



Universidad
Zaragoza

Trabajo Fin de Máster

Desarrollo de un dispositivo wearable para el reconocimiento de actividades de la vida diaria

Autor

Alejandro García Osés

Directores

Álvaro Marco Marco
Bonifacio Martín del Brío

Escuela de Ingeniería y Arquitectura
2016



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. ALEJANDRO GARCÍA OSÉS

con nº de DNI 73112258L en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster Universitario en Ingeniería Electrónica, (Título del Trabajo)

DESARROLLO DE UN DISPOSITIVO WEARABLE PARA EL RECONOCIMIENTO DE ACTIVIDADES DE LA VIDA DIARIA

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 29 de Agosto de 2016

Fdo: Alejandro García Osés

Índice

1	Introducción	4
1.1	Resumen y motivación	4
1.2	Estado del Arte	4
1.2.1	Tecnologías, comunicación de los dispositivos	4
1.2.2	Tipo de actividades	6
1.2.3	Tipos de dispositivos	8
1.2.4	Método experimental	10
1.2.5	Preprocesado y Algoritmos de clasificación	11
1.2.6	Algoritmos de clasificación	14
1.2.7	Setup de evaluación y resultados	17
2	Objetivos y Desarrollo del proyecto	18
2.1	Objetivos	18
2.2	Materiales a utilizar	19
2.2.1	Sensor inercial	19
2.2.2	Programmable System-On-Chip (PSoC)	20
2.2.3	Batería	21
2.2.4	Carcasa	21
2.2.5	Wearable final	21
2.3	Test del dispositivo	22
2.4	Obtención y visualización de los datos	27
2.4.1	Filtrado de servicios y características	27
2.4.2	Base de datos	27
2.4.3	Interfaz de usuario	28
2.4.4	Interfaz de usuario secundaria	29
2.4.5	Visualización en ordenador	30
2.5	Experimentación	30
2.6	Selección de características	32
2.7	Algoritmo de clasificación	33
2.8	Resultados	34
2.8.1	Saltar	34
2.8.2	Hablar por teléfono	34
2.8.3	Levantado	34

2.8.4	Andar.....	35
2.8.5	Barrer	35
2.8.6	Abrir-Cerrar puerta	35
2.8.7	Leer/Pasar página.....	35
2.8.8	Transicionales	35
2.8.9	Resto de actividades.....	37
2.8.10	Orden de las redes escalonadas.....	37
2.8.11	Características utilizadas.....	38
2.9	Algoritmo Embebido.....	38
2.9.1	Obtención y almacenamiento de datos del BNO055	39
2.9.2	Preprocesado de datos y obtención de actividad realizada y de estadísticas	40
2.9.3	Envío de datos mediante Bluetooth a dispositivo Android.....	44
2.10	Visualización de datos en Android.....	44
2.11	Resultados del sistema embebido	46
2.11.1	Levantado.....	46
2.11.2	Saltar	46
2.11.3	Andar.....	46
2.11.4	Hablar por teléfono	46
2.11.5	Abrir/Cerrar puerta.....	47
2.11.6	Barrer	47
2.11.7	Transicionales	47
2.11.8	Beber.....	47
2.11.9	Comer.....	47
2.11.10	Ver TV.....	47
2.11.11	Utilizar Ordenador/Tablet	48
3	Conclusiones	48
4	Referencias.....	50

1 INTRODUCCIÓN

1.1 RESUMEN Y MOTIVACIÓN

El objetivo del trabajo consiste en la detección de actividades realizadas por una persona en su vida diaria. Esta detección se basa en un dispositivo PSoC (*Programmable System-on-Chip*), un circuito de sensado y comunicación Bluetooth Low Energy.

Este trabajo se enmarca en el proyecto de investigación “Memory Lane”, financiado por el Ministerio de Economía y Competitividad, el cual consiste en la captura de imágenes y otros datos de actividades de personas con necesidades especiales, así como su reconocimiento. El objetivo es crear un sistema que, sin interferir en el día a día de la persona, cree un diario personal de su vida con diferentes propósitos. Por ejemplo, en el caso de datos visuales capturados con cámaras, el diario permite a la persona poder acceder a un evento pasado, ejercitando de esta forma la memoria o servir como ayuda o tutorial a la hora de realizar ciertas tareas.

En este marco, el presente trabajo aborda el objetivo de reconocimiento de actividades mediante datos obtenidos con sensores electrónicos.

1.2 ESTADO DEL ARTE

Existen numerosos trabajos orientados al reconocimiento de actividades de la vida diaria. Si bien todos los autores persiguen el mismo objetivo, el desarrollo, evaluación y los métodos empleados difieren. Por ello es necesario conocer las diferentes técnicas utilizadas en la literatura. Uno de los factores más limitantes en el desarrollo de aplicaciones para reconocer actividades es la tecnología. La ley de Moore establece que aproximadamente en un periodo de dos años el número de transistores disponibles en un microcontrolador se duplica. Esto implica que aplicaciones que anteriormente procesaban los datos en un servidor ahora puedan utilizar un microcontrolador de tamaño reducido, susceptible de ser utilizado en un dispositivo *wearable*.

1.2.1 Tecnologías, comunicación de los dispositivos

El proceso de creación de un modelo clasificador requiere de una fase de entrenamiento y otra de evaluación. Para poder llevarlas a cabo se requiere una base de datos del problema suficientemente completa, es decir, debe existir un nexo entre los datos externos y el dispositivo encargado de recopilar y/o procesar los datos. Para el caso de los *wearables* se opta por utilizar comunicación inalámbrica. El protocolo de comunicación no es único, por lo tanto, se pueden encontrar proyectos que utilicen diferentes tecnologías. En el presente trabajo se describirá el protocolo Bluetooth, dado que es un tipo de comunicación en auge y que cualquier dispositivo actual permite manejar.

1.2.1.1 Bluetooth

Bluetooth es un estándar de comunicación inalámbrica orientado a distancias cortas (< 100 m). Las ventajas de una tecnología inalámbrica son evidentes, la inexistencia de cables permite una

mayor movilidad y un menor costo. Además, al tratarse de un estándar, es utilizable en varios dispositivos. La tecnología Bluetooth dispone de varias versiones, siendo las más recientes compatibles con las más antiguas:

- Bluetooth 1.0
- Bluetooth 2.0
- Bluetooth 3.0
- Bluetooth 4.0

A continuación, se explican distintos conceptos clave de la versión 4.0 debido a que es la más reciente y la que se va a utilizar en este proyecto:

El protocolo Bluetooth 4.0 surgió en el año 2010 e incluye tres subconjuntos: Bluetooth clásico, Bluetooth de alta velocidad (basado en Wifi), y Bluetooth Low Energy. El último subconjunto es el más adecuado para la utilización en un dispositivo *wearable*, ya que se caracteriza por su reducido uso de energía.

Se requiere conocer una serie de conceptos de Bluetooth Low Energy para su utilización:

Roles

- **Central y periférico:** Estos dos roles hacen referencia a la creación de un enlace entre dos o más dispositivos. El dispositivo periférico es aquel que es visible al resto de dispositivos, es decir, anuncia al resto su capacidad para conectarse. Por otra parte, el rol central es aquel capaz de mandar una propuesta de conexión para crear un enlace.
- **Servidor y cliente:** Respecto a la transmisión de información, existen el rol de cliente y de servidor. El cliente es aquel que recibe los datos mientras que el servidor es aquel que envía los datos.

No existe relación entre ambos puntos, ya que por ejemplo un servidor puede tener un rol tanto central como periférico, aunque normalmente el rol periférico actúa como servidor y el central como cliente.

Generic Attribute Profile (GATT)

Se trata de una especificación que define el intercambio de datos entre dos dispositivos a través de servicios y características, los cuales serán definidos posteriormente.

Attribute Protocol (ATT)

ATT es un protocolo que define cómo se transmite la información (atributos) entre dispositivos. Los atributos son los bloques de datos que se intercambian, incluyéndose características, servicios y descriptores. Cada atributo tiene un identificador único denominado *Universally Unique Identifier* (UUID). Este identificador consta de 128 bits (p ej. 00000000-0000-1000-8000-00805F9B34FB).

Características

Una característica contiene los valores de los datos a recibir, además de los descriptores. Los descriptores son datos que indican información acerca de los valores de las características, por ejemplo, pueden indicar el número de bits de una característica.

Servicios

Se denomina servicio a un conjunto de características, las cuales definen una función especial. Un servicio de Monitorización de Pulso Cardíaco englobará diferentes características además de la medida de las pulsaciones.

Ventajas y desventajas

Los factores más limitantes son la transferencia de datos y la distancia de funcionamiento, ya que comunicaciones inalámbricas como Wifi, consiguen una transferencia de datos mayor de 100Mb/s y una distancia de trabajo superior a la de Bluetooth Low Energy (BLE). En el campo de los dispositivos *wearable* la distancia no supone un problema en la mayoría de los casos, ya que, para la monitorización de actividades cotidianas no existe una gran distancia entre dispositivo emisor y receptor. Entre las ventajas de BLE se encuentran el reducido uso de energía y la amplia implantación que presenta, ya que cualquier terminal, como un *Smartphone*, con una versión de Android superior a 4.3 o una versión de iOS superior a 5 posee soporte para esta tecnología. En la Tabla I se muestran algunas de las especificaciones de Bluetooth Low Energy.

Energía utilizada	0.01 a 0.5 W
Banda de frecuencia	2.4GHz
Técnica de modulación	Espectro ensanchado por salto de frecuencia
Velocidad de transferencia máxima	1Mb/s
Distancia máxima	250 m

Tabla I. Características de Bluetooth

1.2.2 Tipo de actividades

Las actividades a monitorizar dependen del estudio que se desee realizar. Estudios dedicados a monitorizar la vida diaria de las personas mayores como los de [1]–[3], estarán orientados a mejorar la calidad de vida, detectar el deterioro cognitivo de la persona y evitar accidentes relacionados con la edad. Por ello en la literatura podemos encontrar diversos artículos orientados a detectar caídas o inactividad. Por otra parte, si la investigación aborda las actividades de personas jóvenes, estará orientada a desarrollar modelos para determinar la actividad física desarrollada, el consumo de energía (calorías) o la monitorización del sueño. Actualmente, existen dispositivos comerciales creados para dicho propósito, como las pulseras *Fitbit* o *NikeFuel*.

Referencias	Actividades clasificadas	Nro. de actividades	Nro. de sujetos
Saemi Im et al. [4]	Estar de pie/ Sentarse/Caminar/Correr/Acostarse/Cortar/ Planchar/Peinarse/Cepillarse los dientes/Agitar las manos/ Limpiar/Coger paraguas/Tomar foto/Aspirar el suelo/Hablar por teléfono/Leer/Aplaudir/Saltar a la comba/Aplicarse loción/Empujar carro	20	15
Ling Bao y Stephen S. Intille [5]	Andar/Sentarse/Quedarse quieto/ Ver TV/ Correr/ Estirarse/ Fregar/Doblar ropa/Cepillarse los dientes/Utilizar ascensor/ Andar llevando objetos/Trabajar con el ordenador/Comer o beber/Leer/Andar en bicicleta/Entrenamiento de fuerza/Usar el aspirador/Tumbarse a descansar/Subir escaleras/Utilizar escalera mecánica	20	20
A Moncada-Torres et al. [6]	Andar/Correr/Subir escaleras/Bajar escaleras/Estar de pie/ Sentarse/Tumbarse/Lavarse los dientes/Quitarse o ponerse una chaqueta/Beber/Cortar comida/Quitarse o ponerse zapatos/Pelar zanahoria/Escribir/Prueba de caja y cubos/ Untar mantequilla en pan	16	6
Marten Pijl et al. [7]	Guardar comida/Preparar la cena/Comer/Lavar platos/Usar el aspirador/Preparar una bebida	6	8
Debraj De et al. [8]	Entrar a casa corriendo/Entrar a casa/Usar frigorífico/Lavar utensilio de cocina/Cocinar/Sentarse y comer/Usar el lavabo/Salir de casa/Subir escaleras/Bajar escaleras/Estar de pie/Recostarse en la cama/Sentarse en la cama/Recostarse en el suelo/Sentarse en el suelo/Sentarse en el sofá/Sentarse en el retrete	16	1
Piyush Gupta & Tim Dallas [9]	Saltar/Correr/Andar/Sentarse/Sentarse y levantarse/ Arrodillarse y levantarse	6	9
Nirmalya Roy et al. [10]	Sentarse/Estar de pie/Andar/Correr/Recostarse/Subir escaleras/Lavar/Cocinar/Medicarse/Lavarse las manos/Regar las plantas/Barrer	12	10
Praneeth Vepakomma et al. [11]	Correr/Caminar/Usar frigorífico/Limpiar utensilio/ Cocinar/ Sentarse y comer/Usar lavabo/Levantarse y charlar/Salir de casa/Entrar en casa/Subir escaleras/Bajar escaleras/Estar de pie/Recostarse en la cama/Sentarse en la cama/Sentarse en la silla del escritorio/Recostarse en el suelo/Sentarse en el suelo/Recostarse en el sofá/Sentarse en el sofá/Sentarse en retrete	22	2
Kai Zhan et al. [12]	Caminar/Subir escaleras/Bajar escaleras/Beber/ Levantarse/ Sentarse/Leer/Ver televisión/Escribir/Abrir el grifo/Lavarse las manos	11	5
Kerem Altun et al. [13]	Sentarse/Estar de pie/Recostarse sobre la espalda/Recostarse sobre el lado derecho/Subir escaleras/Bajar escaleras/Estar en el interior de un ascensor/Moverse dentro del ascensor/Andar en un aparcamiento/Andar en cinta para correr a 4 km/h en plano y con 15° de inclinación/Andar en cinta de correr a 8km/h/Hacer ejercicio en <i>stepper</i> /Ejercicio en máquina elíptica/Andar en bicicleta estática horizontal/Andar en bicicleta estática vertical/Remar/Saltar/Jugar al baloncesto	19	8

Tabla II. Comparación de estudios en base a las actividades y los sujetos analizados

Las actividades de la vida diaria son estudiadas para monitorizar el estado cognitivo de las personas mayores, es por esto que se establece una clasificación concreta en el campo de la gerontología, creada por el Doctor Sidney Katz [14]:

- Actividades de la vida diaria básicas (AVDB): Contemplan aquellas actividades que se han aprendido desde la infancia y que están relacionadas con el cuidado personal básico: usar el baño, alimentarse, vestirse, desplazarse, bañarse, arreglarse, etc.
- Actividades de la vida diaria instrumentales (AVDI): A diferencia de las anteriores las AVDI presentan un mayor grado de complejidad. Las AVDI son aquellas actividades que son aprendidas durante la adolescencia y que son necesarias para vivir independientemente. Entre éstas se incluyen habilidades como: usar el teléfono, comprar, preparar la comida, limpiar la casa, hacer la colada, transportarse, medicarse correctamente y manejar la financiación de la casa. Al requerir una capacidad cognitiva mayor, este tipo de actividades se utilizan para determinar un deterioro en las funciones cognitivas de la persona. De acuerdo con [15], cambios en las AVDI son los primeros indicativos de enfermedades como el Alzheimer.

En la Tabla II, se recopilan actividades que han sido motivo de estudio en artículos.

1.2.3 Tipos de dispositivos

Dispositivos de sensado portátiles (*wearables*)

- Acelerómetros

Uno de los sensores más populares en el reconocimiento de actividades o gestos. Pueden ser tanto biaxiales como triaxiales. El procesado de los datos de aceleración de cada uno de los ejes permite clasificar distintos tipos de movimiento. Las medidas se basan en una masa de prueba suspendida mediante suspensión mecánica, la aceleración actuará sobre la masa de acuerdo a la segunda ley de Newton.

- Giróscopos

Este tipo de sensores permiten medir la velocidad angular hasta en tres ejes. Esta información se requiere para conocer la orientación. La velocidad angular se mide típicamente en radianes por segundo o grados por segundo. No es tan popular como el acelerómetro debido a la deriva que presenta.

- Magnetómetros

Los magnetómetros miden el campo magnético de su entorno. El campo magnético existente en la Tierra se mide mediante el dispositivo, proporcionando información de su orientación, tal y como funcionaría una brújula. La unidad de medida utilizada es el Tesla [T]. Estos sensores pueden estar basados en efecto Hall o en elementos magnetorresistivos.

- Sensores de Temperatura

Permiten medir la temperatura ambiente, proporcionando información contextual extra a cierto tipo de actividades. Por ejemplo, cocinar, abrir el frigorífico o salir a la calle son actividades en las que la temperatura presenta cierta relevancia.

- Cámaras portátiles y micrófonos

Las cámaras y micrófonos proporcionan una gran información de la actividad en su contexto. Entre sus inconvenientes se encuentran la falta de privacidad y el coste computacional que conlleva procesar información de imagen y sonido.

- Sensores barométricos

Los sensores barométricos monitorizan la presión atmosférica del ambiente. Son relevantes a la hora de medir actividades como subir o bajar escaleras, ya que las pequeñas variaciones de presión son detectadas por el sensor.

Dispositivos de sensado en la infraestructura

Al contrario que la mayor parte de sistemas de sensado portátiles, los dispositivos de infraestructura no están acoplados al individuo y normalmente otorgan una información más contextual, relevante a la hora de clasificar actividades más complejas. La ubicación de la persona tiene gran relevancia, ya que actividades efectuadas en lugares poco habituales pueden evidenciar problemas cognitivos en una persona.

- Sensores de presión

Para poder conocer la posición de una persona dentro de un recinto, los sensores de presión presentan una buena opción. Colocados en lugares determinados, como una silla u otro tipo de asientos, permite conocer donde está situado el individuo. Dichos sensores son susceptibles a falsos positivos si se sitúa un objeto pesado en el lugar donde se sitúan.

- Balizas Bluetooth

Se trata de dispositivos que mediante comunicación Bluetooth Low Energy, transmiten información a un cliente, como un Smartphone, cuando se encuentran a una distancia adecuada. En el campo del reconocimiento de actividades, las balizas permiten detectar aproximadamente el emplazamiento de la persona en el hogar, lo cual otorga información importante al algoritmo de clasificación.

- Etiquetas y lector RFID

Las etiquetas RFID se han utilizado en varios estudios para la detección de uso de determinados utensilios del hogar. Esto facilita el reconocimiento de actividades relacionadas con el objeto en uso.

- Cámaras y micrófonos montados

Al igual que su equivalente portátil, los micrófonos y las cámaras montadas ofrecen información acerca de la actividad que está realizando el usuario. De igual manera, la privacidad y el coste computacional, son sus principales desventajas.

- Sensores de movimiento

Basados en diferentes tecnologías, como infrarrojos o microondas, los sensores de movimiento permiten localizar a una persona dentro de un recinto.

En la Tabla III se representan diferentes estudios junto con los tipos de sensores que se utilizaron.

Referencias	Dispositivos <i>wearable</i>	Sensores en infraestructura
Saemi Im et al. [4]	Lector RFID y Acelerómetro de 3 ejes	Etiquetas RFID
Ling Bao y Stephen S. Intille [5]	Acelerómetro de 2 ejes	----
A Moncada-Torres et al. [6]	Sensor barométrico , acelerómetro y giróscopo de 3 ejes	----
Marten Pijl et al. [7]	----	Cámara y micrófono
Debraj De et al. [8]	Acelerómetro, giróscopo, sensor de temperatura, humedad y presión atmosférica	Balizas Bluetooth
Piyush Gupta & Tim Dallas [9]	Acelerómetro de 3 ejes	----
Nirmalya Roy et al. [10]	Acelerómetro y Giróscopo de 3 ejes integrados en Smartphone	Detectores infrarrojos de movimiento
Praneeth Vepakomma et al. [11]	Acelerómetro y giróscopo de 3 ejes, GPS, sensores de temperatura, humedad y presión atmosférica	Balizas Bluetooth
Kai Zhan et al. [12]	Cámara portátil y acelerómetro de tres ejes	----
Kerem Altun et al. [13]	Acelerómetro, giróscopo y magnetómetro de 3 ejes	----

TABLA III. Comparación de estudios en base a dispositivos empleados

1.2.4 Método experimental

El método experimental incluye ciertos parámetros que han de tenerse en cuenta, como el número de personas que forman parte del experimento y el tiempo total en el que se han tomado las muestras. El número de sujetos y la cantidad de datos condicionarán el preprocesado y la evaluación de los resultados.

La fase más relevante en el método experimental es la toma y etiquetado de datos. En ocasiones, los datos se toman y almacenan continuamente, para proceder a etiquetar manualmente cada trama

de datos, como en [15]. El procedimiento más habitual consiste en la toma de datos con etiquetado en tiempo real, el cual puede ser llevado a cabo por el propio sujeto, quien mediante una aplicación móvil o similar indicaría qué actividad estaría realizando [16]. Con el uso de un *Smartphone*, la toma de datos y el etiquetado pueden incluirse en el mismo dispositivo, lo cual supone una ventaja. Otra opción consiste en el etiquetado por un supervisor, el cual anotaría la actividad realizada por el sujeto [17].

En cuanto a la frecuencia de muestreo, estudios previos han demostrado que el movimiento del cuerpo humano se puede modelar como señales de 18 Hz e inferior [10]. No obstante, la elección varía en base al estudio, siendo lo habitual escoger un valor superior a 18 Hz.

Gran parte de las investigaciones realizadas se han llevado a cabo bajo en un entorno de laboratorio, es decir, en un entorno concreto, con una duración de actividades estipulada o una supervisión por parte de una o más personas. Este hecho provoca que al realizarse el mismo tipo de actividades en un entorno fuera de laboratorio, la precisión del sistema clasificador disminuya. En [16] se realizan diversas tomas de datos, unas fuera y otras dentro del laboratorio. Se observa como la precisión disminuye al utilizar solamente datos tomados en laboratorio para clasificar actividades fuera de éste.

El tiempo de experimentación es otro parámetro a tener en cuenta. Investigaciones orientadas a obtener un clasificador para observar el desempeño de actividades durante un largo periodo de tiempo cuentan con un mayor número de muestras. Este modo de proceder se debe a que el objetivo es la monitorización personalizada de una o más personas. En estudios como [17] o [18], se tomaron datos durante de 2 a 3 meses y 28 días, respectivamente. Por el contrario, experimentos realizados con un mayor número de personas, como en [9] o [4], tienen una duración limitada de aproximadamente 2-3 minutos y 45 segundos por actividad respectivamente.

1.2.5 Preprocesado y Algoritmos de clasificación

Una vez generada la base de datos (proceso que ya en sí ocupa una enorme cantidad de tiempo y esfuerzo), se procede a realizar un preprocesado en el cuál se extraen diversas características de los datos en crudo obtenidos de los sensores. El procedimiento habitual consiste en dividir los datos en ventanas que abarcan un rango de datos en crudo. De estas ventanas se obtienen ciertas características, como las indicadas en la Tabla IV. No existe una medida estándar respecto a la anchura de la ventana; en la literatura se encuentran diversos valores, desde 1 segundo [19], hasta 25 segundos [7]. Esta diversidad existe debido a que la anchura depende del tipo de actividades a realizar, de la frecuencia de muestreo, etc.

No todas las características son relevantes para cierto tipo de actividades o para una posición concreta del sensor; por ejemplo, en [9] se seleccionan aquellas características independientes de la posición del sensor alrededor de la cintura. Además, un elevado número de características a procesar hará que el sistema sea más lento. Por todo ello se utilizan métodos de selección de características (*feature selection*), los cuales se pueden dividir en tres grupos[20]:

1.2.5.1 Métodos de filtrado

Este tipo de métodos tienen en cuenta solamente el conjunto de características iniciales y no el algoritmo clasificador empleado. Las características se puntúan en base a una métrica determinada, reflejando la capacidad discriminativa de cada una.

Entre estas métricas se encuentran:

- *Fisher Score*
- *T-test*
- *Information Gain*
- *Área bajo la curva ROC*

Características
Media
Percentiles
Varianza
Covarianza
Desviación Estándar
Derivadas
Integrales
<i>Zero-crossing Rate</i>
Máximo/Mínimo
Curtosis
Entropía
Energía
Frecuencia de pico
<i>Skewness</i>
Coefficientes Wavelet

TABLA IV. Características comúnmente empleadas en la literatura

Las características se valoran en base a una de las métricas anteriores y de cara al algoritmo clasificador, se escogen las características que mejor puntuación obtengan. La ventaja que presenta este método es el bajo coste computacional necesario para implementarlo. Sin embargo, normalmente no tienen en cuenta dependencias entre características, siendo necesario a menudo crear nuevas variables que representen combinaciones entre características ya existentes.

Existen distintos tipos de algoritmos específicamente creados para el filtrado de características, en concreto en [21] se presentan tres algoritmos utilizados por su robustez frente a clases con un número de muestras desequilibrado:

- *Relief-F*
- *Correlation-based Feature Selection (CFS)*
- *Fast Correlation Based Filter (FCBF)*

1.2.5.2 Métodos wrapper

Al contrario que los métodos de filtrado, en los métodos *wrapper* se seleccionan las características en base a un algoritmo clasificador que las evalúa según su precisión. Dado que la selección de características viene determinada por un algoritmo, estos métodos están indicados para estudios donde se vaya a utilizar un tipo de algoritmo de clasificación concreto [22]. Los métodos *wrapper* deben ser implementados sobre el subespacio de muestras de entrenamiento, si se aplican sobre el conjunto de muestras en su totalidad (entrenamiento y test), la precisión del clasificador final puede estar sesgada, mostrando una precisión más optimista de la que verdaderamente presenta [23]. Además del algoritmo clasificador, se utiliza un algoritmo de búsqueda, el cual añade o elimina características para encontrar el subespacio óptimo. Si el proceso de selección de características comienza tomando como punto de inicio todo el espacio de variables, la selección se denominará *Sequential Backward Elimination*. Si se comienza con el espacio vacío y se añaden características a éste, se denomina *Sequential Forward Selection*. Ejemplos de distintos algoritmos de búsqueda:

- *Hill Climbing*
- *First Best Search*
- Algoritmos genéticos

1.2.5.3 Métodos embebidos

En este tipo de métodos el propio algoritmo clasificador determina las características más relevantes. Al igual que los métodos *wrapper*, están sujetos a un algoritmo concreto, pero su computación resulta menos costosa. Ejemplos de métodos embebidos son:

- *Random forests*
- Regularización Bayesiana

1.2.5.4 PCA

Además de seleccionar características, también se pueden extraer nuevas a partir de un conjunto existente. Uno de los algoritmos más utilizados para ello es el análisis de componentes principales o PCA (*Principal Component Analysis*), técnica que reduce la dimensionalidad del problema a partir de la matriz de covarianza del espacio de muestras. Los vectores propios de dicha matriz representan las direcciones donde la varianza es mayor, los cuales definen un nuevo espacio de muestras basado en unas nuevas características ortogonales, combinación lineal de las originales, de las cuales se seleccionan las que cuentan con un mayor valor propio (mayor significación estadística). Así, si se proyectan los datos originales en el nuevo espacio, se puede obtener un conjunto reducido de nuevas características que conserven la mayor parte de la información inicial. PCA requiere una capacidad de cálculo considerable, aunque es muy efectivo a la hora de reducir la dimensionalidad de un problema.

1.2.6 Algoritmos de clasificación

A lo largo de los últimos decenios se han ido incorporando nuevos algoritmos de clasificación; en concreto, para el reconocimiento de actividades, las técnicas han avanzado junto con la tecnología. En los estudios realizados a finales de los años 90 se observa el uso de algoritmos sencillos basados en umbrales, que clasifican un número de actividades posturales o locomotoras sencillas [24], [25]. Los umbrales se calculaban en base a la observación de la señal en crudo, estimando de esta manera el cambio transicional entre dos actividades. Con el paso de los años este tipo de procedimientos ha caído en desuso, sustituyéndose por técnicas más sofisticadas de *Machine Learning* [26]. Los algoritmos más utilizados se expondrán en base al estudio realizado en [27].

1.2.6.1 Árboles de decisión

ID3 y C4.5 son dos tipos de árboles de decisión que surgieron en 1970 y 1980 respectivamente. C4.5 probablemente sea el tipo de árbol de decisión más utilizado en la literatura para la clasificación de actividades. Se denominan de esta forma debido a que están formados por hojas (nodos de clase), ramas (bifurcaciones de decisión) y nodos. Cada nodo corresponde con una característica, los nodos superiores se deciden en base a un valor denominado *Information Gain*, de modo que el nodo superior se bifurcará en base a los valores de la característica con mayor *Information Gain*. De esta forma, surgen diferentes subespacios con valores concretos de una característica, en el nuevo subespacio se determina de nuevo la característica con mayor *Information Gain*. Este procedimiento continúa hasta que finalmente las bifurcaciones de decisión lleven a los nodos hoja, correspondientes con las clases. Una de las ventajas de éste método radica en la estructura, ya que es fácilmente interpretable por un humano.

1.2.6.2 Redes Bayesianas

Los algoritmos bayesianos determinan una clase en base a probabilidades condicionales entre distintas variables. Dada una serie de características, las redes bayesianas son capaces de inferir la probabilidad de que se dé una determinada clase. Las redes se representan normalmente en forma de grafos acíclicos, siendo los nodos las características y los enlaces la probabilidad condicional entre dos variables.

1.2.6.3 Aprendizaje basado en instancias

Un algoritmo basado en instancias muy conocido y utilizado en el reconocimiento de actividades es el denominado *k-Nearest Neighbours* (k-NN). En éste, la clase se asigna a una muestra en base al número de muestras vecinas. Si de entre k muestras vecinas, la mayor parte pertenecen a una clase, la muestra de ejemplo será asignada a dicha clase. La cercanía entre muestras se determina utilizando la distancia euclídea u otras, como la de Hamming. El valor de *k* se determina mediante técnicas heurísticas. Este algoritmo no requiere de entrenamiento, lo cual supone una importante ventaja, siendo su inconveniente principal la sensibilidad que presenta ante *outliers* o ruido.

1.2.6.4 Modelos de Markov

Los modelos de Markov son un tipo particular de red bayesiana. Han sido ampliamente utilizados en el campo del reconocimiento de voz, encontrándose estudios actuales que los utilizan [28], no

obstante, se han expandido hacia otros ámbitos, incluido el de reconocimiento de actividades. Es un método relevante debido a que tiene en cuenta los eventos pasados o, dicho de otra forma, relaciona muestras tomadas en distintos instantes de tiempo. Es por esto que en ocasiones se denominan clasificadores secuenciales [29]. Los Campos Aleatorios Condicionales son una variante de este tipo de algoritmos también utilizada en la literatura [8], [12].

1.2.6.5 *Redes Neuronales*

Se trata de algoritmos inspirados en el conexionado del cerebro, siendo el perceptrón multicapa una de las redes neuronales más utilizadas [26], [30]. Constan de una capa de nodos de entrada, una o más capas ocultas (nodos de procesamiento) y una capa de salida. Cada característica de entrada tiene un peso asignado, correspondiente con la importancia que tendrá dicha característica sobre la predicción final de la clase; estos pesos se obtienen mediante un proceso de entrenamiento en el que se hace uso de un conjunto de patrones-ejemplo. Así, la red neuronal crea iterativamente (conforme se presentan ejemplos) una frontera no-lineal que separa un conjunto de muestras en sus diferentes clases. Dentro de las redes neuronales, se encuentran también las *Temporal Delay Neural Networks* (TDNN), las cuales introducen como elementos de entrada muestras de instantes anteriores, además de las del momento presente; esto hace que se tenga en cuenta la secuencia temporal de entradas, por lo que también entran dentro del grupo de los algoritmos secuenciales.

1.2.6.6 *Lógica difusa*

La lógica difusa (*fuzzy logic*) [26] se basa en la forma del pensamiento humano. Mientras en la lógica Booleana solo se contemplan dos valores de clasificación (0 y 1), la lógica difusa establece una serie de valores intermedios (*fuzzy*) entre el 0 y el 1, indicando el grado de pertenencia de un caso a una clase. La clasificación con lógica difusa consta de tres fases. El primer paso se denomina *fuzzification* y consiste en convertir los valores de entrada en valores difusos. Estos valores difusos pasan por un proceso de inferencia en el cual se establecen reglas SI/ENTONCES entre las entradas y las salidas difusas. Por último, mediante combinación de las salidas difusas en un proceso denominado *defuzzification* se obtiene el grado de pertenencia de la muestra de entrada a una clase u otra.

1.2.6.7 *Métodos de ensemble*

Este tipo de métodos se basan en la utilización de distintos modelos de predicción combinados para clasificar muestras. Debido a esto, normalmente se utilizan algoritmos de pequeño coste computacional; por ejemplo, los *Random Forests* utilizan árboles de decisión como algoritmo base. Otros ejemplos de métodos de *ensemble* son *Bagging* [31], *Boosting* [32] y *Stacking* [33].

1.2.6.8 *Deep Learning*

Deep Learning es un tipo de algoritmo perteneciente al *Machine Learning*, que está cobrando mucha importancia en los últimos años, en especial en campos como el reconocimiento de imágenes. Se puede equiparar con una red neuronal con muchas capas ocultas, la cual es capaz de extraer y seleccionar diversas características de los datos de entrada en las primeras capas para posteriormente proceder a la clasificación en las últimas.

Referencias	Métodos de extracción/selección de características	Características	Algoritmos de clasificación
Saemi Im et al. [4]	---	Media/Energía/Entropía/Correlación en el dominio de la frecuencia	Árboles de decisión
Ling Bao y Stephen S. Intille [5]	---	Media/Energía/Entropía/Correlación	Tabla de decisión/C4.5/k-NN/Naive Bayes
A Moncada-Torres et al. [6]	Relief-F	Correlación/Media Cuadrática/Desviación estándar/Percentiles/Medida picopico	k-NN
Tzu-Ping Kao et al. [34]	LDA [35]	Media/Correlación/Desviación media absoluta/Media Cuadrática/Varianza/Desviación estándar	Clasificador basado en <i>Fuzzy-Based-Functions</i> (FBF)
Debraj De et al. [8]	---	Media/Varianza/Derivada primera y segunda	Campos aleatorios condicionales
Piyush Gupta & Tim Dallas [9]	Relief-F/SFFS	Energía/Entropía/Varianza/Diferencia entre máximos Media/Varianza/ <i>Windowed Mean</i> /Coeficientes del <i>Detrended Fluctuation Analysis</i>	k-NN Naive Bayes
Nirmalya Roy et al. [10]	---	Media/Máximo/Mínimo/Cruces por cero/Varianza/Correlación	Modelos ocultos de Markov
Praneeth Vepakomma et al. [11]	---	Media/Varianza/Derivada primera y segunda	Deep Learning
Kai Zhan et al. [12]	---	Magnitud y frecuencia del mayor pico de la transformada de Fourier/Media/Desviación estándar/Máximo/Cruces por un umbral/Tramas de vídeo	Campos aleatorios condicionales
Kerem Altun et al. [13]	PCA	Media/Varianza/ <i>Skewness</i> / <i>Curtosis</i> /Autocorrelación/Picos de las transformada de Fourier	Redes bayesianas /Redes neuronales artificiales/Árboles de decisión/SVM/k-NN/ <i>Dynamic Time Warping</i> (DTW)

TABLA V. Comparación de estudios en base a las actividades analizadas

Debido a su alto coste computacional, por el momento no son susceptibles de ser implementadas en un microcontrolador, pero si en un servidor externo [11].

En la Tabla V se muestran diversos estudios con las características, algoritmos de clasificación y métodos de selección/extracción de características empleados.

1.2.7 *Setup* de evaluación y resultados

Para determinar el desempeño de un clasificador se requiere una metodología y herramientas adecuadas. En primer lugar, una herramienta utilizada en casi la totalidad de estudios de clasificación es la matriz de confusión. Se trata de una matriz de $n \times n$ elementos, correspondientes con las n clases. En la Tabla VI se muestra un ejemplo de matriz de confusión con dos clases (“positivos”, “negativos”).

		Clase predicha	
		p	n
Clase real	p	VP	FN
	n	FP	VN

TABLA VI. Matriz de confusión con dos clases

Se deben tener en cuenta los siguientes conceptos:

- Falsos positivos (FP): dada la clase “positivos”, aquellas muestras que se han clasificado incorrectamente como pertenecientes a dicha clase.
- Falsos negativos (FN): dada la clase “negativos”, aquellas muestras que se han clasificado incorrectamente como pertenecientes a dicha clase.
- Verdaderos positivos (VP): dada la clase “positivos”, aquellas muestras que se han clasificado correctamente como pertenecientes a dicha clase.
- Verdaderos negativos (VN): dada la clase “negativos”, aquellas muestras que se han clasificado correctamente como pertenecientes a dicha clase.

La evaluación del conjunto de muestras se puede efectuar de diferentes formas. Una posibilidad consiste en la división de los datos de todos los sujetos en sus diferentes actividades, para luego ser mezclados aleatoriamente. Este conjunto de datos se divide en dos subconjuntos, uno de entrenamiento y otro de evaluación (en ocasiones, como en el caso del perceptrón, puede haber un tercer conjunto de validación), o bien se divide en distintos grupos para realizar validación cruzada (VC). Otra opción, posiblemente la más utilizada en el ámbito de la detección de actividades debido al reducido número de sujetos disponibles [5], [6], [9], [36], es el método *leave-one-subject-out*, que consiste en el entrenamiento del algoritmo con los datos provenientes de todos los sujetos salvo de uno, que será sobre el que se evaluará el algoritmo (por lo que puede considerarse un caso extremo de VC). Este procedimiento se repite a lo largo de todos los sujetos, es decir, se escoge otro individuo como test y se entrena con el resto. Con el resultado de clasificación de las

diferentes clases, se obtiene la precisión del clasificador. Si el objetivo no es obtener un resultado sobre todos los sujetos, también se puede obtener la precisión sobre un solo sujeto [8], dividiendo el espacio de muestras obtenidas de dicho individuo en dos subespacios, uno de entrenamiento y otro de test.

En la Tabla VII se muestran las precisiones obtenidas y el método de evaluación utilizado en clasificación de actividades.

Referencias	Exactitud
Saemi Im et al. [4]	<i>Leave-one-out</i> : +90%
Ling Bao and Stephen S. Intille [5]	Un usuario & <i>Leave-one-out</i> : 84.26% (promedio)
A Moncada-Torres et al. [6]	<i>Leave-one-out</i> : 57.12% Todas las muestras: 92.78% Un usuario: 95.06%
Marten Pijl et al. [7]	<i>Leave-one-out</i> : +80% Un usuario: +80%
Debraj et al. [8]	Un usuario: 80.48%
Piyush Gupta y Tim Dallas [9]	<i>Leave-one-out</i> : ~98%
Nirmalya Roy et al. [10]	Todas las muestras: ~70%
Praneeth Vepakomma et al. [11]	Un usuario: ~96%
Kai Zhan et al. [12]	Todas las muestras: ~90%
Kerem Altun et al. [13]	Todas las muestras: +95%

TABLA VII. Resultados y evaluaciones obtenidos en la literatura

Como se puede observar, la mayor parte de los estudios superan o se acercan al 90% de precisión, con los mejores resultados entorno al 95%. No obstante, se deben tener en cuenta el tipo de actividades concretas que se clasifican, dado que algunas presentan mucha mayor complejidad; se considera que un desempeño igual o superior al 90% denota un buen modelo clasificador.

2 OBJETIVOS Y DESARROLLO DEL PROYECTO

2.1 OBJETIVOS

Mediante un dispositivo *wearable*, formado por un sensor inercial y un PSoC, situado en la muñeca de un individuo (en el brazo hábil), se tratarán de clasificar las siguientes 17 actividades de la vida diaria:

1. Andar
2. Levantado
3. Levantado-Agachado (Transición)
4. Agachado-Levantado
5. Levantado-Sentado
6. Sentado-Levantado
7. Levantado-Tumbado
8. Tumbado-Levantado
9. Saltando
10. Comer
11. Beber
12. Manejar el ordenador
13. Barrer
14. Ver Televisión
15. Abrir/Cerrar puerta
16. Manejar tablet
17. Hablar por teléfono móvil

Se han elegido estas actividades en concreto dado que se considera que representan una gran parte de las actividades que se realizan en el ámbito doméstico.

Además, se obtendrán estadísticas relacionadas con algunas de las actividades (p ej., número de sorbos, tiempo bebiendo). Las actividades predichas y las estadísticas se enviarán a un Smartphone para la visualización en tiempo real de éstas y su almacenamiento en una base de datos.

2.2 MATERIALES A UTILIZAR

Para el proyecto se utilizará un dispositivo electrónico que consta de los siguientes elementos:

2.2.1 Sensor inercial

El sensor inercial utilizado es el BNO055 de Bosch [37]. Este sensor obtiene tres medidas diferentes en sus tres ejes espaciales: Aceleración, campo magnético y velocidad angular. Los ejes del integrado vienen indicados en la Fig. 1. Además, ofrece el denominado modo fusión, el cual permite obtener directamente medidas de orientación, expresadas bien mediante ángulos de Euler o bien mediante cuaternios. El hecho de que el propio integrado incluya calibración y medidas de orientación supone una ventaja, ya que no es necesario realizar estos procesos externamente, por ejemplo, en un microprocesador.

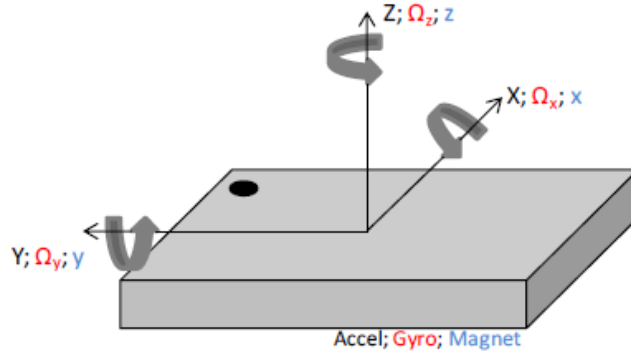


Fig. 1. Espacio cartesiano del integrado

2.2.2 Programmable System-On-Chip (PSoC)

El integrado utilizado para la recepción y manipulación de los datos pertenece a la familia de los PSoC 4000, los cuales incluyen la posibilidad de trabajar con bloques reconfigurables tanto analógicos como digitales.

La CPU está basada en la estructura del Cortex-M0 de ARM. Este es el encargado de obtener los datos provenientes del sensor y transmitirlos a través de Bluetooth Low Energy mediante un transceptor. El microcontrolador también es capaz de implementar un algoritmo de *Machine Learning* y de preprocesar los datos, lo cual se realizará en la fase final del proyecto. Las especificaciones son las siguientes:

- 48 MHz
- 32 KB de memoria flash
- 4 KB de SRAM

Tanto el Microcontrolador como el sensor estarán unidos en una placa PCB, mostrada en la Fig. 2, junto con los componentes pasivos correspondientes.

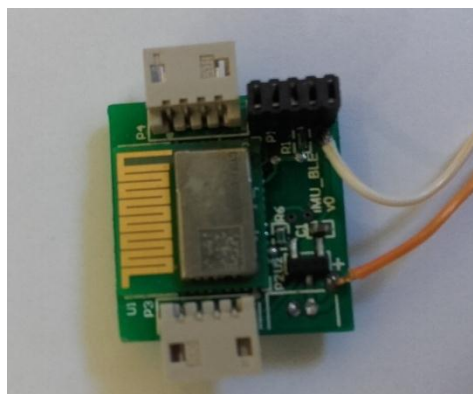


Fig. 2. Placa PCB utilizada

2.2.3 Batería

Para la alimentación del sensor se utilizará una batería de polímero de litio, 3.7V y 150 mAh, de tamaño reducido. Se muestra en la Fig. 3.



Fig. 3. Batería de litio

2.2.4 Carcasa

Para que el dispositivo pueda ser *wearable*, se requiere de una carcasa que se pueda acoplar a la parte del cuerpo que se necesite. Se utilizará una carcasa de material plástico acompañada por una correa. En la Fig. 4 se muestra la carcasa junto con la correa.

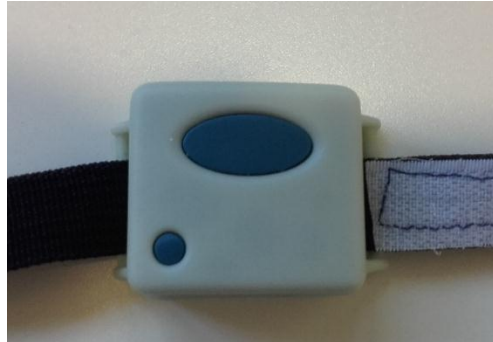


Fig. 4. Carcasa

2.2.5 Wearable final

Una vez ensamblados todos los componentes, el resultado final se puede observar en la Fig. 5.

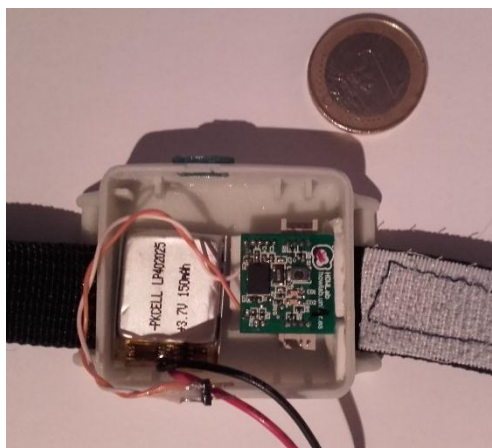


Fig. 5. Dispositivo final

2.3 TEST DEL DISPOSITIVO

Con el objetivo de comprobar el correcto funcionamiento del dispositivo, así como medir posibles derivas, offset o interferencias, se realizan varios experimentos. Las medidas obtenidas en ellos se muestran en las Fig. 6 a 14.

2.3.1.1.1 Test estático

1. 10 minutos quieto en una posición
2. Reiniciar dispositivo y situarlo 10 minutos quieto en aproximadamente la misma posición
3. Breve Inclinación y colocación en posición inicial

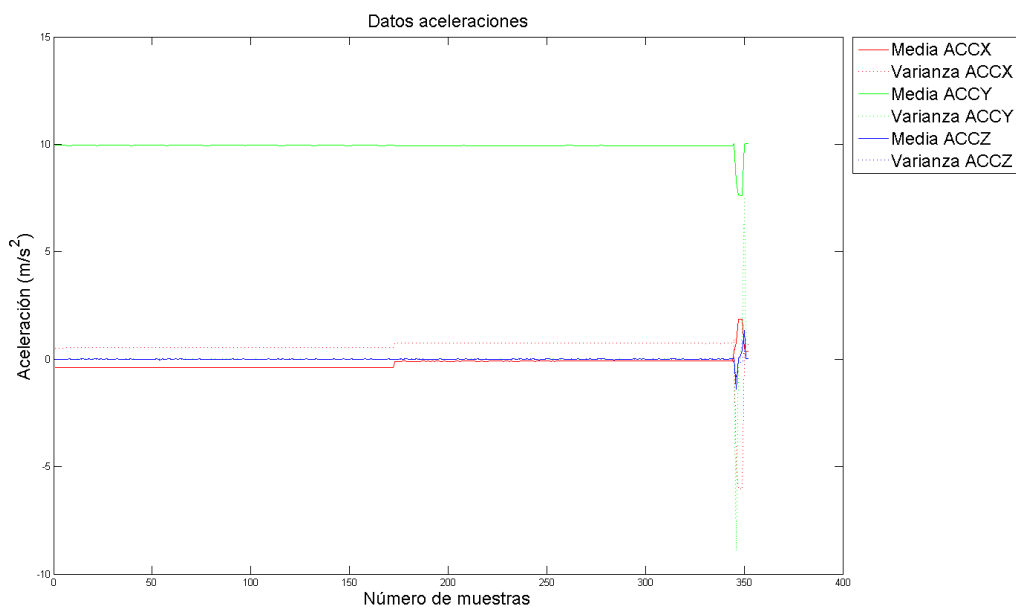


Fig. 6. Medias y varianzas de las aceleraciones

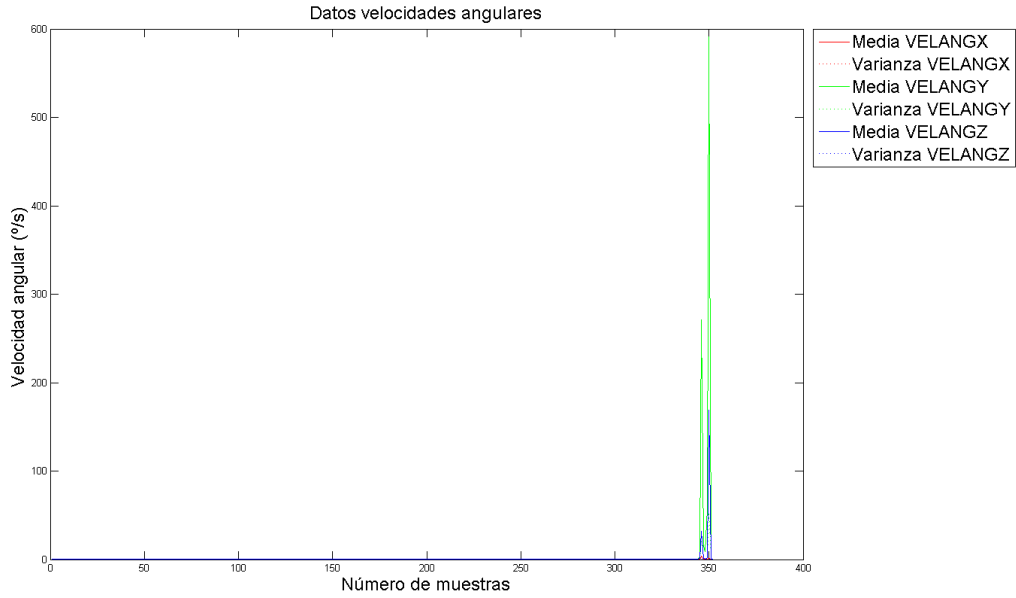


Fig. 7. Medias y varianzas de las velocidades angulares

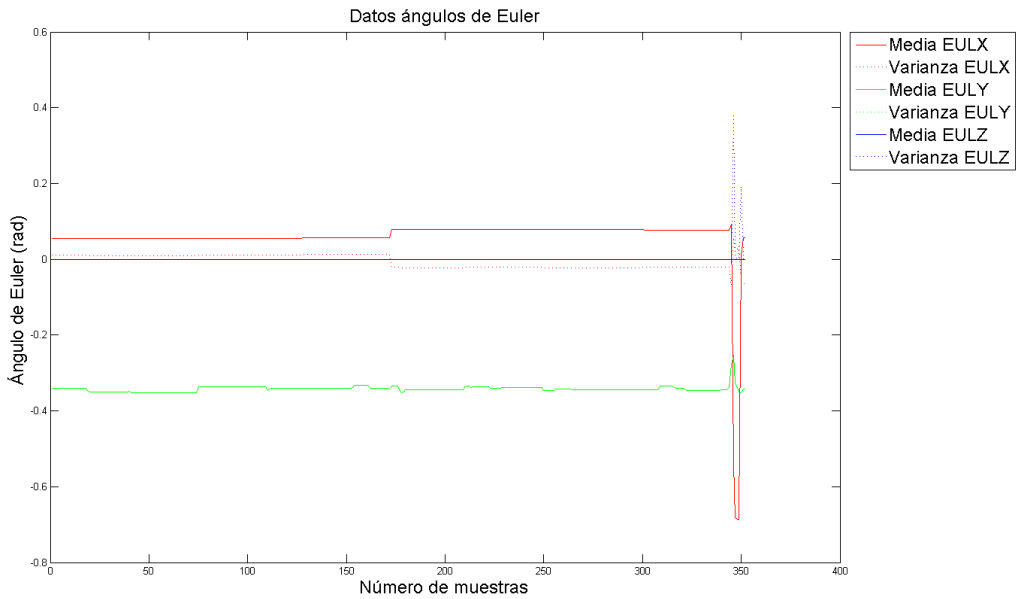


Fig. 8. Medias y varianzas de los ángulos de Euler

2.3.1.1.2 Test dinámico 1

1. Giro de 90° en X, posición inicial, giro de 90° en Y, posición inicial, giro de 90° en Z, posición inicial, movimientos erráticos, posición inicial.
2. Repetir

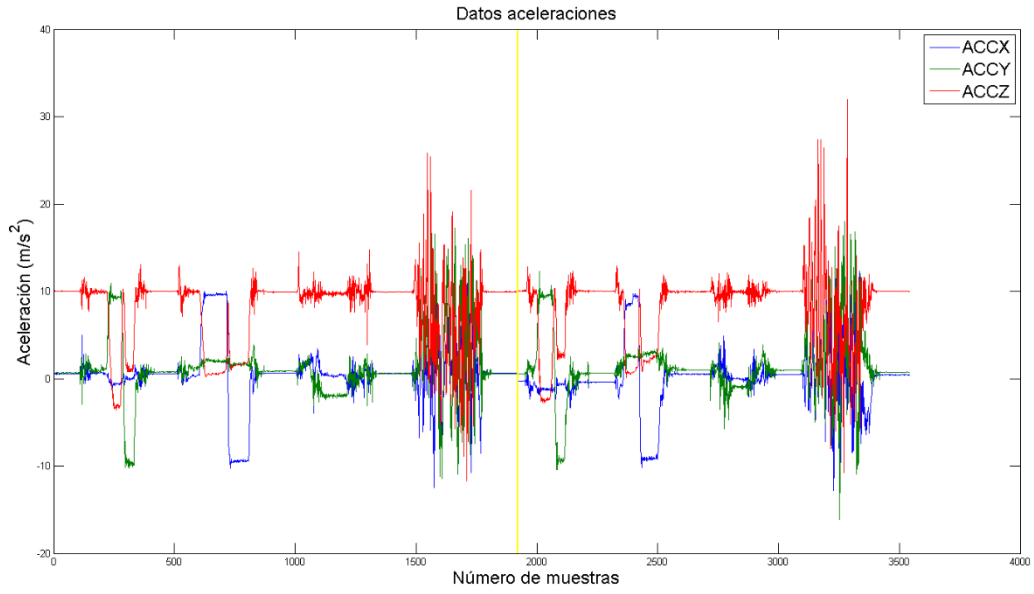


Fig. 9. Medidas de aceleraciones en tres ejes

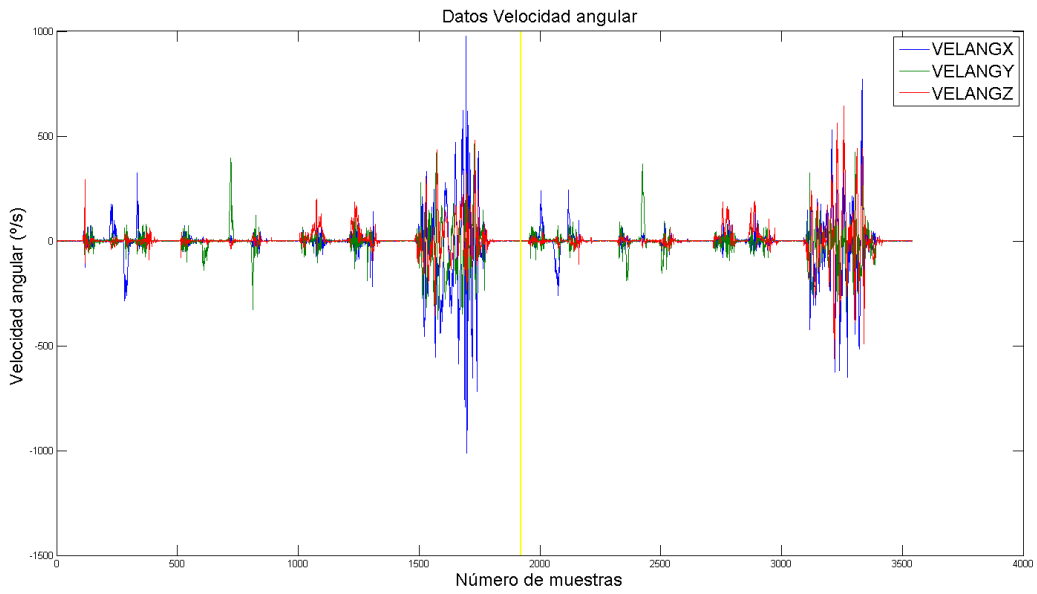


Fig. 10. Medidas de velocidades angulares en tres ejes

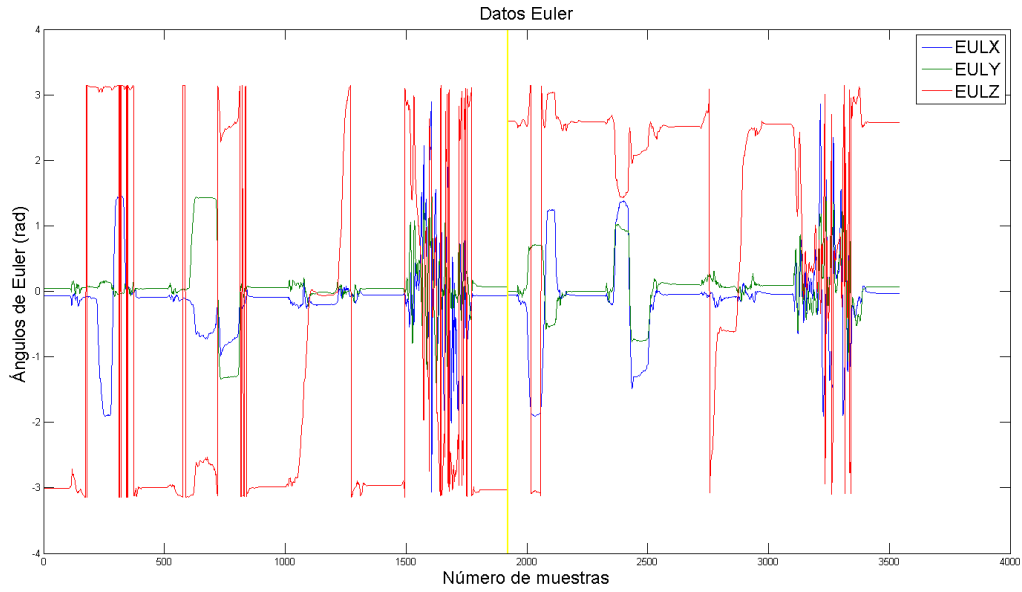


Fig. 11. Medidas de ángulos de Euler en tres ejes

2.3.1.1.3 Test dinámico 2

1. 10 Saltos, 10 sorbos de agua en botella, 10 segundos caminando.
2. Repetir
3. Repetir

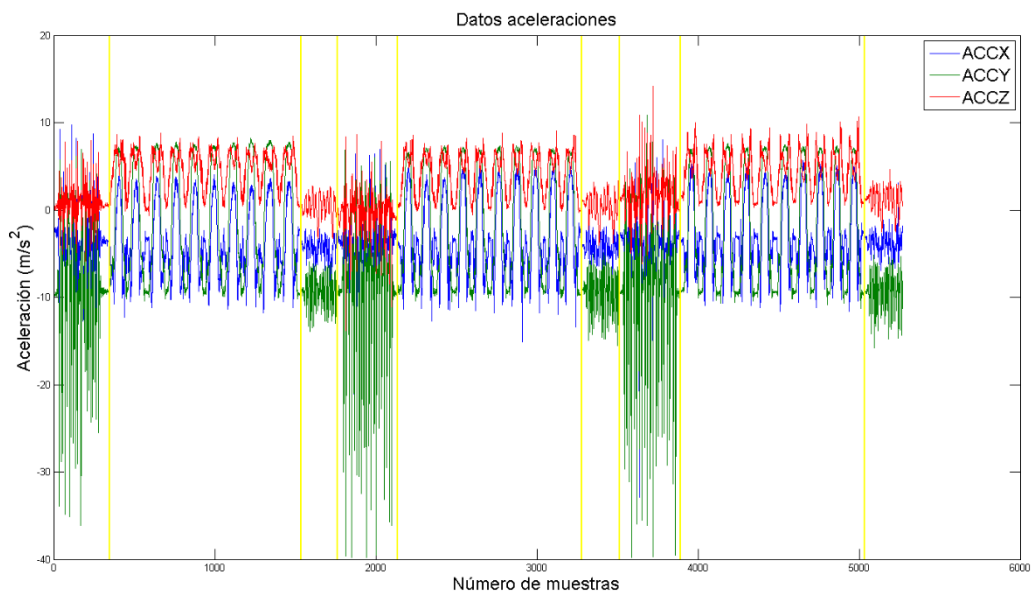


Fig. 12. Medidas de aceleraciones en tres ejes

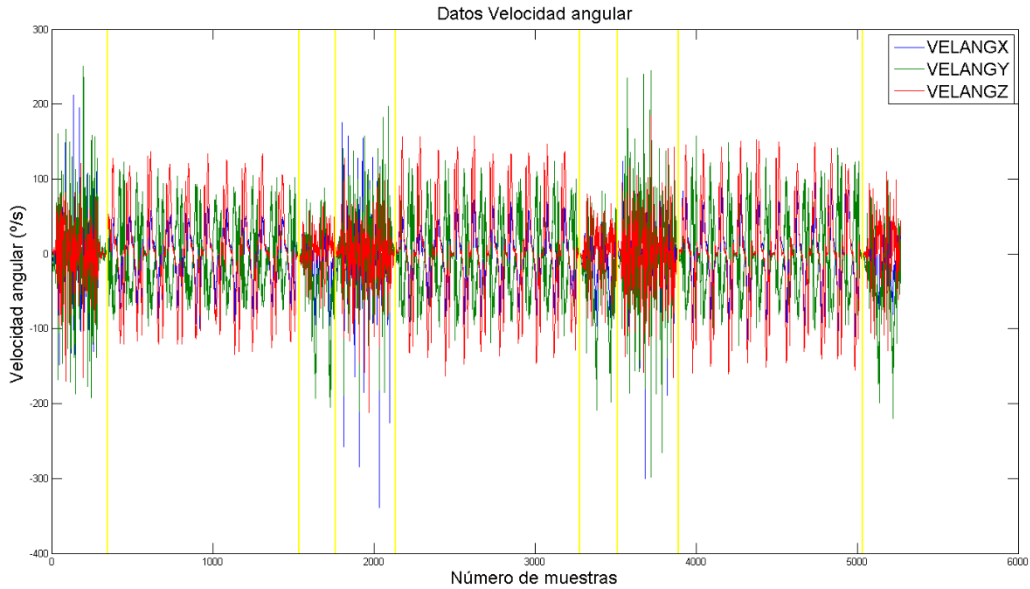


Fig. 13. Medidas de velocidades angulares

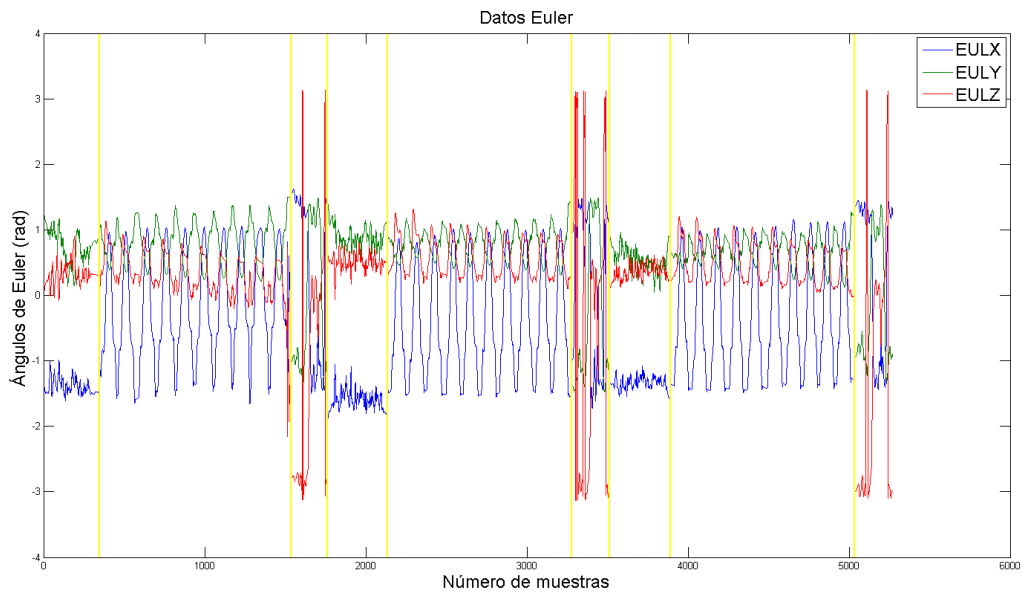


Fig. 14. Medidas de ángulos de Euler

Observando el test estático se concluye que el dispositivo no sufre derivas a lo largo del tiempo. Mediante el test dinámico número uno, se observa que movimientos bruscos no afectan a las medidas del dispositivo. No obstante, se observa que en posiciones extremas los ángulos de Euler pueden variar bruscamente. Finalmente, en el último test, se comprueba que una actividad es consistente independientemente de las veces que se realice.

2.4 OBTENCIÓN Y VISUALIZACIÓN DE LOS DATOS

El primer paso del proyecto consiste en la creación de una aplicación móvil para el almacenamiento de los datos provenientes del dispositivo *wearable* en una base de datos. Para realizar la aplicación, se parte de la aplicación de código abierto *Cysmart* [38] y se utiliza el programa Android Studio. Los cambios más relevantes incluyen el filtrado de los servicios y características ajenos al *wearable*, el almacenamiento de los datos en una base de datos y el desarrollo de una interfaz de usuario.

2.4.1 Filtrado de servicios y características

Como se ha indicado en apartados anteriores, cada característica, servicio y descriptor tiene un identificador único. La aplicación se configura de tal manera que solamente muestre aquellos servicios correspondientes con el servicio y las características deseadas. En concreto, se aisló el servicio denominado *Server_UART*, el cual consta de dos características: *Server_UART_Tx_Data* y *Server_UART_Rx_Data*. La característica que se necesita para la obtención de datos es la primera, la cual otorga datos de aceleración, campo magnético, velocidad angular y ángulos de Euler o cuaternios. La estructura de los datos es la siguiente:

AX1-AX2-AY1-AY2-AZ1-AZ2-MX1-MX2-MY1-MY2-MZ1-MZ2-GX1-GX2-GY1-GY2-GZ1-GZ2- EZ1-EZ2-EY1-EY2-EX1-EX2

Donde cada las letras A, M, G, E significan medidas de acelerómetro, magnetómetro, giróscopo y ángulos de Euler, respectivamente. Las letras X, Y y Z hacen referencia a los ejes cartesianos. Los números hacen referencia al número del bit. Como se puede observar, cada medida en su eje se representa mediante dos bytes.

La codificación es *Little Endian* y complemento a dos, es decir, los bytes 1 y 2 se intercambian entre sí para dar el valor real del sensor y el valor puede ser positivo o negativo. Posteriormente, los valores se convierten a enteros mediante el complemento a dos y se dividen por una escala que dependerá de la variable física. Las escalas se indican en la Tabla VIII.

Variable Física	Factor de escala
Acelerómetro	100 LSB
Magnetómetro	16 LSB
Giróscopo	16 LSB
Ángulos de Euler	16 LSB

TABLA VIII. Factores de escala aplicados a las medidas

2.4.2 Base de datos

Como base de datos se utilizará *SQLite*, ya que es la que está disponible por defecto en Android y existen ya clases y métodos dedicados para su uso desde Java.

La base de datos constará de una tabla con tantas columnas como datos se deseen obtener, en este caso contará con 18 campos, los cuales se muestran en la Fig. 15.

ID	NOMBRE	MANO	ACTIVIDAD	ACCX	ACCY	ACCZ	GYRX	GYRY	GYRZ	MAGX	MAGY	MAGZ	EULX	EULY	EULZ	DATE	TIME	
	Fil...	...	Filter				Fil...	Fil...	Fil...	Fil...	Filter	Filter	Fil...	F...	Filter	Filter	Filter	
1	1	Test	Dcha	Andar	-0.28	-0.8	9.79	-4.0625	13.0	-40.5625	0.0625	-0.1875	-0.125	11.5	-1.0	-4.75	18 abr. 2016	22:43 23 763
2	2	Test	Dcha	Andar	-0.27	-0.76	9.8	-3.25	12.25	-44.375	0.0	-0.1875	0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 23 856
3	3	Test	Dcha	Andar	-0.27	-0.78	9.79	-3.25	14.0625	-40.5625	-0.0625	-0.25	0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 23 954
4	4	Test	Dcha	Andar	-0.28	-0.77	9.8	-3.25	12.5625	-44.0	0.0	-0.125	0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 071
5	5	Test	Dcha	Andar	-0.27	-0.78	9.79	-2.5625	13.375	-40.5625	0.125	0.0	0.0	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 173
6	6	Test	Dcha	Andar	-0.27	-0.79	9.79	-2.5625	13.0	-43.0625	0.125	-0.125	0.0	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 271
7	7	Test	Dcha	Andar	-0.3	-0.77	9.82	-2.5625	13.6875	-40.1875	0.0625	0.0	0.125	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 367
8	8	Test	Dcha	Andar	-0.28	-0.78	9.82	-3.25	13.0	-43.0625	-0.0625	-0.0625	0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 466
9	9	Test	Dcha	Andar	-0.26	-0.77	9.79	-3.6875	14.0625	-40.1875	0.0625	-0.1875	-0.125	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 564
10	10	Test	Dcha	Andar	-0.28	-0.78	9.79	-3.25	11.875	-43.0625	0.125	0.0	-0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 674
11	11	Test	Dcha	Andar	-0.28	-0.76	9.8	-2.875	14.0625	-40.5625	0.0625	0.0	0.125	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 814
12	12	Test	Dcha	Andar	-0.25	-0.78	9.81	-3.6875	12.5625	-44.375	-0.1875	0.25	-0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 863
13	13	Test	Dcha	Andar	-0.26	-0.75	9.78	-2.875	14.0625	-41.0625	-0.1875	0.0	-0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 24 965
14	14	Test	Dcha	Andar	-0.28	-0.76	9.76	-2.875	13.375	-43.5625	0.1875	0.0	0.0	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 110
15	15	Test	Dcha	Andar	-0.26	-0.78	9.79	-3.25	14.0625	-41.0	-0.0625	0.0625	0.0	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 161
16	16	Test	Dcha	Andar	-0.28	-0.78	9.82	-1.75	13.0	-43.875	-0.125	0.0625	-0.125	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 268
17	17	Test	Dcha	Andar	-0.24	-0.79	9.82	-3.6875	13.6875	-40.5625	-0.0625	0.125	-0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 368
18	18	Test	Dcha	Andar	-0.31	-0.77	9.81	-3.25	13.6875	-44.0	-1.0625	0.3125	0.0	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 463
19	19	Test	Dcha	Andar	-0.28	-0.83	9.84	-3.25	13.375	-40.1875	-1.625	0.375	0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 587
20	20	Test	Dcha	Andar	-0.31	-0.77	9.83	-2.875	11.875	-43.5625	1.375	0.25	-0.0625	11.5	-1.0	-4.75	18 abr. 2016	22:43 25 681
21	21	Test	Dcha	Andar	-0.23	-0.79	9.8	-2.1875	14.5	-40.5625	-1.5625	0.625	-1.1875	11.5625	-1.0	-4.75	18 abr. 2016	22:43 25 789
22	22	Test	Dcha	Andar	-0.21	-0.84	9.82	-2.5625	13.375	-44.0	-0.9375	0.25	-6.8125	11.9375	-1.0	-4.8125	18 abr. 2016	22:43 25 872
23	23	Test	Dcha	Andar	-0.25	-0.83	9.79	-4.375	14.5	-39.75	-0.625	0.375	-13.3125	13.0	-0.9375	-4.875	18 abr. 2016	22:43 25 977
24	24	Test	Dcha	Andar	-0.25	-0.87	9.75	-3.25	12.5625	-43.5625	-2.125	0.875	-25.0	14.875	-0.875	-5.0	18 abr. 2016	22:43 26 075

Fig. 15. Base de datos obtenida en SQLite

2.4.3 Interfaz de usuario

Para poder monitorizar y controlar la toma de datos, se crea una interfaz de usuario. En la Fig. 16 se muestra la interfaz, la cual consta de dos campos de texto donde se introducen el nombre y la mano dominante de la persona, dos botones de START y STOP que activan y desactivan las notificaciones respectivamente, varios campos de texto donde se muestran datos del gir6scopo, del aceler6metro y el tiempo transcurrido en segundos desde que se pulsa START y, por 6ltimo, una serie de botones que se establecen qu6 actividad se est6 monitorizando.



Fig. 16. Interfaz de usuario

2.4.4 Interfaz de usuario secundaria

La visualización de los datos en la interfaz anterior no se realiza en tiempo real, sino que se almacenan en la base de datos para su posterior representación en gráficas. Esto hace que la verificación de los datos del sensor sea costosa.

Para solucionar este problema, se crea una nueva interfaz, mostrada en la Fig. 17, en la cual se puedan mostrar los datos de los sensores de manera gráfica y en tiempo real.



Fig. 17. Interfaz de usuario secundaria

Pese a que con esta interfaz no se puede indicar la actividad que se está realizando, permite observar si el sensor está bien calibrado o si las medidas están siendo tomadas correctamente.

2.4.5 Visualización en ordenador

Una vez finalizada la fase de experimentación, los datos registrados quedan almacenados en un archivo con extensión .db en el teléfono móvil. Para poder utilizar estos datos desde Matlab, se utiliza *PyCharm*, un entorno de trabajo para programar con Python, mediante el cual se puede acceder fácilmente a las bases de datos SQLite. En concreto, se lee la base de datos y se convierte a un formato denominado *DataFrame*, al cual se le elimina la columna de índice y se pasa a formato csv, fácilmente manipulable en el entorno de Matlab. El código utilizado se muestra a continuación:

```
#Convert from SQL to CSV

import sqlite3
import pandas.io.sql as sql

con =
sqlite3.connect('C:\\Users\\Alejandro\\Documents\\MATLAB\\Database.db');
table = sql.read_sql("SELECT * FROM Tabla", con);
del table["ID"];

table.to_csv('C:\\Users\\Alejandro\\Documents\\MATLAB\\BaseDatos.csv', header
= False, index=False);
```

Una vez se tengan los datos con las filas y columnas deseadas en formato csv, este archivo se abre con Matlab con el siguiente código:

```
%Extract SQLite Data

fid = fopen('BaseDatos.csv','r');
Data = textscan(fid,'%s %s %s %f %f %f %f %f %f %f %f %f %f %f %s %s',
'Delimiter',' ');
fid = fclose(fid);
```

En la función *textscan* se debe indicar el tipo con el que se desean interpretar los datos almacenados en el archivo csv. En este caso, solamente se han utilizado los tipos float (%f) y string (%s). La variable *Data* es de tipo *cell array*.

2.5 EXPERIMENTACIÓN

Los datos se recogerán mediante una aplicación móvil. En ella se establece el nombre, la mano dominante, se pulsa el botón con la actividad que se vaya a realizar y se pulsa START al comenzar ésta.

Se realizarán una serie de actividades en el orden que la persona prefiera, aunque a la hora de numerar las características, es preferible seguir un orden. La duración y repetición de estas dependerá de la complejidad de la actividad. Debido a esto, actividades repetitivas y continuas tendrán una duración más larga que las transicionales. En el caso de las actividades transicionales,

se toma una duración de 4 segundos para que no solo se capture la transición, sino también el movimiento previo y posterior a ésta. Las actividades son expuestas en la siguiente lista:

1. Andar: El sujeto andará durante 50 segundos por el recinto.
2. Levantado: El individuo permanecerá de pie y sin moverse durante 50 segundos, con brazos relajados, sin cruzarlos.
3. Levantado-Agachado: El sujeto se agachará tras permanecer de pie. La duración de la actividad será de 4 segundos y se repetirá 10 veces de forma no consecutiva, esto es, no encadenando el fin de la actividad con el inicio de la siguiente, sino haciendo una pequeña pausa.
4. Agachado-Levantado: Se realizará de la misma forma que la actividad anterior, invirtiendo el orden de las acciones.
5. Levantado-Sentado: El sujeto se sentará tras permanecer de pie. La duración de la actividad será de 4 segundos y se repetirá 10 veces de forma no consecutiva.
6. Sentado-Levantado: Se realizará de la misma forma que la actividad anterior, invirtiendo el orden de las acciones.
7. Levantado-Tumbado: El sujeto se tumbará tras permanecer de pie. La duración de la actividad será de 4 segundos y se repetirá 10 veces de forma no consecutiva.
8. Tumbado-Levantado: Se realizará de la misma forma que la actividad anterior, invirtiendo el orden de las acciones.
9. Saltando: El sujeto saltará 10 veces de forma no consecutiva.
10. Comer: El sujeto se llevará una cuchara/tenedor a la boca hasta 10 veces de forma no consecutiva.
11. Beber: El individuo beberá de un recipiente, como un vaso o una taza de manera normal. Esta actividad se realizará 10 veces de forma no consecutiva.
12. Manejar el ordenador: Consistirá en utilizar el ordenador la siguiente forma: 30 segundos utilizando el ratón y 30 segundos escribiendo un texto.
13. Barrer: El individuo barrerá durante 50 segundos.
14. Ver Televisión: Consistirá en sentarse, coger el mando y cambiar de canal o bajar/subir el volumen 10 veces de manera no consecutiva.
15. Abrir/Cerrar puerta: El individuo abrirá y cerrará una puerta 10 veces de manera no consecutiva y casual, como al entrar a una habitación.
16. Manejar tablet: Consistirá en utilizar la tablet de la siguiente forma: 30 segundos tecleando un texto y 30 arrastrando el dedo o haciendo clic.
17. Hablar por teléfono móvil: El individuo cogerá el teléfono lo mantendrá pegado a la oreja durante 50 segundos.

Durante el experimento se anotan posibles irregularidades ocurridas durante su desarrollo, para poder ser corregidas posteriormente. Si una actividad se ha realizado de manera errónea, puede eliminarse o puede prolongarse su realización durante más tiempo y desechar el fragmento erróneo.

Las actividades las llevarán a cabo 5 sujetos diestros de diferentes edades, de 22 a 58 años.

2.6 SELECCIÓN DE CARACTERÍSTICAS

Para la obtención de características se utilizan tres componentes inerciales, aceleración, ángulos de Euler y velocidad angular. Los datos en crudo de cada uno de estos se dividen en ventanas de 4 segundos y se extraen las características expuestas en la Tabla IX.

<i>Características extraídas</i>	<i>VARIABLES UTILIZADAS</i>
<i>Media</i>	Datos en crudo/ Derivada primera
<i>Varianza</i>	Datos en crudo/ Derivada primera
<i>Máximo</i>	Datos en crudo/ Derivada primera/ Módulos
<i>Mínimo</i>	Datos en crudo/ Derivada primera/ Módulos
<i>Diferencia entre máximo y mínimo</i>	Datos en crudo/ Derivada primera
<i>Módulo</i>	Ejes XY, XZ e YZ
<i>Error cuadrático medio</i>	Datos en crudo, ejes XY, XZ e YZ.
<i>Máximo absoluto</i>	Datos en crudo/ Derivada primera
<i>Mínimo absoluto</i>	Datos en crudo/ Derivada primera
<i>Diferencia entre máximo y mínimo absolutos</i>	Datos en crudo/ Derivada primera
<i>Área bajo la curva</i>	Datos en crudo
<i>Percentiles</i>	Datos en crudo/ Derivada primera
<i>Arcotangente</i>	Datos en crudo

TABLA IX. Características extraídas

La elección del tamaño de la ventana viene dado por las actividades transicionales “Tumbarse-Levantarse” o “Levantarse-Tumbarse”, que tienen una duración de alrededor de 4 segundos, ya que de otra forma no se podrían clasificar correctamente dichas actividades. Dicha justificación también se da en [9]. Además, la ventana tendrá un solapamiento de un 50% en todas las actividades salvo en las transicionales, para que la red sea capaz de generalizar de mejor forma.

No se utilizarán todas las características, pues ciertas actividades solamente requieren de una o dos características para poder clasificarse de manera precisa. Por ejemplo, en la Fig. 18 se observa que la actividad saltar se puede clasificar notablemente con solamente el mínimo de la aceleración en el eje Y. Debido a esto, se procede a hacer un análisis visual de todas las variables gráficamente, observando cuáles son más relevantes para clasificar cierto tipo de actividades.

Una vez se hayan elegido las características, se entrenarán las redes neuronales mediante validación cruzada. En concreto, se utilizará el método denominado *Leave-one-subject-out*, el cual consiste en dividir los datos en tantos grupos como individuos haya.

En base al desempeño obtenido, se seleccionarán unas características u otras. Este análisis podría ser realizado mediante métodos como los algoritmos genéticos, no obstante, se ha optado por el análisis visual de los datos debido a que ha proporcionado resultados suficientemente buenos y a que los algoritmos genéticos son computacionalmente muy exigentes y se necesitaría entrenar un gran número de redes.

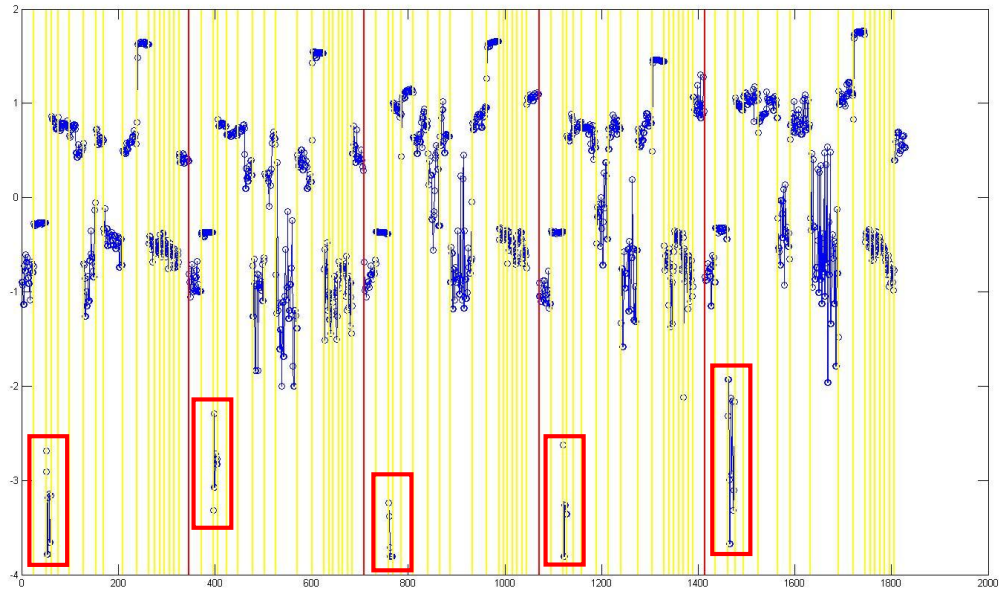


Fig. 18. Mínimos de la aceleración en Y, con líneas amarillas separando actividades y líneas rojas separando individuos

A la hora de seleccionar características, no solo se tiene en cuenta el desempeño. Debido a la implementación en un microcontrolador, tendrán mayor prioridad aquellas características que sean más fáciles de procesar. De hecho, se ha evitado la utilización de variables pertenecientes al dominio de la frecuencia, debido a que para su obtención se requeriría aplicar la transformada rápida de Fourier, con el tiempo de cálculo que conlleva.

2.7 ALGORITMO DE CLASIFICACIÓN

En el presente proyecto, se han utilizado redes neuronales para clasificar las actividades, en concreto el modelo MLP (*MultiLayer Perceptron*), dado que es un método robusto que permite dividir el espacio de muestras entre diferentes clases mediante una frontera no lineal [26]. Además, como se ha establecido en la literatura [30], [39], las redes neuronales son algoritmos adecuados para ser implementados en un microcontrolador con recursos limitados, frente a otros métodos como *k-Nearest Neighbours* (k-NN), que requieren grandes cantidades de memoria, solo disponibles en computadores [26], [30], [39].

En este trabajo se han utilizado diferentes redes MLP para cada una de las actividades para poder realizar una clasificación escalonada. Por ejemplo, en primer lugar, se determina si la actividad es “Saltar” con una red MLP pequeña, dado que dicha actividad no necesita de muchas variables de entrada. Si se obtiene que la muestra no pertenece a dicha actividad, esta pasa a otra red, que hará otro proceso de cribado.

El entrenamiento también se realiza de forma escalonada. Para entrenar la primera red de cribado, se utilizan todas las muestras. La próxima red de cribado solo será entrenada con las muestras que

no pertenezcan a la actividad clasificada anteriormente. Así, si la primera red clasifica la actividad “Saltar”, se utilizarán todas las muestras. Sin embargo, si la segunda clasifica por ejemplo “Beber”, se utilizarán todas las muestras salvo aquellas que pertenezcan a la actividad “Saltar”.

Para el entrenamiento de los MLP se ha utilizado el algoritmo de Levenberg–Marquardt, con 10 repeticiones por red y método *Leave-one-subject-out*. De este modo al haber 5 individuos y producirse 10 repeticiones por red, se entrenan en total 50 redes neuronales. Las redes finales a implementar en el microcontrolador serán entrenadas con todas las muestras disponibles y con las características más significativas.

2.8 RESULTADOS

A continuación, se muestran los desempeños obtenidos en la clasificación de diversas actividades y las características utilizadas. En la matriz de confusión de cada actividad (por ejemplo, “saltar”) se indica primero los aciertos para el resto de actividades (“no saltar”) y a continuación los correspondientes a la actividad que se analiza (“saltar”), de ahí que el número de casos aparezca desbalanceado (hay más casos “no saltar” que “saltar”).

2.8.1 Saltar

Características utilizadas	Variación y mínimo de aceleración en Y
Índice Kappa	0.9706
Número de capas ocultas	1
Neuronas capa oculta	1
Matriz Confusión (No saltar/Saltar)	1532 3 0 51

2.8.2 Hablar por teléfono

Características utilizadas	Media de aceleración en X y media de Euler Y
Índice Kappa	0.9692
Número de capas ocultas	1
Neuronas capa oculta	1
Matriz Confusión (No Hablar tlf./Hablar tlf.)	1291 4 3 121

2.8.3 Levantado

Características utilizadas	Máximo de aceleración en Y
Índice Kappa	0.9830
Número de capas ocultas	1
Neuronas capa oculta	1
Matriz Confusión (No levantado/Levantado)	1583 1 3 124

2.8.4 Andar

Características utilizadas	Máximo de aceleración en Y y percentil 25 de la derivada primera de la aceleración en Y.
Índice Kappa	0.9825
Número de capas ocultas	1
Neuronas capa oculta	2
Matriz Confusión (No andar/Andar)	1406 0 4 122

2.8.5 Barrer

Características utilizadas	Media de aceleración en eje X y media de Euler Y
Índice Kappa	0.9456
Número de capas ocultas	1
Neuronas capa oculta	2
Matriz Confusión (No barrer/Barrer)	930 6 6 119

2.8.6 Abrir-Cerrar puerta

Características utilizadas	Mínimo de Euler X. Mediana de Euler X. Percentil 25 de aceleración en Z. Media de Euler Z y Euler X. Máximo de giróscopo en Z. Diferencia máximo-mínimo de derivadas de Euler X.
Índice Kappa	0.9754
Número de capas ocultas	1
Neuronas capa oculta	4
Matriz Confusión (No Abrir-Cerrar puerta/Abrir-Cerrar puerta)	1059 7 2 217

2.8.7 Leer/Pasar página

Tras numerosos intentos, no se ha podido clasificar esta actividad con una precisión superior al 90%. Esto es debido a que la posición de la mano al leer varía mucho entre personas. Además, el tamaño del libro, o si se trata de una revista, puede afectar a la clasificación.

2.8.8 Transicionales

Las actividades transicionales se agrupan para poder ser clasificadas por separado. La red que permite separar las actividades transicionales del resto es la siguiente:

Características utilizadas	Media de aceleración en X e Y. Medias de derivadas de aceleración en Y y Z
Índice Kappa	1.0000
Número de capas ocultas	1
Neuronas capa oculta	2
Matriz Confusión (No transicional/Transicional)	637 0 0 299

Una vez separadas, se buscan características para poder separar las distintas actividades transicionales, no obstante, no se obtiene un buen resultado, ya que existen muestras muy similares entre las actividades transicionales que incluyen “Sentarse” y “Tumbarse”, como se muestra a modo de ejemplo en la Fig. 19. Es por esto que se decide agrupar las tres transicionales en un solo grupo.

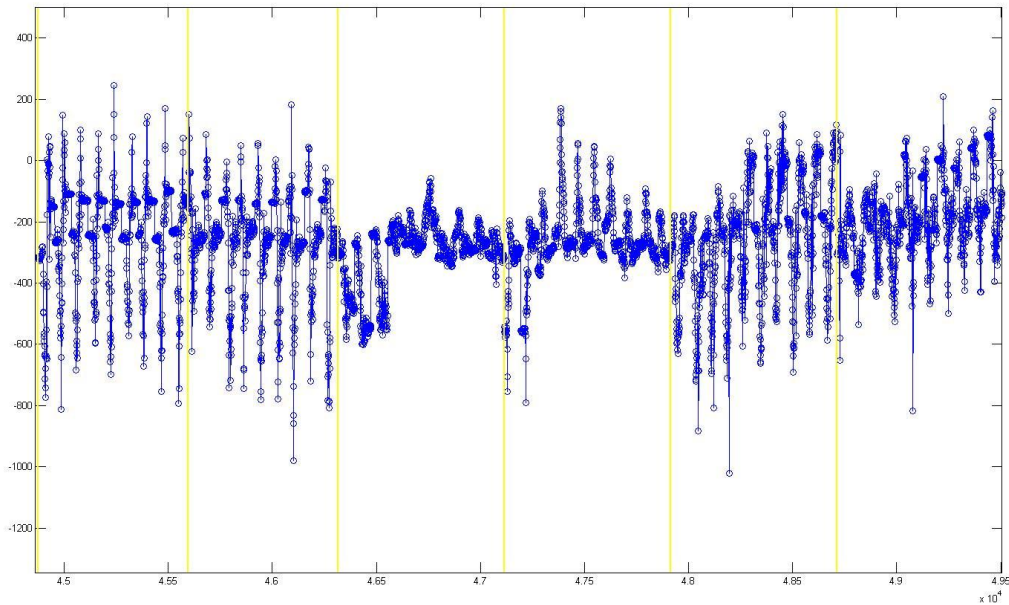


Fig. 19. Aceleración en el eje X en las siguientes actividades separadas mediante líneas amarillas: levantado-agachado, agachado-levantado, levantado-sentado, sentado-levantado, levantado-tumbado y tumbado-levantado

2.8.9 Resto de actividades

El resto de actividades son las siguientes: Comer/Ver TV, beber y manejar el PC/Manejar Tablet.

Características utilizadas	Media de aceleraciones. Media de EUL X e Y. Máximo de aceleraciones. Mínimo de aceleraciones. Mínimo de EUL en X e Y. Diferencia máximo y mínimo de giróscopo en X. Máximo de giróscopo en X. Mínimo de giróscopo en Z.									
Índice Kappa	0.9398									
Número de capas ocultas	1									
Neuronas capa oculta	5									
Matriz Confusión (Comer-Ver TV/Beber/Manejar PC- Tablet)	<table> <tr> <td>198</td> <td>1</td> <td>10</td> </tr> <tr> <td>3</td> <td>119</td> <td>4</td> </tr> <tr> <td>6</td> <td>0</td> <td>296</td> </tr> </table>	198	1	10	3	119	4	6	0	296
198	1	10								
3	119	4								
6	0	296								

Debido a que los datos inerciales son muy similares entre las actividades ‘manejar Tablet’ y ‘manejar el PC’, están tienden a ser incorrectamente clasificadas. Es por esto que se decide agrupar ambas actividades, ya que ambas representan el uso de tecnología. De igual forma, las actividades “comer” y “ver TV” se agrupan, para ser clasificadas por separado:

Características utilizadas	Percentil 80 y 85 de atan(ACCY/ACCZ)				
Índice Kappa	0.8088				
Número de capas ocultas	1				
Neuronas capa oculta	2				
Matriz Confusión (Comer/Ver TV)	<table> <tr> <td>73</td> <td>13</td> </tr> <tr> <td>6</td> <td>115</td> </tr> </table>	73	13	6	115
73	13				
6	115				

2.8.10 Orden de las redes escalonadas

En base a los resultados obtenidos en el apartado anterior, en las que algunas redes han tenido que ser descartadas, las redes neuronales escalonadas resultantes se utilizarán en el siguiente orden:

1. Levantado
2. Saltar
3. Andar
4. Hablar por teléfono
5. Abrir-Cerrar Puerta
6. Barrer
7. Transicionales
8. Comer/Ver TV, beber y manejar PC-Tablet
9. Comer y ver TV

2.8.11 Características utilizadas

Las características elegidas para las redes neuronales se reúnen en la Tabla X. Se requieren en total 30 características.

ÍNDICE	NOMBRE DE CARACTERÍSTICA
1	Media ACC X
2	Media ACC Y
3	Media ACC Z
4	Media EUL X
5	Media EUL Y
6	Media EUL Z
7	Varianza ACC Y
8	Máximo ACC X
9	Máximo ACC Y
10	Máximo ACC Z
11	Máximo GYR X
12	Mínimo ACC X
13	Mínimo ACC Y
14	Mínimo ACC Z
15	Mínimo GYR Z
16	Mínimo EUL X
17	Mínimo EUL Y
18	Diferencia máx/mín GYR X
19	Percentil 25 ACC Z
20	Percentil 50 EUL X
21	Media derivada ACC Y
22	Media derivada ACC Z
23	Varianza derivada ACC Y
24	Varianza derivada ACC Z
25	Máximo GIR Z
26	Percentil 25 derivada ACC Y
27	Diferencia máx/mín EUL X
28	Percentil 80 de atan(ACCY/ACCZ)
29	Percentil 85 de atan(ACCY/ACCZ)
30	Media ACC X

TABLA X. Características utilizadas

2.9 ALGORITMO EMBEBIDO

Los algoritmos desarrollados en MATLAB se implementan a continuación en código C para poder ser incluidos en el PSoC 4200 BLE de *Cypress*. El código principal consiste en una máquina de estados, la cual se muestra en la Fig. 20.

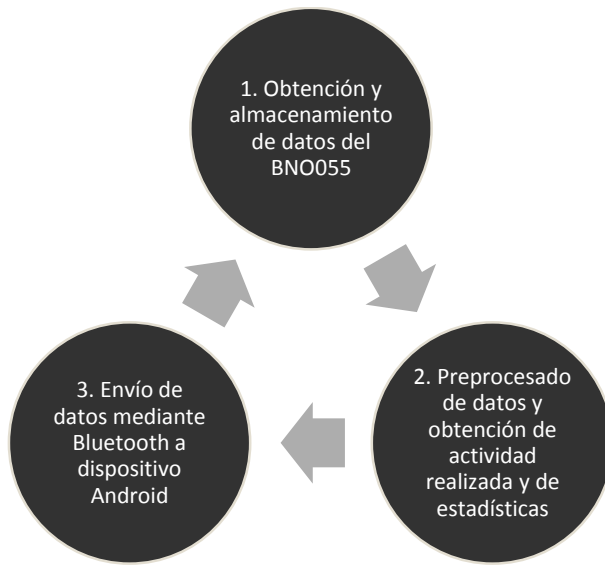


Fig. 20. Máquina de estados

2.9.1 Obtención y almacenamiento de datos del BNO055

La comunicación mediante el microcontrolador y el integrado BNO055 se realiza mediante protocolo I2C, en el cual el microcontrolador actuará como maestro y el dispositivo de sensado como esclavo.

Los datos provenientes del BNO055 se obtienen en forma de enteros sin signo de 8 bits (1 byte). Éstos se almacenarán en la memoria del microcontrolador, ocupando un total de 1.6 KB, ya que los datos constan de 80 muestras (equivalentes a 4 segundos) de 10 datos inerciales con un tamaño de 2 bytes.

La estructura en la que se obtienen los datos se puede observar en la Tabla XI.

A	A	A	A	A	A	G	G	G	G	G	G	M	M	M	M	M	M	C	C	C	C	C	C	C	C
C	C	C	C	C	C	Y	Y	Y	Y	Y	Y	A	A	A	A	A	A	U	U	U	U	U	U	U	U
C	C	C	C	C	C	R	R	R	R	R	R	G	G	G	G	G	G	A	A	A	A	A	A	A	A
X	X	Y	Y	Z	Z	X	X	Y	Y	Z	Z	X	X	Y	Y	Z	Z	T	T	T	T	T	T	T	T
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	W	W	X	X	Y	Y	Z	Z
																		0	1	0	1	0	Z	0	1

TABLA XI. Trama de datos inerciales

Solamente se utilizarán los datos de aceleraciones, velocidades angulares y cuaternios. Las unidades del magnetómetro no son necesarias ya que la orientación es dada por los cuaternios. Finalmente se utilizan cuaternios debido a que el algoritmo de fusión implementado en el firmware del BNO055 presenta resultados inestables al transformar los cuaternios a ángulos de Euler en inclinaciones muy pronunciadas del ángulo de *pitch* (eje X) o del ángulo de *roll* (eje Y) [40]. Al transformar los cuaternios a ángulos de Euler de manera externa (en el propio microcontrolador)

mediante fórmulas trigonométricas, como se verá en el siguiente apartado, éste problema de inestabilidad desaparece.

Antes de poder almacenarse, los datos deben ser procesados para que representen el valor correcto. Los datos están codificados en Little Endian, es decir, los bytes menos significativos se envían primero. Por esta razón los bytes de cada variable deben ser intercambiados para que el dato sea consistente. Así pues, según la Tabla XI, el dato correcto de la aceleración sería: ACCX1-ACCX0.

Para conformar los 16 bits a partir de 2 valores uint8 se crea una función que junte los datos de 8 bits, obteniendo variables tipo *short*:

```
((short) (((short)x << 8) | (uint8) y))
```

Una vez modificados los datos y guardados en una variable se procede a obtener información de estos.

2.9.2 Preprocesado de datos y obtención de actividad realizada y de estadísticas

El preprocesado consiste en la extracción de las características indicadas en la Tabla X. Como se puede observar, ninguna característica pertenece al dominio frecuencial, por lo que no es necesario realizar transformaciones complicadas como la transformada rápida de Fourier. Las computaciones más costosas vienen dadas por las operaciones trigonométricas, utilizadas al hallar las características 28 y 29, además de los ángulos de Euler. En concreto, para la obtención de los ángulos de Euler a partir de los cuaternios, se utilizan las siguientes fórmulas:

$$EULX = \left(\frac{180}{\pi}\right) * \text{atan2}(2 * CUATY * CUATZ + 2 * CUATX * CUATW, 1 - 2 * CUATX^2 - 2 * CUATY^2) \quad (1)$$

$$EULY = \left(\frac{180}{\pi}\right) * \text{asin}(-2 * CUATX * CUATZ + 2 * CUATY * CUATW) \quad (2)$$

$$EULZ = \left(\frac{180}{\pi}\right) * \text{atan2}(2 * CUATX * CUATY + 2 * CUATZ * CUATW, 1 - 2 * CUATZ^2 - 2 * CUATY^2) \quad (3)$$

Las funciones trigonométricas utilizadas son las incluidas en la librería *math* de C. Se utilizan las funciones “atan2f” y “asinf”, por lo que se deberá realizar un *casting* de *short* a *float* para poder utilizar los cuaternios como argumentos en dichas funciones.

Con las variables tipo *short* (excepto ángulos de Euler, que son tipo *float*), se obtienen las diferentes características, de tipo *float*. Se utiliza este tipo para obtener una mayor precisión.

Una vez obtenidas las características, es necesario normalizarlas. Para ello se extrae la media y la desviación estándar de la base de datos obtenida para cada una de las características. Las características se normalizan (*z-scores* o *normal-scores*) mediante la siguiente fórmula:

$$feat_n = \frac{feat - media}{desv_est} \quad (4)$$

El siguiente paso consiste en la implementación del algoritmo de aprendizaje. Las redes neuronales son fácilmente implementables en un microcontrolador cuando están entrenadas, ya que requieren operaciones muy sencillas. Una red neuronal con dos entradas, dos neuronas ocultas con función

de activación tangente hiperbólica y dos salidas sigmoideas (para limitar las salidas al rango 0.0-1.0) se puede implementar con las siguientes operaciones:

$$\begin{bmatrix} b_{11}^{(1)} & w_{11}^{(1)} & w_{12}^{(1)} \\ b_{21}^{(1)} & w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 1 \\ i_1^{(1)} \\ i_2^{(1)} \end{bmatrix} = \begin{bmatrix} o_1^{(1)} \\ o_2^{(1)} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} i_1^{(2)} \\ i_2^{(2)} \end{bmatrix} = \begin{bmatrix} \tanh(o_1) \\ \tanh(o_2) \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} b_{11}^{(2)} & w_{11}^{(2)} & w_{12}^{(2)} \\ b_{21}^{(2)} & w_{21}^{(2)} & w_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 1 \\ i_1^{(2)} \\ i_2^{(2)} \end{bmatrix} = \begin{bmatrix} o_1^{(2)} \\ o_2^{(2)} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} salida_1 \\ salida_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-o_1^{(2)}}} \\ \frac{1}{1+e^{-o_2^{(2)}}} \end{bmatrix} \quad (8)$$

En la Fig. 21 se observa la red anterior de forma esquemática. La salida que posea un valor más alto determinará la actividad predicha.

Para realizar las operaciones de las redes neuronales se utilizará punto flotante con variables tipo *float*, ya que se compararon los resultados obtenidos con Matlab (con variables tipo *double*) con los obtenidos usando variables tipo *float*, y se obtuvieron las mismas salidas de las redes. Los pesos de la red neuronal también se codificarán en variables de tipo *float*. Por último, para realizar las funciones de la tangente hiperbólica y de la función sigmoidea, se utilizan las funciones de la librería *math* “*tanhf*” y “*exp*”.

Una vez obtenida la actividad, se registrarán además una serie de valores estadísticos:

- Número de pasos
- Número de sorbos
- Número de veces que se lleva un tenedor o cuchara a la boca
- Tiempo comiendo
- Tiempo bebiendo
- Tiempo usando ordenador/tablet

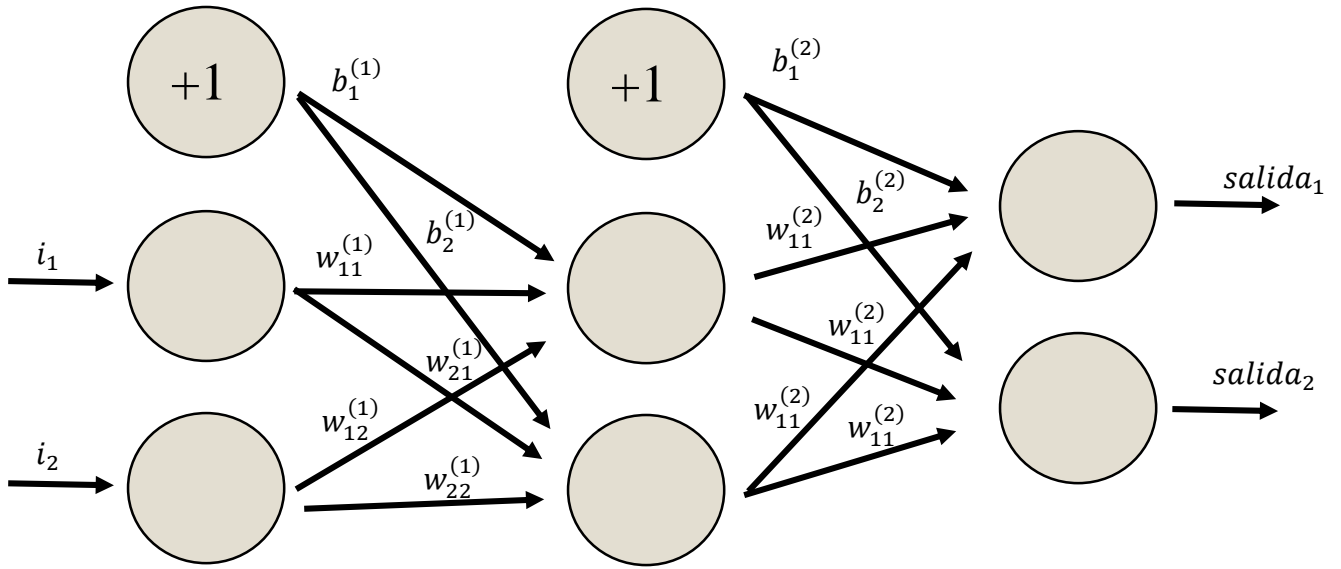


Fig. 21. Ejemplo de red neuronal MLP 2-2-2

Para contar el número de veces que se realizan ciertos movimientos, se utilizan comparadores con histéresis con los umbrales elegidos de forma empírica tras observar las señales en crudo. En las Fig. 22 a 24 se muestran las señales en crudo más representativas para las actividades “Andar”, “Beber” y “Comer”.

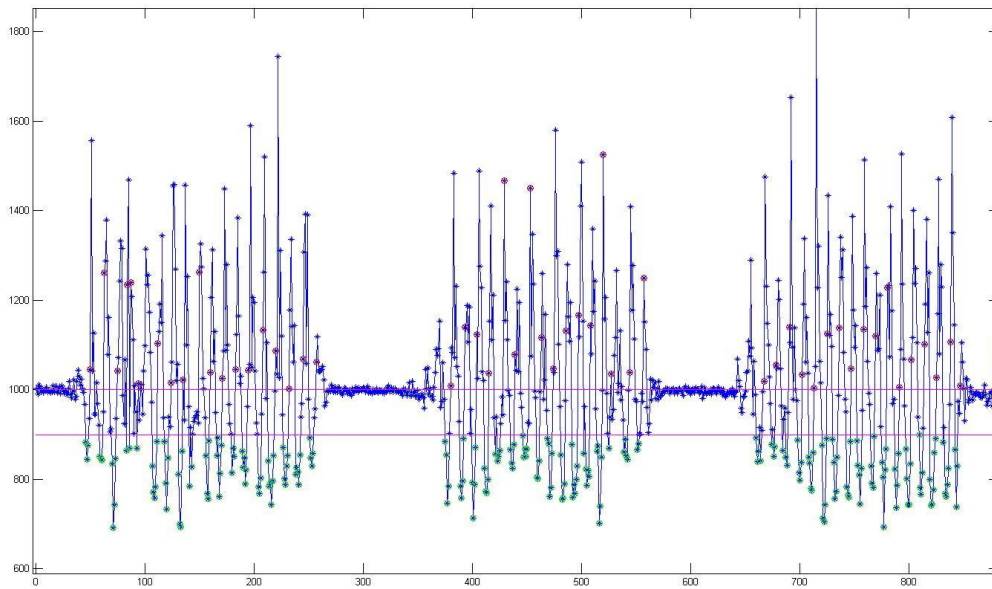


Fig. 22. Módulo de aceleraciones X e Y en la actividad “Andar”

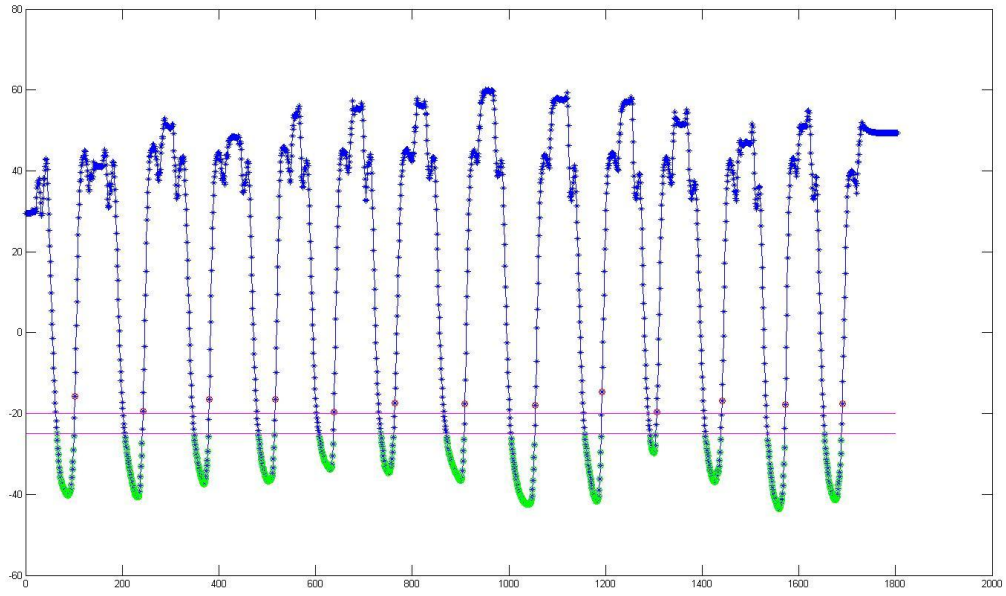


Fig. 23. Ángulo de Euler en Y en la actividad “Beber”

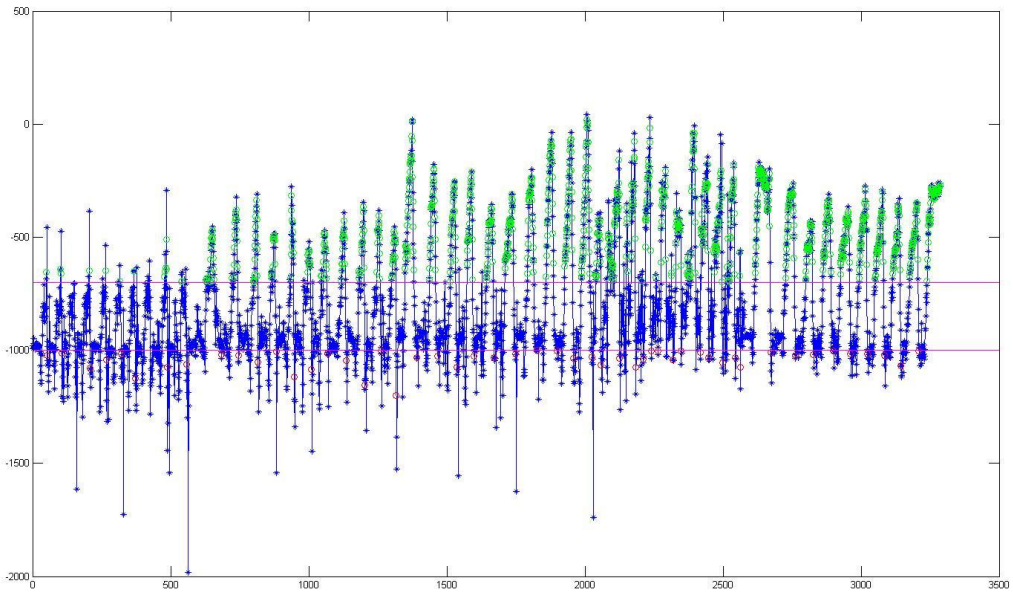


Fig. 24. Aceleración en X en la actividad “Comer”

Es necesario mencionar que en la actividad “Beber”, la aceleración en el eje X ha sido suavizada tomando las medias entre dos muestras consecutivas, ya que la señal tenía picos que podían distorsionar el número de sorbos calculado.

Para calcular el tiempo de cada actividad, simplemente se tiene en cuenta lo obtenido por las redes neuronales. Si las redes determinan que la ventana pertenece a una actividad, el tiempo de dicha actividad aumentará en 4 segundos, que es el tamaño de la ventana.

El número de pasos y cucharadas se obtienen de manera precisa, ya que en las señales correspondientes se aprecia de forma inequívoca en qué momento ocurren ambas acciones. No es ese el caso de la actividad “Andar”, en la que la señal es más rápida y oscilante y, por lo tanto, menos precisa. Se realizó una prueba monitorizando 126 pasos y se comprobó cómo con el método de umbrales se obtuvieron 120 pasos, obteniendo un error del 4.7%.

2.9.3 Envío de datos mediante Bluetooth a dispositivo Android

En la comunicación Bluetooth, el microcontrolador actuará como servidor GATT y como periférico. Para el envío de datos se utilizará un servicio personalizado denominado *Server_UART*, el cual constará de dos características, siendo la denominada *Server_UART_Tx_data* la encargada de enviar los datos al dispositivo Android, que actuará como cliente. En la Tabla XII se muestran los identificadores únicos del servicio y la característica comentados anteriormente.

<i>Server_UART</i> UUID	0000EEE1-0000-1000-8000-00805F9B34FB
<i>Server_UART_Tx_data</i> UUID	00002902-0000-1000-8000-00805F9B34FA

TABLA XII. Identificadores únicos de servicio y característica

La característica se configura para que pueda almacenar un array de 35 *uint8*, suficiente teniendo en cuenta que los datos enviados son 7 *uint8* correspondientes con la actividad y las estadísticas obtenidas en el apartado anterior.

2.10 VISUALIZACIÓN DE DATOS EN ANDROID

Para la visualización de datos, es necesario crear otra aplicación Android. Para ello se utiliza el mismo procedimiento que en el apartado 2.3. En lugar de recibirse los datos inerciales, estos son reemplazados por los datos de actividad y estadísticas.

La interfaz también se modifica de forma que se puedan visualizar adecuadamente todos los datos. El resultado se muestra en la Fig. 25.

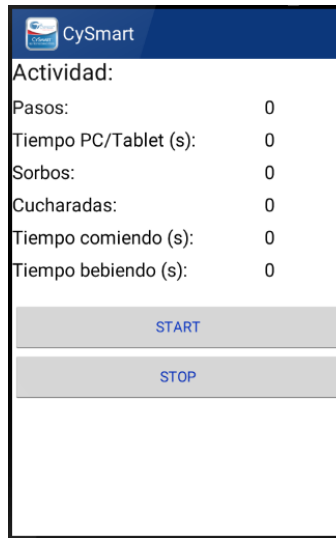


Fig. 25. Interfaz definitiva

Además de mostrar los datos en la interfaz, estos se guardan en la base de datos para su posterior consulta. La actividad y las estadísticas se complementan con la fecha, para poder situar en el tiempo el desarrollo de las actividades. La base de datos se muestra en la Fig. 26.

	ID	ACTIVIDAD	PASOS	SORBOS	CUCHARADAS	T_COMIENDO	T_BEBIENDO	T_PC_TABLET	FECHA
		Filter	...	Fil...	Filter	Filter	Filter	Filter	Filter
19	19	Manejando PC/Tablet	0	0	0	0	0	12	2016-08-04 18:44:11.445
20	20	Manejando PC/Tablet	0	0	0	0	0	12	2016-08-04 18:44:11.452
21	21	Manejando PC/Tablet	0	0	0	0	0	12	2016-08-04 18:44:11.458
22	22	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.549
23	23	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.555
24	24	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.563
25	25	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.568
26	26	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.574
27	27	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.579
28	28	Manejando PC/Tablet	0	0	0	0	0	16	2016-08-04 18:44:15.585
29	29	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.734
30	30	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.742
31	31	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.752
32	32	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.758
33	33	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.763
34	34	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.769
35	35	Estático	0	0	0	0	0	16	2016-08-04 18:44:19.775
36	36	Manejando PC/Tablet	0	0	0	0	0	20	2016-08-04 18:44:23.854
37	37	Manejando PC/Tablet	0	0	0	0	0	20	2016-08-04 18:44:23.869
38	38	Manejando PC/Tablet	0	0	0	0	0	20	2016-08-04 18:44:23.874

Fig. 26. Base de datos obtenida en SQLite para monitorización

2.11 RESULTADOS DEL SISTEMA EMBEBIDO

El algoritmo desarrollado se puede implementar en el PSoC ya que no sobrepasa los límites de memoria RAM o memoria Flash. En ambas memorias ha quedado un espacio libre de en torno a un 18% tal y como se muestra en la Fig. 27. Un algoritmo más complejo, o que conlleve un mayor gasto de memoria, como k-NN, posiblemente hubiera superado el límite de memoria.



Memory	
Flash	82.5 %
SRAM	82.2 %

Fig. 27. Consumo de memoria

Se analiza el tiempo de cálculo requerido para procesar los datos y ejecutar las redes neuronales, es decir, el estado 2 de la máquina de estados expuesta anteriormente. Para ello se utiliza un “Timer” del PSoC, el cual al llegar a un valor determinado hará saltar una interrupción y se reiniciará. En la interrupción se habilita un contador para estimar el tiempo empleado. Tras testear el algoritmo, se establece que el tiempo de ejecución es de aproximadamente 90 ms.

En cuanto a la clasificación, el *wearable* presenta ciertas discrepancias respecto a las obtenidas de forma simulada en Matlab. A continuación se explica la precisión de cada actividad clasificada en el *wearable*.

2.11.1 Levantado

Esta actividad presenta un buen nivel de precisión. No obstante, si en la ventana de datos se captura algún movimiento brusco además de estar levantado, es probable que la actividad obtenida sea incorrecta.

2.11.2 Saltar

Cuando una persona salta, se produce una gran aceleración en el eje Y, lo cual hace que sea una actividad muy característica y que se diferencie con facilidad respecto a otras. Es por esto, que la precisión es buena.

2.11.3 Andar

Andar también presenta buenos resultados. No obstante, al igual que en la actividad “Levantado”, si en la ventana se captura algún otro movimiento, la actividad obtenida puede ser errónea. Respecto al número de pasos dados, se recoge correctamente el número de pasos y lo actualiza cada 4 segundos.

2.11.4 Hablar por teléfono

Utilizar el teléfono fijo o móvil también es una actividad muy característica. La posición de la mano al ponerse el teléfono en la oreja es claramente diferenciada utilizando ángulos de Euler.

2.11.5 Abrir/Cerrar puerta

Esta actividad presenta discrepancias respecto a lo obtenido en Matlab. En la simulación, se obtenía una precisión superior al 90%, sin embargo al implementarlo en el *wearable*, ésta actividad no es capturada correctamente por la ventana y se confunde en la mayoría de las ocasiones con las actividades transicionales. Esto puede deberse a que a la hora de elaborar la base de datos, no se haya capturado la actividad de forma natural, ya que en la base de datos el abrir o cerrar la puerta se hacía desde una posición estática, y en un contexto doméstico una persona abre o cierra puertas mientras está en movimiento, mientras anda.

2.11.6 Barrer

Barrer es una actividad con gran precisión. Esto es debido a que los bruscos movimientos al mover una escoba o similares son muy característicos.

2.11.7 Transicionales

Como se ha explicado anteriormente, las actividades transicionales incluyen aquellas que ocurren entre actividades estacionarias, como estar sentado, tumbado, agachado, etc. Dichas actividades son clasificadas correctamente por el dispositivo. No obstante, un movimiento brusco al andar también puede ser interpretado como una actividad transicional incorrectamente.

2.11.8 Beber

Esta actividad presenta una precisión notable cuando el recipiente del que se bebe es un vaso, ya que la red neuronal ha sido entrenada con personas bebiendo de un vaso de agua. Con otro tipo de recipientes, como una botella de agua, la precisión empeora. Esto es debido a que el recorrido de la mano hacia la boca no es el mismo. Por ejemplo, una botella muy larga hará que apenas se tenga que acercar la mano a la boca.

El número de sorbos calculado por el algoritmo es correcto y coincide con el real.

2.11.9 Comer

La actividad “Comer” presenta una buena precisión. No obstante, presenta ciertos falsos negativos, ya que en ocasiones, dependiendo del movimiento, se puede confundir con la actividad “Beber”.

En las ocasiones en que la ventana es clasificada correctamente, el número de cucharadas calculadas por el algoritmo es el correcto.

2.11.10 Ver TV

Esta actividad puede presentar muchas variaciones, como la distancia de la televisión, la forma de coger el mando, etc. En el contexto en el que se ha entrenado, el algoritmo clasifica correctamente la actividad, aunque con algunos falsos negativos con la actividad “Comer”.

2.11.11 Utilizar Ordenador/Tablet

Utilizar el ordenador o la tablet se reconoce correctamente en la mayor parte de las ocasiones, no obstante, el *wearable* también indica que se está realizando la actividad cuando la persona está simplemente sentada en una silla. Para solucionar esto, se cambia ligeramente el algoritmo: Si la actividad es clasificada como “Utilizar Ordenador/Tablet”, entonces se calcula si el giróscopo en el eje X supera cierto valor umbral un número determinado de veces. Dicho umbral se establece para determinar el uso del ratón o del teclado del dispositivo, ya que el movimiento de la mano hace que el giróscopo cambie respecto al valor que tendría en una actividad estática. En concreto, el umbral se establece en 20 grados por segundo y el número de veces que se tiene que superar el umbral en 13.

3 CONCLUSIONES

De los resultados expuestos podemos concluir que la precisión del algoritmo clasificador cuando se aplica de forma “on-line” sobre datos reales en algunos casos difiere de la alcanzada con la base de datos usada en el desarrollo. Así, de las 11 actividades a clasificar, se puede asegurar una precisión elevada en 6 de ellas: “Levantado”, “Saltar”, “Andar”, “Hablar por teléfono”, “Barrer”, “Beber” y “Utilizar ordenador/tablet”. Estas actividades presentan ciertas características que las hacen más fáciles de clasificar, como el hecho de que sean repetitivas, como en el caso de “Andar” o “Barrer”, o que presenten un valor constante y diferenciado al resto de actividades, como ocurre en el caso de “Hablar por teléfono” o “Levantado”, en las que la posición de la mano es constante. Presentan dificultades para su clasificación actividades cuyo tiempo de ejecución sea corto comparado con la ventana de 4 segundos, como el caso de “Comer” o “Ver TV”.

Cuando no se realiza ninguna de las 11 actividades, el algoritmo dará como resultado la actividad que más se parezca al movimiento realizado. Habitualmente, los movimientos rápidos se clasifican como una actividad transicional y los que incluyen pequeños movimientos de la mano como “Utilizar ordenador/tablet”.

La muñeca demuestra ser una posición adecuada para gran parte de las actividades realizadas en el hogar. Sin embargo, es susceptible a irregularidades, ya que por ejemplo un movimiento errático de la mano puede distorsionar la clasificación final. Es por esto que puede resultar interesante utilizar sensores adicionales colocados en el torso, en la pierna o en la cintura, ya que los movimientos de estas partes del cuerpo son más limitados, además de más adecuados para ciertas actividades, como andar, sentarse, levantarse, etc.

El sensor podría ver aumentada su eficacia si estuviese complementado con un sensado situado en la infraestructura del hogar. Esto es debido a que el lugar donde se realicen las actividades es importante a la hora de determinar el proceso cognitivo de la persona. Por ejemplo, si el sujeto se tumba en el suelo, esto puede indicar que está sufriendo algún tipo de problema.

La recopilación de estadísticas de los movimientos, aun siendo precisa, es susceptible a errores, ya que si una actividad se ha clasificado incorrectamente, el algoritmo almacenará estadísticas de una actividad que no está realizando. Otro de los inconvenientes es la visualización, ya que debido al

tamaño de la ventana, los datos se actualizan cada 4 segundos. Es por esto que si no se desea ver los datos en tiempo real, los resultados se almacenan en una base de datos para su posterior estudio.

Como futuras líneas de trabajo se podrían implementar otros algoritmos, como k-NN, para poder clasificar de mejor forma actividades cuyo movimiento es característico y rápido, como puede ocurrir al leer un libro o abrir una puerta, permitiendo ampliar el reconocimiento a un mayor rango de actividades. Además, se podrían integrar los algoritmos directamente en un reloj compatible con el sistema operativo Android. Otra posibilidad digna de un futuro estudio es aplicar la misma metodología cuando el dispositivo se coloca en otras zonas del cuerpo, como la cintura. Finalmente, indicar que el algoritmo solo ha sido entrenado con personas diestras, por lo que éste podría ser ampliado con una base de datos que incluya también personas zurdas.

4 REFERENCIAS

- [1] T. H. Tan, M. Gochoo, K. H. Chen, F. R. Jean, Y. F. Chen, F. J. Shih, and C. F. Ho, "Indoor activity monitoring system for elderly using RFID and Fitbit Flex wristband," *2014 IEEE-EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2014*, pp. 41–44, 2014.
- [2] S. Chernbumroong, "Multi-sensor Activity Recognition of An Elderly Person," Bournemouth, 2015.
- [3] P. Kumar and P. C. Pandey, "A wearable inertial sensing device for fall detection and motion tracking," *2013 Annu. IEEE India Conf.*, pp. 1–6, 2013.
- [4] S. Im, I. J. Kim, S. C. Ahn, and H. G. Kim, "Automatic ADL classification using 3-axial accelerometers and RFID sensor," *IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, pp. 697–702, 2008.
- [5] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Pervasive Comput.*, pp. 1 – 17, 2004.
- [6] A. Moncada-Torres, K. Leuenberger, R. Gonzenbach, A. Luft, and R. Gassert, "Activity classification based on inertial and barometric pressure sensors at different anatomical locations," *Physiol. Meas.*, vol. 35, pp. 1245–63, 2014.
- [7] M. Pijl, S. van de Par, and C. Shan, "An event-based approach to multi-modal activity modeling and recognition," *IEEE Int. Conf. Pervasive Comput. Commun.*, pp. 98–106, 2009.
- [8] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Comput.*, vol. 19, no. 5, pp. 26–35, 2015.
- [9] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, 2014.
- [10] N. Roy, A. Misra, and D. Cook, "Infrastructure-assisted smartphone-based ADL recognition in multi-inhabitant smart environments," *2013 IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2013*, no. March, pp. 38–46, 2013.
- [11] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," *2015 IEEE 12th Int. Conf. Wearable Implant. Body Sens. Networks*, pp. 1–6, 2015.
- [12] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition," *2014 IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2014*, pp. 51–59, 2014.
- [13] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, 2010.
- [14] L. S. Noelker and R. Browdie, "Sidney Katz, MD: A new paradigm for chronic illness and long-term care," *Gerontologist*, vol. 54, no. 1, pp. 13–20, 2014.
- [15] J. R. Hall, H. T. Vo, L. a Johnson, R. C. Barber, and S. E. O'Bryant, "The Link between Cognitive Measures and ADLs and IADL Functioning in Mild Alzheimer's: What Has Gender Got to Do with It?," *Int. J. Alzheimers. Dis.*, vol. 2011, p. 276734, 2011.
- [16] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, 2008.
- [17] F. R. Allen, E. Ambikairajah, N. H. Lovell, and B. G. Celler, "Classification of a known

- sequence of motions and postures from accelerometry data using adapted Gaussian mixture models,” *Physiol. Meas.*, vol. 27, no. 10, pp. 935–951, 2006.
- [18] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kr, “Accurate Activity Recognition in a Home Setting,” *UbiComp '08 Proc. 10th Int. Conf. Ubiquitous Comput. ACM, Seoul, Korea*, pp. 1–9, 2008.
- [19] R. C. Wagenaar, I. Sapir, Y. Zhang, S. Markovic, L. M. Vaina, and T. D. C. Little, “Continuous monitoring of functional activities using wearable, wireless gyroscope and accelerometer technology,” *Eng. Med. Biol. Soc. 2011 Annu. Int. Conf. IEEE*, pp. 4844–4847, 2011.
- [20] Y. Saeys, I. Inza, and P. Larrañaga, “A review of feature selection techniques in,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [21] N. A. Capela, E. D. Lemaire, and N. Baddour, “Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients,” *PLoS One*, vol. 10, no. 4, pp. 1–18, 2015.
- [22] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” *Int. Conf. Mach. Learn.*, pp. 1–8, 2003.
- [23] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.
- [24] M. Sekine, T. Tamura, M. Ogawa, T. Togawa, and Y. Fukui, “Classification of acceleration waveform in a continuous walking record,” *Proc. 20th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Vol20 Biomed. Eng. Towar. Year 2000 Beyond Cat No98CH36286*, vol. 3, no. 3, pp. 1523–1526, 1998.
- [25] Seon-Woo Lee and K. Mase, “Activity and location recognition using wearable sensors,” *IEEE Pervasive Comput.*, vol. 1, no. 3, pp. 24–32, 2002.
- [26] R. del-Hoyo, B. Martín-del-Brío, N. Medrano, and J. Fernández-Navajas, “Computational intelligence tools for next generation quality of service management,” *Neurocomputing*, vol. 72, no. 16, pp. 3631–3639, 2009.
- [27] O. D. Lara and M. a. Labrador, “A Survey on Human Activity Recognition using Wearable Sensors,” *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [28] a a M. Abushariah, T. S. Gunawan, O. O. Khalifa, and M. a M. Abushariah, “English digits speech recognition system based on Hidden Markov Models,” *Comput. Commun. Eng. ICCCE 2010 Int. Conf.*, no. May, pp. 1–5, 2010.
- [29] A. Mannini and A. M. Sabatini, “Machine learning methods for classifying human physical activity from on-body accelerometers,” *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.
- [30] A. Bono-Nuez, C. Bernal-Ruíz, B. Martín-del-Brío, F. J. Pérez-Cebolla, and A. Martínez-Iturbe, “Recipient size estimation for induction heating home appliances based on artificial neural networks,” *Neural Comput. Appl.*, 2016.
- [31] L. Breiman, “Bagging Predictors,” *Mach. Learn.*, vol. 24, pp. 123–140, 1996.
- [32] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.
- [33] L. Breiman, “Stacked regressions,” *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
- [34] T. P. Kao, C. W. Lin, and J. S. Wang, “Development of a portable activity detector for daily activity recognition,” *IEEE Int. Symp. Ind. Electron.*, no. ISIE, pp. 115–120, 2009.
- [35] S. Balakrishnama and a Ganapathiraju, “Linear Discriminant Analysis - a Brief Tutorial,” *Compute*, vol. 11, pp. 1–9, 1998.

- [36] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier, "Hierarchical, Multi-Sensor Based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms Using a Benchmark Dataset," *PLoS One*, vol. 8, no. 10, 2013.
- [37] Bosch Sensortec GmbH, "Data Sheet BNO055 - Intelligent 9-axis absolute orientation sensor," no. November, 2014.
- [38] D. N. R. B, C. Court, and S. Jose, "CySmart™ Android App User Guide," no. 001, pp. 1–36, 1810.
- [39] B. Martin-Del-Brio and C. Bernal-Ruiz, "A software tool for teaching microcontroller system principles," *Int. J. Electr. Eng. Educ.*, vol. 36, no. 4, pp. 279–286, 1999.
- [40] "BNO055 - Euler angles broken?," 2016. [Online]. Available: <https://forums.adafruit.com/viewtopic.php?f=19&t=88600>.