Universidad del País Vasco Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

MÁSTER UNIVERSITARIO EN

INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

DESARROLLO DE UNA APLICACIÓN PARA REALIZAR LA VALORACIÓN FUNCIONAL DE LA MARCHA USANDO UNA CONTERA

Estudiante *Sinovas Castro, Iker*
Director/Directora *Zubizarreta Pico, Asier*
Departamento *Ingeniería de Sistemas y Automática*
Curso académico *2019-2020*

Bilbao, agosto 2020

RESUMEN TRILINGÜE Y PALABRAS CLAVE

Resumen

El documento que se presenta a continuación recoge un Trabajo Fin de Máster realizado en el marco del proyecto SMARTIP. En este Trabajo Fin de Máster se desarrolla una aplicación Android que permite la interacción con la contera inteligente, a fin de analizar una valoración funcional de la Esclerosis Múltiple. Dicha enfermedad se trata de una patología crónica, neurodegenerativa y progresiva, de la que se desconoce una cura, lo que conlleva un alto gasto económico tanto para el sistema sanitario, como para las familias que conviven con la enfermedad. Existen múltiples tratamientos, pero todos ellos con un enfoque paliativo y/o rehabilitador. Debido a la multiplicidad de formas evolutivas que tiene la enfermedad y la cantidad de síntomas que pueden darse o no en cada uno de los casos, las personas que padecen la enfermedad deben ser tratadas de distintas formas. Por lo que se requiere de tratamientos personalizados a cada cual, y para ello es preciso el conocimiento exhaustivo de la Esclerosis Múltiple que padece cada persona.

Así, gracias a la ayuda de la contera inteligente, se adquiere información que puede ser útil para su tratamiento, ya que se transmiten datos acerca de 14 valores diferentes. La aplicación se desarrolla en Android Studio y la conexión entre los dos dispositivos se realiza mediante Bluetooth Low Energy. El uso de la aplicación Android favorece la obtención de datos en tiempo real, debido a la conexión permanente e inmediata que se da entre la aplicación y la contera sensorizada; además da la posibilidad de realizar en el mismo momento otra prueba si la primera resultase fallida. Asimismo, favorece el seguimiento evolutivo de la enfermedad, ya que durante el proyecto se ha perseguido y logrado el objetivo de creación de una base de datos para cada persona usuaria, y la posibilidad de acceder a estos registros a través de la nube a la cual estos datos están subidos. Logrando con todo ello el conocimiento de distintos valores que posibilitan el ajuste del tratamiento a cada persona.

Palabras claves

Esclerosis múltiple / Contera inteligente / Bluetooth Low Energy / Android Studio / Base de datos / nube

Laburpena

Hurrengo txostenean SMARTIP proiektu barnean dagoen Master Amaierako Lana aurkezten da. Honetan, txurro adimentsuarekin Android aplikazio bat garatzen da,Esklerosi Anizkoitza duten pazienteen martxaren diagnosi funtzionala ebaluatzeko xedearekin. Esklerosi Anizkoitza gaur egun sendapenik ez duen neuroendekapenezko gaixotasun kronikoa eta progresiboa da; eta horrek kostu ekonomiko handiak sortarazten dizkie bai osasun sistemari, bai senideei ere. Badira makina bat tratamendu, baina denak ikuspuntu paliatibo eta errehabilitazio terapiekin lotutakoak. Gaixotasun hau pairatzen duten pertsonek modu ezberdinetako tratamenduak jasan ditzakete, izan ere, honako eritasuna era ezberdinetan bizi baita eta haren hedapena ez da pertsona guztietan modu berean garatzen. Hortaz, pertsona bakoitzak bere tratamendua du, eta horretarako bakoitzaren egoera zehaztea beharrezkoa da.

Txurro adimentsuari esker pertsona bakoitzerantzako tratamendu egokia ezartzeko informazioa jaso daiteke, proiektu honetan garatutako aplikazioaz baliatuta 14 balio ezberdin bidaltzen baititu. Aplikazioa Android Studio programan garatu da eta bi dispositiboen arteko konexioa Bluetooth Low Energy bitartez lortzen da. Android aplikazioaren erabilerak datuak denbora errealean jasotzea baimentzen du, txurro adimentsu duen etengabeko eta bat-bateko konexioa dela eta. Gainera, entsegutan akatsa egotekotan berehalako beste froga bat egiteko ahalmena du. Aplikazioak pertsona bakoitzaren gaixotasunaren eboluzioa jarraitzeko aukera ematen du honetan eratutako datu-basea dela medio. Bestalde, datuak hodeian aurki daitezke frogen erregistroa han gordetzen baita. Honek guztiak, erabiltzaile bakoitzarekiko tratamendu zehatza ezartzeko balioak ezagutarazteko gaitasuna ematen du.

Hitz gakoak

Esklerosi anizkoitza / Txurro adimentsua / Bluetooth Low Energy / Android Studio / Datu-basea / Hodei

Summary

The present thesis fits within the SMARTIP project. An Android App will be developed throughout the next pages. This app is connected to an intelligent tip for the crutches of people with Multiple Sclerosis' (MS), allowing the functional assessment of these patients. MS is a chronic, degenerative and progressive disease with no cure known, that's why is represents such a big economical problem both for Health System and families. Actually, there are many treatments approved, all of them as a palliative approach. The disease itself shows different kinds of evolutive processes and a huge variety of symptoms, so patients need various treatment modalities. As users benefit from personalized treatments, it is important to know how MS has affected each patient.

The intelligent tips helps gathering useful information for these patients' treatment, as it sends 14 values of different data. The App has been developed in Android Studio and connection between the tip and smartphone is provided by Bluetooth Low Energy. The Android App obtains patients' data in real time, thanks to permanent and immediate connection between the two devices. Furthermore, the app allows a second testing in case the first once failed. This system also makes patients' monitoring easier, as it creates a user data base so that this register can be found fully updated in the cloud. This way we can get to know different values in order to adjust each person's treatment.

Key-words

Multiple Sclerosis/ Intelligent Tip / Bluetooth Low Energy/ Android Studio/ Data Base / Cloud

ÍNDICE

TABLAS

FIGURAS

ACRÓNIMOS

AEDEM	Asociación Española de Esclerosis Múltiple
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
COCEMFE	Confederación Española de Personas con Discapacidad Física y Orgánica
EM	Esclerosis múltiple
EME	Federación de Esclerosis Múltiple España
EMPP	Esclerosis múltiple primaria progresiva
EMPR	Esclerosis múltiple progresiva recidivante
EMRR	Esclerosis múltiple remitente- recurrente
EMSP	Esclerosis múltiple secundaria progresiva
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
IoT	Internet of Things
L2CAP	Logical Link Control and Adaptation Protocol
LE PHY	Physical Layer
LL	Link Layer
MAC	Medium Access control
SIG	Bluetooth Special Interest Group
SMP	Security Manager Protocol
SQL	Structured Query Language
UUID	Identificador Único Universal

1 INTRODUCCIÓN

A día de hoy son muchos los dispositivos tecnológicos que se pueden encontrar en las actividades cotidianas de la sociedad. Volviendo la vista 50 años atrás ésta era una situación inesperada, pero a partir del siglo XXI la tecnología ha tenido una evolución muy positiva, inmiscuyéndose en sectores como el cuidado personal, la medicina, la educación, etc. Es esta segunda en la que se centra este proyecto: la medicina.

Las nuevas tecnologías han facilitado el avance en sectores relacionados con la salud, como así señala el Hospital Virgen del Mar en su publicación "Avances tecnológicos en la salud: mejoras aplicadas a la medicina". Y es que el progreso que se ha realizado en las tecnologías aplicadas a la medicina, como la tecnología IoMT, que posibilita la incorporación de sensores inalámbricos en equipos médicos y poder combinarlos con Internet, ha implicado un mejor control de las enfermedades, un perfeccionamiento en análisis de los diagnósticos de estas, y de su posible tratamiento de rehabilitación, o en su caso, de recuperación.

En el contexto que nos ocupa un equipo de investigación del Departamento de Automática y Robótica de la Escuela de Ingeniería de Bilbao de la UPV/EHU, ha desarrollado un proyecto del que forma parte este TFM. Proyecto denominado: 'Contera inteligente para el diagnóstico funcional de la marcha en pacientes con Esclerosis Múltiple' (SMARTIP).

La Esclerosis Múltiple es la enfermedad neurodegenerativa que actualmente más arraigada está en las personas jóvenes, concretamente de entre los 20 y 40 años de edad. Se trata, como así se narra durante el trabajo de una enfermedad crónica, progresiva y degenerativa que afecta al sistema nervioso central, y de la que actualmente se desconoce tanto la cura como una causa concluyente, aunque hay numerosas investigaciones que coinciden en la propiedad autoinmune de la enfermedad como la causante de ella, y por ende, de la destrucción de la capa de mielina por no reconocerla como propia del organismo.

La esclerosis múltiple también conocida como la enfermedad de las mil caras, puede evolucionar en cada persona de formas muy diversas. En concreto, existen cuatro formas evolutivas, según el progreso de la enfermedad -forma remitente-recurrente, forma progresiva secundaria, forma progresiva primaria y forma progresiva recidivante-; y en cada una de las personas son múltiples los síntomas que pueden darse. No necesariamente se dan todos los síntomas en una misma persona, y no todas las personas tienen los mismos síntomas, pero si es habitual que sean más de un síntoma los que se den durante la enfermedad. Así pues, hay múltiples formas de vivir la Esclerosis Múltiple y por ello, múltiples diagnósticos y tratamientos.

Los tratamientos de esta enfermedad son tratamientos paliativos y de rehabilitación, y ahí es donde tiene lugar este proyecto: SMARTIP. Con el objetivo de mejorar la calidad de vida de las personas que padecen la enfermedad y generar un análisis del diagnóstico más completo, perfeccionado y personalizado, para poder asimismo realizar un

tratamiento más acorde a las necesidades de cada persona. Así la contera inteligente que inicio recuperando los datos desde un ordenador, con el avance del proyecto vio necesaria la creación de una aplicación en un dispositivo móvil, debido a los avances y beneficios que podría conllevar.

El objetivo principal de este trabajo fin de máster es desarrollar una aplicación Android que permita la iteración con la contera inteligente del proyecto SMARTIP. Este proyecto se propone los objetivos de visualizar los datos adquiridos a tiempo real permitiendo un seguimiento de los parámetros de la muleta; generar almacenaje de los datos adquiridos en la memoria interna vinculados al ID e Iteración del usuario, para poder llevar un seguimiento de la enfermedad en el paciente; crear acceso directo para el almacenaje de la base de datos en una nube y facilitar el acceso a los datos registrados por la contera; y paliar el esfuerzo físico y monetario que conlleva la recogida y el análisis de datos desde el ordenador para el sistema inteligente a la hora de realizar el diagnóstico funcional de la marcha en pacientes de Esclerosis Múltiple.

Para el desarrollo de aplicación se utilizará la herramienta de Android Studio. A través de la cual se ha creado una aplicación con la capacidad de capturar los datos medidos para 14 valores. Estos datos a su vez serán transmitidos de la contera inteligente a la aplicación a través de la tecnología BLE -Bluetooth Low Energy-, desde donde serán guardados en una base de datos seleccionada y subidos a la nube, con el fin de acceder a ellos en cualquier momento y desde cualquier dispositivo móvil.

En lo que se refiere a la estructuración del trabajo, se encuentran nueve apartados. En primer lugar se realiza un abordaje contextual de los principales conceptos teóricos vinculados a la Esclerosis Múltiple y sus tratamientos, y a la tecnología utilizada, donde se hará hincapié en la Monitorización Remota e Internet of Things. En el segundo apartado se han planteado el objetivo general, como los específicos y el alcance de este trabajo de fin de máster. El tercer apartado aborda los beneficios del proyecto, tanto a nivel general, como a nivel social. El cuarto apartado explica cuáles han sido los requerimientos a la hora del desarrollo de la aplicación, ya que este TFM se ha realizado en el marco del proyecto SMARTIP. El quinto apartado, recoge dos de las elecciones importantes del proyecto como son la base de datos y la nube. En este apartado se aborda una recogida de información con varias alternativas para cada categoría y la decisión final para cada una de ellas. En el sexto apartado se plantea el diseño de la aplicación, comenzando con la explicación del funcionamiento de la muleta y describiendo los sensores de los que dispone la contera inteligente, siguiendo con la tecnología BLE, la pila de protocolos y GATT; tras ello se desarrolla el análisis de la aplicación a nivel general y por ventanas; y por último, se pasa a describir el código diseñado. El séptimo apartado detalla la metodología utilizada en el desarrollo del trabajo: las tareas, las fases y el diagrama de Gantt. Seguido del octavo apartado que describe el presupuesto del proyecto. Y por último, se cerrará este trabajo con el noveno apartado, las conclusiones y principales vías de investigación para el futuro. Seguido de la recopilación bibliográfica utilizada. Cabe destacar que al proyecto irán anexados una serie de documentos entre los que podemos encontrar el manual de usuario.

2 CONTEXTO

2.1 Esclerosis múltiple

Según la Sociedad Española de Neurología, a finales del año 2019 la Esclerosis Múltiple (EM), afectaba a 2.5 millones de personas en todo el mundo, de las cuales 770.000 se encuentran en Europa. A nivel estatal son más de 55.000 personas las que padecen esta enfermedad en España, entre estos casos el 5% se dan en la CAPV. Estos datos son reflejo de la realidad sociosanitaria que viven muchas familias en el mundo. Siendo considerada la enfermedad neurológica crónica más frecuente en adultos jóvenes de entre 20 y 40 años. Aunque también afecta a los hombres, la Esclerosis Múltiple es más prevalente en las mujeres, dándose 2 de cada 3 casos en éstas.

De acuerdo con la Federación de Esclerosis Múltiple España -EME-, la EM es una de las causas que mayores casos de discapacidad genera en España (Weber, 2018). Asimismo, en las últimas décadas los diagnósticos de personas que padecen la enfermedad se han multiplicado (Pérez Menendez, 2016). Según los datos presentados por EME en España, se diagnostican 1.900 casos nuevos cada año, y 1 nueva familia es afectada por esta enfermedad cada 5 horas. A continuación, se visualizan los datos registrados a nivel mundial en referencia a la envergadura de la EM a través de un mapa publicado por la Federación de Esclerosis Múltiple España (Weber, 2018).

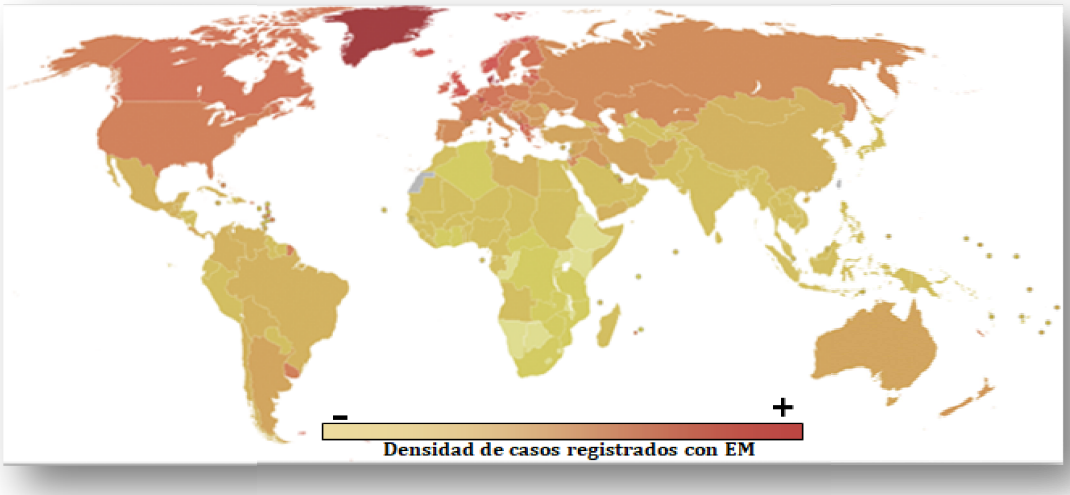

Figura 1. Densidad de casos registrados con EM. Fuente: Federación de Esclerosis Múltiple España, 2020.

La EM es una enfermedad crónica, progresiva y neurodegenerativa, o lo que es lo mismo, que afecta al sistema nervioso. En concreto se trata de una de las enfermedades más comunes del sistema nervioso central, que está formado por el cerebro y la medula espinal. Asimismo, es definida como una patología de origen autoinmune, lo que quiere decir, que el organismo arremete contra su propio tejido no reconociéndolo como propio, sino como un cuerpo invasivo dañino para sí.

Actualmente, no se conoce una causa concluyente, pero muchas de las investigaciones coinciden en que es la característica autoinmune de esta enfermedad, la que causa los síntomas que la EM lleva consigo.

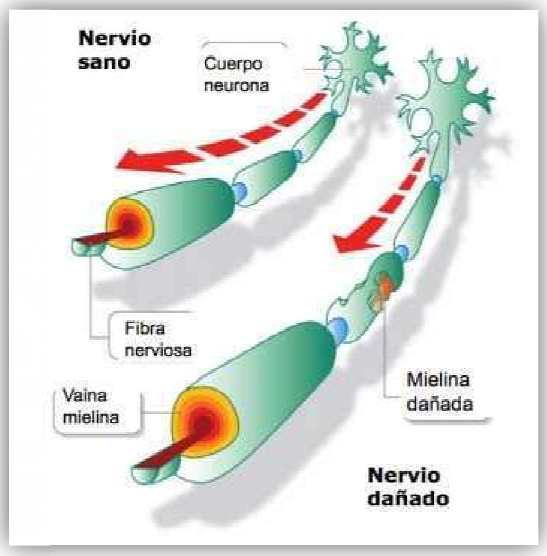

Figura 2. Estructura del SNC

De este modo, el sistema inmune ataca al sistema nervioso, en concreto a la capa aislante que recubre los nervios, conocida como la mielina (ver Figura 2. Estructura del SNC). Esta capa formada por sustancias grasas y proteínas, es la encargada de que los impulsos eléctricos se transmitan eficientemente y de forma rápida a las neuronas. Por tanto, considerando esta situación, son innumerables las investigaciones que señalan a la destrucción de la mielina como causa de la ralentización de los impulsos, y por tanto de la enfermedad de Esclerosis Múltiple.

En cuanto a los síntomas que genera la enfermedad no es coincidencia que la EM sea conocida como la enfermedad de las 1.000 caras, y es que se trata de una patología con múltiples síntomas, pudiendo ser estos diferentes en cada persona e, incluso, en una misma persona en diferentes etapas de la enfermedad. Los síntomas difieren en cuanto a cuáles son las áreas del sistema nervioso central que están dañadas. Aquí podemos observar cuales son los síntomas que pueden padecerse en el transcurso de la enfermedad (EME, Esclerosis Múltiple):

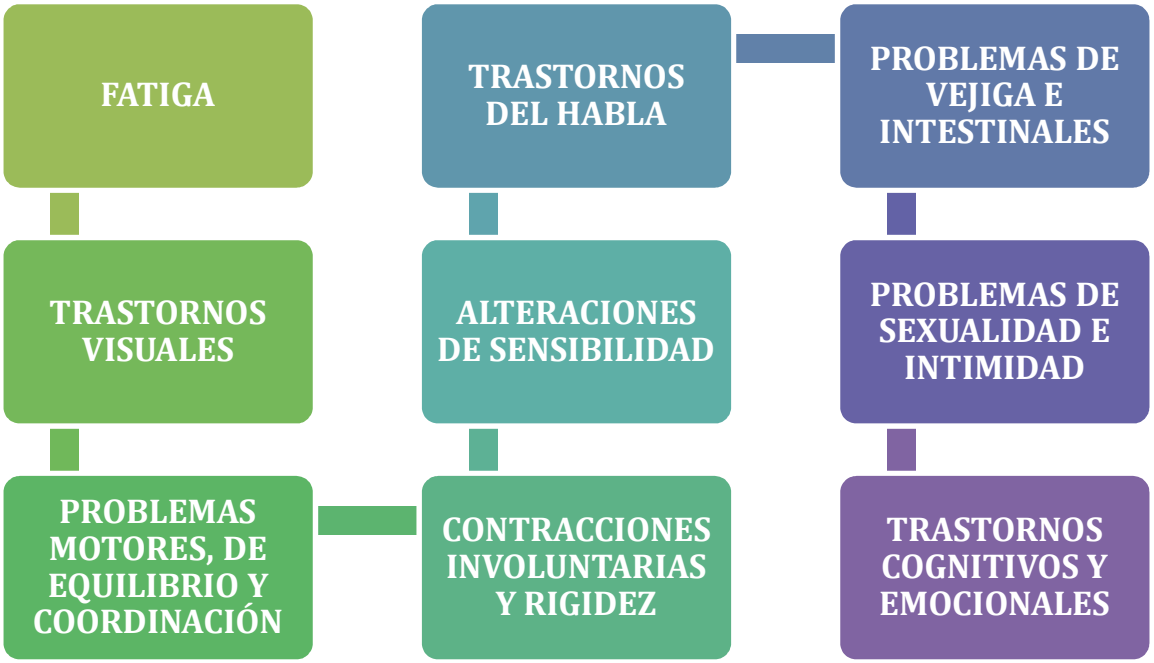

Figura 3. Síntomas de la EM. Fuente: EME, 2020.

Resulta importante destacar que no todos los síntomas son experimentados por una misma persona, pero es frecuente que sean más de uno los que aparezcan en cada caso, y no necesariamente con la misma duración ni gravedad. Asimismo, existen distintos tipos evolutivos de la enfermedad, pero estos no están directamente relacionados con la forma en la que cada persona experimenta los síntomas, sino con el modo en que evoluciona la enfermedad. Estas son las 4 formas que podemos encontrar (EME, Esclerosis Múltiple) (FEGADEM):

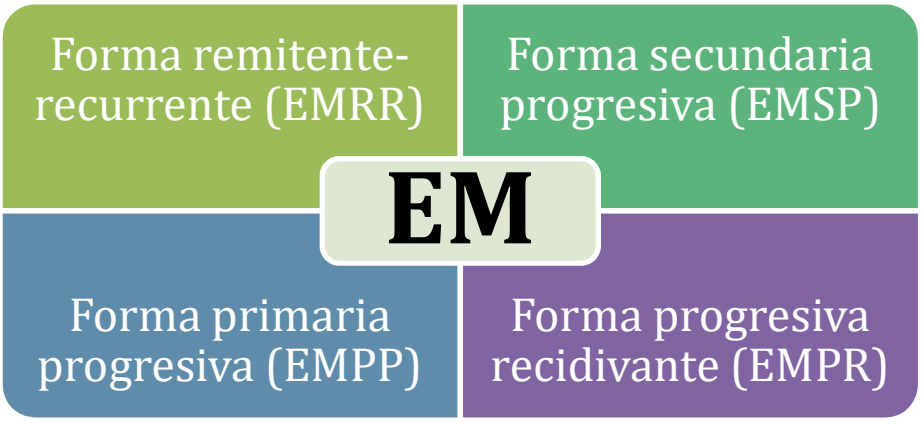

Figura 4. Formas de la EM. Fuente: EME y Esclerosis Múltiple Galicia, 2020

Forma remitente-recurrente (EMRR).

Este tipo se da en un 80% de las personas afectadas por EM, por lo que es la forma más habitual. En el inicio de la enfermedad los síntomas invisibles son muy comunes. Es posible, que incluso durante años, el organismo haya estado sufriendo lesiones inflamatorias en el sistema nervioso central, pero sin llegar a dar lugar a una sintomatología. Esta forma evolutiva se caracteriza por la aparición imprevisible de los síntomas que pueden durar varias semanas y luego desaparecer, sin propiciar ninguna progresión en la evolución de la enfermedad, es decir volviendo, parcial o totalmente, al estado inicial anterior al brote.

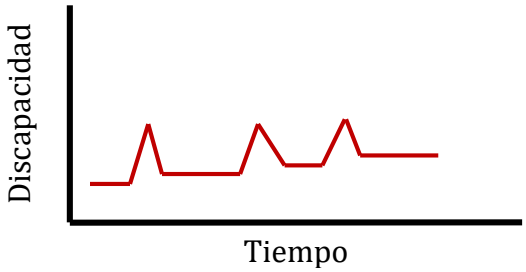

Figura 5. Forma EMRR

Forma progresiva secundaria (EMSP).

Esta forma evolutiva que usualmente se da en personas de entre 35 y 45 años de edad, la padecen entre un 30% y un 50% de las personas que con anterioridad han sufrido la forma remitente-recurrente. Se trata de un tipo evolutivo en que el grado de discapacidad, tanto física como cognitiva, va empeorando o se mantiene, entre los brotes de la enfermedad.

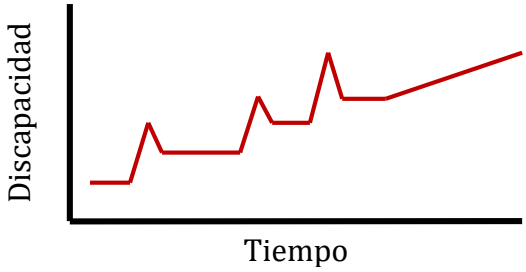

Figura 6. Forma EMSP

Forma progresiva primaria (EMPP).

Este tipo evolutivo lo padecen el 10% de las personas con Esclerosis Múltiple. En este tipo existe un comienzo lento de la enfermedad con un empeoramiento progresivo y constante sin la apariencia de brotes ni periodos de mejora o remisión considerables.

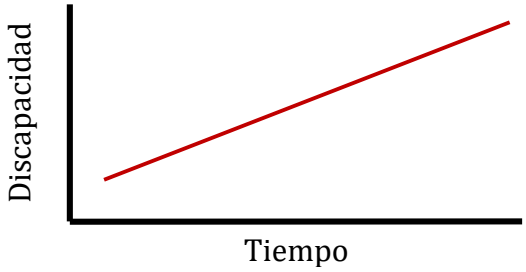

Figura 7. Forma EMPP

Forma progresiva recidivante (EMPR).

Es un tipo evolutivo en el que el empeoramiento de la enfermedad es constante desde el inicio de esta. Durante la evolución existen episodios o periodos de agravamiento claros que no siempre conllevan a una recuperación completa. Además, se caracteriza por ser una forma de la enfermedad en la que no hay momentos de mejora sino una progresión continúa.

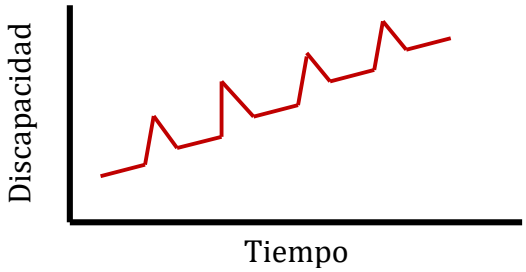

Figura 8. Forma EMPR

Asimismo, junto con las formas o tipos evolutivos cabe destacar un dato importante del trascurso de la enfermedad, y es que es muy frecuente que tras 15 años del primer brote de la enfermedad, la persona requiera de una ayuda técnica para mantener su nivel de autonomía (Souza, 2010), como pueda ser un bastón, muletas, silla de ruedas o andador.

Figura 9. Sistema de ayuda técnica

La Esclerosis Múltiple se trata de una enfermedad crónica de la que no se conoce causa concluyente, y la cual, a día de hoy, carece de cura. Este hecho unido a varios factores supone un elemento clave en el impacto económico tan elevado que tiene la enfermedad en la sociedad. Se estima que en España son 1.200 millones de euros los que se destinan a subsanar el coste sanitario que conlleva esta enfermedad (Izquierdo Ayuso, 2014), siendo el coste empleado en cada caso diferente. Esto se debe a la diferencia del grado de discapacidad y el incremento de esta, como así señalan distintas investigaciones. Ejemplos de ello son las de Karampampa, Gustavsson, Miltenburger y Ecker (2012), o la de Oreja-Guevara et al. (2017), que coinciden en que el incremento de la discapacidad y el deterioro de la autonomía, y por tanto, un peor estado de salud, implican un mayor coste económico.

Así, se puede afirmar que la discapacidad que conlleva la enfermedad es uno de los factores que generan el impacto económico de la EM. Unido a este factor se encuentra el hecho de tratarse de una enfermedad crónica careciente de cura, lo que conlleva una mayor prevalencia y gasto directo en tratamientos continuos, medicación o terapias rehabilitadoras (World Health Organization, 2006). Además, se debe contemplar el factor laboral, o en este caso de exclusión laboral. La Esclerosis Múltiple afecta a personas jóvenes en edad de trabajar, personas que con la evolución de la enfermedad van perdiendo autonomía y encontrándose dificultades para llevar a cabo su trabajo con normalidad, siendo en múltiples ocasiones necesario el retiro o abandono laboral antes de llegar a la edad de jubilación (García-Domínguez J.M. et al. 2019). Con el finde prevenir esta situación y mantener en la medida de lo posible la calidad de vida de las personas y familias afectadas por la Esclerosis Múltiple, son varios los tratamientos que se siguen con un carácter preventivo, rehabilitador y paliativo.

2.2 Tratamientos de la EM

Ante la ausencia de cura para la Esclerosis Múltiple se llevan a cabo tratamientos tanto farmacológicos como rehabilitadores, todos ellos con el objetivo de prevenir o controlar los brotes que puedan ocasionarse, también para ralentizar o disminuir el avance de la enfermedad y para mitigar muchos de los síntomas de la EM. Todo ello, con un fin último: la mejora de la calidad de vida de la persona y de su entorno social, unido al conocimiento exhaustivo de la enfermedad y el proceso evolutivo de esta en cada persona.

No es posible comprender un tratamiento en una enfermedad neurodegenerativa en que no se combinen el tratamiento farmacológico, o dicho de otro modo, la medicación estipulada por el sistema sanitario, y el tratamiento rehabilitador integral. Así, tal y como se cita en una guía redactada con la colaboración de Roche Farma, EME y AEDEM-COCEMFE, *el tratamiento de la enfermedad no es sólo farmacológico* (2019). El tratamiento rehabilitador, donde podemos incorporar terapias de logopedia, psicología, trabajo social, terapia ocupacional y fisioterapia entre otras, tiene un papel determinante en la promoción de la autonomía y prevención de la dependencia de la persona con EM, y por ende en el estado de su calidad de vida (Flachenecker, 2015). Y es en ese punto donde se quiere intervenir con este proyecto.

La terapia rehabilitadora que se lleva a cabo en las enfermedades degenerativas, consiste en una serie de ejercicios físicos y mentales, que dirigidos a los síntomas que la persona padece ayudan a mejorar las funciones y la habilidad de las partes locomotoras del organismo afectadas por la enfermedad, como pueden ser la concentración, el habla, el equilibrio o la movilidad. Es esta última en la que incorporaremos nuestro dispositivo inteligente, la contera.

Estos programas de ejercicios que se plantean en la rehabilitación no son una cura para la Esclerosis Múltiple, ya que como apuntábamos, a día de hoy se carece de dicha cura. Pero aunque estos ejercicios no eliminen los brotes o recaídas, ni el desarrollo de la enfermedad, son una herramienta para recuperar la actividad física (Bennett, 2010) y que el declive de esta se dé de forma desacelerada. Todo ello favorece su integración sociofamiliar y su mantenimiento en la vida laboral, además de aportar un beneficio en el desarrollo de las tareas de su vida diaria y de su calidad de vida (EME, Roche Farma, AEDEM-COCEMFE, 2019).

Pero como veníamos diciendo la obtención de ello depende de la unión de ambos tratamientos, no deben realizarse de forma independiente. Además, teniendo en cuenta lo dicho en el apartado anterior en referencia a la enfermedad de las mil caras, es decir, la multiplicidad de formas de vivir esta enfermedad, es necesario que los tratamientos se personalicen y sean acordes a cada una de las personas a quien se vayan a dirigir (EME, Roche Farma, AEDEM-COCEMFE, 2019), lo que hace que sea imprescindible llevar a cabo distintas pruebas y controles continuados a los pacientes, a fin de conocer de forma holística la enfermedad de cada cual y su evolución, para ajustar a estas cada intervención a realizar.

A día de hoy son muy diversas las formas de obtención de este tipo de datos, como las escalas, cuestionarios, etc. Pero actualmente y desde la última década, es cada vez más habitual la obtención de datos a través del uso diario o frecuente de dispositivos que analizan los movimientos y que registran los datos de los movimientos realizados, como puede ser las pulseras inteligentes que cada vez más personas llevan en sus muñecas a modo de análisis de trayectorias diarias, estado del sueño, etc.

Sin embargo estos dispositivos son más frecuentes en el uso de personas sanas para mejorar el rendimiento físico o, simplemente, conocer los movimientos y otros aspectos para actividades concretas o a lo largo del día. Ahora bien, ¿y si ese uso se diera para el conocimiento del avance de las enfermedades degenerativas? En la Esclerosis Múltiple pudiera ser una tecnología beneficiosa.

2.3 Tecnología

En las últimas décadas especialmente a partir del siglo XXI la tecnología ha tenido una evolución exponencial. A día de hoy el uso de dispositivos tecnológicos esta al orden del día, lo que apenas 50 años atrás era algo impensable, actualmente forma parte de las actividades cotidianas del día a día. Estos mismos avances también se han dado en el sector de la medicina y en los cuidados de la salud, nutriéndose de valiosas nuevas

tecnologías como el *Robot DaVinci*, el cual realiza cirugía robótica facilitando que las intervenciones sean más cómodas; o la *biopsia líquida* que permite identificar células cancerosas tumorales o ADN de células tumorales que están circulando en la sangre (Hospital Virgen del Mar).

Los avances tecnológicos están creando un cambio en el campo sanitario al disponer de tecnología más ventajosa que permite un mejor control y análisis en los diagnósticos o tratamientos de enfermedades.

2.3.1 Monitorización remota

Se denomina monitorización remota (MR) a la tecnología de trasmisión que se aplica para la adquisición de datos de un dispositivo tecnológico implantado en el cuerpo de una persona, sin que esta tenga necesidad de acudir a la consulta del especialista. Para realizar dicha función el recurso tecnológico implantado en el paciente ha de tener la capacidad de transmitir datos y, a su vez, esta persona deberá tener en el domicilio un dispositivo con la capacidad de transferirlos (García Urra, Porres Aracama, & Fontán Martín-Chico, Dispositivos eléctricos y monitorización remota, 2013) (García Urra & Porres Aracama, 2015).

En 2001, Biotronik desarrolló el primer marcapasos que podía ser monitorizado desde un teléfono móvil, lo que facilita que personas con dificultades de movilidad, problemas de desorientación, alto grado de dependencia, y otro tipo causas por las que un seguimiento presencial es dificultoso, puedan tener un seguimiento domiciliario. Debido a ello, y tras analizar los beneficios que esta tecnología conlleva, diversas empresas Europeas, como Medtronic, St. Jude Medical, Boston Scientific y Sorin Group, incluyeron la monitorización entre sus servicios, pero con pequeñas diferencias. Aunque todas tienen el objetivo de obtener información valiosa sin tener que acudir a una consulta presencial (García Urra, Porres Aracama, & Fontán Martín-Chico, Dispositivos eléctricos y monitorización remota, 2013) (García Urra & Porres Aracama, 2015).

PROCEDIMIENTO CONSULTA PRESENCIAL

Figura 10. Procedimiento de consultas presenciales y no presenciales. Fuente: (García Urra, Porres Aracama, & Fontán Martín-Chico, Dispositivos eléctricos y monitorización remota, 2013) (García Urra & Porres Aracama, 2015).

Hoy en día existen varios sistemas subcutáneos que utilizan la monitorización como son los marcapasos, desfibriladores (DAI), Holter y dispositivos de resincronización cardíaca (CRT) que ofrecen aspectos programables y pueden almacenar grandes cantidades de información.

Figura 11. Ejemplos de sistemas tecnológicos subcutáneos

Su funcionamiento es sencillo, el paciente dispone de un dispositivo implantado que transmite la información a una hora previamente establecida por el personal médico o cuando sucede una situación de cambio -evento-, como una variación en el funcionamiento del dispositivo o una arritmia. El transmisor del paciente recoge la información y envía vía multimedia la información codificada al centro de servicios de la compañía. A continuación, los datos se vuelcan a la base de datos, que realiza una diferenciación en el motivo de trasferencia. Así, en caso de que se haya activado por un evento se avisa al médico inmediatamente a fin de que la información sea revisada; y si por el contrario la información no es trasferida debido a un evento esta quedará

guardada a la espera de ser analizada (García Urra, Porres Aracama, & Fontán Martín-Chico, Dispositivos eléctricos y monitorización remota, 2013).

Figura 12. Funcionamiento de la monitorización remota

2.3.2 Internet of Things (IoT)

Internet of Things, Internet de las Cosas, es un concepto desarrollado a partir del año 1999, el cual, según su definición, es conectar todo tipo de dispositivos a través de Internet. Un concepto que inicialmente limitaba su uso al campo industrial y domótico, pero que en la actualidad va más allá. Esto es debido a que este concepto se ha expandido y hoy en día se disponen de diversos dispositivos, incluso elementos cotidianos, capaces de conectarse a Internet (Hu, Xie, & Shen, 2013).

Uno de esos campos es el de la medicina, en el que dicha tecnología adquiere un nuevo nombre: IoMT, o lo que es lo mismo, Internet de las Cosas Médicas. En el ámbito de salud la aparición de la tecnología IoT ha permitido que se promueva un nuevo desarrollo del modelo médico. Esto se debe a la posibilidad de incorporar sensores inalámbricos en equipos médicos y poder combinarlos con Internet. La incorporación de esta tecnología permite que el diagnostico se mejore y agilice, ya que abre un abanico de posibilidades como es el acceso instantáneo a plataformas creadas en la nube donde se almacena la información de cada paciente. Además, permite que se haga una monitorización de esta persona a distancia, ya que utilizando esta tecnología no únicamente se consigue que el material del centro sanitario o de hospitales se transforme en un dispositivo inteligente, sino que permite que herramientas de apoyo o de uso cotidiano de los pacientes también lo sean, y les mantengan continuamente conectados a su personal médico (IAT).

Para vislumbrar más la importancia de la IoT se van a explicar algunas aplicaciones en el sector médico y en el cuidado de la salud.

Equipo médico y control de medicamentos

El IoT posibilita la visualización de la gestión de materiales, se puede monitorizar todo el proceso de producción y entrega de equipos médicos y control de medicamentos, además de combatirla falsificación de estos. Esta monitorización de datos a tiempo real permite que se dé un mayor control y seguridad al poder evitar la falsificación de equipos médicos y medicamentos (Hu, Xie, & Shen, 2013).

Gestión de información medica

Se posibilita una mejor gestión de la información de cada paciente, por lo que el personal médico o de enfermería dispone inmediatamente del historial clínico del paciente, pudiendo evitar errores en el uso de medicamentos o inyecciones (Hu, Xie, & Shen, 2013).

Telemedicina

La telemedicina es un nuevo servicio médico que combina la tecnología informática, tecnología de la comunicación, tecnología multimedia y tecnología médica, es decir, la telemedicina permite el intercambio de información valido entre profesionales sanitarios para el diagnóstico, tratamiento o incluso prevención de enfermedades. En otras palabras, la telemedicina agiliza la coordinación, colaboración y el trabajo multi e interdisciplinar en el ámbito sanitario. Este nuevo servicio tiene el objetivo de mejorar los diagnósticos, ahorro de costes sanitarios, satisfacer las necesidades de las personas y construir un sistema de salud centrado en el paciente (Hu, Xie, & Shen, 2013).

Atención medica móvil

La atención médica móvil establece una base de datos de los datos de monitorización de sus signos vitales, entre los que se incluyen el peso, colesterol, contenido de grasa, etc. Es una base de datos única para cada persona, y los resultados son transferidos a la unidad pertinente, enfermería o unidades médicas relacionadas (Hu, Xie, & Shen, 2013).

Pastillas inteligentes

Abilify MyCite son unas pastillas inteligente desarrolladas por Otsuka Pharmaceutical Co. y Ltd. y Proteus Digital Health. Estas píldoras estás diseñadas para el tratamiento de esquizofrenia o trastorno bipolar.

El usuario toma la pastilla, la cual tiene un sensor digital, que al estar en el estómago debido a los ácidos gástricos se disuelve. Es entonces cuando se activa el sensor y envía una señal a una aplicación móvil. Debido a este sistema el personal médico certifica que el usuario toma la medicación y a qué hora, con lo que el médico puede cerciorar que el paciente sigue el tratamiento y hacer un seguimiento de este (IAT).

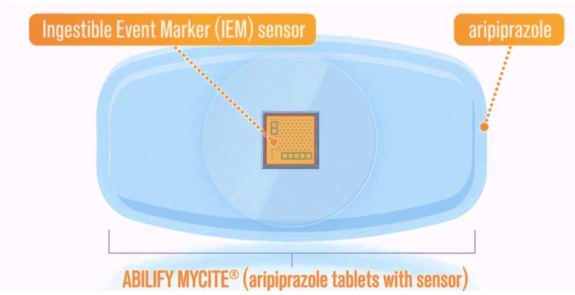

Figura 13 Pastilla inteligente

Como se puede observar, el concepto IoT abarca un abanico de posibilidades, desde la gestión de información hasta el control de la medicación, o un seguimiento continuo e inmediato del tratamiento de las personas usuarias de sistema sanitario.

Así, tras un largo recorrido por lo que podríamos denominar marco teórico o contexto del proyecto es imprescindible acabar este apartado señalando que ateniéndose a todo lo explicado en él y sosteniéndose a los motivos citados, un grupo de investigación del Departamento de Automática y Robótica de la Escuela de Ingeniería de Bilbao comenzó con el proyecto de *Desarrollo De Un Sistema Inteligente Para El Diagnóstico Funcional De La Marcha En Pacientes De Esclerosis Múltiple* (Brull Mesanza, 2020). En este caso, el dispositivo inteligente diseñado centra su objetivo en el ámbito de la movilidad, concretamente en el estudio y análisis de la marcha de la persona afectada. Para ello, el departamento comenzó a desarrollar una muleta inteligente capaz de analizar las fuerzas y los movimientos del propio dispositivo con el objetivo de facilitar el análisis del diagnóstico del paciente.

Tras varios años de investigación se ha diseñado un prototipo de muleta que es capaz de transmitir los valores de fuerza, altitud, orientación, aceleración, velocidad de giro y el campo magnético. Estos datos son enviados al ordenador mediante un programa que crea un archivo que posteriormente es utilizado para realizar estudios.

Así, finalmente motivados por una mejora en la portabilidad, debido a que el ordenador es un objeto pesado y de gran volumen, se ha decidido cambiar el sistema del ordenador a una aplicación móvil, por lo que se llevará a cabo este proyecto.

3 OBJETIVO Y ALCANCE DEL TRABAJO

El objetivo principal de este trabajo fin de máster es desarrollar una aplicación Android que permita la iteración con la contera inteligente desarrollada por un grupo de investigación del Departamento de Automática y Robótica de la Escuela de Ingeniería de Bilbao.

El objetivo principal se desglosa en los siguientes objetivos específicos:

- Desarrollar una aplicación que permita la conexión entre la contera y el móvil.
- Visualizar los datos adquiridos a tiempo real permitiendo un seguimiento de los parámetros de la contera.
- Generar almacenaje de los datos adquiridos en la memoria interna vinculados al ID e Iteración del usuario.
- Crear acceso directo para el almacenaje de la base de datos en una nube.
- Paliar el esfuerzo físico y monetario que conlleva la recogida y el análisis de datos desde el ordenador para el sistema inteligente a la hora de realizar el diagnóstico funcional de la marcha en pacientes de Esclerosis Múltiple.
- Facilitar el acceso a los datos registrados por la contera.

En cuanto al alcance del proyecto se ha decidido desarrollar una aplicación Android que permita la adquisición de los datos transferidos por la contera inteligente. Se opta por la aplicación móvil como sustituta del ordenador, ya que esta favorece la adquisición de los datos a tiempo real, además de su fácil manejo y disponibilidad de movimiento sin esfuerzo físico.

Mediante este proyecto se quiere dar respuesta a los objetivos planteados en el apartado anterior y para ello se deben alcanzar o superar tres etapas o fases:

- Primero, la conexión entre la contera y la aplicación. Esta conexión se realiza introduciendo el MAC y permitiendo realizar la conexión entre los dos sistemas. Posteriormente, se comienza con la transferencia de datos.
- Segundo, el almacenaje interno donde se introducen los datos de la prueba y una vez finalizado se guardan los datos en la memoria interna. Además, se integran gráficos que posibilitan la fácil visualización de los datos.
- Finalmente, se posibilita la subida de la base de datos elegida a una nube, se crea un archivo en formato CSV que se subirá a la plataforma. Este sistema permite la adquisición de los datos recolectados para futuros análisis.

4 BENEFICIOS DEL PROYECTO

Los beneficios que aporta este proyecto son cuantiosos tanto a nivel tecnológico como a nivel social.

4.1 Beneficios técnicos

A nivel tecnológico, ésta herramienta aporta beneficios como los siguientes:

En primer lugar, debido al cambio del dispositivo de captura a un móvil se va a conseguir una mejora en la postura ergonómica al hacer las pruebas. El dispositivo receptor de los datos de la muleta varía de un ordenador a un móvil, por lo que conlleva menos esfuerzo al ser un dispositivo más ligero. El sujetar el ordenador manteniendo una postura ergonómica es dificultoso debido a la fatiga que ocasiona el aparato. En este caso, se mejora la postura previniendo lesiones. Asimismo, a la hora de hacer las pruebas se ha de cambiar de ubicación, teniendo que llevar el equipo hasta el lugar de ensayos, de este modo el volumen de la herramienta disminuirá al ser el teléfono, en comparación al ordenador, un dispositivo más pequeño y a su vez más trasladable.

En segundo lugar, la integración de los gráficos posibilita un análisis de los resultados más visual y rápido. El analizador tiene el comportamiento de la prueba al instante, al final de la misma, por lo que en caso de no satisfacerle la prueba tiene la posibilidad de repetirla. Anteriormente, tenía que crear el gráfico en el archivo que se generaba, por lo que los tiempos eran más largos. De esta se facilita y agiliza el trabajo del equipo de fisioterapeutas.

Por último, los datos son almacenados internamente en el dispositivo móvil. Lo que permite que en caso de que un archivo se haya borrado accidentalmente en la plataforma de la nube, se permitirá volver a recuperar ese archivo con la posibilidad de volver a subirlo a la nube. Además, la memoria interna desarrollada y la nube ofrecen la posibilidad de conocer la evolución del usuario, ya que se puede acceder a los datos obtenidos en pruebas realizadas con anterioridad.

4.2 Beneficios sociales

A nivel social, por otro lado, el presente proyecto también presenta beneficios. El gasto económico que supone la EM para el sistema sanitario es muy elevado, pero esta enfermedad también conlleva un gasto económico, en ocasiones impermisible, para las familias que conviven con la EM. Al tratarse de una enfermedad crónica, progresiva, degenerativa y para la que actualmente se carece de cura, son muchas las pruebas que deben hacerse para suministrar el tratamiento adecuado en cada momento de la enfermedad, o para establecer las terapias rehabilitadoras más idóneas a cada situación. Con el proyecto SMARTIP, se favorece un conocimiento más completo y personalizado de cada caso, lo que implica una terapia rehabilitadora más puntillosa, integral y personalizada. O lo que es lo mismo, facilita un análisis diagnóstico más completo y una

terapia más veraz para cada persona, lo que a la larga se traduce en un menor coste económico para las personas y familias que conviven con ella.

Por otro lado, el hecho de que la aplicación sea móvil no implica un mayor coste, ya que actualmente no es usual que las personas no vayan acompañadas de este aparato telefónico. De ser ese el caso, la aplicación posibilita que desde un móvil ajeno a la persona se pueda acceder a sus datos o realizar pruebas para el paciente y guardar los datos en su base de datos, siempre y cuando, se utilice el ID de la persona afectada. Por tanto, no es necesario que la persona realice las pruebas o acceda a sus datos desde su móvil, sino que en caso necesario podría realizarlas desde el de un familiar o persona cuidadora.

Otro de los beneficios a tener en cuenta a nivel social, se trata de la integración sociolaboral. Muchas personas que padecen EM no pueden continuar en su puesto laboral con normalidad, e incluso se ven en la necesidad de abandonar el trabajo. Ello supone una situación de crisis económica en muchos núcleos convivenciales, o familias. Así, el diagnóstico rápido y completo, y por ende, el tratamiento personalizado y eficaz por el que vela este proyecto, favorecen la mejora de la situación de la persona, e incluso ralentizan la velocidad a la que evoluciona la enfermedad. Lo que puede verse traducido, en un mantenimiento de sus capacidades durante mayor tiempo, y un retraso en el momento de solicitar la incapacitación laboral.

Por último, el coste emocional que conlleva esta enfermedad para las personas y familiares también ha de tenerse en cuenta. La incertidumbre de lo que dirán las pruebas realizadas es un claro punto crisis emocional, nerviosismo, negatividad, tensión, ansiedad, etc. Con la creación de la aplicación estos tiempos se acortan, y se favorece el conocimiento de los datos a tiempo real. Además, si existiesen dudas de los resultados, permite volver a realizar la prueba en el mismo momento comparando los datos obtenidos. Lo que supone una mayor tranquilidad y alivio de la tensión, tanto paras las personas que padecen la enfermedad como para sus familiares.

A fin de cuentas, la aplicación diseñada favorece una mejora de la calidad de vida de las personas afectadas pero también de sus familiares. Entendiendo que a mejor salud mejor calidad de vida, y que la salud se mide tanto a nivel físico, como psicológico y emocional.

5 DESCRIPCIÓN DE REQUERIMIENTOS

Como previamente se ha comentado el proyecto viene derivado del proyecto SMARTIP, es por ello que se han fijado ciertos requisitos para el diseño de la aplicación. En el siguiente apartado se van a explicar los requisitos exigidos para el desarrollo del entorno software.

En primer lugar, la herramienta que se dispone para desarrollar la aplicación es Android Studio, una aplicación gratuita desarrollada por Google.

En segundo lugar, la conexión entre dispositivos se realiza entre la aplicación móvil y la contera inteligente que dispone de sistema de transmisión previamente programado, por lo que es preciso y requisito indispensable ajustar la transmisión al sistema.

Por último, se han exigido la incorporación de 3 funciones específicas y esenciales en el diseño:

1. Se debe realizar una base de datos en la cual serán almacenados los datos de la contera internamente en el móvil.

2. Se deberá disponer de la opción de guardar archivos en una plataforma en la nube, favoreciendo la posibilidad de acceder a estos desde el ordenador.

3. Se deberá disponer la opción de visualizar los datos en tiempo real. Visualizar tanto el último dato como los 10 últimos valores. Para ello se han propuesto diversas opciones, como son la visualización mediante gráficos o tablas.

6 ANÁLISIS DE ALTERNATIVAS

Como se ha observado en los requerimientos se necesita implementar una base de datos para poder almacenar internamente los datos de las pruebas y buscar una plataforma en la nube donde guardar los archivos.

Por lo tanto, en el proyecto la base de datos y la plataforma en la nube son dos puntos clave. Dos apartados donde se debe decidir qué herramientas utilizar según las características de cada una de ellas, por lo que se pasaran a analizar las distintas alternativas a fin de decidir cuál es la más adecuada para este proyecto.

6.1 Base de datos

La base de datos es un apartado importante a tener en cuenta a la hora de realizar el desarrollo, debido a que debe de almacenar los datos de cada prueba manteniéndolos para futuros usos. En referencia a esta característica necesaria para el proyecto se han encontrado diversas alternativas:

6.1.1 SQLite

SQLite es un sistema de gestión de bases de datos de formato Structured Query Language (SQL). Es un sistema de código abierto, ligero, autónomo, de configuración simple y sin servidor. Como su propio nombre indica SQL es de lenguaje de consulta estructurada por lo que los datos son almacenados en tipo tablas. Entre los datos que puede almacenar se encuentran Int, Real, Double, Float, Text, Boolean, Date y Date time (Muradas Maceira, SQLite para Android: La herramienta definitiva, 2018).

Según el blog Open Webinars(2018) estas son varias de las ventajas que puede aportar:

- **Configuración sencilla:** Después de instalarlo, este sistema de base de datos no necesita configuración de rutas, tamaños, puertos entre otros como se debe de realizar en otros sistemas.

- **Es Software libre:** Todos los desarrolladores lo tienen disponible al ser de código abierto.

- **Almacena los datos de forma persistente:** Si el dispositivo de apaga los datos persisten y se encuentran correctos en la aplicación una vez que esta se vuelve a encender.

Resumiendo, SQLite es ideal para datos que se repiten o que mantienen una estructura uniforme, pudiendo almacenar datos a largo plazo y crear libremente la estructura necesaria para la aplicación (Muradas Maceira, SQLite para Android: La herramienta definitiva, 2018).

6.1.2 GreenDAO

GreenDao es un software de código abierto de mapeo objeto-relacional (ORM), permite la conexión entre objeto y la base de datos SQLite. El desarrollador solo ha de definir las tablas de la base de datos, y GreenDAO crea los objetos de datos (entidades) y los objetos de acceso de datos (DAO) facilitando su uso (Greenrobot).

Entre las ventajas que aporta se pueden encontrar las siguientes (Greenrobot):

- **Consumo:** El consumo de la memoria es mínimo.
- **Velocidad**: En comparación a otros software ORM es el más rápido realizando operaciones.
- **Librería poco pesada:** el tamaño de la librería es menos de 100KB.
- **Generación del código:** GreenDAO genera las entidades y los objetos DAO.
- **Seguridad:** Soporta el encriptado de la base de datos protegiendo los datos almacenados.

6.1.3 Realm

Realm es una base de datos de código abierto creado para que los desarrolladores que deseen una base de datos tipo NoSQL, es decir, una base que no requiere de una programación mediante estructuras fijas como son tablas, ya que no usa SQL como lenguaje principal de consultas. Esta base de datos puede usarse en plataformas como Android, IOS, Swift, Objective-C, JavaScript y NET (Realm Database) (Chugh, 2018).

Entre las ventajas que tiene el sistema Realm se encuentran:

- **Simplicidad:** Comparando con SQLite el código es más corto y conciso, solo necesita tratar con objetos y árboles de objetos.

- **Velocidad**: Debido a su diseño cero copia, Realm es más rápida que ORM y a veces más rápida que SQLite, con lo que puede guardar gran cantidad de datos en pocos minutos.

- **Seguridad:** Se puede incluir seguridad a la base de datos al poder usar encriptado.

Por lo tanto Realm es una base de datos no estructurada capaz de guardar gran cantidad de datos en poco tiempo, y dispone de la posibilidad de encriptar los datos aportando seguridad a la base de datos.

6.1.4 Elección de la base de datos

Para la elección de la base de datos se han seleccionado varias bases disponibles: SQLite, GreenDAO, Realm. A la hora de calificar cual es la mejor base de datos para implementar se han tenido en cuenta varios criterios: la funcionalidad, que permita crear una estructura de guardado apropiado para el almacenaje; la velocidad de guardado, que ha de ser alta, y que sea fácil a la hora de implementar.

A la hora de puntuar cada criterio se ha optado por valorarlo del 1 al 5, siendo 1 poco adecuado y 5 excelente.

	Funcionalidad	Velocidad	Implementación	Total
SQLite	5	4	4	13
GreenDAO	5	2	3	10
Realm	2	5	4	11

Tabla 1. Elección base de datos

Como se observa la opción mejor valorada es SQLite, una herramienta fácil de implementar y tiene una velocidad de almacenaje adecuada para grandes cantidades de datos. Asimismo, los datos generados en la aplicación mantienen una estructura uniforme y SQLite tiene una estructura de tipo tabla ideal para el desarrollo.

6.2 Plataforma nube

La plataforma en la nube es otro elemento importante en el proyecto debido a su uso para guardar los archivos de cada prueba en Internet. Se requiere de este elemento a fin de poder acceder a los archivos mediante el ordenador pudiendo descargarlos y realizar modificaciones y ensayos sobre ellos.

6.2.1 Google DriveAPI

Google drive API es una interfaz que comunica una aplicación Android con Google Drive. Esta API facilita la subida de archivos a la plataforma en la nube Google Drive. Además, posibilita la selección de usuario de una cuenta de Google lo que permite seleccionar a que cuenta subirlo (API).

Hasta el 6 de diciembre de 2019 estuvo operativa su utilización, a partir de dicha fecha se debía migrar a otro sistema lo que disminuyo las opciones de programación mediante este método (API).

6.2.2 Firebase

Firebase es una plataforma creada por Google que permite desarrollar la creación de apps en diferentes plataformas como iOS, Android y web. Entre las posibilidades que da esta plataforma se encuentra la del almacenaje en la nube por lo que cumple con el criterio para valorar las distintas alternativas (Firebase).

Centrando la atención en el servicio Cloud Storage, en otras palabras, el servicio que posibilita el almacenamiento de archivos en Firebase, es preciso destacar sus ventajas:

- **Rapidez de desarrollo:** Implementar Firebase es fácil y rápido debido a la facilidad que da la ruta de implementación.
- **Cargas y descargas robusta:** Detiene y reanuda las transferencias de archivos de forma automática cuando la conectividad móvil se recupera.
- **Almacenaje gratuito:** Se dispone hasta 5GB para almacenar los archivos gratuitamente.

6.2.3 Google Cloud Platform

Al igual que la ya mencionada plataforma de Firebase, Google Cloud Platform pertenece a Google. La diferencia entre ambas se encuentra, sobre todo, en que Google Cloud Platform está más enfocada al ámbito profesional y empresarial. La plataforma es de pago y su precio varía dependiendo de los servicios contratados (Google Cloud).

6.2.4 AWS (Amazon Web Services)

AWS (Amazon Web Services) es una plataforma en la nube que dispone de más de 175 servicios. Los servicios ofertados están relacionados con la comunicación entre servicios y el análisis de datos (AWS, Informática en la Nube con AWS, 2019).

AWS ofrece las siguientes ventajas:

- **Precio variable:** El precio varía dependiendo de los servicios contratados y se dispone de una opción gratuita durante 12 meses.
- **Seguro:** La infraestructura principal se diseñó con el objetivo de cumplir los requisitos de seguridad de organizaciones como el ejército o bancos internacionales, por lo que aporta una seguridad muy elevada. Además, ofrece la función de cifrar los datos de entre la información almacenada de los clientes.

6.2.5 Elección de la plataforma en la nube

Para la elección de la plataforma en la nube se han seleccionado varias herramientas disponibles: Firebase, Google Cloud, AWS y Google Drive Api. A la hora de calificar la mejor plataforma se han tenido en cuenta varios criterios: la facilidad de la implementación; la velocidad de guardado en la nube y el precio de una cuenta.

A la hora de puntuar cada criterio se ha optado por valorarlo del 1 al 5, siendo 1 poco adecuado y 5 excelente.

	Implementación	Velocidad	Precio	Total
Firebase	5	4	4	13
Google Cloud	3	3	1	7
AWS	3	5	1	9
Google Drive API	1	3	5	9

Tabla 2. Elección plataforma en la nube

Como se observa la opción mejor valorada es Firebase, una herramienta fácil de implementar debido a la guía paso a paso que ofrece en el programa Android Studio, y tiene una velocidad de almacenaje adecuada para grandes cantidades de datos. Asimismo, se disponen de 5GB de almacenaje gratuitos en la plataforma en la nube, suficientes para almacenar una veintena de archivos.

7 DISEÑO DE LA APLICACIÓN

A continuación se procederá a explicar el diseño de la aplicación. Para ello, es preciso explicar primero los componentes de los que se dispone para el desarrollo del proyecto, de este modo se conocerá el material con el que habrá que conectarse y la configuración dada al sistema de transferencia de los datos.

7.1 Funcionamiento de la muleta

Tal y como se ha comentado previamente el grupo de investigación del Departamento de Automática y Robótica ha desarrollado una contera inteligente, es decir, un dispositivo que registra los movimientos, velocidades, aceleraciones y otros valores que se recogen en él. Usando como base una muleta de la marca FDI France Médical -modelo Access-, se ha incorporado a la misma una contera sensorizada, tal y como se ve en la Figura 14.

Figura 14. Contera sensorizada acoplada a la muleta

En aras de seguir con el objetivo inicial la conversión de la muleta en un dispositivo IoT es imprescindible y para ello, ha sido necesario integrar en la contera sensorizada 3 componentes que pasaran a desarrollarse a continuación:

- Sensor de fuerza C9C
- MTi 1-series Development Board con modulo MTi-3-8A7G6
- Sensor BMP280

7.1.1 Sensor de fuerza C9C

Este sensor es un transductor de fuerza universal para fuerzas de compresión comercializado por HBM. Mide en un rango de 50N a 50KN con una precisión del 0.2%. Además, debido a su elevada frecuencia fundamental puede realizar mediciones muy rápidas. Asimismo, es un sensor con dimensiones reducidas, lo que facilita su integración en la muleta y al estar herméticamente soldado es capaz de soportar climas adversos (García Urra, Porres Aracama, & Fontán Martín-Chico, Dispositivos eléctricos y monitorización remota, 2013).

Figura 15. Sensor de fuerza C9C

Mediante este sensor se registra la fuerza aplicada en la muleta, tanto las fuerzas estáticas como las fuerzas dinámicas en compresión, lo que aporta información del uso que se está dando al dispositivo de ayuda para caminar.

7.1.2 MTi 1-series Development Board con módulo MTi-3-8A7G6

El MTi-1 de X-Sense se trata de un Inertial Measuring Unit (IMU), o lo que es lo mismo un dispositivo que permite obtener información acerca de la velocidad, fuerzas gravitacionales y orientación. Para ello, lleva integrados un acelerómetro 3D con rango de $\pm 156.96 m/s^2$, un giróscopo con rango de $\pm 2000º/s$ y un magnetómetro con rango de $\pm 0.8G$.

Figura 16. Mti-1 de Xsens

7.1.3 Sensor BMP280

El sensor BMP280 permite medir la presión barométrica y la temperatura, por lo que se puede obtener la altitud a la que se encuentra la muleta, y por ende la persona. El sensor cuenta con un rango de presión de 300 a 1100 hPa y una precisión de 1hPa en la presión barométrica; y en referencia a la temperatura dispone de una horquilla de medición -40ºC a 85ºC, y 1.0ºC de precisión.

Figura 17. Sensor de presión barométrico

Además, añadido a estas características su pequeño y el bajo consumo hacen de este sensor un elemento adecuado para el proyecto que se está desarrollando (Prometec).

7.1.4 Esquema de captura

Por tanto, concluyendo con este apartado, y en sintonía con lo descrito en él, se puede decir que mediante los componentes previamente descritos la muleta mide hasta 14 valores:

Figura 18. Datos obtenidos. Fuente: (Brull Mesanza, 2020)

Una vez que se han registrado los 14 valores previamente mencionados estos son transmitidos mediante RedBearNanoV2con un periodo de 10ms. Redbear Nano V2 es una tecnología Bluetooth Low Energy de pequeñas dimensiones y que consume poca batería.

7.2 Bluetooth Low Energy

A continuación se define que es el Bluetooth Low Energy y sus características principales, posteriormente se explica la arquitectura del protocolo, analizando los niveles, y cómo se ha configurado el dispositivo para transferir los datos en la muleta.

El Bluetooth Low Energy (BLE), también conocido como Bluetooth Smart, es una tecnología inalámbrica de bajo consumo, desarrollada por Bluetooth Special Interest Group (SIG), que conecta varios dispositivos tecnológicos entre sí. El BLE proviene de la última versión lanzada en 2010 del protocolo Bluetooth, Bluetooth 4.0, y su uso está enfocado para la aplicación en Internet of Things (IoT) (Bluetooth, 2018).

BLE está diseñado para la transferencia de pequeñas cantidades de datos por lo que los tiempos de transmisión son pequeños y por lo tanto tiene un menor consumo que el Bluetooth clásico. Esta tecnología es económica y la mayoría de móviles o dispositivos inteligentes la aceptan. Además, opera con 2.4 GHz en banda ISM, es decir, usa el mismo espectro que otras tecnologías inalámbricas como el Bluetooth Classic o el Wifi.

7.2.1 Pila de protocolos

El Bluetooth SIG define una pila de protocolos para la gestión de los dispositivos, la conexión y la interfaz de las aplicaciones. La pila del protocolo BLE se divide en tres partes básicas, organizadas por niveles: Controlador, Host y Applications (Figura 19. Pila de Bluetooth BLE). Siendo esta última la de mayor nivel. A continuación, se explicarán estas partes y sus niveles en un orden jerárquico de menor a mayor.

El Controlador, la parte más inferior, es el dispositivo físico que permite transmitir y recibir señales radio e interpretarlas como paquetes con información. Este dispositivo físico contiene la capa física, *Physical Layer (LE PHY),*entendida como el nivel más bajo, que realiza los procesos de modulación y demodulación de señales analógicas para transfórmalas en digital; la capa de enlace, *Link Layer (LL),* que se responsabiliza por un lado, de los procesos de Advertising y Scanning, y por otro lado, de la creación y el mantenimiento de las conexiones; y la interfaz de control de host, *Host Controller Interface (HCI)*, que permite la comunicación entre el host y el controlador. (Jacopo, Taffoni, Santacatterina, Sannino, & Formica, 2017) (ELT) (Akhayad, 2016).

El Host, la parte intermedia, es la pila de software que administra cómo dos o más dispositivos se comunican entre sí. En esta segunda parte se disponen de los niveles *Logical Link Control and Adaptation Protocol (L2CAP)* que actúa como multiplexor encargándose de los datos de niveles inferiores y adaptándolos al protocolo de niveles superiores; *Security Manager (SMP)* el administrador de seguridad; *Attribute protocol*

(ATT) que es el protocolo que permite el intercambio de información con una arquitectura cliente-servidor; *Generic Attribute Profile (GATT)*que define el modo en el que dos dispositivos BLE transfieren información; y *Generic Access Profile (GAP)* que permite que el resto de dispositivos visibilicen el dispositivo (Jacopo, Taffoni, Santacatterina, Sannino, & Formica, 2017) (ELT).

La aplicación (App) es el nivel más alto de la pila de protocolos, viene precedido del nivel GAP y representa el enlace directo con el usuario.

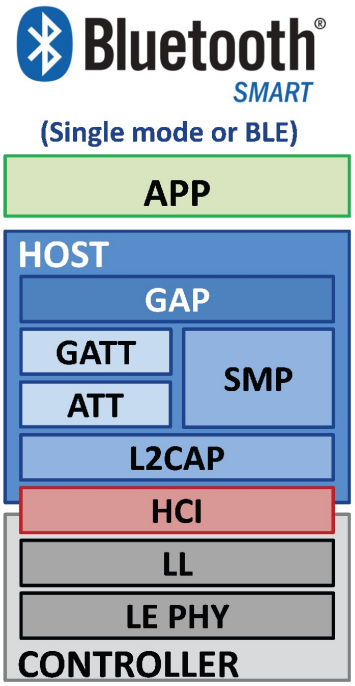

Figura 19. Pila de Bluetooth BLE

7.2.2 GATT

En el BLE se diferencian dos tipos de aparatos: dispositivo Central y dispositivo Periférico. El Central es el dispositivo que dispone de un abanico más amplio de características en el término CPU, memoria y batería. El Periférico es un dispositivo más ajustado. Al tener mejores características la responsabilidad del proceso se enfoca habitualmente en el Central, lo que conlleva a un menor gasto de consumo por parte del periférico. En el proyecto el dispositivo Central será el móvil y el Periférico la contera, debido a que el móvil tiene mayor CPU.

La transferencia de los datos entre dispositivos BLE pertenece al nivel GATT y los datos de transmisión son organizados en Servicio y Característica como se puede observar en la siguiente figura (Jacopo, Taffoni, Santacatterina, Sannino, & Formica, 2017).

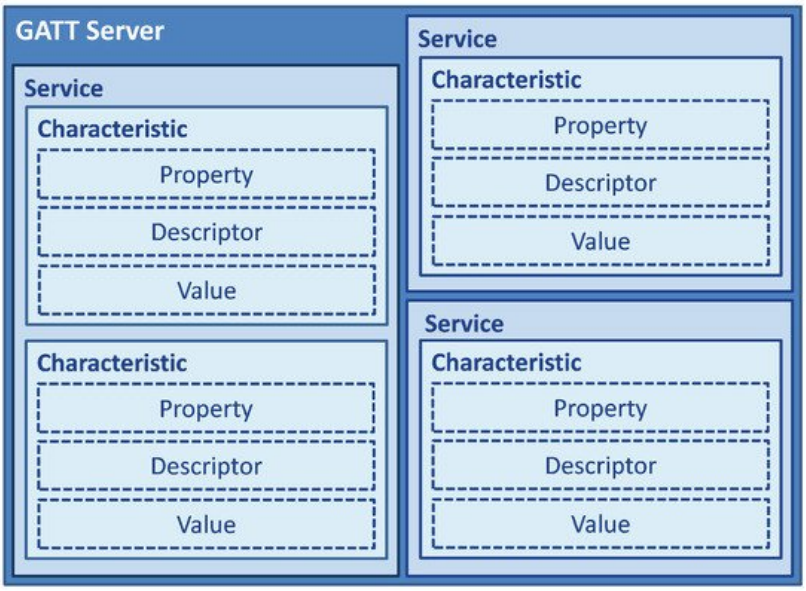

Figura 20. Servidor GATT.Fuente: (Jacopo, Taffoni, Santacatterina, Sannino, & Formica, 2017)

Service:

Los servicios son una colección de características, estos pueden tener una o múltiples características en su interior. Cada servicio se distingue por un único número asignado, este número se llama *identificador único universal (UUID)* y puede tener un tamaño de 16-bit o 128-bit.

Characteristics:

La característica es el nivel más bajo del concepto GATT. Dentro de este concepto se integran el valor de los datos, un descriptor que aporta información adicional del valor de la característica y algunas propiedades. Las propiedades indican que operaciones están permitidas y pueden ser las siguientes:

- Broadcast: permite enviar datos a los dispositivos BLE usando paquetes publicitarios.
- Readable: permite solo el modo lectura de los datos.
- Writable: únicamente permite al cliente escribir un nuevo valor en la característica.
- Notificable: permite a la persona recibir notificaciones si el servidor actualiza la característica, de este modo el cliente puede acceder al nuevo dato.

Cada característica se distingue por un nombre predefinido de 16o 128 bit UUID, al igual que en el servidor.

7.3 Protocolo de comunicación de la Contera Inteligente

Una vez definidos los conceptos más importantes del protocolo Bluetooth Smart se explicará la configuración desarrollada previamente a este proyecto en la contera inteligente.

La contera inteligente está configurada con un único servicio cuyo Identificador Único Universal (UUID) toma el nombre de 0x1000. Dentro de este servicio están integradas dos características con propiedades de notificación, es decir, que cuando haya una actualización de la característica el cliente recibe una notificación pudiendo leer el nuevo dato.

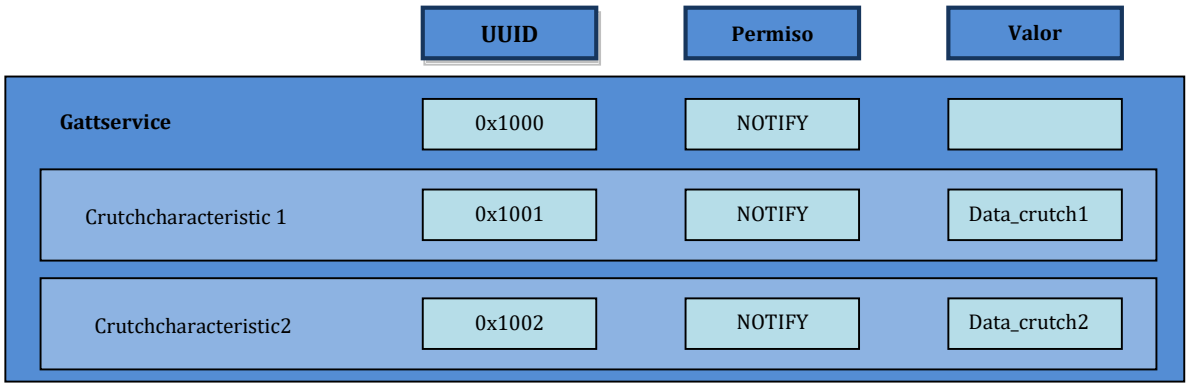

Figura 21. Identificadores universales y características. Fuente: (Gervais)

El tamaño máximo de la característica es de 20 bytes (160 bits) y los 14 valores que necesita la muleta transmitir, los cuales se han concretado en el apartado anterior.Por tanto, a fin de cumplimentar los 14 valores, son necesarios 317 bits compuestos estos por 309 bits de valores, y 8 bits de cada iteración que son usados para verificar la correcta recepción (Gervais).

En la siguiente tabla se puede observar en que característica está integrado cada dato. Además, se especifica en que número de byte está colocado.

Data_crutch1	
Número del byte	Composición
0	Iteración (4 bits) | Force (4 bits)
1	Force (8 bits)
2	Force (4 bits) | Altitude (4 bits)
3	Altitude (8 bits)
4	Altitude (8 bits)

5	Altitude (8 bits)
6	Altitude (4 bits) | Roll (4 bits)
7	Roll (8 bits)
8	Roll (8 bits)
9	Roll (4 bits) | Pitch (4 bits)
10	Pitch (8 bits)
11	Pitch (8 bits)
12	Pitch (8 bits)
13	Yaw (8 bits)
14	Yaw (8 bits)
15	Yaw (8 bits)
16	Yaw (4 bits) | Accx (4 bits)
17	Accx (8 bits)
18	Accx (8 bits)
19	Accx(4 bits) | Accy (4 bits)

Tabla 3. Tabla de datos por característica 1. Fuente: (Gervais)

Data_crutch2	
Número del byte	Composición
0	Iteración (4 bits) | Accy (4 bits)
1	Accy (8 bits)
2	Accy (8 bits)
3	Accz (8 bits)
4	Accz (8 bits)
5	Accz (8 bits)
6	Accz (1 bits) | Avx (7 bits)
7	Avx (8 bits)
8	Avx (5 bits) | Avy (3 bits)
9	Avy (8 bits)
10	Avy (8 bits)
11	Avy (1 bits) | Avz (7 bits)

12	Avz (8 bits)
13	Avz (5 bits) | Magx (3 bits)
14	Magx (8 bits)
15	Magx (5 bits) | Magy (3 bits)
16	Magy (8 bits)
17	Magy (5 bits) | Magz (3 bits)
18	Magz (8 bits)
19	Magz (5 bits)

Tabla 4.Tabla de datos por característica 2.Fuente: (Gervais)

7.4 Diseño de la aplicación móvil

Una vez conocido el medio al que se debe conectar y aprendida la codificación en la que se envían los datos se procede a explicar el diseño del entorno software. Primero se explicará la aplicación en un formato general, y posteriormente se analizará más detalladamente cada ventana.

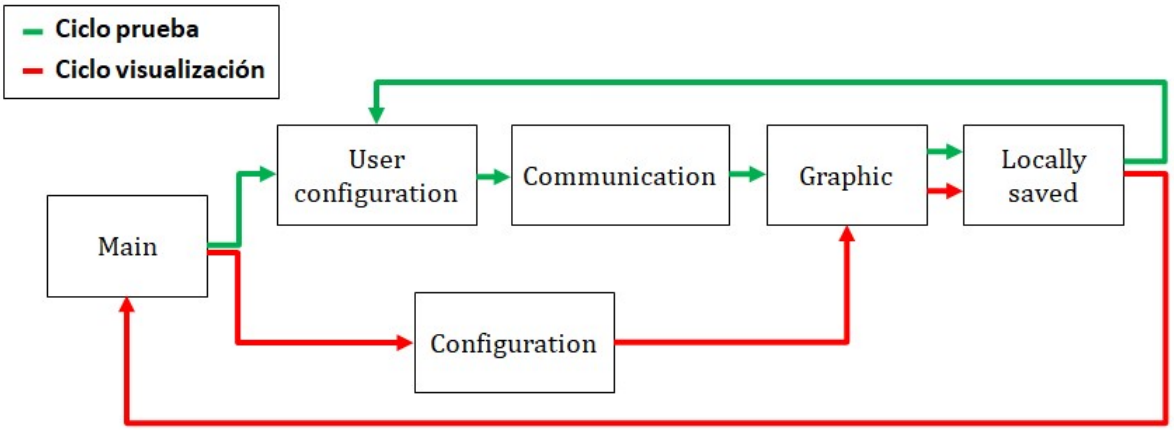

Figura 22. Ciclos de la aplicación

Al iniciar la aplicación se accede a la ventana ***main*** donde comienzan los ciclos. En la aplicación existen dos ciclos de ejecución diferenciados. Ambos ciclos comienzan en la ventana principal, y aunque si llegan a compartir parcialmente las ventanas, estas dos ejecuciones o ciclos de ejecución no se juntan en el resto de la aplicación.

7.4.1 Ciclo de Prueba

El primer ciclo, el circuito verde, el cual se identifica como ciclo prueba, tiene la funcionalidad de realizar y guardar las pruebas formalizadas a las personas usuarias. Durante el proceso se dispone de la opción de conectarse y adquirir los datos de la

muleta, fijar el número de usuario y de prueba, y de grabar los datos recibidos en la memoria interna y subirlos a la plataforma en la nube.

El ciclo cuenta con 4 ventanas y la principal que se ejecuta al inicio del proceso. En cada pantalla se realiza una, o una serie, de acciones quedan la importancia a cada una de ellas.

Al iniciar el programa se ejecuta la ***ventana principal***. Para iniciar el proceso de realizar una prueba se ha de introducir el MAC del dispositivo al que se quiere conectar, seguidamente se comprueba que la conexión ha sido exitosa y seda acceso a la siguiente ventana, user configuration.

En la ventana de ***configuración de usuario*** se introduce el número de identidad del usuario y el número de prueba para seguir avanzando en la aplicación. Al pedir avanzar se comprueba que los datos utilizados no tengan ninguna prueba registrada. En caso contrario, es decir, si existiesen datos previamente guardados de una prueba con el mismo número ID y de iteración, se solicitará la confirmación de que se quiere avanzar borrando todos los datos de la prueba registrada con anterioridad, y así avanzar a la siguiente pantalla: communication.

Al iniciar la ventana ***communication*** comienza la prueba. En esta ventana se dispone de botones que permiten comenzar, pausar y parar la prueba; el botón *cronómetro* que permite medir el tiempo; y varios botones que posibilitan una mejor visualización de los datos. Al pulsar al símbolo *play* comienza la adquisición de los valores que se muestran en la pantalla, estos valores varían rápidamente debido al gran muestreo que da la muleta. Es por ello, por lo que se ha integrado la opción de *visualizar los 10 últimos valores*. Esta opción se visualiza en formato gráfico para una mayor facilidad a la hora de interpretar la tendencia de los valores. Asimismo, en los datos que tienen más de un valor como son orientación, aceleración, velocidad de giro y campo magnético se dispone de la opción de *abrir una mini ventana* en el que se muestran los 3 valores del dato. En esta ventana podemos encontrar el botón *Pause* para pausar la prueba, al pausar la ejecución no se visualiza ni se adquiere ningún valor nuevo. Para terminar la prueba se debe de pulsar el botón *Stop*, de este modo termina la transmisión y adquisición de datos, y se da paso a la siguiente ventana: graphic.

La ventana ***Graphic*** consiste en la visualización de toda la serie de valores obtenidos en la ventana anterior. Se selecciona el dato que se quiere observar (fuerza, altura, orientación...) y este es plasmado en un gráfico, o en 3 gráficos, en el caso de que el dato tenga 3 ejes. Esta proyección se genera siendo el eje Y el valor del dato, y el eje X el número de la serie. En la ventana gráfico se dispone de la opción de volver a repetir la prueba y/o la de guardar los resultados. En caso de pulsar *volver* se retrocede a la ventana comunicación reiniciando la adquisición, es decir, se pierden todos los valores de la prueba, a fin de adquirir unos nuevos. Por otro lado, si se pulsa *guardar* se avanza a la siguiente ventana: saved locally.

La ventana ***saved locally***, o ventana de guardado, tiene como fin guardar los datos en la memoria interna y tener la posibilidad de enviar archivo. La primera acción que realiza al iniciarse es el almacenaje en la memoria interna, se guardan todos los valores de la serie, y los datos introducidos en la configuración de usuario; también el tiempo de cronometro y la fecha de hoy. En la pantalla se visualiza un resumen de los datos más característicos de la prueba y se dispone de 2 botones que permiten el envío de la prueba. El primero, *upload,* sube a una plataforma en la nube un archivo CSV con todos los datos registrados; mientras que el segundo botón, *send by email, crea* un email adjuntando el archivo CSV. Además, se dispone de un botón que lleva de nuevo a la ventana de configuración de usuario rellenando el ID, e incrementa la iteración en 1, con lo que agiliza realizar una nueva prueba al mismo usuario.

7.4.2 Ciclo de Visualización

El segundo ciclo, el circuito rojo, al cual se identifica como ciclo visualizar, tiene el objetivo de visualizar pruebas previas y disponer dela opción de subirlo a la plataforma en la nube. El proceso de visualización cuenta con 3 ventanas, además de la principal. Las dos últimas ventanas son compartidas con el ciclo de crear prueba, el ciclo verde. Sin embargo ciertas funciones que se pueden realizar en el otro ciclo en este caso desaparecen.

Para recuperar o poder visualizar pruebas previas desde la pantalla principal, ***main,*** se ha de avanzar a la ***pantalla de configuración*** mediante un botón. Una vez en la pantalla se ha de seleccionar tanto el *numero ID del usuario* como la *prueba*. Para no cometer una mezcla de números y facilitar el uso, primeramente se selecciona el usuario a fin de que aparezcan las pruebas vinculadas a ese registro.

Finalizada la selección se comienza la búsqueda y se avanza a la ventana ***graphic***. Aquí comienza la fase que comparte pantalla con el otro ciclo pero sin estar entre ellos enlazados. Una vez aquí, se visualiza la serie de datos de la prueba seleccionada al igual que se hacía en el otro ciclo, la diferencia está en que no hay botón de volver atrás y solo se puede avanzar a la pantalla guardar.

La ***pantalla guardar,*** a diferencia del ciclo anterior, tiene como objetivo enviar el archivo con los datos. Ambos ciclos comparten en esta ventana las mismas dos posibilidades de envío, subir a la nube y enviar por email. Por otro lado, debido a que los datos que están siendo observados ya están registrados en la memoria interna, esta ventana, a diferencia del ciclo verde, no da la opción de guardar. Además, otra de las diferencias entre ambos ciclos en esta última parte se da al finalizar la observación de datos, ya que al terminarla aplicación dirige a la pantalla principal pudiendo empezar el mismo proceso desde el inicio, o comenzar el registro de una nueva prueba.

7.5 Layout

Después de analizar el formato general se procede a analizar las pantallas individualmente desgranado todo tipo de detalle logrando conocer más el funcionamiento del programa.

El programa cuenta con 6 pantallas:

7.5.1 Ventana main

La ventana main es la pantalla principal, es decir, la pantalla con la que se inicia la aplicación. El objetivo principal de esta ventana es conectarse con la contera sensorizada para iniciar el proceso de realización de una prueba. Asimismo, esta ventana dispone también de la opción de ir a la ventana configuración.

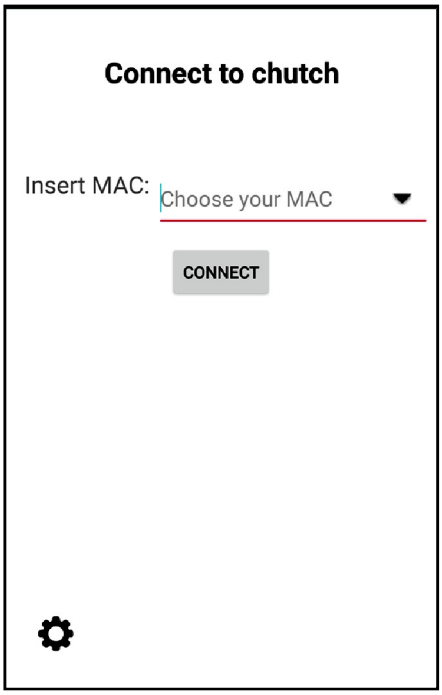

Figura 23. Pantalla principal

Para comenzar el proceso de realización de una prueba a un usuario primero debe establecerse la conexión con la contera. Para ello, se debe introducir el Médium Access Control (MAC) del dispositivo inteligente. Es importante señalar que cada dispositivo BLE tiene su propio nombre MAC. En el momento de introducir el nombre se disponen de dos alternativas:

Por un lado, la inscripción manual. En esta primera alternativa, como su propio nombre indica, se introduce manualmente el identificador de la contera. En caso de que el identificador que se esté introduciendo esté ya guardado en la memoria, al escribir un carácter se dará la opción de seleccionarlo, ya que será reconocido.

Por otro lado, se puede escribir mediante el desplazamiento de lista, en este caso se desplegarán los nombres MAC de todos los dispositivos previamente guardados en la memoria interna, de este modo se logra una mayor rapidez en caso de que se hubiese usado la app con anterioridad.

Al pulsar el botón *connect* si tiene un nombre MAC seleccionado comienza la búsqueda y la conexión con la contera que dispone de ese MAC. En caso de que se conecte con el dispositivo se accederá a la siguiente ventana; sin embargo, en caso contrario, permanecerá en la misma ventana mostrándose un mensaje temporal en el que se indica que no se ha podido conectar por cualquier razón. Además, si el nombre MAC introducido no está en la base de datos se procederá a guardar la nueva dirección, por lo tanto, solo se agregará el nombre MAC a la base de datos en caso de que se haya conseguido conectar una vez con el dispositivo.

A modo de síntesis el proceso seguido en la pantalla queda de la siguiente forma:

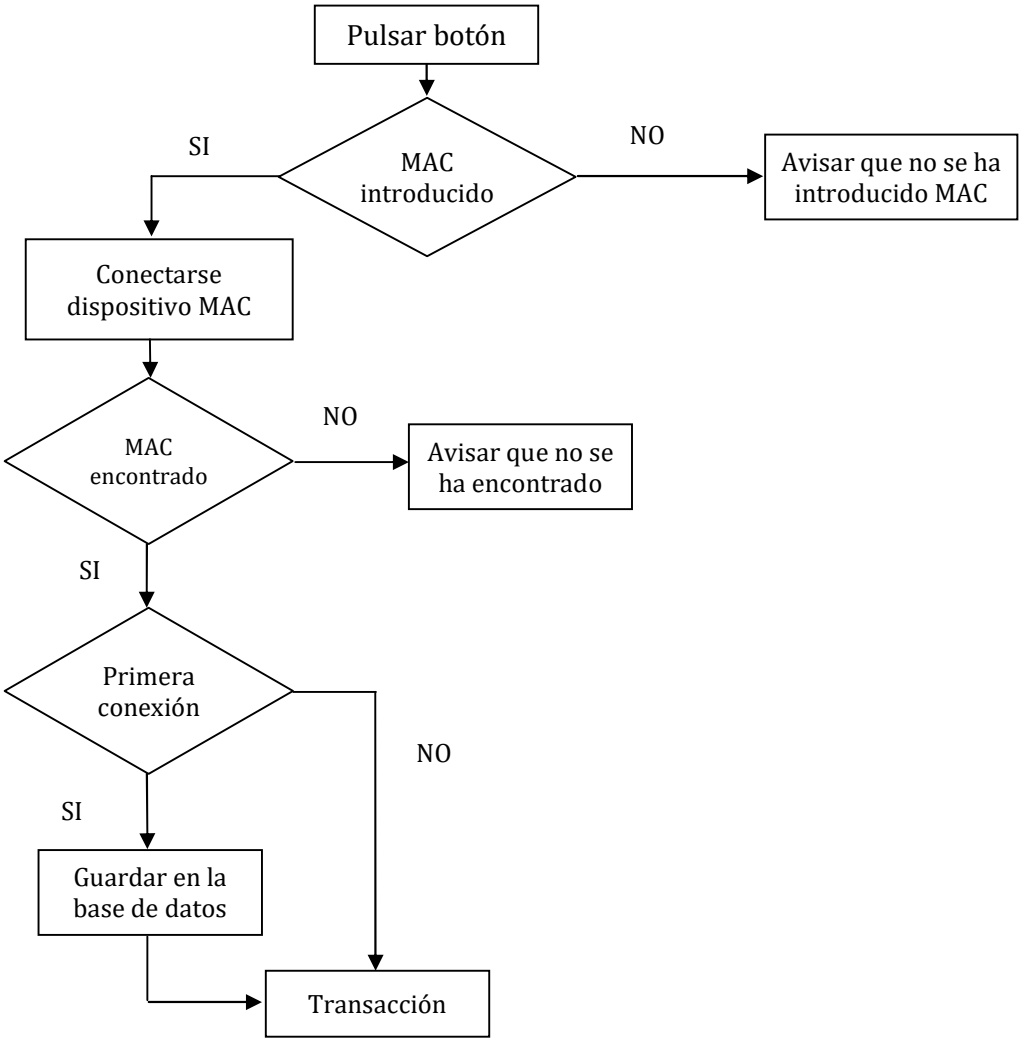

Figura 24. Esquema ventana Main

Por último, es importante resaltar que para acceder a la ventana de configuración, desde esta ventana, se ha de pulsar el botón con forma de engranaje.

7.5.2 Ventana User Configuration

La ventana configuración de usuario es la pantalla previa a conectarse y transmitir con la contera. El fin de la pantalla es definir la ID y la iteración del usuario. Con la definición de estas dos cualidades se puede saber a quién pertenecen y, en concreto, a qué prueba corresponden, los datos almacenados.

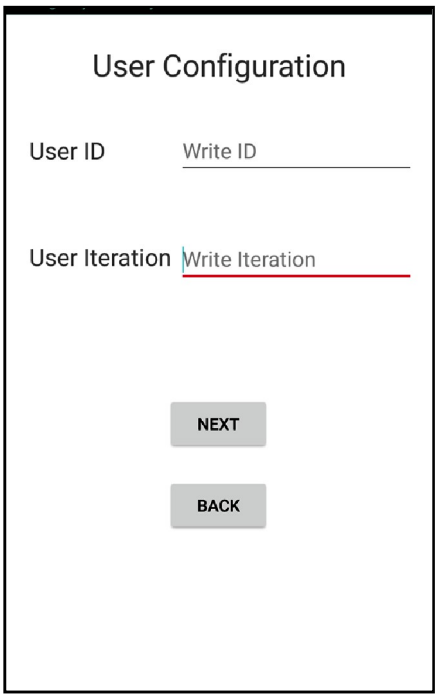

Figura 25. Pantalla configuración de usuario

Por lo tanto, introducir los dos valores es un paso imprescindible en el proceso. Para facilitar este paso se ha desarrollado un sistema que en caso de haber realizado una prueba previa la aplicación lo detecta y rellena automáticamente los campos, dependiendo de la pantalla de la que procede.

Por esta cuestión, la pantalla puede ser rellenada de diversas formas dependiendo de la procedencia, pero todos los casos disponen de la opción de modificar los campos. Estos son los cambios que se pueden dar según la procedencia:

- Si se viene de Main: Hay que introducir los conceptos de forma manual al no rellenase automáticamente.
- Si se viene de Communication: Se introducen automáticamente los mismos valores que se habían introducido a la hora de hacer la prueba, concluyendo que

por alguna razón no se quiere hacer la prueba o se quiere comenzar desde el principio.

- Si se viene de Saved locally: Se introduce automáticamente incrementando en uno el valor de la iteración, facilitando el avance para realizar la siguiente prueba al mismo usuario.

Para introducir o modificar el valor de alguno de los conceptos se dispone de un teclado numérico que se abre al pulsar sobre el campo. El teclado se ha configurado para no permitir la introducción de un carácter que no sea de formato numérico.

Al pulsar el botón *Next* comienza el proceso para verificar que los datos introducidos no estén registrados y poder avanzar a la siguiente pantalla. Se comprueba en la base de datos que no se va a añadir un registro duplicado, para ello se verifican todos los registros previos, comparando los valores ID e iteración.

Si no hay registro previo se avanza a Communication, en caso contrario, se activa un *diálogo de alerta*. La alerta comunica que se ha encontrado un registro anterior con los mismos datos y consulta si se quiere continuar borrando todos los datos del registro, perdiendo los valores de la prueba o si, por el contrario, no desea borrarlos.

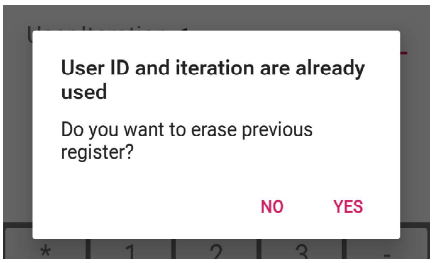

Figura 26. Diálogo de alerta

Por último, se da la opción de retroceder a la pantalla principal pulsando el botón *back*.

7.5.3 Ventana communication

La ventana de comunicación es la pantalla más importante de la aplicación, en esta pantalla se realiza la conexión y transmisión con la contera sensorizada. Además, se dispone de funciones que permiten una mejor visualización de los datos.

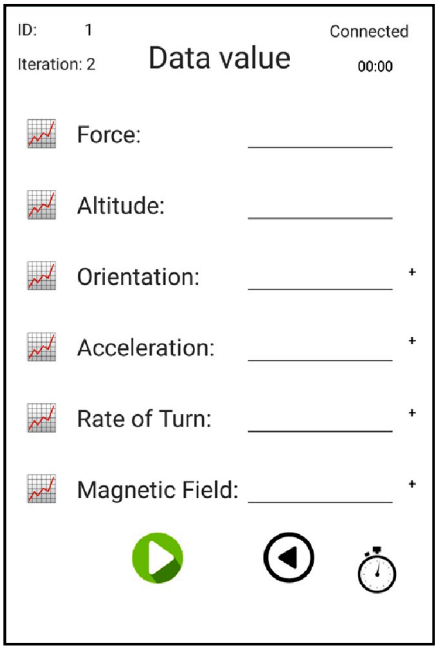

Figura 27. Ventana Communication

Al iniciar la ventana communication comienza la conexión con la contera inteligente. Para iniciar la prueba se dispone del botón *play* que al pulsarlo comienza la adquisición de los datos y el muestreo en la pantalla. Además, se disponen de otros botones para el control de la adquisición que varían dependiendo el seleccionado.

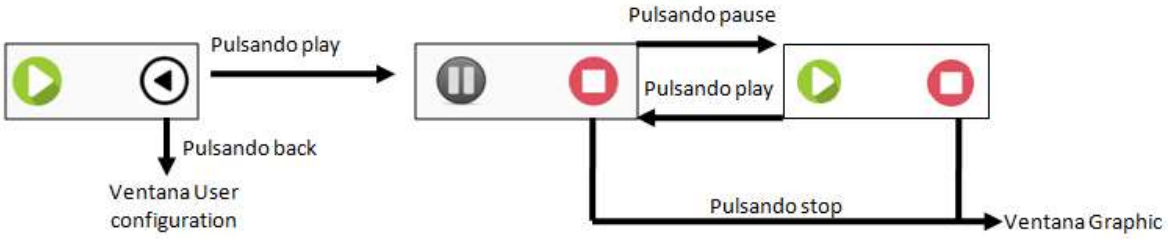

Figura 28. Botones de ejecución

El botón *pause* paraliza la captación de datos y queda en la espera de volver a reanudar la prueba o finalizarla.

Para finalizar la prueba hay que pulsar el botón *stop*, deja de adquirir datos y avanza a la ventana Graphic.

En el caso de querer volver a la ventana configuración usuario se dispone del botón *back,* pero siempre antes de haber iniciado la transmisión de datos.

En la parte central se muestran los últimos valores obtenidos; para una mejor visualización de estos se han incorporado dos métodos:

Pulsando el botón *gráfico* se despliega un gráfico donde se muestran los últimos 10 valores. Mediante este método se puede analizar la tendencia que está teniendo el dato.

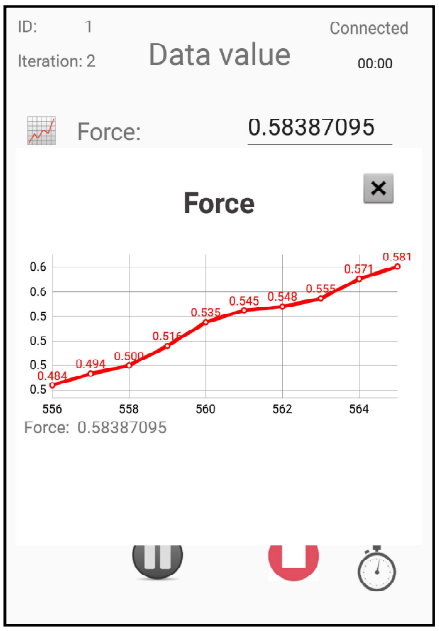

Figura 29. Mini ventana gráfico

En los datos que tienen 3 ejes tienen un botón cruz que al pulsar se abre una mini ventana que permite una mejor visualización de los 3 ejes.

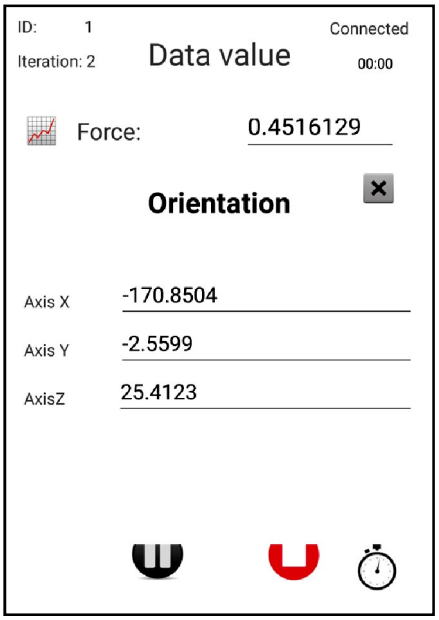

Figura 30. Mini ventana zoom

La ventana tiene a su disposición un cronómetro para medir simultáneamente la prueba. Su código se analizará posteriormente en la sección 7.6.5.

7.5.4 Ventana graphic

La pantalla gráfico tiene como objetivo visualizar todos los valores de los datos adquiridos de la prueba.

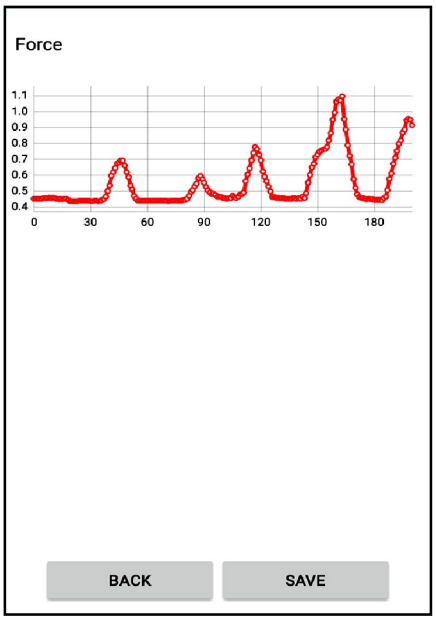

Figura 31. Ventana Graphic

Seleccionando el dato se visualiza la serie de valores. Debido a que algunos datos tienen 3 ejes, en ese caso, se visualizan 3 gráficos independientes lo que facilita una mejor apreciación de los resultados.

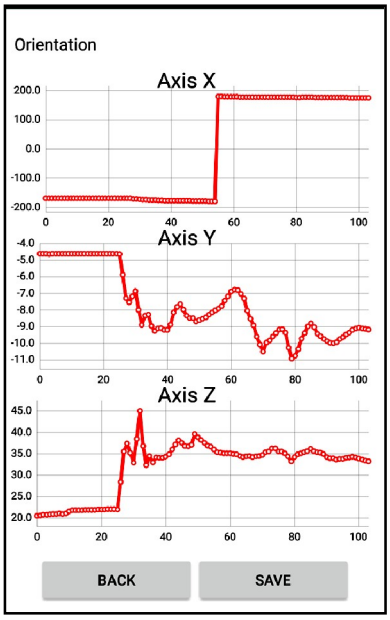

Figura 32. Ventana Graphic II

En el ciclo prueba, al final de la pantalla se dispone de 2 botones:

El botón *Back* finaliza la ventana y retrocede a la ventana anterior: ventana communication. Al retroceder se pierden todos los datos de la captura por lo que la adquisición de los datos comenzarían desde el inicio.

El botón *Save* finaliza la ventana y avanza a la ventana Saved locally, última ventana del ciclo.

Mientras que en el ciclo visualización solo aparece el botón *Next* cuya función es la misma que el botón *Save*: el avance a la siguiente pantalla.

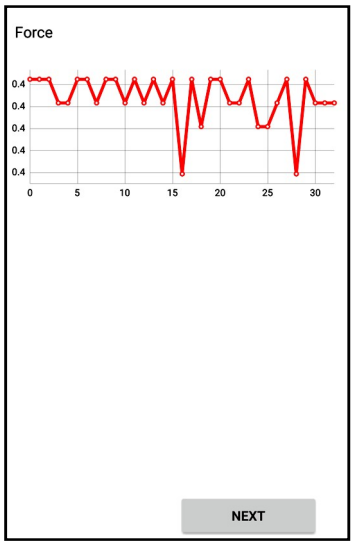

Figura 33. Ventana Graphic ciclo visualización

7.5.5 Ventana saved locally

La ventana saved locally es la última pantalla de los dos ciclos que tiene la aplicación. La función principal en la pantalla es guardar internamente la prueba, aunque también dispone de las opciones de subir a la nube o enviar por email.

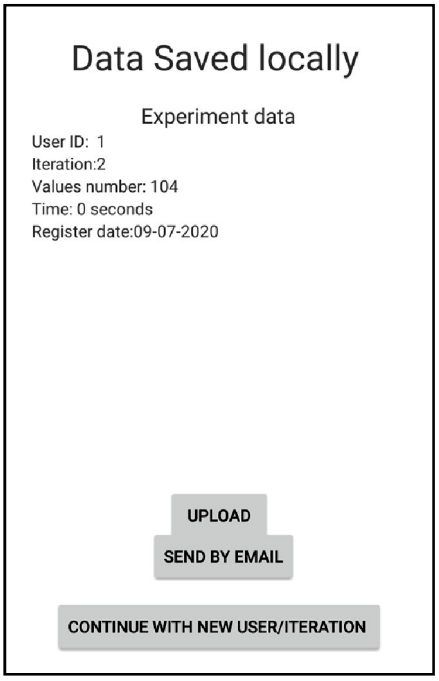

Figura 34. Ventana Saved locally

En la pantalla se visualizan los datos de la prueba.

Se dispone de dos botones, Upload y send by email, cada cual tiene una función pero los dos transfieren los archivos. El botón *upload* sube un archivo CSV con los datos de la prueba a la plataforma en la nube Firebase, mientras que el botón *send by email* crea un correo electrónico con el archivo adjunto y el título y la descripción escrita, está solo a falta de escribir el destinatario para enviarlo.

El botón *continue with user/iteration* finaliza la ventana y devuelve a la ventana de la configuración de usuario, permitiendo realizar una nueva prueba. En el caso del ciclo visualización, el botón cuyo nombre es *Finish review* devuelve a la pantalla principal.

7.5.6 Ventana configuration

La ventana da de configuración es la pantalla donde se selecciona la prueba que se quiere visualizar.

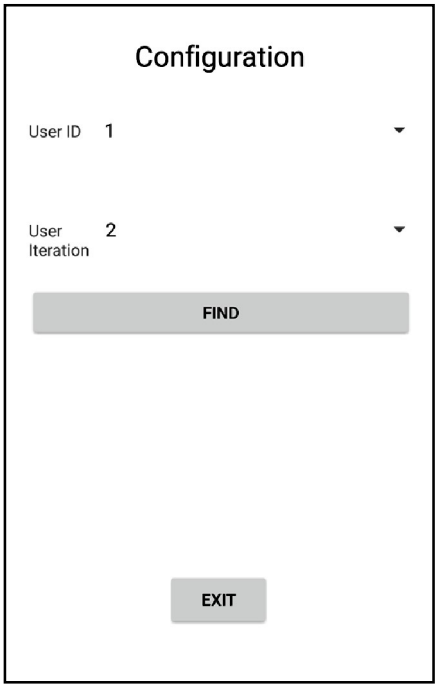

Figura 35. Ventana Configuration

Para ello se dispone de dos listas desplegables, la primera define la identidad del usuario y el segundo el número de iteración. Ambas están interrelacionadas, por lo que al seleccionar el usuario la lista que se despliega de las iteraciones sufre un filtrado dejando únicamente seleccionar las vinculadas al mismo. De este modo se evita cometer errores de selección de prueba.

El botón *Find* inicia la búsqueda recuperando las series de datos de la prueba seleccionada. Una vez obtenida avanza a la ventana de gráficos.

Para volver a la pantalla principal se dispone del botón *Exit*.

7.6 Código

Analizado el funcionamiento de cada pantalla se va a explicar el desarrollo realizado para cumplir las funciones de cada pantalla.

7.6.1 Configuración Android Studio

Android Studio necesita una configuración previa para que la aplicación pueda funcionar. Al crear un nuevo proyecto, Android Studio genera automáticamente un archivo que configura permisos, nombres, etc.

Hay permisos que hay que incorporar para el correcto funcionamiento de la aplicación. La aplicación necesita la implementación del Bluetooth y para su correcta utilización necesita administrar los siguientes 3 permisos:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figura 36 Permisos bluetooth

Mientras que para crear el archivo CSV se ha visto necesario implementar los permisos de escribir y leer en el almacenaje exterior.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Figura 37. Permisos almacenaje externo

7.6.2 Base de datos

La base de datos es el apartado importante en la aplicación, ya que, debe almacenar los datos de la lista de MAC conectadas, la información relacionada al usuario y los datos de la pruebas.

7.6.2.1 Tablas

Debido a que se necesita almacenar diferentes datos en el proyecto se han creado 3 tablas:

MAClist: tabla que almacena todos los MAC que se han conectado con la aplicación. Gracias a este guardado no se deberá meter en cada uso el nombre de la MAC con lo que se agilizará la conexión. Para ello, la tabla cuenta con:

Nombre	Función
ID_MAC	Número de registro
MAC_number	MAC del dispositivo con el que se ha conectado

Tabla 5. Tabla Maclist

User: tabla que guarda los datos de la configuración del usuario y datos de la prueba. Para ello, la tabla cuenta con:

Nombre	Función
_ID	Número de registro
ID	Número de usuario
Iteration	Número de prueba
Date	Fecha en la que se realiza la prueba
Time	Tiempo registrado por el cronometro.

Tabla 6. Tabla user

Datavalue: tabla que almacena los valores transmitidos por la muleta, por lo tanto, se registran los 14 valores de la muleta e ID_user que es la conexión con la información guardada del usuario, es decir, es la conexión con la tabla user.

Nombre	Función
ID_value	Número de registro
ID_user	Número de registro de la prueba realizada
force	Valor force
altitude	Valor altitude
roll	Valor Roll
pitch	Valor pitch
yaw	Valor yaw
accx	Valor accx
accy	Valor accy
accz	Valor accz
gyrx	Valor gyrx
gyry	Valor gyry
gyrz	Valor gyrz
magnx	Valor magnx
magny	Valor magny
magnz	Valor magnz

Tabla 7. Tabla datavalue

7.6.2.2 Guardar, Buscar o Eliminar

Antes de realizar cualquier ejecución en la base de datos, hay que llamar a la clase donde se han creado las tablas.

```
AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(getApplicationContext(), name: "administracion", factory: null, version: 1);
SQLiteDatabase Basededatos = admin.getWritableDatabase();
```

Figura 38. Abrir base de datos

Tras ello, dependiendo de la ejecución que se quiere realizar el proceso varía.

7.6.2.3 Guardar

Esta ejecución guarda los datos en la base de datos. Se debe de definir la variable donde se guarda todo el contenido de valores creando una fila en la tabla.

```
ContentValues registros = new ContentValues();
registros.put("MAC_number", myMAC_Address);
Datebase.insert( table: "MAClist",  nullColumnHack: null, registros);
```

Figura 39. Guardar registro

7.6.2.4 Buscar

Esta ejecución permite buscar en la base de datos todos los registros que tienen esa condición. Una vez que se ha abierto la base de datos, se añaden a un cursor todos los registros.

Para recuperar el valor hay que usar el cursor, obtener Int y definir el nombre de la columna.

```
Cursor cursor = Basededatos.rawQuery( sql: "select * from user WHERE ID =? AND iteration =? ", new String[]{bus_ID,bus_iteration});
cursor.moveToFirst();
IDregister=cursor.getInt(cursor.getColumnIndex( columnName: "_ID"));
```

Figura 40. Buscar registro y obtener valor

7.6.2.5 Eliminar

Esta ejecución permite eliminar todos los registros que tengan las mismas condiciones que se especifiquen. Abierta la base de datos se da la orden de borrar eligiendo en que tabla y la condición que se le aplica.

```
Basededatosdata.execSQL("DELETE  from datavalue  WHERE ID_user =?", new String[]{String.valueOf(IDregister)});
Basededatosdata.close();
```

Figura 41. Borrar registro

7.6.3 Ventana Main

La pantalla principal comienza a ejecutarse una vez se inicializa el programa. Es en ese momento en el que se desarrolla el siguiente flujo:

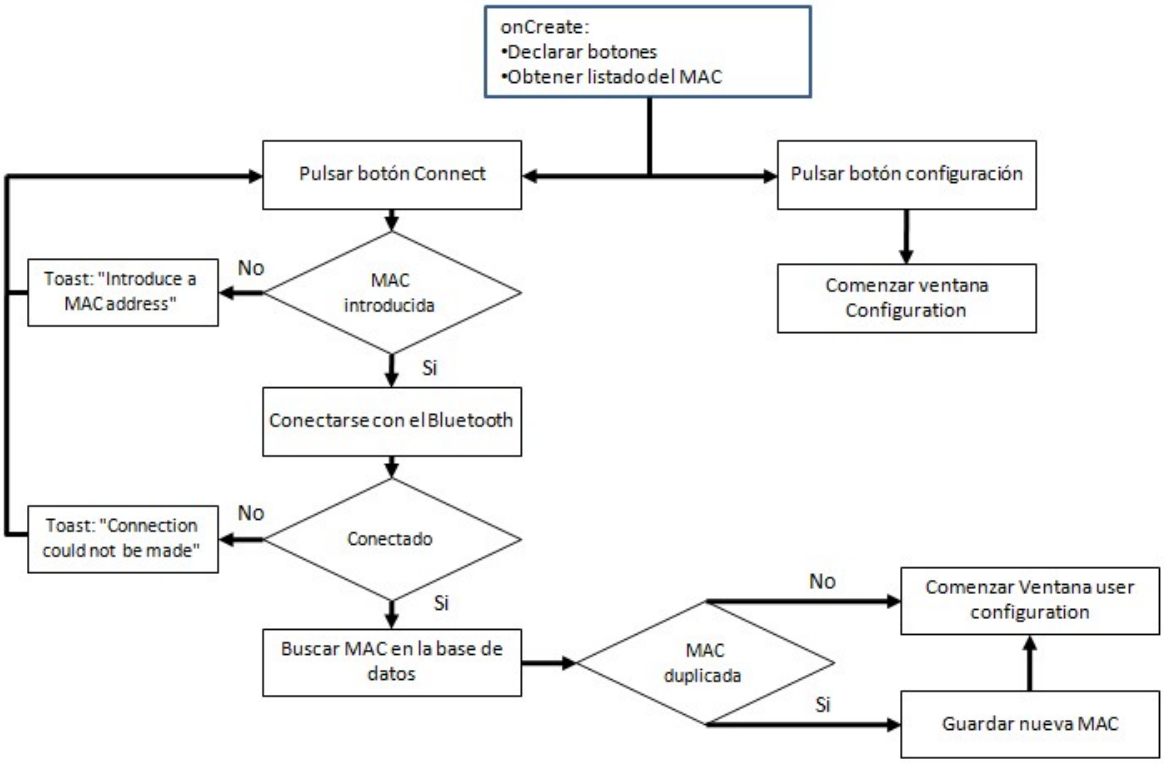

Figura 42. Flujo ventana Main

Las funciones principales se basan en el manejo de la base de datos, las cuales han sido previamente explicadas en un formato general. Lo más interesante de la pantalla main es el análisis de la conexión al bluetooth, pero como se trata de una función que se da en la pantalla communication, se pasará a explicar en ese punto, ya que se analizará el sistema al completo. Así, en este apartado se analizan otros aspectos de menor arraigo.

Toast

Son mensajes temporales que aparecen en la pantalla en una duración corta. A la hora de la configuración hay que indicar el contexto, el mensaje y la longitud del mensaje.

```
Toast.makeText(getApplicationContext(), text: "Introduce a MAC address", Toast.LENGTH_SHORT).show();
```

Figura 43. Toast

Iniciar otra actividad

A la hora de pasar de pantalla el comando a utilizar es Intent en el que se deben identificar el contexto en el que se ubica y la nueva ventana que se quiere iniciar. Además, añadiendo *put Extra* se pueden pasar variables de una ventana a otra.

```
Intent i = new Intent( packageContext: MainActivity.this, Choose_ID_Iteration.class);
i.putExtra( name: "MAC_Adress", myMAC_Address);
startActivity(i);
```

Figura 44. Avanzar a otra ventana

Mientras que para adquirir el valor de la variable se usa el siguiente sistema.

```
String myMAC_Address = getIntent().getExtras().getString( key: "MAC_Adress");
```

Figura 45. Adquirir variable de otra ventana

7.6.4 Ventana User configuration

Al comenzar la ventana de configuración de usuario se desarrolla el siguiente flujo:

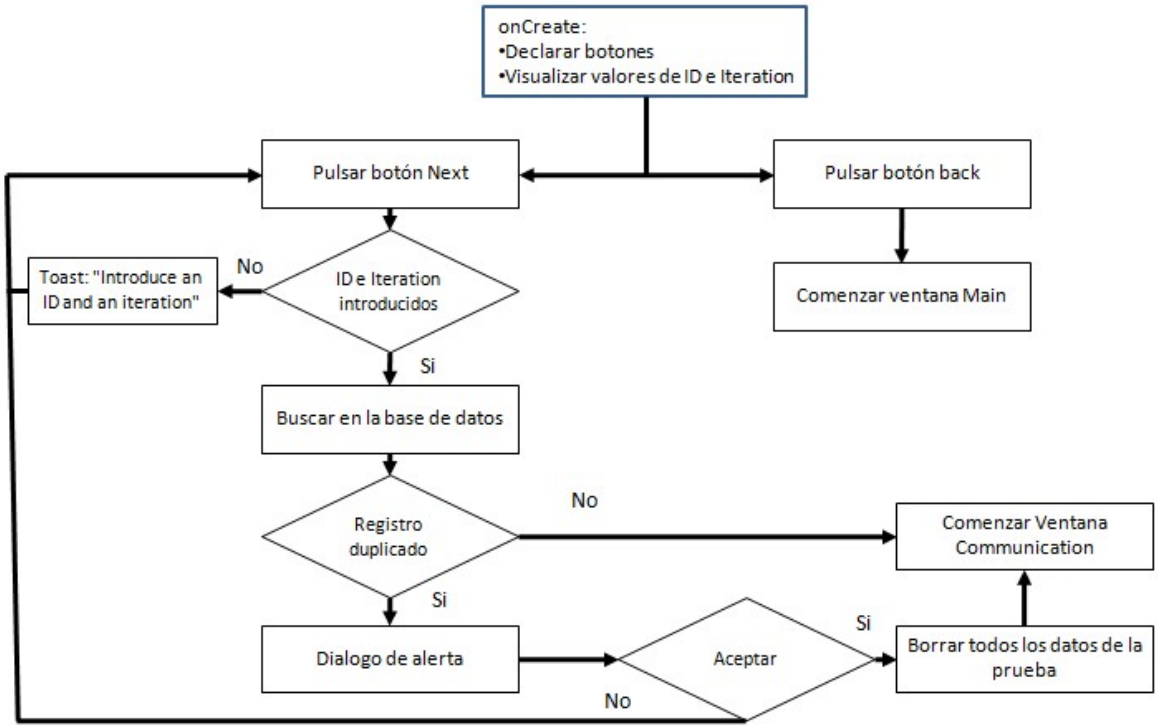

Figura 46. Flujo ventana User configuration

En esta ventana lo que destaca es la búsqueda en la base de datos la cual se ha analizado en el apartado 7.6.2.4 algo nuevo a destacar es la programación del diálogo de alerta.

Dialogo de Alerta

El dialogo de alerta ha sido implementado para advertir al usuario de la aplicación de que en caso de avanzar se pierden datos.

Como se observa en la Figura 47, el desarrollo de esta función es esquemático. Hay que definir el título y el mensaje que debe aparecer y definir los botones que aparecen en la alerta. En los dos casos del proyecto, se han definido dos botones, pulsando el botón *Yes* comienza la siguiente actividad llevando consigo ciertas variables importantes tratadas en la ventana actual.

```
public void alertTwoButtons() {
    new AlertDialog.Builder( context: Graphic.this)
            .setTitle("Are you sure you want to go back?")
            .setMessage("All data values will be deleted without save it")
            .setPositiveButton( text: "YES",
                    new DialogInterface.OnClickListener() {
                        @TargetApi(11)
                        public void onClick(DialogInterface dialog, int id) {

                            Intent i = new Intent( packageContext: Graphic.this, Communication.class);
                            i.putExtra( name: "MAC_Adress",getIntent().getExtras().getString( key: "MAC_Adress"));
                            i.putExtra( name: "ID",getIntent().getExtras().getInt( key: "ID"));
                            i.putExtra( name: "Iteration",getIntent().getExtras().getInt( key: "Iteration"));
                            startActivity(i);
                            finish();

                        }
                    })
            .setNegativeButton( text: "NO", (dialog, id) -> {
                    dialog.cancel();
            }).show();
}
```

Figura 47. Código dialogo alerta

7.6.5 Ventana Communication

Al comenzar la ventana de communication se desarrolla el siguiente flujo. Nótese que además de la función principal, existen numerosos flujos de código asíncronos, dependientes de las acciones y botones varios:

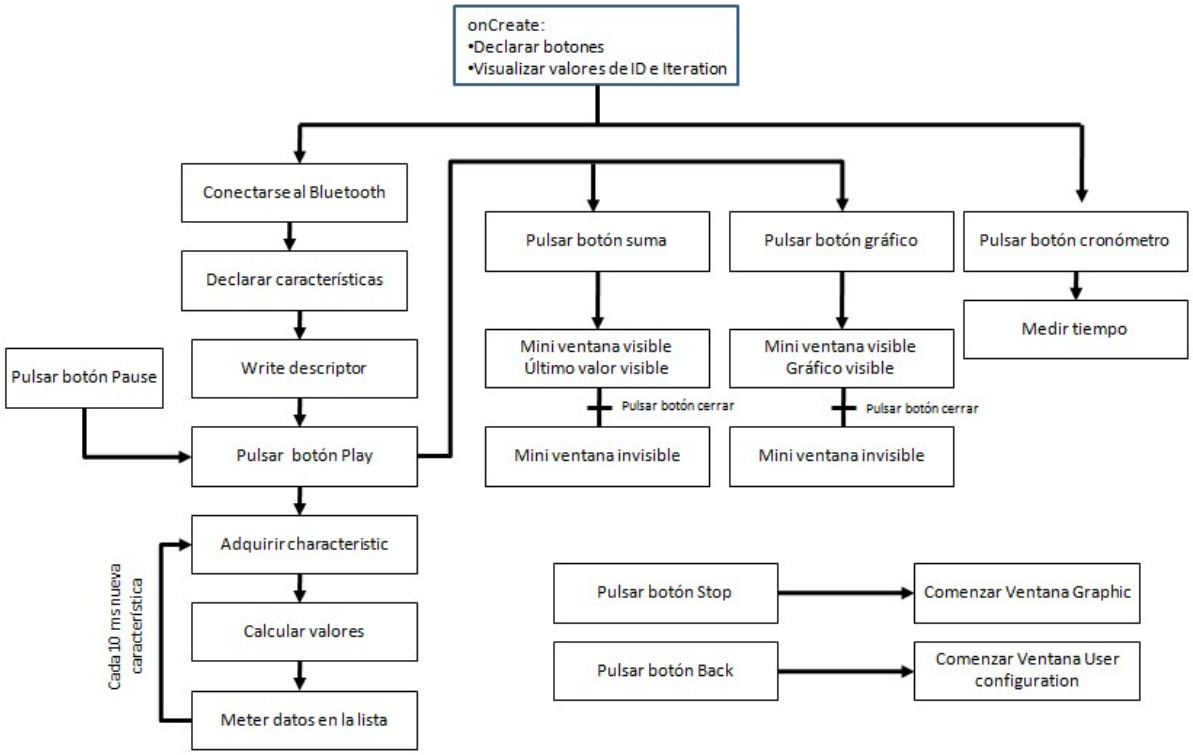

Figura 48. Flujo ventana communication

La tarea principal de esta ventana es adquirir los datos y por ello esta tarea tiene una gran importancia. Por este motivo en el siguiente apartado se va a explicar el desarrollo de la tarea de conexión y la transmisión de datos entre la contera sensorizada y la aplicación.

Conexión y transmisión con la contera

A la hora de conectarse con la contera el primer paso es conectarse con el dispositivo para ello se usa el método configureConnect. Adquiere el nombre del MAC e intenta conectarse.

```
public void configureConnect() {
    myMAC_Address = getIntent().getExtras().getString( key: "MAC_Adress");
    myBluetoothDevice = myBluetoothAdapter.getRemoteDevice(myMAC_Address);
    myBluetoothGatt = myBluetoothDevice.connectGatt( context: Communication.this,  autoConnect: false, myGattCallback);
    Log.i(TAG,  msg: "try to connect to crutch: " + myMAC_Address);

}
```

Figura 49. Método configure Connect

En el ejemplo de la Figura 49 el MAC es obtenido de la pantalla previa debido a que la inserción se hace en la pantalla principal.

Una vez conectado, comienza el método asíncrono myGattCallback en el que se activan acciones dependiendo del estado del Bluetooth. Primero, como el estado de bluetooth ha variado, de desconectado a conectado, se ejecuta el método onConnectionSatateChange, que al conectarse manda ejecutar el método discoverService.

```
@Override
// if the bluetooth Connection state has changed
public void onConnectionStateChange(final BluetoothGatt gatt, int status, int newState) {
    super.onConnectionStateChange(gatt, status, newState);

    if (newState == BluetoothProfile.STATE_CONNECTED) {
        connectionState = STATE_CONNECTED;
        Log.i(TAG,  msg: "Connected to Gatt Server");
        gatt.discoverServices();
        Log.i(TAG,  msg: "Discover Services");

    } else if (newState == BluetoothProfile.STATE_DISCONNECTED) {
        connectionState = STATE_DISCONNECTED;
        Log.i(TAG,  msg: "Disconnected to Gatt Server");
        gatt.close();
    }
    showConnectionState();
}
```

Figura 50. Método onConnectionStateChange

En el método onServiceDiscover se declaran las características 1 y 2 y se añaden a una lista de características.

```
characteristic_1 = gatt.getService(Service_1_UUID).getCharacteristic(characteristic_1_UUID);
characteristic_2 = gatt.getService(Service_1_UUID).getCharacteristic(characteristic_2_UUID);
chars.add(characteristic_1);
chars.add(characteristic_2);

subscribeToCharacteristics(gatt);
```

Figura 51. Declarar características

En el método subscribeToCharacteristics se obtiene la primera característica, characteristic 1, y se le da permisos para notificar a la característica, se declara CCD y se escribe el descriptor. Además, una vez escrito se borra la característica usada de la lista.

```
private void subscribeToCharacteristics(BluetoothGatt gatt) {
    BluetoothGattCharacteristic characteristic = chars.get(0);
    gatt.setCharacteristicNotification(characteristic, enable: true);
    CCCD = gatt.getService(Service_1_UUID).getCharacteristic(CCCD_UUID);
    BluetoothGattDescriptor descriptor = characteristic.getDescriptor(CCCD_UUID);
    if (descriptor != null) {
        descriptor.setValue(BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);
        gatt.writeDescriptor(descriptor);
        chars.remove( index: 0);
        Log.i(TAG,  msg: "descriptor writed "+ descriptor);
    }
}
```

Figura 52. Escribir el descriptor

Cuando se ha escrito el descriptor se activa el método onDescriptorWrite y se manda que realice el mismo proceso que antes, pero esta vez con la característica que queda en la lista, Characteristic 2.

```
@Override
public void onDescriptorWrite(BluetoothGatt gatt, BluetoothGattDescriptor descriptor, int status) {
    super.onDescriptorWrite(gatt, descriptor, status);
    Log.i(TAG,  msg: "Descriptor write find");
    subscribeToCharacteristics(gatt);
}
```

Figura 53. Método onDescriptorWrite

Mientras cada vez que el método onCharacteristicChange sienta un cambio de característica se activa y en caso que se le haya dado al botón *play* para adquirir datos, se activa el método broadcastUpdate.

```
@Override
// as soon as the characteristic has changed, this method is called
public void onCharacteristicChanged(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic) {
    super.onCharacteristicChanged(gatt, characteristic);
    Log.i(TAG,  msg: "characteristic change "+ characteristic);
    if(Running == true) {
        broadcastUpdate(characteristic);
    }
```

Figura 54. Método onCharacteristicChange

Dependiendo de la característica que haya recibido el cambio se adquiere el conjunto de byte y se añaden a la variable char, donde están los datos de la característica y se envía un mensaje al handle para que separe los datos y haga la conversión requerida para que la variable sea float.

```
public void broadcastUpdate(BluetoothGattCharacteristic characteristic) {
    if (characteristic.equals(characteristic_1)) {
        char_1 = characteristic_1.getValue();
        descriptorname = "desc1";
        handler1.sendEmptyMessage( what: 0);
        ArrayForce.add(force);
        ArrayAltitude.add(altitude);
        ArrayRoll.add(roll);
        ArrayPitch.add(pitch);
        ArrayYaw.add(yaw);
        ArrayAccX.add(accx);
        ArrayAccY.add(accy);
        ArrayAccZ.add(accz);
        ArrayRateX.add(turnx);
        ArrayRateY.add(turny);
        ArrayRateZ.add(turnz);
        ArrayMagneticX.add(magneticx);
        ArrayMagneticY.add(magneticy);
        ArrayMagneticZ.add(magneticz);

    } else if (characteristic.equals(characteristic_2)) {
        char_2 = characteristic_2.getValue();
        descriptorname = "desc2";
        handler1.sendEmptyMessage( what: 0);
    }
}
```

Figura 55. Adquisición de las características

En este punto se guardan en arraylist los valores float de cada dato para luego poder manejarlas y guardar en la base de datos.

Al enviar el conjunto de byte al método handle da comienzo el proceso de separación de los datos. Debido a la composición de la estructura, explicada en el apartado 7.3, los datos están unidos y deben ser separados creando una variable para cada uno de ellos.

```
force_i = bitExtracted(char_1[0], k: 4, p: 1) << 12 | (char_1[1] & 255) << 4 | bitExtracted(char_1[2], k: 4, p: 5);
```

Figura 56. Extraer el dato del char

En el proceso de separación se obtienen los bits del dato y cada uno de ello es colocado en su posición creando el valor integer del dato.

Asimismo, hay casos en el que dos datos distintos comparten el mismo número de byte por lo que se extrae la parte correspondiente al dato indicando la posición y la longitud.

```
static int bitExtracted(int number, int k, int p) {

    return (((1 << k) - 1) & (number >> (p - 1)));
}
```

Figura 57. Extraer bits de un byte

Una vez obtenido el valor integer se realiza una operación para convertirlo en un valor float.

```
force = force_i;

force = force / (1024 * 10 / 33);
```

Figura 58. Obtener el valor float del dato

Por último, se visualiza en la pantalla.

```
TextView TV_Force = (TextView) findViewById(R.id.ET_Force);
TV_Force.setText("" + force);
```

Figura 59. Visualizar datos en la pantalla

Cronómetro

Otra de las funciones destacables es la del cronómetro al ser una función única ya que solo aparece en esta pantalla.

El cronómetro es una función que sirve para medir el tiempo. Usando un mismo botón se logra iniciar, pausar y reiniciar el tiempo. Clicando el botón se inicia o se pausa, mientras que si se mantiene pulsado un rato se resetea. Por lo tanto, el flujo queda de la siguiente manera:

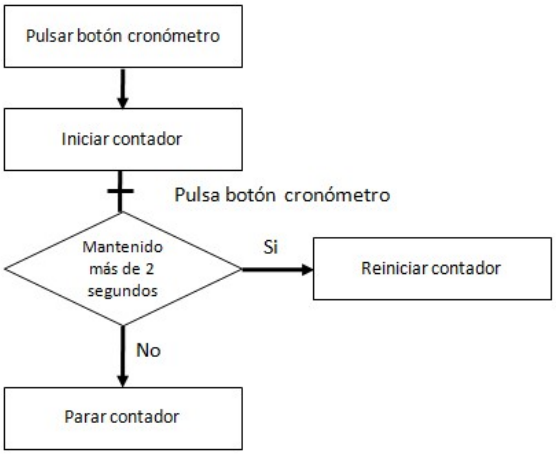

Figura 60. Flujo función cronómetro

Para favorecer el conocimiento del estado de ejecución del cronómetro se le han asignado colores al botón dependiendo de su empleo, cuando está ejecutando está en verde, y en rojo cuando está parado.

```
if(!ChronoRun){ click = false;btn_start.setColorFilter(Color.GREEN); }
else{click =true;btn_start.setColorFilter(Color.RED);}
```

Figura 61. Asignar color al botón cronómetro

En el caso de inicializar o reanudar el tiempo, se asigna a la variable chronometer los milisegundos desde el arranque quitando el tiempo que ha estado ejecutando. Además, se pone en verdadero ChronoRun para que la siguiente vez que se pulse al botón sea para parar.

```
if(!ChronoRun && !click) {
    chronometer.setBase(SystemClock.elapsedRealtime()-Chronostop);
    chronometer.start();
    ChronoRun = true;
}
```

Figura 62. Cronómetro play

En la segunda pulsada al estar las dos variables en verdadero se paraliza el cronómetro, se guarda el tiempo de ejecución y varia ChronoRun.

```
if(ChronoRun && click) {
    chronometer.stop();
    Chronostop = SystemClock.elapsedRealtime() - chronometer.getBase();
    ChronoRun = false;
}
```

Figura 63. Cronómetro stop

Mientras que en el caso de detectar un largo clic la función que se ejecuta es la de reiniciar.

```
btn_start.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        Toast.makeText(getApplicationContext(), text: "Reset", Toast.LENGTH_SHORT).show();
        Chronometer chronometer =(Chronometer)findViewById(R.id.TXT_crono);
        chronometer.stop();
        chronometer.setBase(SystemClock.elapsedRealtime());
        Chronostop = Long.valueOf(0);
        ChronoRun=true;
        return false;
    }
});
```

Figura 64. Cronómetro Reset

Se para el tiempo y se fija como base el tiempo actual, además se indica que el tiempo acumulado es de 0, y por último, se pone la variable ChronoRun en *true* para que la próxima vez inicialice el tiempo.

7.6.6 Ventana graphic

Al comenzar la ventana del gráfico se desarrolla el siguiente flujo:

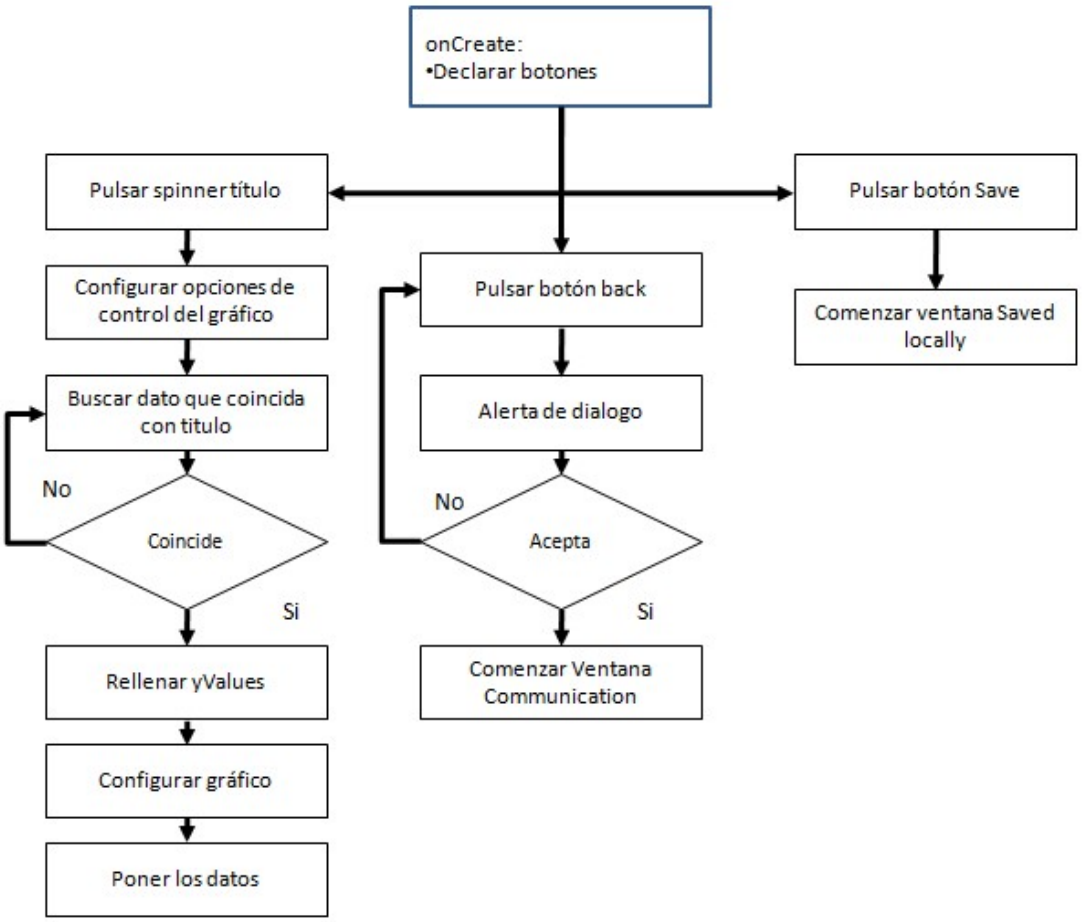

Figura 65. Flujo ventana gráfico

Hay que aclarar que en la Figura 65 en el caso que este ejecutando el ciclo visualización el botón *back* está invisible y se cambia el nombre del botón *save* a *next*. Esto se desarrolla al declarar los botones.

En la pantalla la función principal es la visualización del gráfico pero también es lo más destacable a la hora de analizar el código al ser algo que no se ha comentado aún.

Crear gráfico

A la hora de realizar los gráficos se ha utilizado una librería que se encuentra en la plataforma Github, implementando la librería en dependecies de build.grandle y sincronizándolo se ha logrado incorporar las funciones de dicha librería.

```
implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'
```

Figura 66. Librería gráfico

A la hora de desarrollar el gráfico en el método para crear el gráfico, primero, se define la variable con el cuadro donde aparece el este. Además, se quita y se pone la visibilidad permitiendo que en caso de cambio de datos se actualice automáticamente. Por último, se configuran las opciones de control del eje.

```
mChart= (LineChart) findViewById(R.id.Linearchart);
mChart.setVisibility(View.INVISIBLE);
mChart.setVisibility(View.VISIBLE);
mChart.setDragEnabled(true);    //poder moverse en eje X
mChart.setScaleEnabled(true);   //Ampliar la image
```

Figura 67. Ajustes del gráfico

Luego se define la variable yvalues para guardar los valores que se desean mostrar.

```
ArrayList<Entry> yValues= new ArrayList<>();
```

Figura 68. Declaración de la variable yValues

Creada la variable se añaden los valores que se desean mostrar, para ello comprueba si el título es el nombre de los datos. En caso que sea cierto mete en la variable primero el número de serie y luego todos los valores del dato.

```
if(Title.equals("Force")){
    for  (int i=0; i<ArrayForce.size(); i++) {
        yValues.add( new Entry(i, ArrayForce.get(i)));
          }
```

Figura 69. Inserción de valores en la variable yValues

Completada la variable se define la configuración de la línea de los datos, los ejes, el título y otros aspectos del gráfico.

```
//Line configuration
LineDataSet set1= new LineDataSet(yValues, label: "Data set 1" );
set1.setFillAlpha(110);
set1.setColor(Color.RED);
set1.setLineWidth(3f);
set1.setValueTextColor(android.R.color.transparent);
set1.setCircleColor(Color.RED);
```

Figura 70. Configuración de la línea

Terminada la definición de la configuración, se añaden la variable con los datos de la línea una lista de líneas. En caso de querer incorporar más de una línea al gráfico se añade aquí.

```
ArrayList<ILineDataSet> dataSets= new ArrayList<>();
dataSets.add(set1);
LineData data = new LineData(dataSets);
mChart.setData(data);
```

Figura 71. Inserción de la variable set1 al gráfico

7.6.7 Ventana Saved locally

Al comenzar la ventana de guardar se desarrolla el siguiente flujo:

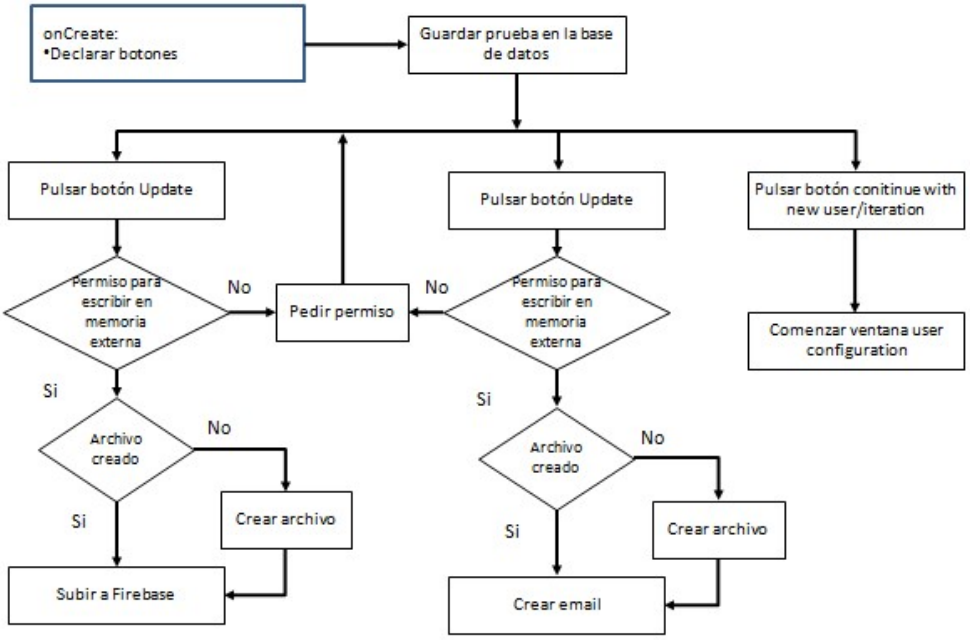

Figura 72. Flujo ventana Saved locally con ciclo prueba

En esta pantalla se encuentran diferencias en las funciones que se desarrollan entre el ciclo de prueba y el ciclo de visualización. Como se observa en la Figura 72 justo después de declarar los botones se hace el guardado de los datos de la prueba, mientras que en el ciclo de visualización no se requiere la función de guardado debido a que se está visualizando una prueba ya guardada en la base de datos.

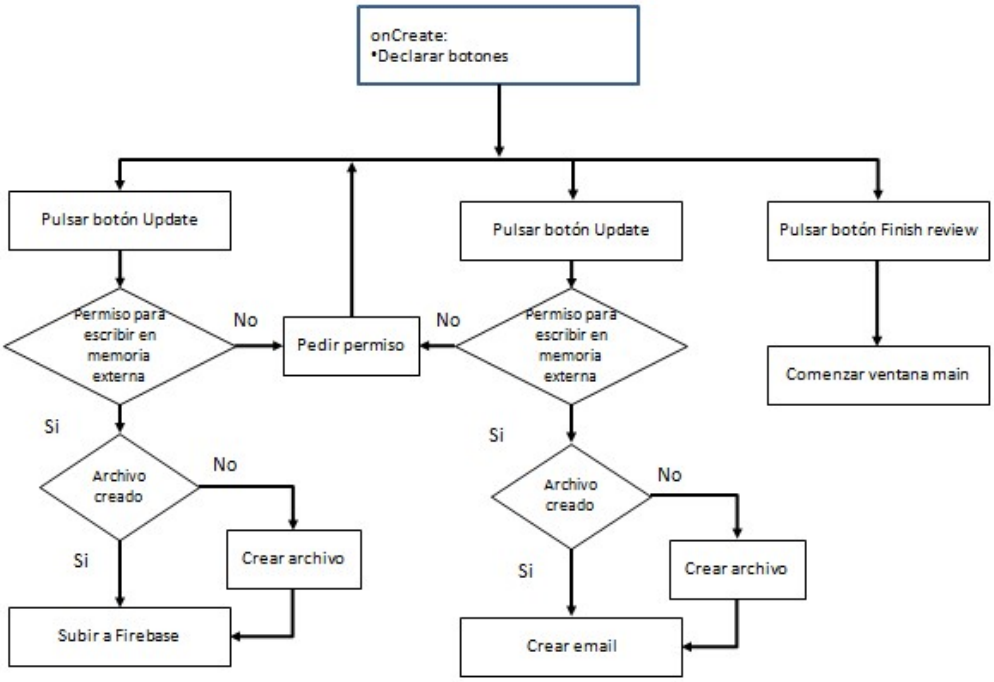

Figura 73. Flujo ventana Saved locally con ciclo visualización

En esta pantalla es de interés conocer cómo se realiza el crear el archivo, subir a la plataforma en la nube y crear el email.

Crear archivo

Primero se define el nombre del archivo con el número de usuario y luego el número de iteración.

```
Archivo = ID + "_" + Iteration + ".xls";
```

Figura 74. Nombre del archivo

Se crea la carpeta Folder en almacén externo en el caso que previamente no exista

```
File folder = new File( pathname: Environment.getExternalStorageDirectory() + "/Folder");
if (!folder.exists()) {
    try {
        folder.mkdir();

    } catch (Exception e) {
        e.printStackTrace();
    }
```

Figura 75. Crear carpeta par el archivo

Se crea el archivo si previamente no ha sido creado.

```
file = new File( pathname: Environment.getExternalStorageDirectory().getAbsolutePath() + "/Folder/" + Archivo);
if (!file.exists()) {
    try {
        file.createNewFile();

    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Figura 76. Crear el archivo

Una vez creado el archivo se escribe línea a línea el archivo. En la primera fila se escriben los títulos de cada columna. Luego se van metiendo fila a fila todos los valores de los datos, cuando termina de escribir todas las iteraciones se cierra el escritor.

Subir a Firebase

Android Studio tiene agregado en el asistente la plataforma Firebase, utilizando este sistema y seleccionando la función Storage dentro de Firebase el primer paso que hay que dar es conectar la aplicación con la plataforma. Para ello se dispone de un botón en el asistente que lo realiza.

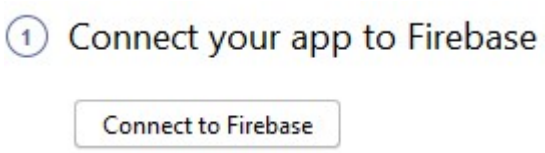

Figura 77. Conectar con Firebase

Luego hay que instalar las dependecies en la aplicación.

```
implementation 'com.google.firebase:firebase-storage:16.0.4'
implementation 'com.google.firebase:firebase-auth:16.0.5'
```

Figura 78. Instalar las librerías de Firebase

Se declara una referencia y se inicializa al crear la pantalla.

```
private StorageReference mStorageRef;

mStorageRef = FirebaseStorage.getInstance().getReference();
```

Figura 79. Declarar la referencia

Una vez creado el archivo CSV con los datos se guarda una referencia con la dirección en la que se quiere guardar y se sube el archivo

```
Uri urifile = Uri.fromFile(new File(file.toString()));
StorageReference riversRef = mStorageRef.child("CSV/"+ Archivo);

riversRef.putFile(urifile)
        .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

                Toast.makeText(getApplicationContext(), text: "File uploaded",Toast.LENGTH_SHORT).show();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {

                Toast.makeText(getApplicationContext(),exception.getMessage(),Toast.LENGTH_SHORT).show();
            }
        }) .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {

    }
});
```

Figura 80. Subir archivo CSV

Email

Una vez que se tiene el archivo CSV con los datos de la prueba se crea el email. Primero, se configura el email definiendo el título, la descripción del email y qué tipo de archivo se va adjuntar, para el formato CSV se ha de definir como pdf. Posteriormente, se adjunta el archivo indicando Identificador de recursos uniforme (URI). Por último, se manda abrir la actividad con la configuración del email.

```
String email="";
    //Send email
    Uri uri = Uri.fromFile(file);
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.setType("text/plain");
    emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[] { email});
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, value: "File " + Archivo);
    emailIntent.putExtra(Intent.EXTRA_TEXT, value: "Hello, I send you the file "+ Archivo +" created in "+ formattedDate);
    emailIntent.setType("application/pdf");
    emailIntent.putExtra(Intent.EXTRA_STREAM, uri);
    startActivity(Intent.createChooser(emailIntent, title: "sent"));
    Toast.makeText(getApplicationContext(), text: "enviado",Toast.LENGTH_SHORT).show();
```

Figura 81. Crear email

```
if (file.exists()) {
    try {
        FileWriter fileWriter = new FileWriter(file);
        BufferedWriter bfWriter = new BufferedWriter(fileWriter);
        bfWriter.write( str: "Fuerza;Altitud;Euler Roll;Euler Pitch;Euler Yaw;Accel X;Accel Y;Accel Z;Gyro X;Gyro Y;Gyro Z;Magen X;Magne Y;Magen Z");
        bfWriter.newLine();
        for (int ind = 0; ind < ArrayForce.size(); ind++) {
            bfWriter.write( str: String.valueOf(ArrayForce.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayAltitude.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayRoll.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayPitch.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayYaw.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayAccX.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayAccY.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayAccZ.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayRateX.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayRateY.get(ind)).replace( target: ".", replacement: ",")+ ";");
            bfWriter.write( str: String.valueOf(ArrayRateZ.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayMagneticX.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayMagneticY.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.write( str: String.valueOf(ArrayMagneticZ.get(ind)).replace( target: ".", replacement: ",") + ";");
            bfWriter.newLine();
        }
        Toast.makeText(getApplicationContext(), text: "Saved at " + file.toString(), Toast.LENGTH_SHORT).show();
        bfWriter.close();
        filewritten=true;
```

Figura 82. Completar el archivo

7.6.8 Ventana Configuration

Al comenzar la ventana de configuración se desarrolla el siguiente flujo:

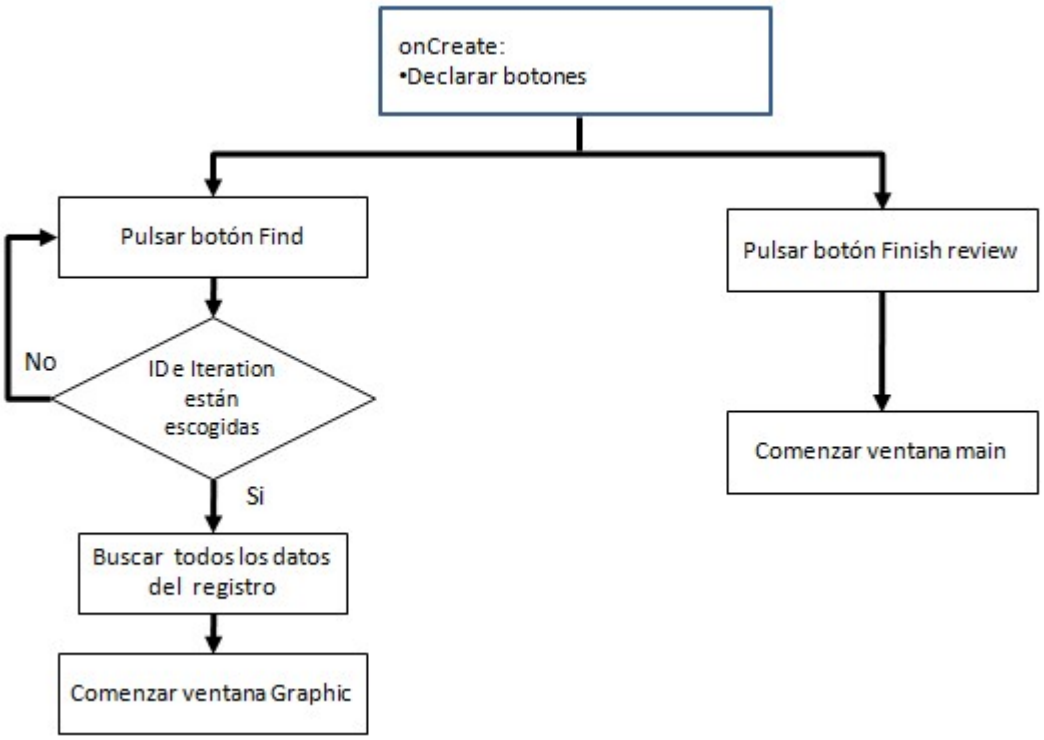

Figura 83. Flujo ventana configuration

8 METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO

8.1 Descripción de tareas, fases, equipos o procedimientos

El proyecto del desarrollo de la aplicación comenzó a principios de marzo y durante los siguientes meses se han seguidos las siguientes fases:

1. Fase: Formación

El proyecto comenzó con la formación tanto en la materia, como en el entorno software que se iba a utilizar.

Para realizar esta fase se utilizaron 52h repartidas de la siguiente forma:

Tarea	Duración
SMARTIP	4h
Java	17h
Android Studio	31h

Tabla 8. Tareas en la formación

2. Fase: Diseño

En esta fase se realizó el diseño las pantallas de la aplicación, definiendo la función de cada ventana y creando un pequeño boceto de cada una. Además, se definieron las conexiones entre pantallas creando los flujos de los procesos.

El diseño ha tenido la siguiente duración:

Tarea	Duración
Diseño y definición de las pantallas	8h

Tabla 9.Tareas en el diseño

3. Fase: Desarrollo

El desarrollo es la fase en la que se han invertido la mayor parte de las horas del proyecto. A la hora de realizar se ha dividido en varias tareas, primero, se han creado los layout de cada ventana y la interconexión creando los flujos; y posteriormente, se ha procedido a desarrollar las funciones de cada ventana.

Las diversas tareas han tenido las siguientes duraciones:

Tarea	Duración
Layout	62h
Main	16h
User configuration	32h
Communication	232h
Graphic	66h
Saved locally	53h
Configuration	12h

Tabla 10. Tareas en el desarrollo de la aplicación

4. Fase: Escribir proyecto

En la última fase se ha escrito la documentación.

Tarea	Duración
Escribir proyecto	78h

Tabla 11. Tareas en la fase 4

8.2 Diagrama de Gantt

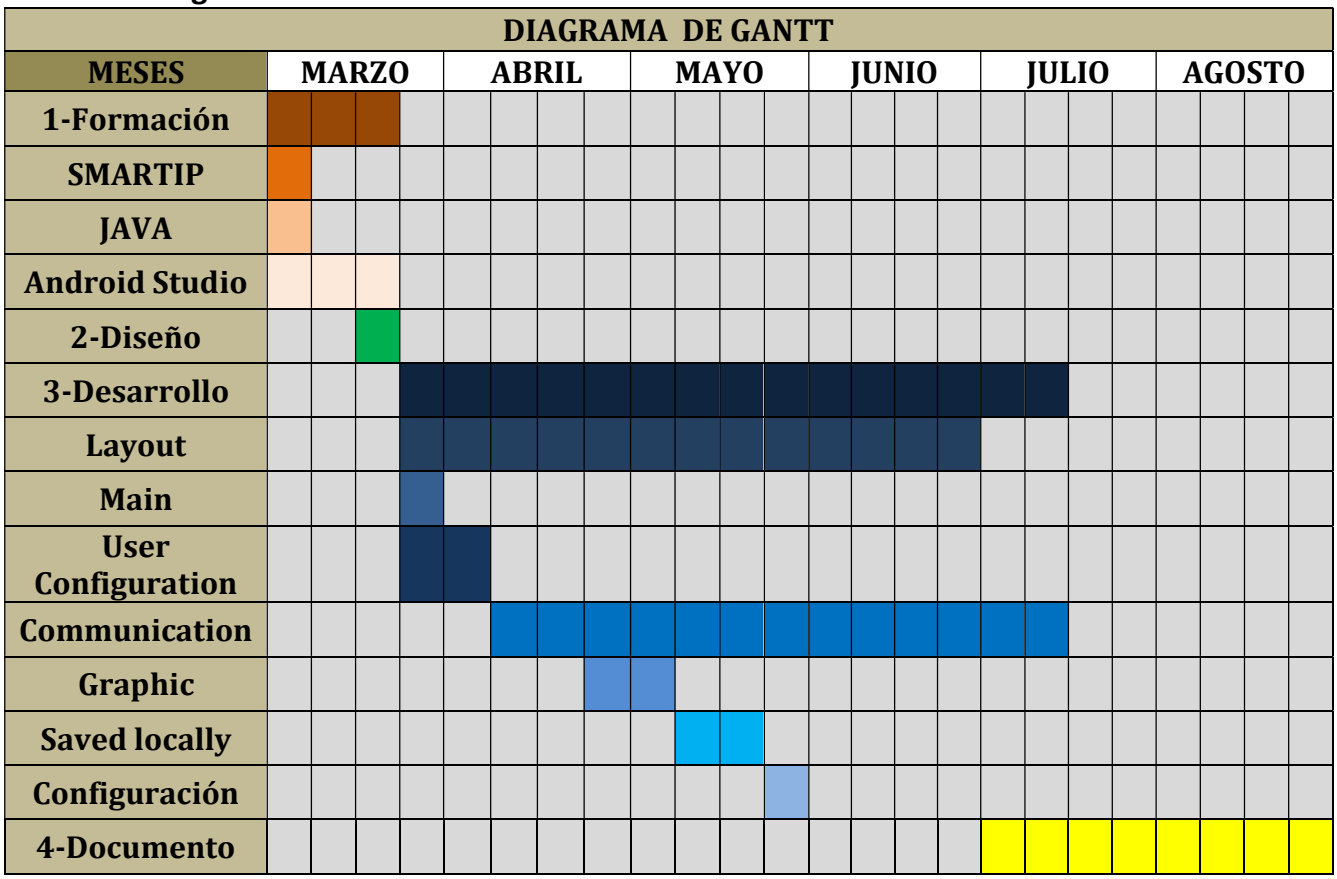

Tabla 12. Diagrama de Gantt

9 ASPECTOS ECONÓMICOS

9.1 Descripción del presupuesto

En el siguiente apartado se va analizar el presupuesto necesario para la ejecución del proyecto. Para la realización de este proyecto se han calculado las horas invertidas por el ingeniero junior cuya tasa está en 16 euros la hora.

Para el desarrollo de la aplicación y elaboración de la memoria se ha empleado un ordenador y un móvil valorados en 400 y 100 euros respectivamente. Además, se ha empleado una licencia gratuita de Android Studio y Firebase.

El porcentaje de los costes directos es del 7%, mientras que el coste de finanza es del 4%.

HORAS EMPLEADAS

	Tasa	Horas	Precio
Ingeniero Junior	35	611	21385

Tabla 13. Horas empleadas

GASTOS Y AMORTIZACIONES

	Cantidad	Coste	Vida útil	Horas utilización	Precio
Ordenador	1	600€	5000 h	225h	27
Móvil	1	120€	8800 h	200h	3
Material de oficina	1	50€			50

Tabla 14. Gastos y amortizaciones

Costes directos	21465€
Costes indirectos	1502,55€
Total costes	22967,55€
Costes de finanza	918,7€
Total	23886,25€

Tabla 15. Presupuesto

Una vez calculado el presupuesto del proyecto se sitúa en los 23886,25€. Correspondiendo un mayor gasto las horas empleadas en el proyecto.

10 CONCLUSIONES

Con la aplicación desarrollada durante este proyecto se ha querido dar respuesta a los requisitos establecidos por un Equipo de Investigación del Departamento de Automática y Robótica de la Escuela de Ingeniería de Bilbao de la UPV/EHU. Así, durante este proyecto se ha desarrollado una aplicación Android que permite conectar el móvil con una contera sensorizada, en concreto, la contera inteligente del proyecto SMARTIP, dentro del que se enmarca este TFM. Para ello se ha utilizado la tecnología Bluetooth Low Energy, a través de la cual, se transfieren un total de 14 valores diferentes desde la Contera sensorizada a la aplicación móvil. Valores que se registran a través de distintos sensores integrados en la contera inteligente y que favorecen un conocimiento más exhaustivo de la enfermedad que padece la persona y de la evolución de la misma, y por ende, una mejor adaptación del tratamiento a la persona.

Finalizado el proyecto se puede concluir que se ha cumplido con los requisitos demandados como es el uso del programa Android Studio para el desarrollo integro de la aplicación. Por otro lado, se ha logrado alcanzar cada objetivo propuesto. En primer lugar, el proyecto se ha ajustado al máximo posible a la transmisión de la contera sensorizada manteniendo totalmente el código de ésta. En segundo lugar, se han incorporado las funciones de almacenaje interno y subida, a través de la creación de una base de datos y la plataforma Firebase en la nube. En tercer lugar, se ha posibilitado la visualización de los datos en transición a tiempo real, pudiendo acceder al último dato, o incluso a un gráfico creado que recoge los 10 últimos valores adquiridos permitiendo verificar la tendencia del dato. Por último, se ha creado acceso a pruebas registradas con anterioridad, pudiendo visualizar la serie completa y su subida a la nube.

Además, aparte de las funciones indispensables, se han desarrollado otras funciones que aportan valor añadido al programa, como son el cronómetro, el envío de emails y los gráficos. El cronómetro permite registrar el tiempo mientras se realiza la prueba sin la necesidad de utilizar otro dispositivo como un reloj o cronómetro, ya que este se encuentra integrado en la aplicación. La opción de enviar por email permite enviar rápidamente los resultados a otra persona (personal sanitario, fisioterapeuta, familiar, etc.), y el gráfico que se genera después de la prueba permite asegurarse de unos buenos resultados y de una visualización más sencilla de estos.

10.1 Líneas futuras

A pesar de haber completado los objetivos planteados, cabe subrayar que se han encontrado algunas dificultades durante el desarrollo del proyecto. A la hora de realizar el envío de las características respetando la programación BLE establecida y de la que dispone la contera inteligente, se aprecia que Android Studio no admite el envío de varias características de forma simultánea, ya que ambas se solapan y no consigue recibirlas. Por lo que ha de hacerse en formato individual, es decir, primero el envío de una y al poco tiempo la segunda. Para observar si esto es posible se han realizado pruebas verificando el correcto funcionamiento del envío a nivel individual de las características, respetando siempre el tiempo de 20 milisegundos entre cada muestra.

Así la verificación del correcto funcionamiento da pie a una de las propuestas a futuro como es el cambio en la programación del BLE de la contera sensorizada, modificando el código para una transferencia individualizada.

Otra de las propuestas se relaciona con el hecho de tener que meter manualmente el MAC en la ventana *main*, ya que se pueden ocasionar problemas o dificultades debido a que todas las personas usuarias de la aplicación no conocen el MAC del dispositivo con el que se quieren conectar. Por ello, con el fin de facilitar ese paso se puede desarrollar un sistema de escaneo que sea capaz de buscar los dispositivos accesibles y mostrar el listado para que el usuario tenga que seleccionar con cual desea hacer la prueba.

Por último, debido a la situación sanitaria que se está viviendo y al distanciamiento social que se debe mantener a consecuencia de esta, no se ha podido implementar la aplicación en pruebas reales con usuarios. Por lo que como propuesta final se plantea la realización de pruebas de verificación del correcto funcionamiento en la captación de datos con personas que padecen EM.

11 BIBLIOGRAFÍA

Akhayad, Y. (2016). *Bluetooth 4.0 Low Energy: Análisis de las prestaciones y aplicaciones para la automoción.* Obtenido de https://upcommons.upc.edu/bitstream/handle/2117/82702/memoria.pdf

API, G. D. (s.f.). *Introduction to Google Drive API*. Recuperado el 16 de marzo de 2020, de Google Drive API: https://developers.google.com/drive/api/v3/about-sdk

AWS. (2019). *Informática en la Nube con AWS*. Recuperado el 16 de marzo de 2020, de AWS: https://aws.amazon.com/es/what-is-aws/

Bennett, S. E. (2010). Effective rehabilitation methods in patients with multiple sclerosis. *US Neurology*, 67-70.

Bluetooth. (9 de abril de 2018). *Intro to Bluetooth Low Energy*. Recuperado el 5 de febrero de 2020, de Bluetooth: https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-low-energy/

Brull Mesanza, A. (2020). *Sistema Inteligente para el Diágnostico Funcional en la marcha de pacientes de Esclerosis Múltiple.*

Chugh, A. (2018). *Android Realm Database*. Recuperado el 16 de marzo de 2020, de Slack: https://www.journaldev.com/23357/android-realm-database

ELT. (s.f.). *BLE (Bluetooth Low Energy)*. Recuperado el 2020, de ELT: Innovation in Lighting Technology: https://www.elt.es/ble-bluetooth-low-energy#:~:text=El%20Bluetooth%20de%20baja%20energ%C3%ADa,dependientes%20de%20bater%C3%ADa%20o%20pila

EME. (s.f.). *Esclerosis Múltiple*. Recuperado el 3 de abril de 2020, de Esclerosis Múltiple España: https://esclerosismultiple.com/esclerosis-multiple/sintomas/

EME, Roche Farma, AEDEM-COCEMFE. (2019). *Cuestiones más frecuentes sobre la Esclerosis Múltiple.* Roche Farma.

EME, ROCHE, AEDEM-COCEMFE. (s.f.). *Cuestiones más frecuentes sobre la Esclerosis Múltiple.* Roche Farma.

FEGADEM. (s.f.). *EM Remitente - Recurrente*. Recuperado el 23 de marzo de 2020, de Fedaración Galega de Esclerose Múltiple: https://esclerosismultiplegalicia.org/esclerosis_multiple_recurrente_remitente_e

s.html#:~:text=Es%20el%20tipo%20m%C3%A1s%20frecuente,luego%20mejoran%20parcial%20o%20totalmente

Firebase. (s.f.). *Cloud Storage*. Recuperado el 16 de marzo de 2020, de Firebase: https://firebase.google.com/docs/storage

Flachenecker, P. (2015). Clinical Implications of Neuroplasticity – The Role of Rehabilitation in Multiple Sclerosis. *Frontiers in Neurology*.

García Urra, F., & Porres Aracama, J. (2015). Monitorización remota: estado actual. *Impulso Revista: Actualidad de Estimulación Cardiaca*(5).

García Urra, F., Porres Aracama, J., & Fontán Martín-Chico, B. (abril de 2013). Dispositivos eléctricos y monitorización remota. *Revista Uruguaya de Cardiología, 28*(1).

García-Domínguez J.M., Maurino J., Martínez-Ginés M.L., Carmona O., Caminero A.B., Medrano N., Ruíz-Beato E.W, IMPACT Clinical Investigators. (2019). Economic burden of multiple sclerosis in a population with low physical disability. *BMC Public Health*.

Gervais, C. (s.f.). *Crutch development: Wiring & Software Programs.*

Google Cloud. (s.f.). *Solucionar problemas forma parte de nuestro ADN. Nuestro objetivo es ponerlo a tu servicio.* Recuperado el 16 de marzo de 2020, de Google Cloud: https://cloud.google.com/why-google-cloud

Greenrobot. (s.f.). *greenDAO Features*. Recuperado el 16 de marzo de 2020, de Greenrobot: https://greenrobot.org/greendao/features/

Hospital Virgen del Mar. (s.f.). *Avances tecnológicos en la salud: mejoras aplicadas a la medicina*. Recuperado el 16 de enero de 2020, de Hospital Virgen del Mar: https://www.hospitalvirgendelmar.es/noticia/la-tecnologia-aplicada-a-la-salud-los-ultimos-y-mejores-avances/20

Hu, F., Xie, D., & Shen, S. (2013). *On the Application of the Internet of Things in the Field of Medical and Health Care.*Obtenido de https://ieeexplore-ieee-org.ehu.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=6682394

IAT. (s.f.). *IoT en Medicina. Que es y Ejemplos de Aplicaciones*. Recuperado el 23 de marzo de 2020, de IAT: https://iat.es/tecnologias/internet-de-las-cosas-iot/medicina/

Izquierdo Ayuso, G. (2014). Esclerosis múltiple: impacto socioeconómico y en la calidad de vida de los pacientes. *Medicina Clínica, 143*, 7-12.

Jacopo, T., Taffoni, F., Santacatterina, M., Sannino, R., & Formica, D. (13 de diciembre de 2017). *Performance Evaluation of Bluetooth Low Energy: A Systematic Review.*Recuperado el 2020, de MDPI: https://www.mdpi.com/1424-8220/17/12/2898/html

Karampampa K, Gustavsson A, Miltenburger C, Eckert B. (2012). Treatment experience, burden and unmet needs (TRIBUNE) in MS study: results from Spain. *Mult Scler*.

Muradas Maceira, Y. (23 de marzo de 2018). *SQLite para Android: La herramienta definitiva*. Recuperado el marzo de 2020, de OpenWebinars: https://openwebinars.net/blog/sqlite-para-android-la-herramienta-definitiva/

Oreja-Guevara C, Kobelt G, Berg J, Capsa D, Eriksson J, Platform EMS. (2017). New insights into the burden and costs of multiple sclerosis in Europe: Results for Spain. *Mult Scler*, 166-178.

Pérez Menendez, A. (18 de diciembre de 2016). *El 70% de los nuevos casos diagnosticados de Esclerosis Múltiple corresponden a personas de entre 20 y 40 años .* Recuperado el enero de 2020, de Sociedad Española de Neurología: https://www.sen.es/saladeprensa/pdf/Link204.pdf

Prometec. (s.f.). *SENSOR DE PRESIÓN Y TEMPERATURA BMP280*. Recuperado el 23 de febrero de 2020, de Prometec: https://www.prometec.net/sensor-de-presion-y-temperatura-bmp280/

Realm Database. (s.f.). Recuperado el 16 de marzo de 2020, de Realm: https://realm.io/docs/

Souza, A. K. (2010). Multiple sclerosis and mobility-related assistive technology: Systematic review of literature. *TheJournal of RehabilitationResearch and Development*.

Weber. (4 de junio de 2018). *Proyecto SROI-EM.Impacto clínico, asistencial, económico y social del abordajeideal de la esclerosis múltiple en comparación con el abordaje actual. Informe Final.* Obtenido de Roche: https://www.roche.es/content/dam/rochexx/roche-es/roche-farma/neurociencias/Informe-SROIEM.pdf

World Health Organization. (2006). *Neurological disorders: public health challenges.*World Health Organization.

12 ANEXO: MANUAL DE USUARIO

12.1 Introducción

Crutch v0 es una aplicación cuyo objetivo es monitorizar el avance de la Esclerosis Múltiple realizando pruebas médicas a usuarios con dicha enfermedad mediante una contera sensorizada.

Este manual se diferencia en distintos apartados. Primero, se van a definir los requisitos necesarios para un buen funcionamiento de la aplicación; después, se explica el funcionamiento de la aplicación narrando paso a paso como realizar los dos ciclos de ejecución posibles y enseñando a utilizar las funciones integradas en las ventanas.

12.2 Requisitos

Para el uso adecuado de la aplicación el móvil ha de disponer de los siguientes requisitos:

- Sistema operativo Android (4.3 o mayor)
- Bluetooth
- Conectividad (3G y/o WIFI)
- Contera inteligente
- Aplicación gmail

12.3 Realizar una prueba

En esta sección se explica paso a paso cómo realizar una prueba registrando los datos de la muleta.

Al inicializarse la aplicación se queda en la pantalla que se observa en la Figura 84.

Paso 1: Insertar el MAC del dispositivo con el que se quiere conectar. Para realizar este paso se dispone de dos alternativas:

1.1Escribir el nombre completo MAC. La aplicación crea una lista con los dispositivos que vayan coincidiendo con el nombre inscrito.

1.2 Pulsando en la flecha 1.2 se despliega todos los MAC previamente usados y selecciona el que se desee.

Paso 2: Pulsar el botón conectar.

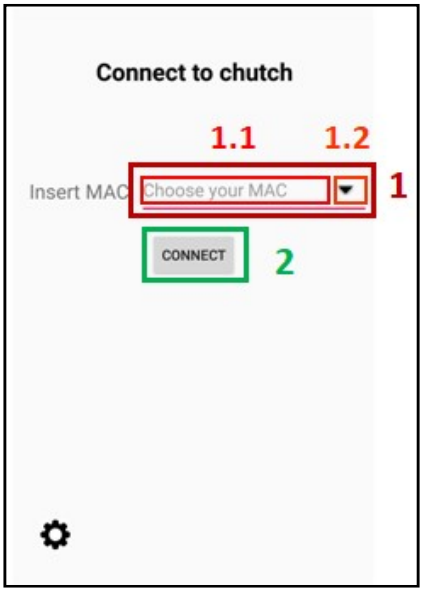

Figura 84. Paso 1 y 2 de realizar prueba

Paso 3: Escribir el número identidad del usuario y el número de la prueba.

Paso 4: Pulsar el botón *next.*

Figura 85. Paso 3 y 4 de realizar prueba

Si la configuración del usuario ha sido previamente registrada salta una advertencia.

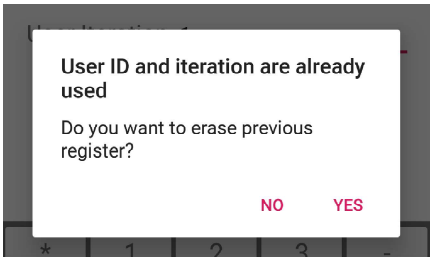

Figura 86. Advertencia

Pulsando *NO* se posibilita el cambio de los números.

Pulsando *YES* borra el registro previo y avanza al siguiente paso.

Paso 5: Adquisición de datos.

En la pantalla se disponen de diferentes botones para el control de la adquisición y otras funciones.

- Play: Comienza o vuelve inicializar la adquisición.
- Pause: Paraliza la recogida de datos.
- Back: Devuelve a la ventana de configuración de usuario. (paso 3)
- Stop: Finaliza la adquisición de datos. (Paso 6)
- Cronometro: control de la función del cronometro. (Ver12.5.1)
- Gráfico: Acceso a visualizar los 10 últimos valores. (Ver12.5.2)
- Zoom: Se muestran el último valor de los 3 ejes. (Ver12.5.3)
- Visualización datos: Se muestra los últimos valores obtenidos.
- Datos usuario: Muestra los datos del usuario.

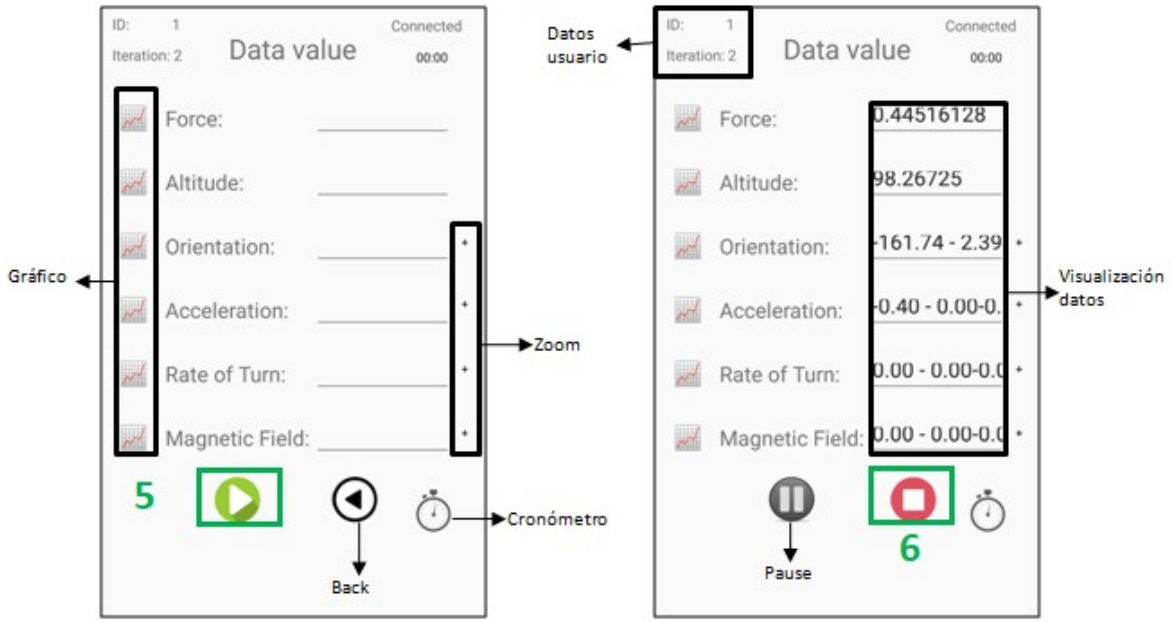

Figura 87. Paso 5 y 6 de realizar prueba

Paso 7: Visualización la serie en el gráfico.

Pulsando en el nombre se abre la lista y se selecciona el grupo que se quiera visualizar.

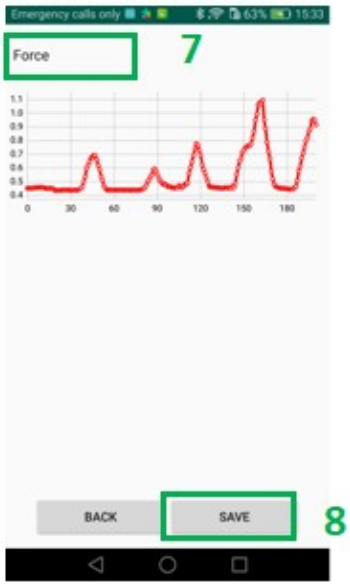

Figura 88. Paso 7 y 8 de realizar prueba

Paso 8: Pulsar *Save* para guardar los registros de la prueba.

Pulsando el botón back se vuelve al paso 5, borrando los datos adquiridos previamente.

Paso 9: Pulsando los botones marcados posibilita subir el archivo de la prueba a la nube (Ver 12.5.4) o enviarlo por email. (Ver 12.5.4)

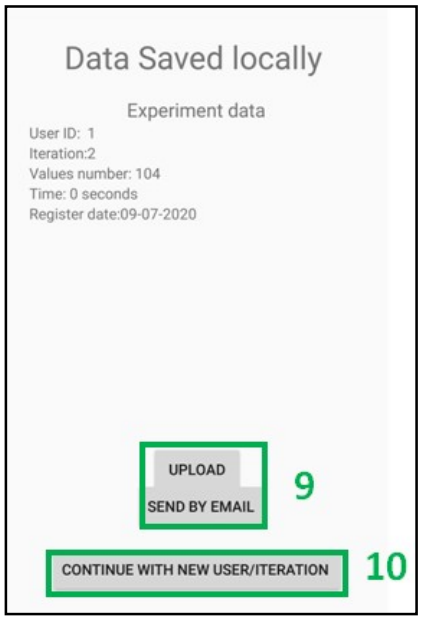

Figura 89. Paso 9 y 10 de realizar prueba

Paso 10: Pulsando el botón *Continue with new user/iteration* se vuelve a la ventana de configuración de usuario pudiendo volver a hacer otra prueba (Paso 3).

12.4 Visualizar prueba previa

En este apartado se explica paso a paso cómo volver a visualizar una prueba realizada previamente. Para realizar el proceso se deben de seguir los siguientes pasos:

Paso 1: Acceder a la ventana configuración pulsando el botón con forma de engranaje.

Figura 90. Paso1 de visualizar prueba

Paso 2: Seleccionar tanto el user ID como el User iteration.

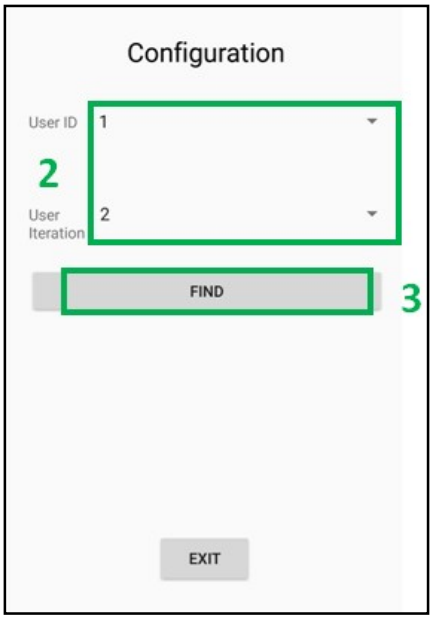

Figura 91. Paso 2 y 3 de visualizar prueba

Paso 3: Pulsar el botón *find*.

Paso 4: Visualizar los gráficos de los diferentes datos. Seleccionando el nombre se despliega una lista con el nombre de los datos.

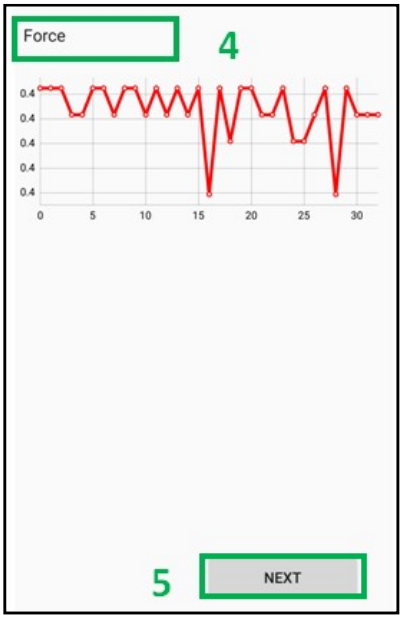

Figura 92. Paso 4 y 5 de visualizar prueba

Paso 5: Pulsar *Next* para avanzar a la siguiente pantalla.

Paso 6: Subir la prueba a Firebase o enviarlo por email.

Figura 93. Paso 6 y 7 de visualizar prueba

Paso 7: Pulsar el botón *Finish review* para finalizar el proceso.

12.5 Otras funciones

Como se ha observado dentro de los ciclos en alguna ventana se haya las funciones secundarias que no han sido explicadas en el proceso principal.

12.5.1 Cronómetro

La función de cronómetro se encuentra integrada en la ventana Communication, su objetivo es medir el tiempo y funciona de modo independiente a la adquisición de datos.

Figura 94. Pantalla con la función cronometro

Su funcionamiento se controla con un único botón con forma de cronometro. Este varia de color dependiendo del estando que se encuentre.

Figura 95. Estado del botón cronómetro

Para iniciar hay que pulsar una vez el botón cronometro, se pone en verde y en la visualización del tiempo comienza a incrementar el tiempo. Para parar se vuelve a pulsar una vez el botón. En caso de querer poner a cero el tiempo se ha de mantener pulsado durante 2 segundos el botón.

12.5.2 Gráfico tiempo real

La función de gráfico en tiempo real se encuentra integrada en la ventana Communication, su objetivo es visualizar los últimos 10 valores obtenidos y así intuir la tendencia que está siguiendo la serie.

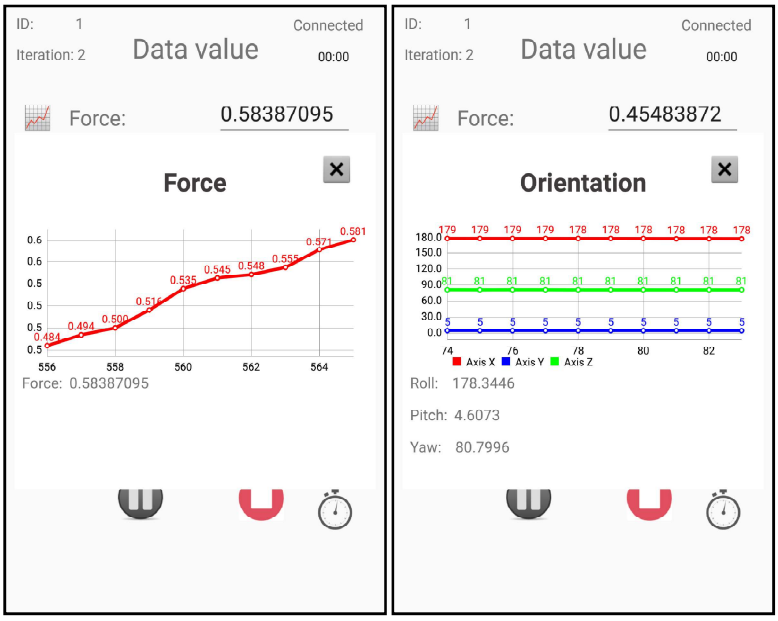

Figura 96. Mini ventana de la función Gráfico tiempo real

Para abrir la mini ventana del gráfico en tiempo real hay que pulsar el botón con la figura de un gráfico.

Mientras que para cerrar la mini ventana hay que pulsar la cruz que se encuentra a la izquierda del título de la mini ventana.

12.5.3 Zoom

La función zoom se encuentra integrada en la ventana Communication, su objetivo es mostrar de una forma más visual los 3 ejes de un dato.

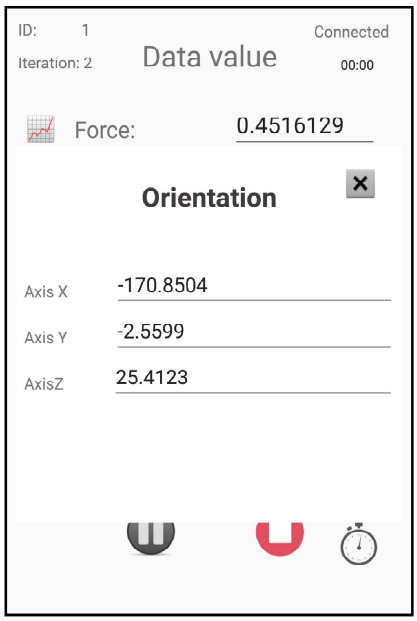

Figura 97. Mini ventana de la función Zoom

Para abrir la mini ventana zoom hay que pulsar el botón con la figura de una suma.

Mientras que para cerrar la mini ventana hay que pulsar la cruz que se encuentra a la izquierda del título de la mini ventana.

12.5.4 Subir la nube

La función subir a la nube se encuentra integrada en la ventana Saved locally, su objetivo es subir la prueba a la plataforma en la nube Firebase.

El único paso que hay que dar es pulsar el botón *Upload* y el archivo se subirá inmediatamente.

En el caso de que la aplicación aun no tenga el permiso de acceso a la tarjeta SD se muestra el aviso de la Figura 98. Aviso para permitir el acceso a la tarjeta SD

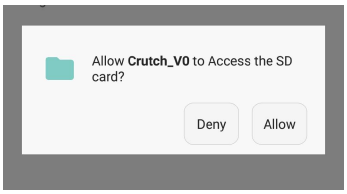

Figura 98. Aviso para permitir el acceso a la tarjeta SD

Pulsa en el botón *allow* para permitir el acceso, una vez que se tenga, vuelve a intentar subirlo volviendo a pulsar el botón.

12.5.5 Enviar por email

La función enviar por email se encuentra integrada en la ventana Saved locally, su objetivo es crear un email con el archivo CSV de la prueba adjunta y a solo a falta de introducir el email del destinatario.

Para crear el email hay que pulsar en el botón *send by email*. En el caso que la aplicación aun no tenga el permiso de acceso a la tarjeta SD se muestra el aviso de la Figura 98. Aviso para permitir el acceso a la tarjeta SD

Pulsa en el botón *allow* para permitir el acceso, una vez que se tenga, vuelve a intentar crear el email.

Una vez pulsado el botón se abre una ventana en la que seleccionar el icono de *gmail.*

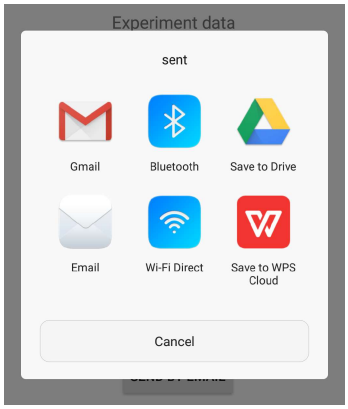

Figura 99. Pantalla de envío

Al seleccionarlo se abre la ventana de un nuevo gmail y se rellenan los datos de título y descripción, además de adjuntarse el archivo de la prueba.

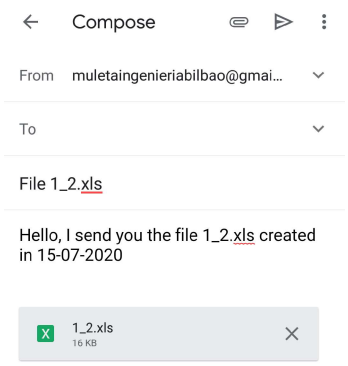

Figura 100. Email creado

Por último hay que introducir el email de destinatario y pulsar *enviar*. El correo se envía a su destino enviando el archivo.