



ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN

**TRABAJO FIN DE MÁSTER**

INGENIERO DE TELECOMUNICACIÓN

INTERVALOS DE NORMALIDAD EN  
MEDIDAS DE ACTIMETRÍA: UN ESTUDIO  
PARA DIFERENTES TIPOS DE  
ACTÍMETROS

---

AUTOR: ELENA MARTÍN GONZÁLEZ

TUTORES: CARLOS ALBEROLA LÓPEZ, RODRIGO DE LUIS GARCÍA

Julio de 2017

---

TÍTULO: INTERVALOS DE NORMALIDAD EN MEDIDAS DE ACTIMETRÍA: UN ESTUDIO PARA DIFERENTES TIPOS DE ACTÍMETROS

AUTOR: ELENA MARTÍN GONZÁLEZ

TUTORES: CARLOS ALBEROLA LÓPEZ, RODRIGO DE LUIS GARCÍA

DEPARTAMENTO: TSCIT

---

**TRIBUNAL**

---

PRESIDENTE: MARCOS A. MARTÍN FERNÁNDEZ

SECRETARIO: JUAN IGNACIO ARRIBAS SÁNCHEZ

VOCAL: JUAN PABLO DE CASTRO FERNÁNDEZ

---

FECHA:

CALIFICACIÓN:

---

## **RESUMEN**

La actimetría es un método de monitorización de actividad no invasivo que está tomando cada vez más protagonismo en la vida cotidiana como marcador de vida saludable. Sin embargo, existen en el mercado diferentes tipos de actímetros, pero sólo algunos de ellos cumplen los requisitos necesarios para su validez en el diagnóstico clínico.

En este Trabajo Fin de Máster se comparan características extraídas mediante estos distintos tipos de actímetros. A fin de hacer comparables los datos proporcionados por un actímetro válido en el ámbito clínico, en este caso el wGT3X-BT de Actigraph, y los datos extraídos de un dispositivo comercial, la pulsera inteligente Microsoft Band 2, se aproximan los datos de actividad (*counts*) del primero a partir de los datos de aceleración del segundo. Para ello, fue necesario el desarrollo de una aplicación móvil que permitiese extraer los datos de aceleración del actímetro comercial. Una vez extraídos dichos datos se llevó a cabo un preprocesado consistente en un remuestreo y alineación de los mismos de manera que las señales quedaran acondicionadas para su posterior procesado.

La aproximación de *counts* a partir de los datos de aceleración se lleva a cabo mediante aprendizaje automático, analizando los resultados obtenidos mediante redes neuronales y bosques aleatorios.

Finalmente, se extraen características de los datos de ambos actímetros, siendo éstas comparables.

## **PALABRAS CLAVE**

Actimetría, *counts*, ActiGraph, Microsoft Band 2, intervalos de normalidad.

## **ABSTRACT**

Actimetry is a method to monitor activity in a non-invasive manner that is increasingly taking a leading role in everyday life as a marker of healthy life. However, there are different types of actimeters on the market, but only some of them meet the requirements for their validity in clinical diagnosis.

In this Master's Thesis we compare features extracted by these different types of actimeters. In order to make the data provided by a valid actimeter in the clinical field, in this case Actigraph's wGT3X-BT, comparable to data extracted from a commercial device, the Microsoft Band 2 smartband, the activity data from the first one (counts) are approximated from the acceleration data of the second. For this, it was necessary to develop a mobile application that allowed to extract the data of acceleration of the commercial actimeter. Once this data was extracted a preprocessing consisting of a resampling and alignment was carried out so that the signals were conditioned for further processing.

The approximation of counts from the acceleration data is performed through machine learning, analyzing the results obtained through neural networks and random forests.

Finally, data characteristics of both actimeters are extracted, being these comparable.

## **KEYWORDS**

*Actimetry, counts, ActiGraph, Microsoft Band 2, intervals of normality.*

# AGRADECIMIENTOS

---

En primer lugar, quisiera dar las gracias a mis tutores, Carlos Alberola López y Rodrigo de Luis García, por todo el tiempo que me han concedido.

También me gustaría hacer especial mención al Laboratorio de Procesado de Imagen de la Universidad de Valladolid por haber puesto a mi disposición los medios necesarios para el desarrollo de este Trabajo Fin de Máster.

Gracias a mis padres, compañeros y amigos, por el apoyo y el ánimo que me han brindado.



# ÍNDICE GENERAL

---

<b>Índice general</b>	<b>vii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Señales biomecánicas y actimetría	1
1.2. Motivaciones	2
1.2.1. Objetivos	2
1.2.2. Fases y Métodos	2
1.2.3. Medios	3
1.3. Estructura del documento	4
<b>2. Actimetría</b>	<b>7</b>
2.1. Descripción	7
2.2. Actímetros	7
2.2.1. Acelerómetros	8
2.2.2. Tasa de muestreo	8
2.2.3. Counts	8
2.2.4. Épocas	9
2.3. Monitor ActiGraph wGT3X-BT	9
2.4. Pulsera inteligente Microsoft Band 2	10
<b>3. Aplicación Registro de Aceleración MSBand</b>	<b>13</b>
3.1. Android	13
3.1.1. Arquitectura de la plataforma de Android	14
3.1.2. Componentes de una aplicación	14
3.1.3. Intenciones	16
3.1.4. Fichero de manifiesto	16
3.1.5. Recursos	16
3.2. SDK de Microsoft Band	17
3.3. Aplicación Registro de Aceleración MSBand	17
3.3.1. Módulos de la app	18
3.3.2. Limitaciones	22
<b>4. Métodos</b>	<b>25</b>
4.1. Introducción	25
4.2. Preprocesado	26
4.2.1. Extracción de datos	26
4.2.2. Remuestreo y alineación	27
4.3. Aproximación de funciones y aprendizaje automático	28
4.3.1. Redes neuronales	28
4.3.2. Bosques aleatorios	31
4.4. Identificación de intervalos de actividad y descanso	32
4.5. Diezmado	32
4.6. Cálculo de características a partir de counts	33

4.6.1. Medidas de centralidad . . . . .	33
4.6.2. Medidas de dispersión . . . . .	34
4.6.3. Medidas de distribución . . . . .	34
4.6.4. Análisis no lineal . . . . .	36
4.7. Intervalos de normalidad . . . . .	38
<b>5. Resultados</b>	<b>39</b>
5.1. Registro y bases de datos . . . . .	39
5.2. Remuestreo y alineación . . . . .	40
5.3. Aproximación de funciones . . . . .	40
5.3.1. Redes neuronales . . . . .	41
5.3.2. Bosques aleatorios . . . . .	43
5.4. Intervalos de normalidad . . . . .	45
<b>6. Conclusiones y líneas futuras</b>	<b>55</b>
6.1. Conclusiones . . . . .	55
6.2. Limitaciones y líneas futuras . . . . .	56
<b>Bibliografía</b>	<b>57</b>
<b>A. Código fuente de la aplicación</b>	<b>61</b>
<b>B. Manual de usuario de la aplicación</b>	<b>75</b>
B.1. Antes de comenzar . . . . .	75
B.2. Instalación de la aplicación . . . . .	75
B.3. Ejemplo de uso . . . . .	76
<b>C. Boxplots</b>	<b>81</b>
C.1. Canal $x$ . . . . .	81
C.2. Canal $y$ . . . . .	87
C.3. Canal $z$ . . . . .	93

# INTRODUCCIÓN

---

## 1.1 SEÑALES BIOMECÁNICAS Y ACTIMETRÍA

---

La función del cuerpo humano está frecuentemente asociada con señales de origen eléctrico, químico o acústico. Dichas señales transportan información que puede no ser percibida inmediatamente y que se encuentra oculta en la propia estructura de la señal. Esta información necesita ser decodificada o extraída de alguna manera antes de que las señales puedan ser interpretadas de forma significativa. Las señales reflejan propiedades de los sistemas biológicos de los que proceden y su decodificación ha sido de ayuda a la hora de explicar e identificar varias condiciones patológicas. De hecho, actualmente es difícil imaginar una situación en la que enfermedades relacionadas con el cerebro, el corazón o los músculos sean diagnosticadas sin incluir información derivada de señales fisiológicas.

Los avances en la miniaturización de dispositivos, así como el cómputo móvil y ubicuo han dado paso a un crecimiento del interés por la tecnología *wearable* [1]. En estos últimos años se ha despertado un interés en la incorporación de señales biomecánicas. Dichas señales se originan a partir de alguna función mecánica del sistema biológico. Estas señales incluyen aquellas producidas por la locomoción y el desplazamiento, señales de flujo y presión, y otras. El fenómeno mecánico no se propaga, como lo hacen los campos magnéticos y eléctricos y las ondas acústicas, por lo que la medición se tiene que realizar normalmente en el lugar exacto donde se origina. Un claro ejemplo del beneficio de la incorporación de señales biomecánicas en los estudios puede verse en la extracción de patrones de actividad obtenidos de acelerómetros, ya sean uniaxiales, biaxiales o triaxiales, además de otros parámetros que aporten información acerca del estilo de vida del paciente. Estas señales pueden proporcionar información que, basada en el contexto, mejore el diagnóstico y seguimiento de varias enfermedades o desórdenes, y pueda desembocar en el diseño de tratamientos personalizados [2].

El análisis las señales biomecánicas se lleva a cabo mediante estudios de actimetría, los cuales se basan en monitorizar la actividad del paciente durante un periodo de tiempo escogido por el facultativo o por el investigador. La monitorización consiste en cuantificar los movimientos realizados por el paciente mediante la aceleración resultante de dichos movimientos [3]. La adquisición de estas señales biomecánicas, con el objetivo de formar parte de aplicaciones clínicas, se realizan con actímetros homologados para tal fin.

Algunas de las patologías o desórdenes que pueden ser estudiados a partir de esta información son el trastorno de déficit de atención e hiperactividad (TDAH o ADHD, por sus siglas en inglés) [4-7], la enfermedad de Parkinson [8-10], el trastorno bipolar [11] o la depresión [12].

---

## 1.2 MOTIVACIONES

---

En el mercado existen diferentes tipos de actímetros, pero sólo algunos de ellos cumplen los requisitos necesarios para su validez en el diagnóstico clínico. El hecho de que cumplan con tales requisitos se manifiesta con el marcado CE (*Conformité Européenne*, Conformidad Europea) en la Unión Europea de acuerdo a la normativa correspondiente: la Directiva 93/42/CEE. En esta directiva se establecen los requisitos esenciales que deben cumplir los productos sanitarios y productos para diagnóstico *in vitro* y sus accesorios, que hacen referencia a todos los aspectos que pueden influir en la seguridad de los productos, incluyendo los de compatibilidad electromagnética y protección de las radiaciones para el caso de productos sanitarios electromédicos. Al igual que el marcado CE en la Unión Europea posee directivas que regulan los productos sanitarios y productos para diagnóstico, en Estados Unidos éstos vienen regulados por un organismo denominado FDA (*Food and Drug Administration*, Administración de Alimentos y Medicamentos).

El interés para la realización de este proyecto radica, principalmente, en averiguar si es posible utilizar dispositivos comerciales, no homologados para uso médico, para obtener información que tenga alguna relevancia clínica, dado que estos dispositivos son de más fácil acceso al público general. Para ello, se deberá encontrar un mapeo entre datos registrados con un actímetro homologado para el diagnóstico clínico, datos que denominaremos *counts*, y datos de aceleración registrados con un actímetro comercial, disponible en el mercado para todo el público. Así como, en dirimir si con actímetros que no poseen los requisitos anteriormente expuestos es posible la obtención de resultados comparables a los obtenidos con actímetros homologados para el diagnóstico clínico, para lo que nos valdremos del cálculo de los intervalos de normalidad en la población adulta, en lo que a actividad respecta.

---

### 1.2.1 OBJETIVOS

---

El objetivo perseguido con la elaboración de este trabajo es, por lo expuesto en la sección anterior, la aproximación de *counts* extraídos del modelo de ActiGraph, poseedor de los requisitos esenciales para poder ser utilizado en el diagnóstico clínico, a partir de datos de aceleración de otros actímetros que no cumplen dichos requisitos. De forma que los intervalos de normalidad obtenidos en base al primer modelo permitan la comparación con los resultados obtenidos a partir de estos dispositivos comerciales.

Este objetivo global puede verse desglosado en los siguientes subobjetivos:

- **Aproximación de *counts*** a partir de datos de aceleración registrados con actímetros sin validez para el diagnóstico clínico.
- **Cálculo de los intervalos de normalidad en adultos** a partir de registros realizados con actímetros válidos para diagnóstico clínico previa segmentación de tramos de actividad o vigilia y descanso.
- **Obtención de resultados comparables** logrados a partir de *counts* válidos y de *counts* estimados.

---

### 1.2.2 FASES Y MÉTODOS

---

Para alcanzar los objetivos planteados, las labores a desarrollar se han ordenado de acuerdo a las siguientes fases:

- A) Estudio de los fundamentos de la actimetría y el uso de actímetros. Propuesta del uso de diferentes actímetros, el monitor wGT3X-BT de ActiGraph y la pulsera Microsoft Band 2.

## B) Tareas respecto de los datos de ActiGraph:

- Lectura de los datos crudos de aceleración (*raw data*, en adelante RD) y de *counts*, los cuales se encuentran en ficheros separados.
- Análisis de dichos datos en cuatro canales, los cuales se corresponden a los ejes X, Y, Z, y al vector magnitud, y en diferentes tramos de actividad y descanso, y con diferentes diezmados.
- Obtención de características en los diferentes tramos mediante medidas tradicionales como media, mediana, desviación estándar, coeficiente de asimetría, curtosis y rango intercuartil, así como mediante medidas no lineales como la medida de la tendencia central y la dinámica simbólica.
- Cálculo de los intervalos de normalidad para cada una de las características obtenidas.

## C) Tareas respecto de Microsoft Band 2:

- Análisis de la metodología necesaria para extraer los datos de dichas pulsera. Propuesta de desarrollo de una aplicación móvil.
- Desarrollo de una aplicación móvil que lea el RD de Microsoft Band 2.
- Búsqueda de la equivalencia entre datos de Microsoft Band 2 y *counts* de ACT, mediante regresores o aproximadores, previo preprocesado consistente en remuestreo y alineación.
- Análisis de estos datos aproximados en los canales antes nombrados y para diferentes diezmados.
- Obtención de características mediante las mismas medidas utilizadas con los datos de ActiGraph.
- Exposición de estos resultados y determinar si son comparables a los obtenidos con los datos de ActiGraph.

### 1.2.3 MEDIOS

---

Será necesario, para la realización de este proyecto, el acceso a las herramientas *software* y *hardware* que se detallan a continuación:

#### Software

- MATLAB [13]: lenguaje de programación técnico de alto nivel y entorno de desarrollo integrado para el desarrollo de algoritmos, visualización y análisis de datos, y computación numérica.
- Eclipse Mars.2: entorno de desarrollo integrado (IDE) que proporciona herramientas para gestionar espacios de trabajo, para construir, lanzar y depurar aplicaciones de diversos lenguajes informáticos.
- Android SDK
- Microsoft Band SDK
- SQLite Database Browser versión 3.9.1
- ActiLife versión 6.13.3 *Lite*
- L<sup>A</sup>T<sub>E</sub>X: sistema de composición de textos, orientado a la creación de documentos escritos.

### Hardware

- PC portátil con las siguientes características:
  - Procesador Intel® Core™ i5-3317U 1.70 GHz.
  - 4 GB de memoria RAM.
  - Disco SSD de 128 GB de capacidad.
- PC sobremesa con las siguientes características:
  - Procesador Intel® Core™ i5-4440 3.10 GHz.
  - 8 GB de memoria RAM.
  - Disco duro de 1 TB de capacidad.
- Microsoft Band 2
- Actigraph wGT3X-BT
- Smartphone Android versión 6.0.1 (BQ Aquaris E5).
- Acceso a servidores de cálculo disponibles en el Laboratorio de Procesamiento de Imagen (LPI) de la Universidad de Valladolid (UVa).

### Bibliografía

- Hemerotecas de la Escuela Técnica Superior de Ingenieros de Telecomunicación.
- Bases de datos electrónicas suscritas por la Universidad de Valladolid.

En cuanto a los registros sobre los que se realizará el análisis y evaluación de los métodos, se hará uso de la base de datos de adultos disponible dentro del grupo en el que se realiza el proyecto, LPI, además de la realización de nuevos registros.

## 1.3 ESTRUCTURA DEL DOCUMENTO

---

El documento se encuentra dividido en 6 capítulos, entre los que se incluye el presente capítulo. El contenido de los capítulos restantes se describe brevemente a continuación:

**Capítulo 2 - Actimetría.** En este capítulo se expone una breve descripción del concepto de actimetría, así como de los actímetros, y los aspectos de éstos que se han de tener en cuenta a la hora de realizar los estudios. Se completa el capítulo con reseñas sobre los actímetros que se usarán en el desarrollo del trabajo: el monitor wGT3X-BT de ActiGraph y la pulsera inteligente Microsoft Band 2.

**Capítulo 3 - Aplicación Registro de Aceleración MSBand.** En este capítulo se presenta la aplicación desarrollada, necesaria para la adquisición de los datos de aceleración proporcionados por la pulsera Microsoft Band 2. Con el fin de aclarar conceptos se incluye una introducción a la programación de aplicaciones Android.

**Capítulo 4 - Métodos.** En este capítulo se expone todo el preprocesado necesario, entendido éste como la extracción de los datos de los diferentes ficheros, remuestreo de las señales de aceleración y alineación de las mismas, de modo que se acondicionen los datos para su procesado. Además, se describen los diferentes métodos llevados a cabo con el objetivo de aproximar la función que proporciona *counts* a partir de datos de aceleración, de manera que sea posible la comparación entre estos *counts* de diferentes actímetros. También se exponen las medidas tradicionales y las medidas no lineales que serán analizadas para el posterior cálculo de los intervalos de normalidad.

**Capítulo 5 - Resultados.** En este capítulo se presentan los resultados obtenidos en diferentes fases del trabajo, como el remuestreo y alineación de las señales, la aproximación de funciones entre datos crudos de aceleración y *counts*, y como en las diferentes medidas trasladadas a intervalos de normalidad.

**Capítulo 6 - Conclusiones y líneas futuras.** Este capítulo final contiene las conclusiones extraídas al término de este Trabajo Fin de Máster, así como limitaciones encontradas. Se plantean las posibles líneas de trabajo futuro surgidas a partir de su realización.

Además de un capítulo en el que se recogen las fuentes bibliográficas y referencias consultadas para la redacción del presente documento, se añaden unos apéndices cuyo contenido se expone en las líneas siguientes:

**Apéndice A - Código fuente de la aplicación.** Contiene algunos de los ficheros de código fuente de la aplicación Android desarrollada.

**Apéndice B - Manual de usuario de la aplicación.** Se exponen los pasos a llevar a cabo para un correcto uso de la aplicación desarrollada, Registro de Aceleración MSBand.

**Apéndice C - *Boxplots*.** Se muestran los *boxplots* extraídos para las diferentes medidas calculadas en los que se superponen los resultados obtenidos con los datos de Microsoft Band 2 para los canales x, y, z, en el tramo de Actividad, con diferentes diezmados.



# ACTIMETRÍA

---

## 2.1 DESCRIPCIÓN

---

La actimetría (o actigrafía) es un método no invasivo que permite, mediante la colocación de un aparato denominado actímetro, normalmente colocado en la muñeca del brazo no dominante, registrar y medir el movimiento de manera objetiva, entendido éste como cambios en la aceleración. La ventaja del uso de la actimetría en estudios radica en su facilidad de uso. Además, el uso del actímetro no distorsiona la dinámica diaria y el entorno social del sujeto [2].

La actimetría es una herramienta valiosa para el diagnóstico objetivo y el seguimiento de diferentes patologías, dado que permite la evaluación de intervalos de actividad y sueño que pueden verse afectados por dichas patologías. Además, esta técnica de monitorización puede ser aplicada durante largos periodos de tiempo de forma mínimamente invasiva, ubicua y no muy cara [2].

Las aplicaciones más inmediatas de estas monitorizaciones son el estudio de biorritmos y de su alteración por diversas causas. Éstas pueden ser el estado emocional y somático, el empleo de fármacos y drogas o el entorno de actividad, incluyendo las condiciones y horarios de trabajo [3]. Por todo ello, la actimetría ha servido para el estudio del TDAH [4–7], la enfermedad de Parkinson [8–10], el trastorno bipolar [11] o la depresión [12].

## 2.2 ACTÍMETROS

---

Un actímetro es un aparato de tamaño similar a un reloj de pulsera, formado por un acelerómetro y una memoria (aunque puede contener otros sensores). En dicha memoria se acumulan las medidas de actividad, que pueden realizarse durante periodos prolongados de tiempo.

Este dispositivo se coloca habitualmente en la muñeca del brazo no dominante del sujeto que lo porta (aunque también puede colocarse en la cintura o en el tobillo). Se han realizado estudios como [14] que concluyen que no existen diferencias significativas en los registros de actimetría en función de la muñeca en que se coloque el actímetro. Sin embargo, y por comodidad para la persona que lo porte, en este trabajo los registros se realizarán con el actímetro en la muñeca no dominante del sujeto [15].

En las siguientes subsecciones se tratarán aspectos tales como las características propias del acelerómetro (2.2.1), la tasa de muestreo a la que se recogen los datos (2.2.2), *counts* (2.2.3) y las épocas a estudiar (2.2.4).

---

### 2.2.1 ACELERÓMETROS

---

Los acelerómetros usados emplean diferentes sistemas microelectromecánicos (*Microelectromechanical Systems*, MEMS) para detectar cambios en la aceleración. Frecuentemente se usan tecnologías que incluyen sensores piezoeléctricos, capacitivos y resistivos [16]. La mayoría de monitores de actividad basados en acelerómetros proporcionan a su salida representaciones de aceleración arbitrarias conocidas como *counts*. Esta unidad carece de significado fisiológico, pero ha sido ampliamente usada para predecir, entre otras cosas, el gasto energético (kilocalorías, METs) mediante modelado estadístico en diferentes grupos de edad [17–19].

El movimiento humano puede ser clasificado en base a planos anatómicos que corresponden a la detección triaxial de la aceleración usando acelerómetros. Cuando el movimiento tiene lugar en el plano transversal o eje vertical, el movimiento se entiende hacia arriba o hacia abajo en línea con la fuerza de la gravedad. Cuando el movimiento ocurre en el plano sagital o eje lateral, el movimiento tiene lugar de derecha a izquierda, o viceversa. Y cuando el movimiento ocurre en el plano frontal o eje anteroposterior, el movimiento es hacia delante o hacia atrás [15].

Los primeros acelerómetros usados para la medida de la actividad física eran uniaxiales y únicamente recogían datos en el eje vertical, dado que el movimiento humano en condiciones diarias, en la mayor parte, puede explicarse como desplazamiento vertical del centro de masa del cuerpo [20]. Actualmente, la mayoría de los acelerómetros comerciales que se incluyen en estos dispositivos son triaxiales, lo que significa que son sensibles al movimiento en tres planos (las tres direcciones ortogonales del espacio), permitiendo medir movimientos que en realizaciones de actímetros anteriores se perdían. Los datos recogidos con estos modelos pueden ser analizados en cada eje por separado o usando el vector resultante de los tres ejes, llamado vector magnitud [21].

---

### 2.2.2 TASA DE MUESTREO

---

La tasa de muestreo es la frecuencia predeterminada a la cual se configuran los acelerómetros para recoger los datos. La unidad utilizada para la tasa de muestreo es el Hertzio, (i.e., número de veces por segundo que el acelerómetro medirá el movimiento). En los modelos actuales de monitores de actividad, por ejemplo, de ActiGraph, los cuales son los más frecuentemente usados por investigadores, la tasa de muestreo puede ser ajustada entre 30 Hz y 100 Hz, mientras que otros dispositivos, como la pulsera inteligente Microsoft Band 2 las frecuencias disponibles para la recuperación de los datos del acelerómetro son 62/31/8 Hz. El RD de aceleración registrado a esas altas frecuencias se presenta normalmente en múltiplos de la fuerza de gravedad  $g$ . La mayoría de los métodos validados emplean *counts* para estimar la intensidad de la actividad física, mientras que estudios recientes han desarrollado métodos para procesar este RD [15].

---

### 2.2.3 COUNTS

---

Los *counts* son una medida de actividad adimensional y están directamente relacionados con la magnitud de la aceleración. Se cree que los orígenes del término *count* se pueden rastrear hasta los primeros monitores de actividad que tenían pocos o ningún dispositivo de estado sólido capaz de cuantificar múltiples niveles de actividad. Estos dispositivos utilizaban una técnica de cruce de umbral para supervisar la actividad. La actividad que provocaba que la señal de aceleración superara el umbral era “contado” como actividad; cualquier aceleración por debajo de este umbral era ignorada. Al final del periodo de medición, se registraba el número de “cuentas” (*counts*) de actividad. Se cree que el uso de este término, *counts*, se ha trasladado a los monitores de actividad modernos [22].

La derivación de los *counts* implica dos pasos importantes:

1. La rectificación de la señal.
2. El filtrado patentado, que puede ser propiedad de la compañía que comercia con los actímetros.

La rectificación de la señal es el proceso de transformar señales de aceleración negativa en señales positivas equivalentes. El filtrado patentado puede implicar varios filtros para diversos fines. El uso de filtros selectivos en frecuencia para eliminar frecuencias y valores de aceleración fuera del rango normal del movimiento humano es común entre los posibles filtros que se pueden emplear en el preprocesado de datos. Los datos filtrados se clasifican en diferentes niveles de aceleración [23], donde cada nivel es equivalente a una cierta proporción de *g/count*. Además, cada fabricante tiene sus propias especificaciones de filtrado para obtener *counts*. Por lo tanto, la comparación directa de los *counts* de monitores de actividad de diferentes marcas no es posible [15].

#### 2.2.4 ÉPOCAS

---

Las épocas son intervalos de tiempo especificados para los cuales los *counts* son sumados para el análisis de datos [24]. Estos intervalos son seleccionados por el investigador de entre los disponibles en el dispositivo y pueden abarcar un rango desde 1 s hasta periodos mayores que 1 min. Las épocas de 60 s han sido tradicionalmente usadas para clasificar la intensidad de una actividad como sedentaria, leve, moderada o vigorosa. Los primeros modelos de monitores de actividad no eran capaces de coleccionar datos en épocas menores de 1 min. Sin embargo, los avances tecnológicos han hecho posible el registrado de *counts* en épocas de 1 s, y de RD a frecuencias de 100 Hz [15].

La selección de las épocas también puede depender de la población con la que se utiliza el monitor. Por ejemplo, en niños, se recomienda que los investigadores adopten épocas cortas (e.g., 15 s) para clasificar la actividad física debido a la naturaleza intermitente de las actividades en este grupo de edad [25]. En adultos y personas mayores, la mayoría de los métodos clasifican la intensidad de la actividad usando épocas de 60 s [21, 26–30]; sin embargo, si se utilizan métodos más recientes, los investigadores necesitan usar épocas más cortas (e.g., 1 s, 5 s).

### 2.3 MONITOR ACTIGRAPH WGT3X-BT

---

El wGT3X-BT es el monitor de actividad emblemático de ActiGraph (Pensacola, FL), en adelante ACT, utilizado por investigadores de todo el mundo para capturar y grabar información continua y de alta resolución de actividad física y de sueño/vigilia. Proporciona medidas como aceleración, gasto energético, pasos dados, intensidad de la actividad física, ritmo cardiaco (para lo que requiere de un pulsómetro compatible sin cables), posición corporal y luz ambiental. Estos datos son accesibles a través del software ActiLife, propiedad de ACT [31]. En la figura 2.1(a) se muestra el diseño del actímetro, y en la figura 2.1(b) se puede observar cómo se encuentra orientado el dispositivo y la forma en que han de interpretarse los ejes, en cuanto a datos de aceleración se refiere.

El wGT3X-BT tiene la aprobación de la FDA de dispositivos médicos de Clase II en los Estados Unidos y el certificado médico Clase I dentro de la Unión Europea [32], lo que significa que, en lo que respecta a invasividad y riesgo, supone un bajo riesgo. Por poseer estos marcados, FDA y CE, en este trabajo se tomará este actímetro como referencia.

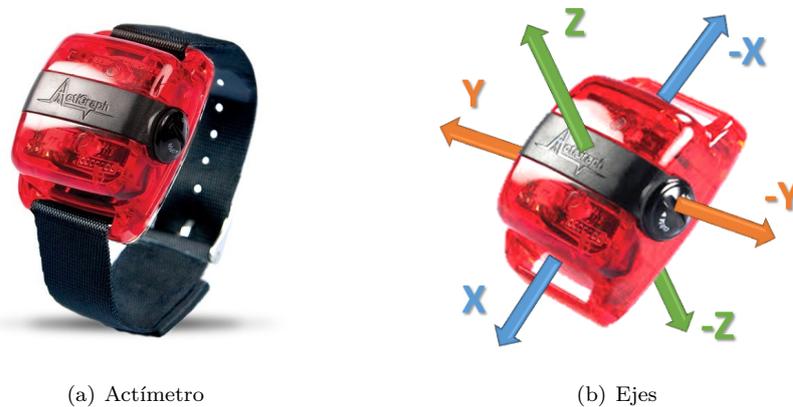


FIGURA 2.1: Monitor ActiGraph wGT3X-BT.

El software ActiLife 6 es la principal plataforma de análisis y gestión de datos de actigrafía de ACT. Se utiliza para preparar los dispositivos de ACT para la recopilación de datos, y para descargar, procesar, y administrar con seguridad los datos recogidos tanto RD como *counts*; para ello, basta con conectar el monitor de ACT al ordenador mediante un cable. ActiLife tiene una extensa selección de características integradas y herramientas de análisis diseñadas para distintos objetivos clínicos y de investigación.

Existen dos versiones del software ActiLife dependiendo de la licencia, la versión *Full* y la *Lite*. La versión *Full* pone a disposición del usuario todas las posibilidades, como inicializar dispositivos, importar la información del paciente, importar datos del dispositivo y analizar los datos. Con la versión *Lite* sólo es posible exportar datos e inicializar el dispositivo de nuevo, por lo que no es posible el análisis. Dado que para la realización de este trabajo se dispone de la versión *Lite*, se requieren otras herramientas externas que permitan ver y procesar esos datos [33].

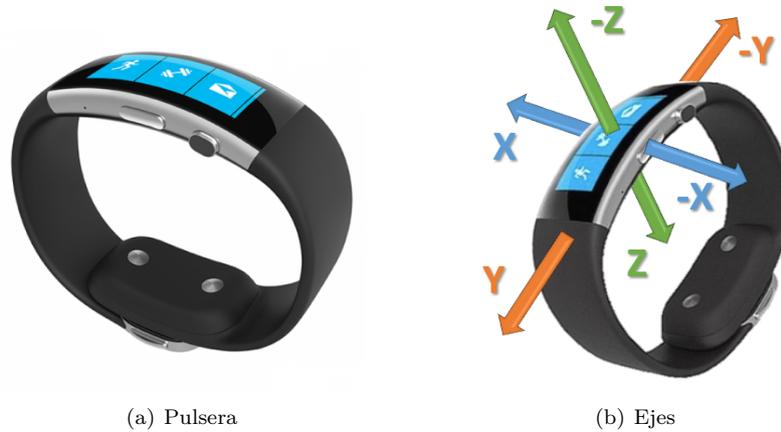
Las soluciones de monitorización de ACT han sido extensamente utilizadas por muchos estudios de investigación para determinar objetivamente los perfiles de actividad física y el comportamiento del sueño en cientos de miles de sujetos investigados por todo el mundo.

## 2.4 PULSERA INTELIGENTE MICROSOFT BAND 2

Microsoft Band 2, en adelante MSB, es una pulsera inteligente que permite a sus usuarios realizar un seguimiento de su actividad física así como recibir notificaciones del smartphone al que se encuentra emparejada. Este dispositivo no cuenta con los marcados FDA y CE; además, se advierte al usuario de que no es para uso en maquinaria, aplicaciones médicas e industriales [34], por lo que no cumple los requisitos necesarios para su validez en el diagnóstico clínico. En la figura 2.2(a) se muestra el diseño de la banda y en la figura 2.2(b) puede observarse la orientación del dispositivo.

Existen tres tamaños disponibles. La diferencia entre los tres modelos radica en la longitud de la propia pulsera, de forma que se adapte de forma ergonómica a su portador (figura 2.3).

Supera a su predecesora, la Microsoft Band, en cuanto a diseño, pantalla, duración de la batería y la adición de nuevos sensores.

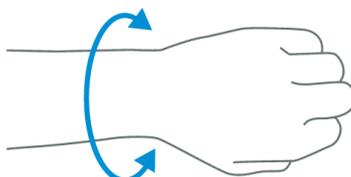


(a) Pulsera

(b) Ejes

FIGURA 2.2: Pulsera inteligente Microsoft Band 2.

La pulsera inteligente MSB cuenta con los siguientes sensores: sensor óptico de ritmo cardiaco, acelerómetro triaxial, giróscopo triaxial, GPS, sensor de luz ambiental, sensor UV, sensor de temperatura corporal, sensor de proximidad, sensor de respuesta galvánica, micrófono y barómetro [35]. El RD capturado por los sensores de esta pulsera inteligente son transmitidos continuamente vía Bluetooth, y recibidos por el dispositivo móvil que esté emparejado a ésta y suscrito a los flujos de datos de dichos sensores.



	Wrist circumference	
	Millimeters	Inches
Small	145 – 165	5.7 – 6.5
Medium	160 – 190	6.3 – 7.5
Large	185 – 215	7.3 – 8.5

FIGURA 2.3: Tallas de la pulsera Microsoft Band 2.

Como se ha dicho, la pulsera MSB se conecta con el smartphone vía Bluetooth, pero sólo con llevar a cabo este emparejamiento no es suficiente para acceder al RD. Por ello, para extraer el RD de esta pulsera es necesaria una aplicación móvil, cuyo desarrollo se ha llevado a cabo en el marco de este trabajo y se presenta en el capítulo 3.



## Capítulo 3

# APLICACIÓN REGISTRO DE ACELERACIÓN MSBAND

---

En este capítulo se presenta la aplicación **Registro de Aceleración MSBand**, necesaria para la adquisición de los datos de aceleración proporcionados por la MSB. Para desarrollar la aplicación se requería del uso del SDK (*Software Development Kit* o kit de desarrollo de software) de Microsoft Band, que posibilita programar aplicaciones que hagan uso de ciertas funcionalidades propias en Android, iOS y Windows Phone.

El motivo de haber elegido Android para la programación de la aplicación es, además de su manejo sencillo e intuitivo, el hecho de que se ha convertido en el gran referente del sector por pertenecer a Google, que decidió adquirir esta empresa en el año 2007 [36]. En ese momento no era tan popular como en la actualidad, dado que no se hizo conocido a nivel mundial hasta que no se popularizaron los smartphones. Actualmente la gran mayoría de smartphones cuentan con Android, teniendo una amplia ventaja en cuota de mercado respecto a otros sistemas como iOS o Windows Phone, como se puede observar en la figura 3.1.

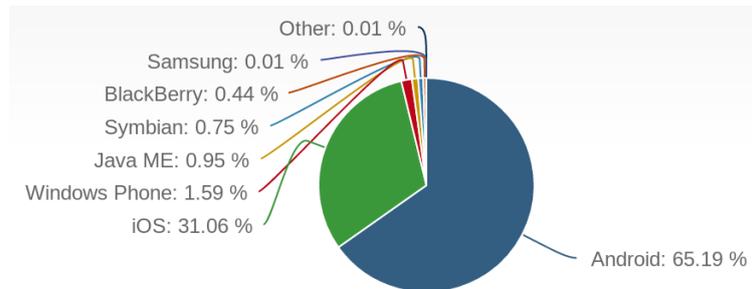


FIGURA 3.1: Mercado mundial de sistemas operativos en móviles y tabletas. Abril 2017 [37].

Para una mejor comprensión de la terminología utilizada en la sección 3.3, se describen a continuación algunos conceptos fundamentales de Android.

## 3.1 ANDROID

---

A lo largo de esta sección se encuentran descritos: la arquitectura de la plataforma de Android (3.1.1), los principales componentes que definen una aplicación (3.1.2), las intenciones que activan algunos de los componentes (3.1.3), el fichero de manifiesto en el que se declaran dichos componentes y

las características del dispositivo necesarias (3.1.4), así como los recursos que están separados del código y que permiten a una aplicación optimizar su comportamiento con gran variedad de configuraciones de dispositivos (3.1.5).

### 3.1.1 ARQUITECTURA DE LA PLATAFORMA DE ANDROID

---

A modo de breve descripción podemos decir que Android es un entorno operativo de código abierto dirigido a dispositivos móviles.

Las aplicaciones Android se escriben en lenguaje de programación Java. Las herramientas del Android SDK compilan el código existente en un APK, un *Android package* (paquete Android), que es un archivo con el sufijo .apk. Un archivo APK contiene todo lo que se incluye en una aplicación Android y es el archivo que utilizan los dispositivos para instalar la aplicación [38].

Desde un punto de vista más técnico, el sistema operativo Android es un sistema Linux multiusuario en el que cada aplicación se entiende como un usuario diferente. De forma predeterminada, cada aplicación se ejecuta en su propio proceso de Linux. El sistema Android inicia el proceso cuando es necesario ejecutar cualquiera de los componentes de la aplicación y, a continuación, cierra el proceso cuando ya no se necesita o cuando el sistema debe recuperar la memoria para otras aplicaciones.

El sistema Android implementa el principio del privilegio mínimo; es decir, cada aplicación, por defecto, tiene acceso sólo a los componentes que requiere para hacer su trabajo, por lo que no puede acceder a partes del sistema para las que no se le da permiso. Sin embargo, una aplicación puede solicitar permiso para acceder a los datos del dispositivo, como los contactos del usuario, los mensajes SMS, el almacenamiento, la cámara y el Bluetooth. El usuario tiene que conceder explícitamente estos permisos. Dependiendo de la versión de Android, el usuario concede los permisos al instalar la aplicación o bien al ejecutarla.

### 3.1.2 COMPONENTES DE UNA APLICACIÓN

---

Los componentes son los principales elementos de una aplicación Android. Cada componente es un punto de entrada a través del cual el sistema o un usuario puede acceder a la aplicación. Algunos componentes dependen de otros [38].

Hay cuatro tipos diferentes de componentes de aplicación:

- Actividades (*Activities*).
- Servicios (*Services*).
- Receptores de difusión (*Broadcast receivers*).
- Proveedores de contenido (*Content providers*).

Cada tipo tiene un propósito distinto y tiene un ciclo de vida distinto. Las siguientes secciones describen los cuatro tipos de componentes de la aplicación.

## Actividades

Una actividad representa una sola pantalla con una interfaz de usuario.

La clase `Activity` sirve como punto de entrada para la interacción de una aplicación con el usuario, proporcionando la ventana en la que la aplicación dibuja su interfaz de usuario (UI, *user interface*). Esta ventana normalmente llena la pantalla, pero puede ser más pequeña que la pantalla y flotar en la parte superior de otras ventanas. Las actividades se implementan como subclases de la clase `Activity`.

La mayoría de las aplicaciones contienen varias pantallas, lo que significa que abarcan varias actividades. Normalmente, se especifica una actividad en una aplicación como actividad principal, que es la primera que aparece cuando el usuario inicia la aplicación. Cada actividad puede iniciar otra actividad para realizar diferentes acciones.

Para utilizar actividades en una aplicación, se ha de registrar información sobre ellas en el fichero de manifiesto de la aplicación.

## Servicios

Un servicio es un componente de aplicación que puede realizar operaciones de larga duración en segundo plano y no proporciona una interfaz de usuario.

Un servicio puede ser iniciado por otro componente de la aplicación y continúa ejecutándose en segundo plano incluso si el usuario cambia a otra aplicación. Además, un componente puede enlazar a un servicio para interactuar con él e incluso realizar comunicación entre procesos (IPC, *interprocess communication*).

Para crear un servicio se debe crear una subclase de `Service` o usar una de las subclases existentes. El sistema Android fuerza la detención un servicio sólo cuando la memoria es baja y debe recuperar los recursos del sistema para la actividad que tiene el foco del usuario.

Se deben declarar todos los servicios en el manifiesto de la aplicación, al igual que ocurre con las actividades.

## Receptores de difusión

Un receptor de difusión es un componente que permite al sistema entregar eventos a la aplicación fuera de un flujo regular, permitiendo que la aplicación responda a anuncios de difusión de todo el sistema. Las aplicaciones también pueden iniciar transmisiones.

Un receptor de difusión se implementa como una subclase de `BroadcastReceiver`. Debido a que los receptores de difusión son otra entrada bien definida en la aplicación, el sistema puede entregar transmisiones incluso a aplicaciones que no se estén ejecutando actualmente. Aunque los receptores de difusión no muestran una interfaz de usuario, pueden crear una notificación de barra de estado para alertar al usuario cuando se produce un evento de difusión.

Los receptores de difusión también deben declararse en el fichero de manifiesto de la aplicación.

## Proveedores de contenido

Un proveedor de contenido administra un conjunto compartido de datos, los cuales identifica mediante un esquema URI, que puede almacenar en el sistema de archivos, en una base de datos SQLite,

en la Web o en cualquier otra ubicación de almacenamiento permanente a la que pueda acceder la aplicación. A través del proveedor de contenido, otras aplicaciones pueden consultar o modificar los datos si el proveedor de contenido lo permite.

Un proveedor de contenido se implementa como una subclase de `ContentProvider` y debe implementar un conjunto estándar de API (*Application Programming Interface*) que permita a otras aplicaciones realizar transacciones. Del mismo modo que el resto de componentes, los proveedores de contenido deben aparecer en el fichero de manifiesto.

### 3.1.3 INTENCIONES

---

Una intención es un mensaje asíncrono que representa la voluntad de realizar alguna acción o tarea. Las actividades, servicios y receptores de difusión se activan mediante estos mensajes, los cuales pueden contener información extra que comunicar al componente destinatario del mensaje.

Se crea una intención con un objeto `Intent`. Para las actividades y los servicios, una intención define la acción a realizar. Para los receptores de difusión, la intención simplemente define el anuncio que se emite.

A diferencia de las actividades, servicios y receptores de difusión, los proveedores de contenido no se activan por intenciones.

### 3.1.4 FICHERO DE MANIFIESTO

---

Antes de que el sistema Android pueda iniciar un componente de aplicación, el sistema, previa consulta al fichero de manifiesto de la aplicación `AndroidManifest.xml`, debe saber que el componente existe. Cada aplicación debe declarar todos sus componentes en este fichero, el cual debe estar localizado en la raíz del directorio del proyecto de la aplicación.

El manifiesto contiene información adicional además de las declaraciones de los componentes de la aplicación, como puede ser:

- Identificación de los permisos de usuario que requiera la aplicación, como acceso a Internet o acceso de lectura a los contactos del usuario.
- Declaración del nivel mínimo de API marco de Android requerido por la aplicación, lo cual se traduce en la versión mínima de Android que soportará la aplicación.
- Declaración de las características de hardware y software utilizadas o requeridas por la aplicación, como una cámara, servicios Bluetooth o una pantalla multitáctil.
- Declaración de las bibliotecas API con las que la aplicación debe enlazarse (distintas de las API marco de Android), como la biblioteca de Google Maps.

### 3.1.5 RECURSOS

---

Cualquier aplicación de Android requiere de una serie de recursos adicionales al código fuente, como pueden ser imágenes, archivos de audio y cualquier elemento relacionado con la presentación visual de la aplicación. Se pueden definir menús, estilos, colores y el diseño de las interfaces de usuario de las actividades a partir de archivos XML. El uso de estos recursos facilita al programador la actualización de varias características de la aplicación sin necesidad de modificar el código. Proporcionar conjuntos de recursos

alternativos permite optimizar la aplicación para una variedad de configuraciones de dispositivos, como idiomas y tamaños de pantalla diferentes.

## 3.2 SDK DE MICROSOFT BAND

---

El SDK de Microsoft Band ofrece acceso a los sensores disponibles en la banda o pulsera, así como la capacidad de crear y enviar notificaciones [39].

Las aplicaciones que funcionan con MSB utilizan el SDK de Microsoft Band para comunicarse con la banda [40]. La lógica de la aplicación se ejecuta en el sistema operativo anfitrión, en este caso Android, y controla de forma remota la interfaz de usuario de la banda. También recibe información contextual y datos de los sensores de la banda.

La MSB cuenta con muchos sensores de los que los desarrolladores de aplicaciones pueden obtener datos. El SDK expone los datos de estos sensores como flujos, y las aplicaciones pueden suscribirse a estos flujos. Una suscripción es esencialmente un mecanismo de devolución de llamada específico de la plataforma. Entregará datos a intervalos específicos del sensor. Algunos sensores tienen intervalos dinámicos, como el acelerómetro (en Android y Windows), que permiten a los desarrolladores especificar a qué velocidad quieren que se entreguen los datos. En el caso de múltiples frecuencias soportadas, sólo la tasa más alta está disponible para iOS.

Para el problema que nos ocupa, el sensor de interés es el acelerómetro. Proporciona datos de aceleración en unidades g ( $1\text{ g} = 9.81\text{ metros por segundo al cuadrado } (m/s^2)$ ) en los ejes X, Y, y Z, por tratarse de un acelerómetro triaxial. Las frecuencias disponibles para la recuperación de los datos del acelerómetro son 62/31/8 Hz [40].

## 3.3 APLICACIÓN REGISTRO DE ACELERACIÓN MSBAND

---

Para el desarrollo de esta aplicación en concreto, se han requerido las siguientes tecnologías o herramientas:

- Plataforma en la que programar: Eclipse (*Java Integrated Development Environment*, IDE).
- Herramientas desarrolladoras de Android (*Android Development Tools* o ADT y Android SDK).
- Herramientas desarrolladoras de Microsoft Band (Microsoft Band SDK).
- Código fuente.

Además, para poder hacer uso de las características ofrecidas por el SDK de Microsoft Band, es necesaria una cuenta de Microsoft y vincular a ella la aplicación Microsoft Band, que debe estar instalada en el smartphone.

La aplicación desarrollada consiste en dos actividades, dos receptores de difusión y un servicio; todo ello indicado en el fichero de manifiesto. Asimismo, en este mismo fichero se declara la necesidad de los siguientes permisos: Bluetooth, emparejamiento con la pulsera (propio del SDK de Microsoft band) y almacenamiento, como se muestra en el Apéndice A en las líneas 9 a 11 del fichero `AndroidManifest.xml`.

El funcionamiento queda descrito a modo de diagrama en la figura 3.2, en la que las flechas grises han de ser entendidas como intenciones. Desde la primera actividad o actividad principal se lanza la segunda actividad. En la segunda actividad se configura una alarma para programar el comienzo del registro y, además, se lanza una notificación que proporciona la posibilidad de cancelar el registro. Si el

registro es cancelado, se anula la alarma, mientras que si no se cancela se lanzará el servicio en segundo plano, que es el que realiza el registro en sí mismo. En la sección 3.3.1 se describe en detalle cada uno de los componentes y acciones llevados a cabo.

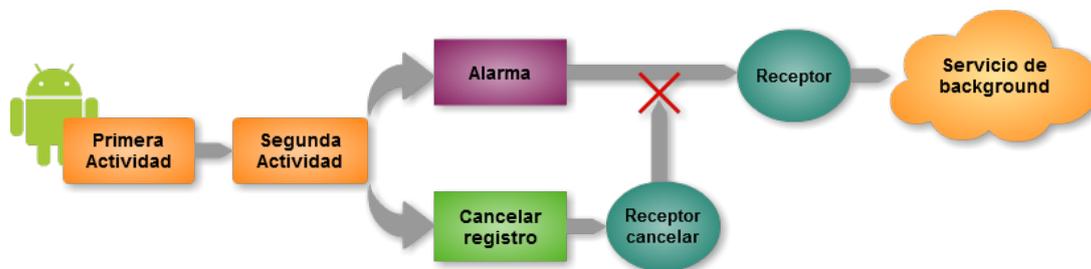


FIGURA 3.2: Esquema o diagrama descriptivo del flujo de la aplicación.

### 3.3.1 MÓDULOS DE LA APP

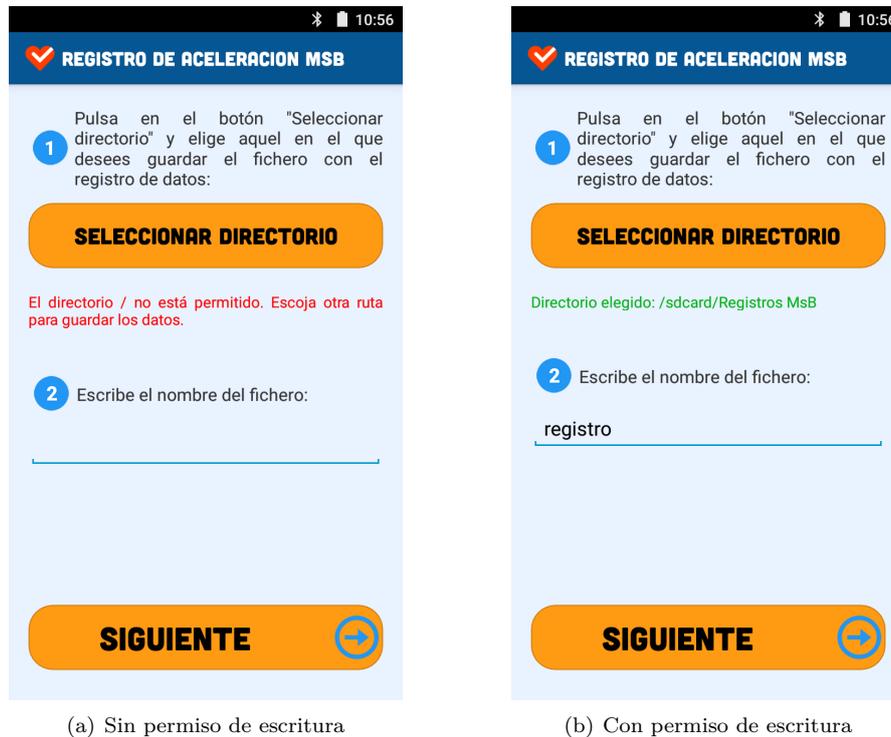
Como se expuso anteriormente, la aplicación consiste en dos actividades, dos receptores de difusión y un servicio. A continuación se detallan las acciones que se llevan a cabo en cada uno de estos componentes y cómo interaccionan entre ellos.

#### Primera actividad o actividad principal

Esta actividad, representada como una ventana que llena la pantalla del dispositivo, es el primer punto de interacción del usuario con la aplicación; es decir, será lo primero con lo que se encuentre el usuario al iniciar la aplicación. El diseño de esta actividad se puede observar en la figura 3.3.

En ella, el usuario debe seleccionar el directorio en el que desee guardar el fichero con los datos de aceleración. Para ello, al pulsar el botón *Seleccionar directorio* se lanza una ventana que permite navegar por el árbol de directorios, además de proporcionar la posibilidad de crear un nuevo directorio; su implementación puede observarse en las líneas 35 a 61 del fichero `Primera_Actividad.java` contenido en el Apéndice A. Esta funcionalidad fue extraída de [41], bajo licencia CPOL (*Code Project Open License*). Una vez seleccionado el directorio deseado se realiza una comprobación de los permisos de escritura en ese directorio, avisando al usuario de la validez de ese directorio. Si el directorio no tiene permisos de escritura se mostrará un mensaje como el que aparece en la figura 3.3(a), mientras que si el directorio elegido sí que permite escritura aparecerá un mensaje como se muestra en la figura 3.3(b). Tras la elección del directorio se ha de escribir el nombre que recibirá el fichero que contendrá los datos de aceleración.

Para confirmar el nombre del fichero y su ubicación se ha de pulsar en *Siguiente*. Se comprueba que se ha elegido un directorio válido y se ha proporcionado un nombre al fichero. Si no se ha seleccionado un directorio válido o no se ha escrito el nombre del fichero, se lanza un diálogo para alertar al usuario (figura 3.4), de modo que pueda completar estos campos para continuar. En caso de que ninguno de los campos esté vacío se lanza la segunda actividad, que recibe tanto la ruta del directorio como el nombre del fichero. La implementación de esta funcionalidad queda expuesta en el Apéndice A, fichero `Primera_Actividad.java`, líneas de código 63 a 79.



(a) Sin permiso de escritura

(b) Con permiso de escritura

FIGURA 3.3: Actividad principal. Permiso de escritura en el directorio seleccionado.

## Segunda actividad

Como se acaba de indicar, la segunda actividad recibe de la actividad principal la ruta del directorio y el nombre del fichero como cadenas de caracteres. La función de esta actividad es dar al usuario la posibilidad de programar un registro, esto es, permite elegir fecha y hora de comienzo del registro y fecha y hora de fin del registro (figura 3.5). La implementación de la selección de fecha y hora se ha llevado a cabo mediante botones, los cuales lanzan diálogos en los que el usuario puede escoger estos valores de forma sencilla. En la figura 3.5(b) se muestra una posible configuración.

Existen restricciones en cuanto a tiempo máximo de registro, determinado en 24 horas, así como de tiempo mínimo de registro, determinado en 1 minuto. En el caso de no respetar estas condiciones se alerta al usuario con un diálogo como el indicado en la figura 3.6(b). También se contemplan algunos errores o incoherencias, como pueden ser que no se seleccione alguno de los parámetros (figura 3.6(a)) o que trate de programarse un registro en el pasado o en el instante actual (figura 3.6(c)).

Una vez configurado el registro se ha de pulsar en *Comenzar*. En ese momento, se programa una alarma para el instante de comienzo del registro. Además, se muestra una notificación con la fecha y hora de comienzo del registro, permitiendo al usuario cancelarlo, como se muestra en la figura 3.7. Las diferentes intenciones que permiten lanzar los receptores al pulsar el botón *Comenzar*, junto con los parámetros necesarios en cada una de ellas, quedan reflejadas en las líneas 186 a 246 del fichero `Segunda_Actividad.java` contenido en el Apéndice A.



FIGURA 3.4: Actividad principal. Alerta de ruta incompleta.



(a) Diseño



(b) Configuración del tiempo de registro

FIGURA 3.5: Segunda actividad.

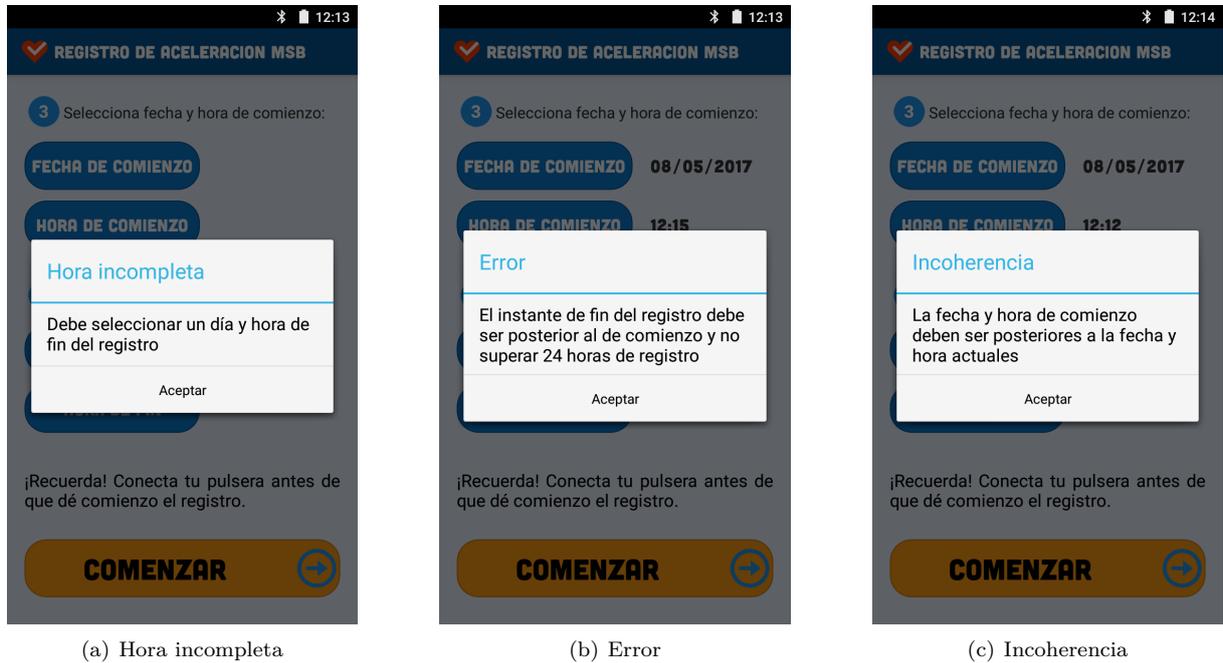


FIGURA 3.6: Posibles alertas al configurar el registro.

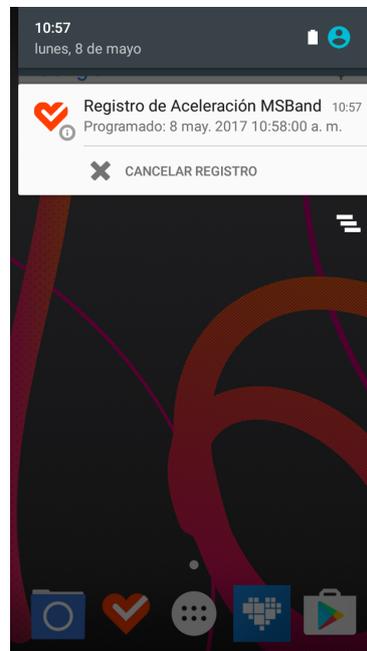


FIGURA 3.7: Notificación de registro programado.

### Receptores de difusión

Se han implementado dos receptores de difusión. Uno de ellos se hará cargo de recibir la alarma y lanzar el servicio que se ejecuta en segundo plano, mientras que el otro recibirá la acción de cancelar el registro desde la notificación.

Por lo tanto, si el usuario decide cancelar el registro (pulsando en el correspondiente botón de la notificación 3.7), esta acción será recibida por un receptor de difusión que anulará la alarma y, por consiguiente, no lanzará el servicio que lleva a cabo el registro.

Por otro lado, si el usuario no cancela el registro y permite que se dispare la alarma, ésta será recibida por otro receptor de difusión, que lanzará el servicio en segundo plano, el cual se encargará del registro.

El código fuente de cada uno de los receptores se muestra en el Apéndice A, en los ficheros `ReceptorCancelar.java` y `Receptor.java`, respectivamente.

### Servicio en segundo plano

Lo primero que realiza este servicio es la comprobación de que la pulsera está conectada. En caso de no estarlo, se detiene el servicio y se muestra una notificación, como se puede observar en la figura 3.8(a). Si la pulsera está conectada, el servicio se suscribe al flujo de datos del acelerómetro (Apéndice A, `ServicioBackground.java`, línea 101). Recoge los datos de aceleración que recibe por Bluetooth y los va guardando en un fichero de texto con el siguiente formato: `ejeX ejeY ejeZ ejeX ejeY...`, como cadena de caracteres. Mientras se lleva a cabo el registro aparece una notificación permanente, la cual muestra dinámicamente en pantalla los datos que se van recibiendo (figura 3.8(b)).

El código que se ejecuta cada vez que un evento del sensor de aceleración de la pulsera es recibido, se expone en las líneas 18 a 66 del `ServicioBackground.java` contenido en el Apéndice A.

Cuando finaliza el registro se muestra otra notificación al usuario para darle a conocer el fin del registro, como en la figura 3.8(c). A partir de ese momento, el usuario podrá acceder al fichero de texto con los datos de aceleración.

### 3.3.2 LIMITACIONES

Han de entenderse las limitaciones como aquellas situaciones con una elevada probabilidad de ocurrencia que dan lugar a errores que interrumpen un registro y que, por simplificar la aplicación, no han sido contemplados en esta primera versión.

La primera situación a tener en cuenta es que alguno de los dispositivos implicados en el registro, ya sea la MsB o el smartphone, se queden sin batería. Para prevenirlo, antes de comenzar un registro es necesario asegurarse de que ambos dispositivos disponen de batería suficiente para completarlo.

Por último, es necesario saber que, una vez que da comienzo el registro, no es posible cancelarlo directamente desde la aplicación. Si durante un registro el usuario quiere cancelarlo deberá acceder a él a través de los *Ajustes* del propio smartphone: *Ajustes* → *Aplicaciones* → *Registro de Aceleración MSBand* → *Forzar detención*.

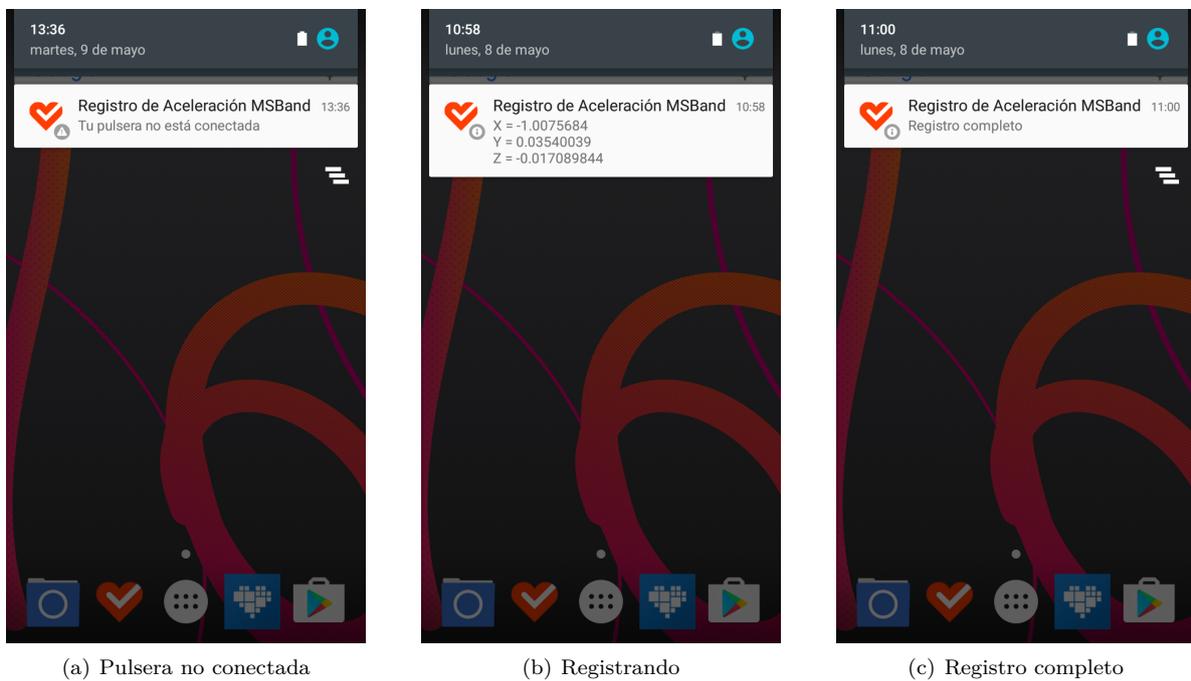


FIGURA 3.8: Notificaciones desde el servicio en segundo plano.



# MÉTODOS

---

## 4.1 INTRODUCCIÓN

---

El objetivo de este trabajo conseguir aproximar el RD de un actímetro comercial, en este caso MSB, a *counts* de ACT. A fin de validar tal aproximación se hallarán los intervalos de normalidad en adultos a partir de *counts* de actividad, cuya metodología consiste en una serie de etapas consecutivas entre las que se encuentran la identificación de intervalos de actividad y descanso, para lo cual se hará uso de algoritmos de clasificación basados en descriptores calculados a partir de *counts* de ACT [5, 6, 42], y el diezmado previo al procesado de señal propiamente dicho. Tal procesado consiste en la extracción de características mediante la aplicación de medidas tradicionales como media, desviación estándar, mediana, rango intercuartil, coeficiente de asimetría, curtosis y coeficiente de variación, además de otros métodos no lineales de cuantificación de variabilidad, como la medida de la tendencia central y la dinámica simbólica.

ACT, a través de su software ActiLife, proporciona los *counts* para unos datos de aceleración registrados. Estas unidades son obtenidas a partir de un procesado, consistente en un filtro digital, del que es propietario ACT y que, por lo tanto, es desconocido para el público, aunque sí es cierto que se conocen algunas de sus características. Se sabe que este filtro digital limita los datos registrados a un rango de frecuencias de 0.25 a 2.5 Hz, elegido para detectar movimiento humano normal y eliminar cambios de aceleración fuera de la banda de paso, pero se desconoce la forma exacta de éste. Después, las muestras se suman a lo largo de épocas [43].

Un estudio llevado a cabo en 2014 [44] muestra la forma aproximada del filtro utilizado por ACT. En el presente trabajo se trató de reproducir tal filtro pero no se obtuvieron resultados comparables, por lo que se decidió tratar de aproximar la función que mapea entre RD y *counts* mediante aprendizaje automático.

La idea puede ser descrita del siguiente modo:

$$\begin{array}{ccc}
 \mathbf{x} \longrightarrow f_{\mathbf{x}}^i (i = 1 \dots M) \longrightarrow I_N(f_{\mathbf{x}}^i) & & \\
 & \uparrow & \\
 \mathbf{y} \longrightarrow \hat{\mathbf{x}} \simeq g(\mathbf{y}) \longrightarrow f_{\hat{\mathbf{x}}}^i (i = 1 \dots M) & & (4.1)
 \end{array}$$

donde  $\mathbf{x}$  son los *counts* de ACT, para los que se hallan M medidas  $f_{\mathbf{x}}^i (i = 1 \dots M)$ , cada una de las cuales posee su intervalo de normalidad  $I_N(f_{\mathbf{x}}^i)$ . Por otro lado,  $\mathbf{y}$  son los datos de aceleración de MSB, que a través de la aproximación  $g(\mathbf{y})$  se transforman en *counts*  $\hat{\mathbf{x}}$ . Se calcularán las mismas medidas para estos *counts* y se comprobará si tales  $f_{\hat{\mathbf{x}}}^i (i = 1 \dots M)$  pertenecen a los intervalos hallados con anterioridad.

La aproximación  $g(\mathbf{y})$  se llevará a cabo mediante aprendizaje automático.

## 4.2 PREPROCESADO

A fin de mapear el RD de MSB a *counts* de ACT se registran datos de forma simultánea de ACT y MSB, puesto que de ACT es posible extraer el RD y los *counts*. Los registros de MSB se configuran en intervalos más cortos, con el objetivo de alinearlos de la forma más precisa posible a los datos de ACT. No todos los registros fueron válidos (no se consiguieron alinear todos los registros, de forma visual se discriminaron aquellos que ni tan siquiera se aproximaban). Se construyó una base de datos para llevar a cabo el entrenamiento de las técnicas de aprendizaje automático comentadas en la sección 4.3.

### 4.2.1 EXTRACCIÓN DE DATOS

Una vez realizados los registros simultáneos de ambos actímetros se procede a la extracción de los datos. Por un lado, cada registro de ActiGraph contiene dos ficheros, uno de ellos contiene los *counts*, además de otros datos, y otro contiene el RD, previo al procesado de ActiLife. Por otro lado, cada registro de Microsoft Band 2 consiste en un único fichero de texto que contiene el RD. Se deben adecuar estos datos para el posterior procesado.

#### REGISTROS ADQUIRIDOS CON ACTIGRAPH

Los datos recogidos por el monitor de ACT se encuentran guardados en la memoria del actímetro, por lo que se han de descargar los dos ficheros de que consta cada registro a través del software ActiLife. Puesto que la versión de ActiLife a la que se tiene acceso no permite la propia extracción y visualización de los datos, se ha de recurrir a software externo. Para extraer los datos de interés de tales ficheros se ha optado por seguir una metodología distinta en función de la extensión del fichero, la cual se expone a continuación:

- **GT3X** - El fichero con la extensión \*.gt3x contiene el *raw data*, es decir, los datos crudos de aceleración previos al procesado de ActiLife en los tres ejes.  
Con la ayuda de la aplicación GT3XParser.jar<sup>1</sup>, la cual es un fichero jar ejecutable y accesible desde línea de comandos, convertimos cada fichero \*.gt3x a \*.csv [45]. Dicho fichero \*.csv será leído por Matlab para su volcado a \*.mat.
- **AGD** - El fichero con la extensión \*.agd contiene, además de otros datos, los *counts* proporcionados por ActiLife para los tres ejes tras el procesado del *raw data*.  
Dado que los ficheros AGD son bases de datos con arquitectura SQLite [46], se ha optado por utilizar *DB Browser for SQLite* para exportar los *counts* a un fichero \*.csv. Ese fichero \*.csv será leído por Matlab para su volcado a \*.mat.

Como se ha comentado, los ficheros \*.csv resultantes se leen con Matlab para realizar el volcado de los datos a ficheros \*.mat y que, de este modo, sean accesibles de manera inmediata. Para ello, se hará uso de una función propia que, a partir de la ruta del fichero \*.csv y el tipo de datos a ser leídos, guarda los datos en una matriz. El código de tal función se observa a continuación:

```

1 function B = lecturaCSV(rutaFichero,tipo)
2 % Selección del tipo de datos: counts/raw
3     switch lower(tipo)
4         case {'counts','c'}

```

<sup>1</sup>Hubo que hacer una pequeña modificación

```

5         % Lee el fichero en el struct A y guarda los datos en la matriz B
6         A = importdata(rutaFichero, ',', 1);
7         B = A.data(:, 2:4); % Axis 1, Axis 2, Axis 3 (Ejes Y, X, Z de ActiGraph)
8     case {'raw', 'r'}
9         A = importdata(rutaFichero, ',', 1);
10        B = A.data; % Ejes X, Y, Z de ActiGraph
11    end
12 end

```

## REGISTROS ADQUIRIDOS CON MICROSOFT BAND 2

El fichero de texto que contiene los datos de aceleración registrados con MSB consiste en una cadena de texto en la que los datos se encuentran separados por espacios en blanco. Los datos se encuentran organizados de forma que cada tres números constituyen los datos de los tres ejes de una misma muestra. La lectura del fichero \*.txt y el volcado de los datos a una matriz se realizará con una función propia, cuyo código se expone a continuación:

```

1 function B = lecturaTXT(rutaFichero)
2     % Lee el archivo de datos numericos ASCII delimitado en el vector A
3     A = dlmread(rutaFichero);
4     % Convierte el vector A en una matriz con 3 columnas, creando una fila cada vez
5     B = vec2mat(A, 3); % Ejes X, Y, Z de Microsoft Band 2
6 end

```

### 4.2.2 REMUESTREO Y ALINEACIÓN

El RD de ACT se encuentra muestreado a una frecuencia de 30Hz, mientras que los datos de MSB lo hacen a una frecuencia mayor, especificada en 31.25Hz. El RD de MSB no cuenta con un sello temporal que indique cuándo fue capturado por la pulsera. Por ello y por retardos aleatorios en la comunicación vía Bluetooth entre pulsera y smartphone, los instantes en los que se registran los datos en el smartphone no pueden ser asumidos como exactos, por lo que se propone alinear ambas señales. Para llevar a cabo la alineación de las señales se utiliza la correlación, que proporciona una medida de la similitud de las señales. Se llevará a cabo un submuestreo de los datos de MSB con el objetivo de reducir la frecuencia de muestro a 30Hz para adecuar las señales y poder alinear dichos registros simultáneos.

Para alinear los registros de los diferentes actímetros se utilizará el vector magnitud resultante del RD de los tres ejes, es decir,  $r = \|(x, y, z)\| = \sqrt{x^2 + y^2 + z^2}$ .

Se debe tener en cuenta que ACT proporciona el RD denominándolo como Eje X, Eje Y y Eje Z, mientras que los *counts* los referencian como Axis 1, Axis 2, y Axis 3. Por otro lado, como se mostró en las figuras 2.1(b) y 2.2(b), los ejes del monitor de ACT y los de la pulsera MSB no representan las mismas direcciones. La correspondencia entre todos estos datos se encuentra expuesta en la tabla 4.1<sup>2</sup>.

<i>counts</i>	ACT	MSB
Axis 1	Y	X
Axis 2	X	Y
Axis 3	Z	Z

TABLA 4.1: Correspondencia entre ejes.

<sup>2</sup>Se contactó con el soporte de ACT para confirmarlo

Una vez alineadas las señales de ACT y de MSB, y considerando válidos los *counts* de ACT, debemos seleccionar los *counts* de ACT que se correspondan con el registro de MSB, teniendo en cuenta que dadas las frecuencias de muestreo de unos y otros, se seleccionará una muestra de *counts* por cada 30 muestras de RD.

### 4.3 APROXIMACIÓN DE FUNCIONES Y APRENDIZAJE AUTOMÁTICO

---

Tras la obtención de los datos crudos de ACT y MSB alineados y la extracción de los correspondientes *counts* (una muestra de *counts* por cada 30 muestras de RD), el siguiente paso consiste en obtener *counts* a partir de los datos crudos de MSB. Puesto que contamos con el RD de MSB y los *counts* de ACT que se corresponden tras alinear las señales, los cuales suponemos válidos, sólo queda aproximar la función que proporciona *counts* a partir de RD. Para ello, se hará uso de aprendizaje automático o aprendizaje máquina (*machine learning*).

El aprendizaje automático enseña a los ordenadores a aprender de la experiencia. Los algoritmos de aprendizaje automático utilizan métodos computacionales para aprender información directamente de los datos sin basarse en un modelo determinado por una ecuación. Por lo general, los algoritmos mejoran su rendimiento de manera adaptativa al tiempo que aumenta el número de muestras disponibles para el aprendizaje; es decir, a mayor número de muestras disponibles, mejores resultados [47].

El aprendizaje automático utiliza dos tipos de técnicas: aprendizaje supervisado, que entrena un modelo a partir de entradas y salidas conocidas de tal forma que puede predecir futuras salidas, es decir, un conjunto conocido de datos de entrada-salida se utiliza para ajustar de forma iterativa los pesos de la red; y aprendizaje no supervisado, que encuentra patrones ocultos o estructuras intrínsecas en los datos de entrada, donde únicamente se disponen de datos de entrada y una función de coste a minimizar. En ocasiones, también se habla de aprendizaje reforzado, en el cual la información proporcionada a la red es mínima, limitándose a indicar si la respuesta de la red es correcta o incorrecta.

Por un lado, el aprendizaje supervisado utiliza técnicas de clasificación y regresión para desarrollar sus modelos predictivos. Mientras que las técnicas de clasificación predicen respuestas discretas, es decir, clasifican los datos de entrada en categorías, las técnicas de regresión predicen respuestas continuas, aproximan funciones. Por otro lado, el aprendizaje no supervisado utiliza, entre otros métodos, algoritmos de *clustering*, que consisten en particionar los datos en grupos de instancias tales que las instancias en un *cluster* sean similares entre sí, y diferentes de las instancias en otros *clusters*.

El presente problema se ajusta mejor a la técnica de aprendizaje supervisado, pues las entradas y las salidas son conocidas, queriendo predecir futuras respuestas. Además, tratamos de aproximar una función, y no clasificar, por lo que nos centraremos en técnicas de regresión; y entre las disponibles, utilizaremos redes neuronales y bosques aleatorios. En la figura 4.1 se puede observar una posible jerarquía de las técnicas expuestas.

#### 4.3.1 REDES NEURONALES

---

Las redes neuronales artificiales, son una técnica de aprendizaje y procesamiento automático inspirada en el funcionamiento del cerebro humano, es decir, tratan de reproducir el proceso de solución de problemas del cerebro. Un red neuronal se puede definir como una estructura de procesamiento paralelo masivo constituida por unas unidades muy sencillas, denominadas neuronas, que tienen la capacidad de almacenar conocimiento experimental y ponerlo a disposición para su uso; toma como ejemplos problemas resueltos para construir un sistema que toma decisiones y realiza clasificaciones o aproximaciones [48]. Se

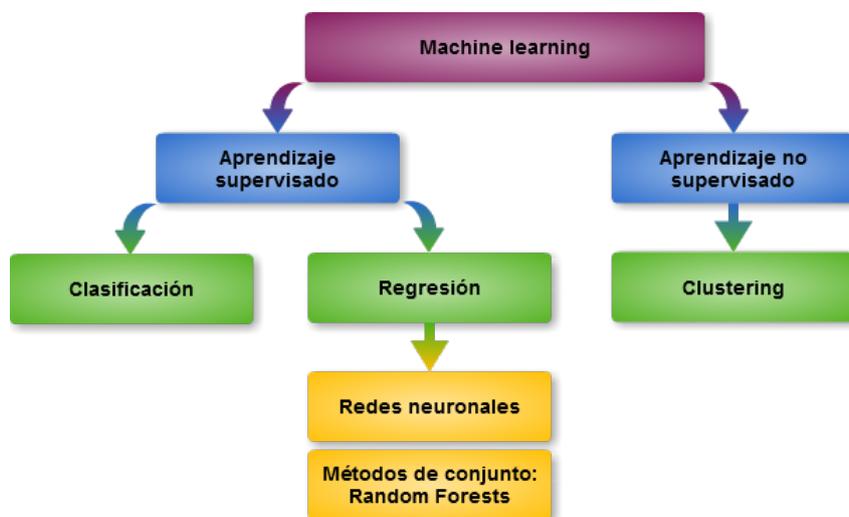


FIGURA 4.1: Jerarquía de técnicas de aprendizaje automático.

pueden implementar tanto para aprendizaje supervisado como para aprendizaje no supervisado.

Como se ha dicho, las redes neuronales están compuestas de unos elementos de cómputo denominados neuronas. Cada neurona está compuesta de dos unidades: la primera unidad suma los productos de las entradas por sus respectivos pesos, y la segunda unidad contiene la función de activación, también denominada función de transferencia [48].

La arquitectura (o topología) de la red hace referencia a la disposición de las neuronas en la red. Las neuronas se organizan formando capas, de modo que la red neuronal puede consistir en una o más capas de neuronas. La capa que contiene las salidas de la red se conoce como capa de salida y el resto de capas como capas ocultas.

Para “enseñar” a la red neuronal es necesario entrenar un conjunto de datos, el cual consiste en señales de entrada asignadas con objetivos correspondientes (salidas deseadas). El entrenamiento es un proceso iterativo. En cada iteración los pesos de los nodos se modifican usando nuevos datos del conjunto de entrenamiento [48].

Los tipos de redes neuronales más utilizados son el perceptrón simple, el perceptrón multicapa (MLP, *MultiLayer Perceptron*), las redes de base radial (RBF, *Radial Basis Function*) y los mapas auto-organizados (SOM, *Self-Organizing Map*) [48–50]. Los tres primeros se sirven de métodos de aprendizaje supervisado, mientras que el último, típicamente, se basa en aprendizaje no supervisado.

Si nos centramos en los perceptrones, el perceptrón simple es un tipo de red neuronal monocapa constituida por un conjunto de sensores de entrada que reciben los patrones de entrada a reconocer o clasificar y una neurona de salida que se ocupa de clasificar los patrones de entrada en dos clases, según que la salida de la misma sea 1 (activada) o 0 (desactivada). Es limitado en el sentido de que sólo sirve para clasificar problemas linealmente separables. Por otro lado, el perceptrón multicapa es una extensión y generalización del perceptrón simple en el que se añaden una o más capas ocultas y las funciones de activación son, comúnmente, de tipo sigmoide. Dado que el problema a resolver es un problema de regresión y no de clasificación, se usarán redes MLP.

El entrenamiento de los MLP está basado en algoritmos de “retropropagación” del error o *backpro-*

*pagation*, donde los pesos se actualizan mediante el uso de un conjunto ordenado de entradas y salidas deseadas y la comparación entre dicha salida y la salida real de la red neuronal [48]; es decir, la salida de la red se compara con el valor objetivo deseado. La diferencia se denomina error de la señal. Es imposible conocer el error en las neuronas de las capas internas directamente, debido a que los valores de salida de estas neuronas son desconocidos. El algoritmo de retropropagación propaga el error de regreso a todas las neuronas, cuya salida fue la entrada de la última neurona. Posteriormente, el error se va propagando a las neuronas de capas anteriores considerando los pesos de las conexiones.

En la figura 4.2 se observa un simple esquema del entrenamiento de redes neuronales:

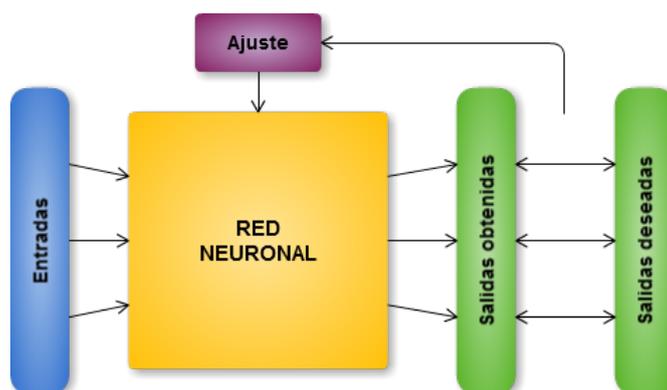


FIGURA 4.2: Entrenamiento de redes neuronales artificiales.

Existen diferentes variaciones de algoritmos de retropropagación donde cada uno tiene ciertas ventajas y desventajas dependiendo de la arquitectura de la red y de la complejidad del problema. Algunos de los algoritmos de entrenamiento con retropropagación del error son *Levenberg-Marquardt backpropagation* (LM), *Bayesian regularization backpropagation* (BR), *Scaled conjugate gradient backpropagation* (SCG) y *Resilient backpropagation* (RP).

RP se utiliza en reconocimiento de patrones. LM tiene un rendimiento pobre en reconocimiento de patrones, pero bueno en aproximación de funciones con redes de hasta pocos cientos de pesos.

SCG funciona bien sobre una amplia variedad de problemas, particularmente para redes con un gran número de pesos. El algoritmo SCG es casi tan rápido como el algoritmo LM en problemas de aproximación de funciones y es casi tan rápido como RP en problemas de reconocimiento de patrones. Su rendimiento no se degrada tan rápidamente como el de RP cuando se reduce el error [51].

BR es un algoritmo de entrenamiento que actualiza los pesos y valores de sesgo según la optimización de Levenberg-Marquardt. Minimiza una combinación de errores cuadrados y pesos, y luego determina la combinación correcta para producir una red que generaliza bien. El proceso se denomina regularización bayesiana.

En cuanto a las funciones de activación a utilizar, cabe destacar que varias capas de neuronas con funciones de transferencia no lineal permiten a la red aprender relaciones no lineales entre los vectores de entrada y de salida. La capa de salida lineal se usa con mayor frecuencia para problemas de ajuste de funciones (o regresión no lineal). Por otro lado, si se desea restringir las salidas de una red (como entre 0 y 1), entonces la capa de salida debe utilizar una función de transferencia sigmoide (como la función logística,  $a = \frac{1}{1+e^{-x}}$ ). Este es el caso cuando la red se utiliza para problemas de reconocimiento de patrones (en los que se está tomando una decisión por parte de la red).

Para tratar de encontrar solución al presente problema, se implementarán redes de una y dos capas ocultas con función de activación logística y capa de salida con función de activación lineal. Estas redes serán entrenadas con los algoritmos BR y SCG.

### 4.3.2 BOSQUES ALEATORIOS

Un árbol de decisión es una estructura de datos simple y determinista para modelar reglas de decisión para un problema de clasificación específico. En cada nodo, se selecciona una característica para realizar una decisión. Puesto que los árboles de decisión no clasifican o aproximan particularmente bien dado que tienen tendencia a sobreajustar, se proponen los métodos de conjunto o *ensemble methods*, existiendo dos estrategias principales: *bagging* y *boosting* [52].

Los bosques aleatorios o *random forests* son un conjunto de árboles de decisión tales que cada árbol depende de los valores de un vector de muestras aleatorias independientes y con la misma distribución para todos los árboles del bosque [53], siguiendo la estrategia *bagging*.

Los bosques aleatorios combinan los resultados de muchos árboles de decisión, lo que reduce los efectos del sobreajuste y mejora la generalización. No sobreajustan más cuantos más árboles se añadan, aunque sí existe un valor límite para el error de generalización [53].

Para entrenar los bosques aleatorios siguiendo la estrategia *bagging* se generan distintos subconjuntos de entrenamiento a partir del conjunto total de datos mediante el método de *bootstrapping*, lo que significa que se generan nuevos conjuntos de datos con el mismo tamaño usando un muestreo con reemplazo, y se ajusta un modelo o árbol para cada uno de ellos. Una simplificación se observa en la figura 4.3.

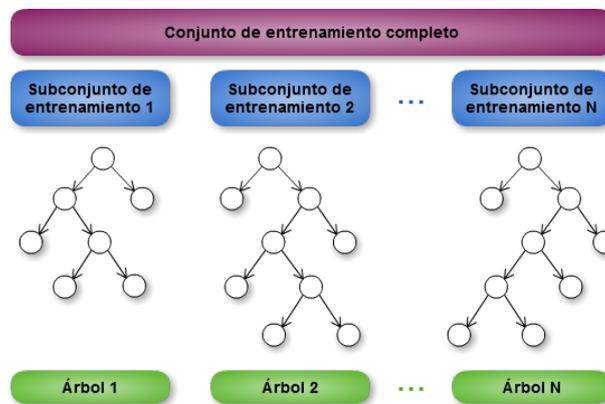


FIGURA 4.3: Entrenamiento de bosques aleatorios.

En cada paso del entrenamiento, se utilizan aproximadamente dos tercios de los datos para ajustar el árbol correspondiente y el resto de datos se consideran observaciones *out of bag* (OOB). Cada árbol se utiliza para predecir los valores de las observaciones OOB, de manera que las predicciones se hacen siempre para observaciones que no han sido usadas para construir el modelo. Finalmente, para cada observación del conjunto total de los datos se tendrá una serie de predicciones (correspondientes a las veces en las que la observación ha formado parte del conjunto OOB), que se promediarán para obtener una predicción de “consenso”. Se ajustan árboles muy profundos (sin podar), lo que implica una gran varianza, pero cuando se toman en conjunto, se reduce la varianza y se aumenta la precisión de las predicciones.

Un sencillo ejemplo de predicción puede observarse en la figura 4.4. En este ejemplo tenemos tres árboles. A los tres árboles entra el mismo dato. Cada árbol toma su decisión. Finalmente se promedian todas las decisiones.

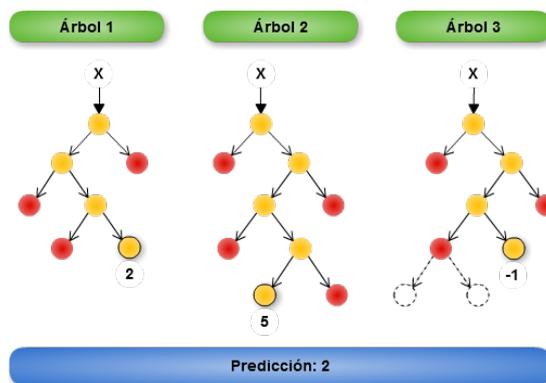


FIGURA 4.4: Ejemplo de predicción con bosque aleatorio de 3 árboles.

#### 4.4 IDENTIFICACIÓN DE INTERVALOS DE ACTIVIDAD Y DESCANSO

Dado que la actimetría es un método ampliamente utilizado en estudios de patologías del sueño, la identificación de intervalos o tramos de actividad y descanso facilita el análisis de tales trastornos.

Esta identificación de intervalos se ha llevado a cabo de forma automática con el método propuesto en [5, 42], el cual está basado en un suavizado de la señal seguido de comparación con un umbral, llevándose a cabo la selección de dicho umbral de forma iterativa. Esta identificación de intervalos de actividad/descanso únicamente ha sido posible para los registros de ACT, dada su duración (aproximadamente 24 h). De esta forma se ha podido analizar independientemente la señal completa, así como la actividad y los intervalos de descanso. Como periodo de descanso se ha considerado el intervalo de tiempo que el sujeto pasa en la cama durante la noche; el resto del registro se considera como periodo de actividad.

Los tramos de sólo actividad se han dividido en actividad durante la mañana y actividad durante la tarde, y los tramos de descanso se han dividido en tres periodos de la misma longitud (inicio, parte intermedia y final del descanso).

#### 4.5 DIEZMADO

El diezclado de los registros es una etapa necesaria para mejorar la información relacionada con la actividad que pueda estar oculta por una gran cantidad de muestras de valor cero en la señal original, además permite interpretar los resultados en función de la actividad asociada a movimientos o a tareas. En [6] se propone crear una nueva señal que consiste en una secuencia de intervalos promediados del siguiente modo:

$$x_{dec}[k] = \frac{1}{M} \sum_{i=0}^{M-1} x[kM + i], 0 \leq k \leq \left\lfloor \frac{N}{M} \right\rfloor \quad (4.2)$$

con  $N$  el número total de muestras. Cada muestra de  $x_{dec}$  suma la información de  $M$  muestras de  $x[n]$ , es decir, las muestras de  $x_{dec}$  son épocas de  $M/fs$  segundos, siendo  $fs$  una muestra por segundo.

El parámetro  $M$  se ha seleccionado entre las siguientes opciones: 1 s, 5 s, 15 s, 30 s, 1 min, 5 min, 10 min y 15 min. Esta elección nos permite abordar las épocas cortas (del orden de los segundos) que representan la actividad de movimientos y grandes épocas (varios minutos) que representan la actividad de las tareas.

## 4.6 CÁLCULO DE CARACTERÍSTICAS A PARTIR DE COUNTS

Las siguientes medidas serán aplicadas a la base de datos disponible de ACT, que únicamente contiene *counts*, con el objeto hallar los intervalos de normalidad en adultos, previa segmentación de las señales en diferentes intervalos o tramos. Las mismas medidas serán aplicadas a los registros de MSB de modo que sea posible la comparación entre ambos actímetros. Se distinguen dos bloques: el bloque de métodos tradicionales de estadística descriptiva, al cual pertenecen las medidas de centralidad (4.6.1), de dispersión (4.6.2) y de distribución (4.6.3), y el bloque de análisis no lineal (4.6.4), que permite extraer características adicionales a las obtenidas mediante el primer bloque.

### 4.6.1 MEDIDAS DE CENTRALIDAD

Las medidas de centralidad permiten identificar y ubicar los valores más representativos de los datos.

#### Media

La media se define como la suma de todos los valores observados, dividido por el número total de observaciones. Su principal desventaja radica en su sensibilidad al cambio de uno de sus valores o a los valores extremos demasiado grandes o pequeños.

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i \quad (4.3)$$

#### Mediana

La mediana representa el valor de la variable de posición central en un conjunto de datos ordenados. Esta medida nos indica que la mitad de los datos se encuentran por debajo de este valor y la otra mitad por encima del mismo.

Sean  $x_1, x_2, x_3, \dots, x_n$  los datos de una muestra ordenada en orden creciente, distinguimos dos casos:

- Si  $n$  es impar, la mediana es el valor que ocupa la posición  $(n+1)/2$  una vez que los datos han sido ordenados (en orden creciente o decreciente), porque éste es el valor central. Es decir:  $M_e = x_{(n+1)/2}$ .
- Si  $n$  es par, la mediana es la media aritmética de los dos valores centrales. Cuando  $n$  es par, los dos datos que están en el centro de la muestra ocupan las posiciones  $n/2$  y  $n/2 + 1$ . Es decir:  $M_e = (x_{n/2} + x_{n/2+1})/2$ .

### 4.6.2 MEDIDAS DE DISPERSIÓN

---

Las medidas de dispersión permiten reconocer cómo se desvían las observaciones alrededor de su media. Las medidas de dispersión a estudiar serán la desviación estándar (o típica), el rango intercuartílico y el coeficiente de variación.

#### Desviación estándar

La desviación estándar permite determinar el promedio aritmético de fluctuación de los datos respecto a su punto central o media. La desviación estándar da como resultado un valor numérico que representa el promedio de diferencia que hay entre los datos y la media. Para calcular la desviación estándar basta con hallar la raíz cuadrada de la varianza, por lo tanto su ecuación sería:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |A_i - \mu|^2} \quad (4.4)$$

donde  $\mu$  es la media de A.

#### Rango intercuartil

El rango intercuartil es una estimación estadística de la dispersión de una distribución de datos. Consiste en la diferencia entre el tercer y el primer cuartil. Mediante esta medida se eliminan los valores extremadamente alejados. El rango intercuartil es altamente recomendable cuando la medida de centralidad utilizada es la mediana (ya que este estadístico es insensible a posibles irregularidades en los extremos).

$$IQR = Q_3 - Q_1 \quad (4.5)$$

#### Coefficiente de variación

El coeficiente de variación (CV) es una medida de la dispersión relativa de un conjunto de datos, que se obtiene dividiendo la desviación típica de una muestra entre su media. Permite comparar las dispersiones de dos distribuciones distintas, siempre que sus medias sean positivas.

$$CV = \frac{\sigma}{\mu} \quad (4.6)$$

### 4.6.3 MEDIDAS DE DISTRIBUCIÓN

---

Las medidas de distribución permiten identificar la forma en que se separan o concentran las observaciones. Su utilidad radica en la posibilidad de identificar las características de la distribución sin necesidad de generar un gráfico.

### Coefficiente de asimetría

La asimetría o *skewness* es una medida que permite identificar si los datos se distribuyen de manera uniforme alrededor de la media. Si el coeficiente de asimetría es negativo, los datos se distribuyen más a la izquierda de la media que a la derecha. Si el coeficiente de asimetría es positivo, los datos se distribuyen más hacia la derecha. El coeficiente de asimetría de la distribución normal (o cualquier distribución perfectamente simétrica) es cero.

El coeficiente de asimetría de una distribución se define como:

$$s = \frac{E\{(x - \mu)^3\}}{\sigma^3} \quad (4.7)$$

donde  $\mu$  es la media de  $x$ ,  $\sigma$  es la desviación estándar de  $x$ , y  $E\{t\}$  representa el valor esperado de la cantidad  $t$  (esperanza matemática). Por lo tanto, puede expresarse como sigue:

$$s = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}\right)^3} \quad (4.8)$$

Los resultados pueden interpretarse del siguiente modo:

- Si  $s = 0$ : se acepta que la distribución es simétrica, es decir, existe aproximadamente la misma cantidad de valores a los dos lados de la media. Este valor es difícil de conseguir por lo que se tiende a tomar los valores que son cercanos ya sean positivos o negativos ( $\pm 0,5$ ).
- Si  $s > 0$ : la curva es asimétricamente positiva por lo que los valores tienden a reunirse más en la parte derecha que en la izquierda de la media.
- Si  $s < 0$ : la curva es asimétricamente negativa por lo que los valores tienden a reunirse más en la parte izquierda de la media.

### Curtosis

La curtosis o *kurtosis* es una medida que sirve para analizar el grado de concentración que presentan los valores de una variable alrededor de la región central de la distribución. La curtosis de la distribución normal es 3. Las distribuciones que son más concentradas que la distribución normal tienen curtosis mayor que 3, mientras que las distribuciones que son menos concentradas tienen curtosis menor que 3.

La curtosis de una distribución se define como:

$$k = \frac{E\{(x - \mu)^4\}}{\sigma^4} \quad (4.9)$$

donde  $\mu$  es la media de  $x$ ,  $\sigma$  es la desviación estándar de  $x$ , y  $E\{t\}$  representa el valor esperado de la cantidad  $t$  (esperanza matemática). Puede calcularse con la ecuación que aparece a continuación:

$$k = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}\right)^2} \quad (4.10)$$

Los resultados pueden interpretarse del siguiente modo:

- Si  $k = 3$ : la distribución es mesocúrtica, lo que supone una concentración normal. Al igual que ocurre con la asimetría, se suelen aceptar valores cercanos ( $\pm 0,5$ ).
- Si  $k > 3$ : la distribución es leptocúrtica, lo que identifica una gran concentración de valores.
- Si  $k < 3$ : la distribución es platicúrtica, determinando una baja concentración de valores.

#### 4.6.4 ANÁLISIS NO LINEAL

El análisis no lineal es una herramienta valiosa para cuantificar la variabilidad y/o complejidad de los registros de actimetría. La cuantificación de la variabilidad de las señales de actividad mediante métodos no lineales permite extraer conclusiones adicionales a las obtenidas mediante métodos tradicionales. Existen numerosos métodos que permiten cuantificar la variabilidad de una serie temporal considerando que dicha serie representa la evolución temporal de un sistema complejo constituido por un número determinado de variables [2].

#### Medida de la Tendencia Central

La medida de tendencia central (CTM) es una medida de la regularidad de una serie temporal basada en el cálculo del número de incrementos consecutivos de la señal que están por debajo de un determinado umbral a definir. Si dicho número es alto, la serie temporal se puede considerar como regular, mientras que si es bajo, la complejidad de la serie será alta. Obviamente, el valor del umbral determina qué se puede considerar como complejo o regular a efectos de la medida. Para obtener la CTM se parte de un diagrama de dispersión que representa la serie de diferencias de primer orden cuyos ejes son (para una señal  $x[n]$ ,  $n = 0, 2, \dots, N - 1$ )  $x[n + 2] - x[n + 1]$  y  $x[n + 1] - x[n]$ . La CTM se calcula como el número de puntos que caen dentro de un círculo de radio  $\rho$  alrededor del origen, dividido por el número de puntos total. El valor de  $\rho$  se relaciona de forma directa con el umbral mencionado.

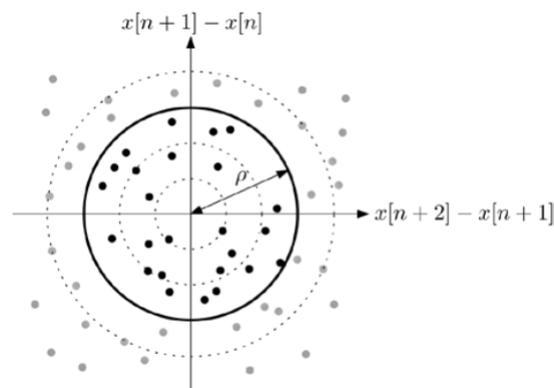


FIGURA 4.5: Representación esquemática del cálculo de la CTM.

La CTM se define como:

$$CTM = \frac{1}{N-2} \sum_{i=0}^{N-3} \delta_n \quad (4.11)$$

donde

$$\delta_n = \begin{cases} 1 & \text{si } \sqrt{(x[n+2] - x[n+1])^2 + (x[n+1] - x[n])^2} \leq \rho \\ 0 & \text{en otro caso} \end{cases} \quad (4.12)$$

### Dinámica simbólica

La dinámica simbólica (DS) se basa en el cálculo de las probabilidades de ocurrencia de palabras en una versión cuantificada y codificada (denotada como  $c[n]$ ) de la señal original. La longitud del alfabeto y el número de símbolos por palabra son los parámetros de esta técnica.

El proceso de cuantificación consiste en dividir la señal en determinados rangos (actividad baja, media o alta en este caso) y la codificación en asignar un símbolo a dichos rangos (“0”, “1” o “2”). Agrupando los diferentes símbolos en palabras formadas por dos o más de ellos, es posible cuantificar la complejidad de la serie mediante el cómputo de la mencionada frecuencia. Por ejemplo, una probabilidad (frecuencia relativa) alta de la cadena “00” denota una baja complejidad (símbolos iguales seguidos) asociada a una baja actividad (símbolo “0”). Por el contrario, la aparición de numerosas instancias de la cadena “02” revela una alta complejidad de la serie (numerosas transiciones) con saltos de gran amplitud (transiciones del mínimo al máximo rango de actividad).

En este trabajo, y tomando como referencia [6] se ha elegido un alfabeto de tres símbolos y palabras de dos símbolos de longitud. Cada símbolo representa un estado, es decir, un nivel de actividad (bajo, medio y alto) y las palabras representan transiciones entre dos estados, es decir, variabilidad. En cuanto a la cuantificación de señales, se definen dos umbrales,  $q_1$  y  $q_2$ , como los percentiles 33 y 66 de  $x[n]$ . Cada muestra de la señal cuantificada se define de la siguiente manera:

$$c[n] = \begin{cases} 0 & \text{si } x[n] \leq q_1 \\ 1 & \text{si } q_1 < x[n] < q_2 \\ 2 & \text{en otro caso} \end{cases} \quad (4.13)$$

Se obtienen palabras de dos símbolos del siguiente modo:

$$w_2[k] = \begin{bmatrix} c[k] \\ c[k+1] \end{bmatrix}. \quad (4.14)$$

dentro del rango de valores posibles de  $k$ . Con estos parámetros existen  $3^2 = 9$  posibles palabras de dos símbolos que se pueden formar con el alfabeto seleccionado.

Otro índice utilizado habitualmente en el análisis basado en DS es la entropía de Shannon  $E_{Shannon}$ , obtenida como:

$$E_{Shannon} = \sum_{i=1}^{N_A} p_i \log(p_i) \quad (4.15)$$

siendo  $p_i$  la probabilidad de aparición de la palabra  $i$ -ésima. Esta magnitud toma mayores valores a mayor complejidad de la señal.

## 4.7 INTERVALOS DE NORMALIDAD

---

Considerando que los sujetos de estudio son normales, es decir, que pertenecen a un grupo de control, se hallarán los intervalos de normalidad para cada una de las medidas o características expuestas en la sección 4.6.

Además, dado que las señales disponibles de ACT consisten en registros de 24 horas, aproximadamente, lo que posibilita su fragmentación en diferentes tramos [5, 42], se hallarán los intervalos de normalidad en estos tramos. Los periodos sujetos a estudio serán, por lo expuesto en la sección 4.4:

- Día completo, es decir, análisis de la señal original en su totalidad.
- Actividad
  - Actividad durante la mañana
  - Actividad durante la tarde
- Descanso
  - Inicio del descanso
  - Parte intermedia del descanso
  - Última parte del descanso

## Capítulo 5

# RESULTADOS

---

Una vez explicados los métodos utilizados en este estudio, en este capítulo se exponen los resultados obtenidos, previa breve descripción de las bases de datos de registros de ACT y MSB (5.1). En primer lugar, se incluirán algunos resultados en lo que a remuestreo y alineación respecta (5.2). Después, se expondrán los resultados obtenidos en el intento de aproximar la función que traduce el RD de MSB a *counts* (5.3). En último lugar, se presentarán los resultados obtenidos tras llevar a cabo las medidas expuestas en la sección 4.6 sobre estos datos y los de ACT (5.4).

## 5.1 REGISTRO Y BASES DE DATOS

---

Para la realización de este estudio se dispone de una base de datos de 436 adultos, cuyos registros fueron realizados con el monitor de ACT durante periodos de más de 20 horas y que únicamente contiene los *counts*. Dicha base de datos comprende sujetos con edades entre 16 y 66 años ( $47.5636 \pm 13.8283$ ) y será utilizada para la extracción de características e intervalos de normalidad de referencia.

Con objeto de adquirir medidas de actividad simultáneas con ambos actímetros, ACT y MSB, se contactó con 12 voluntarios, 5 mujeres y 7 hombres, con edades comprendidas entre los 15 y los 56 años. Estos registros son de corta duración (entre 13 minutos y 83 minutos), y representan ejercicios tales como andar y actividad física espontánea. En la tabla 5.1 se exponen características de estos voluntarios como son su sexo, edad y su mano no dominante.

Sujeto	Sexo	Edad	Mano no dominante
1	M	24	Derecha
2	M	24	Derecha
3	H	29	Izquierda
4	H	25	Izquierda
5	H	25	Izquierda
6	H	25	Izquierda
7	M	15	Izquierda
8	H	16	Izquierda
9	M	48	Izquierda
10	H	56	Izquierda
11	M	22	Izquierda
12	H	24	Izquierda

TABLA 5.1: Base de datos con registros simultáneos de ACT y MSB.

Cabe destacar que existen diferencias significativas entre las edades de ambos grupos, obteniéndose un  $p$ -valor de  $2.4469 \cdot 10^{-5}$  mediante la prueba de suma de rangos de Wilcoxon, también denominada prueba U de Mann-Whitney.

## 5.2 REMUESTREO Y ALINEACIÓN

Como muestra del resultado obtenido tras alinear los registros simultáneos de ACT y MSB se exponen las figuras 5.1 y 5.2, las cuales corresponden al RD de la misma sección de una señal. La primera de ellas representa el vector magnitud  $r$ , que es el utilizado como referencia para llevar a cabo la alineación. La segunda representa la aceleración registrada en un eje.

En las figuras 5.1(a) y 5.2(a) puede observarse la falta de alineación entre los registros, mientras que tras la alineación, los registros coinciden, de igual forma, independientemente del canal a observar (figuras 5.1(b) y 5.2(b)).

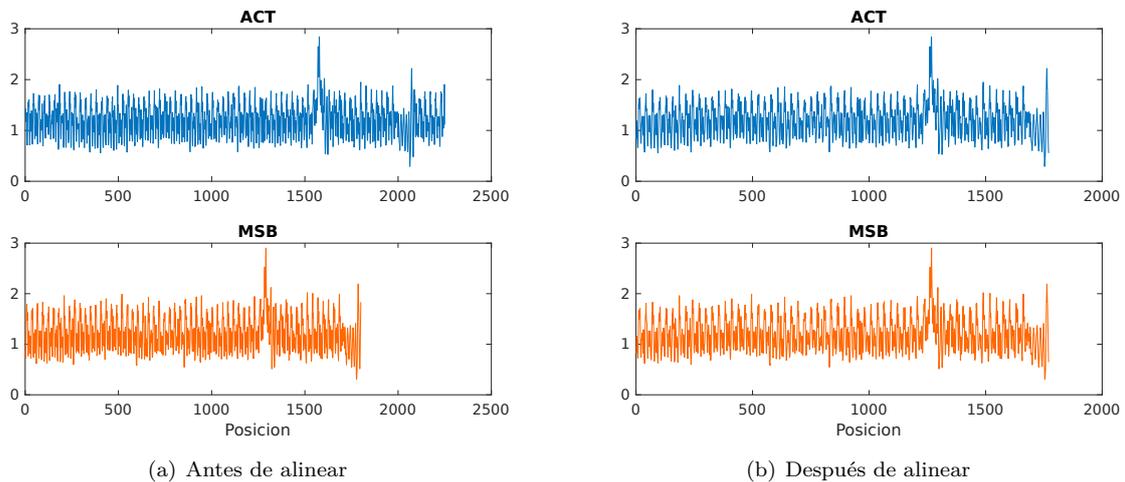


FIGURA 5.1: Alineación de señales de ACT y MSB. Vector magnitud  $r$ .

## 5.3 APROXIMACIÓN DE FUNCIONES

Con el objetivo de aproximar la función de paso de RD de MSB a *counts* se probaron diferentes técnicas de aprendizaje supervisado. Para discernir cuál de ellas es la adecuada nos basaremos en la correlación existente entre los *counts* obtenidos y los *counts* esperados, siendo estos últimos los *counts* de ActiGraph, de este modo evaluamos la similitud de las señales.

Sin embargo, el coeficiente de correlación solo mide la relación entre las dos variables, no su grado de acuerdo o concordancia. Para saber si este aproximador es fiable en comparación con los datos originales, se construye el gráfico de Bland-Altman en el que se representan el promedio de cada pareja de datos (*counts* estimados y *counts* deseados) frente a su diferencia. Además, en este gráfico se refleja el intervalo de confianza del 95 % de las diferencias [54].

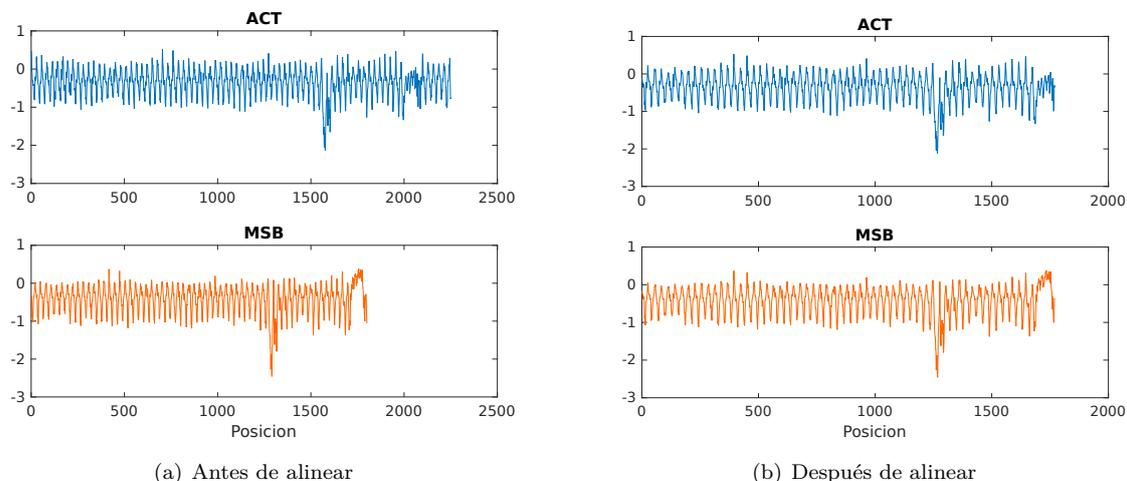


FIGURA 5.2: Alineación de señales de ACT y MSB. Eje.

El entrenamiento de redes neuronales y bosques aleatorios se llevó a cabo con los mismos datos.

### 5.3.1 REDES NEURONALES

En lo que a redes neuronales respecta, se entrenaron redes de una y dos capas ocultas, con función de activación logística en las capas ocultas y función lineal en la capa de salida. Los algoritmos de retro-propagación usados fueron BR y SCG. Los resultados se muestran en las tablas 5.2 y 5.3.

En la tabla 5.2 se muestran los coeficientes de correlación obtenidos tras entrenar redes neuronales con una capa oculta. Se realizó un barrido del número de neuronas para dicha capa oculta, desde 10 neuronas hasta 70 neuronas con incrementos de 10, para ambos algoritmos de entrenamiento (BR y SCG). Al incrementar el número de neuronas se incrementa la dimensionalidad del problema y, por lo tanto, el tiempo de entrenamiento. Puede observarse que el coeficiente de correlación más cercano a la unidad se obtuvo con la red neuronal con 10 neuronas en su capa oculta y algoritmo de entrenamiento BR, siendo éste 0.893005 (figura 5.3).

En la figura 5.4(a) se observa el diagrama de dispersión para esta red, la pendiente de la recta de regresión es de  $45.01^\circ$ , prácticamente la ideal. Por otro lado, en la figura 5.4(b) se observa el gráfico de Bland-Altman, en el que el 95.13 % de las diferencias se encuentran dentro del intervalo de confianza del 95 %.

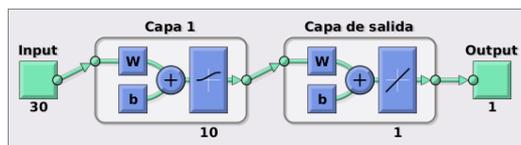


FIGURA 5.3: Diagrama de la red neuronal de una capa oculta con 10 neuronas.

Por otro lado, en la tabla 5.3 se muestran los coeficientes de correlación obtenidos tras entrenar redes neuronales con dos capas ocultas. Para la primera capa se realizó un barrido del número de neuronas

Número de neuronas	Correlación	
	BR	SCG
10	<b>0.893005*</b>	0.308166
20	0.874619	0.796977
30	0.882618	0.793715
40	0.838835	0.792789
50	0.856261	0.344939
60	0.812043	0.448926
70	0.766837	0.811687

TABLA 5.2: Correlación. Redes neuronales con 1 capa oculta.

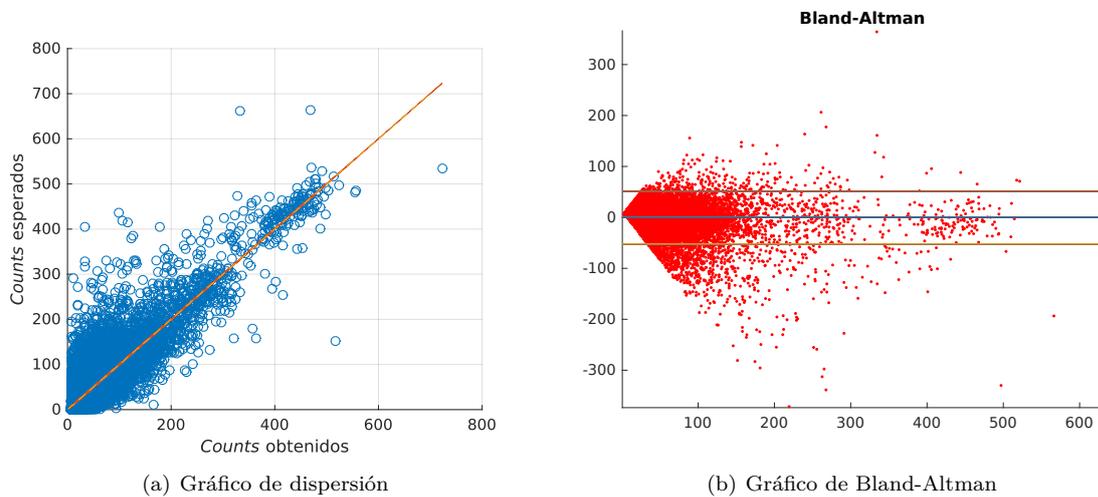


FIGURA 5.4: Gráficos de dispersión y de Bland-Altman de la red neuronal con una capa oculta de 10 neuronas con algoritmo de entrenamiento BR.

desde 10 neuronas hasta 40 neuronas con incrementos de 10, mientras que para la segunda el barrido se realizó desde 2 neuronas hasta 8 con incrementos de 2.

El mayor coeficiente de correlación, 0.899548, se obtuvo con la red neuronal con 10 neuronas en la primera capa oculta y 6 neuronas en la segunda capa oculta, y algoritmo de entrenamiento BR (figura 5.5). El diagrama de dispersión de esta red (figura 5.6(a)) posee una recta de regresión con una pendiente de  $44.98^\circ$ , muy próxima a la ideal, sin embargo, dentro del intervalo de confianza del 95 % representado en la figura 5.6(b) se encuentran el 95.11 % de las diferencias (valor inferior al obtenido con la red neuronal de una capa con 10 neuronas y algoritmo BR).

Número de neuronas		Correlación	
Capa oculta 1	Capa oculta 2	BR	SCG
10	2	0.899502	0.207650
10	4	0.890943	0.703861
10	6	<b>0.899548*</b>	0.843431
10	8	0.899389	0.815811
20	2	0.704879	0.836079
20	4	0.893136	0.793904
20	6	0.893450	0.822111
20	8	0.892642	0.211970
30	2	0.882286	0.254304
30	4	0.880957	0.798889
30	6	0.880210	0.844208
30	8	0.879387	0.844541
40	2	0.814554	0.181264
40	4	0.877849	0.740674
40	6	0.775199	0.824916
40	8	0.806261	0.846973

TABLA 5.3: Correlación. Redes neuronales con 2 capas ocultas.

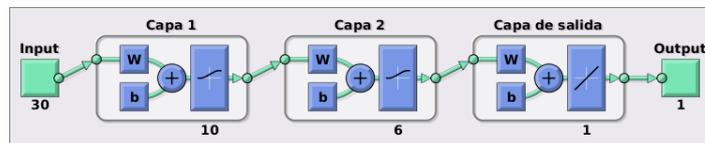


FIGURA 5.5: Diagrama de la red neuronal de dos capas ocultas con 10 neuronas en la primera capa y 6 en la segunda.

### 5.3.2 BOSQUES ALEATORIOS

En cuanto a bosques aleatorios, se entrenaron bosques con un barrido en el número de árboles, desde 10 árboles hasta 300, con incrementos de 10. En la tabla 5.4 se muestran los coeficientes de correlación obtenidos tras entrenar dichos bosques aleatorios.

El coeficiente de correlación más cercano a la unidad se obtuvo con el bosque aleatorio de 280 árboles, siendo éste 0.905360. En la figura 5.7(a) se puede observar el gráfico de dispersión junto con la recta de regresión, la cual posee una pendiente próxima a la unidad, 1.0472, es decir, de  $46.32^\circ$ . Además, en este gráfico de Bland-Altman, expuesto en la figura 5.7(b) se refleja el intervalo de confianza del 95 % de las diferencias, dentro del cual se encuentran el 95.28 % de estas diferencias.

Se puede decir entonces que este aproximador proporciona unos resultados razonables. Por lo tanto, se usará dicho bosque para aproximar los *counts* de MSB y, a partir de tales aproximaciones, calcular las medidas mencionadas en la sección 4.6.

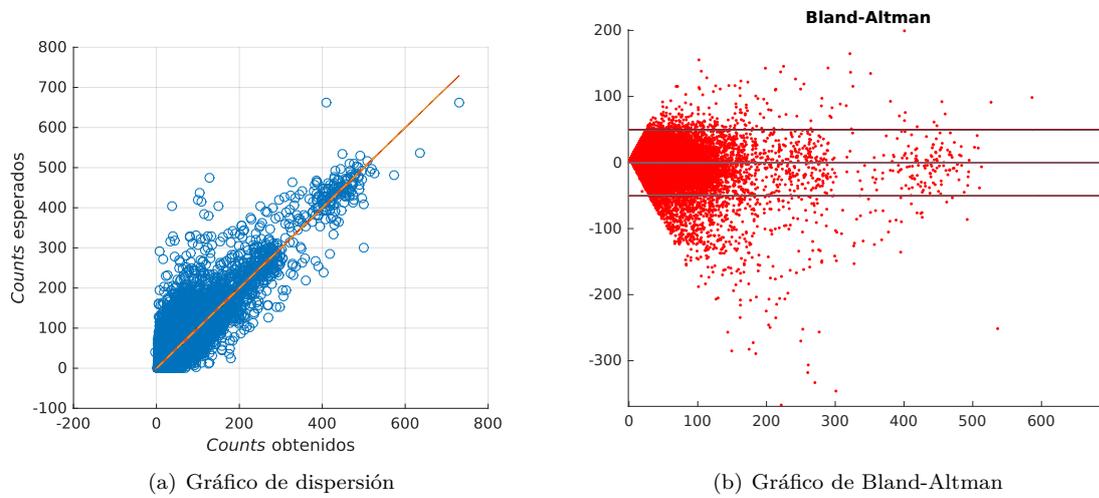


FIGURA 5.6: Gráficos de dispersión y de Bland-Altman de la red neuronal con dos capas ocultas con 10 neuronas en la primera capa y 6 en la segunda, y algoritmo de entrenamiento BR.

Número de árboles	Coefficiente de correlación	Número de árboles	Coefficiente de correlación
10	0.893031	160	0.904764
20	0.899341	170	0.904871
30	0.901706	180	0.904922
40	0.902370	190	0.904972
50	0.902888	200	0.905051
60	0.903280	210	0.905075
70	0.903629	220	0.905149
80	0.903960	230	0.905194
90	0.904013	240	0.905247
100	0.904164	250	0.905267
110	0.904317	260	0.905311
120	0.904429	270	0.905339
130	0.904536	280	<b>0.905360*</b>
140	0.904680	290	0.905358
150	0.904599	300	0.905349

TABLA 5.4: Correlación. Bosques aleatorios.

En la figura 5.8 puede observarse un ejemplo de aproximación de *counts* con el bosque aleatorio de 280 árboles entrenado. Los *counts* esperados, en azul, son aquellos proporcionados por ActiGraph, con sus datos de aceleración y tras el procesado propio. Los *counts* obtenidos, en naranja, son aquellos resultantes de aproximar los datos de MSB correspondientes con el bosque aleatorio finalmente elegido como aproximador.

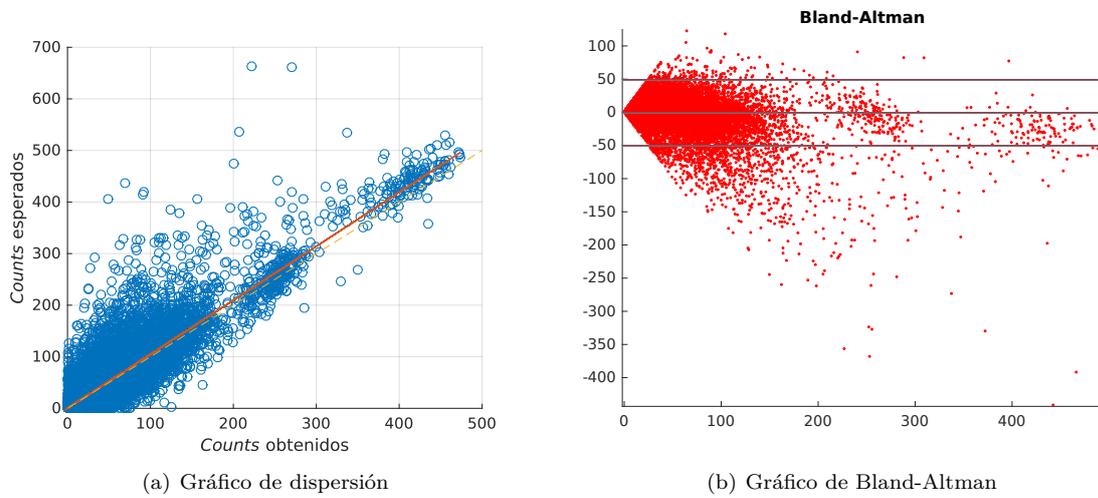


FIGURA 5.7: Justificación del uso del bosque aleatorio de 280 árboles como aproximador.

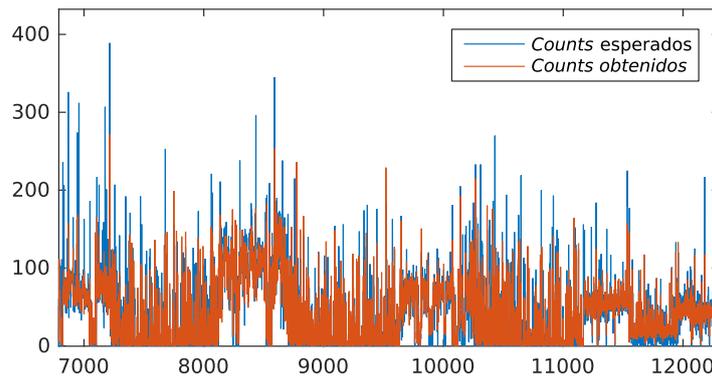


FIGURA 5.8: Aproximación obtenida con el bosque aleatorio de 280 árboles.

## 5.4 INTERVALOS DE NORMALIDAD

En esta sección, primero se abordarán los resultados de ACT obtenidos para las medidas tradicionales, después aquellos logrados para características no lineales y, por último, cómo los resultados de MSB encajan en tales resultados.

En la tabla 5.5 pueden observarse los resultados obtenidos para las medidas tradicionales media, desviación estándar ( $Dt$ ), mediana, coeficiente de asimetría ( $Sk$ ), curtosis ( $Kt$ ), rango intercuartil ( $interq$ ) y coeficiente de variación ( $CV$ ). Se exponen tres diezmados: 1 s, 1 min y 15 min, para cada uno de los cuales se encuentran representados los resultados de los cuatro canales ( $x$ ,  $y$ ,  $z$ ,  $r$ ) de la siguiente manera:  $media \pm desviación$  si los resultados siguen una distribución gaussiana o  $[mediana; rango intercuartil]$  en caso contrario. Se exponen los resultados obtenidos en el tramo de actividad, el cual será comparado con MSB.

Medida	Diezmado	Canal	Resultados	Medida	Diezmado	Canal	Resultados
Media	1 s	x	25.8332±25.4284	Dt	1 s	x	88.3647±124.6692
		y	25.9634±25.4166			y	87.5966±124.9823
		z	27.2277±26.0547			z	88.7540±124.8239
		r	60.6625±72.7785			r	147.5320±208.3608
	1 min	x	25.7806±25.3192		1 min	x	[24.8571;19.8489]
		y	25.9104±25.3066			y	[23.2927;18.3465]
		z	27.1734±25.9443			z	[24.9574;20.4288]
		r	60.5247±72.4367			r	[46.5643;32.7804]
	15 min	x	25.3984±24.5188		15 min	x	[16.0892;13.7699]
y		25.5208±24.5010	y	[14.9000;12.4853]			
z		26.7700±25.1272	z	[16.0903;14.0192]			
r		59.2418±69.7595	r	[31.7023;23.3760]			
Mediana	1 s	x	[0.0000;0.0000]	Sk	1 s	x	[4.5091;14.1046]
		y	[0.0000;0.0000]			y	[4.4431;13.5496]
		z	[0.0000;1.0000]			z	[4.4686;12.5892]
		r	[1.0000;16.6241]			r	[3.0089;8.8619]
	1 min	x	17.8812±27.0005		1 min	x	[1.2653;2.2636]
		y	18.9176±27.9953			y	[1.3469;2.2039]
		z	20.1319±28.5831			z	[1.1337;2.2165]
		r	43.6948±81.7561			r	[1.0947;2.5307]
	15 min	x	20.5231±25.3853		15 min	x	[0.7052;1.5108]
y		20.9803±25.2769	y	[0.6483;1.5356]			
z		22.3894±26.2767	z	[0.6085;1.3402]			
r		46.9156±70.8762	r	[0.6131;1.6396]			
Kt	1 s	x	[30.3042;441.1902]	Interq	1 s	x	[1.0000;39.0000]
		y	[30.3415;425.1567]			y	[1.6250;41.0000]
		z	[30.8378;393.7149]			z	[2.0000;44.0000]
		r	[15.2463;204.3128]			r	66.8021±51.7361
	1 min	x	21.9869±42.0487		1 min	x	29.8597±30.0355
		y	[5.3114;14.9004]			y	28.4217±29.0267
		z	22.8651±49.3992			z	30.7127±29.8808
		r	25.5219±53.4542			r	63.8188±83.7186
	15 min	x	[3.0280;4.5515]		15 min	x	22.6693±28.9075
y		[3.0597;4.4127]	y	21.4768±29.0432			
z		[2.8985;3.8560]	z	23.0200±29.1544			
r		[2.9246;4.8436]	r	50.3005±85.1994			
CV	1 s	x	[2.7427;2.3787]				
		y	[2.7048;2.5051]				
		z	[2.7065;2.5557]				
		r	[1.8846;1.4395]				
	1 min	x	[1.0127;0.4639]				
		y	[0.9876;0.4156]				
		z	[0.9653;0.4165]				
		r	[0.9520;0.4650]				
	15 min	x	[0.7046;0.3232]				
y		[0.6795;0.3271]					
z		[0.6672;0.3263]					
r		[0.6675;0.3448]					

TABLA 5.5: Medidas tradicionales en el tramo de actividad.

En la tabla 5.6 se muestran algunas de las características no lineales obtenidas a partir del análisis de cada intervalo de los registros. Se han elegido estas características tomando como referencia la Tabla 3 de [6]. Los valores se representan como  $\text{media} \pm \text{desviación}$  si existe gaussianidad en los resultados obtenidos o como [mediana;rango intercuartil] en caso contrario. Las características se indican como  $P_{Palabra}^{Canal}$  para DS y  $CTM_{\rho}^{Canal}$  para CTM, y los intervalos se denominan de la siguiente manera: D: día completo, A: actividad o vigilia, AM: actividad durante la mañana, AT: actividad durante la tarde, Desc: descanso, ID: inicio del descanso, MD: mitad del descanso, FD: final del descanso.

Característica	(Época, Intervalo)	Resultado	Característica	(Época, Intervalo)	Resultado
$P_{00}^x$	(15 min, D)	[0.2421;0.0526]	$CTM_{\rho=0}^y$	(1 min, A)	[0.0030;0.0315]
$P_{10}^x$	(15 min, D)	[0.0737;0.0421]	$CTM_{\rho=0}^r$	(1 min, A)	[0.0075;0.0479]
$P_{00}^z$	(15 min, D)	[0.2421;0.0526]	$P_{21}^r$	(30 s, A)	[0.0834;0.0191]
$P_{00}^r$	(15 min, D)	[0.2421;0.0632]	$CTM_{\rho=0}^r$	(30 s, A)	[0.0186;0.0727]
$P_{01}^x$	(15 min, D)	[0.0632;0.0421]	$CTM_{\rho=0,1}^r$	(30 s, A)	0.0881 $\pm$ 0.0847
$CTM_{\rho=2,7}^x$	(1 min, D)	[0.3526;0.1544]	$P_{12}^r$	(30 s, A)	[0.0826;0.0192]
$P_{12}^x$	(15 s, AM)	[0.0948;0.0253]	$P_{00}^x$	(15 min, AT)	[0.1860;0.0706]
$P_{12}^x$	(30 s, AM)	[0.0894;0.0251]	$CTM_{\rho=4,4}^r$	(5 min, AT)	[0.0862;0.1316]
$P_{22}^x$	(15 s, AM)	[0.2185;0.0360]	$CTM_{\rho=3,5}^r$	(5 min, AT)	[0.0667;0.1107]
$P_{20}^x$	(30 s, AM)	0.0215 $\pm$ 0.0221	$CTM_{\rho=3,6}^r$	(5 min, AT)	[0.0702;0.1137]
$P_{21}^x$	(15 s, AM)	[0.0946;0.0251]	$CTM_{\rho=4,3}^r$	(5 min, AT)	[0.0846;0.1317]
$P_{22}^x$	(30 s, AM)	[0.2235;0.0328]	$CTM_{\rho=4,2}^r$	(5 min, AT)	[0.0814;0.1252]
$P_{22}^r$	(5 min, Desc)	[0.1802;0.0800]	$P_{22}^r$	(1 min, ID)	[0.1836;0.1399]
$CTM_{\rho=4,9}^x$	(5 min, Desc)	[0.7906;0.3197]	$P_{22}^y$	(1 min, ID)	[0.1750;0.1500]
$CTM_{\rho=1,5}^x$	(15 min, Desc)	[0.5000;0.4992]	$P_{22}^z$	(1 min, ID)	[0.1750;0.1500]
$CTM_{\rho=5,3}^x$	(5 min, Desc)	[0.8028;0.3032]	$P_{22}^x$	(1 min, ID)	[0.1749;0.1536]
$CTM_{\rho=6,8}^x$	(5 min, Desc)	[0.8439;0.2565]	$P_{00}^y$	(30 s, ID)	[0.1750;0.7139]
$P_{22}^z$	(5 min, Desc)	[0.1732;0.0797]	$E_{Shannon}^y$	(15 s, ID)	[0.9054;1.2947]
$P_{20}^r$	(15 s, MD)	[0.0248;0.0319]	$CTM_{\rho=0,3}^z$	(5 min, FD)	[0.2162;0.4355]
$P_{02}^r$	(15 s, MD)	[0.0248;0.0311]	$P_{21}^r$	(5 min, FD)	[0.1250;0.1225]
$P_{02}^z$	(15 s, MD)	[0.0241;0.0269]	$CTM_{\rho=0,2}^r$	(5 min, FD)	[0.1667;0.3636]
$P_{20}^z$	(15 s, MD)	[0.0247;0.0258]	$CTM_{\rho=0,3}^r$	(5 min, FD)	[0.1716;0.3898]
$CTM_{\rho=0,4}^z$	(15 s, MD)	[0.8465;0.6400]	$CTM_{\rho=0,4}^z$	(5 min, FD)	[0.2500;0.4232]
$P_{20}^y$	(15 s, MD)	[0.0248;0.0271]	$CTM_{\rho=0,1}^r$	(5 min, FD)	[0.1579;0.3333]

TABLA 5.6: Características no lineales

Para presentar una comparativa de los resultados de MSB respecto a los hallados con ACT, se generarán unos *boxplots* con las distribuciones que siguen cada una de las características de ACT dentro del periodo de actividad, a los que se superpondrán los resultados obtenidos para los distintos voluntarios cuyos datos fueron registrados con MSB. Únicamente se expone el periodo de actividad dadas las condiciones de registro de las señales de MSB (actividad física espontánea y andar). A continuación, se muestran los *boxplots* correspondientes a dicho intervalo para el canal  $r$ . El resto de canales se encuentran representados en el Apéndice C.

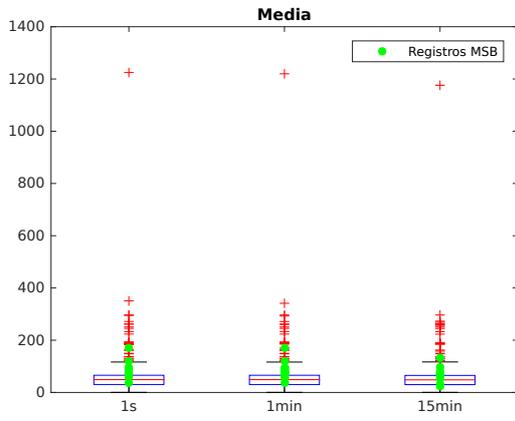


FIGURA 5.9: Media. Canal r. Actividad.

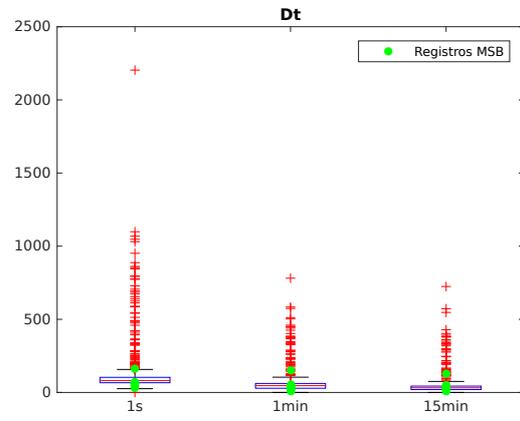


FIGURA 5.10: Desviación estándar. Canal r. Actividad.

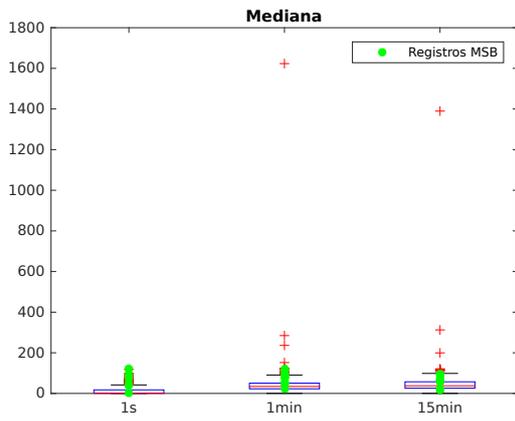


FIGURA 5.11: Mediana. Canal r. Actividad.

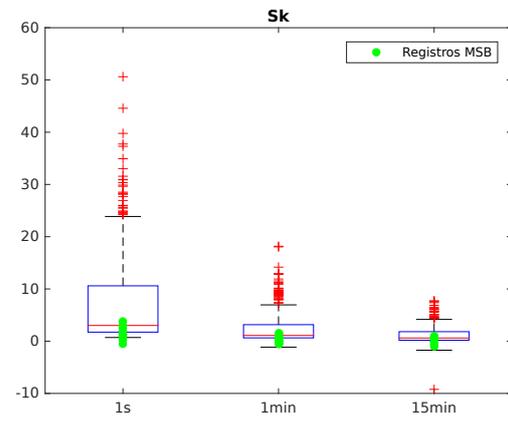


FIGURA 5.12: Asimetría. Canal r. Actividad.

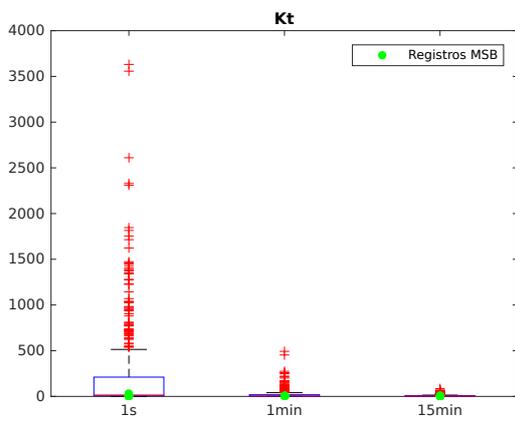


FIGURA 5.13: Curtosis. Canal r. Actividad.

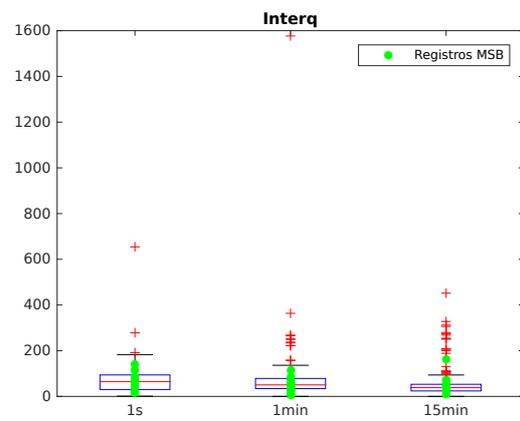


FIGURA 5.14: Rango intercuartil. Canal r. Actividad.

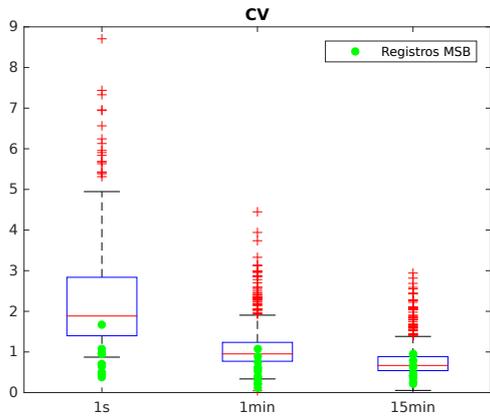


FIGURA 5.15: Coeficiente de variación. Canal r. Actividad.

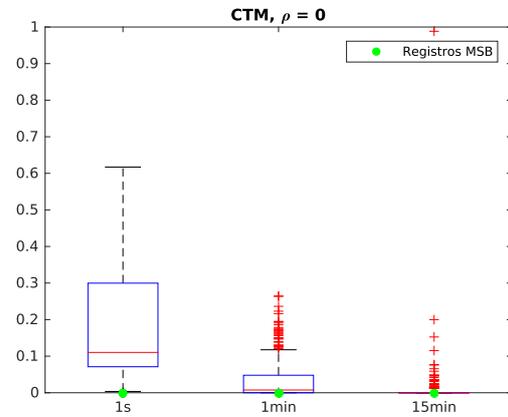


FIGURA 5.16:  $CTM_{\rho=0}^r$ . Actividad.

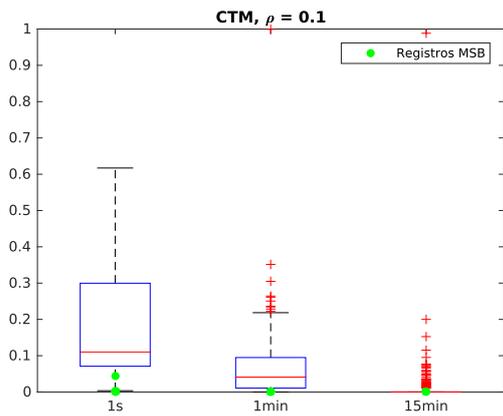


FIGURA 5.17:  $CTM_{\rho=0.1}^r$ . Actividad.

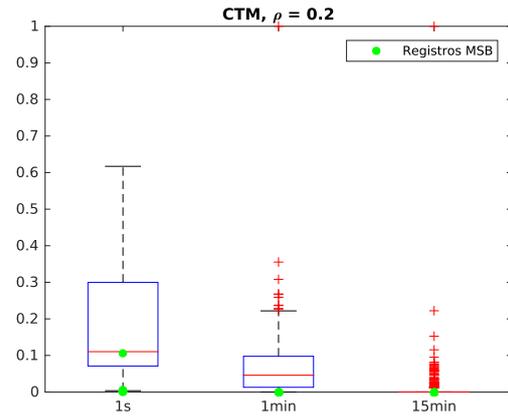


FIGURA 5.18:  $CTM_{\rho=0.2}^r$ . Actividad.

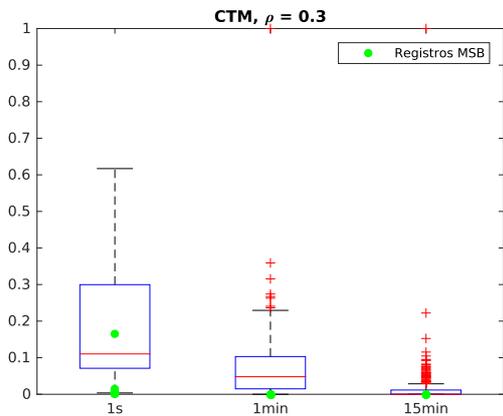


FIGURA 5.19:  $CTM_{\rho=0.3}^r$ . Actividad.

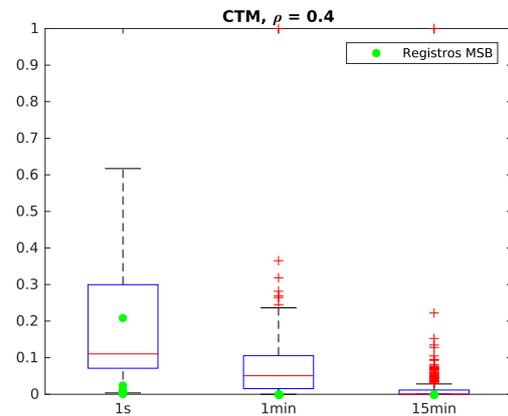


FIGURA 5.20:  $CTM_{\rho=0.4}^r$ . Actividad.

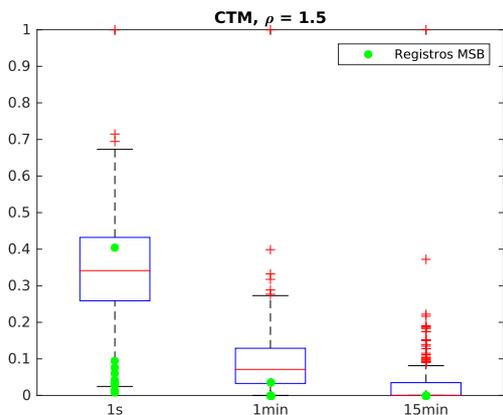


FIGURA 5.21:  $CTM_{\rho=1,5}^r$ . Actividad.

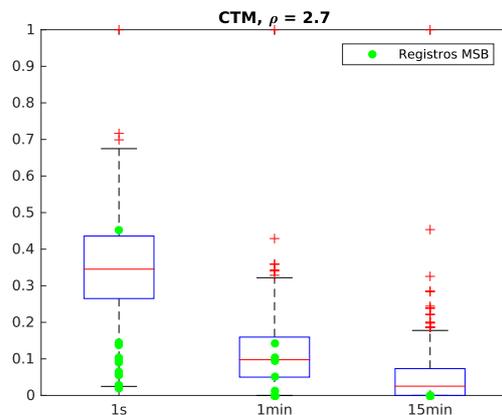


FIGURA 5.22:  $CTM_{\rho=2,7}^r$ . Actividad.

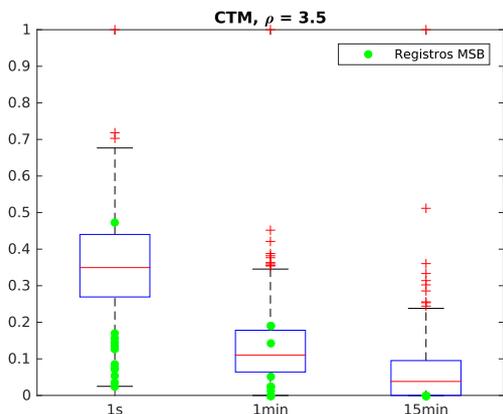


FIGURA 5.23:  $CTM_{\rho=3,5}^r$ . Actividad.

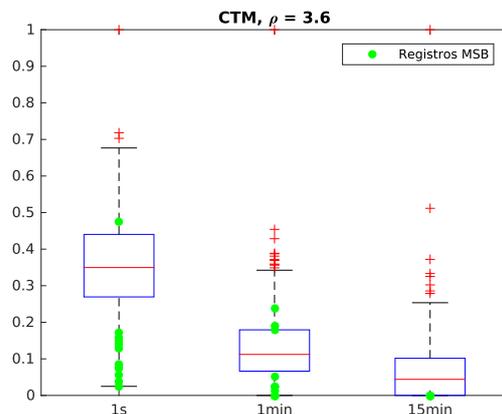


FIGURA 5.24:  $CTM_{\rho=3,6}^r$ . Actividad.

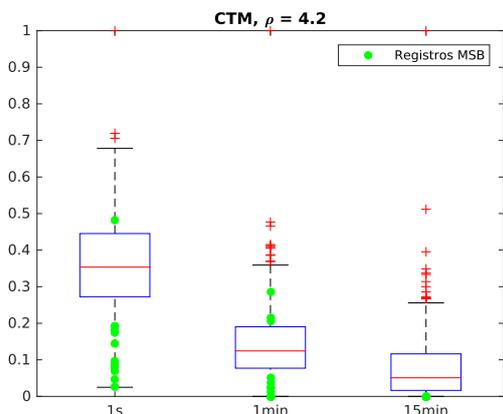


FIGURA 5.25:  $CTM_{\rho=4,2}^r$ . Actividad.

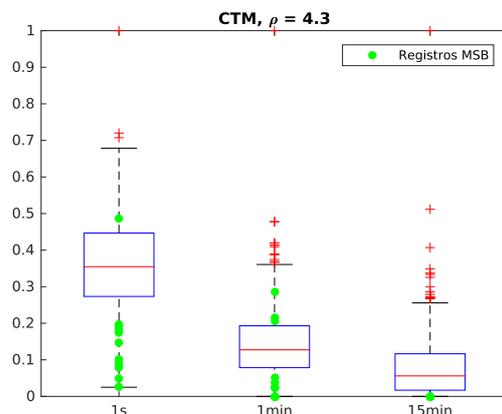


FIGURA 5.26:  $CTM_{\rho=4,3}^r$ . Actividad.

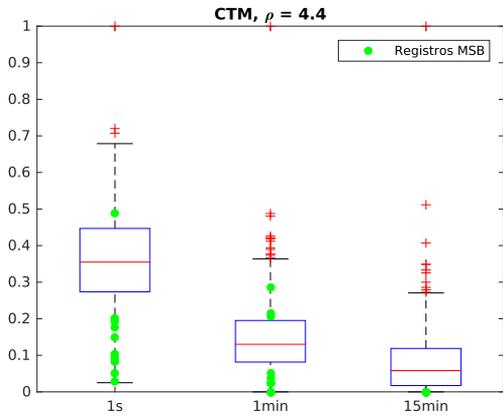


FIGURA 5.27:  $CTM_{\rho=4,4}^r$ . Actividad.

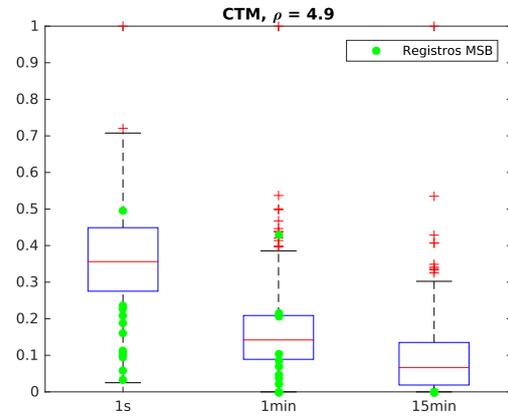


FIGURA 5.28:  $CTM_{\rho=4,9}^r$ . Actividad.

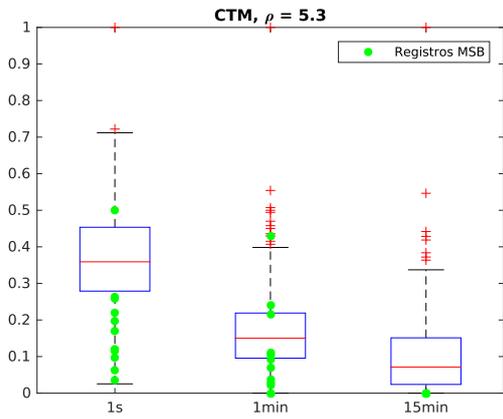


FIGURA 5.29:  $CTM_{\rho=5,3}^r$ . Actividad.

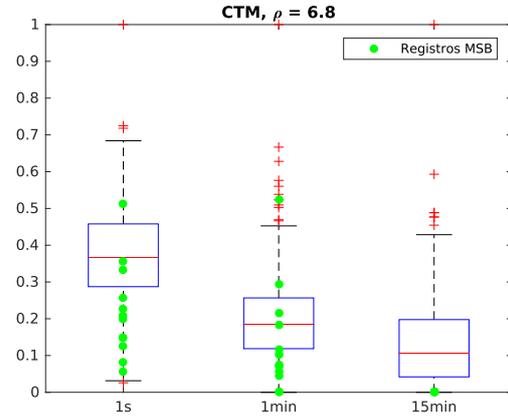


FIGURA 5.30:  $CTM_{\rho=6,8}^r$ . Actividad.

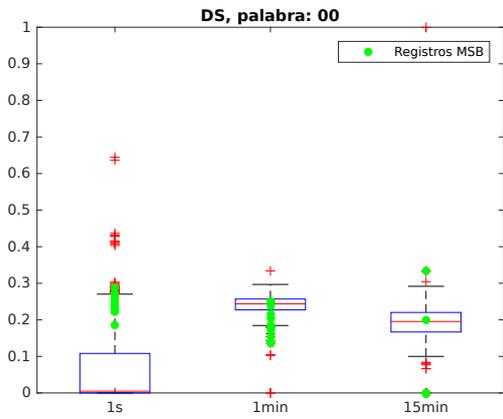


FIGURA 5.31:  $P_{00}^r$ . Actividad.

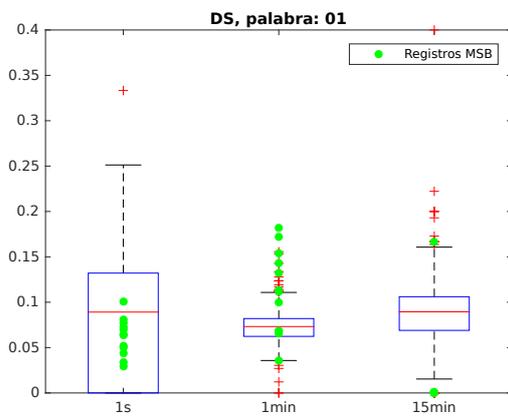


FIGURA 5.32:  $P_{01}^r$ . Actividad.

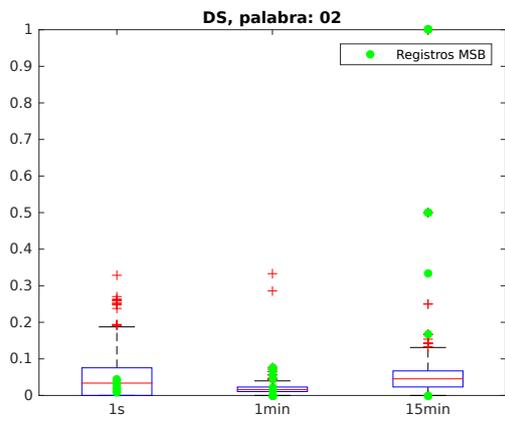


FIGURA 5.33:  $P_{02}^r$ . Actividad.

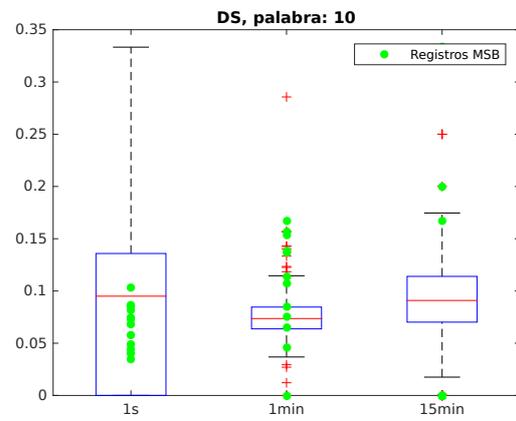


FIGURA 5.34:  $P_{10}^r$ . Actividad.

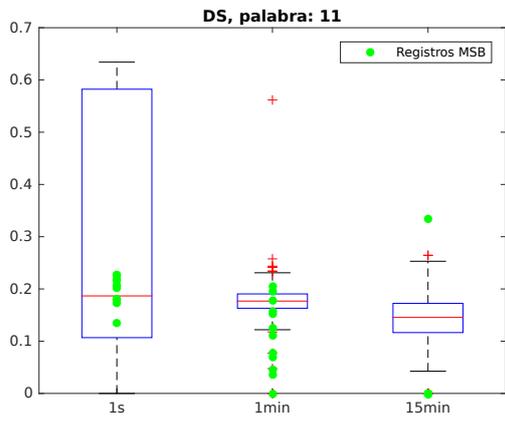


FIGURA 5.35:  $P_{11}^r$ . Actividad.

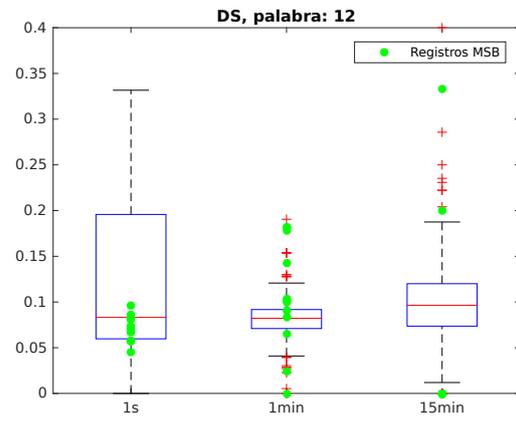


FIGURA 5.36:  $P_{12}^r$ . Actividad.

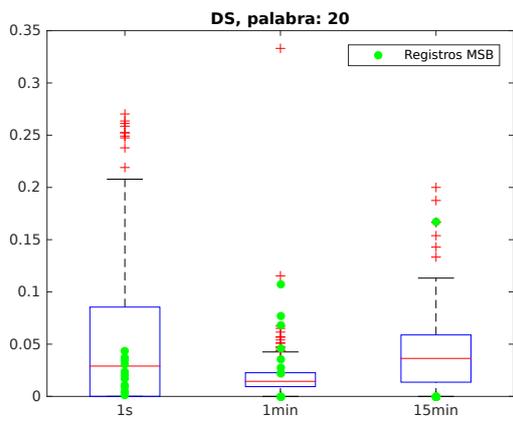


FIGURA 5.37:  $P_{20}^r$ . Actividad.

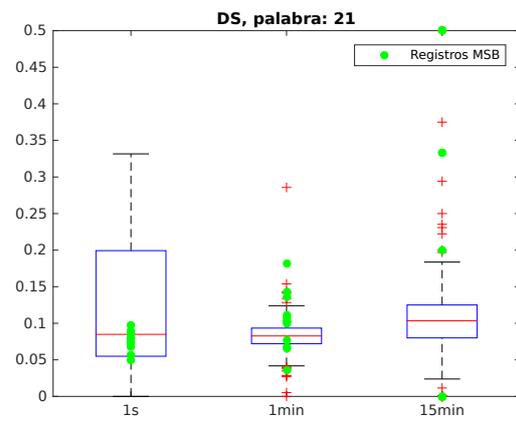
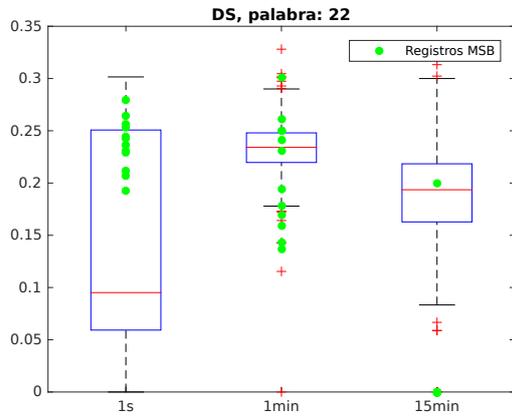
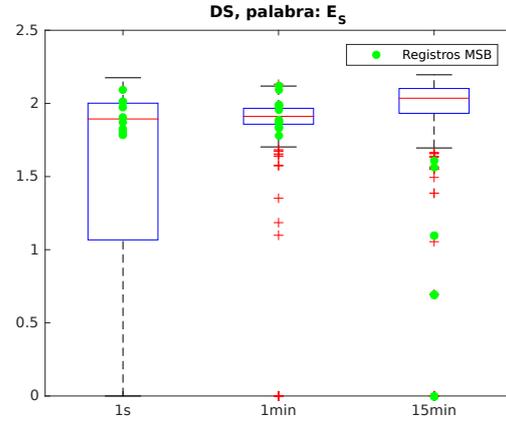


FIGURA 5.38:  $P_{21}^r$ . Actividad.

FIGURA 5.39:  $P_{22}^r$ . Actividad.FIGURA 5.40:  $E_{Shannon}^r$ . Actividad.

En cuanto a las medidas media, mediana, desviación estándar, coeficiente de asimetría, curtosis y rango intercuartil, se puede decir que los datos de MSB se ajustan a los límites impuestos por los bigotes del *boxplot*. Sin embargo, el coeficiente de variación dista de la distribución correspondiente en el diezmado de 1 s, no así en el de 1 min y 15 min.

Por otro lado, en lo que respecta a la CTM puede observarse que a medida que  $\rho$  aumenta, se incrementan el número de registros de MSB que se mantienen dentro de los límites. Los diezmados de 1 s y 1 min se ajustan mejor a la distribución descrita por ACT que el diezmado de 15 min.

Por último, en la dinámica simbólica encontramos unos resultados similares a los obtenidos con la CTM, es decir, los diezmados de 1 s y 1 min se ajustan mejor a la distribución descrita por ACT que el diezmado de 15 min.



## CONCLUSIONES Y LÍNEAS FUTURAS

---

### 6.1 CONCLUSIONES

---

Al comienzo de este documento se expuso el principal objetivo de este trabajo, siendo éste la aproximación de *counts* extraídos del modelo de ACT, poseedor de los requisitos esenciales para poder ser utilizado en el diagnóstico clínico, a partir de datos de aceleración de otros actímetros que no cumplen dichos requisitos. De esta forma los intervalos de normalidad obtenidos en base al primer modelo permitan comparar los resultados obtenidos a partir de estos dispositivos comerciales.

Con el fin de llevarlo a cabo, se desarrolló una aplicación móvil que permite el registro del RD de MSB. Este RD tuvo que ser preprocesado, consistiendo este preprocesado en un remuestreo y alineación, de forma que las señales de MSB fueran adecuadas para el posterior procesado. Una vez acondicionadas estas señales, se hizo uso de aprendizaje automático para aproximar la función de paso de RD a *counts*. Tanto redes neuronales como bosques aleatorios presentaron buenos resultados, las primeras a costa de un mayor tiempo de entrenamiento y los segundos a costa de un mayor tamaño en memoria. Finalmente, se optó por los bosques aleatorios dada su mejor concordancia entre *counts* conocidos y *counts* estimados, es decir, la diferencia entre pares se ajustaba mejor al intervalo de confianza.

Una vez lograda una aproximación razonable de tales *counts* era necesario saber cómo de comparables son los datos obtenidos a partir de ambos actímetros. Para ello, se extrajeron diferentes características de los *counts* de ACT, tanto tradicionales como de cuantificación de la complejidad, para los diferentes canales, diferentes tramos de actividad/descanso y varios diezmados. A partir de éstas fueron calculados los intervalos de normalidad. Las mismas características fueron extraídas de los datos de MSB, sin embargo, estos datos no fueron segmentados en tramos.

Teniendo en cuenta la brevedad de los registros de MSB y las condiciones en las que fueron llevados a cabo, lo cual condiciona la comparativa al tramo de sólo actividad, los resultados obtenidos para MSB pueden considerarse razonablemente comparables a los obtenidos con datos de ACT, aun existiendo diferencias significativas entre las edades de ambos grupos.

Por lo tanto, al término de este Trabajo Fin de Máster, podemos decir que, considerando válidos los *counts* de ACT, es posible aproximar la función que proporciona *counts* a partir de RD de aceleración de MSB, sugiriendo que es posible aproximar tal función para cualquier actímetro disponible en el mercado (aunque no cuente con validez en el diagnóstico clínico). Este hecho favorece el uso de estos actímetros en estudios de la materia, dado que estos dispositivos se encuentran disponibles en el mercado al alcance del público general, no únicamente limitado a médicos e investigadores.

---

## 6.2 LIMITACIONES Y LÍNEAS FUTURAS

---

Durante el desarrollo de este Trabajo Fin de Máster se han encontrado diversas limitaciones, la mayoría de ellas relacionadas con el actímetro MSB, lo cual puede ser solventado en trabajos futuros.

En primer lugar, la base de datos de registros de MSB cuenta con pocas entradas, y todos los registros fueron llevados a cabo con el dispositivo en la mano no dominante del sujeto. Por otro lado, la brevedad de los registros condiciona el análisis de los resultados al tramo de sólo actividad, además de impedir el correcto estudio de los resultados para diezmados grandes, como puede ser el de 15 min, dado que el número de muestras resultantes es muy pequeño.

En lo que respecta a la aplicación móvil desarrollada en el marco de este trabajo, cabe destacar que ésta se encuentra en una primera versión, y que podría ser mejorada incluyendo, por ejemplo, una cabecera en el fichero que contenga la información del registro (fecha y hora de inicio y fin, edad y sexo del sujeto, muñeca en la que portará el dispositivo, ...). Además, podrían programarse funciones que den aviso al usuario en caso de pérdida de la conexión con la pulsera o que le quede poca batería.

## BIBLIOGRAFÍA

---

- [1] Paolo Bonato. Wearable sensors/systems and their impact on biomedical engineering. *IEEE Engineering in Medicine and Biology Magazine*, 22(3):18–20, 2003.
- [2] Diego Martín Martínez. *Contribuciones en Machine Learning y Modelado Estocástico Orientadas al Análisis de Señales Biomédicas*. PhD thesis, Universidad de Valladolid, 2015.
- [3] Francesca Iniesta Tortosa, Miguel García, Josep Colominas Balagué, and Xavier Corbera. Sensor de aceleración para realizar estudios de actimetría. *Buran*, (6):56–59, 1995.
- [4] Penny Corkum, Rosemary Tannock, Harvey Moldofsky, Sheilah Hogg-Johnson, and Tom Humphries. Actigraphy and parental ratings of sleep in children with attention-deficit/hyperactivity disorder (adhd). *Sleep*, 24(3):303–312, 2001.
- [5] Diego Martín, Pablo Casaseca, Susana Alberola, José Antonio López, Francisco Carlos Ruiz, Jesús María Andrés, José Ramón Garmendia, and Julio Ardura. Automatic diagnosis of adhd based on nonlinear analysis of actimetry registries. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 685–688. IEEE, 2011.
- [6] D. Martín Martínez, S. Casaseca de la Higuera, P. Alberola López, J. Andrés de Llano, J. A. López Villalobos, J. Ardura Fernández, and C. Alberola López. Nonlinear analysis of actigraphic signals for the assessment of the attention-deficit/hyperactivity disorder (ADHD). *Medical Engineering and Physics*, 34(9):1317–1329, 2012.
- [7] Jessica Waldon, Esmot Begum, Melissa Gendron, Benjamin Rusak, Pantelis Andreou, Malgorzata Rajda, and Penny Corkum. Concordance of actigraphy with polysomnography in children with and without attention-deficit/hyperactivity disorder. *Journal of sleep research*, 25(5):524–533, 2016.
- [8] Eus JW Van Someren. Actigraphic monitoring of movement and rest-activity rhythms in aging, alzheimer’s disease, and parkinson’s disease. *IEEE Transactions on Rehabilitation Engineering*, 5(4):394–398, 1997.
- [9] K Stavitsky, JL Saurman, P McNamara, and A Cronin-Golomb. Sleep in parkinson’s disease: a comparison of actigraphy and subjective measures. *Parkinsonism & related disorders*, 16(4):280–283, 2010.
- [10] Weidong Pan, Shin Kwak, Fuzhong Li, Chunlan Wu, Yiyun Chen, Yoshiharu Yamamoto, and Dingfang Cai. Actigraphy monitoring of symptoms in patients with parkinson’s disease. *Physiology & behavior*, 119:156–160, 2013.
- [11] Franco De Crescenzo, Alexis Economou, Ann L Sharpley, Aynur Gormez, and Digby J Queded. Actigraphic features of bipolar disorder: A systematic review and meta-analysis. *Sleep Medicine Reviews*, 33:58–69, 2017.

- [12] Kaitlin Hanley White, Meredith E Rumble, and Ruth M Benca. Sex differences in the relationship between depressive symptoms and actigraphic assessments of sleep and rest-activity rhythms in a population-based sample. *Psychosomatic medicine*, 79(4):479–484, 2017.
- [13] The MathWorks - MATLAB and Simulink for Technical Computing. Website. Último acceso mayo 2017. <http://www.mathworks.com/>.
- [14] J.J. Van Hilten, H.A.M. Middelkoop, S.I.R. Kuiper, C.G.S. Kramer, and R.A.C. Roos. Where to record motor activity: an evaluation of commonly used sites of placement for activity monitors. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 89(5):359–362, 1993.
- [15] J Sasaki, K da Silva, B da Costa, and D John. *Measurement of Physical Activity Using Accelerometers*. Elsevier Inc., 2016.
- [16] Kong Y. Chen and David R. Bassett. The technology of accelerometry-based activity monitors: Current and future. *Medicine and Science in Sports and Exercise*, 37(Supplement):S490–S500, 2005.
- [17] Patty S. Freedson, Edward Melanson, and John Sirard. Calibration of the computer science and applications, inc. accelerometer. *Medicine Science in Sports Exercise*, 30(5):777–781, 1998.
- [18] Scott J. Strath, Karin A. Pfeiffer, and Melicia C. Whitt-Glover. Accelerometer use with children, older adults, and adults with functional limitations. *Medicine and Science in Sports and Exercise*, 44(SUPPL. 1):77–85, 2012.
- [19] Stewart G Trost, Paul D Loprinzi, Rebecca Moore, and Karin A Pfeiffer. Comparison of accelerometer cut points for predicting activity intensity in youth. *Medicine & Science in Sports & Exercise*, 43(7):1360–1368, 2011.
- [20] Charles E Matthew. Calibration of accelerometer output for adults. *Medicine and science in sports and exercise*, 37(11 Suppl):S512–22, 2005.
- [21] Jeffer E Sasaki, Dinesh John, and Patty S Freedson. Validation and comparison of actigraph activity monitors. *Journal of Science and Medicine in Sport*, 14(5):411–416, 2011.
- [22] ActiGraph White Paper : What is a Count?. Último acceso mayo 2017. [http://actigraphcorp.com/wp-content/uploads/2015/06/ActiGraph-White-Paper\\_What-is-a-Count\\_.pdf](http://actigraphcorp.com/wp-content/uploads/2015/06/ActiGraph-White-Paper_What-is-a-Count_.pdf).
- [23] Warren W Tryon and Robert Williams. Fully proportional actigraphy: a new instrument. *Behavior Research Methods*, 28(3):392–403, 1996.
- [24] David R Bassett, Ann V Rowlands, and Stewart G Trost. Calibration and validation of wearable monitors. *Medicine and science in sports and exercise*, 44(1 Suppl 1):S32, 2012.
- [25] Ann V. Rowlands. Accelerometer assessment of physical activity in children: An update. *Pediatric Exercise Science*, 19(3):252–266, 2007.
- [26] Jennifer L Copeland and Dale W Esliger. Accelerometer assessment of physical activity in active, healthy older adults. *Journal of aging and physical activity*, 17(1):17–30, 2009.
- [27] Patty Freedson, David Pober, and Kathleen F. Janz. Calibration of accelerometer output for children. *Medicine and Science in Sports and Exercise*, 37(Supplement):S523–S530, 2005.
- [28] D Hendelman, K Miller, C Baggett, E Debold, and P Freedson. Validity of accelerometry for the assessment of moderate intensity physical activity in the field. *Medicine and science in sports and exercise*, 32(9 Suppl):S442–9, 2000.

- [29] Nora E. Miller, Scott J. Strath, Ann M. Swartz, and Susan E. Cashin. Estimating absolute and relative physical activity intensity across age via accelerometry in adults. *Journal of Aging and Physical Activity*, 18(2):158–170, 2010.
- [30] Ann M Swartz, Scott J Strath, DAVID R Bassett Jr, WILLIAM L O’Brien, George A King, and BARBARA E Ainsworth. Estimation of energy expenditure using csa accelerometers at hip and wrist sites. *Medicine and science in sports and exercise*, 32(9 Suppl):S450–6, 2000.
- [31] Choose a Multilayer Neural Network Training Function - Mathworks. Último acceso junio 2017. <http://actigraphcorp.com/products-showcase/activity-monitors/actigraph-wgt3x-bt/>.
- [32] wGT3X-BT Monitor - ActiGraph. Website. Último acceso mayo 2017. <http://actigraphcorp.com/products/wgt3x-bt-monitor/>.
- [33] ActiLife 6 Data Analysis Software - ActiGraph. Último acceso junio 2017. <https://www.actigraph.nl/en/product/11/actilife-6.html>.
- [34] Microsoft Band 2 product regulatory, warranty, and safety summary - Microsoft Band Documents. Último acceso mayo 2017. <https://support.microsoft.com/en-us/help/4000309/band-documents>.
- [35] Microsoft Band - Tech specs. Último acceso junio 2017. <https://www.microsoft.com/microsoft-band/en-us/featurespecs-hero>.
- [36] CNET. Google buys Android. Website. Último acceso mayo 2017. <https://www.cnet.com/news/google-buys-android/>.
- [37] Net Market Share. Mobile/Tablet Operating System Market Share. Website. Último acceso mayo 2017. <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8qpcustomd=1>.
- [38] Android Developers - Aspectos fundamentales de la aplicación. Website. Último acceso mayo 2017. <https://developer.android.com/guide/components/fundamentals.html>.
- [39] Microsoft Health - For Developers. Website. Último acceso mayo 2017. <https://developer.microsoftband.com/>.
- [40] Microsoft. *Microsoft Band SDK*, 2015. <https://developer.microsoftband.com/Content/docs/MicrosoftBandSDK.pdf>.
- [41] Code Project. Android: Ready to use simple directory chooser dialog with new folders creation. Website. Último acceso mayo 2017. <http://www.codeproject.com/ARTICLES/547636/Android-Ready-to-use-simple-directory-chooser-dial>.
- [42] D. Martín Martínez, P. Casaseca de la Higuera, J. Andrés de Llano, J.R. Garmendia Leiza, S. Alberola López, and C. Alberola-López. Automatic detection of wakefulness and rest intervals in actigraphic signals: A data-driven approach. *Medical engineering & physics*, 36(12):1585–1592, 2014.
- [43] ActiGraph Data Conversion Process - ActiGraph Support Center. Último acceso mayo 2017. <https://actigraph.desk.com/customer/en/portal/ARTICLES/2515508-actigraph-data-conversion-process>.
- [44] D. Peach, J. Van Hoomissen, and H.L. Callender. Exploring the actilife® filtration algorithm: Converting raw acceleration data to counts. *Physiological measurement*, 35(12):2359, 2014.
- [45] mHealth-GT3X Converter - GitHub. Último acceso mayo 2017. <https://github.com/SPADES-PUBLIC/mHealth-GT3X-converter-public>.
- [46] ActiGraph Software Department. *ActiLife 6 User’s Manual*, 2012.

- [47] The Mathworks. What is Machine Learning. page 12, 2016.
- [48] Pedro Ponce Cruz. *Inteligencia artificial con aplicaciones a la ingeniería*. Marcombo, 1 edition, 2011.
- [49] M.Y. Rafiq, G. Bugmann, and D.J. Easterbrook. Neural network design for engineering applications. *Computers & Structures*, 79(17):1541–1552, 2001.
- [50] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [51] Choose a Multilayer Neural Network Training Function - Mathworks. Último acceso junio 2017. <https://es.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html>.
- [52] Bosques aleatorios (Random Forests). Último acceso mayo 2017. <http://bis.ifc.unam.mx/es/ensenanza/metodos-computacionales-para-clasificacion/presentacion-random-forests>.
- [53] Leo Breiman. Random forests. *Machine Learning*, 2001.
- [54] J Lluís Carrasco and Lluís Jover. Métodos estadísticos para evaluar la concordancia. *Medicina Clínica*, 122(Supl. 1):28–34, 2004.

## Apéndice A

# CÓDIGO FUENTE DE LA APLICACIÓN

---

```
AndroidManifest.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="org.elena.tfm.registrodeaceleracionmsband"
4      android:versionCode="2"
5      android:versionName="1.1" >
6      <uses-sdk
7          android:minSdkVersion="17"
8          android:targetSdkVersion="22" />
9      <uses-permission android:name="android.permission.BLUETOOTH" />
10     <uses-permission android:name="com.microsoft.band.service.access.BIND_BAND_SERVICE" />
11     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
12     <application
13         android:allowBackup="true"
14         android:icon="@drawable/ic_heart"
15         android:label="@string/app_name"
16         android:resizeableActivity="false"
17         android:theme="@style/Theme.Estiloapp" >
18         <activity
19             android:name=".Primera_Actividad"
20             android:label="@string/app_name"
21             android:launchMode="singleTask"
22             android:screenOrientation="portrait" >
23             <intent-filter>
24                 <action android:name="android.intent.action.MAIN" />
25                 <category android:name="android.intent.category.LAUNCHER" />
26             </intent-filter>
27         </activity>
28         <activity
29             android:name=".Segunda_Actividad"
30             android:screenOrientation="portrait" >
31         </activity>
32         <service android:name=".ServicioBackground" />
33         <receiver android:name=".Receptor" />
34         <receiver android:name=".ReceptorCancelar" />
35     </application>
36 </manifest>
```

```

1 package org.elena.tfm.registrodeaceleracionmsband;
2 import java.io.File; // ...
3
4 public class Primera_Actividad extends Activity {
5
6     private Button botonDirectorio;
7     private Button botonSiguiente;
8     private EditText nombreFichero;
9     private String ficheroTXT;
10    private String m_chosenDir = "";
11    private boolean m_newFolderEnabled = true;
12    private TextViewEx instruccion_directorio;
13    private TextViewEx rutaDirectorio;
14
15    @Override
16    protected void onCreate(Bundle savedInstanceState) {
17        super.onCreate(savedInstanceState);
18        setContentView(R.layout.activity_primera_actividad);
19
20        SpannableString s = new SpannableString("Registro de Aceleracion MsB");
21        s.setSpan(new TypefaceSpan(getApplicationContext(), "Cubano-Regular.otf"), 0, s.length(),
22        ↪ Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
23        s.setSpan(new ForegroundColorSpan(Color.WHITE), 0, s.length(),
24        ↪ Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
25        ActionBar actionBar = getActionBar();
26        actionBar.setTitle(s);
27        botonDirectorio = (Button) findViewById(R.id.botonDirectorio);
28        botonSiguiente = (Button) findViewById(R.id.botonSiguiente);
29        nombreFichero = (EditText) findViewById(R.id.nombreFichero);
30        rutaDirectorio = (TextViewEx) findViewById(R.id.rutaDirectorio);
31        Typeface fuenteBotones = Typeface.createFromAsset(getAssets(), "Fuentes/Cubano-Regular.otf");
32        botonDirectorio.setTypeface(fuenteBotones);
33        botonSiguiente.setTypeface(fuenteBotones);
34        instruccion_directorio = (TextViewEx) findViewById(R.id.instruccion_directorio);
35        instruccion_directorio.setText(getResources().getString(R.string.instruccion_directorio), true);
36
37        botonDirectorio.setOnClickListener(new OnClickListener(){
38            @Override
39            public void onClick(View v) {
40                DirectoryChooserDialog directoryChooserDialog =
41                new DirectoryChooserDialog(Primera_Actividad.this,
42                new DirectoryChooserDialog.ChosenDirectoryListener() {
43                    @Override
44                    public void onChosenDir(String chosenDir) {
45                        File vacio = new File(chosenDir, "vacio.txt");
46                        try {
47                            vacio.createNewFile();
48                            vacio.delete();
49                            m_chosenDir = chosenDir;
50                            rutaDirectorio.setText("Directorio elegido: " + chosenDir);
51                            rutaDirectorio.setTextColor(Color.parseColor("#00B116"));
52                        } catch (IOException e) {
53                            m_chosenDir = "";
54                            rutaDirectorio.setText("El directorio " + chosenDir + " no está permitido. Escoja otra
55                            ↪ ruta para guardar los datos.", true);

```

```
53         rutaDirectorio.setTextColor(Color.RED);
54     }
55 }
56 });
57 directoryChooserDialog.setNewFolderEnabled(m_newFolderEnabled);
58 directoryChooserDialog.chooseDirectory(m_chosenDir);
59 m_newFolderEnabled = !m_newFolderEnabled;
60 }
61 });
62
63 botonSiguiente.setOnClickListener(new OnClickListener(){
64     @Override
65     public void onClick(View v) {
66         ficheroTXT = nombreFichero.getText().toString();
67         if (!ficheroTXT.isEmpty() && !m_chosenDir.isEmpty()){
68             Intent intent = new Intent(Primera_Actividad.this, Segunda_Actividad.class);
69             intent.putExtra("nombreFichero", ficheroTXT);
70             intent.putExtra("directorio", m_chosenDir);
71             startActivity(intent);
72             Primera_Actividad.this.finish();
73         }else{
74             FragmentManager fragmentManager = getFragmentManager();
75             DialogoNombreFichero dialogoNombreFichero = new DialogoNombreFichero();
76             dialogoNombreFichero.show(fragmentManager, "tagAlerta");
77         }
78     }
79 });
80 }
81 }
```

```

Segunda_Actividad.java
1 package org.elena.tfm.registrodeaceleracionmsband;
2 import android.app.Notification; // ...
3
4 public class Segunda_Actividad extends Activity {
5
6     protected static final int ALARM_REQUEST_COMIENZO = 1;
7     private Button botonHora_comienzo;
8     private Button botonFecha_comienzo;
9     private Button botonHora_fin;
10    private Button botonFecha_fin;
11    private Button botonComenzar;
12    private TextView txtFecha_comienzo;
13    private TextView txtHora_comienzo;
14    private TextView txtFecha_fin;
15    private TextView txtHora_fin;
16    private TextViewEx ms_conexion;
17    public long tiempoComienzoMilisegundos;
18    public long tiempoFinMilisegundos;
19    public long tiempoRegistroMilisegundos;
20    private int pHora_comienzo;
21    private int pMin_comienzo;
22    private int mAno_comienzo;
23    private int mMes_comienzo;
24    private int mDia_comienzo;
25    private int pHora_fin;
26    private int pMin_fin;
27    private int mAno_fin;
28    private int mMes_fin;
29    private int mDia_fin;
30    private String nombreFichero;
31    private String directorio;
32    public NotificationManager nManager;
33    DateFormat fmtDateAndTime = DateFormat.getDateTimeInstance();
34    final int NOTIFICATION_ID = 333;
35
36    @Override
37    protected void onCreate(Bundle savedInstanceState) {
38        super.onCreate(savedInstanceState);
39        setContentView(R.layout.activity_segunda_actividad);
40        SpannableString s = new SpannableString("Registro de Aceleracion MsB");
41        s.setSpan(new TypefaceSpan(getApplicationContext(), "Cubano-Regular.otf"), 0, s.length(),
42            ↳ Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
43        s.setSpan(new ForegroundColorSpan(Color.WHITE), 0, s.length(),
44            ↳ Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
45        ActionBar actionBar = getActionBar();
46        actionBar.setTitle(s);
47        nManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
48        botonHora_comienzo = (Button) findViewById(R.id.botonHora_comienzo);
49        botonFecha_comienzo = (Button) findViewById(R.id.botonFecha_comienzo);
50        txtFecha_comienzo = (TextView) findViewById(R.id.txtFecha_comienzo);
51        txtHora_comienzo = (TextView) findViewById(R.id.txtHora_comienzo);
52        botonHora_fin = (Button) findViewById(R.id.botonHora_fin);
53        botonFecha_fin = (Button) findViewById(R.id.botonFecha_fin);
54        txtFecha_fin = (TextView) findViewById(R.id.txtFecha_fin);
55        txtHora_fin = (TextView) findViewById(R.id.txtHora_fin);

```

```

54     botonComenzar = (Button) findViewById(R.id.botonComenzar);
55     ms_conexion = (TextViewEx) findViewById(R.id.ms_conexion);
56     ms_conexion.setText(getResources().getString(R.string.conexion),true);
57     nombreFichero = getIntent().getExtras().getString("nombreFichero");
58     directorio = getIntent().getExtras().getString("directorio");
59     Typeface fuenteBotones = Typeface.createFromAsset(getAssets(), "Fuentes/Cubano-Regular.otf");
60     botonHora_comienzo.setTypeface(fuenteBotones);
61     botonFecha_comienzo.setTypeface(fuenteBotones);
62     botonHora_fin.setTypeface(fuenteBotones);
63     botonFecha_fin.setTypeface(fuenteBotones);
64     botonComenzar.setTypeface(fuenteBotones);
65     txtFecha_comienzo.setTypeface(fuenteBotones);
66     txtHora_comienzo.setTypeface(fuenteBotones);
67     txtFecha_fin.setTypeface(fuenteBotones);
68     txtHora_fin.setTypeface(fuenteBotones);
69
70     botonHora_comienzo.setOnClickListener(new OnClickListener(){
71         @Override
72         public void onClick(View v){
73             Calendar calendar = Calendar.getInstance();
74             int hora = calendar.get(Calendar.HOUR_OF_DAY);
75             int min = calendar.get(Calendar.MINUTE);
76             min = min+1;
77             TimePickerDialog tpd = new TimePickerDialog(Segunda_Actividad.this, new
78                 ↪ TimePickerDialog.OnTimeSetListener() {
79                 @Override
80                 public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
81                     pHora_comienzo = hourOfDay;
82                     pMin_comienzo = minute;
83                     String aux = Integer.toString(minute);
84                     StringBuilder ceroIzq = new StringBuilder("0");
85                     if(minute<10){
86                         ceroIzq.append(minute);
87                         aux = ceroIzq.toString();
88                     }
89                     txtHora_comienzo.setText(hourOfDay +":"+ aux);
90                 }
91             }, hora, min, true);
92             tpd.show();
93         }
94     });
95
96     botonHora_fin.setOnClickListener(new OnClickListener(){
97         @Override
98         public void onClick(View v){
99             Calendar calendar = Calendar.getInstance();
100             int hora = calendar.get(Calendar.HOUR_OF_DAY);
101             int min = calendar.get(Calendar.MINUTE);
102             min = min+1;
103             TimePickerDialog tpd = new TimePickerDialog(Segunda_Actividad.this, new
104                 ↪ TimePickerDialog.OnTimeSetListener() {
105                 @Override
106                 public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
107                     pHora_fin = hourOfDay;
108                     pMin_fin = minute;
109                     String aux = Integer.toString(minute);

```

```

108         StringBuilder ceroIzq = new StringBuilder("0");
109         if(minute<10){
110             ceroIzq.append(minute);
111             aux = ceroIzq.toString();
112         }
113         txtHora_fin.setText(hourOfDay + ":"+ aux);
114     }
115     }, hora, min, true);
116     tpd.show();
117 }
118 });
119
120 botonFecha_comienzo.setOnClickListener(new OnClickListener(){
121     @Override
122     public void onClick(View v){
123         Calendar calendar = Calendar.getInstance();
124         int ano = calendar.get(Calendar.YEAR);
125         int mes = calendar.get(Calendar.MONTH);
126         int dia = calendar.get(Calendar.DAY_OF_MONTH);
127         DatePickerDialog dpd = new DatePickerDialog(Segunda_Actividad.this, new
128             ↳ DatePickerDialog.OnDateSetListener() {
129             @Override
130             public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
131                 mAno_comienzo = year;
132                 mMes_comienzo = monthOfYear;
133                 mDia_comienzo = dayOfMonth;
134                 int mes = mMes_comienzo+1;
135                 String aux1 = Integer.toString(mDia_comienzo);
136                 String aux2 = Integer.toString(mes);
137                 StringBuilder ceroIzq1 = new StringBuilder("0");
138                 StringBuilder ceroIzq2 = new StringBuilder("0");
139                 if(mDia_comienzo<10){
140                     ceroIzq1.append(mDia_comienzo);
141                     aux1 = ceroIzq1.toString();
142                 }
143                 if(mes<10){
144                     ceroIzq2.append(mes);
145                     aux2 = ceroIzq2.toString();
146                 }
147                 txtFecha_comienzo.setText(aux1 + "/" + aux2 + "/" + year);
148             }
149             }, ano, mes, dia);
150         dpd.show();
151     }
152 });
153
154 botonFecha_fin.setOnClickListener(new OnClickListener(){
155     @Override
156     public void onClick(View v){
157         Calendar calendar = Calendar.getInstance();
158         int ano = calendar.get(Calendar.YEAR);
159         int mes = calendar.get(Calendar.MONTH);
160         int dia = calendar.get(Calendar.DAY_OF_MONTH);
161         DatePickerDialog dpd = new DatePickerDialog(Segunda_Actividad.this, new
162             ↳ DatePickerDialog.OnDateSetListener() {
163             @Override

```

```

162     public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
163         mAno_fin = year;
164         mMes_fin = monthOfYear;
165         mDia_fin = dayOfMonth;
166         int mes = mMes_fin+1;
167         String aux1 = Integer.toString(mDia_fin);
168         String aux2 = Integer.toString(mes);
169         StringBuilder ceroIzq1 = new StringBuilder("0");
170         StringBuilder ceroIzq2 = new StringBuilder("0");
171         if(mDia_fin<10){
172             ceroIzq1.append(mDia_fin);
173             aux1 = ceroIzq1.toString();
174         }
175         if(mes<10){
176             ceroIzq2.append(mes);
177             aux2 = ceroIzq2.toString();
178         }
179         txtFecha_fin.setText(aux1 + "/" + aux2 + "/" + year);
180     }
181     }, ano, mes, dia);
182     dpd.show();
183     }
184     });
185
186     botonComenzar.setOnClickListener(new OnClickListener(){
187         @Override
188         public void onClick(View v) {
189             if(mAno_comienzo!=0 && mMes_comienzo!=0 && mDia_comienzo!=0){
190                 if(mAno_fin!=0 && mMes_fin!=0 && mDia_fin!=0){
191                     Calendar calendar_comienzo = Calendar.getInstance();
192                     calendar_comienzo.set(mAno_comienzo, mMes_comienzo, mDia_comienzo, pHora_comienzo,
193                         ↪ pMin_comienzo, 0);
194                     tiempoComienzoMilisegundos = calendar_comienzo.getTimeInMillis();
195                     Calendar calendar_fin = Calendar.getInstance();
196                     calendar_fin.set(mAno_fin, mMes_fin, mDia_fin, pHora_fin, pMin_fin, 0);
197                     tiempoFinMilisegundos = calendar_fin.getTimeInMillis();
198                     if (System.currentTimeMillis() < tiempoComienzoMilisegundos){
199                         tiempoRegistroMilisegundos = tiempoFinMilisegundos - tiempoComienzoMilisegundos;
200                         if(tiempoRegistroMilisegundos>0 && tiempoRegistroMilisegundos<=(24*3600000)){
201                             AlarmManager manager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
202                             Intent intent = new Intent(getApplicationContext(), Receptor.class);
203                             Intent intentCancelar = new Intent(getApplicationContext(), ReceptorCancelar.class);
204                             Bitmap icon = BitmapFactory.decodeResource(getResources(), R.drawable.ic_heart);
205                             intentCancelar.putExtra("notificationId", NOTIFICATION_ID);
206                             intentCancelar.putExtra("alarmId", ALARM_REQUEST_COMIENZO);
207                             PendingIntent pendingIntentCancelar =
208                                 ↪ PendingIntent.getBroadcast(getApplicationContext(), 1, intentCancelar,
209                                 ↪ PendingIntent.FLAG_ONE_SHOT);
210                             intent.putExtra("nombreFichero", nombreFichero);
211                             intent.putExtra("directorio", directorio);
212                             intent.putExtra("milisegundos", tiempoRegistroMilisegundos);
213                             intent.putExtra("notificationId", NOTIFICATION_ID);
214                             PendingIntent pIntent = PendingIntent.getBroadcast(getApplicationContext(),
215                                 ↪ ALARM_REQUEST_COMIENZO, intent, PendingIntent.FLAG_CANCEL_CURRENT);
216                             manager.set(AlarmManager.RTC_WAKEUP, tiempoComienzoMilisegundos, pIntent);

```

```

213     NotificationCompat.Action action = new
        ↳ NotificationCompat.Action.Builder(android.R.drawable.ic_delete, "Cancelar
        ↳ registro", PendingIntentCanceler).build();
214     Notification notificacionProgramacion = new
        ↳ NotificationCompat.Builder(getApplicationContext())
215         .setContentTitle("Registro de Aceleración MSBand")
216         .setTicker("Registro de Aceleración MSBand")
217         .setContentText("Programado: " +
        ↳ fmtDateAndTime.format(calendar_comienzo.getTime()))
218         .setSmallIcon(android.R.drawable.ic_dialog_info)
219         .setLargeIcon(Bitmap.createScaledBitmap(icon, 128, 128, false))
220         .setDefaults(Notification.DEFAULT_VIBRATE)
221         .addAction(action)
222         .setWhen(System.currentTimeMillis()).build();
223     nManager.notify(NOTIFICATION_ID, notificacionProgramacion);
224     Segunda_Actividad.this.finish();
225     }else{
226         FragmentManager fragmentManager = getFragmentManager();
227         DialogoFechaError dialogoFechaError = new DialogoFechaError();
228         dialogoFechaError.show(fragmentManager, "tagAlerta");
229     }
230     }else{
231         FragmentManager fragmentManager = getFragmentManager();
232         DialogoTiempo dialogoTiempo = new DialogoTiempo();
233         dialogoTiempo.show(fragmentManager, "tagAlerta");
234     }
235     }else{
236         FragmentManager fragmentManager = getFragmentManager();
237         DialogoFechaFin dialogoFechaFin = new DialogoFechaFin();
238         dialogoFechaFin.show(fragmentManager, "tagAlerta");
239     }
240     }else{
241         FragmentManager fragmentManager = getFragmentManager();
242         DialogoFechaComienzo dialogoFechaComienzo = new DialogoFechaComienzo();
243         dialogoFechaComienzo.show(fragmentManager, "tagAlerta");
244     }
245     }
246     });
247 }
248 }

```

```
ReceptorCancelar.java
1 package org.elena.tfm.registroidceceleracionmsband;
2 import android.app.AlarmManager; // ...
3
4 public class ReceptorCancelar extends BroadcastReceiver {
5
6     @Override
7     public void onReceive(Context context, Intent intent) {
8         int notificationId = intent.getIntExtra("notificationId", 0);
9         int alarmId = intent.getIntExtra("alarmId", 0);
10        Intent alarmIntent = new Intent(context, Receptor.class);
11        PendingIntent pendingIntent = PendingIntent.getBroadcast(context, alarmId, alarmIntent,
12        ↳ PendingIntent.FLAG_UPDATE_CURRENT);
13        AlarmManager almanager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
14        pendingIntent.cancel();
15        almanager.cancel(pendingIntent);
16        NotificationManager manager = (NotificationManager)
17        ↳ context.getSystemService(Context.NOTIFICATION_SERVICE);
18        manager.cancel(notificationId);
19    }
20 }
```

```
Receptor.java
1 package org.elena.tfm.registrodeaceleracionmsband;
2 import android.app.NotificationManager; // ...
3
4 public class Receptor extends android.content.BroadcastReceiver {
5
6     @Override
7     public void onReceive(android.content.Context context, android.content.Intent intent) {
8         String nombreFichero = intent.getExtras().getString("nombreFichero");
9         String directorio = intent.getExtras().getString("directorio");
10        long tiempoRegistroMilisegundos = intent.getExtras().getLong("milisegundos");
11        int notificationId = intent.getExtras().getInt("notificationId", 333);
12        Intent startIntent = new Intent(context, ServicioBackground.class);
13        startIntent.putExtra("nombreFichero", nombreFichero);
14        startIntent.putExtra("directorio", directorio);
15        startIntent.putExtra("milisegundos", tiempoRegistroMilisegundos);
16        NotificationManager manager = (NotificationManager)
17        ↪ context.getSystemService(Context.NOTIFICATION_SERVICE);
18        manager.cancel(notificationId);
19        context.startService(startIntent);
20    }
}
```

```

1 package org.elena.tfm.registrodeaceleracionmsband;
2 import android.app.IntentService; // ...
3
4 public class ServicioBackground extends IntentService{
5
6     protected static final String TAG = null;
7     private OutputStreamWriter fout;
8     public long tiempoComienzo_aceleracion;
9     private BandClient client = null;
10    public final int minutosMax = 24*60;
11    private long tiempoRegistroMilisegundos;
12    public int iteracion = 0;
13    public String nombreFichero;
14    public String directorio;
15    public NotificationManager nManager;
16    final int NOTIFICACION_ID = 444;
17
18    private BandAccelerometerEventListener mAccelerometerEventListener = new
19    ↪ BandAccelerometerEventListener() {
20        @Override
21        public void onBandAccelerometerChanged(final BandAccelerometerEvent event) {
22            if((SystemClock.uptimeMillis() <= (tiempoComienzo_aceleracion + tiempoRegistroMilisegundos))){
23                if (event != null) {
24                    // Aceleración en unidades g -> 1g = 9.81 metros por segundo2
25                    String texto = "X = "+Float.toString(event.getAccelerationX())
26                        + "\nY = "+Float.toString(event.getAccelerationY())
27                        + "\nZ = "+Float.toString(event.getAccelerationZ());
28
29                    Bitmap icon = BitmapFactory.decodeResource(getResources(), R.drawable.ic_heart);
30                    Intent notificationIntent = new Intent();
31                    PendingIntent pendingIntent = PendingIntent.getActivity(getApplicationContext(), 12,
32                    ↪ notificationIntent, 0);
33                    Notification.Builder notificacionInicio = new
34                    ↪ Notification.Builder(getApplicationContext())
35                        .setSmallIcon(android.R.drawable.ic_dialog_info)
36                        .setTicker("Registro de Aceleración MSBand")
37                        .setLargeIcon(Bitmap.createScaledBitmap(icon, 128, 128, false))
38                        .setContentIntent(pendingIntent)
39                        .setContentTitle("Registro de Aceleración MSBand")
40                        .setContentText("Registrando...")
41                        .setStyle(new Notification.BigTextStyle().bigText(texto));
42                    startForeground(NOTIFICACION_ID, notificacionInicio.build());
43
44                    File ficheroAceleracion = new File(directorio, nombreFichero + ".txt");
45                    try {
46                        fout = new OutputStreamWriter(new FileOutputStream(ficheroAceleracion, true));
47                        fout.append(Float.toString(event.getAccelerationX()));
48                        fout.append(" ");
49                        fout.append(Float.toString(event.getAccelerationY()));
50                        fout.append(" ");
51                        fout.append(Float.toString(event.getAccelerationZ()));
52                        fout.append(" ");
53                        fout.close();
54                    } catch (FileNotFoundException e) {
55                        e.printStackTrace();
56                    }
57                }
58            }
59        }
60    };
61
62 }

```

```

53         } catch (IOException e) {
54             e.printStackTrace();
55         }
56     }
57     }else{
58         try {
59             client.getSensorManager()
60                 .unregisterAccelerometerEventListener(mAccelerometerEventListener);
61         } catch (BandIOException e) {
62             e.printStackTrace();
63         }
64     }
65 }
66 };
67
68 public ServicioBackground() {
69     // Used to name the worker thread, important only for debugging.
70     super("ServicioBackground");
71 }
72
73 @Override
74 public void onCreate() {
75     super.onCreate();
76     nManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
77 }
78
79 @Override
80 protected void onHandleIntent(Intent intent) {
81     nombreFichero = intent.getExtras().getString("nombreFichero");
82     directorio = intent.getExtras().getString("directorio");
83     tiempoRegistroMilisegundos = intent.getExtras().getLong("milisegundos");
84
85     Bitmap icon = BitmapFactory.decodeResource(getResources(), R.drawable.ic_heart);
86     Notification notificacionError = new Notification.Builder(getApplicationContext())
87         .setContentTitle("Registro de Aceleración MSBand")
88         .setTicker("Registro de Aceleración MSBand")
89         .setContentText("Tu pulsera no está conectada")
90         .setSmallIcon(android.R.drawable.ic_dialog_alert)
91         .setLargeIcon(Bitmap.createScaledBitmap(icon, 128, 128, false))
92         .setDefaults(Notification.DEFAULT_VIBRATE)
93         .setDefaults(Notification.DEFAULT_LIGHTS)
94         .setWhen(System.currentTimeMillis()).build();
95
96     try {
97         getConnectedBandClient();
98
99         if (getConnectedBandClient()) {
100             tiempoComienzo_aceleracion = SystemClock.uptimeMillis();
101             client.getSensorManager().registerAccelerometerEventListener(mAccelerometerEventListener,
102                 ↪ SampleRate.MS32);
103
104             while(SystemClock.uptimeMillis() <= (tiempoComienzo_aceleracion +
105                 ↪ tiempoRegistroMilisegundos)){
106                 iteracion++;
107             }

```

```

107     stopForeground(true);
108
109     Notification notificacionFin = new Notification.Builder(getApplicationContext())
110         .setContentTitle("Registro de Aceleración MSBand")
111         .setTicker("Registro de Aceleración MSBand")
112         .setContentText("Registro completo")
113         .setSmallIcon(android.R.drawable.ic_dialog_info)
114         .setLargeIcon(Bitmap.createScaledBitmap(icon, 128, 128, false))
115         .setDefaults(Notification.DEFAULT_VIBRATE)
116         .setDefaults(Notification.DEFAULT_LIGHTS)
117         .setWhen(System.currentTimeMillis()).build();
118     nManager.notify(NOTIFICACION_ID, notificacionFin);
119 }else{
120     nManager.notify(14, notificacionError);
121     Log.e("Pulsera", "Error");
122     stopService(intent);
123 }
124 } catch (BandException e) {
125     nManager.notify(15, notificacionError);
126     Log.e("Pulsera", "Error");
127     stopService(intent);
128 } catch (InterruptedException e) {
129     e.printStackTrace();
130 }
131 }
132
133 @Override
134 public void onDestroy() {
135     if (client != null) {
136         try {
137             client.disconnect().await();
138         } catch (InterruptedException e) {
139             // Do nothing as this is happening during destroy
140         } catch (BandException e) {
141             // Do nothing as this is happening during destroy
142         }
143     }
144     super.onDestroy();
145 }
146
147 public boolean getConnectedBandClient() throws InterruptedException, BandException {
148     if (client == null) {
149         BandInfo[] devices = BandClientManager.getInstance().getPairedBands();
150         if (devices.length == 0) {
151             Toast.makeText(this, "La pulsera no está emparejada con tu
152             ↵ smartphone", Toast.LENGTH_SHORT).show();
153             return false;
154         }
155         client = BandClientManager.getInstance().create(getBaseContext(), devices[0]);
156     } else if (ConnectionState.CONNECTED == client.getConnectionState()) {
157         return true;
158     }
159     return ConnectionState.CONNECTED == client.connect().await();
160 }

```



## Apéndice B

# MANUAL DE USUARIO DE LA APLICACIÓN

---

En este apéndice se encuentran recogidos los pasos necesarios para llevar a cabo la puesta a punto del entorno para realizar registros con la aplicación **Registro de Aceleración MSBand**.



FIGURA B.1: Icono de de la aplicación Registro de Aceleración MSBand.

## B.1 ANTES DE COMENZAR

---

Lo primero es vincular o emparejar la pulsera con el smartphone. Para ello:

- Desde la pulsera:
  - Ajustes →Bluetooth →Emparejamiento.
  - *¿Emparejar con iPhone?* **NO**
- Desde el smartphone (figura B.2):
  - Ajustes →Bluetooth →Activar.
  - Seleccionar la pulsera Microsoft Band de entre los dispositivo disponibles.

Aparecerá un código de vinculación en las pantallas de ambos dispositivos. El código de vinculación que aparezca en la pantalla de la pulsera debe coincidir con el que aparezca en el smartphone. Lo siguiente será disponer de la aplicación Microsoft Band (o Microsoft Health, en versiones anteriores). Para descargarla e instalarla basta con acceder a la tienda de Android, Play Store, como se observa en la figura B.3.

## B.2 INSTALACIÓN DE LA APLICACIÓN

---

Para proceder a la instalación de la aplicación Registro de Aceleración MSBand es necesario disponer del archivo APK, que contiene el código compilado (figura B.5(a)). Mediante el instalador de paquetes de Android se pone en marcha la instalación, que advierte al usuario de los permisos que proporciona a la aplicación (figura B.5(b)).

Una vez que la aplicación ha sido instalada ya se encuentra disponible para el usuario. Puede ser lanzada inmediatamente desde la pantalla que indica el fin de la instalación (figura B.5(c)) o desde el menú de aplicaciones (figura B.6).

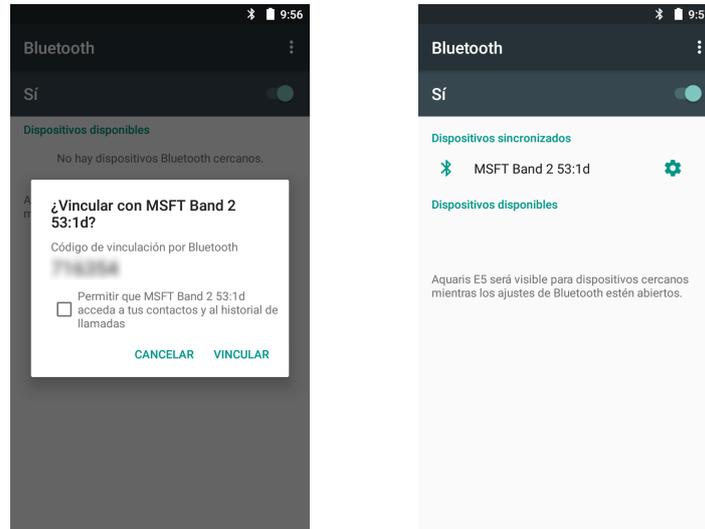


FIGURA B.2: Emparejar Microsoft Band 2 y smartphone.



FIGURA B.3: Play Store. Aplicación Microsoft Band.

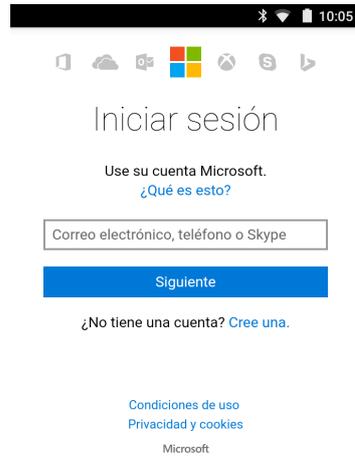
## B.3 EJEMPLO DE USO

A continuación, se propone un ejemplo de registro:

1. Iniciamos la aplicación. En la primera pantalla seleccionamos el directorio en el que se guardará el fichero con los datos del registro, y escribimos el nombre de tal fichero. Pulsamos en *Siguiente* (figura B.7(a)).
2. En la segunda pantalla seleccionamos la fecha y hora de comienzo del registro, que debe ser posterior al instante actual. También seleccionamos la fecha y hora de fin del registro, y pulsamos en *Comenzar* (figura B.7(b)).
3. Aparecerá una notificación mostrando la programación del registro y permitiendo cancelarlo. Puesto que no queremos cancelarlo, deslizaremos la notificación hasta que desaparezca o, simplemente, esperaremos a que comience el registro (figura B.7(c)).

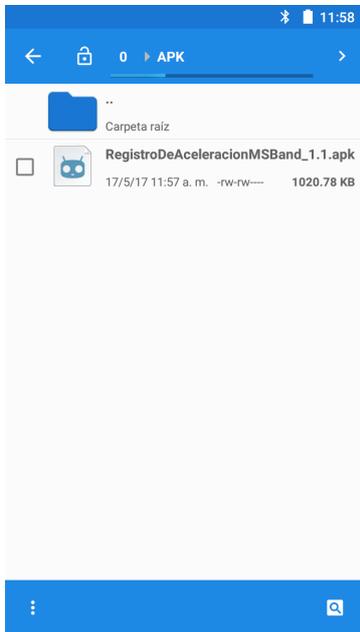


(a) Necesario iniciar sesión

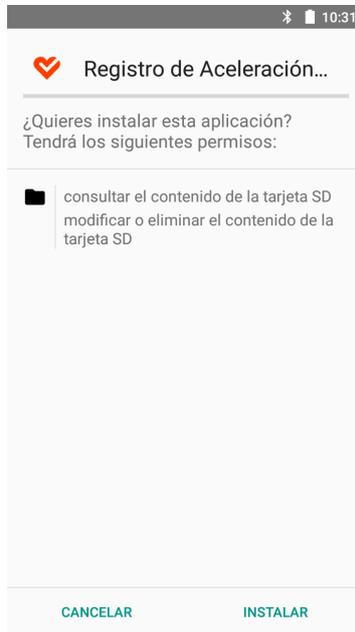


(b) Iniciar o crear cuenta

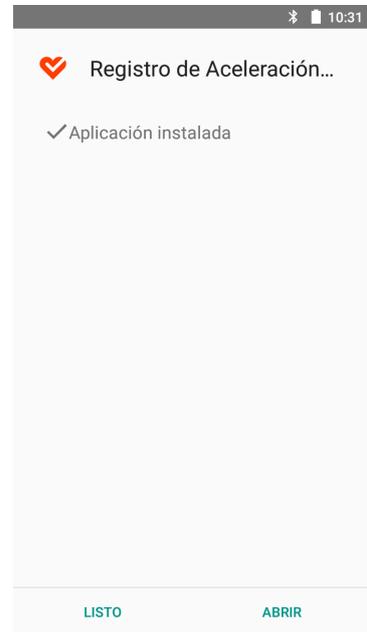
FIGURA B.4: Aplicación Microsoft Band. Primer uso.



(a) Archivo APK



(b) Confirmar permisos



(c) Instalación finalizada

FIGURA B.5: Instalación de la aplicación Registro de Aceleración MSBand.

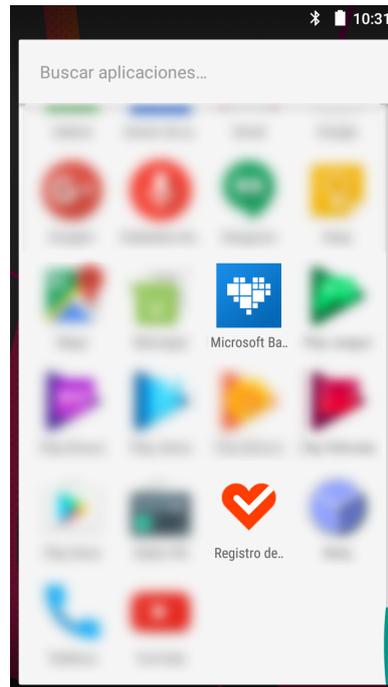
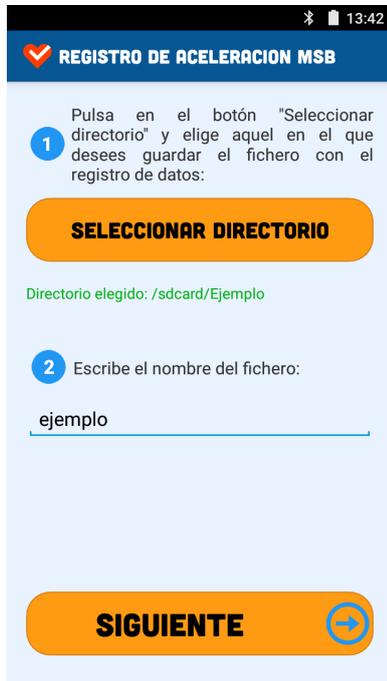


FIGURA B.6: Menú del smartphone.

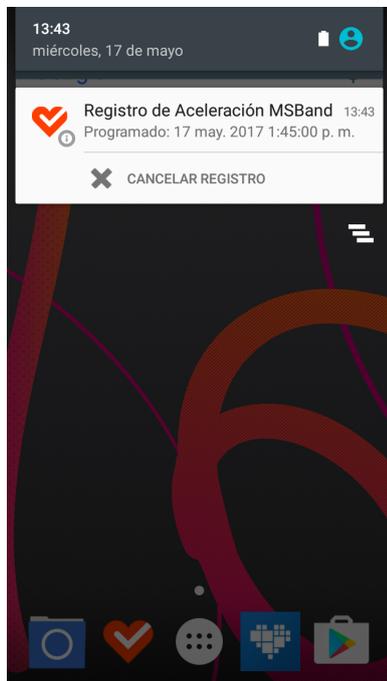
4. Una vez que el registro comienza y, durante el tiempo que hayamos configurado, se mostrarán en una notificación los datos que van siendo recogidos en el fichero (figura B.7(d)).
5. Cuando finalice el registro, se avisa al usuario mediante otra notificación (figura B.7(e)).
6. El fichero con los datos de aceleración registrados estará disponible en el directorio seleccionado en el primer paso (figura B.7(f)).



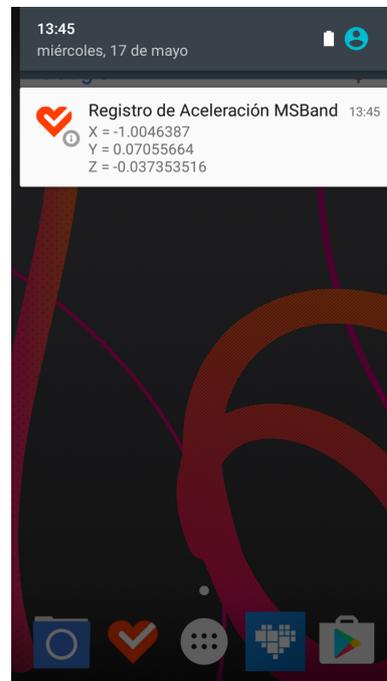
(a) Directorio y nombre del fichero



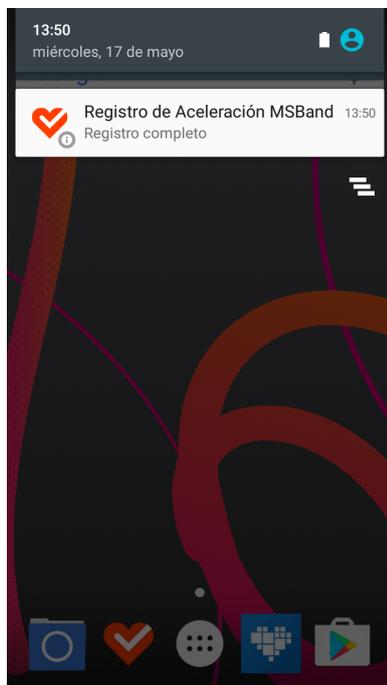
(b) Configuración del registro



(c) Registro programado



(d) Registrando



(e) Registro completo



(f) Datos recogidos

FIGURA B.7: Ejemplo de registro.

## Apéndice C

# BOXPLOTS

---

En este apéndice se presenta la comparación de los resultados obtenidos de ACT mediante *boxplots* en el tramo de sólo Actividad para los diezmados de 1 s, 1 min y 15 min, en los canales x (C.1), y (C.2), z (C.3). A estos *boxplots* se superponen los resultados obtenidos para los diferentes sujetos registrados con MSB.

### C.1 CANAL X

---

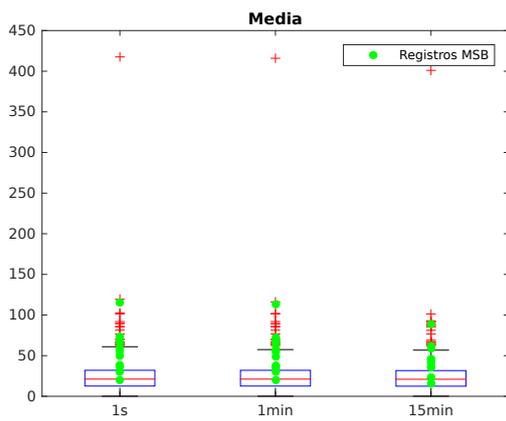


FIGURA C.1: Media. Canal x. Actividad.

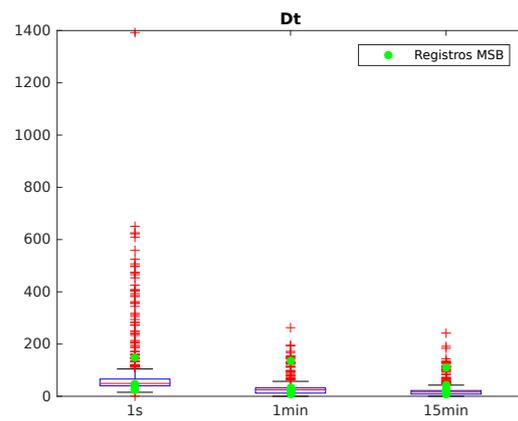


FIGURA C.2: Desviación estándar. Canal x. Actividad.

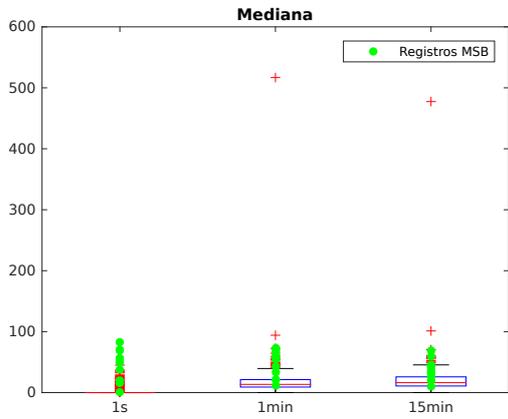


FIGURA C.3: Mediana. Canal x. Actividad.

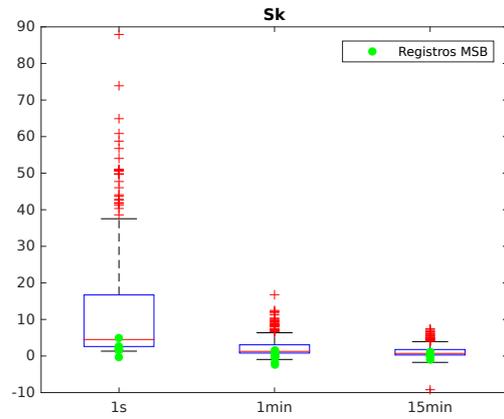


FIGURA C.4: Asimetría. Canal x. Actividad.

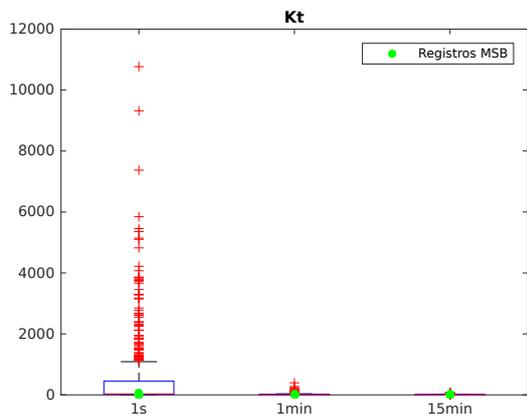


FIGURA C.5: Curtosis. Canal x. Actividad.

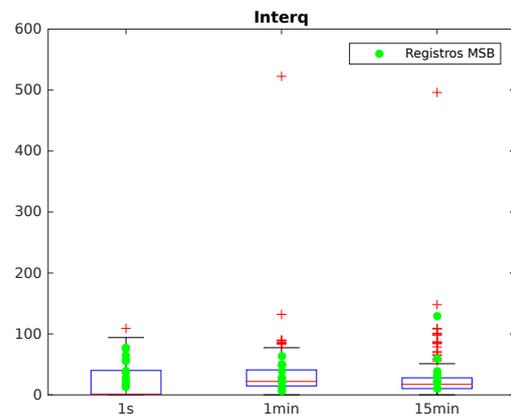


FIGURA C.6: Rango intercuartil. Canal x. Actividad.

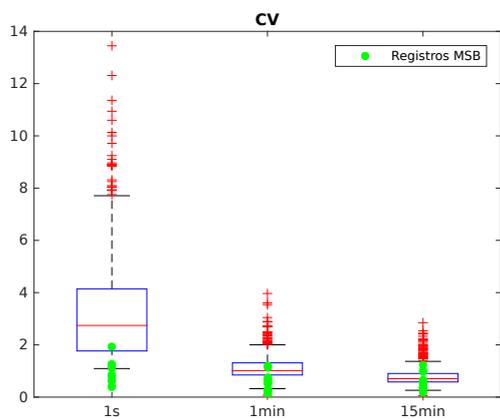


FIGURA C.7: Coeficiente de variación. Canal x. Actividad.

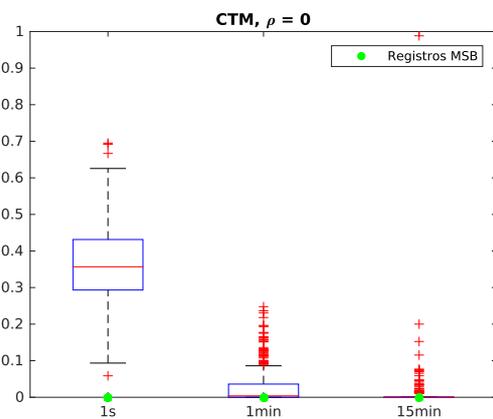


FIGURA C.8:  $CTM_{\rho=0}^x$ . Actividad.

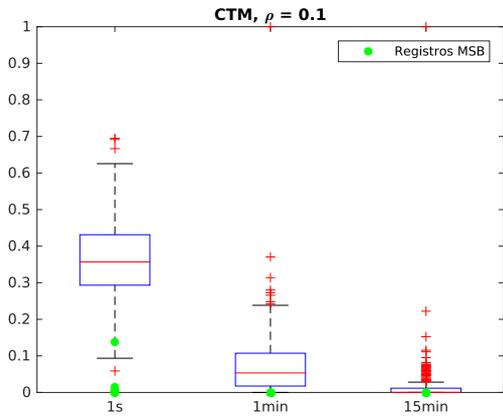


FIGURA C.9:  $CTM_{\rho=0,1}^x$ . Actividad.

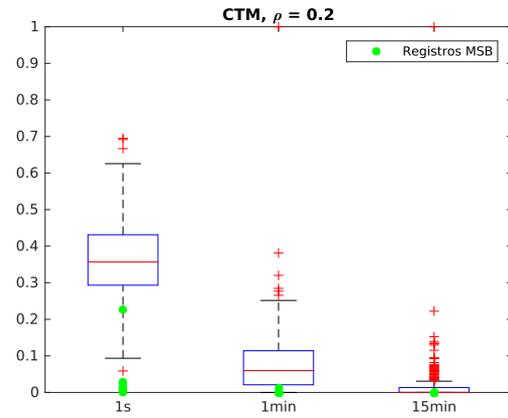


FIGURA C.10:  $CTM_{\rho=0,2}^x$ . Actividad.

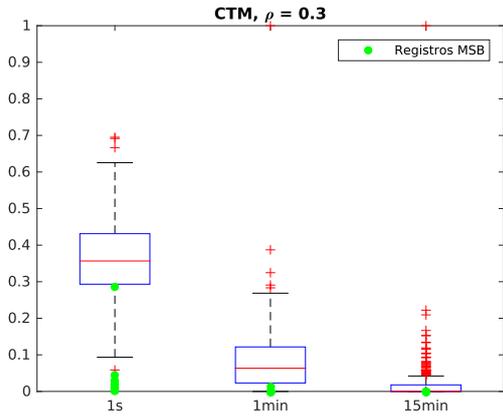


FIGURA C.11:  $CTM_{\rho=0,3}^x$ . Actividad.

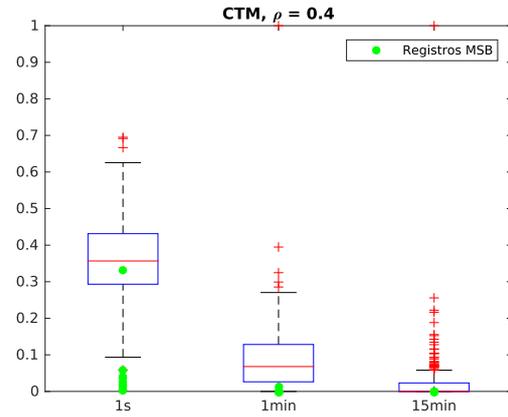


FIGURA C.12:  $CTM_{\rho=0,4}^x$ . Actividad.

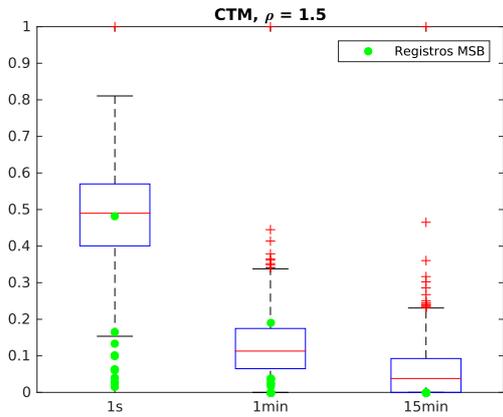


FIGURA C.13:  $CTM_{\rho=1,5}^x$ . Actividad.

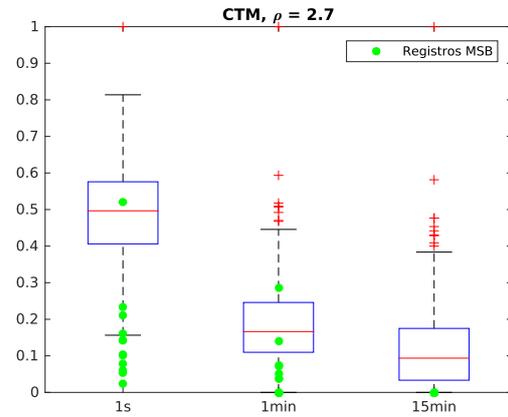


FIGURA C.14:  $CTM_{\rho=2,7}^x$ . Actividad.

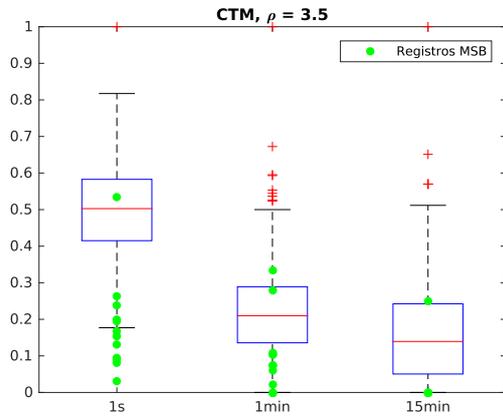


FIGURA C.15:  $CTM_{\rho=3,5}^x$ . Actividad.

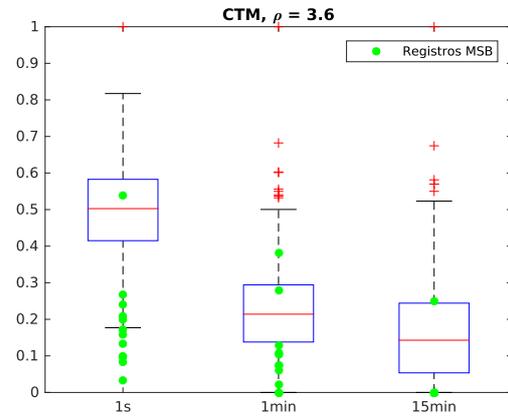


FIGURA C.16:  $CTM_{\rho=3,6}^x$ . Actividad.

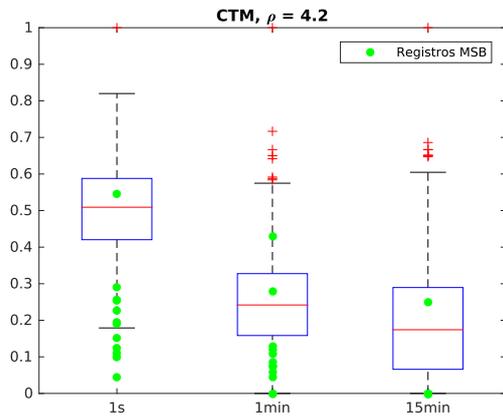


FIGURA C.17:  $CTM_{\rho=4,2}^x$ . Actividad.

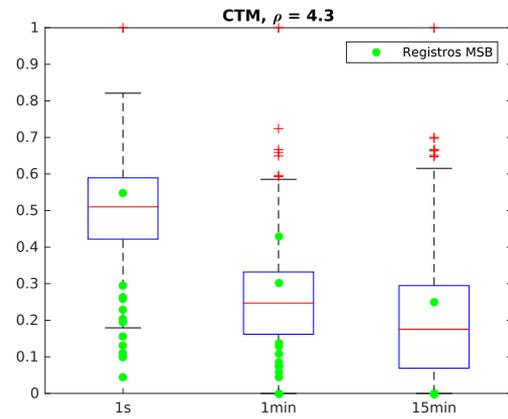


FIGURA C.18:  $CTM_{\rho=4,3}^x$ . Actividad.

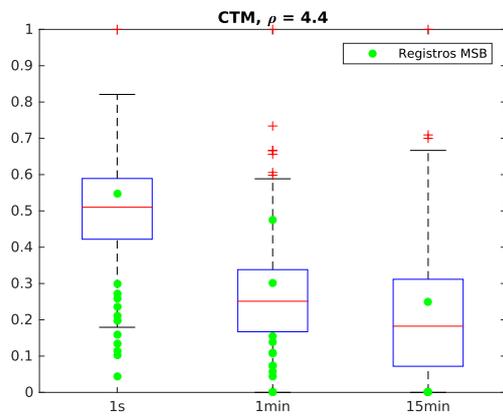


FIGURA C.19:  $CTM_{\rho=4,4}^x$ . Actividad.

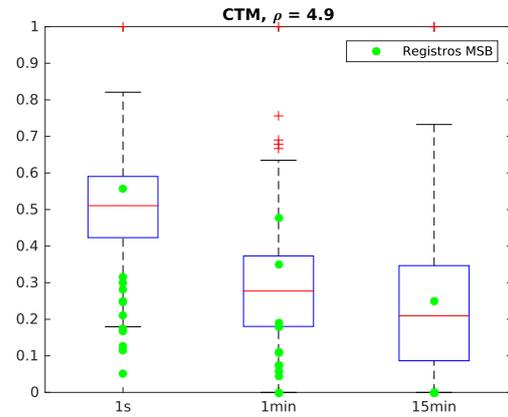


FIGURA C.20:  $CTM_{\rho=4,9}^x$ . Actividad.

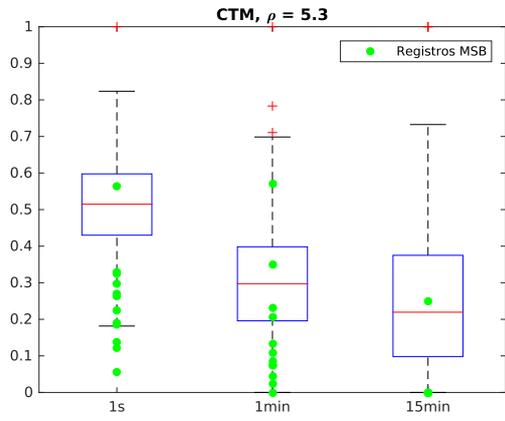


FIGURA C.21:  $CTM_{\rho=5,3}^x$ . Actividad.

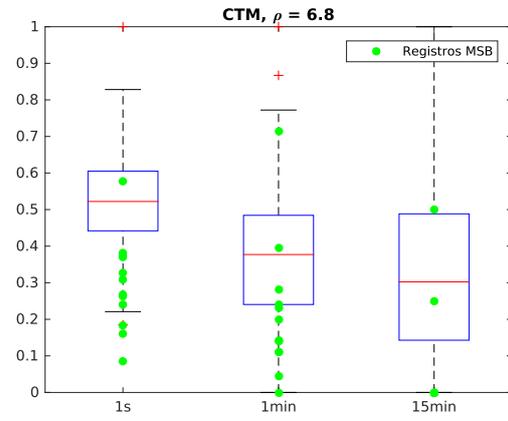


FIGURA C.22:  $CTM_{\rho=6,8}^x$ . Actividad.

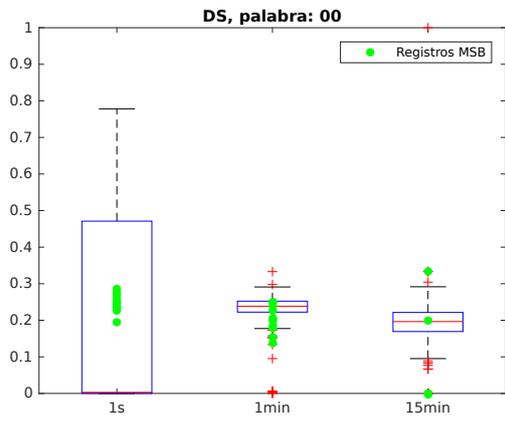


FIGURA C.23:  $P_{00}^x$ . Actividad.

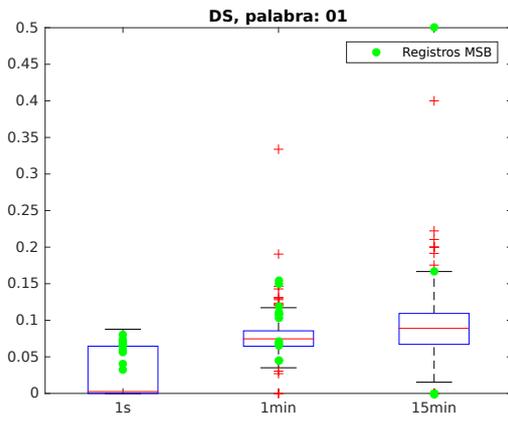


FIGURA C.24:  $P_{01}^x$ . Actividad.

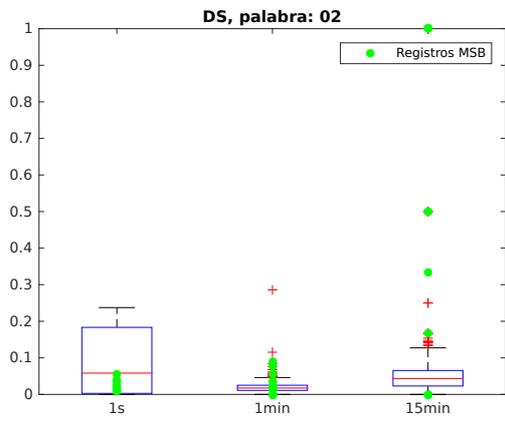


FIGURA C.25:  $P_{02}^x$ . Actividad.

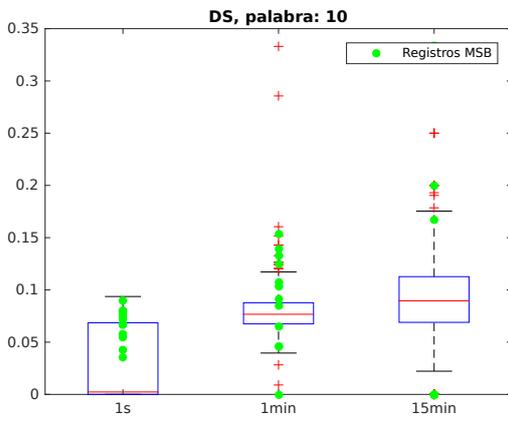


FIGURA C.26:  $P_{10}^x$ . Actividad.

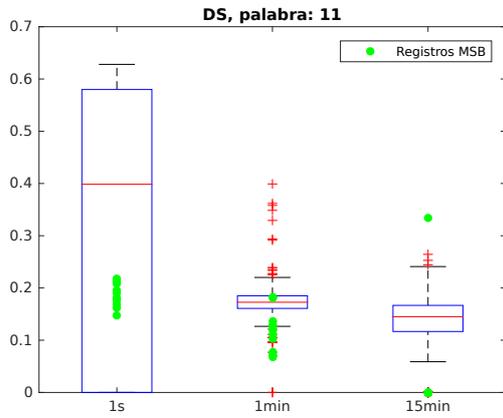


FIGURA C.27:  $P_{11}^x$ . Actividad.

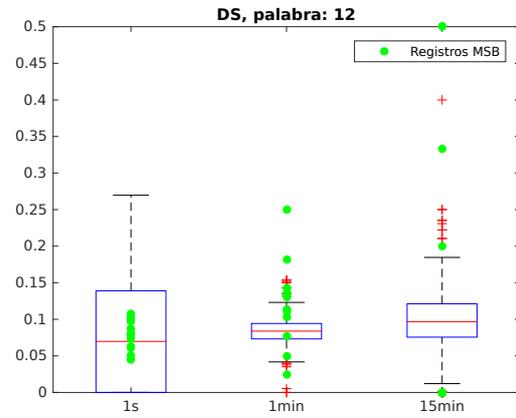


FIGURA C.28:  $P_{12}^x$ . Actividad.

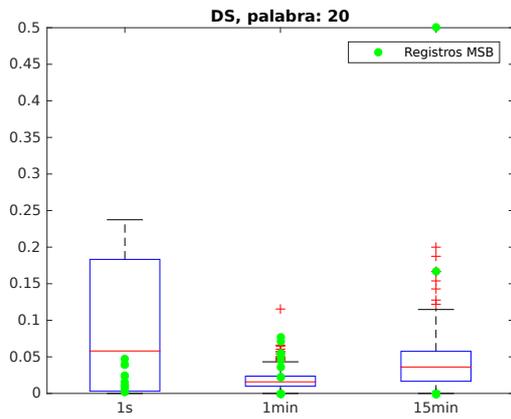


FIGURA C.29:  $P_{20}^x$ . Actividad.

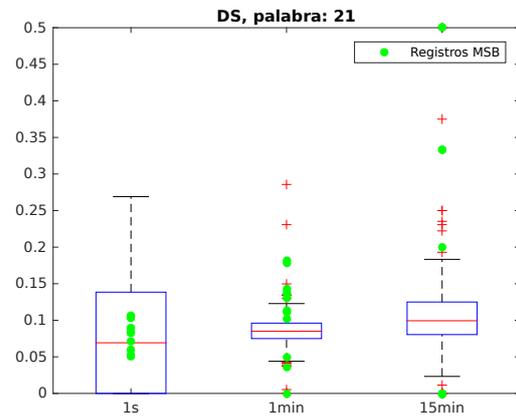


FIGURA C.30:  $P_{21}^x$ . Actividad.

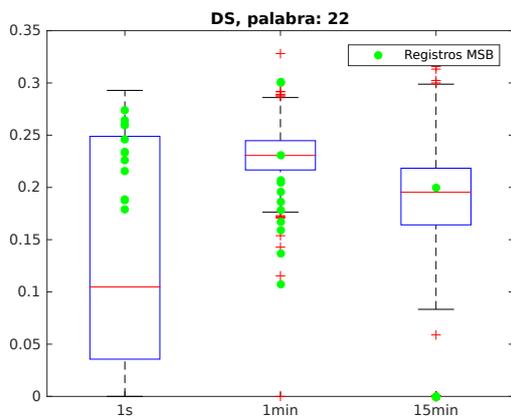


FIGURA C.31:  $P_{22}^x$ . Actividad.

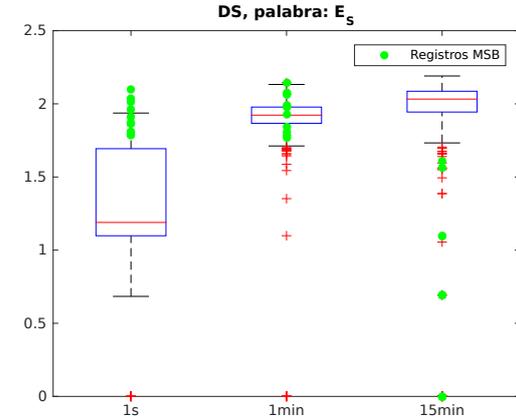


FIGURA C.32:  $E_{Shannon}^x$ . Actividad.

C.2 CANAL Y

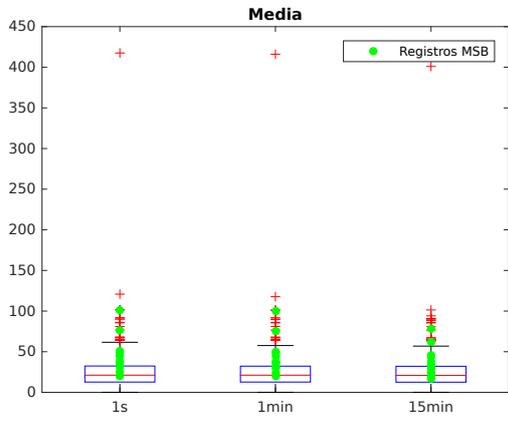


FIGURA C.33: Media. Canal y. Actividad.

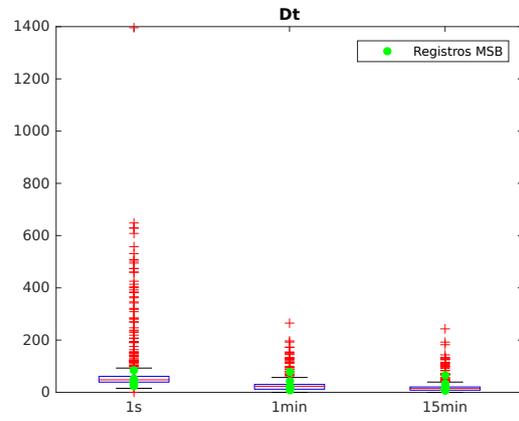


FIGURA C.34: Desviación estándar. Canal y. Actividad.

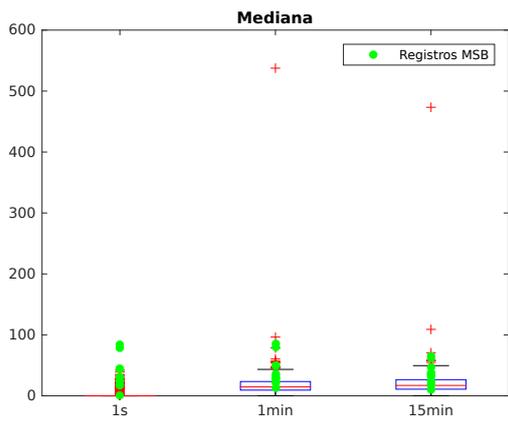


FIGURA C.35: Mediana. Canal y. Actividad.

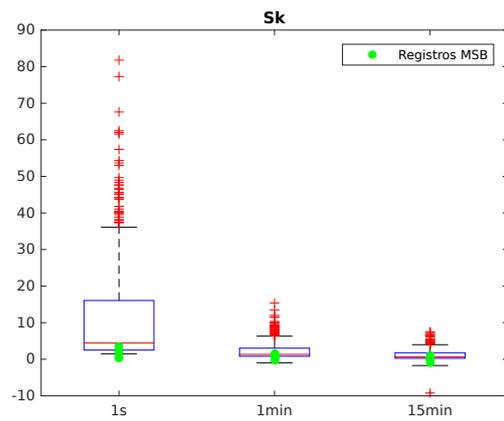


FIGURA C.36: Asimetría. Canal y. Actividad.

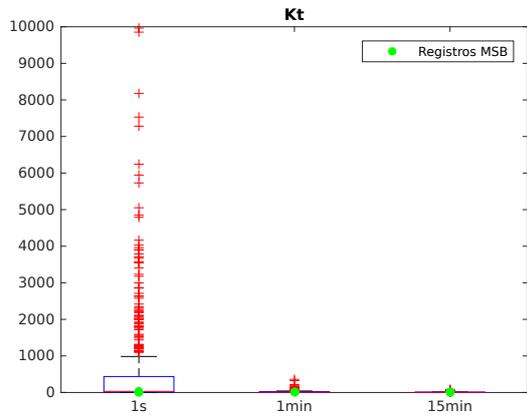


FIGURA C.37: Curtosis. Canal y. Actividad.

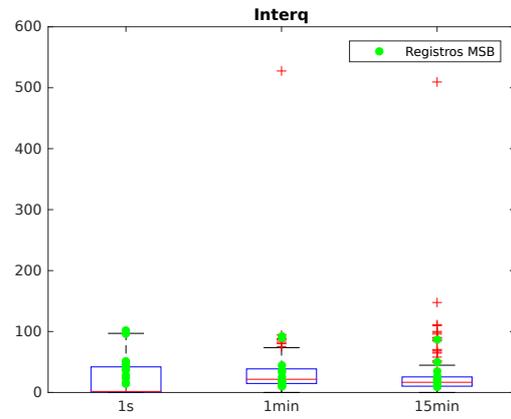


FIGURA C.38: Rango intercuartil. Canal y. Actividad.

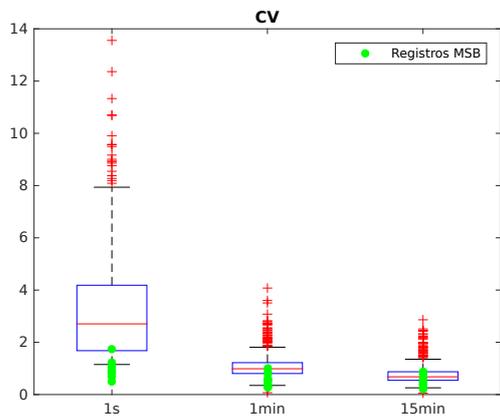


FIGURA C.39: Coeficiente de variación. Canal y. Actividad.

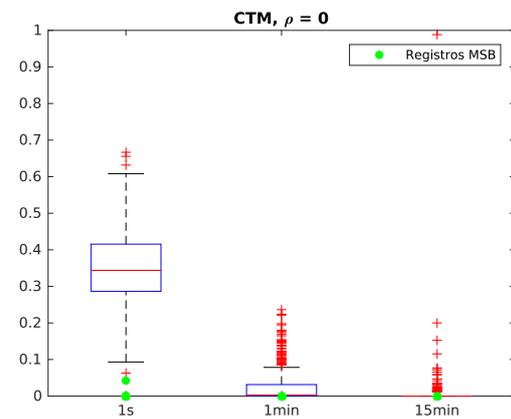


FIGURA C.40:  $CTM_{\rho=0}^y$ . Actividad.

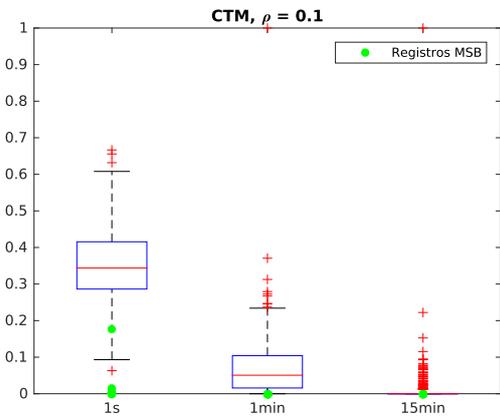


FIGURA C.41:  $CTM_{\rho=0,1}^y$ . Actividad.

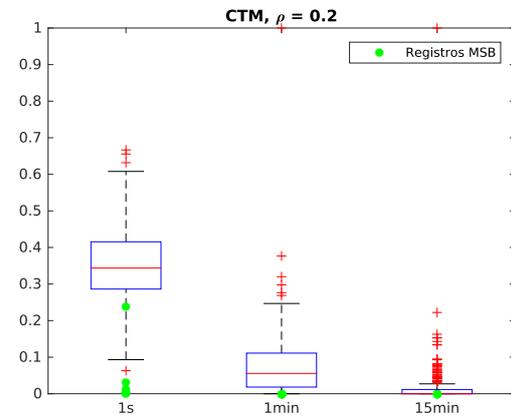


FIGURA C.42:  $CTM_{\rho=0,2}^y$ . Actividad.

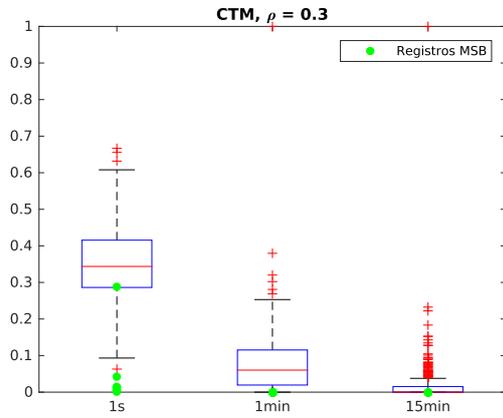


FIGURA C.43:  $CTM_{\rho=0,3}^y$ . Actividad.

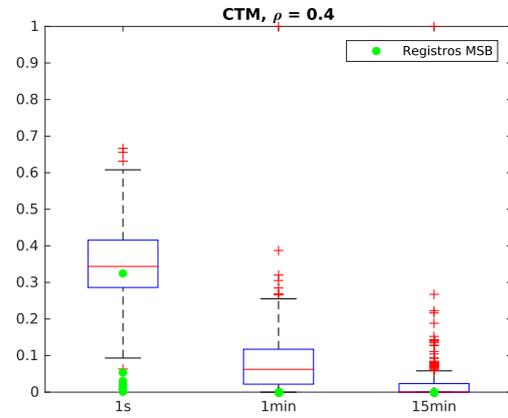


FIGURA C.44:  $CTM_{\rho=0,4}^y$ . Actividad.

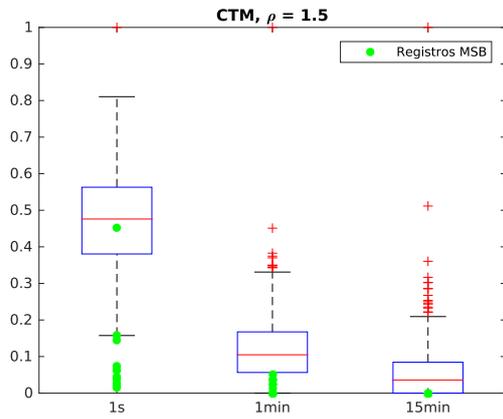


FIGURA C.45:  $CTM_{\rho=1,5}^y$ . Actividad.

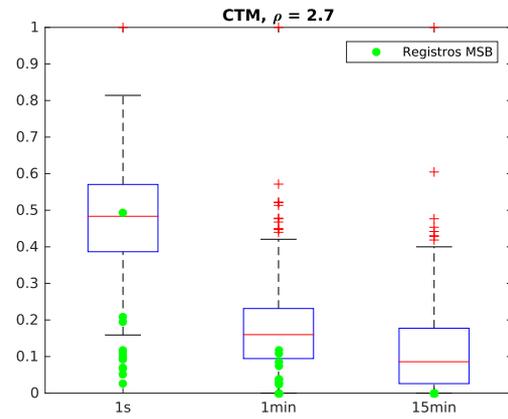


FIGURA C.46:  $CTM_{\rho=2,7}^y$ . Actividad.

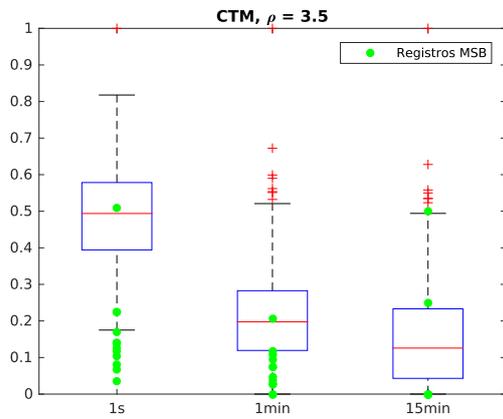


FIGURA C.47:  $CTM_{\rho=3,5}^y$ . Actividad.

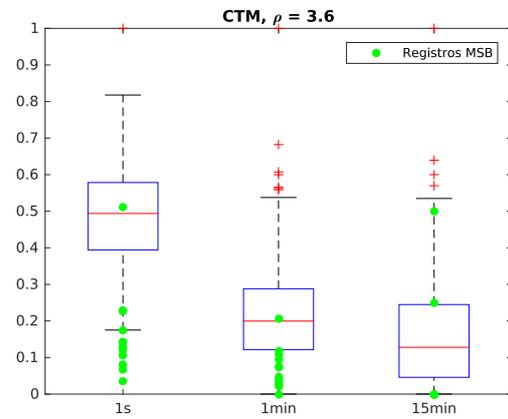


FIGURA C.48:  $CTM_{\rho=3,6}^y$ . Actividad.

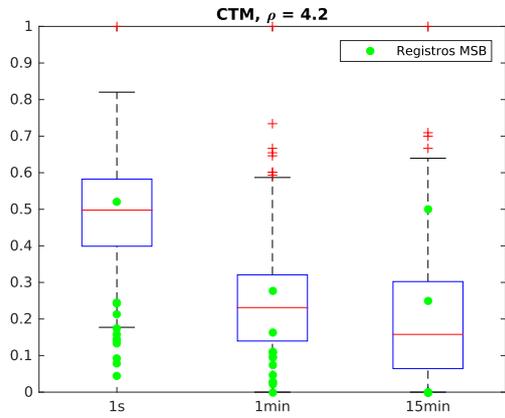


FIGURA C.49:  $CTM_{\rho=4,2}^y$ . Actividad.

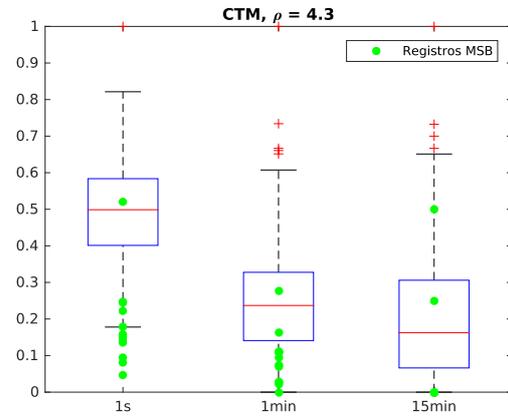


FIGURA C.50:  $CTM_{\rho=4,3}^y$ . Actividad.

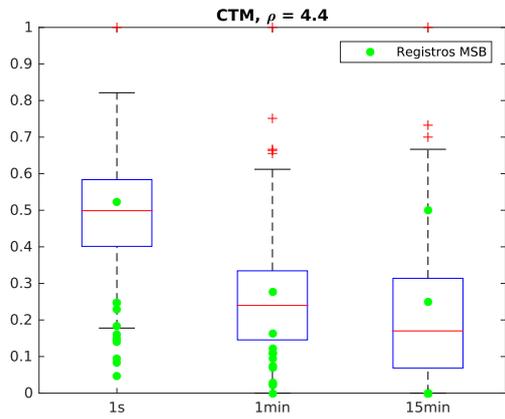


FIGURA C.51:  $CTM_{\rho=4,4}^y$ . Actividad.

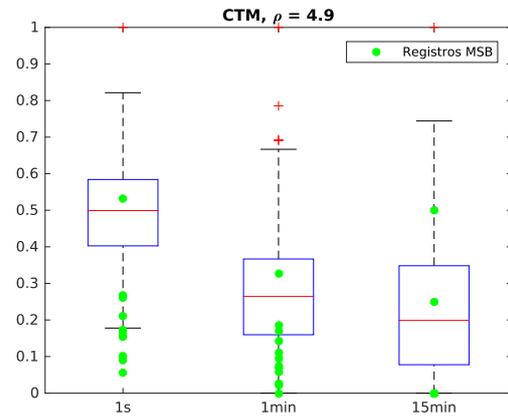


FIGURA C.52:  $CTM_{\rho=4,9}^y$ . Actividad.

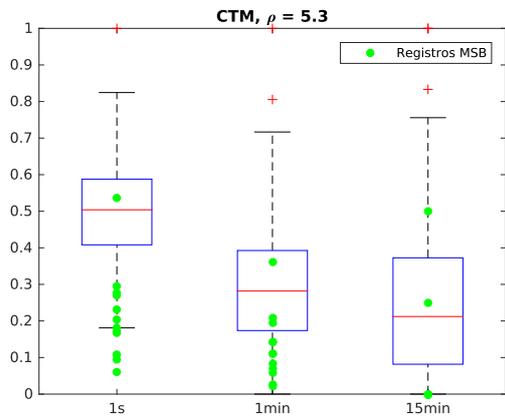


FIGURA C.53:  $CTM_{\rho=5,3}^y$ . Actividad.

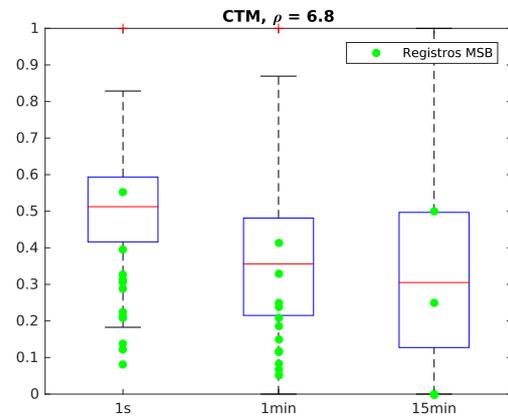


FIGURA C.54:  $CTM_{\rho=6,8}^y$ . Actividad.

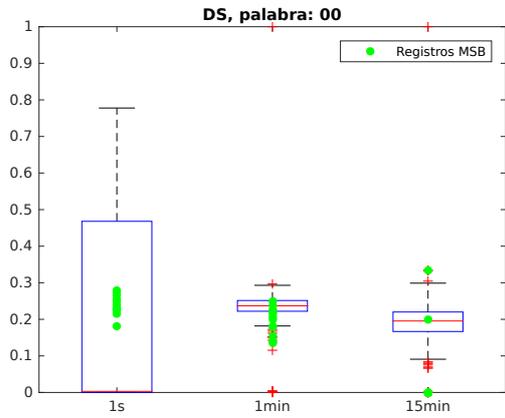


FIGURA C.55:  $P_{00}^y$ . Actividad.

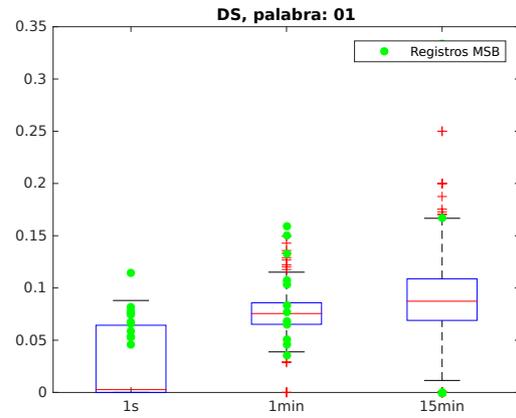


FIGURA C.56:  $P_{01}^y$ . Actividad.

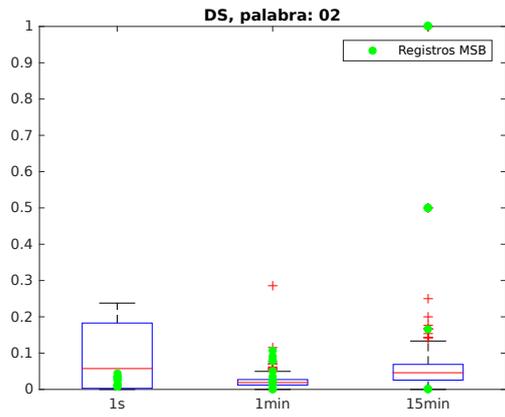


FIGURA C.57:  $P_{02}^y$ . Actividad.

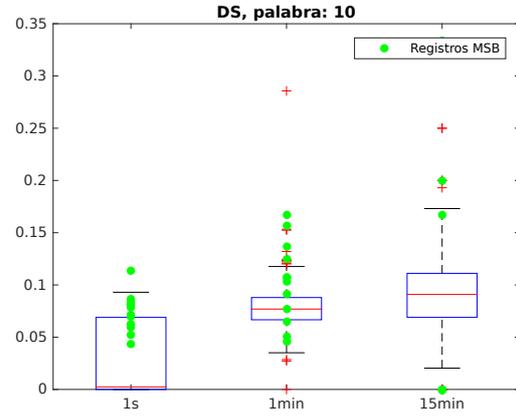


FIGURA C.58:  $P_{10}^y$ . Actividad.

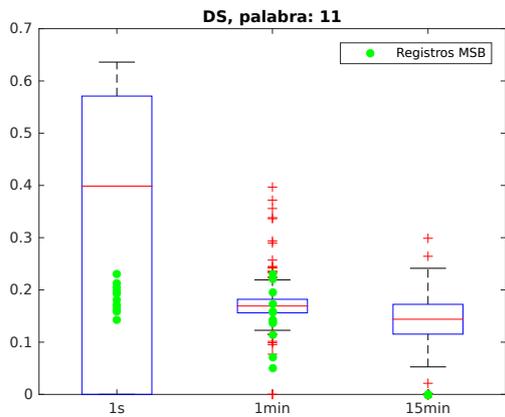


FIGURA C.59:  $P_{11}^y$ . Actividad.

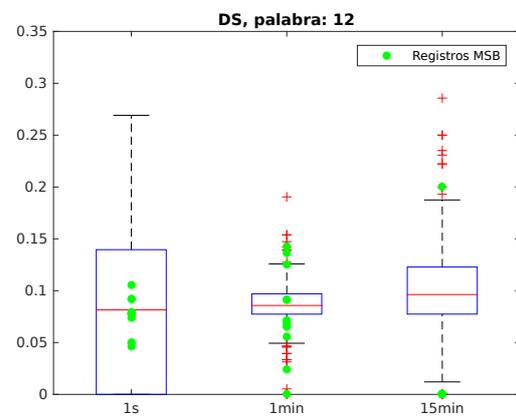


FIGURA C.60:  $P_{12}^y$ . Actividad.

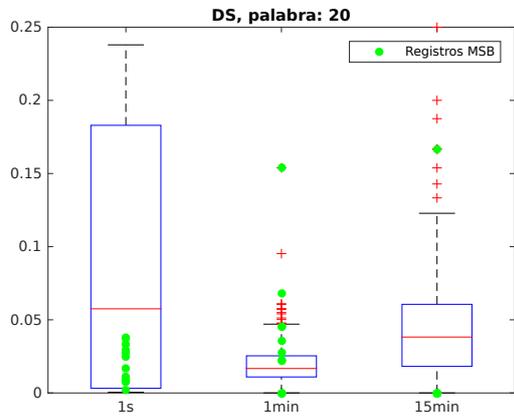


FIGURA C.61:  $P_{20}^y$ . Actividad.

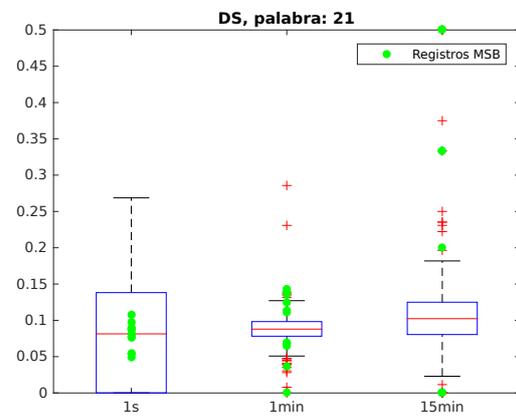


FIGURA C.62:  $P_{21}^y$ . Actividad.

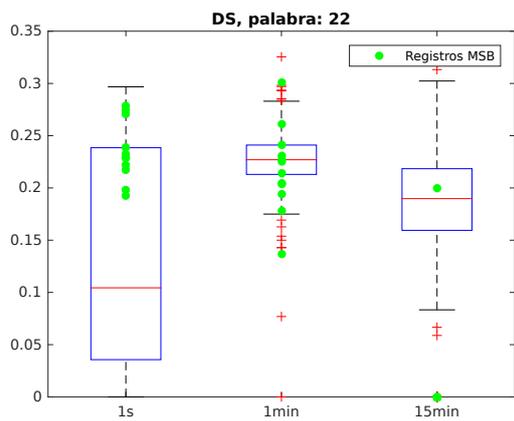


FIGURA C.63:  $P_{22}^y$ . Actividad.

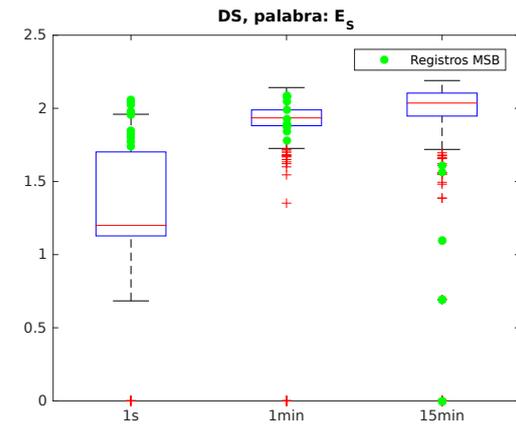


FIGURA C.64:  $E_{Shannon}^y$ . Actividad.

## C.3 CANAL Z

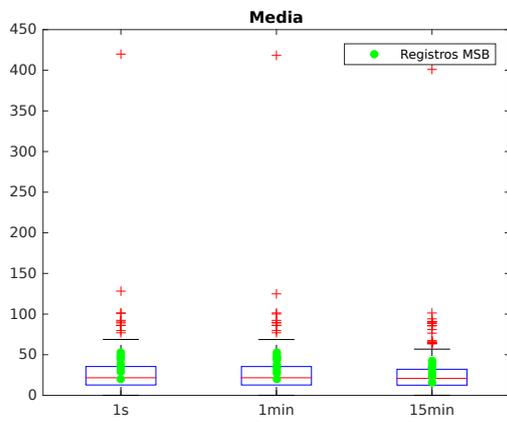


FIGURA C.65: Media. Canal z. Actividad.

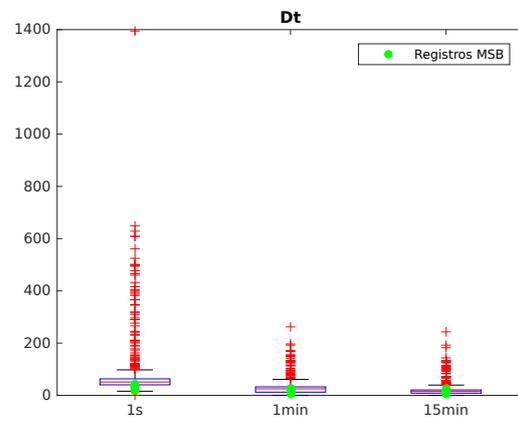


FIGURA C.66: Desviación estándar. Canal z. Actividad.

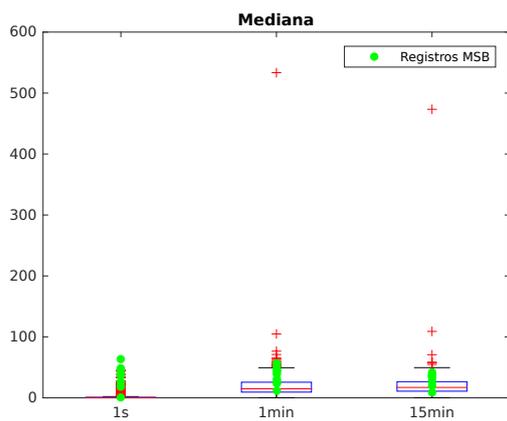


FIGURA C.67: Mediana. Canal z. Actividad.

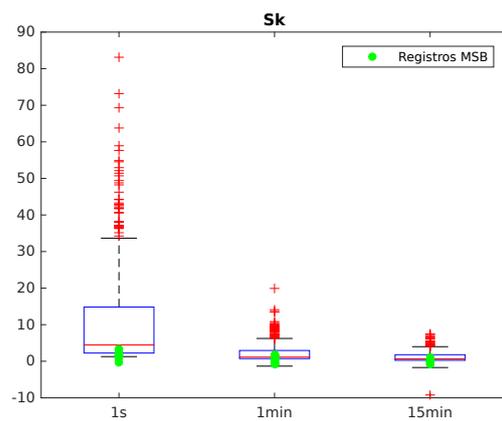


FIGURA C.68: Asimetría. Canal z. Actividad.

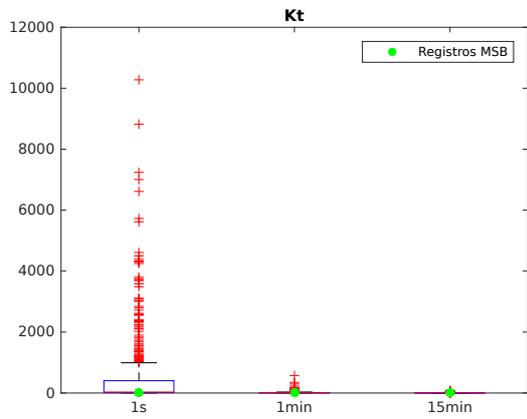


FIGURA C.69: Curtosis. Canal z. Actividad.

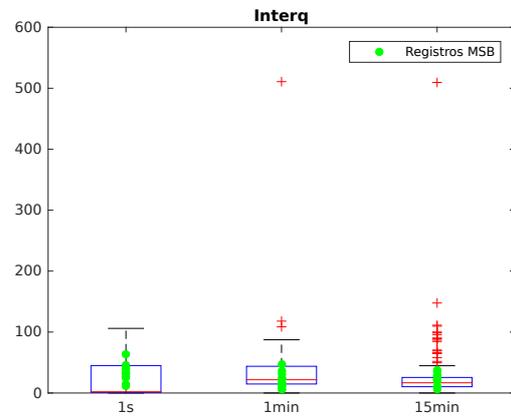


FIGURA C.70: Rango intercuartil. Canal z. Actividad.

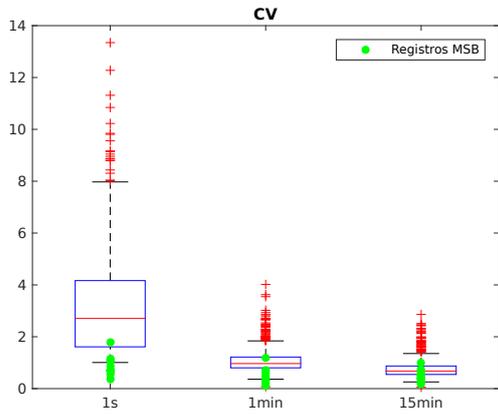


FIGURA C.71: Coeficiente de variación. Canal z. Actividad.

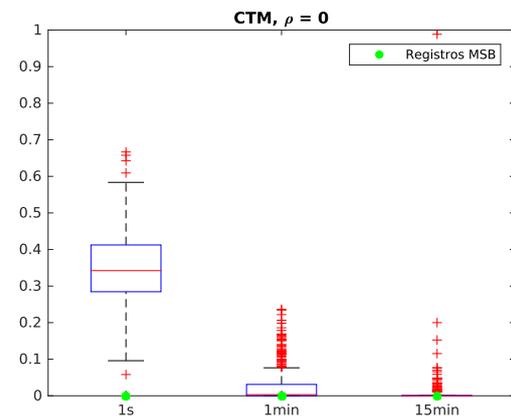


FIGURA C.72:  $CTM_{\rho=0}^z$ . Actividad.

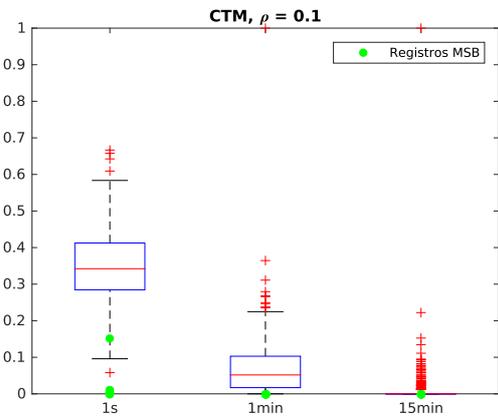


FIGURA C.73:  $CTM_{\rho=0,1}^z$ . Actividad.

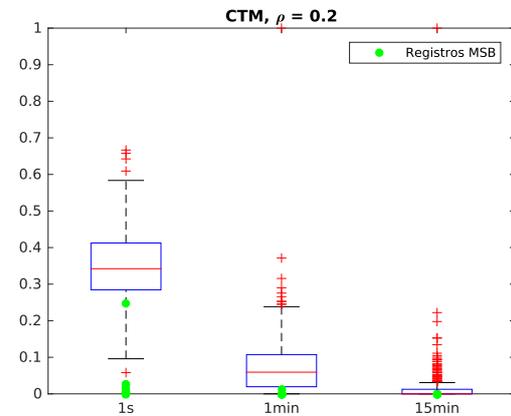


FIGURA C.74:  $CTM_{\rho=0,2}^z$ . Actividad.

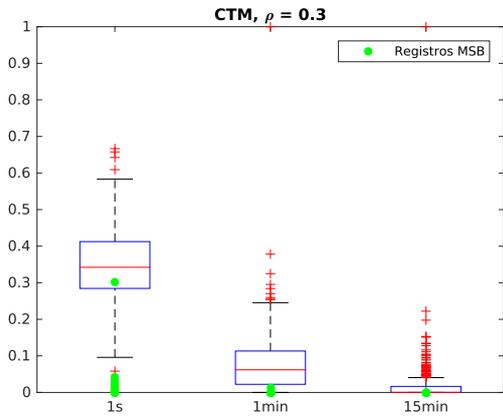


FIGURA C.75:  $CTM_{\rho=0,3}^z$ . Actividad.

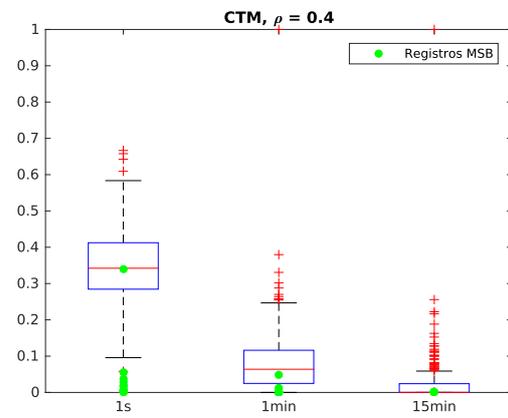


FIGURA C.76:  $CTM_{\rho=0,4}^z$ . Actividad.

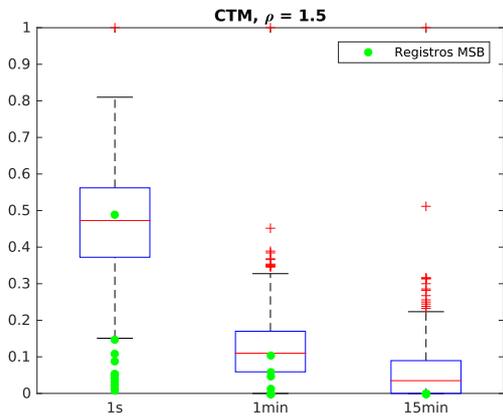


FIGURA C.77:  $CTM_{\rho=1,5}^z$ . Actividad.

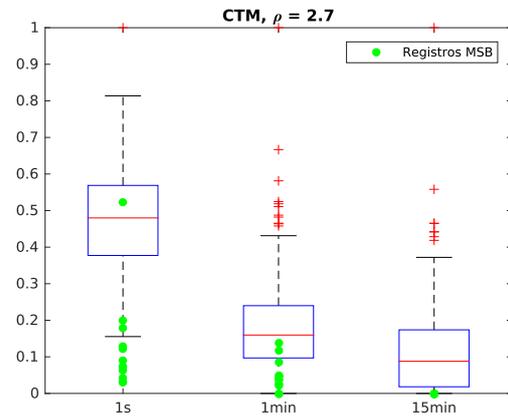


FIGURA C.78:  $CTM_{\rho=2,7}^z$ . Actividad.

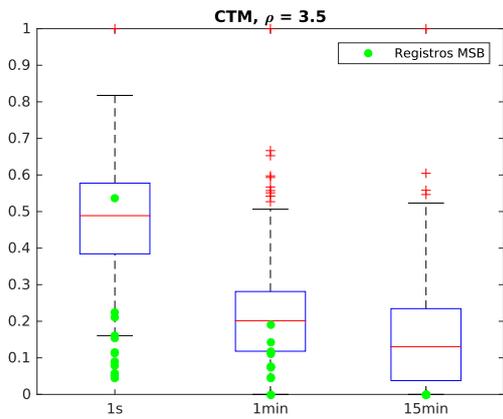


FIGURA C.79:  $CTM_{\rho=3,5}^z$ . Actividad.

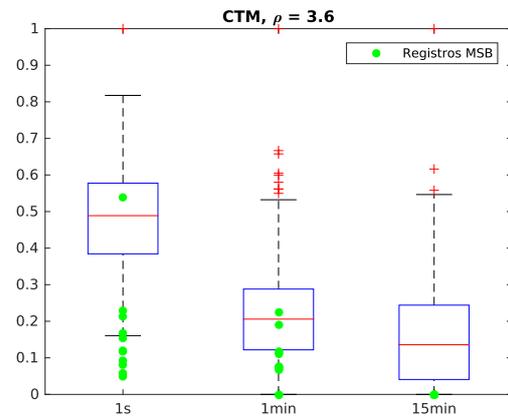


FIGURA C.80:  $CTM_{\rho=3,6}^z$ . Actividad.

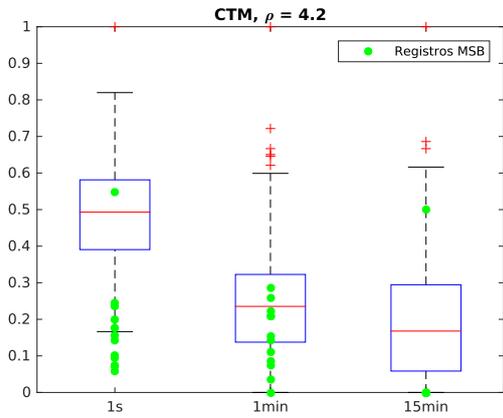


FIGURA C.81:  $CTM^z_{\rho=4,2}$ . Actividad.

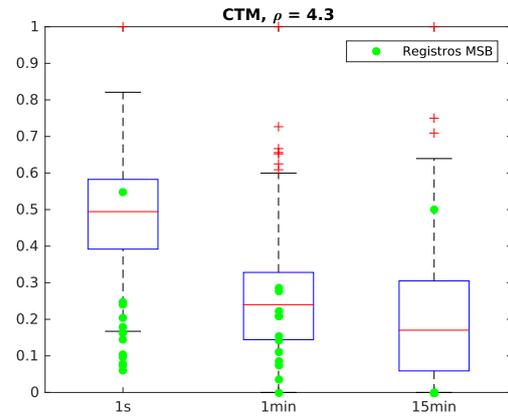


FIGURA C.82:  $CTM^z_{\rho=4,3}$ . Actividad.

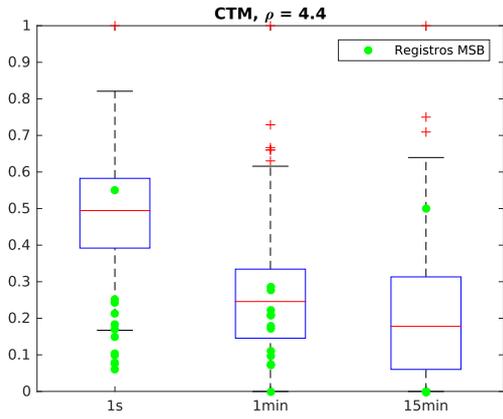


FIGURA C.83:  $CTM^z_{\rho=4,4}$ . Actividad.

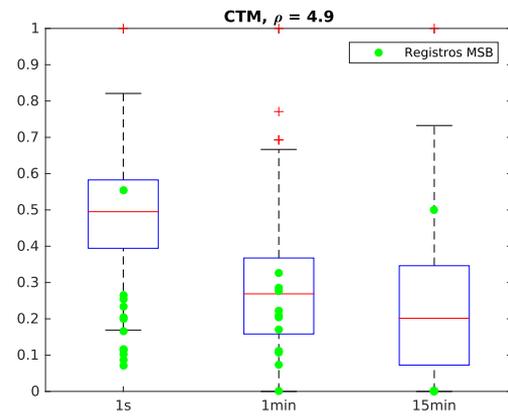


FIGURA C.84:  $CTM^z_{\rho=4,9}$ . Actividad.

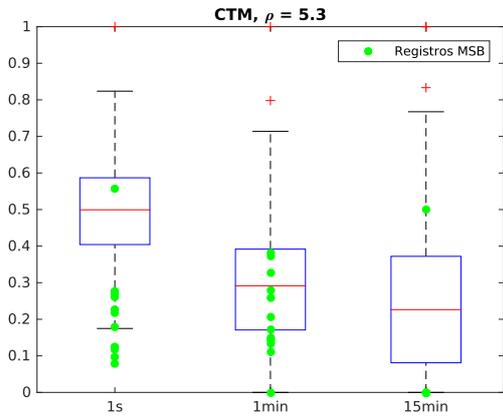


FIGURA C.85:  $CTM^z_{\rho=5,3}$ . Actividad.

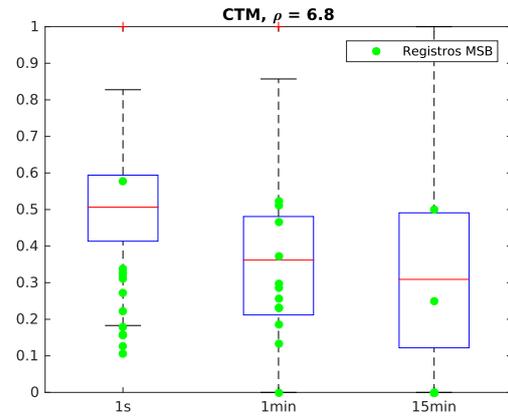


FIGURA C.86:  $CTM^z_{\rho=6,8}$ . Actividad.

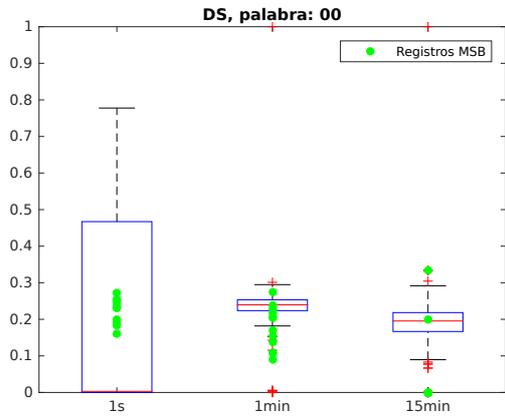


FIGURA C.87:  $P_{00}^z$ . Actividad.

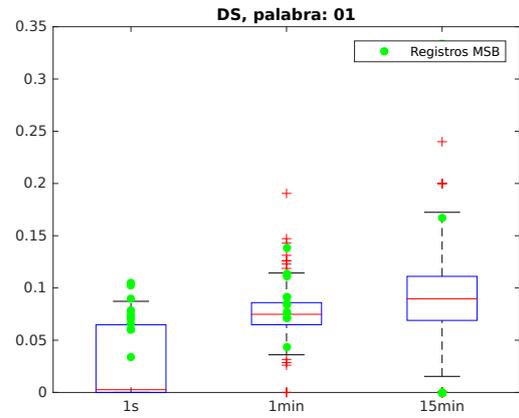


FIGURA C.88:  $P_{01}^z$ . Actividad.

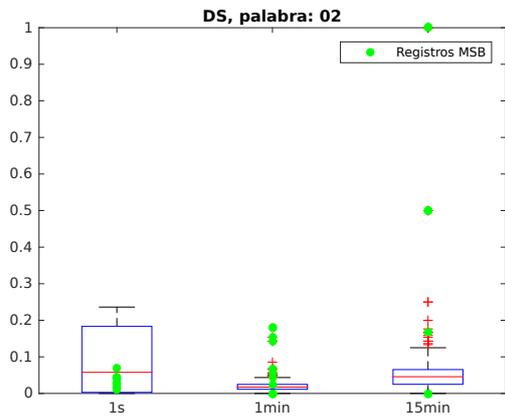


FIGURA C.89:  $P_{02}^z$ . Actividad.

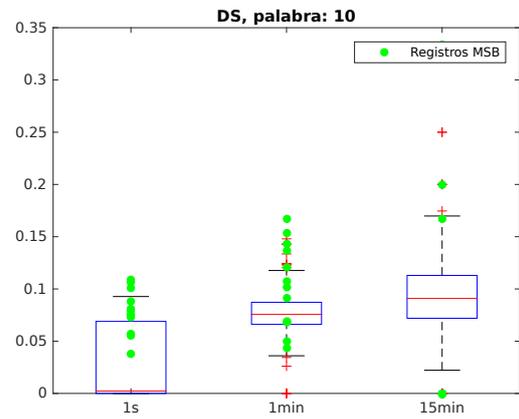


FIGURA C.90:  $P_{10}^z$ . Actividad.

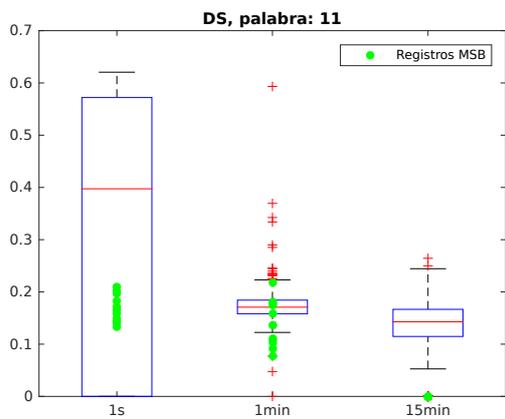


FIGURA C.91:  $P_{11}^z$ . Actividad.

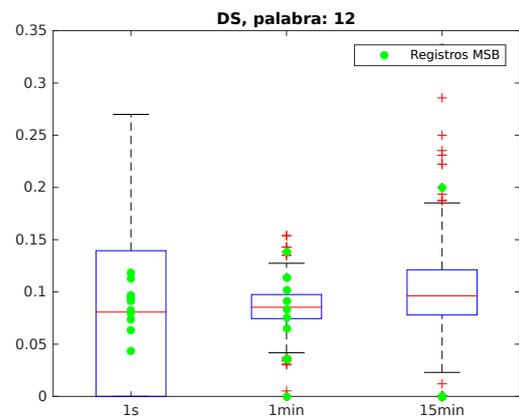


FIGURA C.92:  $P_{12}^z$ . Actividad.

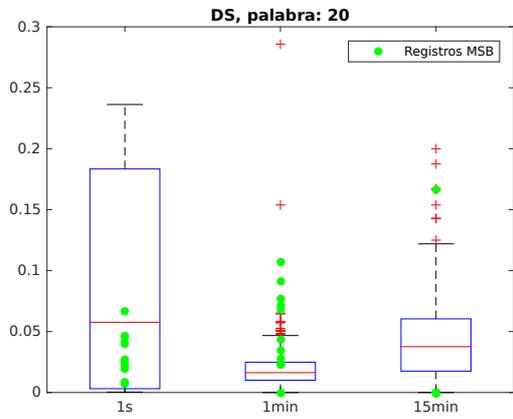


FIGURA C.93:  $P_{20}^z$ . Actividad.

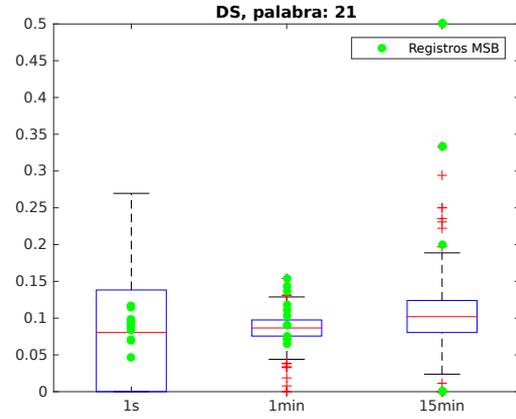


FIGURA C.94:  $P_{21}^z$ . Actividad.

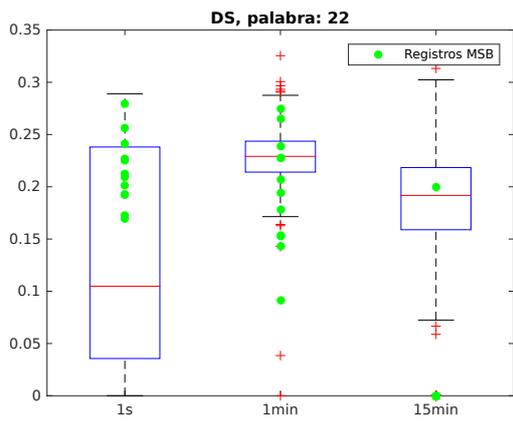


FIGURA C.95:  $P_{22}^z$ . Actividad.

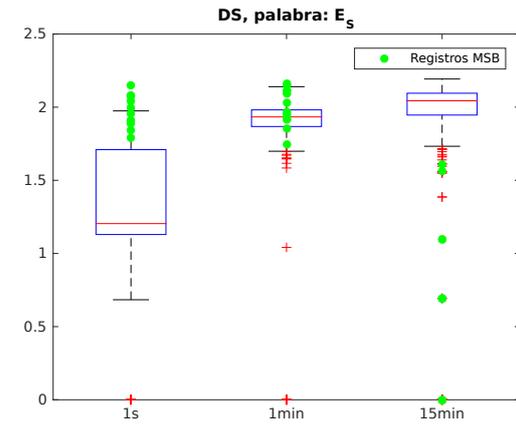


FIGURA C.96:  $E_{Shannon}^z$ . Actividad.