. (2024, Junio 19). *Funcionamiento del Prototipo del Trabajo de Fin de Grado*. YouTube. <https://youtu.be/zBzYZqZ2yws>