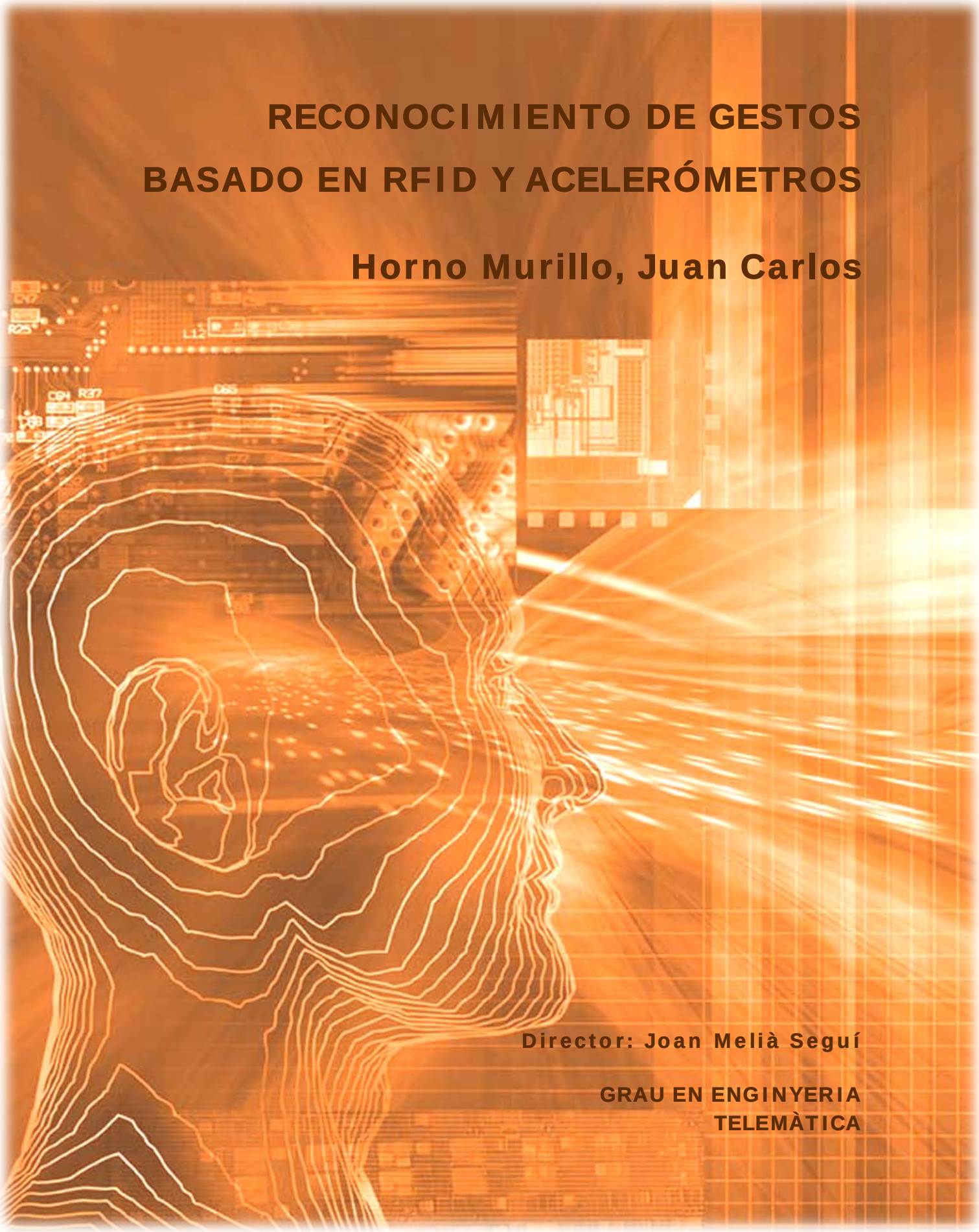
RECONOCIMIENTO DE GESTOS BASADO EN RFID Y ACELERÓMETROS

Horno Murillo, Juan Carlos



Director: Joan Melià Seguí

GRAU EN ENGINYERIA TELEMÀTICA

Treball de Fi de Grau

upf. Universitat Pompeu Fabra Barcelona
Escola Superior Politècnica

RECONOCIMIENTO DE GESTOS BASADO EN RFID Y ACELERÓMETROS

Juan Carlos Horno Murillo

TRABAJO FINAL DE GRADO

GRADO EN INGENIERÍA TELEMÁTICA

ESCUELA SUPERIOR POLITÉCNICA UPF

AÑO 2014

DIRECTOR DEL TFG

Joan Melià Seguí

A mi familia que me ha dado la oportunidad de formarme y estudiar la carrera que siempre quise hacer y que me ha apoyado en todo momento con paciencia, animándome y confiando en mí, siempre.

Agradecimientos

En el campo académico quiero agradecer sinceramente a mi tutor, Joan Melià Seguí, por haber hecho posible la realización de este proyecto y por haberme guiado en todo momento para acabarlo con éxito.

Destacar su gran labor para resolver todas mis dudas y el apoyo y soporte que he tenido para desarrollar, de la mejor manera posible, la memoria.

¡Muchas gracias de nuevo por ayudarme a llevar a cabo este proyecto!

Resumen

La tecnología RFID (*Radio Frecuency Identification*) presenta vulnerabilidades de seguridad en la lectura de los *tags RFID*. Supone un reto proporcionar mecanismos de defensa sin que implique un cambio en el paradigma de uso de la tecnología RFID.

Mediante *tags* RFID pasivos y un dispositivo móvil equipados con sensores acelerómetros, se obtiene información en cuanto a un gesto realizado por un usuario. Esto ofrece la posibilidad de realizar gestos e identificar qué gesto se está produciendo y a quien pertenece.

La primera parte del proyecto se centra en procesar y caracterizar el conjunto de gestos realizados por el móvil y el *tag* RFID. En la segunda parte, las características extraídas de los gestos son estudiadas por el algoritmo *k-Nearest Neighbour,* basado en *Machine Learning*, para poder clasificar e identificar el gesto realizado.

Los resultados obtenidos permiten implementar un sistema de autenticación basado en *tags* RFID pasivos y un dispositivo móvil. Este estudio permitirá crear aplicaciones para garantizar la lectura segura de los *tags* RFID y aplicaciones de identificación de personas.

Abstract

RFID technology has important security problems in the tag reading process. This weakness restricts the commercial application of this technology. The aim is to provide security systems without modifying the RFID technology usage paradigm. It used RFID tags and a Smartphone, both provided with accelerometer sensors, to collect data sets based on user's gestures to identify what gesture they are doing and who they belong to.

The first part of the project was focused on processing and featuring accelerometer data sets to model the gestures. In the second part, we used k-Nearest Neighbour as Machine Learning method for classifying and deciding what gesture was performed.

Looking forward, we will be able to recognize a gesture performed by a tag. This gesture will be linked to a mobile device, where a second authentication will be performed based on the uniqueness of the gesture. It will work in future applications to identify people and to guarantee secure tag reading operations.

Índice

Pág.

Lista de Figuras

Pág.

Lista de Tablas

Pág.

CAPÍTULO 1

INTRODUCCIÓN

1.1. Descripción del estudio

Este estudio pretende reconocer patrones en los gestos descritos por un dispositivo acelerómetro utilizando nuevos prototipos de tecnología RFID (*Radio Frecuency Identification*) equipados con sensores acelerómetros pasivos. Se trata de estudiar y modelar la interacción humana, bajo el contexto del campo de estudio de la Computación Ubicua (*Ubiquitous Computing*).

En una aproximación más concreta, este proyecto recopila, analiza, caracteriza y decide si se trata de un gesto u otro de un conjunto de gestos reconocidos, en base a los datos de posición que va generando el acelerómetro.

En capítulos posteriores se verá de qué trata la tecnología RFID, cómo funciona un acelerómetro, la metodología seguida en el laboratorio para la generación de los gestos, las técnicas matemáticas utilizadas para caracterizar y analizar los gestos y la herramienta de decisión basada en métodos de aprendizaje automático (*Machine Learning)*.

Al tratarse de un proyecto final de carrera, la entrega se hace de forma individual pero se contará con el apoyo del grupo de investigación *Ubiquitous Computing Applications Laboratory (UbiCA Lab)* (1), con el propósito de formalizar el trabajo previamente citado y con la posibilidad de realizar un artículo científico.

1.2. Coyuntura Actual

En la última década, la tecnología RFID ha tenido un auge en cuanto a su implantación debido al desarrollo técnico, a la disposición de estándares internacionales y a la aceptación de las Administraciones Públicas, que son las que tienen competencias en la asignación de las frecuencias, impulsando así su uso en el mercado global (2). La Unión Europea impulsó en 2006 el proyecto BRIDGE (*Building Radio Frequency IDentification for the Global Environment*) con la ayuda de un consorcio de 30 *partners*, universidades, proveedores de servicios y empresas privadas, con el objetivo de investigar, desarrollar e implementar herramientas para el desarrollo de aplicaciones RFID y para ofrecer soluciones a los procesos de negocio dentro de la Comunidad Europea (3).

Actualmente el concepto de *"Internet of things"* creado por la ITU, pretende conectar los objetos a lo largo del mundo de una forma sensorial e inteligente con tecnologías de identificación, redes de sensores y nanotecnología. Eso supone una ventana entre el mundo de los objetos y de los servicios web. Un nuevo concepto, una nueva dimensión en Internet.

Hoy en día existen 1500 millones de PC's con acceso a Internet y cerca de 1000 millones de dispositivos móviles y se prevé que en 2020 se conecten a Internet de 50 a 100 mil millones de dispositivos. Si se consideran todas las posibles conexiones con todo tipo de objetos se podría estar hablando de 1 billón de conexiones a Internet (4).

"Internet of things" tiene a RFID como la tecnología clave y a pesar de que esta tecnología lleva más de medio siglo conviviendo en nuestra sociedad, ha tenido que esperar unos cuantos años hasta situarse en el contexto actual, en el que su uso ha sido extendido debido al bajo coste y tamaño de los circuitos integrados, que han permitido que el *tag* RFID *(*elemento utilizado para la identificación del objeto*)* se pueda producir de forma masiva sin suponer un gasto considerable.

En el siglo XIX las máquinas aprendían a producir, en el siglo XX aprendían a pensar y en el siglo XXI aprenden a percibir.

1.3. Motivación

La tecnología RFID tiene su principal uso en el ámbito de la localización, control y seguimiento de bienes materiales y en la identificación de personas. Por ello, muchas empresas que poseen cadenas de suministro, pueden optar por esta tecnología para mejorar sus servicios aportando automatización a sus procesos y aumentar así la precisión en su control y trazabilidad del stock, como por ejemplo el caso de la empresa japonesa de cuero llamada Penta (5).

Este proyecto pretende extender la funcionalidad de la tecnología RFID, para que no quede limitada a la identificación, localización y trazabilidad de los bienes y utilice información del medio físico mediante sensores incorporados a los *tags* RFID. En el ámbito alimenticio, un problema grave se produce en la logística del pollo, debido a que si éste se encuentra durante un periodo considerable expuesto a temperaturas cálidas, puede producir salmonelosis. Equipando el *tag* RFID que identifica el stock de pollo con un sensor de temperatura, podríamos saber las unidades potencialmente expuestas a salmonella y evitar así un problema muy serio de salud (6).

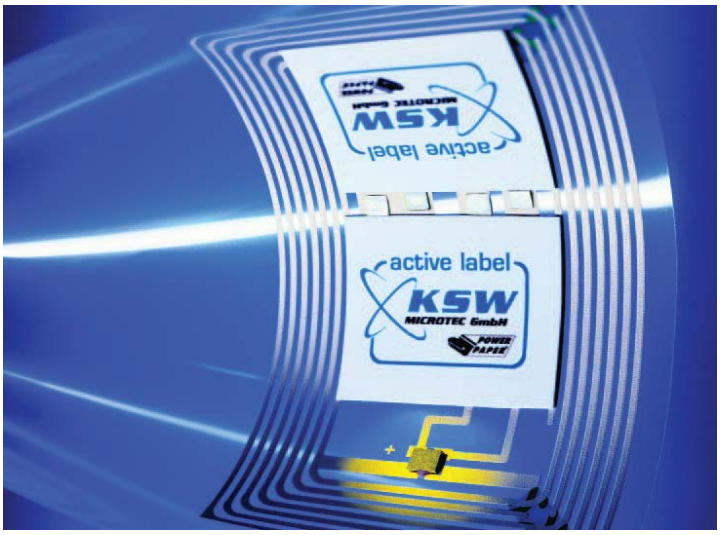


Figura 1.1. *Tag* RFID con *threshold* basado en temperatura (6).

Los sensores acelerómetros también pueden servir para controlar el estrés sufrido por la mercancía durante su transporte, empleando unas etiquetas no electrónicas basadas en colorantes que pueden cambiar de color al recibir golpes o vibraciones excesivas (6).

CAPÍTULO 2

TECNOLOGÍA RFID

2.1 Historia del RFID

La tecnología RFID (Identificación por radiofrecuencia) está creciendo en los últimos años y se está estableciendo como la tecnología más importante en cuanto a la identificación y la agilización de procesos industriales de bienes y materiales, sustituyendo a otras tecnologías como los códigos de barras que requieren una distancia muy corta y una visión directa entre lector y el código de barras impreso en el objeto. Incluso con la introducción de sensores se puede aportar información adicional de las condiciones externas del objeto como la temperatura, tal y como se ha mencionado con anterioridad.

La gran mayoría de las fuentes sitúan el origen de la tecnología RFID en la segunda Guerra Mundial. Hay quienes afirman que el primer uso de la tecnología RFID se llevó a cabo por el ejército inglés para detectar aviones enemigos, como una mejora del uso del radar.

Gracias al *Manhattan Project*, que sirvió entre otras cosas para desarrollar la bomba atómica, se creó el término *Identify Friend or Foe (IFF)* al utilizar el radar como método de identificación de enemigos (7).

Otras teorías apuntan que los alemanes descubrieron que si el piloto realizaba balanceos con el avión al regresar a la base, podían cambiar la señal de radio reflejada y saber que el avión estaba de su bando (8).

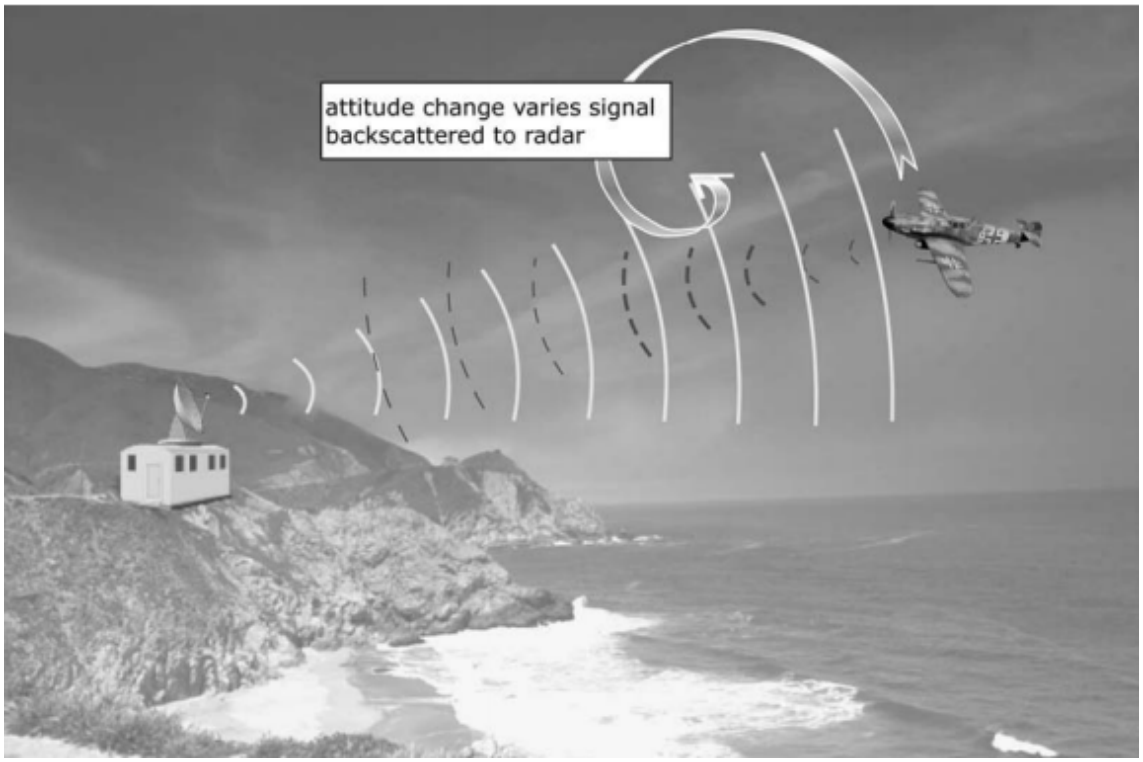


Figura 2.1. Funcionamiento del radar empleado en la Segunda Guerra Mundial para identificar aviones de flotas enemigas (9).

Existen diversas teorías sobre los orígenes del RFID. El primer trabajo de estudio de RFID fue un artículo publicado en 1948 por Harry Stockman titulado *"Communication by Means of Reflected Power"*, que se conoce como la génesis de la idea.

En 1960 se iniciaron las primeras actividades comerciales y varias empresas crearon EAS (*Electronic Article Surveillance*) que se conoce como el primer uso comercial de la tecnología RFID, y que sirvió como sistema de alarma ante el robo de productos en tiendas.

En 1970, instituciones educativas, empresas y gobiernos trabajaron de forma activa en el desarrollo de la tecnología RFID. En 1975 se crean los *tags* pasivos con un rango aproximado de 10 metros.

En 1980, se produce la comercialización de la tecnología RFID gracias al desarrollo del PC, que ofrecía la posibilidad de almacenar información útil proveniente del sistema RFID.

En los años 90, el uso de RFID se extiende a gran escala y los investigadores empiezan a enfocar el uso de la tecnología al seguimiento e identificación en cadenas de producción y logística. Se pretende reducir costes en la producción de *tags*, optimizar las redes de datos y desarrollar estándares abiertos. El *Auto-ID Center*, fundado en 1999 por un conjunto de empresas e instituciones educativas como el MIT o la Universidad de Cambridge, se crea para tal propósito (10).

En 2003, el *Auto-ID Center* traspasa toda su tecnología desarrollada a *EPC Global*, una organización destinada a la creación de un estándar internacional para gestionar el uso y la mejora de la tecnología RFID en las cadenas de abastecimiento, permitiendo la interoperabilidad entre las empresas (11).

2.2 Definición de la tecnología RFID

La tecnología RFID, *"Radio Frecuency Identification"*, es una tecnología que permite identificar objetos, animales, personas mediante el uso de ondas electromagnéticas. La tecnología RFID se podría concebir como un conjunto de tecnologías, en las que existen múltiples variaciones en cuanto al rango de frecuencias, o la incorporación de sensores que extraen características singulares del entorno perteneciente al objeto a identificar.

En un sistema RFID, el objeto a identificar se etiqueta con un circuito integrado equipado con una antena (conocido como '*tag*') para que un dispositivo transmisor/receptor (conocido como '*reader*'), también equipado con una antena, pueda comunicarse y extraer un código que permita a un *host* identificar el objeto. Éste es el sistema básico del RFID pero puede ser ampliado por elementos de red para operar con la información leída del *tag*.

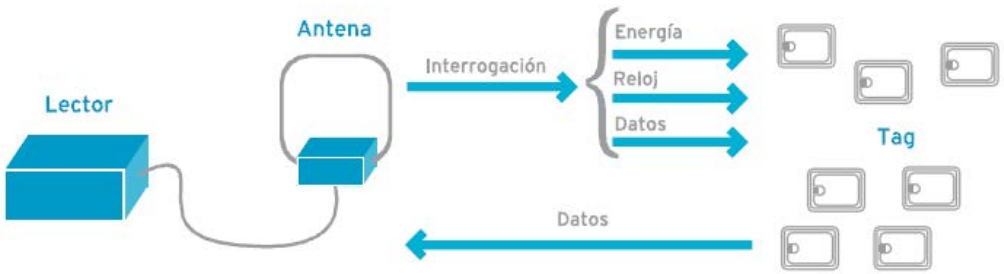


Figura 2.2. Sistema RFID básico (2)

2.3 Transpondedor (*tag* RFID)

El *tag* RFID, se denomina transpondedor debido a su capacidad para recibir y transmitir señales, pero sólo puede transmitir cuando previamente ha sido estimulado por un dispositivo transmisor. Un *tag* RFID está formado por el chip o circuito integrado (*IC-integrated Circuit*), la antena y un sustrato (*inlay*).

Circuito integrado (IC): Es el dispositivo electrónico que puede ser concebido como el computador del *tag*, encargado de ejecutar las instrucciones correspondientes para extraer de la memoria y transmitir el código que permite la identificación. Este código de objeto formado por 96 bits se denomina EPC "*Electronic Product Code*".

Antena: La función de la antena es absorber la energía de las ondas de RF y transmitirlas al circuito integrado. El tamaño de la antena influye en la forma de operar de nuestro *tag*, ya que amplía el alcance de lectura del *tag*.

Sustrato: Es el material donde se imprime el circuito junto con la antena. Este material puede variar en función de las características del objeto que va a identificar. Existen *tags* especiales para textil, líquidos, metales, libros, etc.

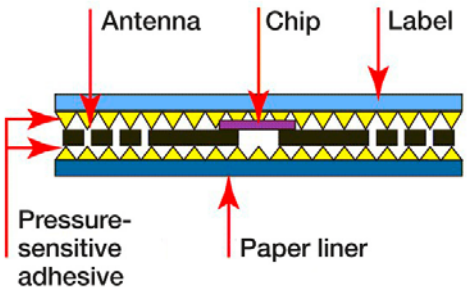


Figura 2.3. Composición etiqueta RFID (Fuente: Schreiner Group)

Figura 2.4. Etiquetas RFID (Fuente: *Alien Technology*)

Los *tags* RFID pueden tener múltiples tamaños, formas, materiales que caracterizan al *tag* y le proporcionan una ganancia dependiendo del ámbito de uso. Esto invita a plantear una clasificación dependiendo del comportamiento o del modo de operar de los *tags* RFID.

Según el tipo de acceso a memoria:

- ***Read Only*:** El identificador del objeto viene grabado de fábrica y no puede ser modificado.
- ***Write Once:*** Puede modificarse el identificador del objeto que introdujo el fabricante. Una vez cambiado, es imposible volverlo a cambiar.
- ***Write many***: Puede modificarse el valor de identificación tantas veces como se desee.

Figura 2.5. Estructura del código EPC (Fuente: GS1 Dominicana).

Según la fuente de alimentación:

- **Pasivos:** Obtienen la energía contenida en las ondas electromagnéticas transmitidas por el lector, para alimentar mediante ondas eléctricas los circuitos integrados del *tag*. Este tipo de *tag* reduce los costes del sistema RFID, pero su rango de aplicación se reduce.

- **Activos:** Poseen una fuente de alimentación en forma de batería encargada de alimentar el circuito integrado y realizar la comunicación mediante RF. Tienen un coste más elevado pero proporcionan más fiabilidad al sistema RFID, permitiendo un mayor alcance y mayor capacidad de almacenar información.

Esto le otorga un valor de funcionalidad que no se reduce a la transmisión del código EPC, como por ejemplo, almacenar el origen y el destino de la comunicación, o equipar sensores de todo tipo.

- **Semiactivos o Semipasivos:** Incorporan una batería para alimentar el circuito integrado (IC), pero para establecer la comunicación utilizan la energía de las ondas electromagnéticas. Este tipo de *tag* es una implementación intermedia para integrar las ventajas de los *tags* pasivos y activos. También incluyen sensores para ampliar la funcionalidad del sistema RFID.

Tag Pasivo	Tag Activo
Funciona sin batería	Funciona con batería
Relativamente económico	Relativamente costoso
Ciclo de vida ilimitado	Ciclo de vida limitado por la batería
Poco peso	Mayor peso
Alcance limitado (3 - 5m)	Mayor alcance (100 m)
Sensible al ruido	Mayor inmunidad ante presencia de ruido
Dependencia de la señal del dispositivo lector	Trasmisor propio
Requiere dispositivos lectores potentes	Relaja el requisito de potencia de los lectores
Velocidad de transmisión baja	Velocidad de transmisión alta
Lectura simultánea baja	Lectura simultánea alta
Alta sensibilidad de orientación	Menor sensibilidad de orientación

Figura 2.6. Comparación *tag* pasivo vs activo (2)

En este proyecto se utilizaran *tags* pasivos equipados con sensores, que se prevé que sea una herramienta que ayude al crecimiento de la tecnología RFID (12).

2.4 Transmisor/Receptor (lector RFID)

El lector, en ámbito electrónico “transreceptor”, es el encargado de emitir las ondas electromagnéticas hacia los *tags* y de recibir la señal generada por éstos.

Según el tipo de lector, las antenas pueden ser integradas, o bien, pueden ser conectadas mediantes puertos de entradas coaxiales. También disponen de interfaces de conexión *Ethernet* o puerto serial RS-232 para poderse conectar al computador.

En función de su ámbito de aplicación y diseño, se pueden encontrar diferentes tipos de lectores RFID. Por ejemplo, en una cadena de montaje donde los productos van circulando mediante una cinta transportadora, sería conveniente situarlo en una posición fija orientada hacia al *tag*. En el control de accesos de personas, se reviste el marco que conforma la puerta.

No obstante, teniendo en cuenta la variedad de diseños de los lectores, cabe destacar los factores relativos a la transmisión de la señal de radiofrecuencia como la potencia, la amplificación y el ancho de banda en el que operan.

2.5 Bandas de Frecuencia de RFID

Los sistemas RFID dependen de forma crucial de la frecuencia en que emiten sus ondas electromagnéticas, tanto para aportar la energía suficiente a las etiquetas, como para determinar el protocolo usado para establecer la comunicación entre *tag* y lector. La elección de una determinada frecuencia, la potencia y el protocolo tendrán una influencia directa en el alcance, el coste y las características a nivel de aplicación que se van a proporcionar al usuario.

Cabe mencionar que la gestión del espectro está regulada por la administración estatal y ésta depende de la normativa del país o continente donde uno se sitúe. Existen bandas de frecuencia licenciadas o privadas (pago) y bandas de frecuencia no licenciadas o comunes (libres).

La tecnología RFID opera en bandas de frecuencias comunes y van variando desde los 100Khz hasta los 5GHZ en espacios reducidos del espectro de frecuencias:

- **LF (*Low Frecuency*)[100kHz-500kHz]:** El estándar más común para RFID se encuentra en **125/134 kHz.** Los sistemas que operan bajo este rango de frecuencias tienen las siguientes características: un alcance de **45 cm**, baja velocidad de transmisión, costes bajos y trabajan bien junto a materiales líquidos y metálicos.
    - **Aplicaciones** → EAS (antirrobo), llaves para automóviles.

- **HF (High Frecuency):** La banda internacional de frecuencias los fija en **13,56 MHz.** Los sistemas se caracterizan por un rango de alcance de **1 a 3 metros**, velocidades de transmisión medias, trabajan bien con materiales líquidos pero resulta problemático ante materiales metálicos.
    - **Aplicaciones** → Telefonía móvil (NFC)

- **UHF (Ultra High Frecuency)[400 MHz-1000 MHz]:** El estándar común internacional es de **850-950 MHz.** Los sistemas que trabajan bajo este rango de frecuencias se caracterizan por un alcance de **3 a 10 metros**, alta velocidad de transmisión, resulta problemático con materiales líquidos y metálicos así como en entornos húmedos y proveen mecanismos de anticolisión.
    - **Aplicaciones** → Gestión de la cadena de suministros y trazabilidad de objetos.

- **Microondas [2,4 -6GHz]:** El estándar fija la banda de uso en **2.4-2.45Ghz.** Sus sistemas poseen características similares a UHF pero con mayores velocidades de transmisión y un precio mayor. El rango de alcance se sitúa en más de **10 metros**.
    - **Aplicaciones** → Peajes y control ferroviario.

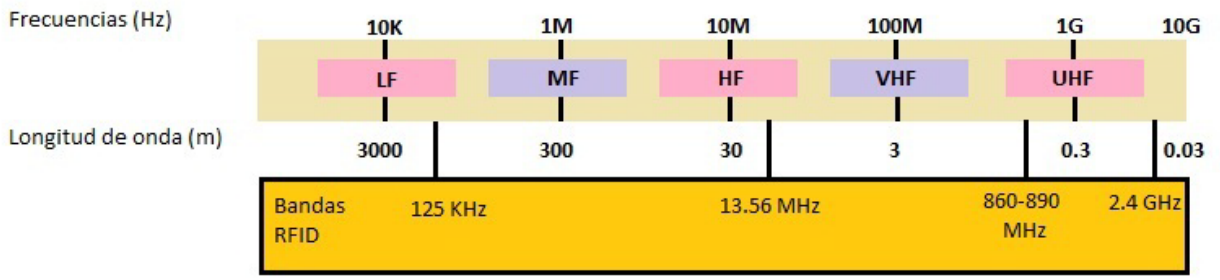


Figura 2.7. Rango de frecuencias reservadas para RFID

2.6 Estándares y protocolos RFID

2.6.1 ¿Por qué un estándar?

En los últimos años, el uso de la tecnología RFID va en aumento, sus posibilidades técnicas dan lugar a nuevas soluciones y, al mismo tiempo, implica que se establezca una normativa internacional que permita la interoperabilidad de los sistemas RFID de todos los proveedores de soluciones, teniendo en cuenta la globalización de la economía.

No existe una única organización internacional que se encargue de la normalización de la tecnología RFID, debido a que se debe respetar la legislación de cada país en cuanto a la asignación del espectro. Estas organizaciones de regularización, dentro de los espacios legales para cada país, deben fijar una frecuencia de trabajo, la potencia de emisión y el tiempo máximo de comunicación entre etiquetas y lectores. En definitiva una guía que especifica sobre cómo debe operar el sistema RFID.

Los estándares desarrollados más importantes para la tecnología RFID son (13):

- **ISO** (*International Organitzation for Standaritzation*)
- **EPC Global**
- **IEC** (*Internation Electrotechnical Commission*)
- **JTC 1** (*Joint Technical Committee*)
- **FCC** (*Federal Communication Commission*) en USA.
- **ETSI** (*European Telecommunication Standards Institute*) en Europa.

2.6.2 Protocolo *EPC Class-1 Gen-2*

EPC Class-1 Gen-2 (11) es el protocolo de comunicación desarrollado por *EPC Global*, tanto a nivel físico (señalización de la comunicación, modulación, protocolo de acceso al medio de anticolisión), como a nivel lógico (comandos operacionales compartidos entre *tag-interrogador*), utilizado para la tecnología RFID que opera con dispositivos pasivos y en la banda de frecuencia UHF de 860-960 MHZ. El sistema compromete lectores (*interrogador*es) y a *tags* (*transpondedor*es).

El protocolo especifica que, un *interrogador* (*reader*) transmite la información al *tag* bajo una frecuencia de 860-960MHz. El *tag* recibe la información y la energía suficiente para excitar sus circuitos, sin emplear una batería o fuente de alimentación extra y devolver la señal en forma de respuesta, modulando el coeficiente de reflexión de su antena. La comunicación entre *tag-interrogador* es del tipo *half-duplex*, lo que significa que el *interrogador* habla, el *tag* escucha y viceversa.

Esta especificación, la *Class-1* (clase-1), define:

- Un código de identificación (EPC).
- Un identificador de etiqueta (TID).
- Funcionalidad *"kill"* que permite desactivar de forma permanente el *tag*.
- Memoria opcional de usuario.
- *Password* de acceso al *tag*, de forma opcional.

En cuanto a Gen-2 cabe destacar:

- Los tipos de modulación empleados para la comunicación:
    - DB-ASK (*Double Sideband-Amplitude Shift Keying*)
    - SS-ASK (*Single Sideband-Amplitude Shift Keying*)
    - PR-ASK (*Phase-Reversal Amplitude Shift Keying*)

- Las velocidades de transmisión de los *tags* se sitúan en :
    - 80 Kbps
    - 160 Kbps
    - 320 Kbps
    - 640 Kbps

- La longitud del EPC varía considerablemente de 96 bits a 256 bits.
- Los *tags* Gen2 aportan EPC (*Electronic Product Code*) de 256 bits, mientras que la Gen1 soportaba hasta 96 bits.
- En entornos con más de un interrogador, las respuestas de los *tags* pueden colisionar con las transmisiones de los interrogadores. Gen-2 introduce los métodos de *Dense-Interrogator channelized signaling* para evitar la colisión. Se basan en la Multipexación por División de la Frecuencia (FDM), esto supone el uso de unas frecuencias para las respuestas de los *tags*, y otras frecuencias distintas para las preguntas de los interrogadores.

En el presente proyecto, se trabaja con prototipos fabricados con los requisitos de la especificación de *EPC Class-1 Gen 2 v02*.

2.7 Aplicaciones y usos de RFID

2.7.1 Tipos de soluciones

Los sistemas RFID se clasifican en base al ciclo de vida del *tag*:

- Ciclo cerrado → Son soluciones en los que el *tag*, una vez transcurre el ciclo de vida del producto, considerando el tiempo que tiene sentido tener reconocimiento de su identificación, se vuelve a reutilizar y su ubicación vuelve al punto de partida inicial. En este tipo de soluciones, no se estima un gran coste en el mantenimiento del sistema, puesto que los elementos de identificación se vuelven a utilizar.

- Ciclo abierto → Una vez realizado el proceso identificación a lo largo del ciclo de vida del producto, no hay posibilidad de reutilizar el *tag*, por la integridad del mismo, o bien porque se produce un traspaso de una empresa a otra. En este tipo de soluciones los costes son más incidentes en el mantenimiento del sistema, y la identificación se plantea de una forma conjunta, en lugar de identificar de una forma más precisa la unidad.

2.7.2 Aplicaciones RFID

Existe una gran variedad de soluciones y aplicaciones RFID. Si se analizan los beneficios de la tecnología RFID, se pueden mejorar los procesos de negocio de las empresas ofreciendo soluciones a sus problemas. Antes de observar qué beneficios aporta esta tecnología en sus diferentes ámbitos de uso y sectores de mercado, se van a mostrar una serie de aplicaciones, para analizar el tipo y nivel de utilidad de la tecnología RFID:

Control de acceso

A pesar de que el control de la privacidad y la autenticación son dos vulnerabilidades de los sistemas RFID, existen soluciones para identificar personas y permitir los accesos autorizados:

- **Identificación de clientes:** En gimnasios, hoteles, empresas, clubes deportivos**,** las tarjetas incluyen datos personales, fotografías, o los servicios contratados por el cliente, horarios, etc. De este modo se puede realizar un acceso personalizado.

- **Identificación de pacientes.** La identificación inequívoca de pacientes es uno de los problemas que suceden en los centros sanitarios, incluso tiene mención en el Plan de Calidad para el Sistema Nacional de Salud del Ministerio de Sanidad Servicios Sociales e Igualdad (14). Un ejemplo de uso se produce en el contexto de una enfermera que desconoce la medicación del paciente. La enfermera mediante la lectura de la etiqueta asignada al paciente, puede averiguar *"in situ"* qué plan de medicación tiene asignado ese paciente e incluso tener conocimiento de los medicamentos que le producen reacción alérgica.

Gestión de tienda inteligente

La tecnología RFID agiliza los sistemas de cobro en las cajas de los comercios. Esto supone evitar largas colas de espera a los usuarios. Además, el control de las estanterías a tiempo real permite un mayor control de la mercancía y de los sistemas de stock.

Gestión de almacén inteligente

Automatizar los procesos de logística supone un uso optimizado del stock. Esto permite ajustar la producción, aproximarla a las necesidades reales y garantizar la eficacia en los plazos de entrega. Un sistema de gestión de almacén debe tener:

- Zona de etiquetado para los productos que carezcan de su identificación.
- Terminales lectores móviles para identificar los productos desde cualquier ubicación.
- Terminales lectores fijos que identifiquen las entradas y salidas de productos.

Cadena de suministro

EPC Global se centra en este ámbito de aplicación donde la tecnología RFID, permite la identificación de cada producto substituyendo al sistema de códigos de barras. En el momento de fabricación del producto se insertará el *tag*, y a lo largo del ciclo de vida del producto hasta su comercialización, se podrá tener el control del mismo. Durante el ciclo de vida, intervienen una gran cantidad de empresas, otorgándole valor al producto. Este hecho implica la existencia de un estándar internacional que permita que las empresas cooperen entre sí compartiendo información y en definitiva agilizar los procesos. *EPC Global* es la encargada de cumplir esa tarea y su propuesta es codificar cada objeto mediante un único código EPC (*Electronic Product Code*).

Figura 2.8. Control de productos por RFID (Fuente: AREF35).

RFID y sensores ambientales

La integración de RFID y sensores aumenta el grado de aplicación de la tecnología y las posibilidades de ésta. En los controles portuarios de mercancías, se emplea el uso del GPS junto con RFID como solución de control y seguridad de mercancías muy valiosas. La armada de los Estados Unidos, por ejemplo, equipa los *tags* RFID junto con sensores de temperatura y shock [1] para controlar la integración de su stock en munición.

Sistemas de cobro en autopistas

En la red de peajes de las autopistas, se reservan carriles especiales para evitar la aglomeración de vehículos basado en tecnología RFID. Cuando el coche se aproxima a la zona de peaje reduce la velocidad y cuando llega a una distancia dentro de la cobertura de alcance de RFID, permite la identificación del dispositivo. El dispositivo está vinculado a una cuenta bancaria donde posteriormente se realiza el cobro.

2.7.3 Beneficios del sistema RFID

El estándar *EPC Global UHF Class1* tiene su principal ámbito de aplicación en la cadena de suministros. A continuación se muestran los beneficios que aporta RFID:

- Visibilidad y trazabilidad del stock a lo largo de la cadena de suministro. Permite un control fiable de todos los productos y ajustar la producción del stock de forma más precisa.
- Mejora del nivel de servicio. Al tener trazabilidad de los productos, la equivocación en los pedidos se reduce. Esto evita los sobrecostes que suponen el tener que devolver el producto y reenviarlo de forma correcta.
- Mejora de los procesos logísticos. Al no necesitar una línea de visión directa con los productos, se pueden introducir arcos de lectura para identificar los productos. Ante un volumen grande de productos simplifica la tarea de identificación al no tener que ir uno a uno identificando su código con un lector (Ver figura 2.8).

[1] Sensor de shock: Mide las vibraciones producidas al agitarlo.

- Se reduce el impacto de pérdidas o hurtos de los productos durante el ciclo de vida del producto.
- Ante la posible retirada de productos por atentar contra la salud o por seguridad, RFID permite agilizar este proceso al tener un mejor control del stock.

En definitiva, RFID aporta una mejora sustancial en cuanto al seguimiento de envíos y trazabilidad de los productos a lo largo de la cadena de distribución. Permite a los clientes y distribuidores saber con exactitud el número de productos y los plazos en las entregas.

CAPÍTULO 3

ACELERÓMETROS

3.1 Definición de acelerómetro

Un acelerómetro es un instrumento capaz de medir la aceleración, normalmente de objetos, en los que va acoplado. Esto se produce mediante el uso de una masa inercial unida a un elemento elástico, introducida en el dispositivo acelerómetro.

La aceleración es la fuerza que se aplica a la masa y ésta puede ser estática, como en el caso de la gravedad, o dinámicas, producidas por una agitación o vibración.

En este apartado se van a revisar los principios básicos en los que se fundamenta el uso del acelerómetro, los tipos de acelerómetros y aplicaciones con acelerómetros. Se pretende proporcionar unas nociones básicas a cerca de los acelerómetros para entender su funcionamiento y su utilidad para entender el propósito de estudio.

3.2 Principios básicos

El principio físico que explica el funcionamiento de este dispositivo electrónico es la segunda ley de Newton. Newton introduce $F = m.a$. Esto implica que en el sistema compuesto de una masa y un resorte (muelle o material elástico) se aplique una fuerza de acuerdo a la ecuación. La fuerza hace que el material elástico se expanda y comprima a lo largo de una dirección. Por lo tanto, si se quiere calcular el movimiento en un plano, se deberá duplicar el sistema, y si se realiza en 3 dimensiones, se deberá triplicar.

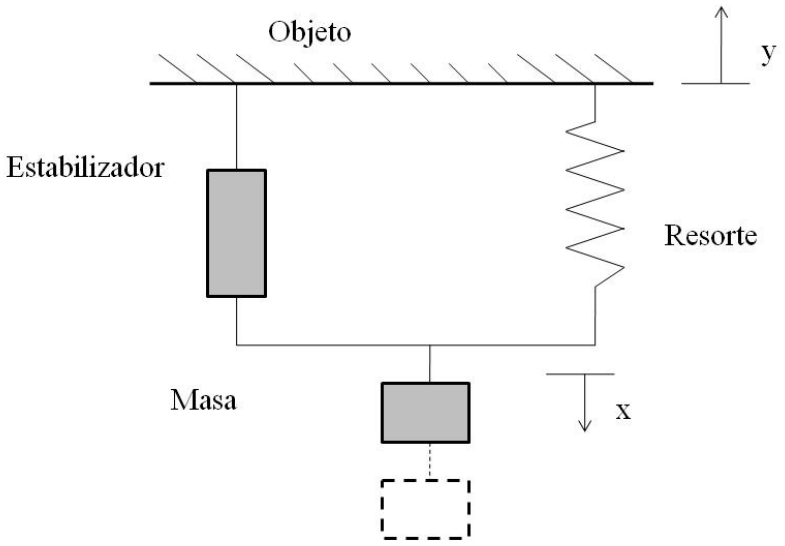


Figura 3.1. Principio físico funcionamiento acelerómetro

En la figura anterior se ilustra el funcionamiento del acelerómetro. Al aplicar una fuerza al objeto en el sentido del gesto (y), el resorte se expande y la masa experimenta un desplazamiento en el sentido (x). Cuando la masa conectada al resorte llega a su máxima expansión, el resorte empieza a comprimirse produciendo una fuerza en el sentido (y) y una aceleración en el sentido (y). El uso del estabilizador es necesario para evitar que el sistema oscile.

3.3 Tipos de acelerómetros

3.3.1 Acelerómetros mecánicos

Son los acelerómetros que más se aproximan al principio físico comentado anteriormente. Emplean un resorte unido a una masa inercial y en función de la fuerza que se le aplica, la masa va deformando el material contenido en la galga extensométrica[2], que genera una variación de corriente que puede traducirse en una aceleración.

[2] Galga Extensiométrica: Es un dispositivo electrónico que mide la deformación. Ante la deformación producida por la masa inercial, se produce una variación de la resistencia eléctrica del dispositivo.

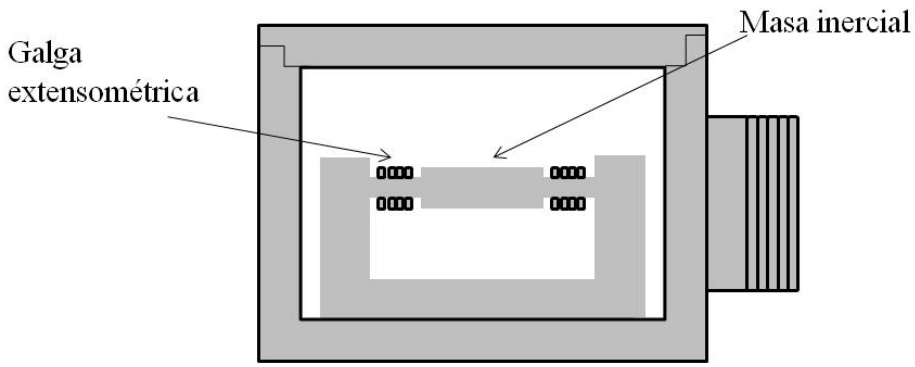


Figura 3.2. Funcionamiento acelerómetros mecánicos

3.3.2 Acelerómetros capacitivos

Emplean microcondensadores y modifican la posición de las placas que los conforman. Al aplicar una aceleración al objeto que contiene el acelerómetro se produce un desplazamiento de una de las placas que conforman el condensador, produciendo una modificación en su capacidad. Esta modificación de la capacidad produce un voltaje de salida.

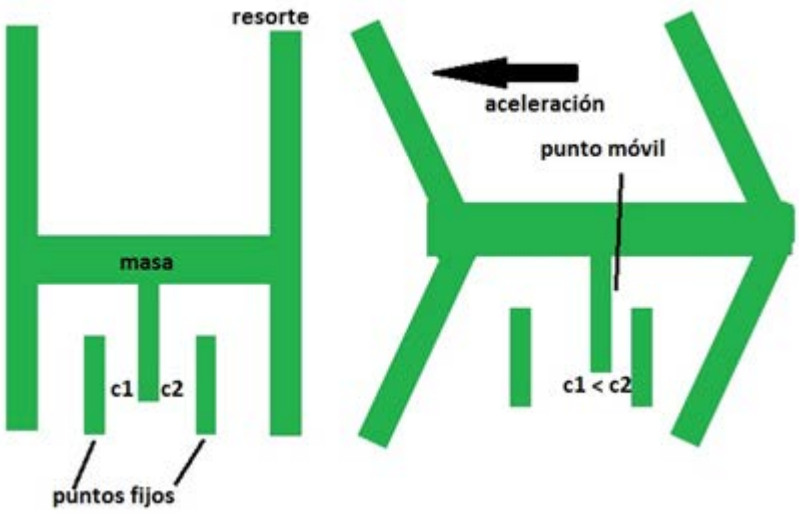


Figura 3.3. Funcionamiento acelerómetros capacitivos

En la figura anterior en forma de “H” se puede ver el estado en reposo del acelerómetro. Los puntos fijos pertenecen a las placas de los condensadores.

Al aplicar una aceleración, el resorte produce un desplazamiento contrario y la placa que permanece en movimiento (punto móvil) produce una variación en las capacidades de los condensadores. La capacidad de los condensadores depende de las distancias que separan las placas (puntos fijos-puntos móviles). Al producir una fuerza sobre una de las placas (punto móvil), se produce una variación proporcional en el campo eléctrico.

El acelerómetro utilizado en este estudio está fundamentado en este tipo de acelerómetro.

3.4 Aplicaciones

La aceleración es una magnitud física que se manifiesta en forma de gravedad, vibración o actividad sísmica, entre otros. La medición de la aceleración da a lugar a numerosas aplicaciones y utilidades.

Smartphones

En la actualidad muchos dispositivos tecnológicos llevan incorporado un acelerómetro. La gran mayoría de los Smartphones lo llevan incorporado. En este tipo de dispositivos es utilizado para determinar la posición, por ejemplo, durante el uso del GPs asistido. Mediante la ayuda del GPs y considerando que se conoce la posición y la velocidad original de un cuerpo, calculando la aceleración, se pueden deducir los desplazamientos y averiguar la posición en todo momento.

Personal Computer

Las empresas de computadoras, más exactamente en el mercado de *laptops,* buscan sistemas para garantizar la integridad de los dispositivos de almacenamiento masivo. Los constantes cambios de ubicación y sacudidas a las que están sometidos los *laptops* pueden dañar los discos duros. Mediante el uso de acelerómetros se podría calcular el impacto de la sacudida y proveer mecanismos para proteger los posibles daños en el *hardware.*

Videojuegos: Nintendo Wii

La Nintendo Wii se basa en la utilización de un mando de control basado en un acelerómetro capaz de detectar los movimientos del usuario en 3D, ajustando al máximo la experiencia del juego a la realidad.

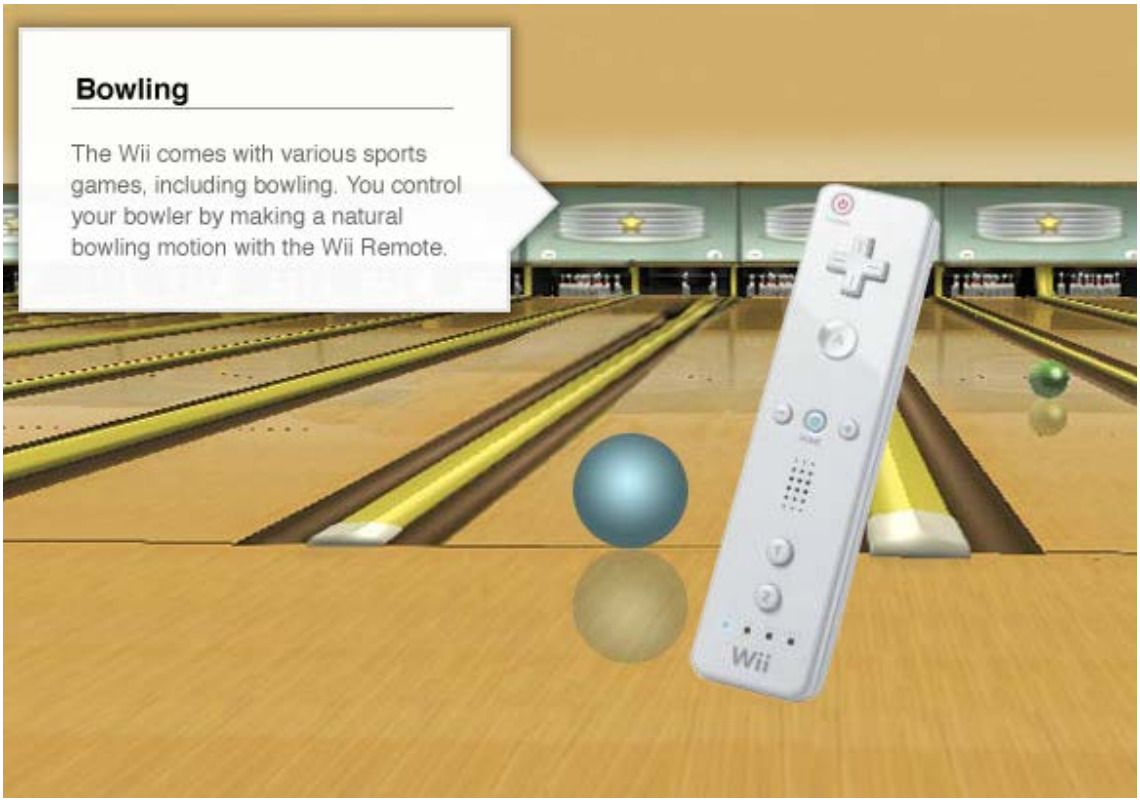


Figura 3.4. Utilización del acelerómetro en la consola WII (Fuente: www.technologyreview.com)

CAPÍTULO 4

MATERIAL A UTILIZAR

4.1 Kineo RFID *tag*

El *tag* de Kineo es un prototipo de la empresa Farsens S.L, encargado de transmitir el código EPC a un lector homologado por *EPC C1G2* sin la necesidad de una batería, pudiendo alcanzar 1,5 metros de distancia. Está equipado con un sensor acelerómetro y no utiliza baterías para alimentar el sensor. En una cadena de distribución permitiría conocer la posición del producto y garantizar la integridad del mismo a lo largo de su ciclo de vida.

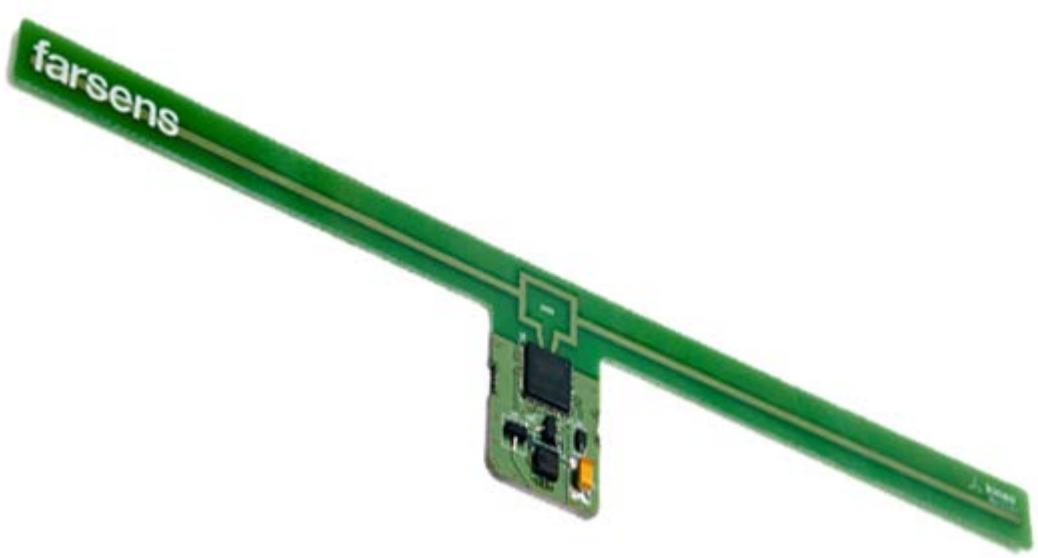


Figura 4.1. Prototipo Farsens Kineo (Fuente: Farsens)

4.1.1 ANDY100

Es el encargado de alimentar los sensores equipados mediante la energía recibida por la señal de radiofrecuencia y de la comunicación con los sensores equipados. Transmite las operaciones dirigidas por el lector al acelerómetro para poder iniciar la medición.

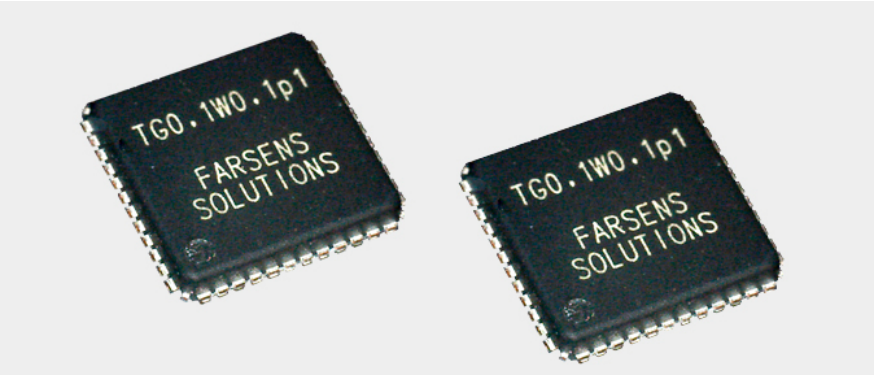


Figura 4.2. ANDY 100 circuito integrado del *tag* Farsens (Fuente: Farsens)

4.1.2 LIS3DH

Es el circuito integrado que genera los datos del acelerómetro en las escalas ±2g/±4g/±8g/±16g y es capaz de producir salidas de datos a tasas comprendidas entre 1khz -5khz.

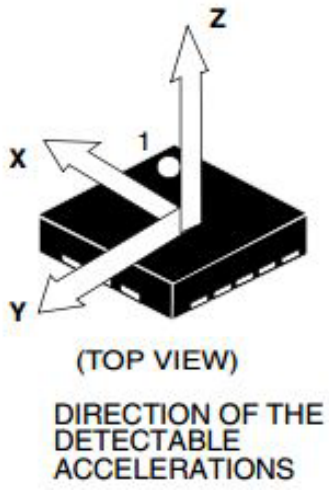


Figura 4.3. Acelerómetro LIS3DH (Fuente: STMicroelectronics)

4.1.3 MAX6427

Es el circuito integrado que actúa como sistema de inicio y el encargado de controlar la alimentación del sistema.

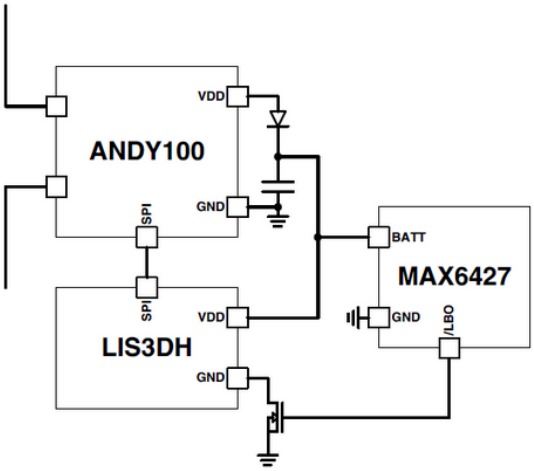


Figura 4.4. Circuito integrado MAX6427 (Fuente: Farsens)

4.2 Lector del *tag*

Sirit Infinity 610 (44), compatible con *EPC Gen2*, es un lector que soporta UHF y que opera en la banda de frecuencia de 902-928 Hz. Incorpora un software de configuración sencilla e intuitiva, lo que le permite agilizar la gestión de las lecturas de los *tags*.

Figura 4.5. Lector *Sirit Infinity 610* (Fuente: www.atlasrfidstore.com)

Se pueden conectar hasta 4 antenas pero para este caso de estudio, con una antena se pueden tomar las mediciones que se precisan. Se utiliza en cadenas de distribución, control de portales de acceso y sistemas de transporte.

4.3 Any*Reader*

Es el software proporcionado por la empresa Farsens S.L. para realizar las lecturas proporcionadas por sus prototipos. Cabe mencionar que este software es compatible con un número limitado de lectores entre los que se encuentra el *Sirit Infinity 610*.

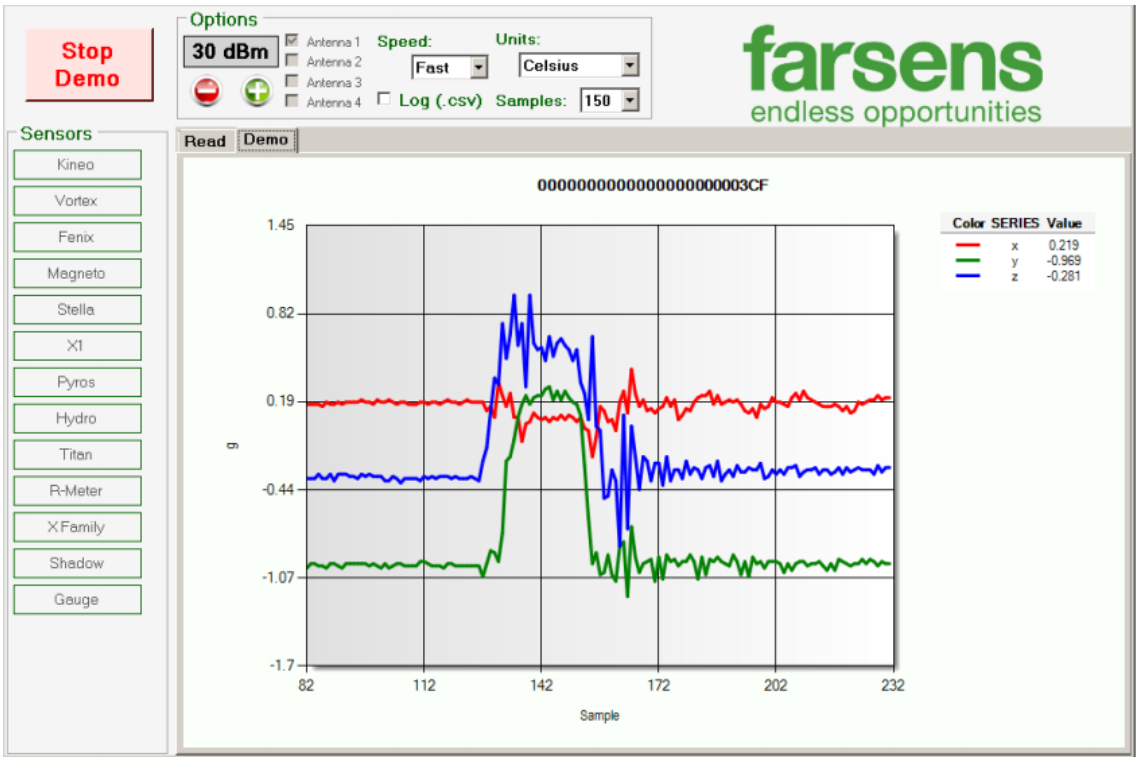


Figura 4.6. Monitorización del *tag* Kineo de Farsens mediante el software AnyReader (Fuente: Farsens)

En la figura anterior se puede ver ilustrado su comportamiento al monitorizar los datos generados por el sensor acelerómetro Kineo. Una vez realizada la lectura de los datos, éstos se almacenan en un archivo con extensión ".csv".

4.4 *Kinetic Pro App*

Como elemento adicional en este caso de estudio se introdujo la lectura de los datos acelerómetros generados por un dispositivo móvil. El software elegido para ese propósito se trata de *Kinetic Pro App* (15) bajo la plataforma Android. Este software permite generar datos del acelerómetro y almacenarlos con extensión ".csv". Considerando que es el mismo formato que emplea Any*Reader*, se puede estandarizar el proceso de lectura en el momento de analizar este tipo de archivos y extraer la información.

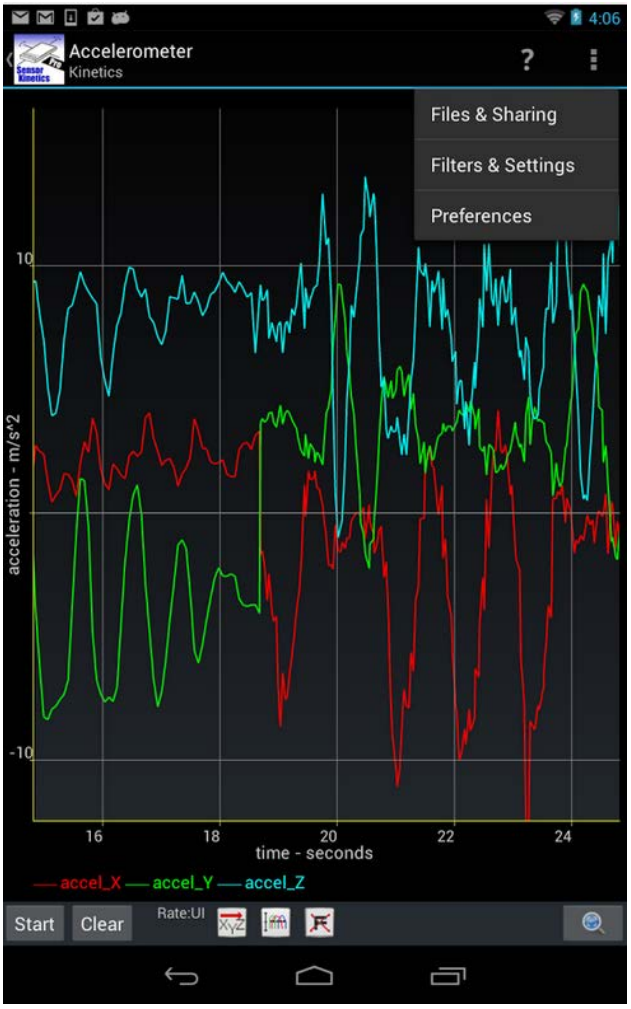


Figura 4.7. Perspectiva de monitorización de *Kinectic Pro App*

4.5 *R Studio*

Es el *ide*[3] utilizado para desarrollar *scripts* en lenguaje R. Se emplearan para extraer las muestras, analizar los datos con su conjunto de herramientas matemáticas y aplicar el algoritmo de *Machine Learning* para la clasificación de los datos generados por el acelerómetro.

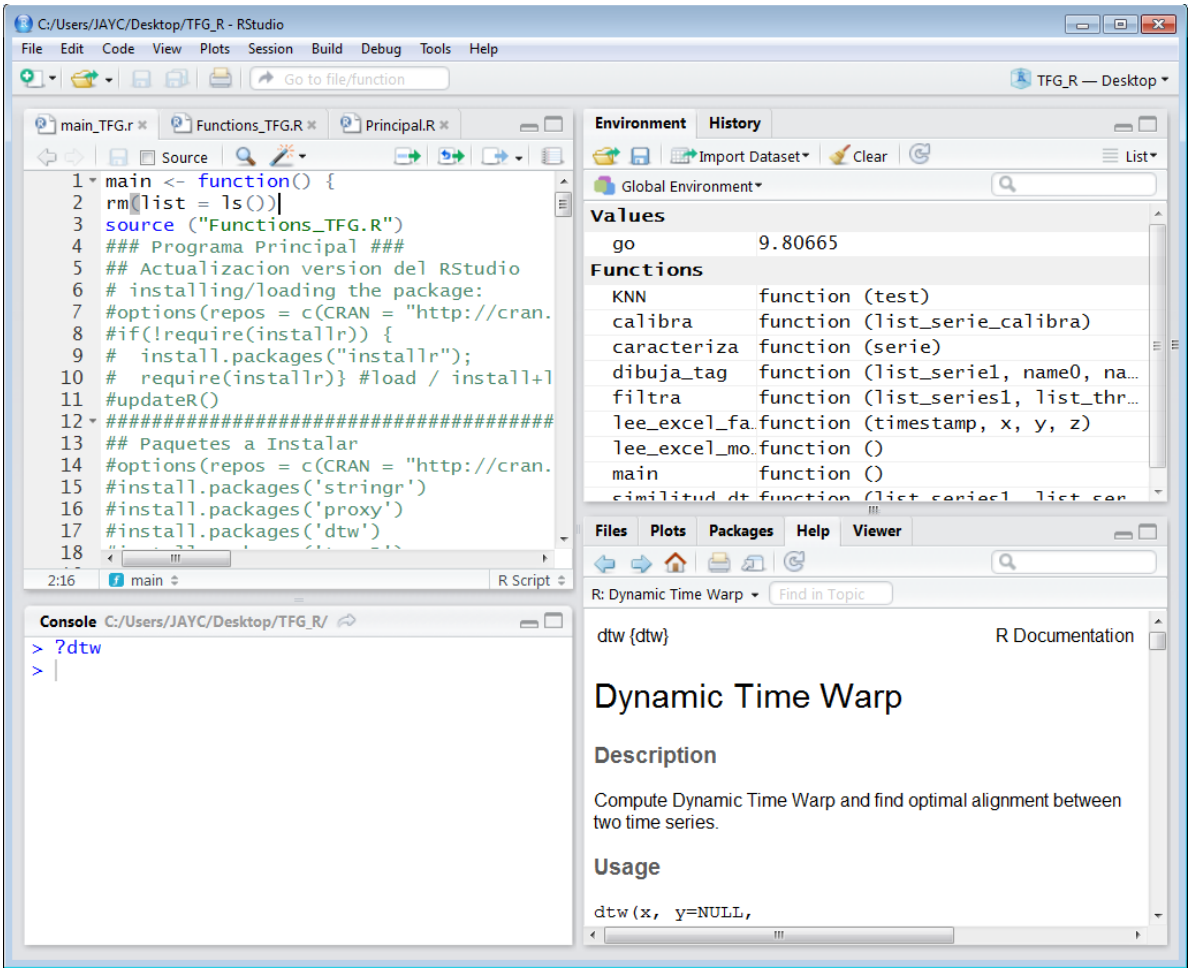


Figura 4.8. Perspectiva principal de la herramienta de desarrollo *RStudio*

[3] Entorno de desarrollo integrado. Es un software que contiene un conjunto de herramientas para el desarrollo de lenguajes de programación.

CAPÍTULO 5

PROBLEMÁTICA Y ESTUDIO DEL CASO

5.1 Introducción al problema

En este proyecto se pretende estudiar el funcionamiento de la tecnología RFID y sensores en el campo de *Ubiquitous Computing* y *Human Interaction Recognition*.

Esta tecnología es vulnerable a ataques tipo '*man on the middle*'[4] o '*ghost and leech*'[5] que pueden interferir entre el *tag* y el lector (16). Es muy sencillo introducir una intromisión en el sistema RFID y realizar actividades malintencionadas.

5.2 Seguridad en RFID

Como se ha comentado en apartados anteriores, la seguridad es una de las debilidades que conllevan los sistemas RFID. Es sabido que cualquier persona en posesión de un lector apropiado puede interferir en la comunicación e interceptar la información contenida en la etiqueta. Un sistema RFID debe protegerse de:

- Lecturas/escrituras indeseadas en las etiquetas
- Falsificación de etiquetas que permitan el acceso a servicios restringidos.

Las investigaciones acerca de la seguridad y privacidad de la tecnología RFID se centran en proveer al propietario del sistema, el control sobre cuándo efectuar la comunicación entre los dispositivos RFID. Existen tres técnicas para afrontar este reto:

- *Reader-to-tag authentication*: Técnicas para detectar cuando un lector RFID está autorizado a transmitir.

[4] En criptografía se refiere al tipo de ataque en que un enemigo tiene la capacidad de interceptar y suplantar la identidad del emisor.
[5] Ataque criptográfico en el que un *leech* (sanguijuela) intercepta la información del emisor (*tag*) y la retransmite a un *ghost* (fantasma) que a su vez se la transmite al receptor (*reader*).

- *Tag-to-reader authentication:* Técnicas utilizadas para impedir la clonación de etiquetas y suplantación de identidad. La aplicación de protocolos criptográficos en los *tags* RFID requieren recursos en energía, memoria y capacidad de procesamiento de los circuitos. Las investigaciones se centran en desarrollar nuevos protocolos de autenticación basados en operaciones bit a bit (17) (18).

- *Context Recognition:* Técnicas para detectar información del entorno e iniciar la comunicación cuando se produzcan una serie de condiciones determinadas. Estas técnicas están dirigidas a afrontar problemas de privacidad, autenticidad y ataques '*ghost and leech*'.

5.2.1 Tipos de ataques

Los ataques a RFID más comunes son **(19)** :

- ***Counterfeiting***

  Modificar el código EPC por uno falso cuando el sistema espera una respuesta correcta. Esto da lugar a la clonación de etiquetas, por lo tanto, a falsificar la identidad del producto etiquetado.

- ***Replay o Eavesdropping***

  Interceptar datos de identificación de un *tag*, grabar los datos y retransmitirlos al lector. En los sistemas RFID, interceptar la información se consigue con el uso de un lector compatible con el estándar de los *tags*. En el caso de robo de un producto, utiliza la información grabada para que el sistema detecte que el producto permanece en su localización.

- **Denegación de Servicio**

  En este tipo de ataque, una fuente maliciosa introduce ruido en la frecuencia de comunicación con el objetivo de sabotearla e inhabilitar el sistema RFID. *EPC Global Class1 Gen 2,* ante una comunicación con ruido, inhabilita el sistema RFID.

- ***Spoofing: Man in the Middle***

  Es el ataque de suplantación de identidad. El atacante figura como un *tag* reconocido del sistema. RFID es vulnerable a este tipo de ataque debido a que no se produce ningún tipo de acuerdo previo o *handshake* para establecer la comunicación. La automatización de las lecturas y la interoperabilidad entre diferentes *tags* y lectores, hace que este sistema sea vulnerable a este tipo de ataque.

5.2.2 Soluciones de seguridad actuales

5.2.2.1 Comando "KILL"

Existen técnicas para evitar la lectura no autorizada de los *tags* RFID que se basan en bloquear la comunicación mediante la inhabilitación del *tag* (20).

En el estándar *EPC Class-1 Gen-2*, como se ha visto en puntos anteriores, se proporciona una opción de seguridad conocida como *kill command*. Esta opción se puede habilitar con un password de 32 bits y permite la desactivación del *tag* RFID de forma permanente. Con este sistema se evitan todo tipo de ataques, excepto el de *DoS*, en el que el principal objetivo es la desactivación del *tag* RFID. El inconveniente de este sistema de seguridad es que una vez desactivado el *tag* ya no puede volverse a identificar. Esta funcionalidad es de utilidad una vez el producto u objeto a identificar ha llegado a su destinatario final, ya que el proceso de identificación no tiene sentido que se produzca. Adicionalmente, el estándar provee una contraseña para la modificación del código EPC, que evita ataques del tipo *counterfeiting*, ya que si el atacante desconoce la contraseña no puede modificar el código EPC.

5.2.2.2 Caja de "Faraday[6]"

Se trata de una técnica basada en el bloqueo de las comunicaciones RFID. El *tag* se recubre con un blindaje magnético para evitar que las radiocomunicaciones accedan a los circuitos del *tag*. Esta técnica evita ataques del tipo *eavesdropping, spoofing* y

[6] Michael Faraday (22 de septiembre de 1791 - 25 de Agosto de 1867) fue un científico inglés que contribuyó en la investigación de los campos de electroquímica y electromagnetismo.

counterfeiting. No tiene aplicación para ataques de DoS, ya que durante el uso del blindaje, el sistema no puede leer el contenido del *tag*. Adicionalmente, el atacante puede robar el objeto etiquetado sin ser detectado por el sistema RFID.

5.2.2.3 Interferencia activa de señales de radiofrecuencia

Esta técnica conocida como *Active Jamming Approach* (19) consiste en transmitir en los canales de frecuencia en los que se establece la comunicación, como si se tratara de un DoS, para evitar ataques de suplantación de identidad o de interceptación de la información. Este sistema no es aplicable para ataques del tipo *DoS* o de manipulación ya que se desconoce el momento en que el atacante va a realizar el ataque. El principal inconveniente es que interrumpe la operación de lectura en el sistema RFID.

5.2.2.4 El bloqueador de *tags*

El *blocker tag* (21) es un mecanismo que sirve para garantizar la privacidad en el sistema RFID. Se establece una zona privada predefinida de *tags* RFID con sus respectivos identificadores EPC. En esta zona, los *tags* están protegidos por el *blocker tag* que se encarga de simular el conjunto de todos los identificadores EPC, de tal forma que el lector no tiene conocimiento de los productos u objetos comprados por los clientes. El principal inconveniente de esta técnica es la introducción de *blockers tags* maliciosos que crean confusión a los lectores autorizados.

5.2.2.5 Técnicas basadas en proxy

Estas técnicas añaden un nuevo dispositivo para garantizar la privacidad y la autenticación en el sistema RFID. Un *watchdog tag* (22) se encarga de decodificar las acciones de los lectores y proporcionar esta información a los usuarios mediante una interfaz gráfica. El RFID *Guardian* (23) se trata de un dispositivo que gestiona y regula la actividad del sistema RFID. Este dispositivo puede ser un PDA o bien un *Smartphone*. Entre las diferentes funciones que puede desarrollar un RFID *Guardian* comentadas en la referencia anterior, cabe destacar su uso como sistema de

autenticación. El control de accesos puede ser regulado por el RFID *Guardian* empleando autenticación *reader-to-tag*, es decir, el sistema es capaz de reconocer un lector legítimo. El riesgo de esta técnica es garantizar la autenticidad del RFID *Guardian*.

5.2.2.6 Técnicas basadas en protocolos de autenticación

Estas técnicas se basan en la capacidad de los *tags* para almacenar información secreta (20). Existen tres tipos de protocolos de autenticación:

- *Crypto tag* → Aplican técnicas básicas de criptografía en las que se comparte una clave entre *reader* y *tag*.
- *Light tag* → Aplican técnicas bit a bit. (17)
- *Gen2-tag* → Técnicas de codificación y detección de errores introducidas en el protocolo *EPC Class1 Generation 2* como CRC (*Cyclic Redundancy Check Operation*) y PRNG (*Pseudo Random Number Generator*) (24).

5.2.2.7 Otras posibles soluciones

Botón "a bordo"

Esta solución consiste en equipar el *tag* RFID con un botón para permitir el proceso de lectura. Se requiere la presencia del propietario del *tag* para presionar el botón y permitir la lectura del *tag* RFID.

La ventaja principal de este sistema es que se requiere la autorización humana para realizar la lectura del *tag* RFID y el usuario decide cuándo se puede acceder al *tag* RFID.

Este sistema no provee un mecanismo que autentique a la persona propietaria del objeto que contiene el *tag* RFID. Cualquier persona que active el botón autoriza al sistema a proceder con la lectura del *tag*.

El diseño del botón es otro hándicap ya que debido a las dimensiones y múltiples formas que puede adoptar, difieren con las dimensiones reducidas que requieren los *tags* RFID.

Teléfonos NFC (*Near Field Communication*)

NFC (24) permite integrar *tags* RFID en los dispositivos móviles y utilizar el teléfono como si se tratara de un *tag* RFID. La lectura del *tag* RFID integrado en el móvil se autorizaría habilitando la conexión NFC en el menú de ajustes del dispositivo.

NFC no es compatible con el estándar EPC que utilizan los prototipos Farsens utilizados en el caso de estudio. En la siguiente referencia (25) se plantea las barreras que suponen la integración de las tecnologías EPC y NFC.

5.3 Análisis de la solución

La solución que se plantea en este proyecto se basa en el reconocimiento del gesto realizado por el acelerómetro incorporado en el *tag* RFID de Farsens y por el acelerómetro del dispositivo móvil. El lector debe ser capaz de reconocer los gestos realizados mediante un conjunto de gestos reconocidos. Reconocer gestos con los dos dispositivos ofrece la posibilidad de implementar un sistema de autenticación.

Este sistema de autenticación se basa en la idea de realizar un primer gesto con el prototipo *tag* Farsens. El lector reconoce el gesto y asocia el código EPC con un dispositivo móvil. A continuación el dispositivo móvil es alertado y el usuario genera un segundo gesto. El sistema reconoce este segundo gesto e identifica por segunda vez al usuario.

5.3.1 Ventajas de la solución

Esta solución permite autenticar a la persona poseedora del *tag* RFID mediante el gesto realizado como si se tratara de un *password*. A su vez se autentica el *tag* RFID con el sistema, es decir, garantiza que el *tag* RFID y la persona propietaria son las autorizadas.

Esto supone un mecanismo de defensa ante ataques de suplantación de identidad como *man in the middle*, añadiendo la posibilidad de realizar una lectura autorizada del *tag* EPC. Si el *tag* efectúa un gesto reconocido, se habilita la lectura del código EPC. Si por el contrario, el *tag* no efectúa un gesto reconocido, se desactiva la posibilidad de lectura del código EPC.

El uso de una segunda autenticación con el dispositivo móvil le añade un nivel de seguridad al sistema de autenticación. Se considera que el móvil es un dispositivo personal e intransferible. En caso de la substracción de éste, el atacante debería proporcionar dos gestos reconocidos, tanto con el *tag* como con el dispositivo, lo que dificulta el ataque de suplantación de la identidad.

5.3.2 Inconvenientes de la solución

Ante un posible ataque de DoS, la solución propuesta no ofrece ningún tipo de inmunidad, ya que los datos generados por el acelerómetro se transmiten por el mismo canal de frecuencia en que se emite el código EPC y, por lo tanto, los datos del acelerómetro pueden no ser reconocidos por el lector.

Ante un ataque de manipulación del código EPC o *Counterfeiting*, el atacante podría manipular la identidad del *tag*. Esto implica que el código EPC asociado al gesto y a su vez al dispositivo móvil no sea el correcto, y el usuario realice la primera autenticación con el *tag* pero no con el dispositivo móvil.

El sistema tampoco muestra inmunidad ante un ataque del tipo *replay o eavesdropping*, debido a que la información del *tag* y los datos generados por el acelerómetro pueden ser interceptados y reproducidos por un atacante, ya que no existe ningún tipo de encriptación de la información. En los sistemas de autenticación web se garantiza la privacidad de los datos en el proceso de autenticación, con protocolos de encriptación como TLS/SSL. En la siguiente referencia (26) se pueden comprobar las estadísticas de uso de las múltiples técnicas de encriptación que emplean estos algoritmos.

El principal problema que ofrece este sistema es la reproducción del gesto por parte de un atacante, es decir, cualquier persona puede plagiar el gesto realizado y autenticarse

con el sistema. Aunque puede resultar improbable, considerando que el *tag* RFID equipado con el sensor y el dispositivo móvil son personales e intransferibles, el atacante debería substraer el *tag* y el dispositivo móvil y reproducir los gestos con ambos dispositivos. Esto no ocurre en la detección biométrica por huella dactilar (27).

5.4 Objetivos

El objetivo se centra en reconocer un gesto acorde a un conjunto de gestos reconocidos mediante un *tag* pasivo y, posteriormente, realizar el mismo reconocimiento desde un dispositivo móvil. De esta forma se implementan las bases de un sistema de autenticación.

En una primera fase se generan todos los datos correspondientes a los modelos de gestos que se van a considerar. A continuación, mediante el uso de *scripts* en R, se van a extraer los datos de los acelerómetros del formato ".csv" para poder trabajar sobre ellos. También se va a realizar una representación gráfica en los 3 ejes del conjunto de muestras analizadas.

Teniendo en cuenta el problema mencionado en el apartado anterior, las fases principales que se van a seguir son:

- Adquisición, análisis y procesado de datos (ver capítulo 6).
    - *Scripts* y métodos propios en lenguaje R.
- Caracterización de los datos (ver capítulo 7).
    - Característica → *Dynamic Time Warping*
- Clasificación y decisión de los datos (ver capítulo 8).
    - Algoritmo de clasificación → *K-Nearest Neighbour*

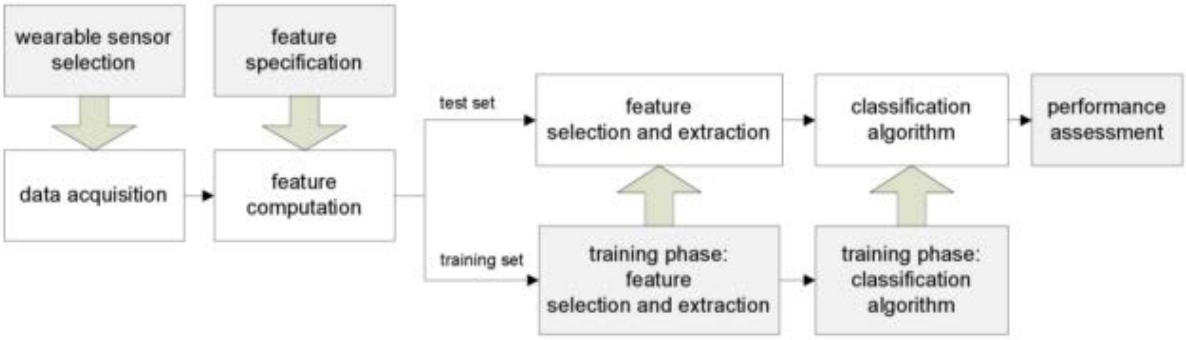


Figura 5.1. Esquema general de un sistema de clasificación basado en aprendizaje supervisado (28)**.**

CAPÍTULO 6

ANÁLISIS Y PROCESADO DE DATOS

6.1 Trabajo en el laboratorio

Las sesiones de laboratorio tienen como objetivo capturar las muestras generadas por el acelerómetro utilizando los prototipos de Farsens y el dispositivo móvil. A continuación se muestra el escenario utilizado para recoger las muestras:

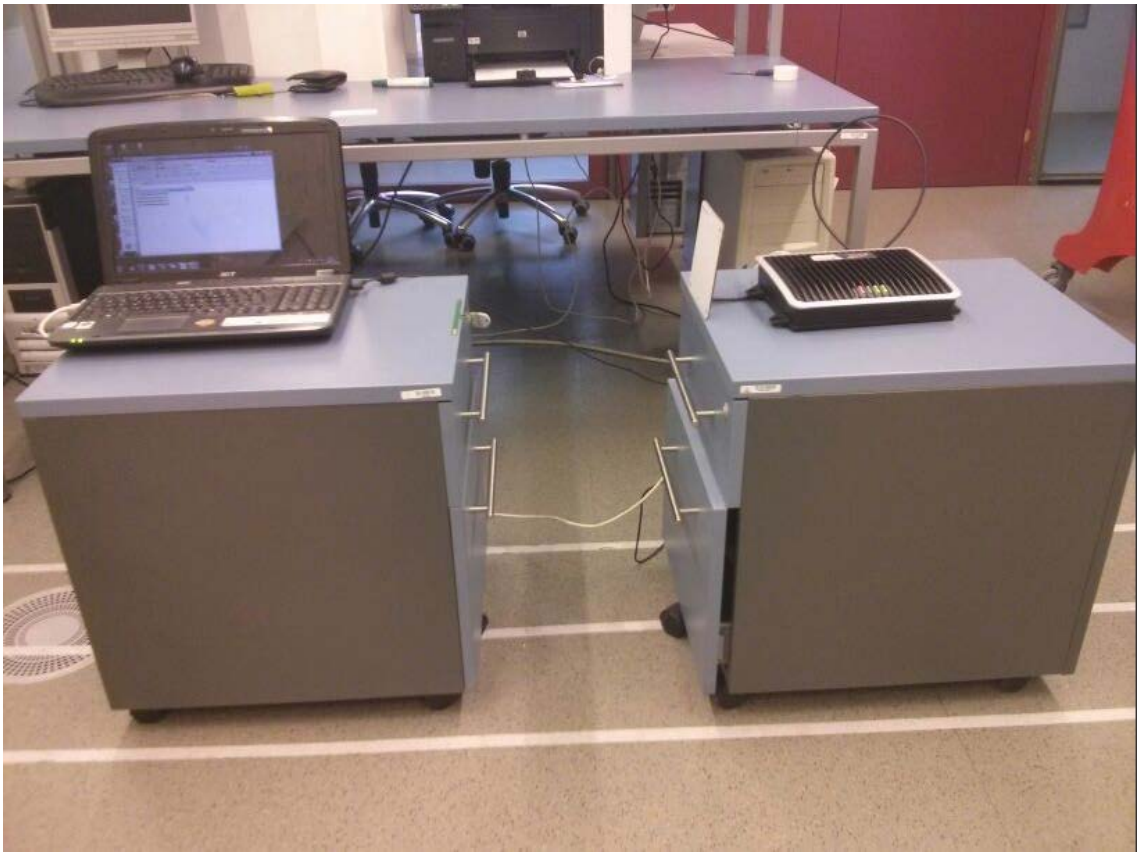


Figura 6.1. Escenario de recogida de muestras

En la parte izquierda de la imagen, se ejecuta el programa *AnyReader*. En la parte derecha de la imagen se sitúa el lector/interrogador SIRIT con su respectiva antena acoplada. A continuación, se sujeta el *tag* Farsens y se realiza uno de los gestos patrones predefinidos.

6.2 Cartera de gestos

Para elaborar una cartera de gestos, se contará con 3 patrones predefinidos. Debido al volumen de datos generados por los dispositivos acelerómetros, se fijan 3 modelos para agilizar la tarea de lectura de los datos.

6.2.1 Gesto “W”

El gesto “W”, tomando como posición inicial el prototipo Farsens, traza la trayectoria mostrada según se muestra en la figura 6.2 y en la figura 6.3:

Figura 6.2. Gesto modelo “W” instante inicial

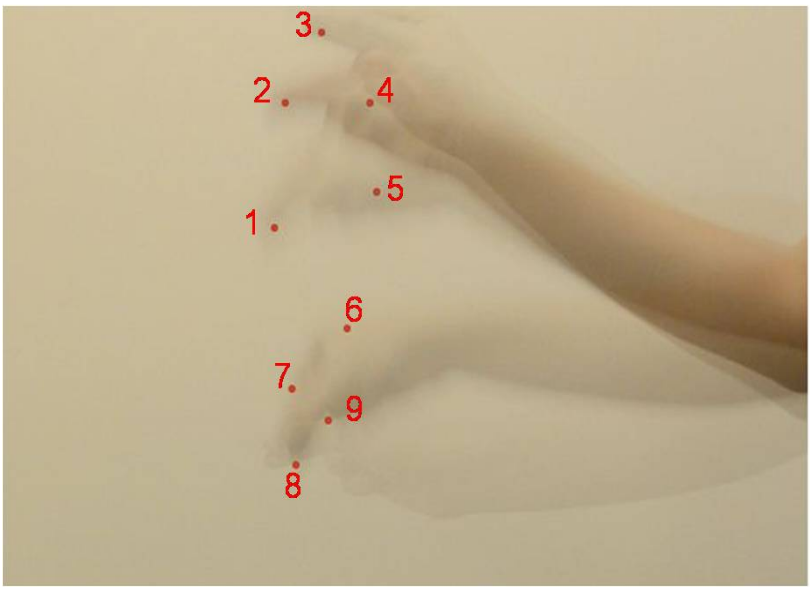


Figura 6.3. Gesto modelo "W" secuencia del gesto

El gesto se ejecuta siguiendo la secuencia de números 1,2,3,4,5,6,7,8,9. La secuencia se realiza un total de 4 repeticiones.

En el Anexo A.3 se pueden consultar las gráficas del conjunto de pruebas realizadas para este gesto.

6.2.2 Gesto "R"

Este gesto, tal y como muestra la figura siguiente, se efectúa en un total de 10 repeticiones:

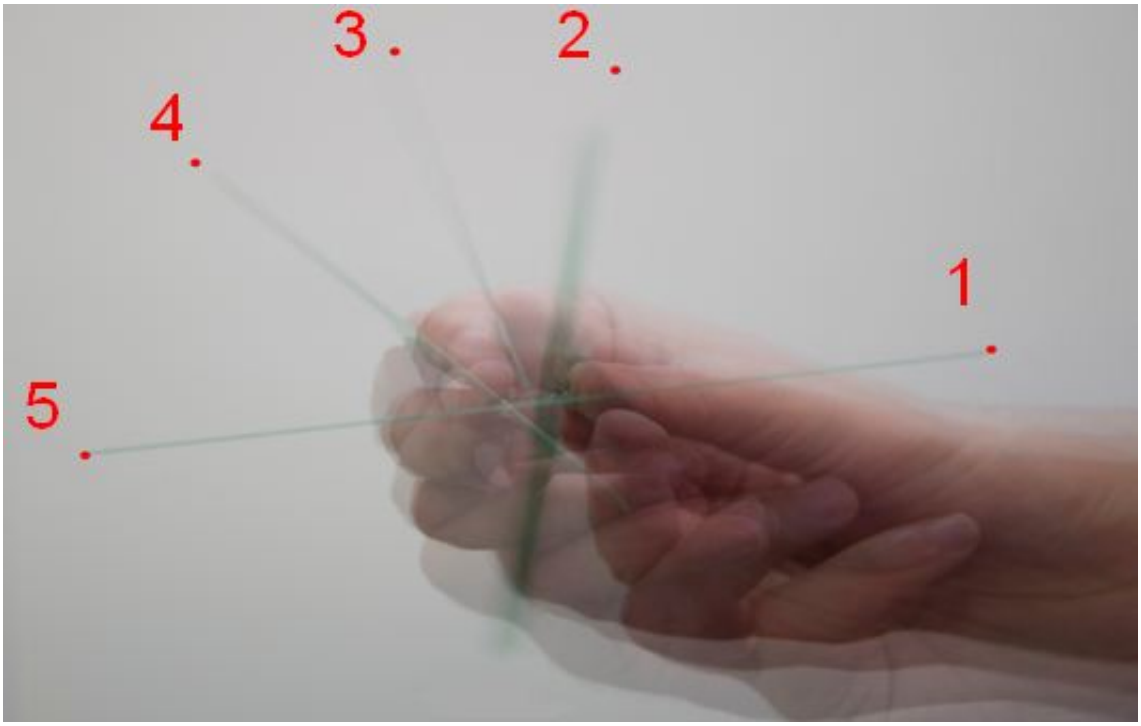


Figura 6.4. Gesto modelo "R"

La secuencia 1,2,3,4,5,4,3,2,1 se repite un total de 10 veces.

En el Anexo A.2 se pueden consultar las gráficas del conjunto de pruebas realizadas para este gesto.

6.2.3 Gesto "C"

El gesto "C" se denota por esta letra porque se trata de un combo de gestos:

- El primer gesto realiza un total de 4 repeticiones considerando la secuencia 1,2,3,2,1 (como muestra la figura 6.5) :

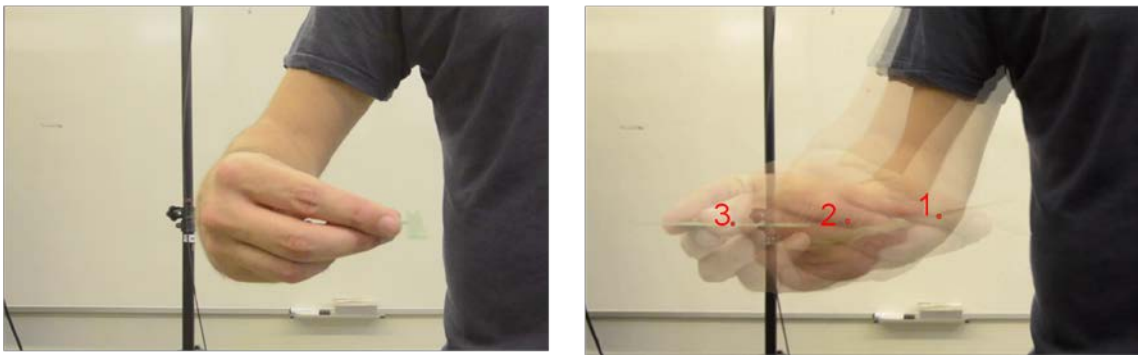


Figura 6.5. Primer gesto modelo "C"

- El segundo gesto de 4 repeticiones realiza la secuencia 1,2,3,2,1 (figura 6.6 derecha) tomando el punto 1 como instante inicial del gesto (figura 6.6 izquierda):

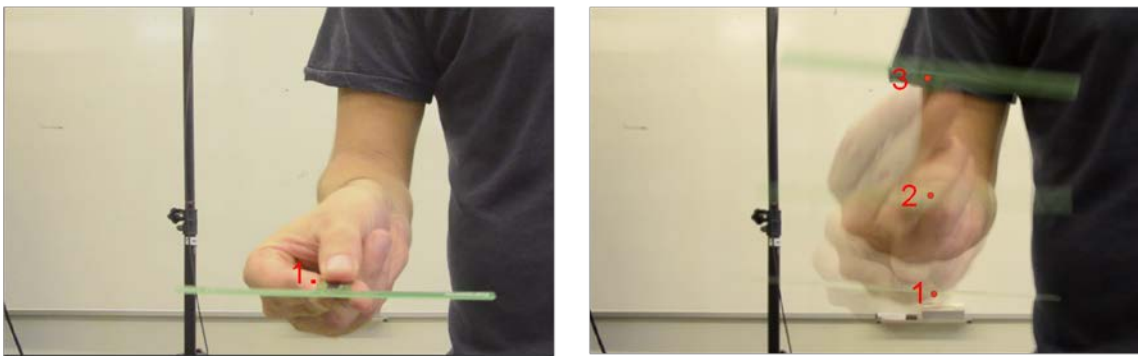


Figura 6.6. Segundo gesto modelo "C"

- En el tercer gesto partiendo como inicio el punto 1 (figura 6.7 izquierda), se ejectua la secuencia 1,2,3,4,5,4,3,2,1 durante 3 repeticiones (figura 6.7 derecha):

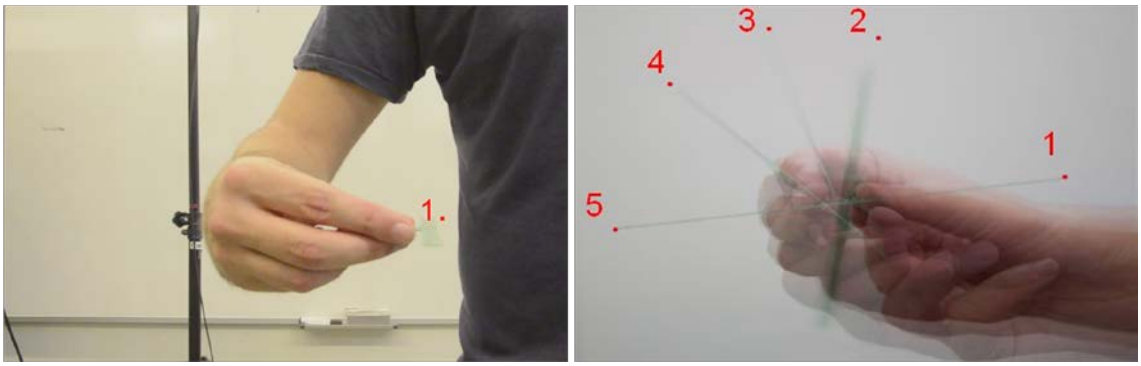


Figura 6.7. Tercer gesto modelo "C"

En el Anexo A.1 se pueden consultar las gráficas del conjunto de pruebas realizadas para este gesto.

6.3 *Scripts* de extracción y representación de datos

Para extraer la información generada por el acelerómetro y contenida en los archivos ".csv" se realizan dos *scripts* en lenguaje R. En el primer *script* llamado *"main.R"* (Ver Anexo B.1) se ejecuta la instancia principal del programa. Se invoca a dos funciones escritas en R, una para la lectura realizada sobre un ".csv" generado por el *tag* de Farsens (ver Anexo B.2), y la otra para la lectura realizada sobre un ".csv" generado por el dispositivo móvil (ver Anexo B.3).

Una vez automatizada la lectura de los datos, se crea una base de datos de muestras formadas por 10 muestras de cada gesto ("W", "R", "C"), 15 generadas por el *tag* de Farsens y 15 generadas por el dispositivo móvil. Se dispone de un conjunto total de 30 muestras. Para cada muestra recogida del total de 30, se realiza una representación gráfica de la aceleración en los 3 ejes del acelerómetro mediante una función en R (Ver anexo B.4).

6.4 Filtrado del gesto

6.4.1 Principios básicos de filtrado: "Los estados"

Antes de efectuar un gesto, el acelerómetro se encuentra en un estado estable, sin movimiento. Durante este periodo, el acelerómetro genera datos muy cercanos a los 0 m/s^2 , pero es una cantidad de información negligente y que se debe extraer puesto que interesa enfocar el análisis de los datos generados por el gesto a realizar. Al tratarse de un acelerómetro hay que tener en cuenta que en el estado estable, que se describía anteriormente, actúa la fuerza de la gravedad, por lo tanto actúa una aceleración g[7].

[7] Es el valor adoptado por el Servicio Internacional de Pesos y Medidas. Representa la aceleración que la tierra ejerce sobre cualquier objeto, con valor de 9.80665 m/s^2

En la siguiente figura se puede ver un movimiento registrado del tipo “W”, con los estados estables (sin efectuar gesto) y los estados en movimiento (efectuando el gesto):

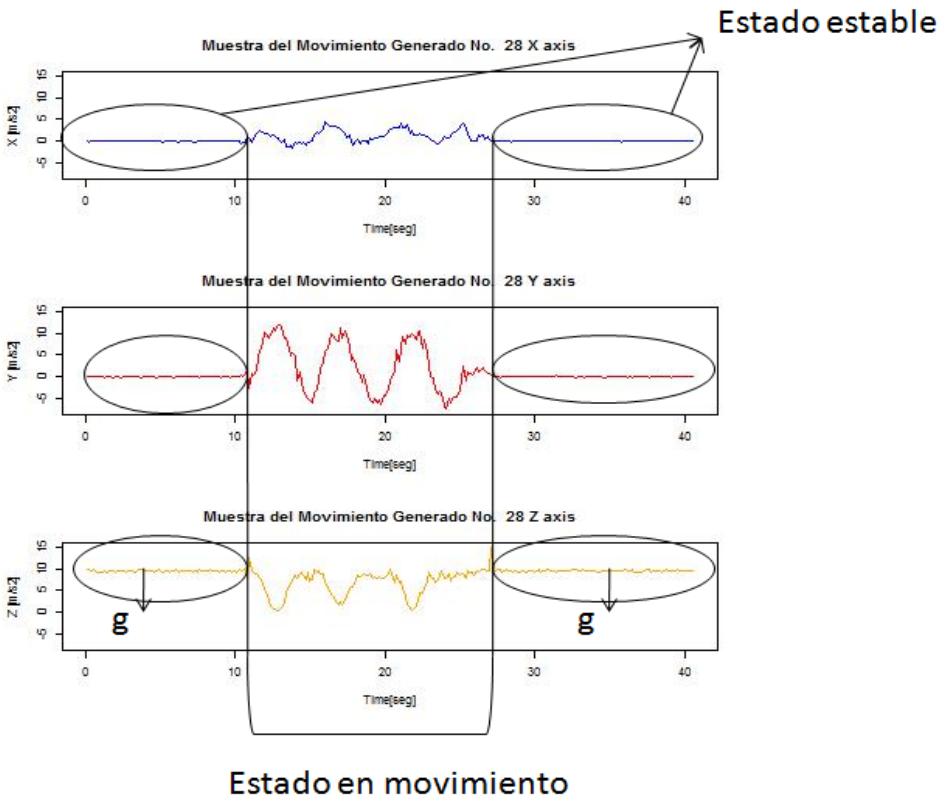


Figura 6.8. Estados del gesto efectuado

El propósito del filtrado es extraer los estados estables del gesto para trabajar con las muestras que pertenecen al gesto generado.

6.4.2 Calibración

Para realizar el filtrado es imprescindible aplicar el método denominado “calibración”. Este método consiste en realizar una muestra de datos del acelerómetro sin efectuar un gesto, tan solo sujetando el acelerómetro. Se trata de medir las pequeñas variaciones producidas por el pulso de la persona que “calibra” al sujetar el acelerómetro. Primero se calibra el *tag* y posteriormente el dispositivo móvil. Estas pequeñas variaciones en la aceleración, que para cada persona pueden resultar diferentes, pertenecen a un estado estable propio de la persona que sujeta (no produce ningún gesto).

Estableciendo un valor máximo y mínimo de la muestra de calibración (considerando los 3 ejes juntos), todos los valores comprendidos entre los valores máximo y mínimo pertenecen al estado estable. Este estado estable contiene las pequeñas variaciones producidas por el pulso de la persona pero en ningún caso se consideran parte del gesto.

6.4.3 Filtrado

El filtrado es la técnica que permite detectar los valores de la aceleración que pertenecen al gesto realizado (estado en movimiento) y extraer aquellos valores de la aceleración en los que no se produce el gesto (estado estable). Esta técnica permite averiguar el instante inicial y final del gesto realizado. En la figura 6.9 se ilustra el funcionamiento de esta técnica.

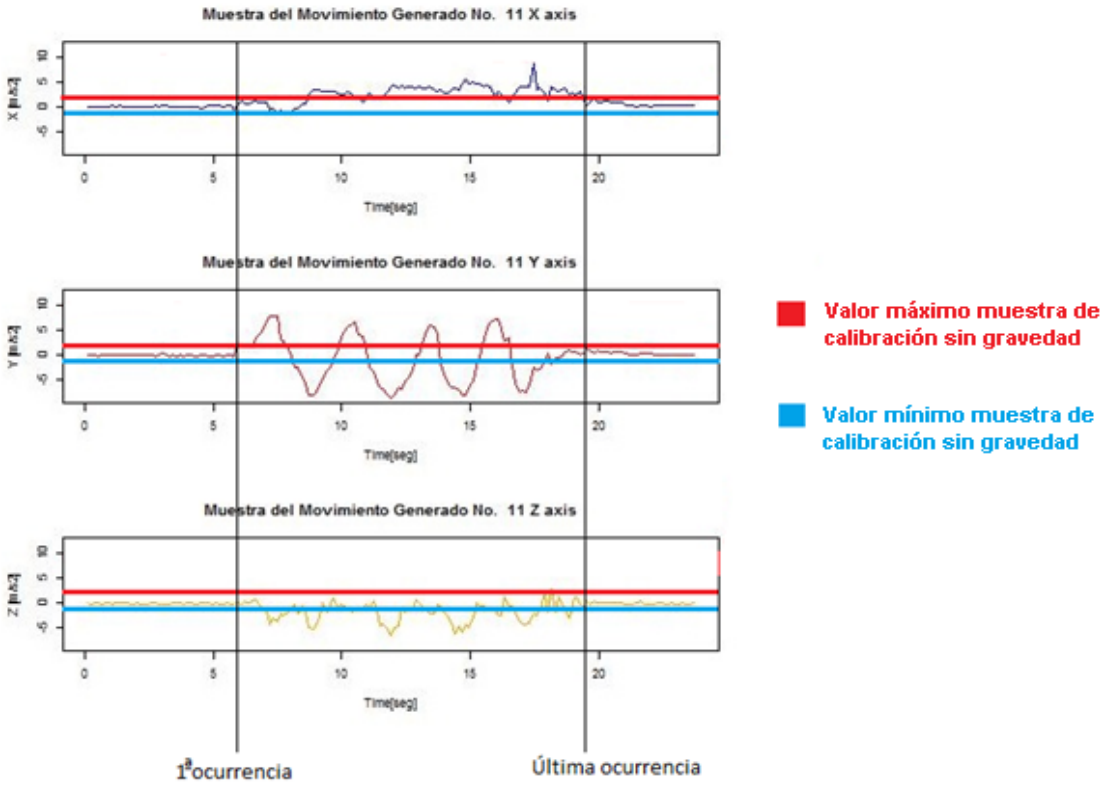


Figura 6.9. Aplicación de los *threshold* sin gravedad sobre el gesto "W"

Los valores máximos y mínimos de la muestra calibrada permiten establecer un *threshold*[8]. Considerando el eje temporal de izquierda a derecha, los valores no considerados como parte del gesto (estado estable) se sitúan en la parte que comprende los valores máximo y mínimo. Estos valores deben ser extraídos ya que no pertenecen al gesto.

Cuando un valor de la aceleración sale de la zona comprendida por el valor máximo y mínimo se considera 1ª ocurrencia, es decir, el instante de tiempo donde se inicia el gesto. A partir de ahí las muestras siguientes se consideran parte del gesto hasta llegar a la última ocurrencia. Del mismo modo la última ocurrencia es el instante de tiempo que pertenece al último valor de la aceleración en situarse fuera de la zona comprendida entre los valores máximo y mínimo (estado en movimiento). A partir de ese instante, el resto de muestras no se consideran parte del gesto, y por tanto deben ser eliminadas. La primera ocurrencia y la última marcan la separación entre el estado estable y el estado en movimiento (ver figura 6.9).

Establecer el rango de valores del estado estable, permite saber qué valores pertenecen al estado en movimiento (rango de valores pertenecientes al gesto). Esto es clave para extraer los datos pertenecientes al gesto y eliminar los valores del estado estable, contemplados como ruido[9]. En la figura 6.10 se puede observar el gesto una vez aplicado el filtro.

[8] Límite o linde.
[9] En el contexto de telecomunicaciones se utiliza para indicar la señal que interfiere de forma no deseada con la señal que se desea transmitir.

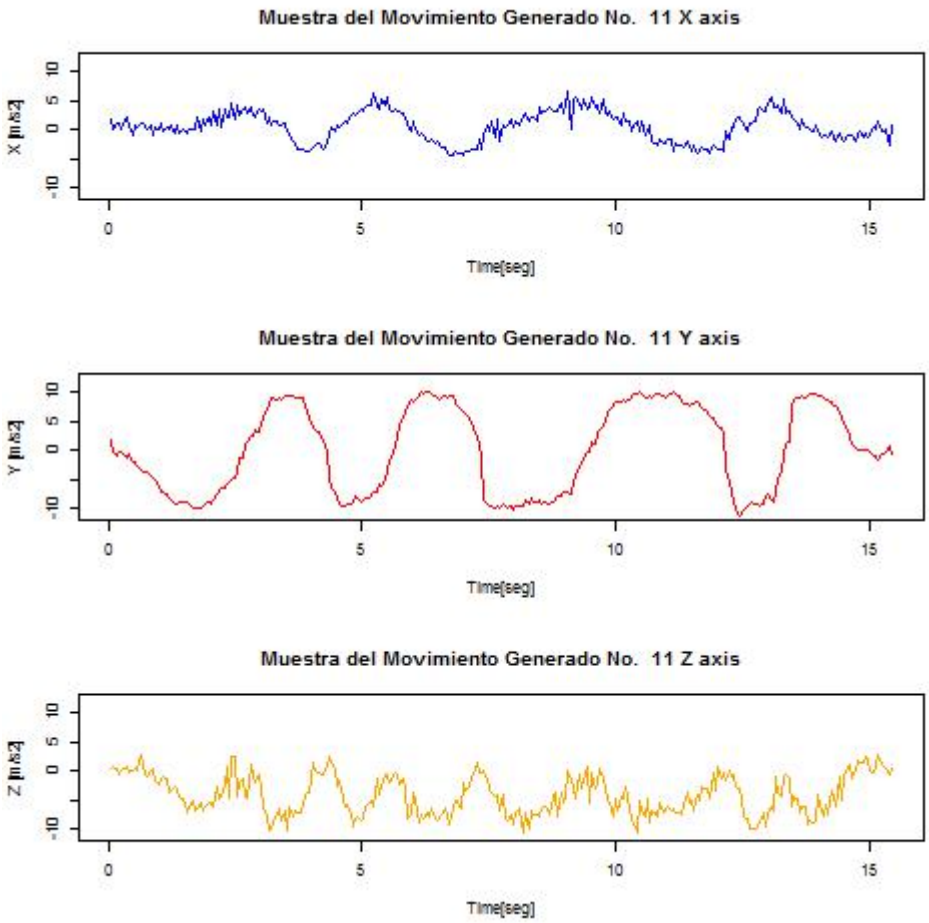


Figura 6.10. Gesto "W" filtrado

6.4.4 Efecto de la gravedad

Se debe tener en cuenta al medir la calibración el efecto de la gravedad. La muestra de calibración incluye el efecto de la gravedad, pero como se quiere medir las variaciones de la aceleración producidas por el pulso de la persona, la gravedad debe ser extraída ya que se trata de una fuerza ejercida por la tierra sobre el objeto.

Cuando se sujeta el *tag* o el dispositivo móvil para calibrar, se permanece en un estado de equilibrio, en el que se aplica una aceleración de la misma magnitud y dirección, pero de sentido contrario a la aceleración de la gravedad. Esta aceleración tiene un valor vectorial[10] de $+9.80665\ \mathrm{m}/s^2$.

[10] Representación geométrica de una magnitud física. Además del módulo, se indica la dirección y el sentido del valor que representa.

En el apartado anterior, al calibrar, se calcula un valor máximo y un valor mínimo del conjunto de valores que conforman la muestra de calibración. Estos dos valores son extraídos de forma conjunta en los 3 ejes dimensionales (x,y,z). En el eje z, mientras se produce la muestra de calibración, los valores de la aceleración oscilan en torno al valor +9.80665 m/s^2, valor de la aceleración que compensa la aceleración de la gravedad. Sin el efecto de la gravedad en el eje z, las variaciones del pulso oscilarían en torno a los 0 m/s^2 , tal y como sucede en los ejes x, y.

Al considerar la gravedad, el valor máximo de la muestra de calibración siempre resulta ser mayor o igual que +9.80665 m/s^2. Al realizar el filtrado se considera "estado en movimiento" todos los valores superiores al valor máximo de la muestra de calibración o todos los valores inferiores al valor mínimo de la muestra de calibración. Si el valor máximo contiene el efecto de la gravedad al fijar el valor máximo con un valor cercano a +9.80665 m/s^2, la zona que comprende el máximo y el mínimo es mayor y esto supone menos precisión para detectar la primera ocurrencia, ya que se pueden omitir valores pertenecientes al gesto. Esta circunstancia se ilustra en la figura 6.11.

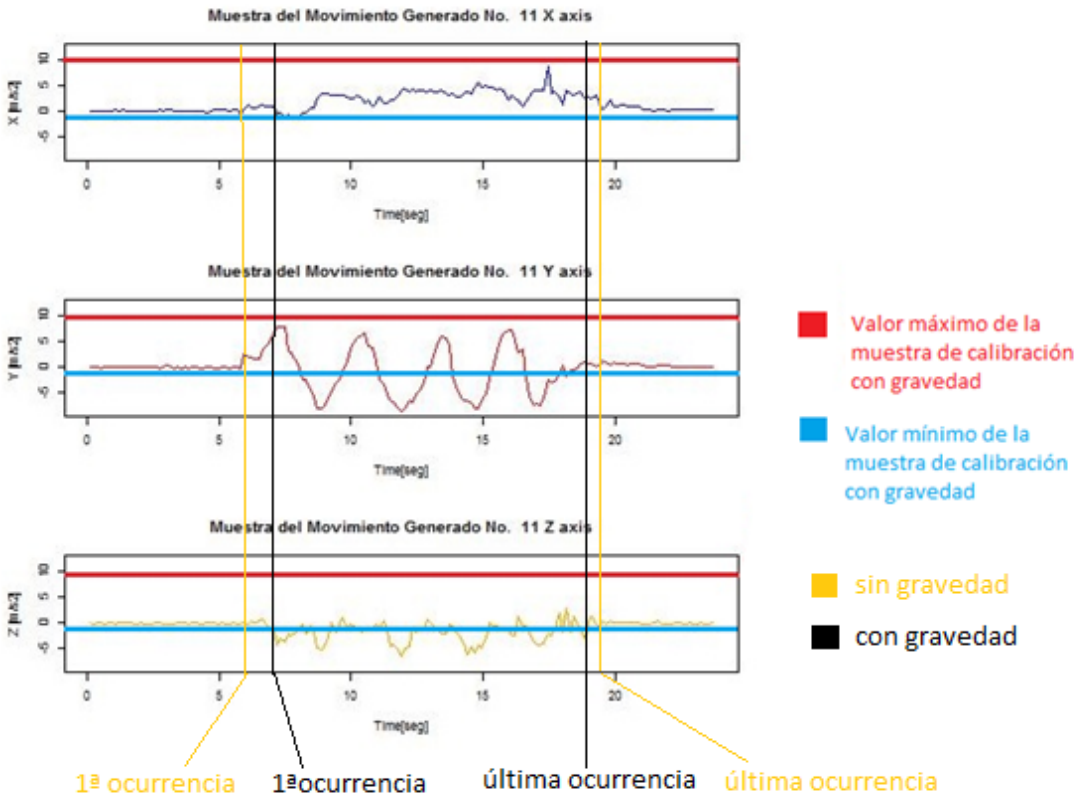


Figura 6.11. Aplicación de los *threshold* con gravedad sobre el gesto "W"

Si se comparan las gráficas contenidas en la Figura 6.9 y 6.11, se obtienen los siguientes rangos de valores para el estado estable:

- Con gravedad :
  - Valor máximo = 10.33 m/s^2 , Valor mínimo = -0.933 m/s^2
- Sin gravedad :
  - Valor máximo = 1.99 m/s^2 , Valor mínimo = -0.933 m/s^2

6.4.5 *Scripts* y resultados

Desde el programa principal "*main_TFG.R*" (ver Anexo B.1) se carga la muestra de calibración en formato ".csv". A continuación se llama a la función "calibra" (ver Anexo B.5) para extraer los valores máximos y mínimos de la muestra de calibración para establecer un *threshold*. Finalmente se realiza una llamada a la función "filtra" pasando una muestra a filtrar y los valores del *threshold* (ver Anexo B.6).

Los resultados de esta operación se pueden comprobar visualizando la gráfica del gesto a tratar (ver Figura 6.12) y la gráfica del gesto filtrado (ver figura 6.13).

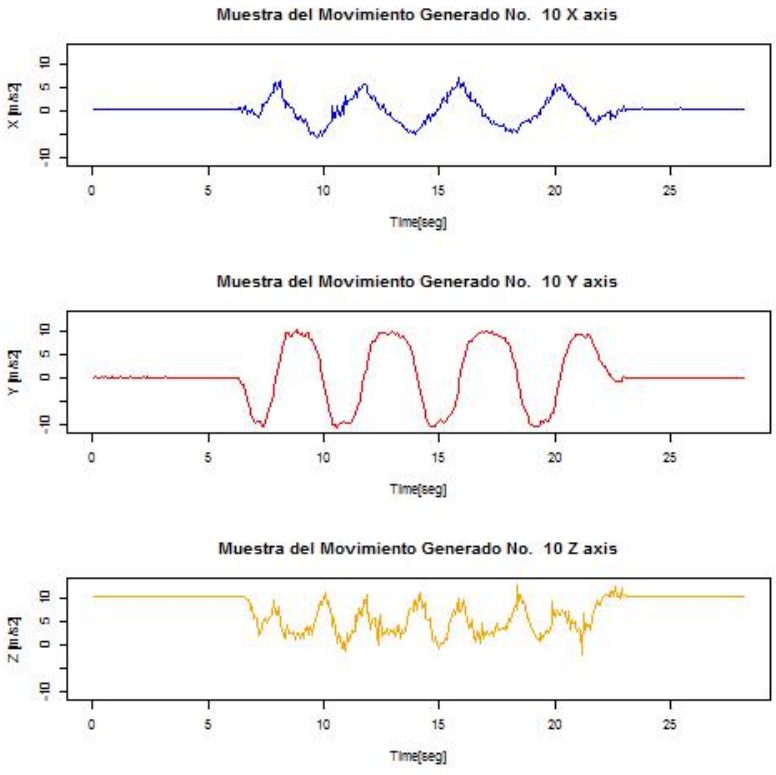


Figura 6.12. Muestra modelo "C" sin filtrar

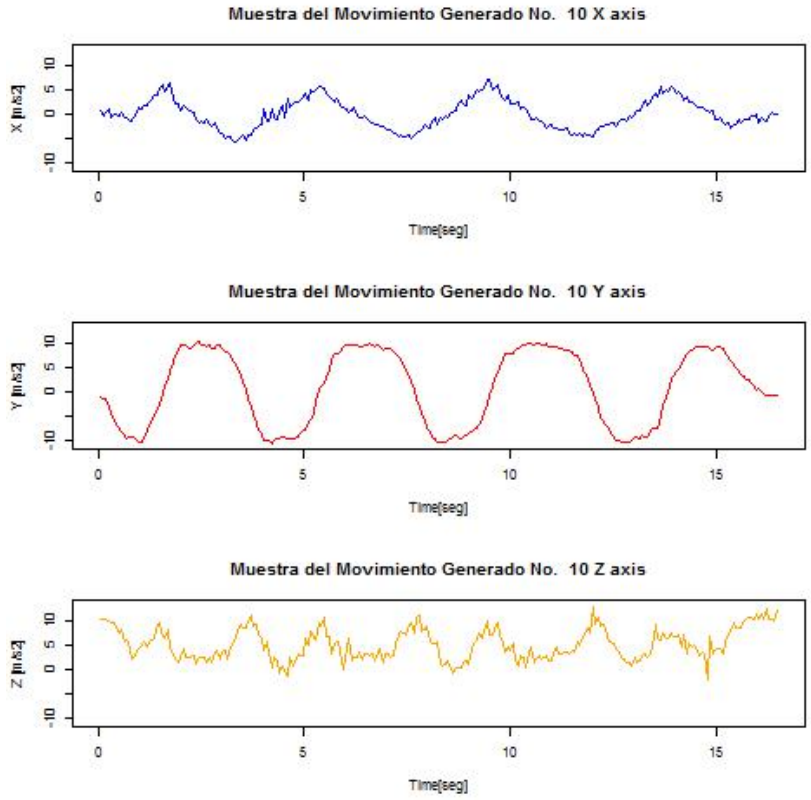


Figura 6.13. Muestra modelo "C" filtrada

Los resultados obtenidos en las gráficas anteriores reflejan los resultados del filtrado. Se ha conseguido extraer el "estado del movimiento", que corresponde al conjunto de datos generados durante la realización del gesto, en este caso un gesto del modelo "C".

CAPÍTULO 7

CARACTERIZACIÓN DE LOS DATOS

7.1 Introducción

En el campo de la computación ubicua, dispositivos como por ejemplo los *smartphones*, han dado lugar a servicios *context-aware*[11] para responder ante actividades o contextos de la vida cuotidiana de los usuarios. Consultar la meteorología y la temperatura de la localidad en la que uno se encuentra o bien las rutas de transporte público que llevan desde una ubicación hasta un domicilio y el tiempo en hacerlo, son algunos ejemplos de este tipo de servicio. El denominador común de este tipo de servicios es la habilidad de detectar y extraer información física en el contexto del usuario. En los dispositivos móviles, los acelerómetros pueden informar de la posición del usuario a tiempo real y, combinado con una posición GPS, se puede saber si está caminando por la calle o está corriendo. Recopilando toda esta información durante un cierto periodo de tiempo se puede, incluso, determinar los hábitos diarios o las tendencias de una persona.

Se centrará en determinar cómo se van a tratar los datos generados por el acelerómetro para poder caracterizar los movimientos generados y establecer parámetros entre las diferentes muestras recogidas.

[11] *Context-aware*: Se refiere a los sistemas móviles que adaptan su comportamiento en función de lo que perciben del entorno físico.

7.2 Evaluación de las características

Un sistema de reconocimiento de patrones nunca realiza la clasificación de los datos usando los datos originales que adquiere del sensor (29). Para poder realizar una clasificación y una decisión sobre el conjunto de datos, la selección de las características o parámetros asociados a los gestos resulta clave y puede ser específica para cada caso de estudio. Por ejemplo, en un estudio para determinar si una persona está corriendo o bien caminando (30), los datos recopilados mostraran ciertas características de periodicidad y de velocidad, puesto que los gestos de caminar y correr muestran esas naturalezas. Existen muchos artículos que estudian la naturaleza de los datos generados por los acelerómetros, con el fin de extraer información relacionada con la actividad que se están llevando a cabo. Un buen ejemplo es el artículo *Preprocessing Techniques for Context Recognition from Accelerometer Data* (31).

7.3 *Dynamic Time Warping* (*DTW*)

Se trata de un algoritmo que mide el grado de similitud entre dos secuencias de datos que pueden diferir en el tiempo y en el número de elementos. El algoritmo DTW (32) alinea dos secuencias de datos de diferente longitud utilizando programación recursiva. Alineando las dos series de datos generados por el acelerómetro, se obtiene un valor de distancia que resulta ser el grado de similitud entre ellas. Por lo tanto, ese grado de similitud es la característica para clasificar y viene definida por el *Dynamic Time Warping*.

Esta métrica resulta muy útil para el objeto de estudio puesto que trata de reconocer gestos con dos dispositivos, *tag* farsens y móvil, donde los gestos pueden ser realizados en instantes de tiempo diferente. A esto se añade que las tasas de salida de datos pueden variar, lo que implica que para un mismo movimiento tomado por los dos dispositivos (*tag* farsens y móvil), el número de muestras es diferente. El *Dynamic Time Warping* no tiene en cuenta este efecto (33).

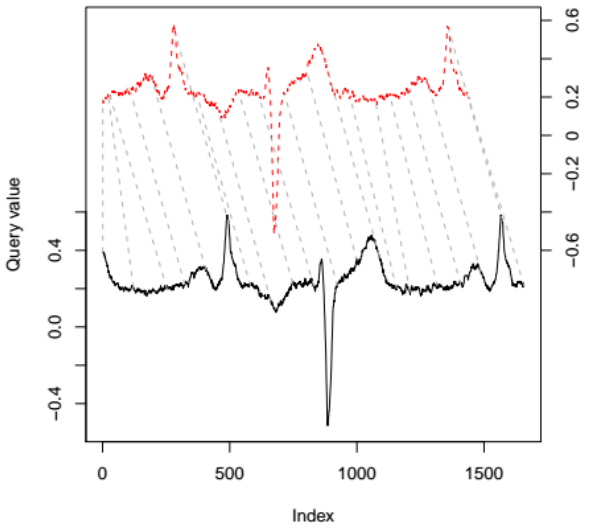


Figura 7.1. Alineamiento temporal de dos series mediante DTW (34)

7.3.1 Fundamento matemático del algoritmo DTW

Si se consideran dos secuencias temporales:

$$A = \{x1,x2,x3\ldots..xi...xn\}$$

$$B = \{y1,y2,y3\ldots.yj.....ym\}$$

Se pueden representar en una cuadrícula considerando el extremo izquierdo, el inicio temporal, y el extremo derecho, el final temporal.

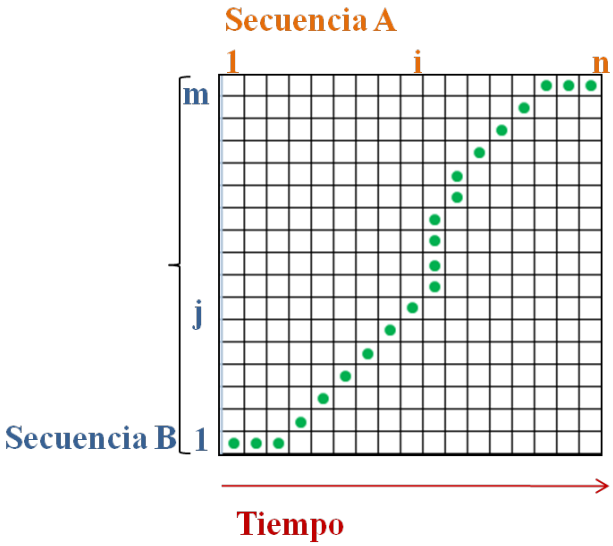


Figura 7.2. Camino de alineamiento óptimo o *warping path*

Se considera una función positiva de similitud:

$$\mathrm{d(i,j)} = \mathrm{f}(x_i y_i) \geq 0$$

La función d es la matriz de distancia cruzada entre los vectores x, y. Para cada elemento de $x_i y_i$ calcula la diferencia y cuando ésta es muy pequeña, significa que el grado de similitud es muy alto. Normalmente se suele usar la distancia euclidiana[12] para el cálculo. Esta matriz se conoce como la matriz de costes y el algoritmo trata de buscar el camino óptimo que recorre las zonas con menor coste, conocido como camino óptimo o *warping path* (ver figura 7.2).

El núcleo de esta técnica se encuentra en la curva de deformado, denominada *warping curve o warping function*:

$$\phi(k) = (\phi_x(k), \phi_y(k))$$

$$k = 1 \dots T$$

$$\phi_x(k) \in \{1 \dots N\}$$

$$\phi_y(k) \in \{1 \dots M\}$$

Las funciones ϕ_x ϕ_y reasignan los valores del tiempo de x, y respectivamente. Dada la función ϕ se calcula la media acumulada de la distorsión entre las dos señales x, y desfasadas en el tiempo.

$$d_\phi(X,Y) = \sum_k^T d((\phi_x(k), \phi_y(k))) \cdot m_\phi / M_\phi$$

El parámetro m_ϕ es un coeficiente que marca el peso del paso y el que permitirá saber el camino óptimo (con menor peso asignado). M_ϕ es la constante de normalización que permite que todas las distorsiones acumuladas se puedan comparar a lo largo de los diferentes caminos óptimos.

[12] Es la distancia mínima entre dos puntos situados en un plano. Sean dos puntos definidos como ($x1 - y1$, *(x2−y2)* las distancia euclidiana *d=x1−y12+(x2−y2)2*

Se imponen una serie de restricciones sobre ϕ; es decir, sobre los pasos a seguir a lo largo del *warping path*:

1. **Condición de límite:**

$$\phi_x(1) = \phi_y(1) = 1$$

$$\phi_x(T) = N, \qquad \phi_y(T) = M$$

El primer elemento y el último deben corresponder al primer y último instante de tiempo de las secuencias de entrada.

2. **Condición de monotonía:**

$$\phi_x(k+1) \geq \phi_x(k)$$
$$\phi_y(k+1) \geq \phi_y(k)$$

Esta condición asegura que las muestras que conforman el camino más óptimo deben sucederse en el tiempo.

3. **Condición de salto temporal:**

$$|\phi_x(k+1) - \phi_x(k)| \leq 1$$
$$|\phi_y(k+1) - \phi_y(k)| \leq 1$$

La diferencia temporal entre un elemento y el siguiente no puede ser mayor a 1. Por tanto, fija el salto temporal en el máximo de 1.

Comentadas las restricciones anteriores, la idea del DTW es encontrar el *warping path* ϕ tal que:

$$DTW(X,Y) = \min_{\phi} d_{\phi}(X,Y)$$

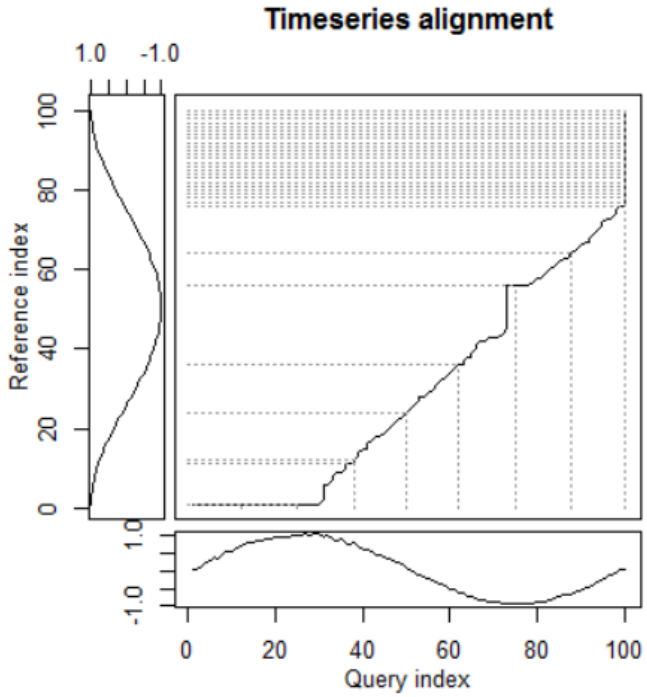


Figura 7.3. Gráfica "3 *plot way*" de dos series a alinear (34)

La salida que proporciona DTW resulta muy interesante ya que es una métrica que caracteriza un par de secuencias en función del grado de similitud. En base a ese grado de similitud, se utilizan los algoritmos de *Machine learning* para aprender cómo las diferentes muestras recogidas se parecen entre sí. Ante una muestra a clasificar, primero se calculará cómo se asemeja con el conjunto de muestras ya clasificadas, y en base a esa similitud, se podrá saber de qué muestra se trata. Cómo aprender es lo que se plantea en el capítulo 8.

7.3.2 *Scripts* en R para el cálculo DTW

Después de filtrar las 30 muestras que conforman el conjunto de pruebas, se ejecuta la función "*similitud_DTW*" (ver Anexo B.7), pasando como parámetro cada par de muestras de las 30x30 que conforman el conjunto de pruebas. El resultado da lugar a una matriz de 30 x 30 con 900 cálculos de la distancia DTW, o en este caso de estudio, 900 "características de similitud".

Hay que tener en cuenta que el cálculo del algoritmo DTW se realiza para cada eje del plano 3dimensional, lo que implica que una muestra se compara con las 30 muestras que forman el conjunto y, a su vez, por cada eje. Esto implica que 1 muestra se compare 90 veces con el resto de muestras y se calculen un total de 30 x 90 = 2700 distancias DTW (ver Anexo C Subcarpeta:“tables”, Archivo:“dtw_xyz.csv”).

Para realizar el cálculo de la similitud DTW se ha recurrido al paquete DTW de lenguaje R. Este paquete invoca a la función DTW que realiza el cálculo del algoritmo y devuelve un objeto DTW. Sobre este objeto, se extrae el valor de la distancia DTW normalizada por la longitud del camino óptimo o *warping path*.

El conjunto de datos caracterizados (2700 distancias DTW) correspondientes a las muestras son la entrada al algoritmo *Machine Learning*.

CAPÍTULO 8

CLASIFICACIÓN Y DECISIÓN

8.1 *Machine Learning*

8.1.1 Introducción a *Machine Learning*

Machine learning (35) es la herramienta encargada de analizar de forma automatizada un conjunto de datos. Se basa en un conjunto de métodos que automáticamente detectan patrones o comportamientos en los datos a analizar. Este conocimiento de los patrones o conductas de los datos se utiliza para predecir nuevos datos o tomar decisiones sobre los datos.

En este caso de estudio se quieren analizar muestras de datos generados por los acelerómetros del *tag* de Farsens y el dispositivo móvil para poder predecir si el gesto producido se trata de un conjunto de gestos conocidos y, más concretamente, averiguar de qué gesto se trata.

8.1.2 Tipo de *Machine Learning*

Existen dos tipologías o modelos generales de métodos *Machine Learning* (36):

- *Unsupervised Learning*
- *Supervised Learning*

La tipología utilizada en el caso de estudio es:

Supervised learning

Los métodos de aprendizaje supervisado se basan en la idea de aprender a partir de la experimentación. Estos métodos emplean un conjunto de datos de muestra (*training set)* y de un conjunto de datos de prueba (*test set*). El objetivo del método o algoritmo es aprender a partir de muestras ya clasificadas del *training set* e identificar las muestras del *test set* que todavía no están clasificadas con la máxima precisión.

En el aprendizaje supervisado, el objetivo es aprender como las muestras del *training set* se corresponden con lo que se quiere clasificar. Se trata de proporcionar un conjunto de muestras de entrada x_i , clasificarlas o asignarles una etiqueta y_i y crear un conjunto de muestras etiquetadas (clasificadas) $T = \{(x_i, y_i)\}_{i=1}^{N}$. Este conjunto de pruebas clasificadas es lo que se denomina *training set*.

En este caso de estudio, el vector x_i contiene 90 distancias normalizadas del DTW $DTW(x_i)$ = 30 * 3 ejes = 90 distancias normalizadas calculadas por el DTW, como se ha visto en el apartado anterior.

A continuación, para constituir el *training set* se necesita saber del total de 30 muestras capturadas y caracterizadas, qué muestras x_i pertenecen a los gestos “C”, “W” y “R”, y asignar la correspondiente etiqueta y_i.

Ya se tiene el total de muestras x_i y sus etiquetas asignadas y_i. Se va a formular el conjunto de muestras de entrenamiento *training set*:

$$Tx = \left(\{(xi, yi)\}_{i=1}^{30}\right), Ty = \left(\{(xi, yi)\}_{i=1}^{30}\right), Tz = \left(\{(xi, yi)\}_{i=1}^{30}\right)$$

$$Txyz = \left(\{(xi, yj)\}_{i=1\ \ j=1}^{30\ \ 30}\right),$$

A continuación, se debe averiguar sobre una muestra de prueba x'_i , su respectiva etiqueta y'_i , basándose en el conjunto Txyz. Esta tarea de decidir y clasificar pertenece al método de *machine learning* elegido.

8.2 *K-Nearest Neighbor*

Es uno de los métodos que se encuentra entre los 10 mejores algoritmos de minería de datos (37).

Este algoritmo se encarga de analizar todos los datos que conforman el conjunto de muestras (*training set*) y busca un subconjunto de k muestras (*neighborhood*) que se asemejen lo máximo a la muestra a comprobar (*test*). Esta semejanza normalmente se calcula mediante la distancia euclidiana. Cada una de las muestras que conforman el conjunto k (*neighborhood*) pertenecen a una etiqueta o clase (*class*). La etiqueta (*class*) que más predomine de ese subconjunto (*neighborhood*) es la que se le asigna a la muestra a probar (*test*). El algoritmo se fundamenta en:

- Conjunto de pruebas (*training set*) etiquetadas con su clase (*class*).
- Cálculo de la distancias de la muestra a comprobar (*test*) y el conjunto de pruebas.
- El número de k vecinos para determinar si pertenece a una clase u otra.

En la figura siguiente se ilustra el funcionamiento del *K-Nearest Neighbor*. El parámetro k=4 fija el subconjunto (*neighborhood*) en 4 muestras. Una nueva muestra a clasificar (*test*) llega al espacio definido por el conjunto de muestras (*training set*). La elección de la nueva clase se fundamenta en el recuento de las clases que conforman el subconjunto k=4 (*neighborhood*). La selección de la clase queda bien demostrada en la figura 8.1.

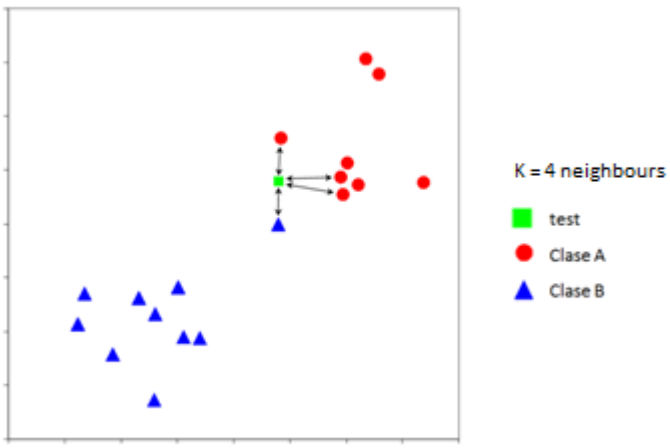


Figura 8.1. Clasificación del KNN en el espacio

Formulación matemática

T : Conjunto de muestras (*training set*).

T_z : Subconjunto de k muestras (*neighborhood*).

$x^{'}$: Vector a comprobar.

$y^{'}$: Clase del vector a comprobar.

v : Clase.

y_i: Clase del elemento i-ésimo del "vecindario".

$I(\cdot)$: Función que devuelve 1 o 0 si la condición es cierta o falsa respectivamente.

$$\text{KNN } (Txyz, z = x^{'}, y')$$

Entradas:

$$\text{T}$$

$$z = x^{'}, y^{'}$$

Proceso:

$$d(x^{'}, x) = \text{ z } \forall\, x, y \in T$$

$$T_z \subseteq T$$

Salida:

$$y^{'} = argmax \sum_{(x_i, y_i) \in T_z} I(v = y_i)$$

En $y^{'}$ se resuelve finalmente la clase o etiqueta a la cual pertenece la muestra $x^{'}$ o muestra de *test.*

8.3 *Scripts* en R del *K-Nearest Neighbor*

Para realizar la clasificación de la muestra de pruebas, se ha recurrido a la función "KNN" (38) elaborada en R (ver en Anexo B.8). Esta función se encarga de ejecutar el método *K-Nearest-Neighbor*, pasándole como parámetros:

- El conjunto *training set* de 30 muestras con sus 90 características.
- Muestra a comprobar o *test* con sus 90 características calculadas.
- Conjunto de etiquetas o *class*:
    - 10 primeras muestras → 10 * "C"
    - 10 muestras siguientes → 10 * "R"
    - 10 muestras últimas → 10 * "W"

Esta función devuelve el resultado de la etiqueta a la cual pertenece, la muestra de prueba y la probabilidad con la que fue clasificada.

CAPÍTULO 9

EVALUACIÓN DE LOS RESULTADOS

9.1 *K-Fold Cross Validation*

Los métodos de validación cruzada o *K-fold cross validation* se utilizan para evaluar los resultados analíticos entre un conjunto de datos *training* y un conjunto de datos *test*. Este método sirve para medir la calidad de la técnica de clasificación comentada anteriormente. La técnica *k-Nearest Neighbour* utiliza el parámetro k[13] (número de vecinos). Mediante la evaluación cruzada se puede saber qué valor de k es el que mejor resultados ofrece.

En la validación cruzada de K[14] iteraciones (*K-Fold Cross Validation*), los datos obtenidos (muestras) se dividen en K subconjuntos. Un subconjunto se utiliza como prueba (*test*) y el resto de subconjuntos K-1, se utilizan como datos de entrenamiento (*training set*). Este proceso se repite durante K iteraciones, con los subconjuntos *test* y *training set* seleccionados por el método de *K-Fold Cross Validation*.

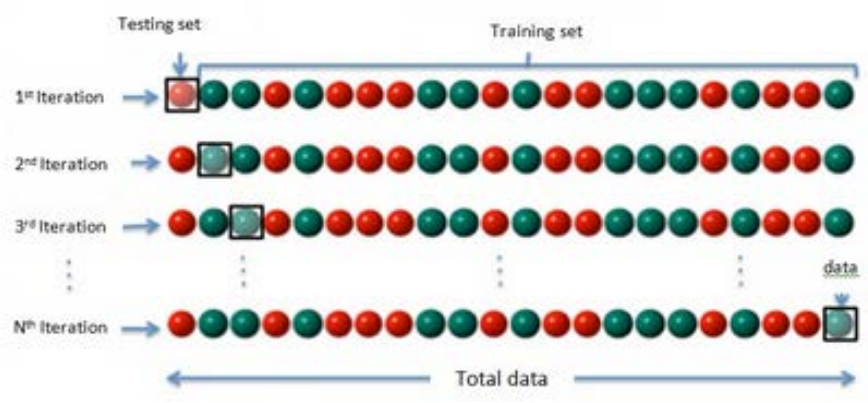


Figura 9.1. Esquema de evaluación de *K Fold Cross Validation* (39)

[13] Parámetro k del algoritmo *k-Nearest Neighbour* que indica el número de vecinos considerados para la clasificación de la muestra *test*.
[14] Parámetro K del algoritmo *K-Fold Cross Validation* que indica el número de iteraciones y el número de subconjuntos en que se debe dividir el conjunto de datos.

9.2 Aplicación del *K-Fold Cross Validation*

Para aplicar esta herramienta de evaluación en este caso de estudio, se va a utilizar el conjunto de datos de prueba o *training set* calculado de 30 muestras. Estas 30 muestras están distribuidas:

1. 10 Muestras "C" :
   a. [1,5] = *tag* Farsens
   b. [6,10] = dispositivo Móvil
2. 10 Muestras "R" :
   a. [11,15] = *tag* Farsens
   b. [16,20] = dispositivo Móvil
3. 10 Muestras "W" :
   a. [21,25] = *tag* Farsens
   b. [26,30] = dispositivo Móvil

1ª Iteración:

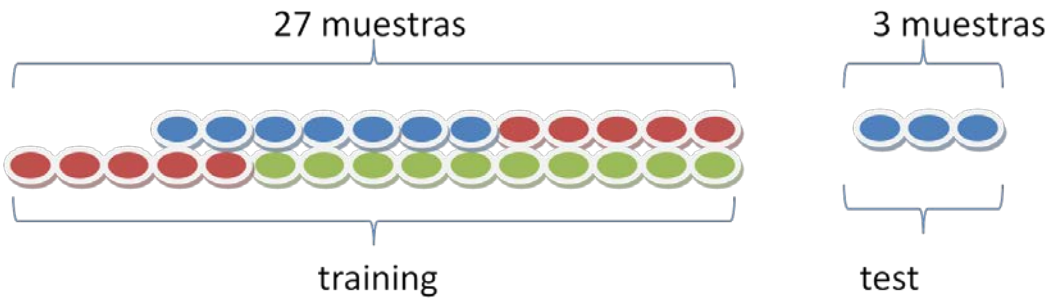


Figura 9.2. 1ª Iteración de la técnica *10-Fold Cross Validation*

El resto de iteraciones del algoritmo se pueden visualizar en el documento anexo (Ver Anexo D).

9.3 *Scripts* en R de *K-Fold Cross Validation*

Para calcular la técnica *K-Kold Cross Validation*, se implementa una función en R llamada "KNN" (ver Anexo B.8). Además de clasificar una muestra *test*, realiza la evaluación del algoritmo KNN mediante la técnica *K-fold Cross Validation*. Para efectuar esta tarea la función se ayuda de la librería *"class"* (ver anexo C Subcarpeta: "librerias", Subcarpeta II: "*class*") con el método *"knn.cv"* (ver Anexo B.8).

9.4 Resultados *K-Fold Cross Validation*

- *True Positives* (TP): Muestras con etiqueta "C", "R", "W" que se clasifican correctamente con sus valores respectivos "C", "R", "W".
- *True Negative* (TN): Muestras que no pertenecen al valor de "C", "R", "W" y no se clasifican en sus valores respectivos "C", "R", "W".
- *False Positive* (FP): Muestras que no pertenecen al valor "C", "R", "W" y se clasifican respectivamente como "C", "R", "W".
- *False Negative* (FN): Muestras que pertenecen a sus clases "C", "R", "W" y no se clasifican respectivamente como "C", "R", "W".

TP		TN		FP		FN	
Muestra	Clasifica	Muestra	Clasifica	Muestra	Clasifica	Muestra	Clasifica
"C"	"C"	No "C"	"R" o "W"	No "C"	"C"	"C"	"R" o "W"
"R"	"R"	No "R"	"C" o "W"	No "R"	"R"	"R"	"C" o "W"
"W"	"W"	No "W"	"C" o "R"	No "W"	"W"	"W"	"C" o "R"

Tabla 9.1. Esquema que define TP, TN, FP y FN

k	10	
“C”		
	TP = 10	FP = 8
	FN = 0	TN = 12
“R”		
	TP = 3	FP = 0
	FN = 7	TN= 20
“W”		
	TP = 9	FP = 0
	FN = 1	TN = 20

Tabla 9.2. Resultados evaluación *30-Fold Cross Validation* (Ver Figura 9.3)

Precision: Evalúa que las muestras que pertenecen a sus clases “C” o “W” o “R” , se clasifiquen como “C” o “W” o “R”, aunque hayan muestras que no se clasifiquen como “C” o “W” o “R”.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Evalúa que las muestras que pertenecen a sus clases “C” o “W” o “R” , se clasifiquen como “C” o “W” o “R”, aunque hayan muestras que se clasifiquen como “C” o “W” o “R” sin serlo.

$$Recall = \frac{TP}{TP + FN}$$

F-score: Es una medida de precisión que pondera las dos métricas *Precision* y *Recall* en un único resultado.

$$F - Score = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Classification Accuracy (ACC) = Calcula la proporción de muestras clasificadas que pertenecen a “C” , “R”, “W” y realmente lo son, y las que no pertenecen a “C”, “R”, “W” no las clasifica como tal. Es la métrica encargada de medir el porcentaje de muestras de *test* que son clasificadas correctamente:

$$ACC = \frac{tp + tn}{tp + tn + fp + fn}$$

```
> knn_cv
[[1]]
 [1] C C C C C C C C C C C C C C C R C R R C W W W W W W W W W W
attr(,"prob")
 [1] 0.7 0.6 0.5 0.6 0.8 0.6 0.8 0.8 0.9 0.8 0.7 0.7 0.7 0.6 0.6 0.5 0.5 0.5 0.5 0.5 0.5
[22] 0.7 0.7 0.8 0.9 0.7 0.7 0.7 0.7 0.7
Levels: C R W
```

Figura 9.3. Resultados evaluación *10-Fold Cross Validation* de la función “KNN” (Ver Anexo B.8)

Para evaluar los resultados de *K-Fold Cross Validation* con diferentes valores de k vecinos (k=10, k=8, k=5, k=2) y para diferentes valores de K iteraciones (K=10, K=30) consultar documentación anexa (Ver Anexos F).

9.5 Puesta a prueba

En el punto anterior, se ha evaluado el rendimiento del método de aprendizaje con las métricas utilizadas en el campo de *pattern recognition*. Sin embargo, el propósito de estudio es poder reconocer un gesto realizado y que no pertenece a nuestro conjunto de gestos entrenados o *training set.* Se va a comprobar que el programa en R *“main_TFG”* (ver anexo B.1) es capaz de reconocer un gesto.

El programa lee un gesto producido con el *tag* Farsens y lo clasifica. A continuación se realiza otro gesto independiente del primero, pero esta vez efectuado con el dispositivo móvil. Este segundo gesto debe ser del mismo tipo que el primer gesto hecho por el prototipo de Farsens.

La idea es que un gesto vaya asociado con una persona o usuario. Al realizar la primera autenticación, el sistema debe permitir realizar una segunda autenticación desde el dispositivo móvil. Por lo tanto no se trata de comparar el gesto del *tag* Farsens con el

gesto del dispositivo móvil, ya que han sido realizados en instantes de tiempo diferentes y con dispositivos diferentes. El sistema debe ser capaz de entrenar los gestos realizados por un usuario con ambos dispositivos (*tag* y móvil), para que cuando realice una autenticación (gesto entrenado que no pertenece al *training set*) con cualquier dispositivo pueda reconocer a dicho usuario.

9.5.1 Prueba 1 k =10 gesto "W"

Tag Farsens

Se realiza un movimiento del tipo "W" (ver punto 6.2.1) producido con el *tag* Farsens. Al realizar un gesto de *test*, el ".csv" generado se introduce en la carpeta "test/Farsens" del entorno de trabajo (ver Anexo C). El programa se encarga de ir al directorio para buscar la muestra de *test*. Carga el movimiento de *test* de la figura 9.4:

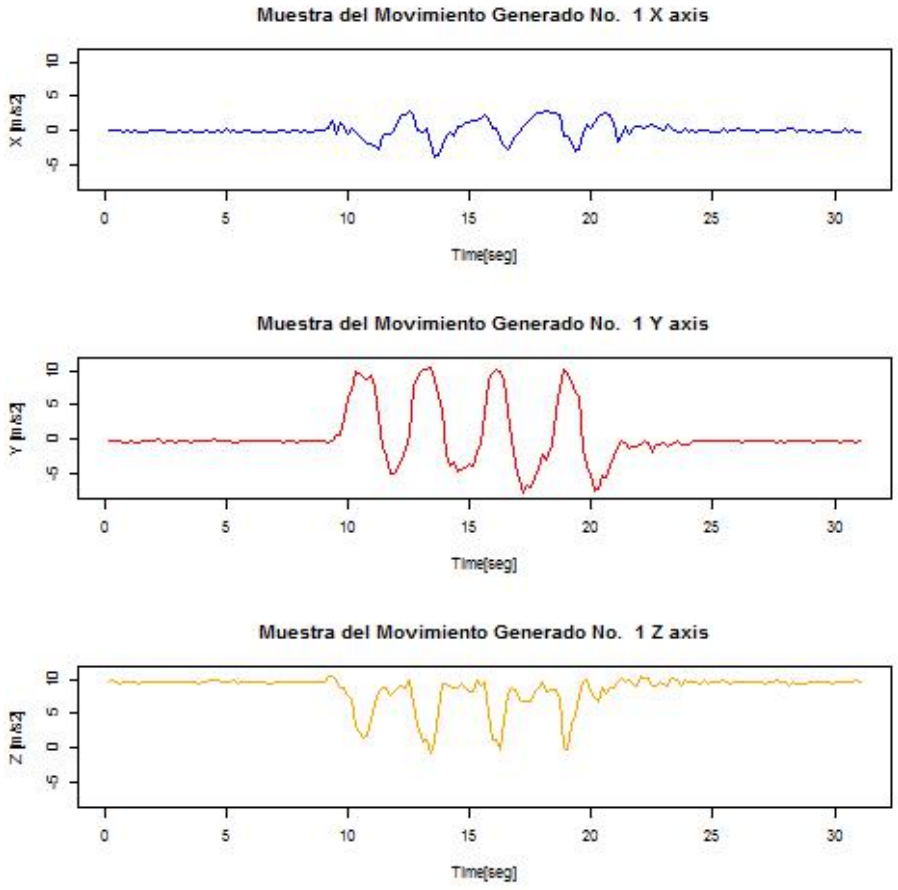
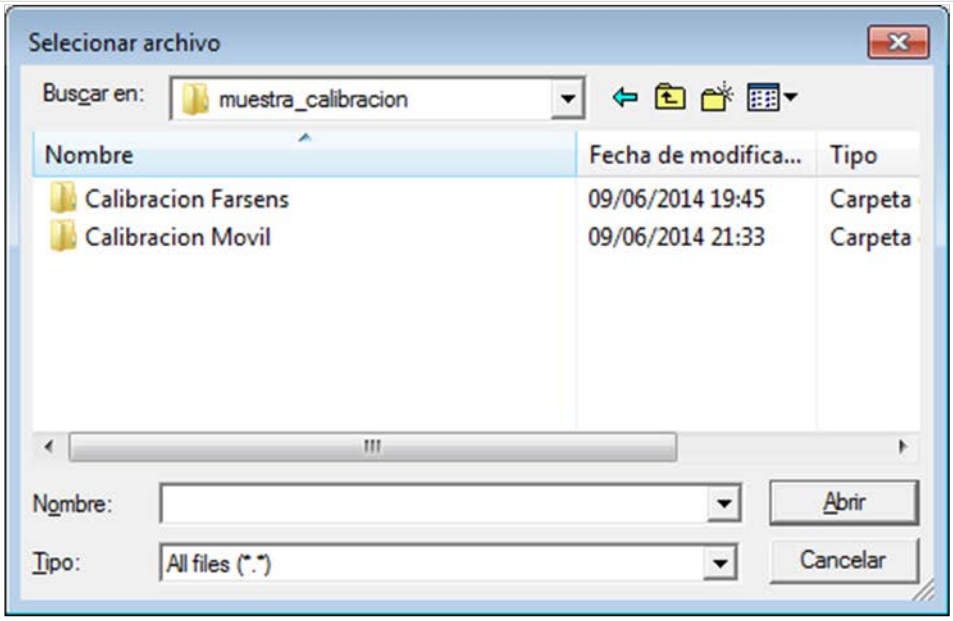


Figura 9.4. Gráfica de la aceleración del gesto *test* del *tag* Farsens

A continuación se solicita una muestra de calibración del *tag* Farsens:

```
> ##############################################################################################################################
#######
> #### Lectura  .... [TRUNCATED]
[1] "Introduzca una muestra calibrada Farsens"

> path<-file.choose()
```



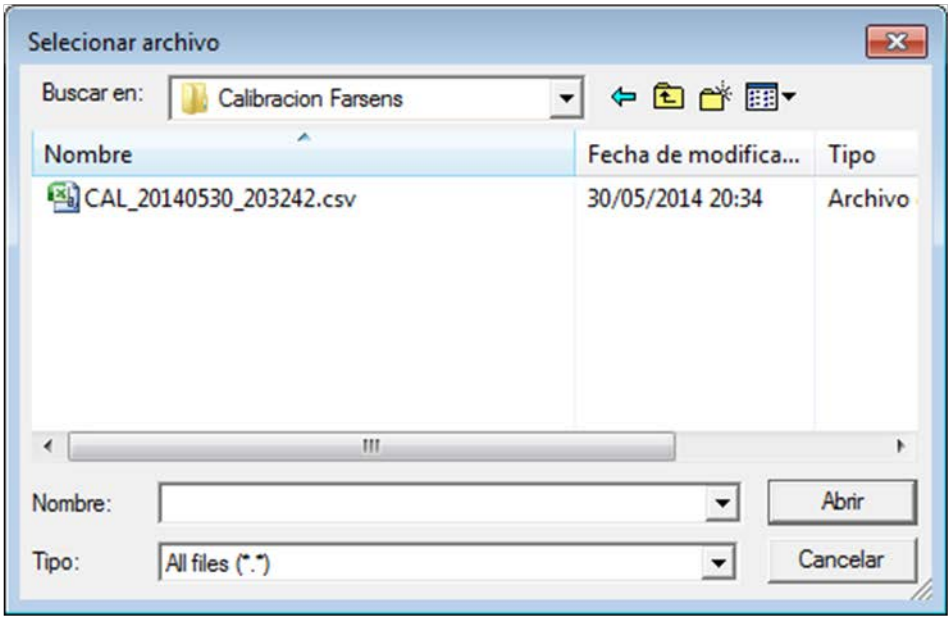


Figura 9.5. Selección por consola de la muestra a calibrar del *tag* Farsens

Se visualiza la muestra *test* filtrada (ver figura 9.6)

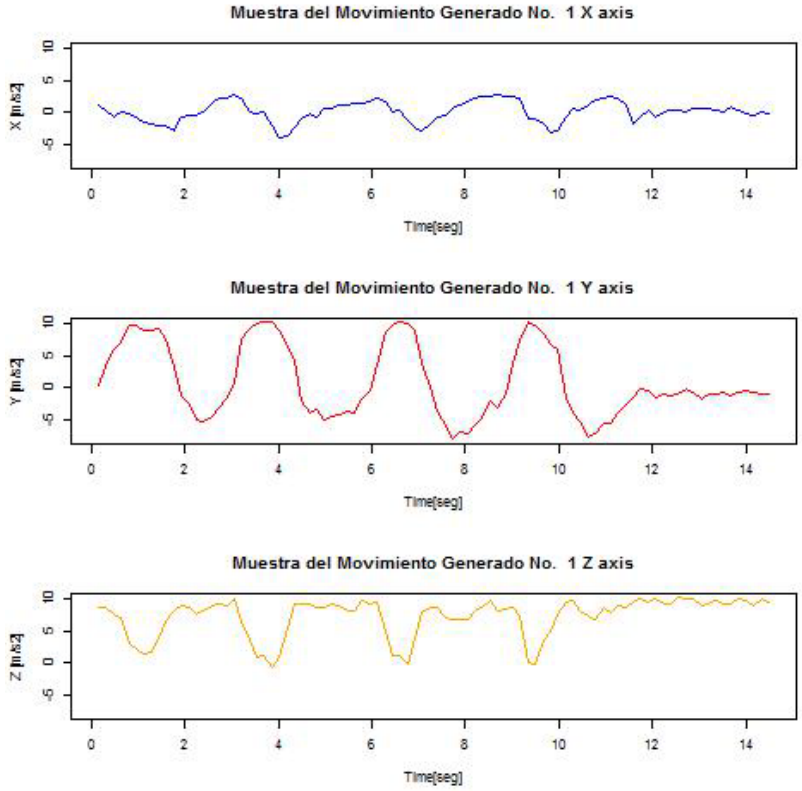


Figura 9.6. Gráfica de la aceleración del gesto *test* filtrado del *tag* Farsens

Por último, se solicita el tipo de muestra de *test* que se quiere comprobar (ver figura 9.7).

```
> ##################################################################
##################################
> # CONSOL .... [TRUNCATED]
[1] "Introduzca una prueba: "

>  print ('Escriba: Farsens o Movil ')
[1] "Escriba: Farsens o Movil "

>  test <- readline()
Farsens
```

Figura 9.7. Selección del tipo de muestra de *test*, *tag* Farsens o dispositivo móvil

El programa retorna el resultado de la decisión. Se trata de un gesto realizado del tipo de movimiento “W” con la probabilidad del 60% (ver figura 9.8):

```
[[1]]
[1] W
attr(,"prob")
[1] 0.6
Levels: C R W
```

Figura 9.8. Resultado de la decisión del método KNN de la muestra *test* del *tag* Farsens

Dispositivo Móvil

Se efectúa un gesto del tipo "W" con el dispositivo móvil. El gesto *test* producido se almacena en formato ".csv" en la carpeta "test/Movil" del entorno de trabajo (Ver Anexo C). El programa se encarga de ir al directorio para buscar la muestra de *test.*

El programa solicita una muestra de calibración del dispositivo móvil (ver figura 9.9).

```
> ##############################################################################################################
#######
> ### Calibramo .... [TRUNCATED]
[1] "Introduzca una muestra calibrada del dispositivo Movil"

> path<-file.choose()
```

Figura 9.9. Selección por consola de la muestra a calibrar del dispositivo móvil

La muestra del gesto producido sin filtrar y filtrado se puede visualizar en la figuras 9.10 y 9.11:

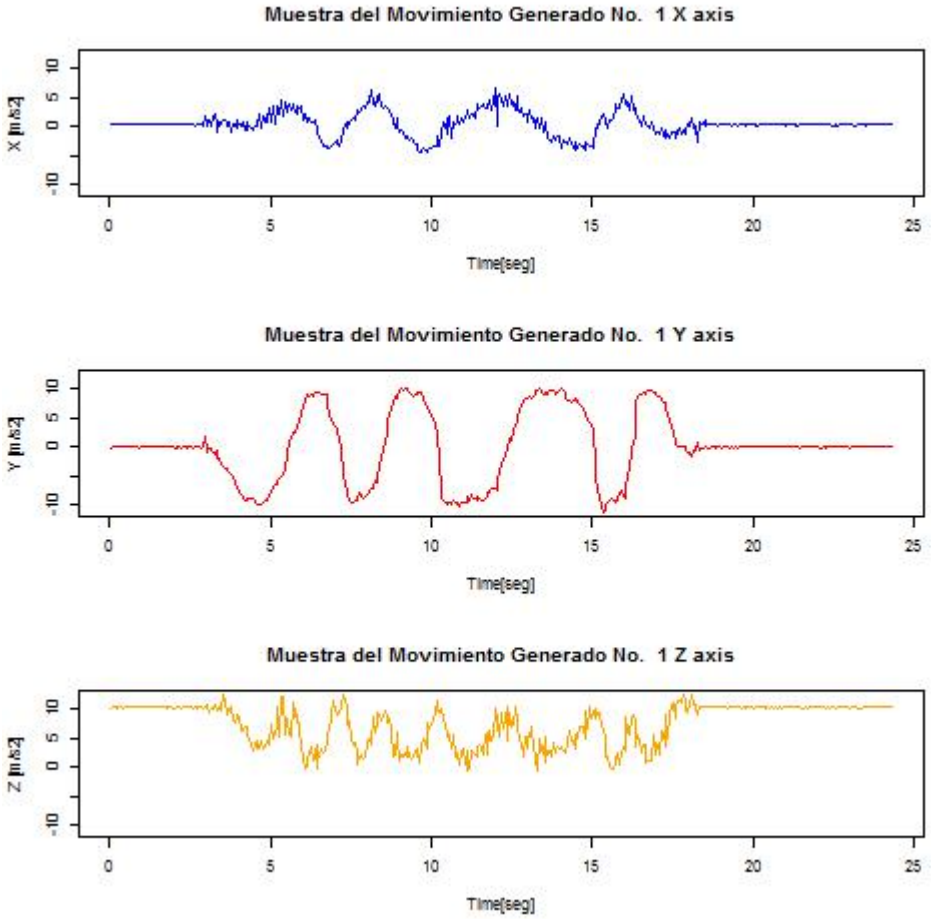


Figura 9.10. Gráfica de la aceleración del gesto *test* sin filtrar del dispositivo móvil

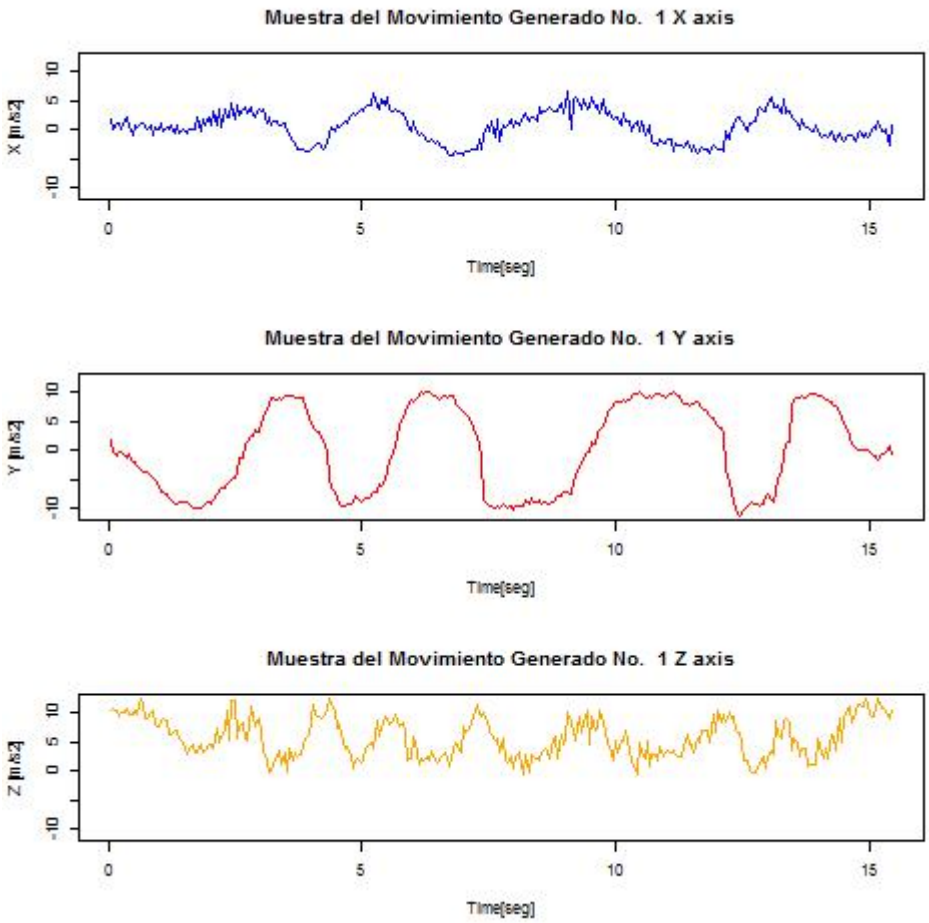


Figura 9.11. Gráfica de la aceleracióndel gesto *test* filtrado del dispositivo móvil

El programa retorna el resultado de la decisión. Se trata de un gesto realizado del tipo de movimiento “W” con la probabilidad del 80% (ver figura 9.12):

```
[[1]]
[1] W
attr(,"prob")
[1] 0.8
Levels: C R W
```

Figura 9.12. Resultado de la decisión del método KNN muestra *test* del dispositivo móvil

En la documentación anexa se puede consultar los resultados de las pruebas para el mismo gesto con diferentes valores de k (k=8, k= 5) (ver Anexo E).

CONCLUSIÓN

En este proyecto se ha realizado un estudio sobre el reconocimiento de gestos realizados por *tags* RFID pasivos y dispositivos móviles, ambos equipados con acelerómetros, como mecanismo de autenticación e identificación.

En la primera parte, el proyecto se centra en procesar y caracterizar de forma automatizada los datos generados por los dispositivos acelerómetros. Resulta importante detectar el momento en el que se produce un gesto para poder determinar con más exactitud la similitud de las muestras.

En la segunda parte del proyecto, se ha focalizado en el aprendizaje de los datos y en cómo ajustar los parámetros del algoritmo *Machine Learning*. Analizando las pruebas realizadas se concluye que para ciertos parámetros del algoritmo de aprendizaje y un conjunto reducido de muestras, se puede determinar con gran precisión qué gesto se está realizando y, por lo tanto, cumplir con las expectativas del caso de estudio.

Un aspecto a destacar es la calidad de los datos generados entre el prototipo *tag* de Farsens y el dispositivo móvil. Las pruebas demuestran que la clasificación de las muestras generadas por el dispositivo móvil se realiza de forma más precisa que la clasificación de las muestras generadas por el *tag* Farsens.

Para poder ser una realidad como sistema de autenticación, se debe reducir las dimensiones del *tag*, mejorar la generación de datos a tiempo real e incorporar una base de datos que identifique usuario, código EPC, gestos realizados y el móvil del usuario.

Se puede concluir que la tecnología RFID equipada con sensor acelerómetro y con el uso adecuado de métodos de aprendizaje puede constituir un sistema de autenticación de garantías y puede mejorar los sistemas actuales aportando información del entorno, en este caso del gesto realizado. Este proyecto incentiva a estudios futuros sobre cómo utilizar el *context recognition*, aporta mecanismos de seguridad basados en autenticación y manifiesta la necesidad de mejorar la privacidad y la integridad de la información en el uso de la tecnología RFID.

BIBLIOGRAFÍA

1. **Ubiquitous Computing Applications Laboratory (UbiCA Lab).** UbiCA Lab. *UbiCA Lab.* [Online] Department of Information and Communication Technologies (DTIC). [Cited: Octubre 23, 2013.] http://ubicalab.upf.edu/.

2. **ONSI , AETIC , AT4WIRELESS S.A.** Observatorio Nacional de las Telecomunicaciones y de la SI. [En línea] [Citado el: 26 de 12 de 2013.] http://www.ontsi.red.es/ontsi/ca/estudios-informes/la-tecnolog%C3%ADa-rfid-usos-y-oportunidades.

3. **Comisión Europea.** BRIDGE. *BRIDGE*. [En línea] [Citado el: 29 de Enero de 2013.] http://www.bridge-project.eu/index.php/mainpage/en/.

4. **European Commission Information Society and Media.** *Vision and Challenges for Realising the Internet of things.* [ed.] Harald Sundmaeker Patrick Guillemin Peter Friess Sylvie Woelfflé. Luxembourg : Publications Office of the European Union, 2010.

5. **Swedberg, Claire.** RFID JOURNAL. [En línea] 30 de Enero de 2014. [Citado el: 02 de Febrero de 2014.] http://espanol.rfidjournal.com/noticias/vision?11393.

6. *Enabling ubiquitous sensing with RFID.* **Want, Roy.** [ed.] IEEE. 4, 2014, Computer, Vol. 37, págs. 84-86.

7. *Implementation of Smart Tags of RFID Technology in Poisonous Area*. **Davood Karimzadgan Moghaddam, Davood Vahdat,Pejman Ravand.** 2010, LATEST TRENDS on COMPUTERS, págs. 320-324.

8. **Roberti, Mark.** RFID JOURNAL. *RFID JOURNAL.* [En línea] 16 de Enero de 2005. [Citado el: 05 de Enero de 2014.] http://www.rfidjournal.com/articles/view?1338.

9. **Dobkin, Daniel.** *THE RF IN RFID.* Burlington,USA : Newnes, 2008.

10. **MIT,CAMBRIDGE,ST.GALLEN,KAIST,FUDAN,ADELAIDE,KEIO.** AUTO-ID LABS. *AUTO-ID LABS.* [En línea] [Citado el: 06 de Enero de 2014.] http://www.autoidlabs.org/.

11. **EPC Global Inc and GS1 Data Excellence Inc.** GS1 THE GLOBAL LANGUAJE OF BUSSINES. *GS1 THE GLOBAL LANGUAJE OF BUSSINES.* [En línea] [Citado el: 08 de Enero de 2014.] http://www.gs1.org/epcglobal.

12. **Choperena, Mikel.** RFID Journal. *RFID Journal.* [En línea] 23 de Junio de 2013. [Citado el: 25 de Enero de 2014.] http://www.rfidjournal.com/articles/view?10784.

13. *RFID:Normativas y estandares.* **Dessenne, Gerard-André.** diciembre de 2005, DATA Collection, págs. 20-30.

14. **España, Ministerio de Sanidad y Politica Social.Gobierno de.** *Informe Plan de Calidad para el Sistema Nacional de Salud 2006-2010.* s.l. : Ministerio de Sanidad y Politica Social.Gobierno de España, 2009.

15. **INNOVENTIONS, Inc. .** Google play. *Google play.* [En línea] INNOVENTIONS, Inc. , 18 de Septiembre de 2013. [Citado el: 22 de Febrero de 2014.] https://play.google.com/store/apps/details?id=com.innoventions.sensorkineticspro.

16. **Voris, Nitesh Saxena and Jonathan.** Motion Detection for Enhanced RFID Security and Privacy without Changing the Usage Model. *Radio Frequency Identification: Security and Privacy Issues.* Instanbul,Turkey : Springer Berlin Heidelberg, 2010, págs. 2-21.

17. *HB++: a Lightweight Authentication Protocol Secure against Some Attacks.* **Julien BRINGER, Herve CHABANNE,Emmanuelle DOTTAX.** Eragny sur Oise : IEE, 2006. págs. 28-33. 0-7695-2549-0.

18. **Ari Juels, Stephen A. Weis.** Authenticating Pervasive Devices with Human Protocols. [aut. libro] Victor Shoup. *Advances in Cryptology – CRYPTO 2005.* Bedford,USA : Springer, 2005, Vol. 3621, págs. 293-3008.

19. **Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador,Arturo Ribagorda.** RFID Systems: A Survey on Security Threats and Proposed Solutions. [aut. libro] FIP-TC6 11th International Conference on Personal Wireless Communications. [ed.] Luiz Orozco-Barbosa Pedro Cuenca. *Personal Wireless Communications.* Madrid : Springer, 2006, Vol. 4217, págs. 159-170.

20. **Paris Kitsos, Yan Zhang.** *RFID Security:Techniques, Protocols and System-On-Chip Design*. s.l. : Springer, 2008. pág. 458. 038776481X, 9780387764818.

21. **Ari Juels, Ronald L Rivest,Michael Szydlo.** *The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy*. s.l. : ACM , 2003. págs. 103-111.

22. **Christian Floerkemeier, Roland Schneider,Marc Langheinrich.** Scanning with a purpose–supporting the fair information principles in RFID protocols. *Ubiquitous Computing Systems*. Berlin,Heidelberg : Springer, 2005, págs. 214-231.

23. **Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum.** RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. *Information Security and Privacy*. Amsterdam : Springer Berlin Heidelberg, 2005, págs. 184-194.

24. *For an EPC-C1 G2 RFID compliant Protocol, CRC with Concatenation: No; PRNG with Concatenation: Yes.* **Masoumeh Safkhani, Nasour Bagheri.** Teherán,Irán : s.n., 2013, IACR Cryptology ePrint Archive, Vol. 2013, pág. 490.

25. **NXP Semiconductors, Sony and Nokia.** NFC Forum. *NFC Forum.* [En línea] 18 de Marzo de 2004. [Citado el: 30 de Junio de 2014.] http://nfc-forum.org/.

26. **Thomas J. P. Wiechert, Frédéric Thiesse, Florian Michahelles,.** Auto-ID Labs. *Auto-ID Labs.* [En línea] 07 de Enero de 2008. [Citado el: 30 de Junio de 2014.] http://www.autoidlabs.org/single-view/dir/article/6/291/page.html.

27. **Homin K. Lee, Tal Malkin,Erich Nahum.** Cryptographic strength of ssl/tls servers: current and recent practices. *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* . s.l. : ACM, 2007, págs. 83-92.

28. *Enhancing security and privacy in biometrics-based authentication systems.* **N.K. Ratha, J.H. Connell, R.M. Bolle.** 3, s.l. : IBM systems Journal, 2001, Vol. 40, págs. 614-634.

29. *Machine Learning Methods for Classifying Human Physical.* **Sabatini, Andrea Mannini and Angelo Maria.** 2010, Sensors 2010, págs. 1154-1175.

30. *Activity Recognition using Cell Phone Accelerometers.* **Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore.** 2010, SensorKDD '10, págs. 10-18.

31. *Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach.* **Mi Zhang, Alexander A.Sawchuk.** 2012, IHI '12, págs. 631-640.

32. *Preprocessing Techniques for Context Recognition from Accelerometer Data.* **Figo, Davide, y otros.** 7, Porto : Springer-Verlag, 2010, Vol. 14, págs. 645-662.

33. *Dynamic Programming Algorithm Optimization for Spoken Word Recognition .* **Hiroaki Sakoe, Seibi Chiba.** 1978, IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, págs. 43-49.

34. *Everything you know about Dynamic Time Warping is Wrong.* **Chotirat Ann Ratanamahatana, Eamonn Keogh.** Seattle : s.n., 2004, Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, págs. 22-25.

35. *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.* **Giorgino, Toni.** 7, 2009, Journal of Statistical Software, Vol. 31, págs. 1-24.

36. **Bishop, Christopher M.** *Pattern Recognition and Machine Learning.* New York : Springer Verlag, 2006. pág. 740.

37. **Murphy, Kevin P.** *Machine Learning: A Probabilistic Perspective.* s.l. : Mit Press, 2012. pág. 1067.

38. *Top 10 algorithms in data mining.* **Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang,Hiroshi Motoda, Geoffrey J. McLachlan , Angus Ng, Bing Liu , Philip S. Yu,Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg.** s.l. : Springer, 2008, IEEE International Conference on Data Mining (ICDM) in December 2006, págs. 1-37.

39. **ETH Zurich.** k-Nearest Neighbour Classification. *k-Nearest Neighbour Classification.* [En línea] http://stat.ethz.ch/R-manual/R-devel/library/class/html/knn.html.

40. **SCHOOL OF ENGINEERING AND COMPUTER SCIENCE.** SCHOOL OF ENGINEERING AND COMPUTER SCIENCE. *SCHOOL OF ENGINEERING AND COMPUTER SCIENCE.* [En línea] UNIVERSITY OF WELLINGTON. [Citado el: 31 de Mayo de 2014.] http://ecs.victoria.ac.nz/Courses/COMP307_2014T1/Lect13-team1.

41. *An introduction to RFID technology.* **Want, Roy.** [ed.] IEEE. 1, 2006/1, Pervasive Computing, IEEE, Vol. 5, págs. 25-33.

42. *The history of RFID.* **Landt, Jeremy.** OCTUBRE-NOVIEMBRE de 1995, Potentials, IEEE, págs. 8-11.

43. **GS1 EPCglobal Inc.** GS1 the Global language of bussines. *GS1 the Global language of bussines.* [En línea] Noviembre de 2013. [Citado el: 30 de Enero de 2014.] http://www.gs1.org/sites/default/files/docs/uhfc1g2/uhfc1g2_2_0_0_standard_20131101.pdf.

44. *Classifying of RFID Attacks and defenses.* **Aikaterini Mitrokotsa, Melanie R. Rieback, Andrew S. Tanenbaum.** 5, Amsterdam : Springer, 2009, Vol. 12, págs. 491-505.

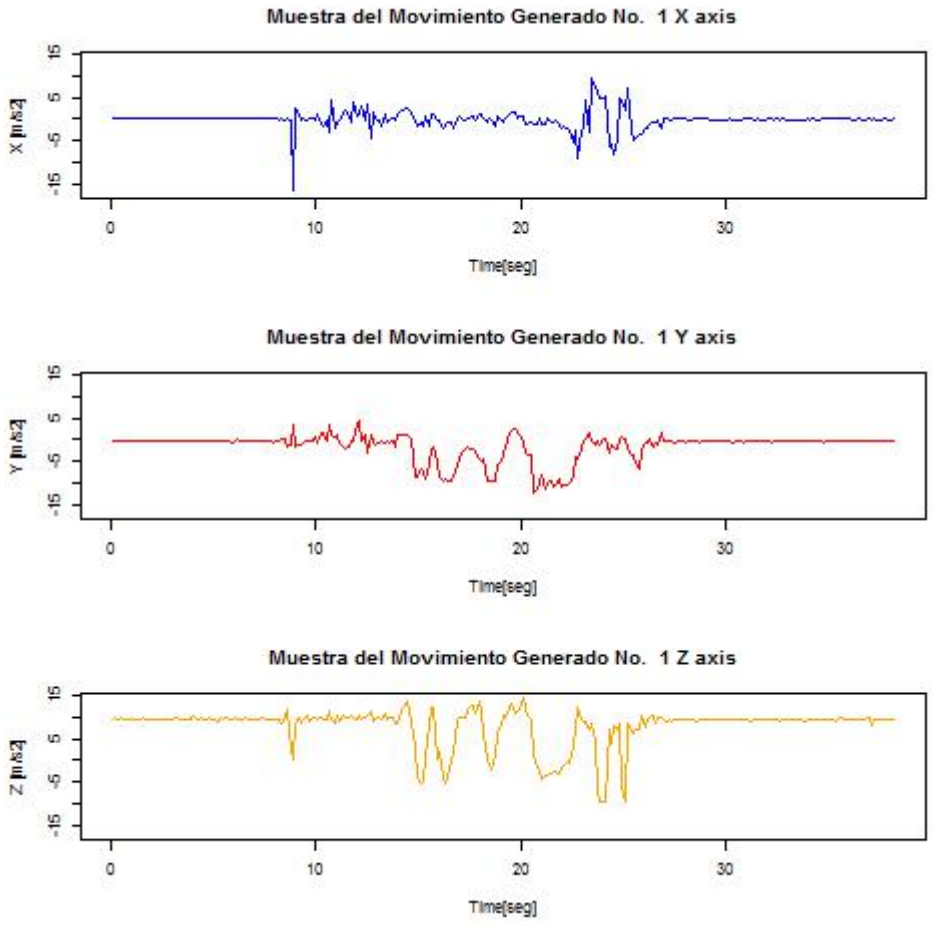
45. **Barcodes Inc.** barcodesinc. *barcodesinc.* [En línea] [Citado el: 22 de Marzo de 2014.] http://www.barcodesinc.com/pdf/Sirit/infinity-610.pdf.

ANEXOS

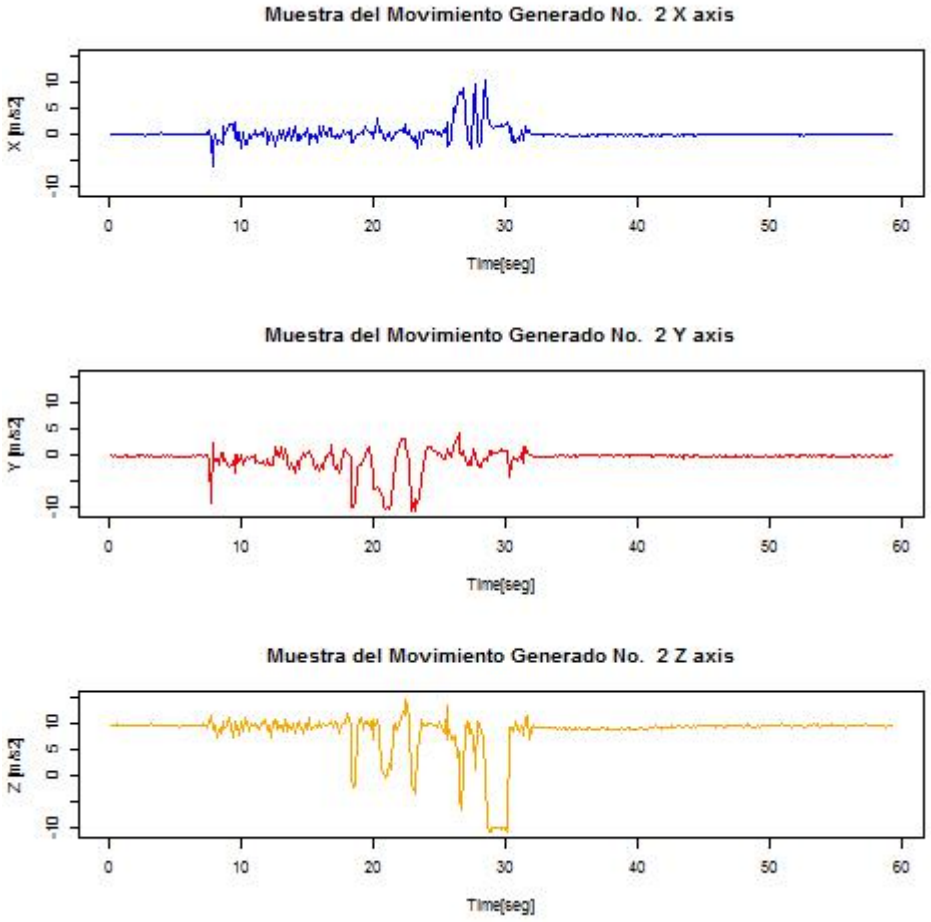
A. BASE DE DATOS DE MUESTRAS

A.1 Gestos modelo “C”

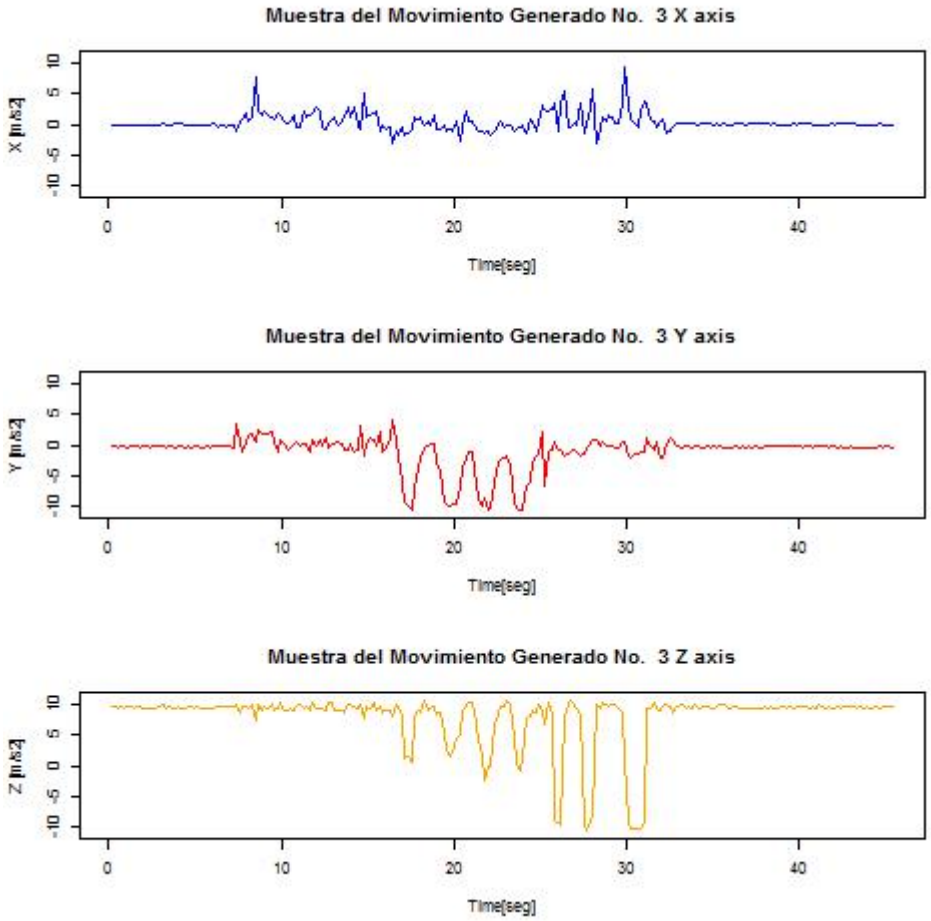
A.1.1 Muestra 1 del *Tag Farsens*



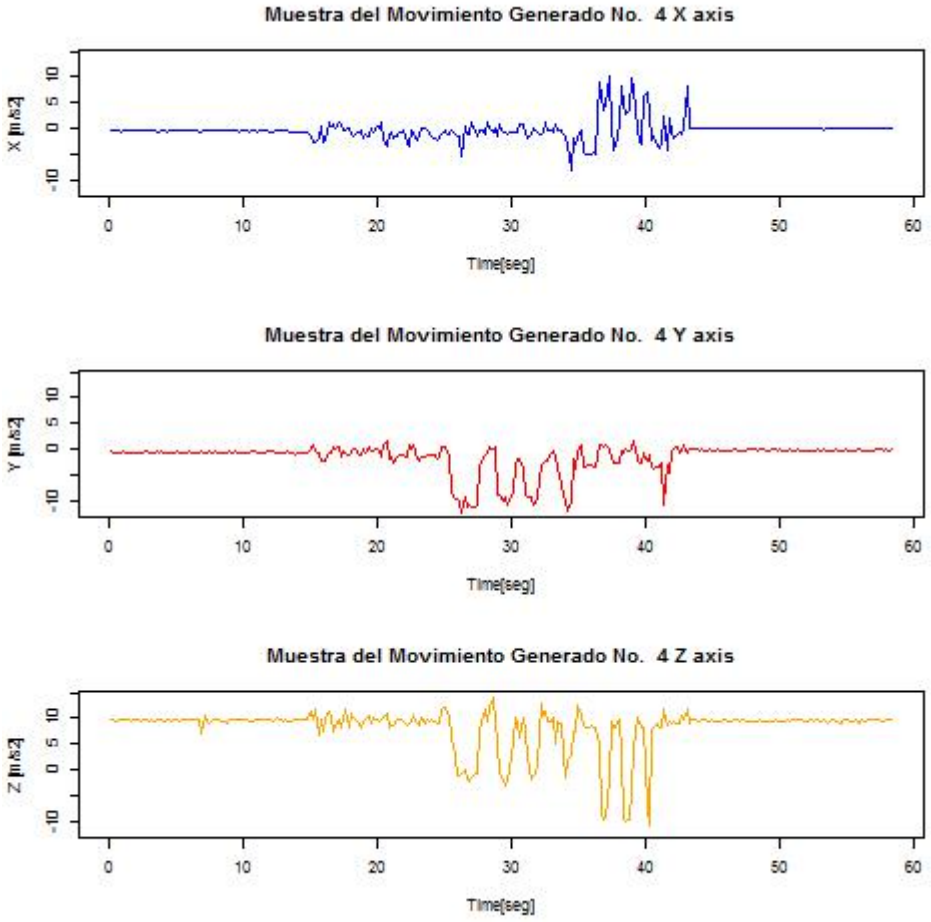
A.1.2 Muestra 2 del *tag* Farsens



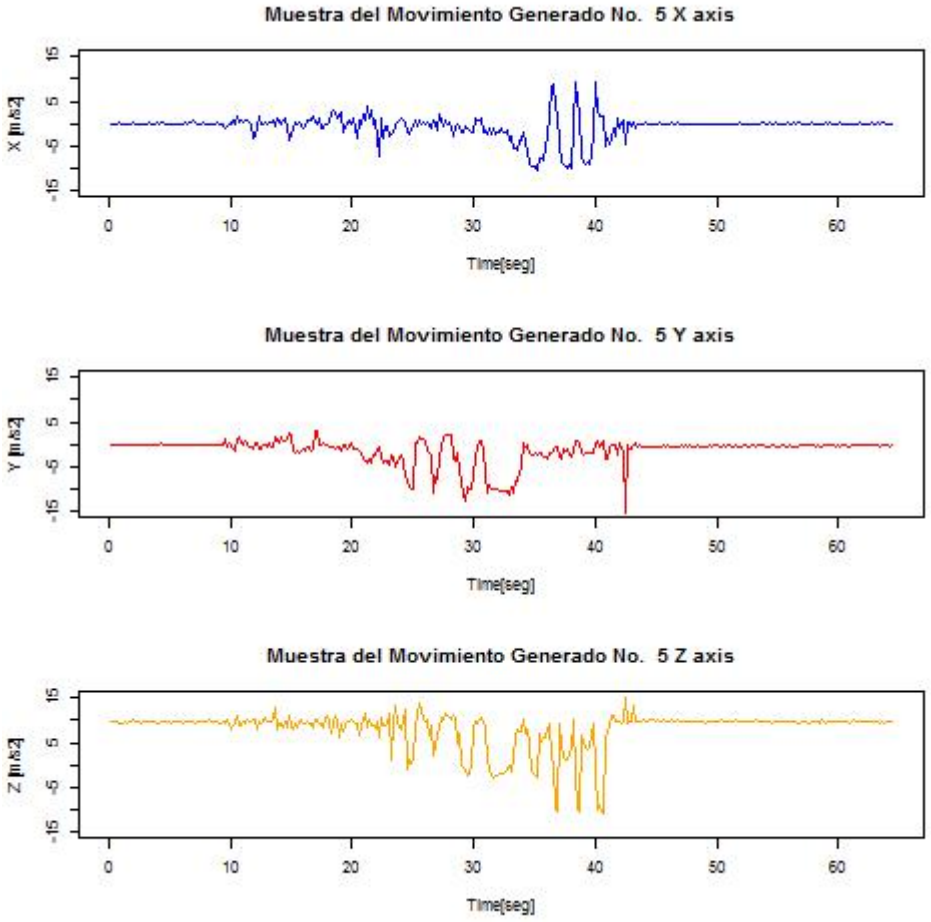
A.1.3 Muestra 3 del *tag* Farsens



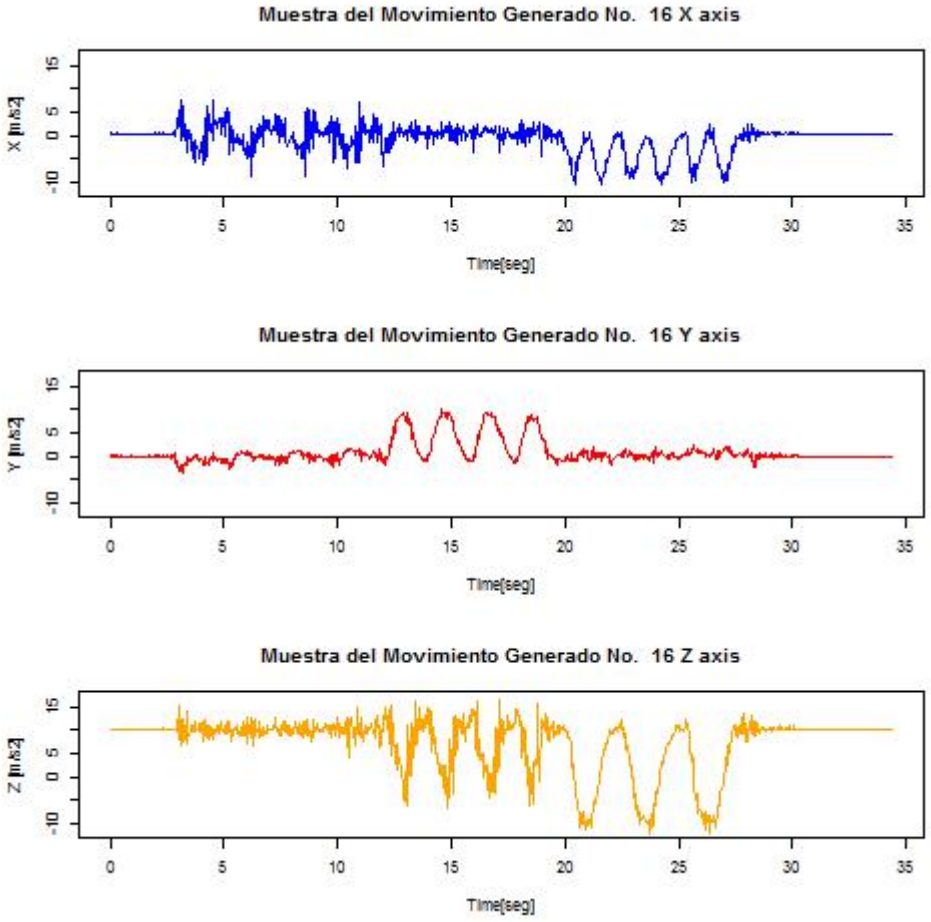
A.1.4 Muestra 4 del *tag* Farsens



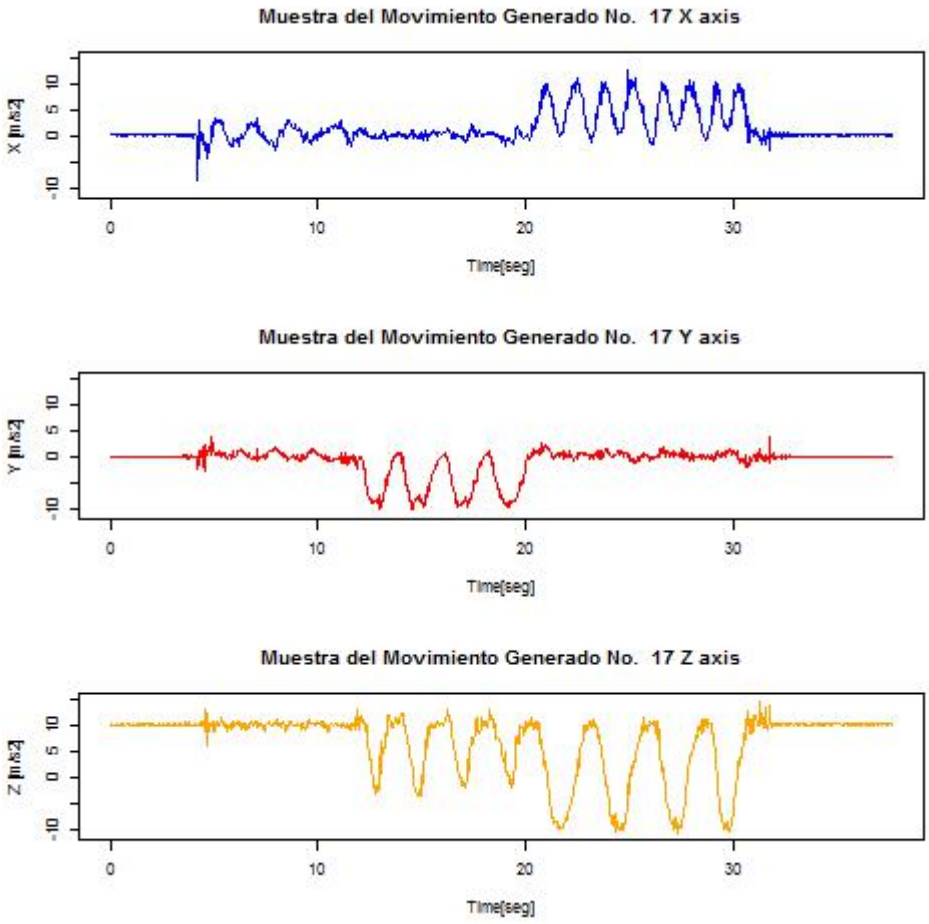
A.1.5 Muestra 5 del *tag* Farsens



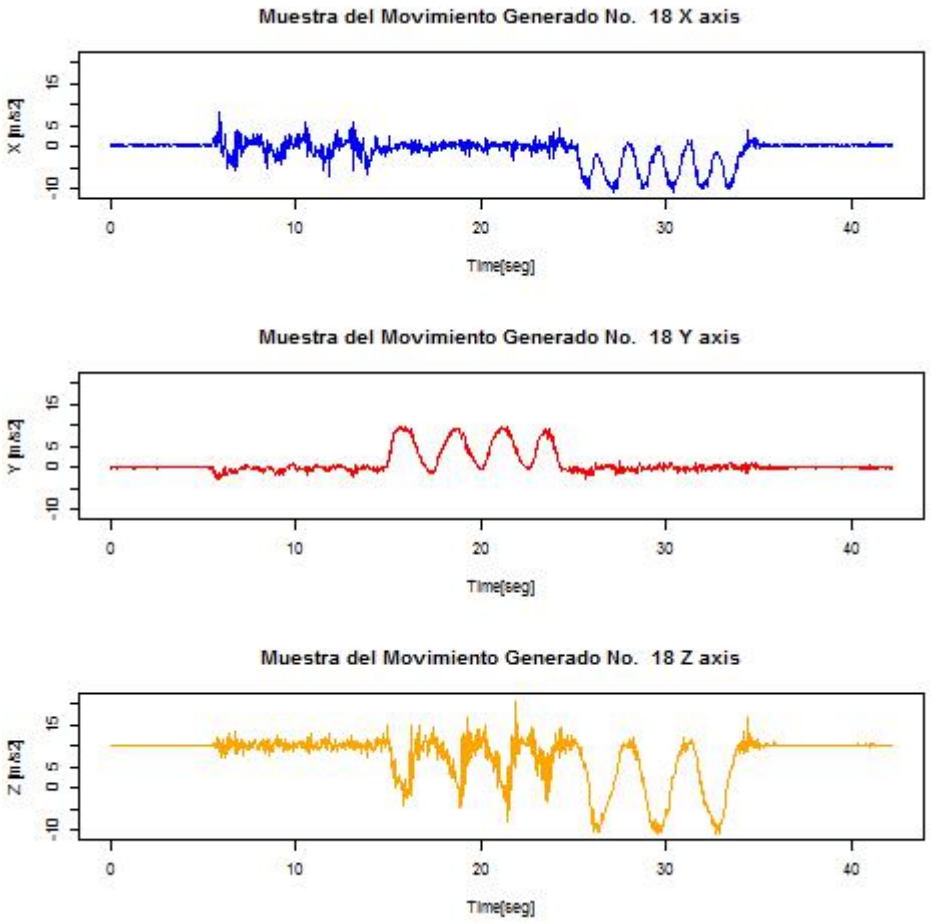
A.1.6 Muestra 6 del dispositivo móvil



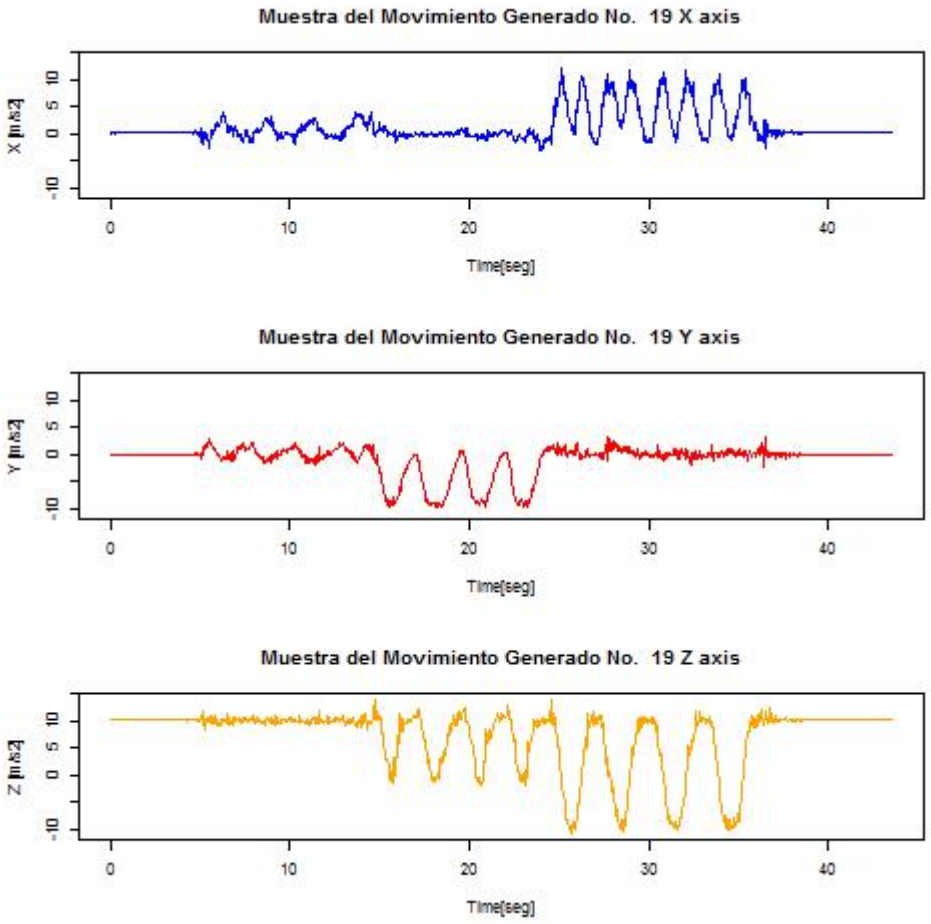
A.1.7 Muestra 7 del dispositivo móvil



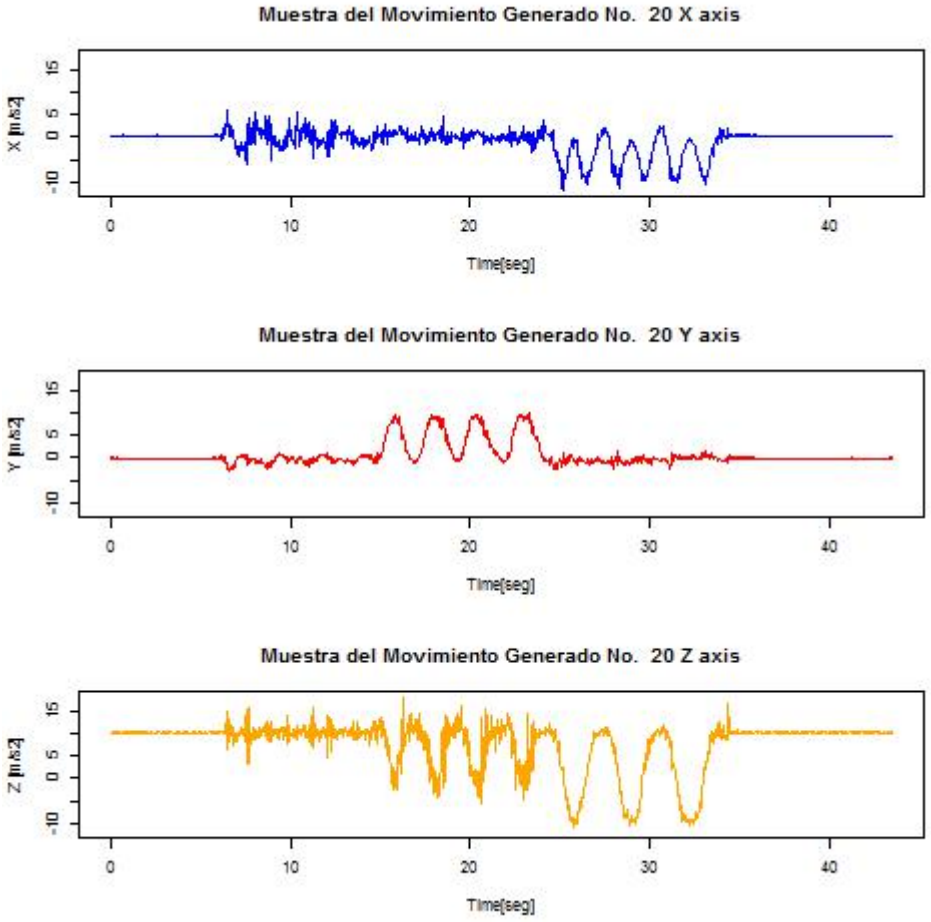
A.1.8 Muestra 8 del dispositivo móvil



A.1.9 Muestra 9 del dispositivo móvil

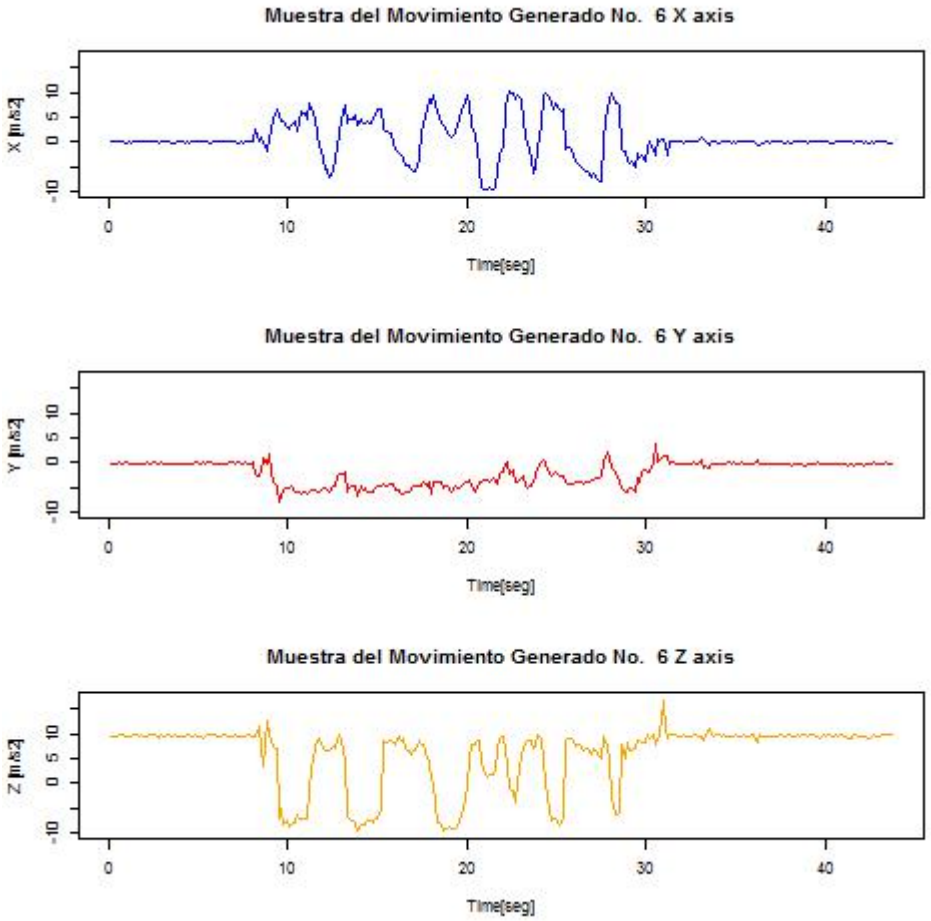


A.1.10 Muestra 10 del dispositivo móvil

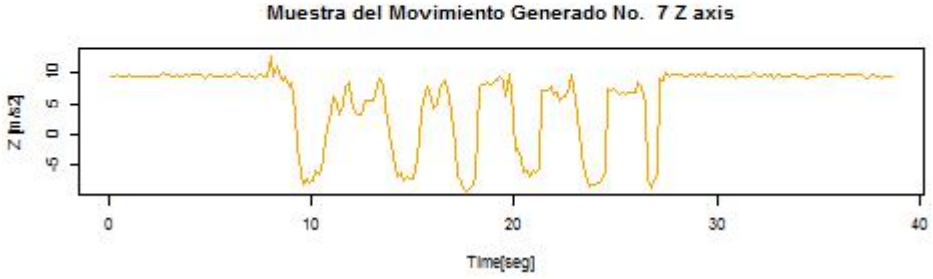


A.2 Gestos modelo “R”

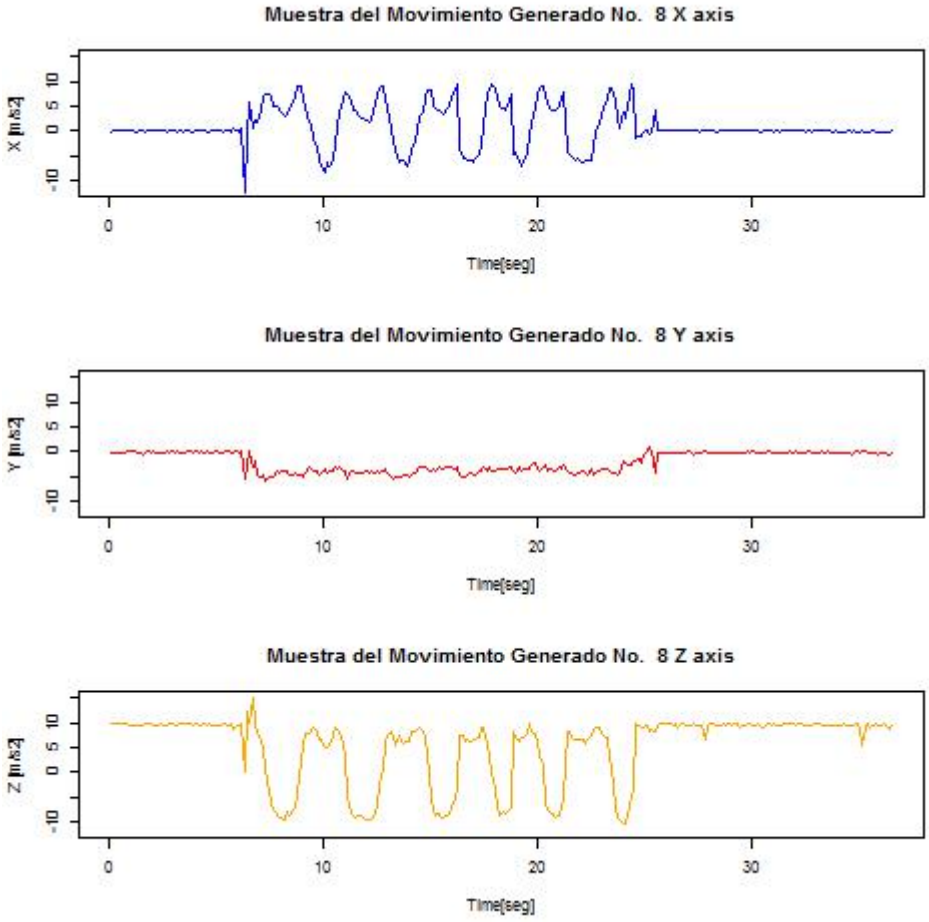
A.2.1 Muestra 1 del *tag* Farsens



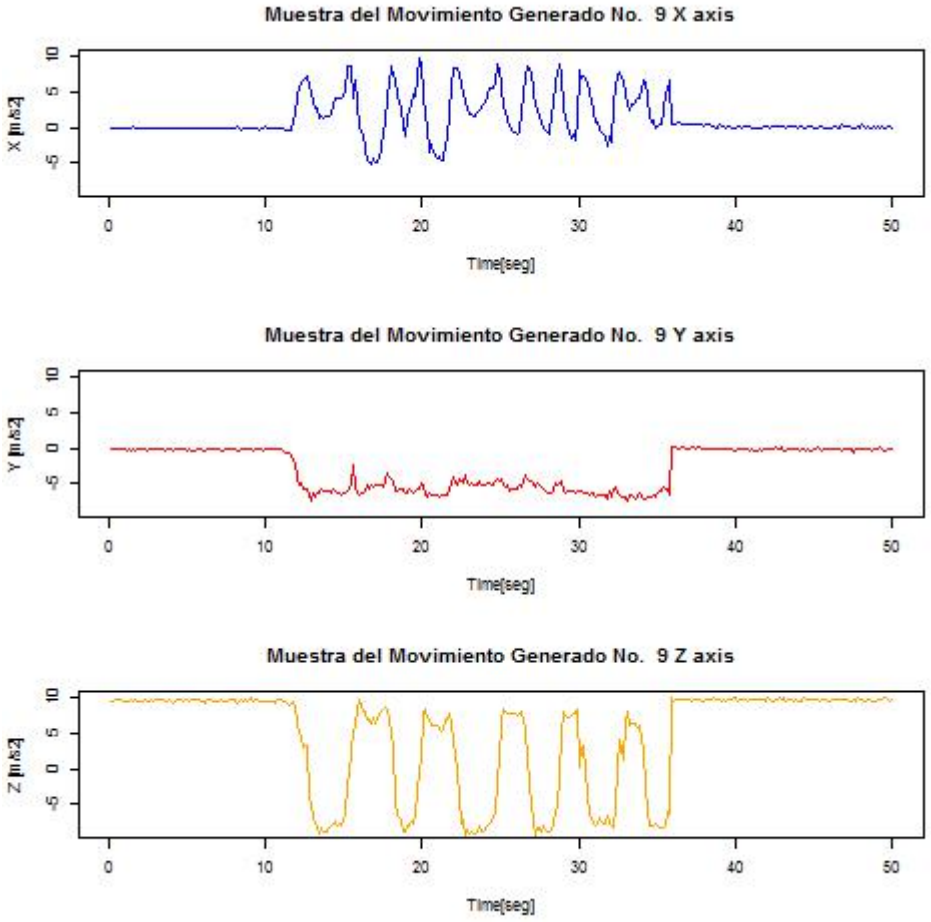
A.2.2 Muestra 2 del *tag Farsens*



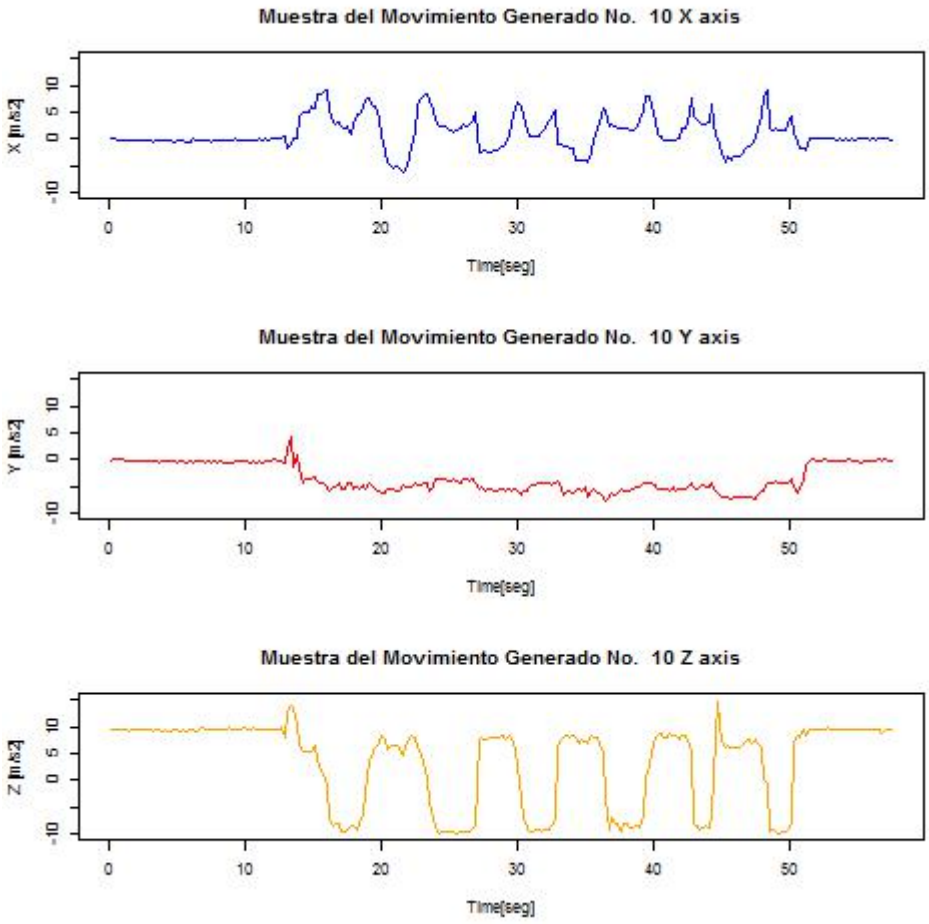
A.2.3 Muestra 3 del *tag* Farsens



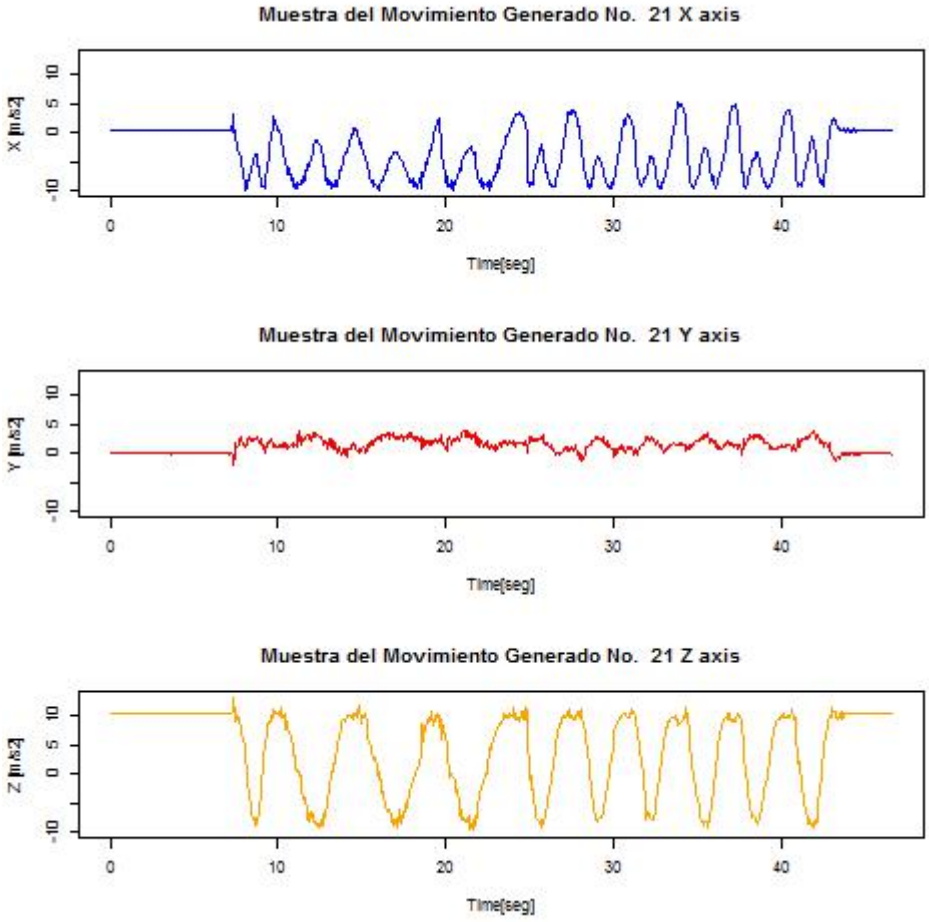
A.2.4 Muestra 4 del *tag* Farsens



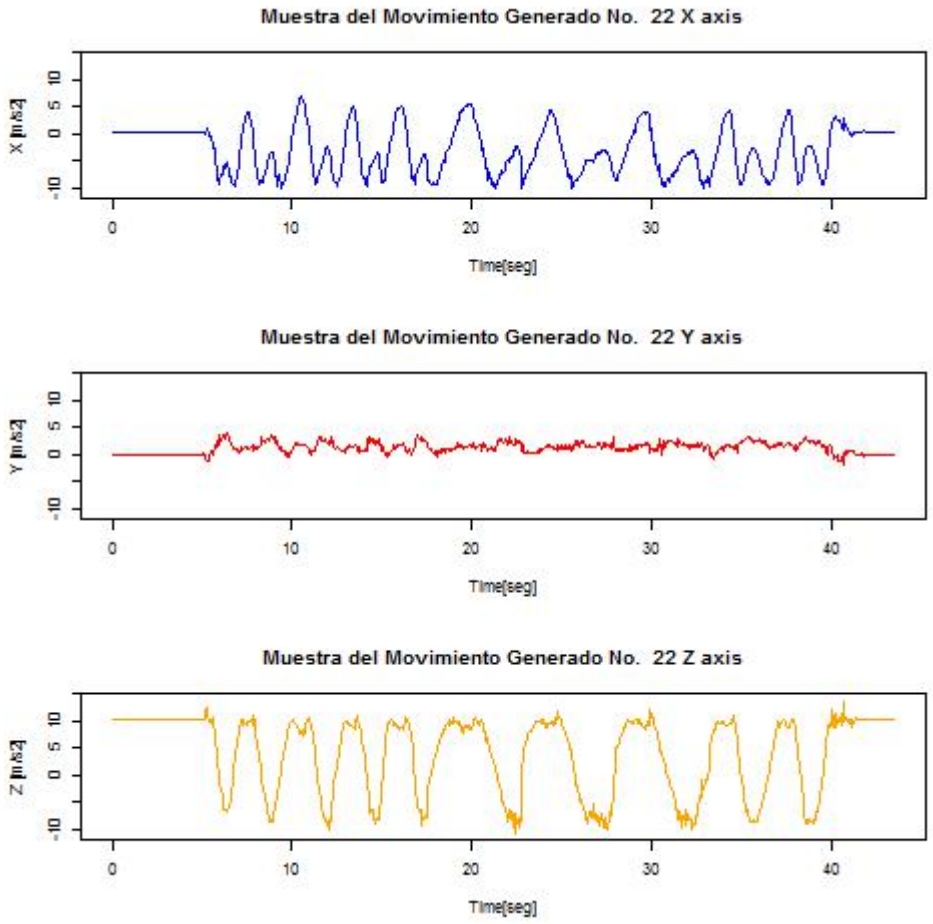
A.2.5 Muestra 5 del *tag* Farsens



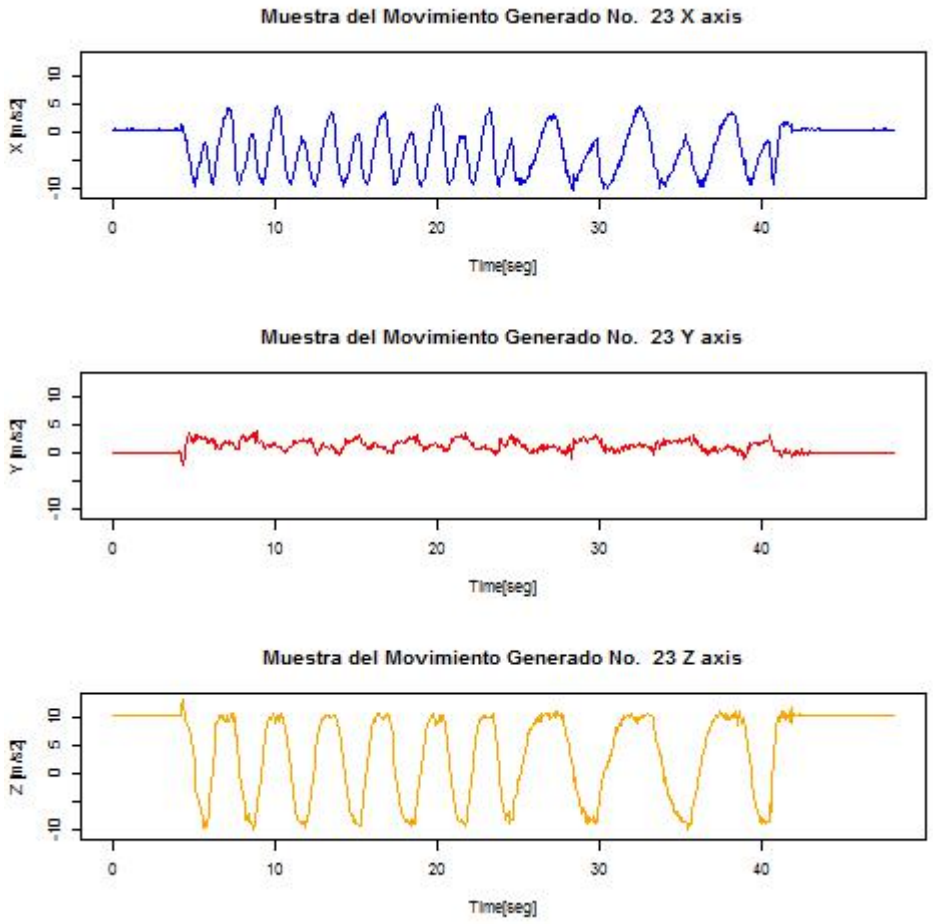
A.2.6 Muestra 6 del dispositivo móvil



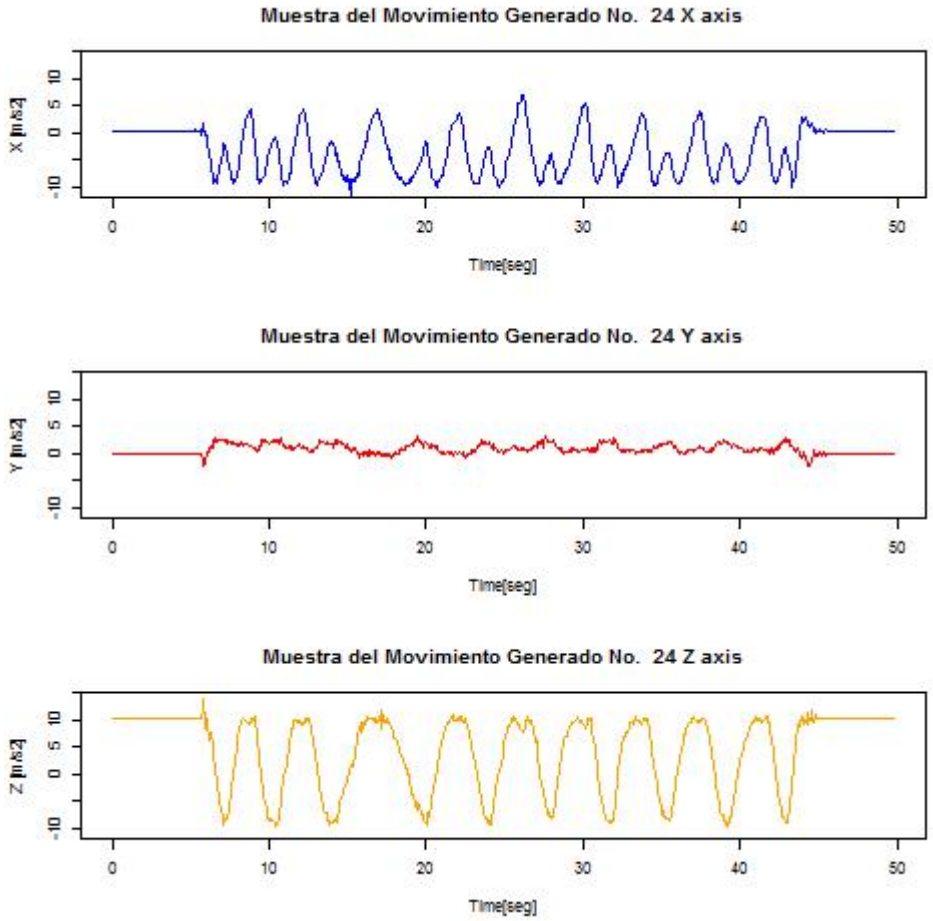
A.2.7 Muestra 7 del dispositivo móvil



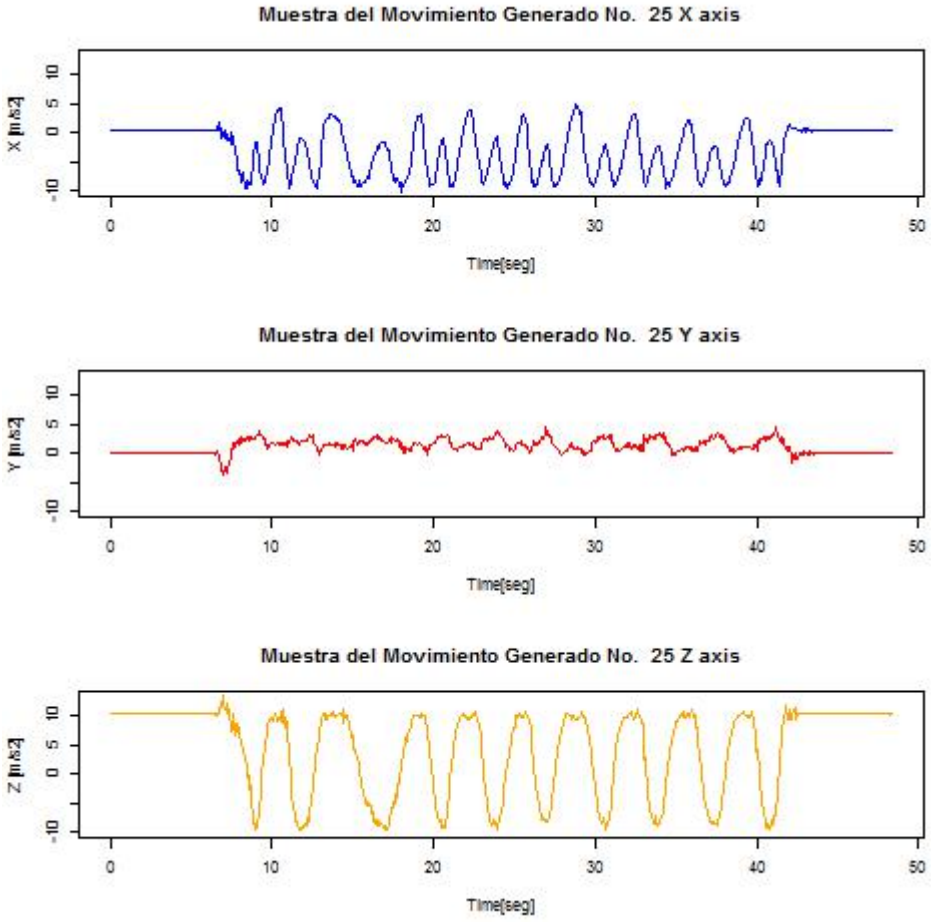
A.2.8 Muestra 8 del dispositivo móvil



A.2.9 Muestra 9 del dispositivo móvil

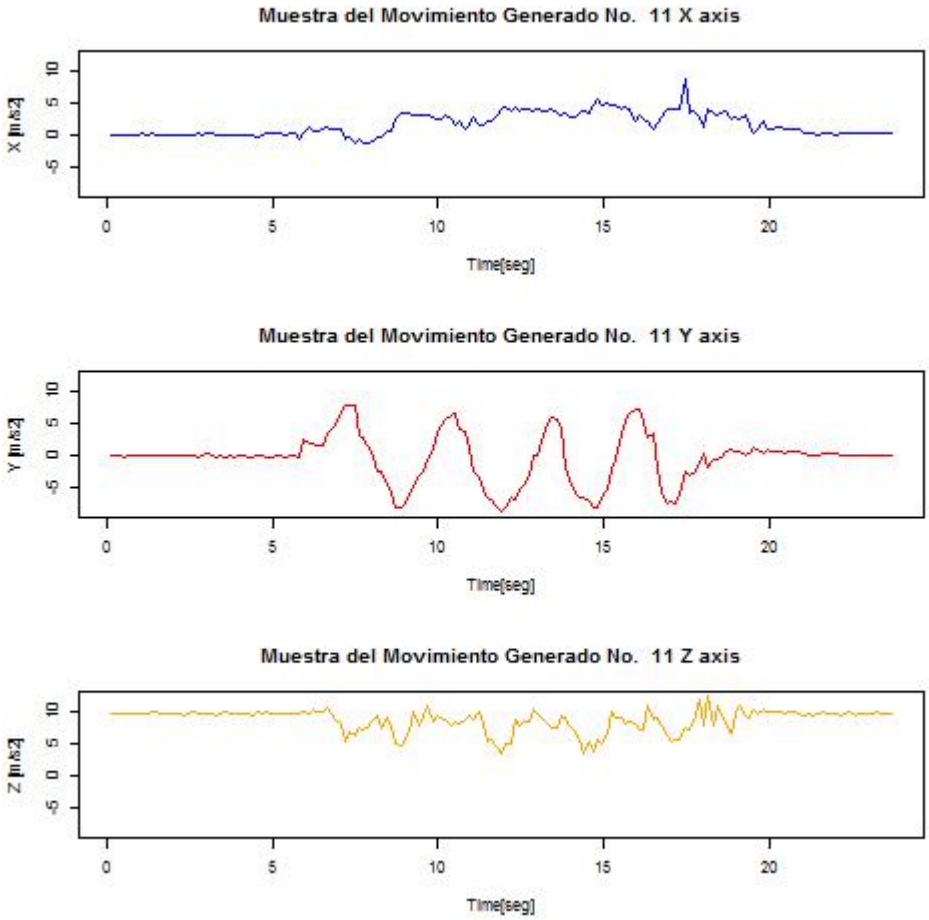


A.2.10 Muestra 10 del dispositivo móvil

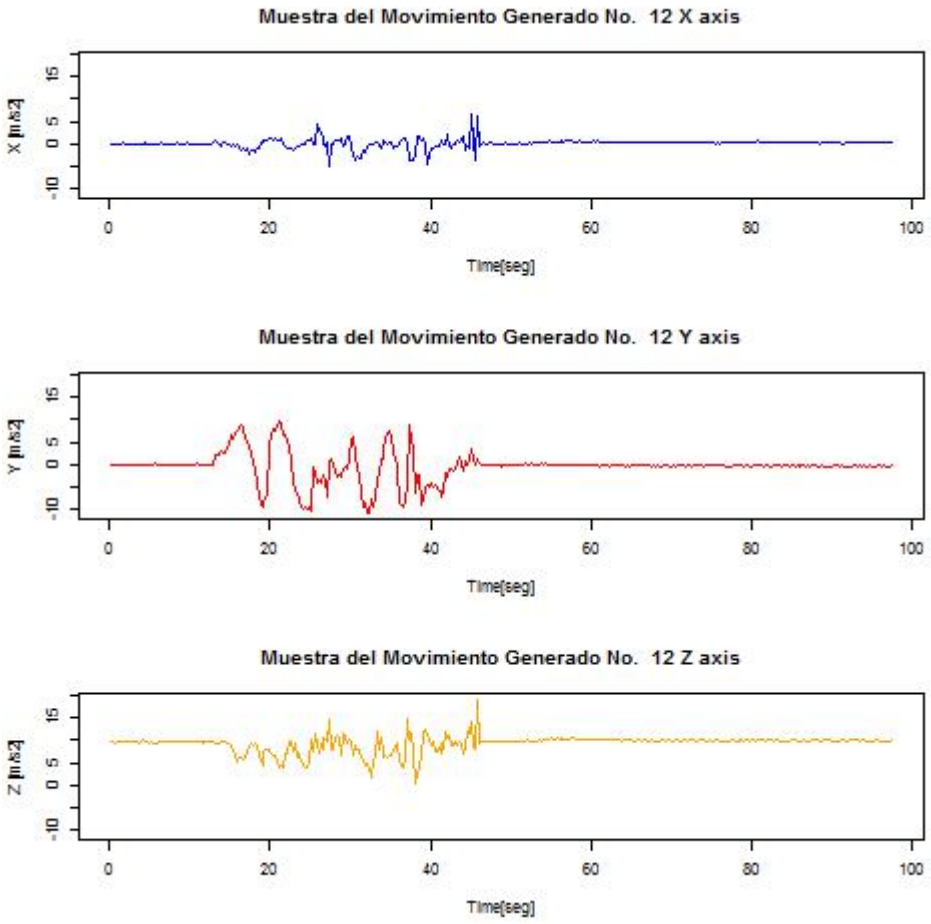


A.3 Gestos modelo “W”

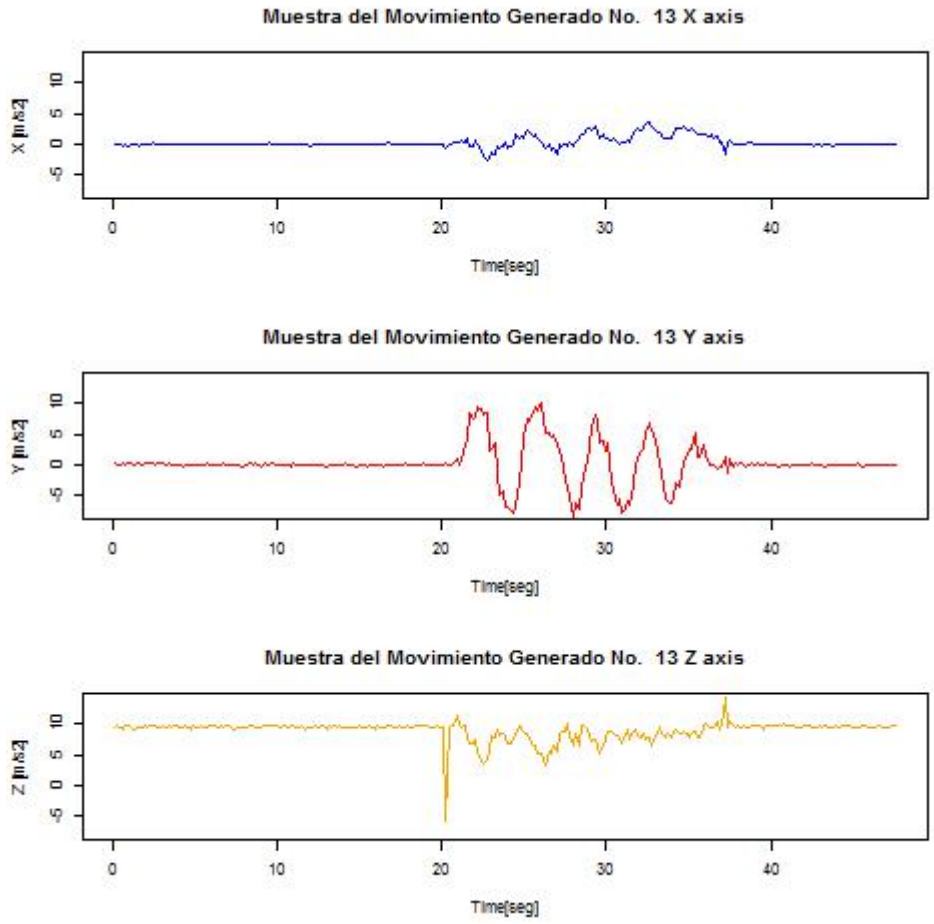
A.3.1 Muestra 1 del *tag* Farsens



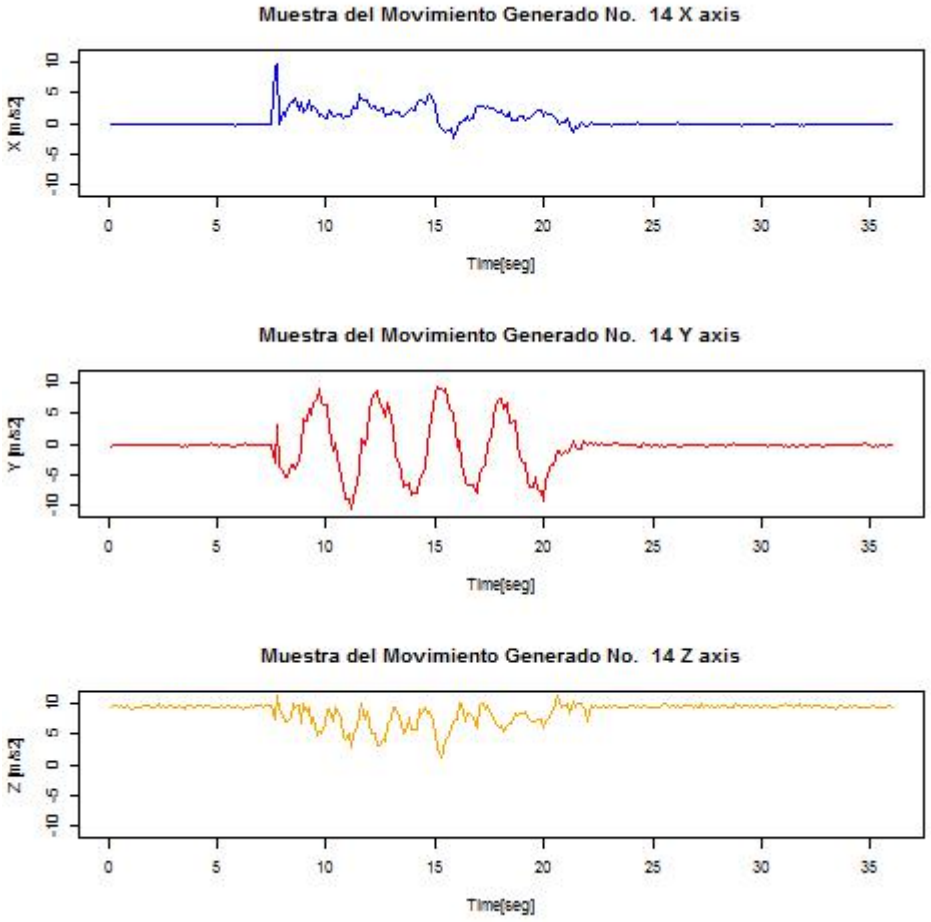
A.3.2 Muestra 2 del *tag* Farsens



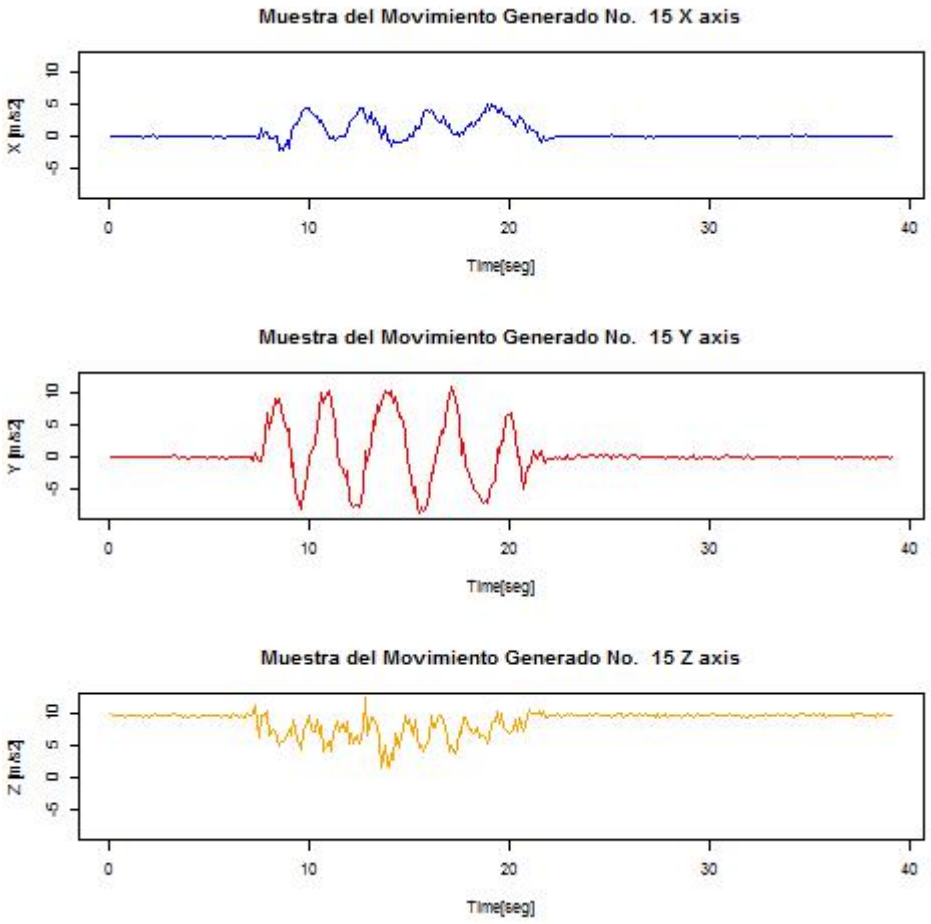
A.3.3 Muestra 3 del *tag* Farsens



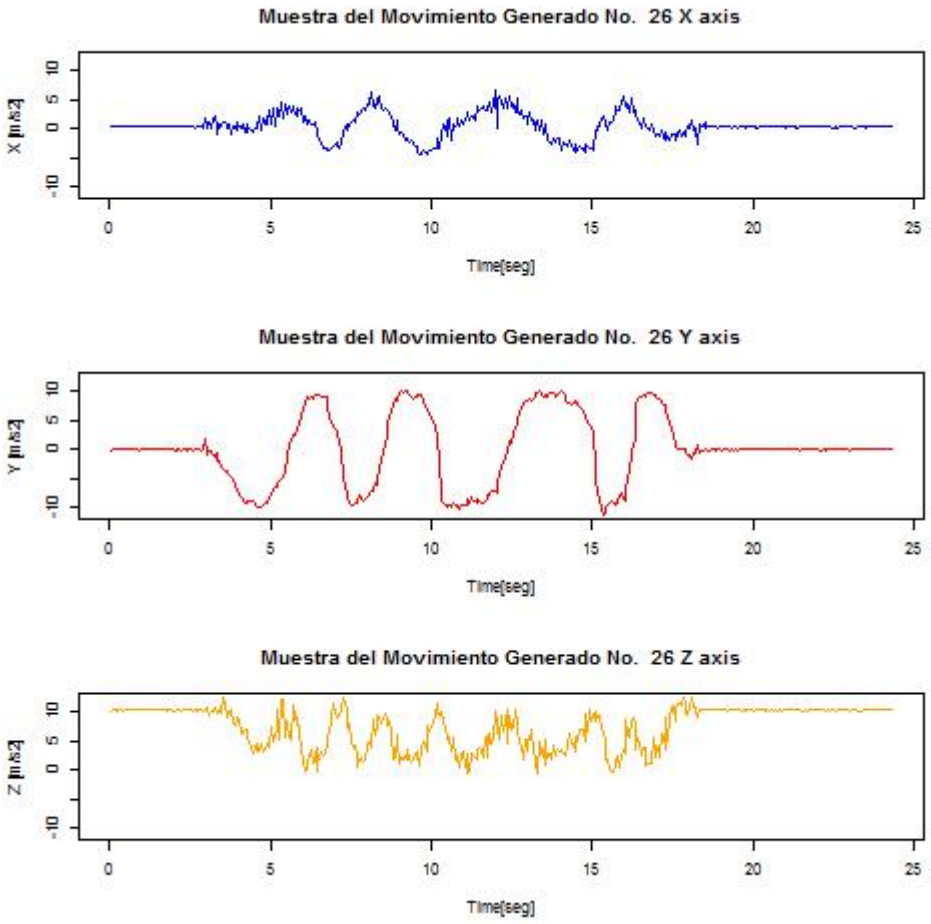
A.3.4 Muestra 4 del *tag* Farsens



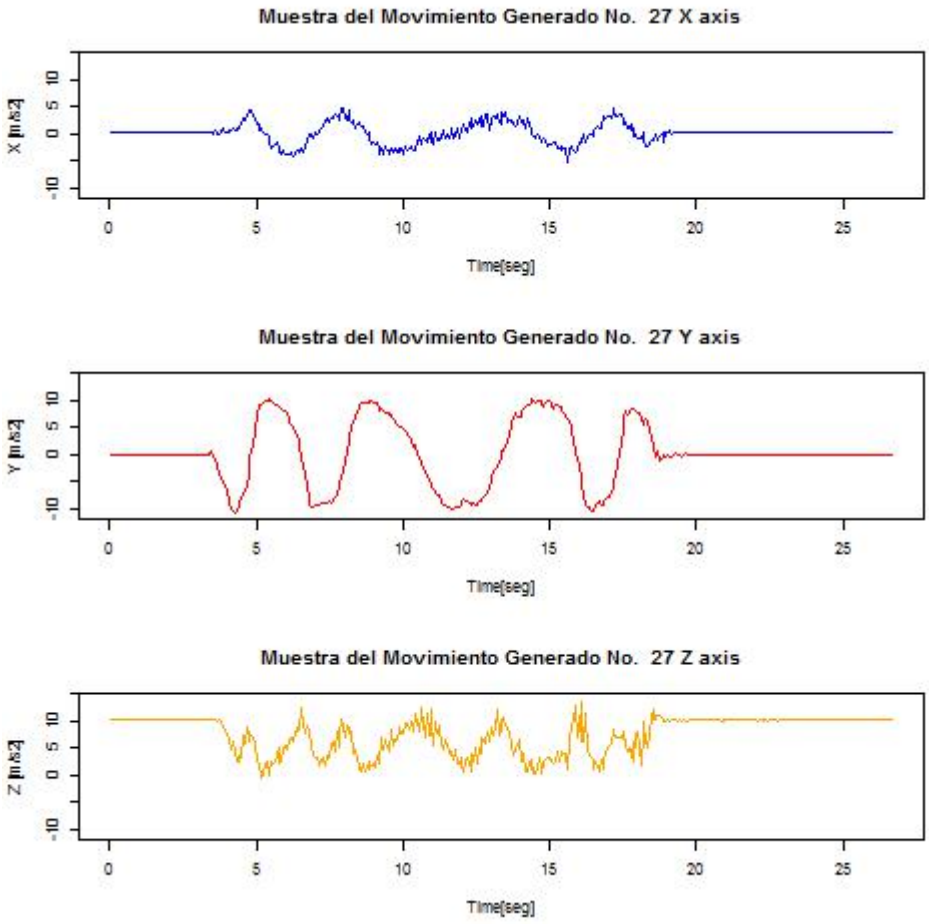
A.3.5 Muestra 5 del *tag* Farsens



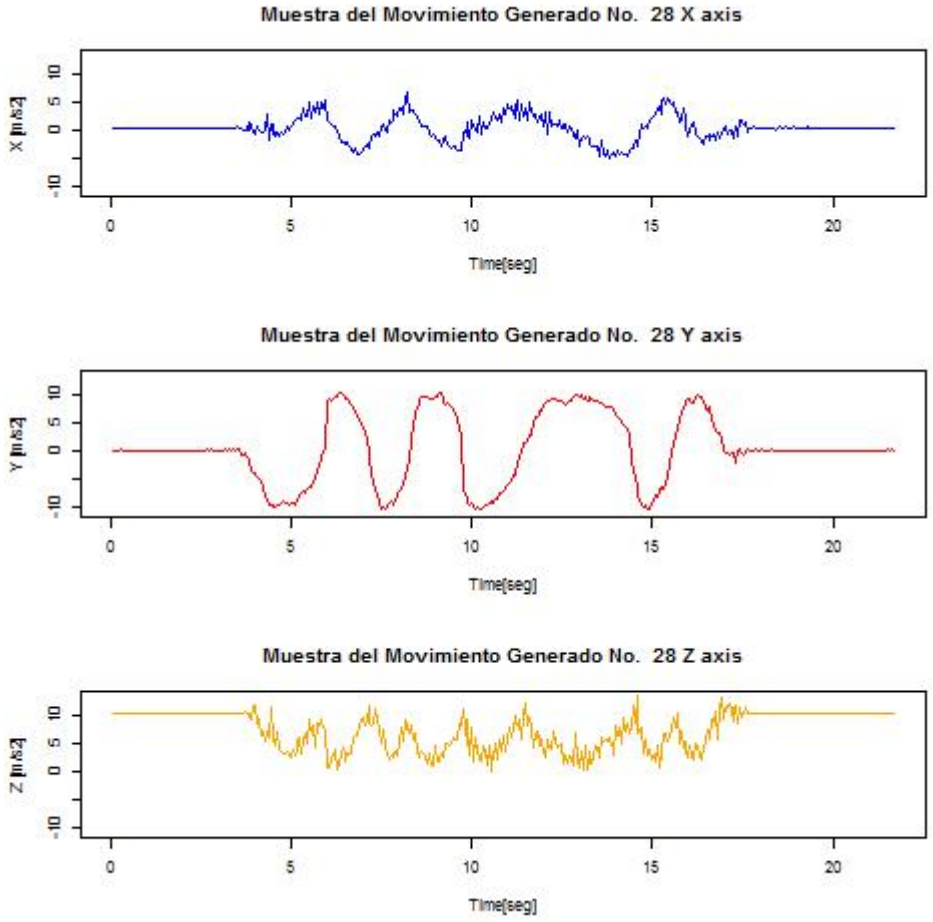
A.3.6 Muestra 6 del *tag* móvil



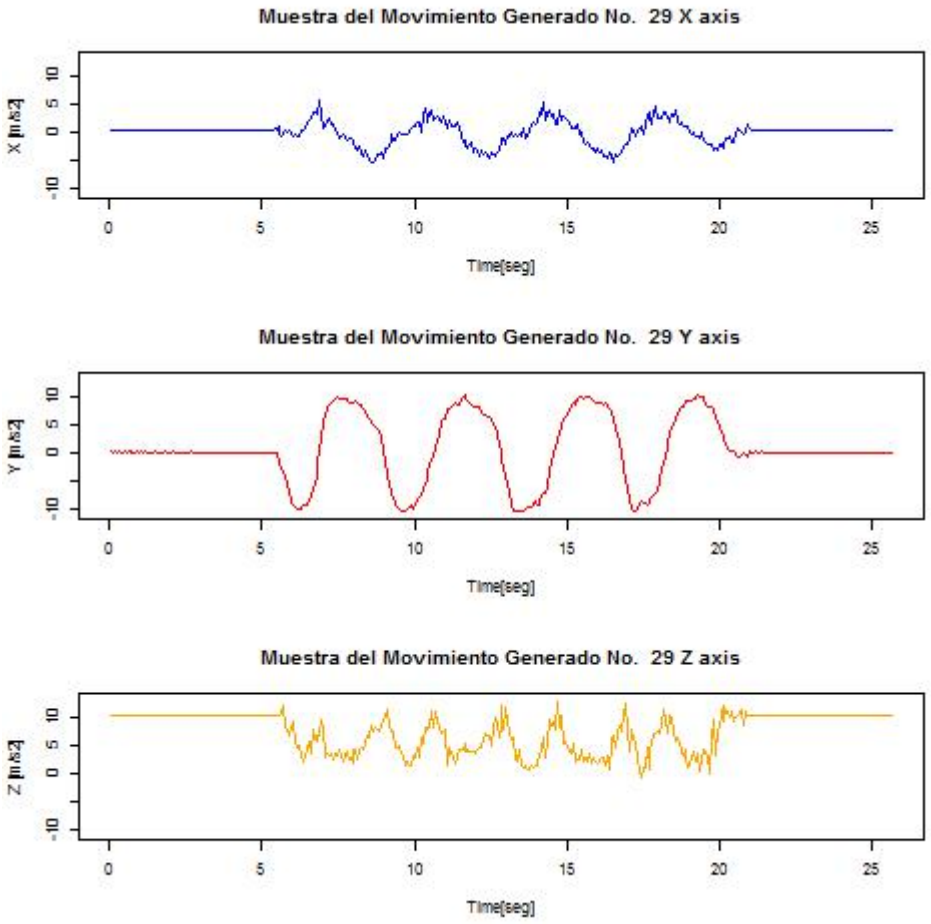
A.3.7 Muestra 7 del *tag* móvil



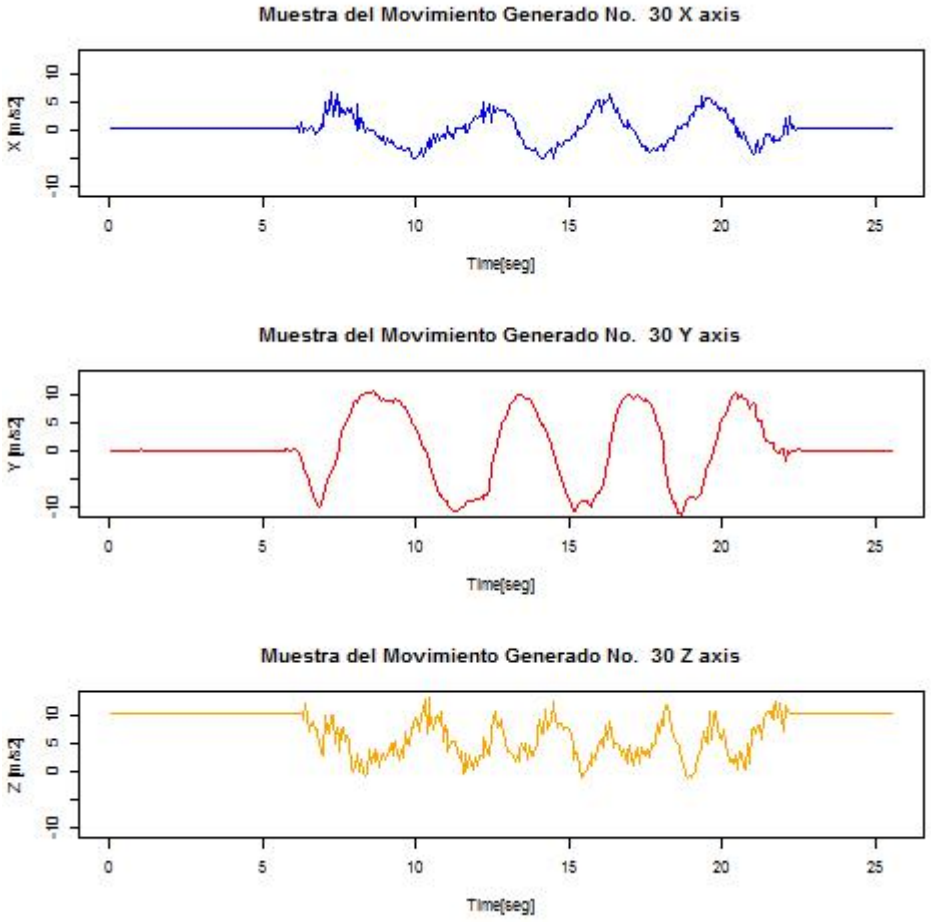
A.3.8 Muestra 8 del *tag* móvil



A.3.9 Muestra 9 del *tag* móvil



A.3.10 Muestra 10 del *tag* móvil



B. *Scripts* y Funciones en R

B.1 Programa Principal “main_TFG.r”

```
rm(list = ls())

setwd("C:/Users/JAYC/Desktop/TFG_R")

source ("Functions_TFG.R")


#######################################################################
############################################################

### Programa Principal ###


#######################################################################
############################################################

## Actualizacion version del RStudio

# installing/loading the package:

#options(repos = c(CRAN = "http://cran.rstudio.com"))

#if(!require(installr)) {

#  install.packages("installr");

#  require(installr)} #load / install+load installr

#updateR()


#######################################################################
############################################################

## Paquetes a Instalar

#options(repos = c(CRAN = "http://cran.rstudio.com"))

#install.packages('stringr')

#install.packages('proxy')

#install.packages('dtw')

#install.packages("RWeka", dependencies = TRUE)

#install.packages("rJava", dependencies = TRUE)
```

```
#install.packages('gWidgets')

#install.packages('tcltk')

########################################################################
############################################################

## Importamos las libreerias necesarias desde

location = paste(getwd(),'/librerias',sep="",collapse=NULL)

library("stringr", lib.loc=location)

library("proxy", lib.loc=location)

library("dtw", lib.loc=location)

library("class", lib.loc=location)

library("RWeka", lib.loc=location)

library("rJava", lib.loc=location)

library("grid", lib.loc=location)

library("splines", lib.loc=location)

library("gWidgets", lib.loc=location)

library("tcltk", lib.loc=location)



########################################################################
############################################################

### Variables Globales

go <- 9.80665 # m/s^2

num_muestras_farsens <- 0

num_muestras_movil <- 0

num_muestras_total <- 0



########################################################################
############################################################
```

```
#### Lectura de N muestras del tag KINEO FARSENS ####

#########################################################################
#################################################################

#########################################################################
#################################################################

### Calibramos para establecer threshold

print(as.character("Introduzca una muestra calibrada Farsens"))

path<-file.choose()

s_calibra<-
as.data.frame(read.delim(path,header=F,quote="",stringsAsFactors
=F,dec="."))

list_serie_calibra                                                    <-
lee_excel_farsens(s_calibra[2],s_calibra[3],s_calibra[4],s_calibra[5])

name1="Farsens"

name2="_muestra_"

name0="calibracion"

dibuja_tag(list_serie_calibra,name0,name1,name2,1)

list_threshold <- calibra(list_serie_calibra)

#########################################################################
#################################################################

location = paste(getwd(),'/BD_muestras/Farsens/',sep="",collapse=NULL)

path_directorio<- location

ruta_csv <- paste(path_directorio,'*.csv',sep="",collapse=NULL)

files <- list.files(path_directorio)

files_full<-paste(path_directorio,files,sep="")

listOfFiles         <-          lapply(files_full,          function(x)
read.delim(x,header=F,quote="",stringsAsFactors =F,dec="."))

### listado de N series

listado_n_x <- list()

listado_n_y <- list()

listado_n_z <- list()
```

```
### Serie filtrada

mov_filtrado_x <- list()

mov_filtrado_y <- list()

mov_filtrado_z <- list()

mov_filtrado_frecuency <- list()


### Calculamos todas las N series

print("Extraemos todas las muestras")

for (i in 1:length(listOfFiles)){


  muestra_frame <-as.data.frame(listOfFiles[i])

  ### Leemos datos del tag farsens

  serie                                                              <-
lee_excel_farsens(muestra_frame[2],muestra_frame[3],muestra_frame[4],m
uestra_frame[5])

  #features <- caracteriza(serie)

  list_series1                                                       <-
list(unlist(serie[1]),unlist(serie[2]),unlist(serie[3]),unlist(serie[4
]))

  ### Capturamos las imagenes de las muestras , para ello preparamos
los directorios

  if(i<6){

    name1="C"

  }else{

    if (i>5 && i<11){

      name1 = "R"

    }else{

      name1 = "W"

    }

  }
```

```
  name2="_muestra_"
  name0="Movimiento_Farsens"
  dibuja_tag(list_series1,name0,name1,name2,i)



  ### Filtramos los 3 ejes de las 2 senyales para quedarnos con el
movimiento
  filtrado_serie1 <- filtra(list_series1,list_threshold)


  mov_filtrado_x[i] <- list(unlist(filtrado_serie1[1]))
  mov_filtrado_y[i] <- list(unlist(filtrado_serie1[2]))
  mov_filtrado_z[i] <- list(unlist(filtrado_serie1[3]))
  mov_filtrado_frecuency[i]    <- list(unlist(filtrado_serie1[4]))


  ### Dibujamos todas las muestras filtradas
  name2="_filtrada_"
  name0="filtrado"
  filtrado                                                          <-
list(unlist(mov_filtrado_x[i]),unlist(mov_filtrado_y[i]),unlist(mov_fi
ltrado_z[i]),unlist(mov_filtrado_frecuency[i]))
  dibuja_tag(filtrado,name0,name1,name2,i)



  listado_n_x[i] <- list(unlist(mov_filtrado_x[i]))
  listado_n_y[i] <- list(unlist(mov_filtrado_y[i]))
  listado_n_z[i] <- list(unlist(mov_filtrado_z[i]))
```

```
}

#########################################################################
#################################################################

### Calibramos para establecer threshold

print(as.character("Introduzca una muestra calibrada del dispositivo
Movil"))

path<-file.choose()

s_calibra<-(read.delim(path,
header=F,skip=2,quote="",sep=';',stringsAsFactors =F,dec="."))

list_serie_calibra                                                    <-
lee_excel_movil(s_calibra[1],s_calibra[2],s_calibra[3],s_calibra[4])

list_threshold <- calibra(list_serie_calibra)

#########################################################################
#################################################################

#########################################################################
#################################################################

#### Lectura de N muestras del Movil ####

#########################################################################
#################################################################

num_muestras_farsens<-length(listado_n_x)

location = paste(getwd(),'/BD_muestras/Movil/',sep="",collapse=NULL)

path_directorio<- location

ruta_csv <- paste(path_directorio,'*.csv',sep="",collapse=NULL)

files <- list.files(path_directorio)

files_full<-paste(path_directorio,files,sep="")

listOfFiles         <-          lapply(files_full,          function(x)
read.delim(x,header=F,skip=2,quote="",sep=';',stringsAsFactors
=F,dec="."))
```

```
### Calculamos todas las N series y

### todas las caracterÃ-sticas para las N muestras x cada eje

for (i in 1:(length(listOfFiles))){


  muestra_frame <-as.data.frame(listOfFiles[i])

  ### Leemos datos del tag farsens

  serie                                                                <-
lee_excel_movil(muestra_frame[1],muestra_frame[2],muestra_frame[3],mue
stra_frame[4])



  list_series1                                                         <-
list(unlist(serie[1]),unlist(serie[2]),unlist(serie[3]),unlist(serie[4
]))



  j <- num_muestras_farsens + i



  ### Capturamos las imagenes de las muestras , para ello preparamos
los directorios

  if(j<21){

    name1="C"

  }else{

    if (j>20 && j<26){

      name1 = "R"

    }else{

      if(j > 25 && j < 31){

        name1 = "W"

      }


    }

  }
```

```
  name2="_muestra_"
  name0="Movimiento_Movil"
  dibuja_tag(list_series1,name0,name1,name2,j)



  ### Filtramos los 3 ejes de las 2 senyales para quedarnos con el
movimiento
  filtrado_serie1 <- filtra(list_series1,list_threshold)


  mov_filtrado_x[j] <- list(unlist(filtrado_serie1[1]))
  mov_filtrado_y[j] <- list(unlist(filtrado_serie1[2]))
  mov_filtrado_z[j] <- list(unlist(filtrado_serie1[3]))
  mov_filtrado_frecuency[j]    <- list(unlist(filtrado_serie1[4]))


  ### Dibujamos todas las muestras filtradas
  name2="_filtrada_"
  name0="filtrado"
  filtrado                                                          <-
list(unlist(mov_filtrado_x[j]),unlist(mov_filtrado_y[j]),unlist(mov_fi
ltrado_z[j]),unlist(mov_filtrado_frecuency[j]))
  dibuja_tag(filtrado,name0,name1,name2,i)



  listado_n_x[j] <- list(unlist(mov_filtrado_x[j]))
  listado_n_y[j] <- list(unlist(mov_filtrado_y[j]))
  listado_n_z[j] <- list(unlist(mov_filtrado_z[j]))
```

```
}

num_muestras_movil <- i

num_muestras_total <- length(listado_n_x)


#listado_dtw_x     <-array(data.frame(),     dim=c(length(listado_n_x),
length(listado_n_x)))

#listado_dtw_y     <-array(data.frame(),     dim=c(length(listado_n_y),
length(listado_n_y)))

#listado_dtw_z     <-array(data.frame(),     dim=c(length(listado_n_z),
length(listado_n_z)))


listado_dtw_x <- list()

listado_dtw_y <- list()

listado_dtw_z <- list()


dtw_xyz<- data.frame(row.names=NULL)

dtw_x <-  data.frame(row.names=NULL)

dtw_y <-  data.frame(row.names=NULL)

dtw_z <-  data.frame(row.names=NULL)


for (i in 1:num_muestras_total){


  for (j in 1:num_muestras_total){


    listado_dtw_x[j] <- similitud_dtw(listado_n_x[i],listado_n_x[j])
```

```
    listado_dtw_y[j] <- similitud_dtw(listado_n_y[i],listado_n_y[j])
    listado_dtw_z[j] <- similitud_dtw(listado_n_z[i],listado_n_z[j])


  }
  dtw_x <- rbind(dtw_x,unlist(listado_dtw_x))
  dtw_y <- rbind(dtw_y,unlist(listado_dtw_y))
  dtw_z <- rbind(dtw_z,unlist(listado_dtw_z))
}

names<- c()
for (i in 1:30){
  names[i] <- paste("V",i,sep="")
}
colnames(dtw_x)<-names
colnames(dtw_y)<-names
colnames(dtw_z)<-names
write.csv2(dtw_x, file = "tables/DTW_x.csv",row.names=FALSE,na="")
write.csv2(dtw_y, file = "tables/DTW_y.csv",row.names=FALSE,na="")
write.csv2(dtw_z, file = "tables/DTW_z.csv",row.names=FALSE,na="")
names<- c()

for (i in 1:90){
  names[i] <- paste("V",i,sep="")
}

dtw_xyz<-cbind(dtw_x,dtw_y,dtw_z)
colnames(dtw_xyz)<-names
```

```
write.csv2(dtw_xyz, file = "tables/DTW_xyz.csv",row.names=FALSE,na="")
```

```
#########################################################################
#########################################################################
# CONSOLA DE MANDOS
#########################################################################
#########################################################################

 print('Introduzca una prueba: ')
 print ('Escriba: Farsens o Movil ')
 test <- readline()

 KNN<-KNN(test,10)

 kfold_cv<-KNN[1]
 decision<-KNN[2]
 knn_cv <- KNN[3]


kfold_cv
knn_cv
decision
```

B.2 Función Lectura CSV *Tag* Farsens “lee_excel_farsens”

```
#Lectura del tag Farsens

lee_excel_farsens <- function(timestamp,x,y,z) {

  #### Calculamos los timestamps de cada dato recojido
  #### y los guardamos en un vector de strings "timestamp"

  V2 <- as.character(timestamp$V2)
  #####Formateo de los datos del accelerometro ##################
  ##### X axis values ###############
  X <- as.numeric(str_replace_all((as.character(x$V3)),",","."))
  ##### Y axis values ###############
  Y <- as.numeric(str_replace_all((as.character(y$V4)),",","."))
  ##### Z axis values ###############
  z1 <- str_replace_all((as.character(z$V5)),"\"","")
  Z <- as.numeric(str_replace_all(z1,",","."))
  ################################################################



  DATA_TIME <- c(1:(length(V2)))
  DATA_TIME <- V2
  fecha <- substring(DATA_TIME,1,10)
  timestamp <- substring(DATA_TIME,12,23)
  #string
  s_timestamp_hh  <- (substring(timestamp,1,2))
  s_timestamp_mm  <- (substring(timestamp,4,5))
```

```
  s_timestamp_ss  <- (substring(timestamp,7,8))

  s_timestamp_ddd <- (substring(timestamp,10,13))

  #numericos

  n_timestamp_hh  <- as.numeric(s_timestamp_hh)

  n_timestamp_mm  <- as.numeric(s_timestamp_mm)

  n_timestamp_ss  <- as.numeric(s_timestamp_ss)

  n_timestamp_ddd <- as.numeric(s_timestamp_ddd)

  ### Creamos vector de timestamps numericos

  ts_vector <- c(1:length(V2))

  ts_vector                                                              <-
as.numeric(paste(s_timestamp_hh,s_timestamp_mm,s_timestamp_ss,s_timest
amp_ddd,sep=""))

#####################################################################

##### Calculamos tiempo/frecuencia de mostreo #####

  t_total               =               ((n_timestamp_hh[length(V2)]-
n_timestamp_hh[1])*3600)+((n_timestamp_mm[length(V2)]-
n_timestamp_mm[1])*60)+((n_timestamp_ss[length(V2)]-
n_timestamp_ss[1]))+((n_timestamp_ddd[length(V2)]-
n_timestamp_ddd[1])/100)

  f_mostreig = length(V2)/t_total

  t_mostreig = 1/ f_mostreig

  ###################################################################




#### Calculamos serie temporal en función de los datos ######

#### y de la frecuencia de mostreo Time [s] f[hz]

x <- ts(X, start = c(t_mostreig), frequency = f_mostreig)*go

y <- ts(Y, start = c(t_mostreig), frequency = f_mostreig)*go

z <- ts(Z, start = c(t_mostreig), frequency = f_mostreig)*go

#####################################################################

series <- list(x,y,z,f_mostreig)
```

```
return(series)


}
```

B.3 Función Lectura CSV Móvil “lee_excel_movil”

```
####Lectura documento csv generado por el dispositivo movil


lee_excel_movil <- function() {
  ####Importamos libreria stringr para el tratamiento de cadenas
  KINEO <- read.delim(file.choose(), header=F,skip=2,quote="",sep=';')
  attach(KINEO)


  # ...leyendo
  ######################################################################




  #####Formateo de los datos del giroscopio ##################
  ##### X axis values ###############
  X <- as.numeric(str_replace_all(as.character(V2),",","."))
  ##### Y axis values ###############
  Y <- as.numeric(str_replace_all((as.character(V3)),",","."))
  ##### Z axis values ###############
  Z <- as.numeric(str_replace_all(as.character(V4),",","."))
  ######Time                                                    values
############################################################
  tiempo <- as.numeric(str_replace_all(as.character(V1),",","."))


  ##### Calculamos tiempo/frecuencia de mostreo #####
  t_total <- (tiempo[length(tiempo)]) - (tiempo[1])
```

```
  f_mostreig = length(X)/t_total
  t_mostreig = 1/ f_mostreig
  ##################################################################
  #### Calculamos serie temporal en función de los datos ######
  #### y de la frecuencia de mostreo Time [s] f[hz]
  x <- ts(X, start = c(t_mostreig), frequency = f_mostreig)
  y <- ts(Y, start = c(t_mostreig), frequency = f_mostreig)
  z <- ts(Z, start = c(t_mostreig), frequency = f_mostreig)
  ##################################################################

  series <- list(x,y,z,f_mostreig)
  return(series)

}
```

B.4 Función gráfica aceleración 3 axis “dibuja_tag”

```
###         FUNCION         PARA         GRAFICAR         EL         TAG
##########################################

### Introducimos una lista de 3 objetos correspondiente a X,Y,Z y los
representamos

dibuja_tag <- function(list_serie1,name0,name1,name2,num){
  x <- (unlist(list_serie1[1])) # m/s2
  y <- (unlist(list_serie1[2])) # m/s2
  z <- (unlist(list_serie1[3])) # m/s2
  f_mostreig <- unlist(list_serie1[4]) # Hz
  ymax <- round(as.numeric(max(x,y,z)))
  ymin <-round(as.numeric(min(x,y,z)))
  ### Mostrar 3 ejes en [m/s2] por cada muestra
```

```
  #plot.ts(x,ylim=range(ymax:ymin),xlab    =    "Muestra",ylab=    "X
[rad/s]",col = "blue")

  #plot.ts(y,ylim=range(ymax:ymin),xlab    =    "Muestra",ylab=    "Y
[rad/s]",col = "red")

  #plot.ts(z,ylim=range(ymax:ymin),xlab    =    "Muestra",ylab=    "Z
[rad/s]",col = "yellow")


  ### Mostrar 3 ejes en [m/s2] cada [seg]

  #### Calculamos serie temporal en función de los datos ######

  #### y de la frecuencia de mostreo Time [s] f[hz]

  x <- ts(x, start = c(1/f_mostreig), frequency = f_mostreig)

  y <- ts(y, start = c(1/f_mostreig), frequency = f_mostreig)

  z <- ts(z, start = c(1/f_mostreig), frequency = f_mostreig)

  mypath   <-   file.path(getwd(),"images",name0,paste(name1,name2,num,
".jpg", sep = ""))

  jpeg(file=mypath)

  mytitle = paste("Muestra del Movimiento Generado No. ",num)

  ######MOSTRAMOS LAS GRAFICAS DE LOS EJES DEL GIROSCOPIO#######

  attach(mtcars)

  par(mfrow=c(3,1))

  plot(as.vector(time(x)),as.vector(x),ylim=range(ymax:ymin),xlab    =
"Time[seg]",ylab=      "X      [m/s2]",col      =      "blue",type='l'
,main=paste(mytitle,"X axis"))

  plot(as.vector(time(y)),as.vector(y),ylim=range(ymax:ymin),xlab    =
"Time[seg]",ylab=             "Y             [m/s2]",col             =
"red",type='l',main=paste(mytitle,"Y axis"))

  plot(as.vector(time(z)),as.vector(z),ylim=range(ymax:ymin),xlab    =
"Time[seg]",ylab=             "Z             [m/s2]",col             =
"orange",type='l',main=paste(mytitle,"Z axis"))

  dev.off()

#########################################################################
##############

}”
```

B.5 Función “calibra”

```
calibra <- function(list_serie_calibra) {
  x<-unlist(list_serie_calibra[1])
  y<-unlist(list_serie_calibra[2])
  z<-unlist(list_serie_calibra[3])-go
  f_mostreig <- list_serie_calibra[4]
 ### Calculamos la derivada para cada uno de los ejes
  xt1 <-  smooth.spline(x)
  yt1 <-  smooth.spline(y)
  zt1 <-  smooth.spline(z)
  threshold_max <- max(xt1$y,yt1$y,zt1$y)
  threshold_min <- min(xt1$y,yt1$y,zt1$y)
  threshold <- list(threshold_max,threshold_min)
  return (threshold)

}
```

B.6 Función “filtra”

```
filtra <- function(list_series1,list_threshold) {
  threshold_max <- unlist(list_threshold[1])
  threshold_min <- unlist(list_threshold[2])
  xt1<-unlist(list_series1[1])
  yt1<-unlist(list_series1[2])
  zt1<-unlist(list_series1[3])
  f_mostreig <-unlist(list_series1[4])
```

```
  ##EJE X

  ### Para cada eje establecemos un threshold y damos valor 1 o 0

  vector_threshold<-ifelse((xt1    >=    threshold_max    |    xt1    <=
threshold_min), xt1, 0)

  ### Buscamos la primera ocurrencia que cumple el threshold

  index_ocurrence <- which(vector_threshold !=0)

  x_first_ocurrence <- index_ocurrence[1]

  ### Buscamos la segunda ocurrencia que cumple el threshold

  x_last_ocurrence <- index_ocurrence[length(index_ocurrence)]

  ### Para cada eje establecemos un threshold y damos valor 1 o 0

  vector_threshold<-ifelse((yt1    >=    threshold_max    |    yt1    <=
threshold_min), yt1, 0)

  ### Buscamos la primera ocurrencia que cumple el threshold

  index_ocurrence <- which(vector_threshold !=0)

  y_first_ocurrence <- index_ocurrence[1]

  ### Buscamos la segunda ocurrencia que cumple el threshold

  y_last_ocurrence <- index_ocurrence[length(index_ocurrence)]

  ### Para cada eje establecemos un threshold y damos valor 1 o 0

  vector_threshold<-ifelse(((zt1-go)  >=  threshold_max  |  (zt1-go)  <=
threshold_min), zt1, 0)

  ### Buscamos la primera ocurrencia que cumple el threshold

  index_ocurrence <- which(vector_threshold !=0)

  z_first_ocurrence <- index_ocurrence[1]

  ### Buscamos la segunda ocurrencia que cumple el threshold

  z_last_ocurrence <- index_ocurrence[length(index_ocurrence)]


  first_ocurrence                                                    <-
as.numeric(min(na.omit(x_first_ocurrence),na.omit(y_first_ocurrence),n
a.omit(z_first_ocurrence)))
```

```
  last_ocurrence                                                    <-
as.numeric(max(na.omit(x_last_ocurrence),na.omit(y_last_ocurrence),na.
omit(z_last_ocurrence)))



  ##EJE X


  ### Creamos un vector (vector_movement1) con todos los valores
comprendidos

  ### desde la primera ocurrencia hasta la Ãºltima y el resto a nulos.

  condition             <-             first_ocurrence             <=
(which(vector_threshold==vector_threshold             ))             &
(which(vector_threshold==vector_threshold )<=last_ocurrence)

  vector_movement1 <- ifelse(condition,xt1,NaN)

  ### Seleccionamos el subconjunto que pertenece a los valores del
movimiento

  x1<- subset(xt1,condition,select=xt1)




  ##EJE Y


  ### Creamos un vector (vector_movement1) con todos los valores
comprendidos

  ### desde la primera ocurrencia hasta la Ãºltima y el resto a nulos.

  condition             <-             first_ocurrence             <=
(which(vector_threshold==vector_threshold             ))             &
(which(vector_threshold==vector_threshold )<=last_ocurrence)

  vector_movement2 <- ifelse(condition,yt1,NaN)

  ### Seleccionamos el subconjunto que pertenece a los valores del
movimiento

  y1<- subset(yt1,condition,select=yt1)
```

```
  ##EJE Z


  ### Creamos un vector (vector_movement1) con todos los valores
comprendidos

  ### desde la primera ocurrencia hasta la Ãºltima y el resto a nulos.

  condition             <-             first_ocurrence             <=
(which(vector_threshold==vector_threshold             ))            &
(which(vector_threshold==vector_threshold )<=last_ocurrence)

  vector_movement3 <- ifelse(condition,zt1,NaN)

  ### Seleccionamos el subconjunto que pertenece a los valores del
movimiento

  z1<- subset(zt1,condition,select=zt1)



  serie_movimiento <- list(x1,y1,z1,f_mostreig)

  #serie_movimiento                                                  <-
list(vector_movement1,vector_movement2,vector_movement3)

  ### Dibujamos el subconjunto de la senyal que pertenece al
movimiento en los

  ### tres ejes (X1,Y1,Z1)

  #dibuja_tag(serie_movimiento)




  return (serie_movimiento)


}
```

B.7 Función “similitud_dtw”

```
similitud_dtw <- function(list_series1,list_series2) {

  ### Calculamos la funcion de similitud o warping function DTW

  alignment                                                              <-
dtw(unlist(list_series1[1]),unlist(list_series2[1]),k=TRUE)

  distance_normalized <- as.numeric(alignment$normalizedDistance)

  return (distance_normalized)



}
```

B.8 Función “KNN”

```
KNN <- function(test,k) {



  training_x <-as.matrix(listado_dtw_x)

  training_y <-as.matrix(listado_dtw_y)

  training_z <-as.matrix(listado_dtw_z)



  # The object to be classified

  if(test=='Farsens'){

  print('Seleccionamos datos generados por el tag Farsens')

########################################################################
#################################################################

  location =
paste(getwd(),'/muestra_pruebas/Farsens/',sep="",collapse=NULL)

  path_directorio<- location

  ruta_csv <- paste(path_directorio,'*.csv',sep="",collapse=NULL)

  files <- list.files(path_directorio)

  files_full<-paste(path_directorio,files,sep="")

  listOfFiles <- lapply(files_full, function(x)
read.delim(x,header=F,quote="",stringsAsFactors =F,dec="."))
```

```
  x_test <- list()
  y_test <- list()
  z_test <- list()
  f_test <- list()



  test_read <-as.data.frame(listOfFiles[1])
  test<-
lee_excel_farsens(test_read[2],test_read[3],test_read[4],test_read[5])
  name1="test"
  name2="_movimiento_"
  name0="test"
  dibuja_tag(test,name0,name1,name2,1)


  list_test_filtered <- filtra(test,list_threshold)
  x_test<-(list_test_filtered [1])
  y_test<-(list_test_filtered [2])
  z_test<-(list_test_filtered [3])
  f_test<-(test[4])


  name1="test"
  name2="_filtrado_"
  name0="test"
  dibuja_tag(list_test_filtered,name0,name1,name2,1)







  }else{
    if (test == 'Movil'){
    print('Seleccionamos datos generados por el movil')

######################################################################
#############################################################
```

```
    location =
paste(getwd(),'/muestra_pruebas/Movil/',sep="",collapse=NULL)

    path_directorio<- location

    ruta_csv <- paste(path_directorio,'*.csv',sep="",collapse=NULL)

    files <- list.files(path_directorio)

    files_full<-paste(path_directorio,files,sep="")

    listOfFiles <- lapply(files_full, function(x)
read.delim(x,header=F,skip=2,quote="",sep=';',stringsAsFactors
=F,dec="."))

    x_test <- list(length(listOfFiles))

    y_test <- list(length(listOfFiles))

    z_test <- list(length(listOfFiles))

    f_test <- list(length(listOfFiles))




    test_read <-as.data.frame(listOfFiles[1])

    test<-
lee_excel_movil(test_read[1],test_read[2],test_read[3],test_read[4])

    name1="test"

    name2="_movimiento_"

    name0="test"

    dibuja_tag(test,name0,name1,name2,1)

    list_test_filtered <- filtra(test,list_threshold)

    x_test<-(list_test_filtered [1])

    y_test<-(list_test_filtered [2])

    z_test<-(list_test_filtered [3])

    f_test<-(test[4])

    name1="test"

    name2="_filtrado_"

    name0="test"

    dibuja_tag(list_test_filtered,name0,name1,name2,1)
```

```
    }else{
      print('Opcion Incorrecta')
      print('Introduzca una prueba: ')
      print ('Escriba: Farsens o Movil ')
      string<- readline()
      KNN(string)}
  }



  test_dtw_x<-c(length(listado_n_x))
  test_dtw_y<-c(length(listado_n_y))
  test_dtw_z<-c(length(listado_n_z))


  for (j in 1:length(listado_n_x)){


    test_dtw_x[j] <-
as.vector(similitud_dtw(x_test[1],listado_n_x[j]))
    test_dtw_y[j] <-
as.vector(similitud_dtw(y_test[1],listado_n_y[j]))
    test_dtw_z[j] <-
as.vector(similitud_dtw(z_test[1],listado_n_z[j]))


  }



  c1<-cbind(dtw_x[1:5,1:30],dtw_y[1:5,1:30],dtw_z[1:5,1:30])
  r1<-cbind(dtw_x[6:10,1:30],dtw_y[6:10,1:30],dtw_z[6:10,1:30])
  w1<-cbind(dtw_x[11:15,1:30],dtw_y[11:15,1:30],dtw_z[11:15,1:30])
  c2<-cbind(dtw_x[16:20,1:30],dtw_y[16:20,1:30],dtw_z[16:20,1:30])
  r2<-cbind(dtw_x[21:25,1:30],dtw_y[21:25,1:30],dtw_z[21:25,1:30])
  w2<-cbind(dtw_x[26:30,1:30],dtw_y[26:30,1:30],dtw_z[26:30,1:30])
```

```
  c <- rbind(c1,c2)

  r <- rbind(r1,r2)

  w <- rbind(w1,w2)


  train <- rbind(c,r,w)

  names<- c()

  for (i in 1:90){

    names[i] <- as.list(paste("V",i,sep=""))

  }

  colnames(train)<-names


  cl <- factor(c(rep("C",10), rep("R",10), rep("W",10)))


  tests <-
c(test_dtw_x[1:(length(test_dtw_x))],test_dtw_y[1:(length(test_dtw_y))
],test_dtw_z[1:(length(test_dtw_z))])


  classifier <- IBk(cl~., data = train,

                    control = Weka_control(K = 20, X = TRUE))


  evaluation<-evaluate_Weka_classifier(classifier, numFolds =
10,class=TRUE)




  decisor <-knn(train, tests, cl, k = k, prob=TRUE,use.all=TRUE)

  attributes(.Last.value)


  knn_cv<-knn.cv(train, cl, k = k, prob = TRUE)







 return (list(evaluation,decisor,knn_cv))

}
```

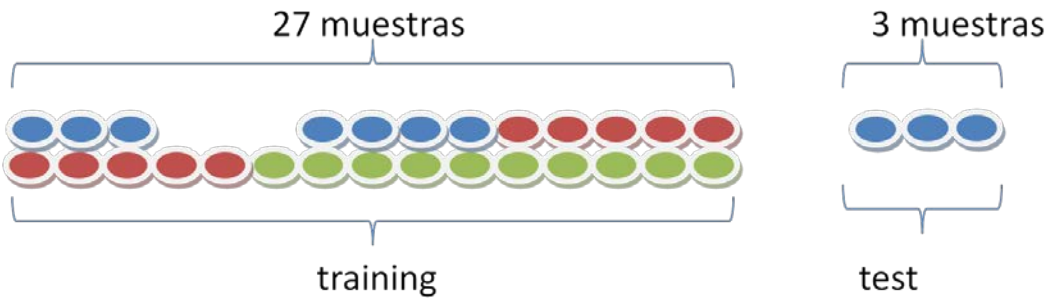
C. Contenido del CD

Carpetas	Subcarpeta I	Subcarpeta II	Descripción
TFG_R	**BD_muestras**	Farsens	Conjunto de 15 muestras correspondientes a las capturas de gestos realizados por el *tag* Farsens en formato .csv.
		Movil	Conjunto de 15 muestras correspondientes a las capturas de gestos realizados por el dispositivo móvil en formato .csv.
	Images	filtrado	Todas las gráficas que contienen los gestos con los filtros aplicados de las 30 muestras.
		Movimiento_Farsens	Conjunto de 15 gráficas correspondientes a las capturas de gestos sin filtro realizados por el *tag* Farsens
		Movimiento_movil	Conjunto de 15 gráficas correspondientes a las capturas de gestos sin filtro realizados por el dispositivo móvil
		test	Contiene las gráficas del gesto filtrado y sin filtrar correspondientes a la muestra a testear
	librerias	dtw	Librería que contiene todos los métodos para el cálculo del algoritmo DTW
		proxy	Librería requerida por la librería DTW
		stringr	Librería que contiene herramientas para el tratamiento de cadenas de caracteres *stringr*
		class	Librería que contiene un conjunto de herramientas *machine learning como knn y knn.cv*
		rJava	Librería que sirve de *interface* entre R y Java VM que permite acceder a objetos y llamadas a métodos del lenguaje Java
	muestra_calibracion	Calibracion_Farsens	Contiene la muestra de

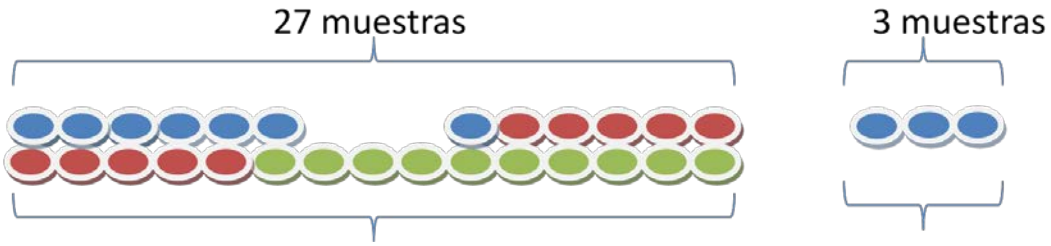
			calibración generada por el *tag* Farsens utilizada para filtrar los gestos en formato .csv.
		Calibracion_Movil	Contiene la muestra de calibración generada por el dispositivo móvil utilizada para filtrar los gestos en formato .csv.
	muestra_pruebas	Farsens	Contiene la muestra a probar generada por el *tag* Farsens en formato .csv.
		Movil	Contiene la muestra a probar generada por el dispositivo móvil en formato .csv.
	tables		Contiene los Excels con todos los cálculos de las distancias de DTW de todo el conjunto de muestras

D. Resto de iteraciones *10-Fold-Cross-Validation*

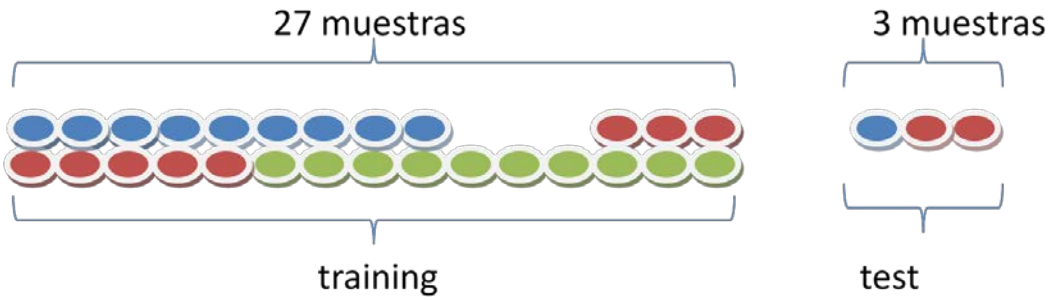
D.1 2ª Iteración



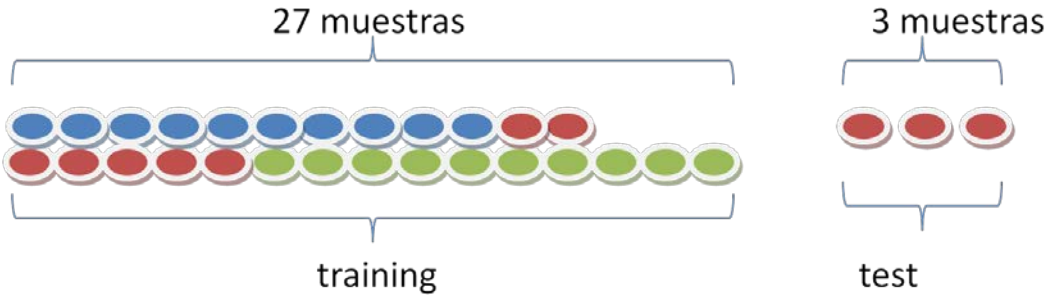
D.2 3ª Iteración



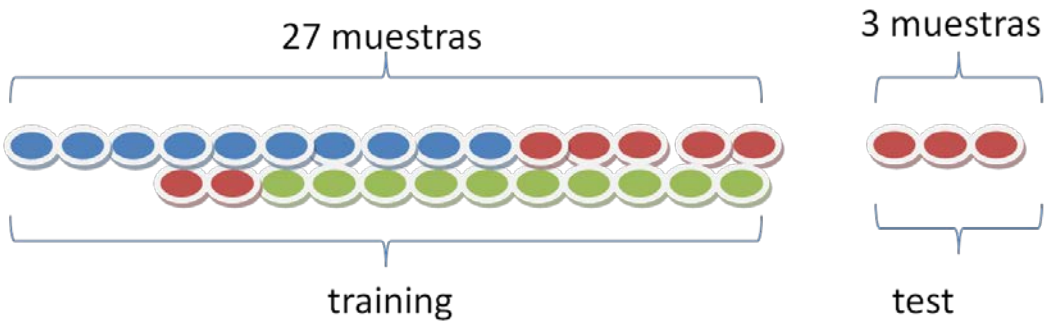
D.3 4ª Iteración



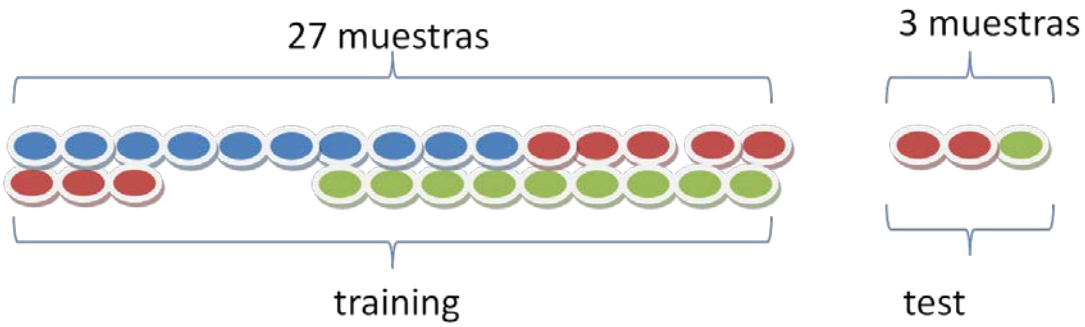
D.4 5ª Iteración



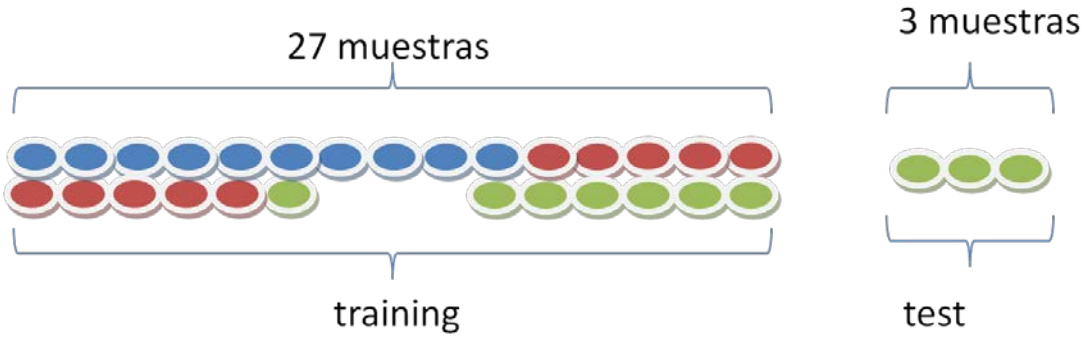
D.5 6ª Iteración



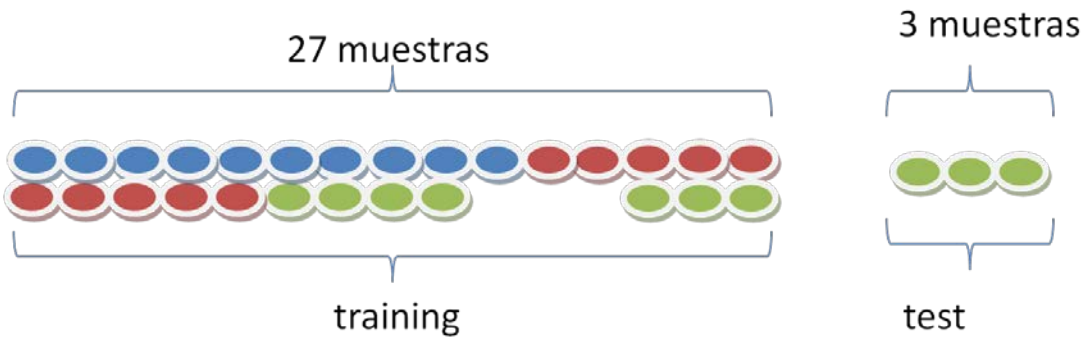
D.6 7ª Iteración



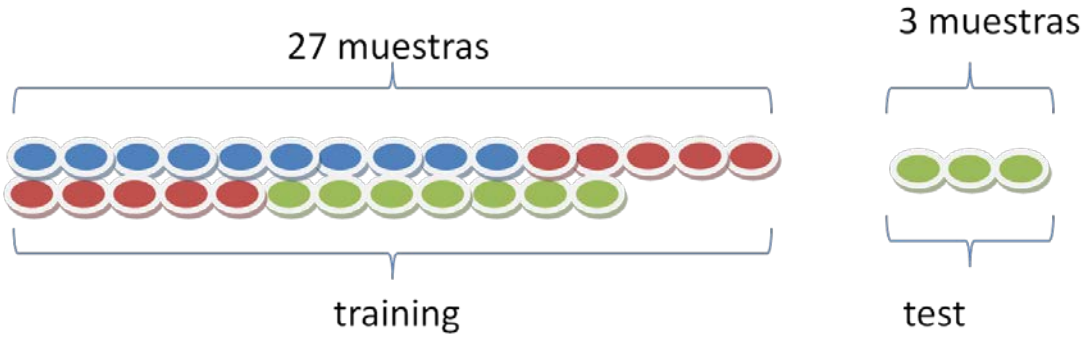
D.7 8ª Iteración



D.8 9ª Iteración



D.9 10ª Iteración



E. Resto de pruebas

Se realizan un conjunto de 10 gestos de *test*. Estos gestos no pertenecen al conjunto de datos entrenados *training set*. Cada gesto es independiente, es decir, ha sido realizado en instantes de tiempo diferentes. En la siguiente tabla se muestra el resultado de la clasificación para los gestos realizados, ya sean por el *tag* Farsens como por el dispositivo móvil. La clasificación se realiza para diferentes valores de k (nº de vecinos), indicando la probabilidad del gesto clasificado.

			Clasificación					
			k=10		k=8		k=5	
#	Gesto test realizado	Generador	Gesto	Probabilidad	Gesto	Probabilidad	Gesto	Probabilidad
1	"W"	tag Farsens	W	06	W	0.625	W	1
2	"W"	móvil	W	0.8	W	1	W	1
3	"R"	tag Farsens	C	0.6	R	0.5	R	0.8
4	"R"	móvil	R	0.6	R	0.625	R	1
5	"C"	tag Farsens	C	0.5	C	0.625	C	1
6	"C"	móvil	C	0.9	C	0.875	C	1
7	“C”	tag Farsens	C	0.5	C	0.625	C	1
8	“R”	móvil	R	0.6	R	0.625	R	1
9	“R”	tag Farsens	C	0.6	C	0.5	R	0.8
10	“W”	móvil	W	0.7	W	0.875	W	1

F. Resto de resultados K-Fold Cross Validation

F.1 K = 30

K=30	muestra in	dispositivo	k=10	Prob	k=8	Prob	k=5	Prob	k=2	Prob
1	C_muestra_1_2014_05_13.csv	farsens	C	0,7	C	0.75	C	0.8	C	1
2	C_muestra_2_2014_05_13.csv	farsens	C	0,6	C	0.75	C	0.8	C	1
3	C_muestra_3_2014_05_13.csv	farsens	C	0.5	C	0.5	C	0.8	C	1
4	C_muestra_4_2014_05_13.csv	farsens	C	0,6	C	0.75	C	0.8	C	1
5	C_muestra_5_2014_05_13.csv	farsens	C	0.8	C	0.75	C	0.8	C	1
6	c_01_mov_acc.csv	movil	C	0.6	C	0.625	C	1	C	1
7	c_02_mov_acc.csv	movil	C	0.8	C	0.875	C	0.8	C	1
8	c_03_mov_acc.csv	movil	C	0.8	C	0.875	C	1	C	1
9	c_04_mov_acc.csv	movil	C	0.9	C	0.875	C	1	C	1
10	c_05_mov_acc.csv	movil	C	0.8	C	0.750	C	1	C	1
11	R_muestra_1_2014_05_08.csv	farsens	C	0.7	C	0.625	R	0.6	R	1
12	R_muestra_2_2014_05_08.csv	farsens	C	0.7	C	0.625	R	0.6	R	1
13	R_muestra_3_2014_05_08.csv	farsens	C	0.7	C	0.625	R	0.6	R	1
14	R_muestra_4_2014_05_08.csv	farsens	C	0.6	C	0.5	R	0.8	R	1
15	R_muestra_5_2014_05_08.csv	farsens	C	0.6	C	0.625	R	0.6	R	1
16	r_01_mov_acc.csv	movil	R	0.5	R	0.625	R	0.8	R	1
17	r_02_mov_acc.csv	movil	C	0.5	R	0.625	R	0.8	R	1
18	r_03_mov_acc.csv	movil	R	0.5	R	0.625	R	0.8	R	1
19	r_04_mov_acc.csv	movil	R	0.5	R	0.625	R	0.8	R	1
20	r_05_mov_acc.csv	movil	R	0.5	R	0.625	R	0.8	R	1
21	W_muestra_1_2014_05_08.csv	farsens	W	0.5	C	0.5	W	0.8	W	1
22	W_muestra_2_2014_05_08.csv	farsens	W	0.7	W	0.75	W	0.8	W	1
23	W_muestra_3_2014_05_08.csv	farsens	W	0.7	W	0.625	W	0.8	W	1
24	W_muestra_4_2014_05_08.csv	farsens	W	0.8	W	0.750	W	0.8	W	1
25	W_muestra_5_2014_05_08.csv	farsens	W	0.9	W	1	W	0.8	W	1
26	w_01_mov_acc.csv	movil	W	0.7	W	0.875	W	1	W	1
27	w_02_mov_acc.csv	movil	W	0.7	W	0.875	W	1	W	1
28	w_03_mov_acc.csv	movil	W	0.7	W	0.875	W	1	W	1
29	w_04_mov_acc.csv	movil	W	0.7	W	0.875	W	1	W	1
30	w_05_mov_acc.csv	movil	W	0.7	W	0.875	W	1	W	1

F.1.1 k = 10

k=10		#	Precision	Recall	F-Score	Acc
"C"	TP	10	55,6%	100,0%	71,4%	73,3%
	TN	12				
	FP	8				
	FN	0				
"R"	TP	3	100,0%	30,0%	46,2%	76,7%
	TN	20				
	FP	0				
	FN	7				
"W"	TP	9	100,0%	90,0%	94,7%	96,7%
	TN	20				
	FP	0				
	FN	1				
			85,2%	73,3%	70,8%	82,2%

F. 1.2 k = 8

k=8		#	Precision	Recall	F-Score	Acc
"C"	TP	10	62,5%	100,0%	77%	80%
	TN	14				
	FP	6				
	FN	0				
"R"	TP	5	100,0%	50,0%	67%	83%
	TN	20				
	FP	0				
	FN	5				
"W"	TP	9	100,0%	90,0%	95%	97%
	TN	20				
	FP	0				
	FN	1				
			87,5%	80,0%	79,4%	86,7%

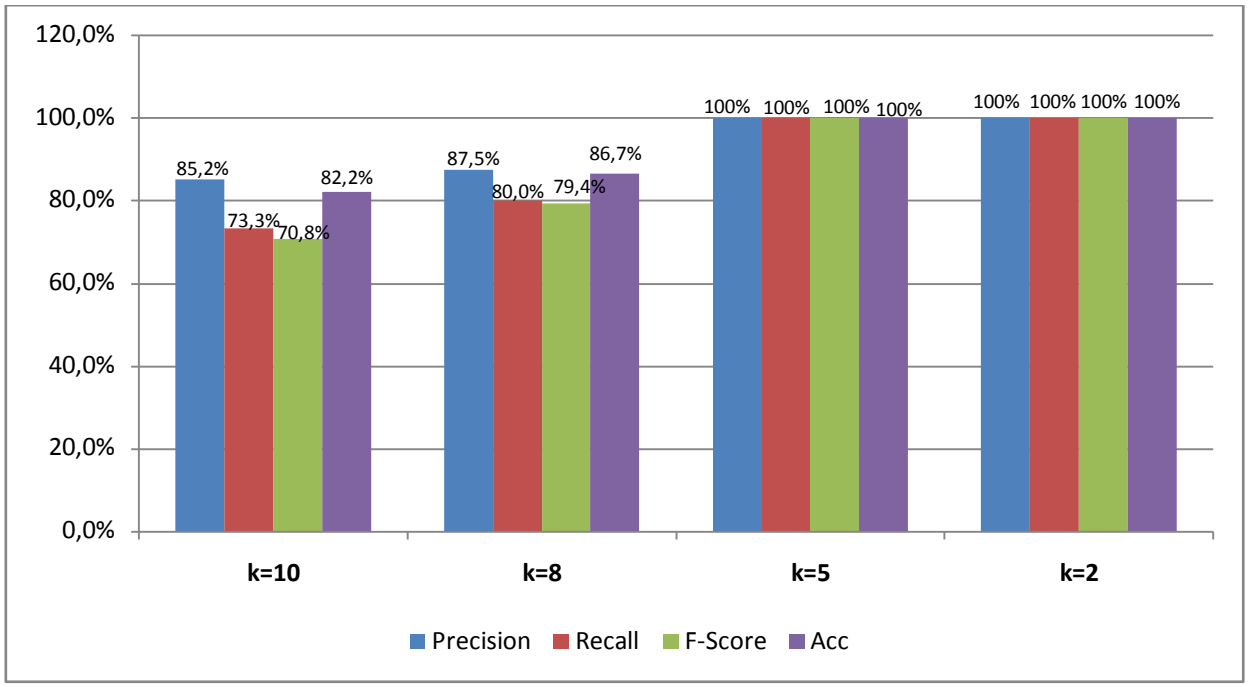
F. 1.3 k = 5

k=5		#	Precision	Recall	F-Score	Acc
"C"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
"R"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
"W"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
			100,0%	100,0%	100,0%	100,0%

F. 1.4 k = 2

k=2		#	Precision	Recall	F-Score	Acc
"C"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
"R"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
"W"	TP	10	100,0%	100,0%	100%	100%
	TN	0				
	FP	0				
	FN	0				
			100,0%	100,0%	100,0%	100,0%

F.1.5 Porcentaje métricas de evaluación K=30



F.2 K = 10

K=10	muestra in	dispositivo	k=10	Prob	k=8	Prob	k=5	Prob	k=2	Prob
1	C_muestra_1_2014_05_13.csv	farsens	C	0,6	C	0.625	C	0.8	C	1
2	C_muestra_2_2014_05_13.csv	farsens	C	0,6	C	0.625	C	0.8	C	1
3	C_muestra_3_2014_05_13.csv	farsens	C	0.5	C	0.625	C	0.8	C	1
4	C_muestra_4_2014_05_13.csv	farsens	C	0,7	C	0.75	C	0.8	C	1
5	C_muestra_5_2014_05_13.csv	farsens	C	0.7	C	0.75	C	0.8	C	1
6	c_01_mov_acc.csv	movil	C	0.5	C	0.625	C	0.8	C	1
7	c_02_mov_acc.csv	movil	C	0.7	C	0.75	C	0.8	C	1
8	c_03_mov_acc.csv	movil	C	0.6	C	0.75	C	0.8	C	1
9	c_04_mov_acc.csv	movil	C	0.7	C	0.875	C	0.8	C	1
10	c_05_mov_acc.csv	movil	C	0.8	C	1	C	1	C	1
11	R_muestra_1_2014_05_08.csv	farsens	C	0.8	C	0.75	C	0.8	C	1
12	R_muestra_2_2014_05_08.csv	farsens	C	0.7	C	0.75	C	0.6	R	1
13	R_muestra_3_2014_05_08.csv	farsens	C	0.8	C	0.75	C	0.6	R	1
14	R_muestra_4_2014_05_08.csv	farsens	C	0.8	C	0.75	C	0.6	R	1
15	R_muestra_5_2014_05_08.csv	farsens	C	0.8	C	0.75	C	0.6	R	1
16	r_01_mov_acc.csv	movil	C	0.6	R	0.5	R	0.6	R	1
17	r_02_mov_acc.csv	movil	C	0.6	R	0.5	R	0.6	R	1
18	r_03_mov_acc.csv	movil	R	0.5	R	0.5	C	0.6	R	1
19	r_04_mov_acc.csv	movil	R	0.5	C	0.5	R	0.6	R	1
20	r_05_mov_acc.csv	movil	C	0.6	C	0.5	R	0.6	R	1
21	W_muestra_1_2014_05_08.csv	farsens	W	0.5	W	0.5	W	0.8	W	1
22	W_muestra_2_2014_05_08.csv	farsens	W	0.5	C	0.625	C	0.8	C	1
23	W_muestra_3_2014_05_08.csv	farsens	W	0.5	C	0.626	C	0.6	W	1
24	W_muestra_4_2014_05_08.csv	farsens	W	0.7	W	0.75	W	0.6	W	1
25	W_muestra_5_2014_05_08.csv	farsens	W	0.7	W	0.75	W	1	W	1
26	w_01_mov_acc.csv	movil	W	0.5	W	0.625	W	1	W	1
27	w_02_mov_acc.csv	movil	C	0.5	W	0.625	W	1	W	1
28	w_03_mov_acc.csv	movil	W	0.5	W	0.625	W	1	W	1
29	w_04_mov_acc.csv	movil	C	0.5	W	0.625	W	1	W	1
30	w_05_mov_acc.csv	movil	C	0.6	C	0.5	W	0.8	W	1

F.2.1 k = 10

k=10		#	Precision	Recall	F-Score	Acc
"C"	TP	10	47,6%	100,0%	65%	63%
	TN	9				
	FP	11				
	FN	0				
"R"	TP	2	100,0%	15,4%	27%	63%
	TN	17				
	FP	0				
	FN	11				
"W"	TP	7	100,0%	38,9%	56%	63%
	TN	12				
	FP	0				
	FN	11				
			82,5%	51,4%	49,1%	63,3%

F.2.2 k = 8

k=8		#	Precision	Recall	F-Score	Acc
"C"	TP	10	50,0%	100,0%	67%	67%
	TN	10				
	FP	10				
	FN	0				
"R"	TP	3	100,0%	23,1%	38%	67%
	TN	17				
	FP	0				
	FN	10				
"W"	TP	7	100,0%	41,2%	58%	67%
	TN	13				
	FP	0				
	FN	10				
			83,3%	54,8%	54,2%	66,7%

F.2.3 k = 5

k=5		#	Precision	Recall	F-Score	Acc
"C"	TP	10	55,6%	100,0%	71%	73%
	TN	12				
	FP	8				
	FN	0				
"R"	TP	4	100,0%	33,3%	50%	73%
	TN	18				
	FP	0				
	FN	8				
"W"	TP	8	100,0%	50,0%	67%	73%
	TN	14				
	FP	0				
	FN	8				
			85,2%	61,1%	62,7%	73,3%

F.2.4 k = 2

k=2		#	Precision	Recall	F-Score	Acc
"C"	TP	10	83,3%	100,0%	91%	93%
	TN	18				
	FP	2				
	FN	0				
"R"	TP	9	100,0%	81,8%	90%	93%
	TN	19				
	FP	0				
	FN	2				
"W"	TP	9	100,0%	81,8%	90%	93%
	TN	19				
	FP	0				
	FN	2				
			94,4%	87,9%	90,3%	93,3%

F.2.5 Porcentaje métricas de evaluación K=10