

Máster Universitario en Ingeniería Industrial

Trabajo Fin de Máster

Fusión de información de sistemas de localización para la
evaluación de rutinas de comportamiento de personas

ESCUELA POLITECNICA
SUPERIOR

Autor: Ismael Miranda Gordo

Tutora: Ana Jiménez Martín

Cotutor: David Gualda Gómez

2021

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

Máster Universitario en Ingeniería Industrial

Trabajo Fin de Máster

**Fusión de información de sistemas de localización para la
evaluación de rutinas de comportamiento de personas**

Autor: Ismael Miranda Gordo

Directores: Ana Jiménez Martín
David Gualda Gómez

Tribunal:

Presidente: Álvaro Hernández Alonso

Vocal 1º: Saturnino Maldonado Bascón

Vocal 2º: Ana Jiménez Martín

CALIFICACIÓN:

FECHA:

A mi familia, a mis amigos. . .

Agradecimientos

En primer lugar, agradecer a mis tutores Ana y David por ayudarme y apoyarme durante el inicio, desarrollo y final de este Trabajo de Fin de Máster. Agradecerles también el buen trato desde que comencé mi etapa en el grupo de investigación GEINTRA hasta el final. De igual manera, gracias a Juan Jesús por el apoyo y ayuda en el desarrollo de este trabajo. Por último, dar las gracias a todos los miembros del grupo de investigación con los que ha sido muy sencillo adaptarse y trabajar durante estos años.

Dar las gracias a todos los compañeros que me han acompañado en la etapa del máster que se cierra con este TFM. En especial a Jorge de Vicente por ser mi compañero en la mayoría de los trabajos, sin su ayuda esto no hubiera sido posible. Gracias a Sergio, Cris y Ángel por apoyarme desde el primer día que nos conocimos en la carrera hasta hoy en el que se acaba este largo proceso.

A mis amigos, Sebas, Raquel, Román, Laura, Leti, Edu y Marina, por ser mi apoyo día a día, en las situaciones fáciles y las no tan fáciles, por hacerme crecer en todos los aspectos, gracias.

Por último, agradecer a mi hermana y a mis padres por el apoyo incondicional que he tenido siempre. Por aguantarme en mis peores momentos de este camino y creer en mí ni cuando yo mismo lo hacía.

En definitiva, agradecer a toda la gente que ha estado durante este largo y duro camino, sin duda, sin ellos este trabajo no habría sido posible.

Resumen

Este trabajo de Fin de Máster se centra en la fusión de información de sistemas de localización para la evaluación de rutinas de comportamiento. Como tecnologías base para los sistemas que se desarrollan en este trabajo se van a utilizar dos tecnologías de radiofrecuencia, WiFi y *Bluetooth*. Además, se va a utilizar la tecnología inercial para la fusión de información con el objetivo de mejorar los resultados de localización. A partir de la localización obtenida de la fusión sensorial se van a evaluar las posibles rutinas de comportamiento monitorizando a un usuario dentro de un entorno controlado.

Palabras Clave: Sistemas de localización, WiFi, *Bluetooth*, fusión sensorial, rutinas de comportamiento.

Abstract

This Master's thesis focuses on the fusion of information from localization systems for the evaluation of behavioural routines. Two radio frequency technologies, WiFi and Bluetooth, will be used as base technologies for the systems developed in this work. In addition, inertial technology will be used for the fusion of information with the aim of improving the localization results. Based on the localization obtained from the sensory fusion, possible behavioural routines will be evaluated by monitoring a user within a controlled environment.

Keywords: Localization systems, WiFi, Bluetooth, sensory fusion, behavioural routines.

Resumen extendido

En este Trabajo de Fin de Máster (TFM) se presenta el desarrollo de dos sistemas de localización utilizando como base tecnologías de radiofrecuencia (*WiFi* y *Bluetooth*) y empleando la tecnología inercial como complemento para mejorar los resultados de ambos sistemas. A partir de la localización obtenida por ambos sistemas se realiza la evaluación de rutinas de comportamiento. Para ello se han utilizado dispositivos comerciales tanto para el receptor, con interfaz WiFi o Bluetooth y una unidad de medida inercial (IMU), como los emisores de radiofrecuencia desplegados en las habitaciones a monitorizar.

Desde el punto de vista de la algoritmia se distinguen tres grandes bloques: el bloque de localización simbólica mediante WiFi, el bloque de localización simbólica mediante *Bluetooth* y el bloque de generación de rutinas de comportamiento. La localización simbólica mediante WiFi se basa en el uso del porcentaje de conexión mientras que la localización simbólica mediante *Bluetooth* se basa en el uso de técnicas de *Machine Learning*. En ambos bloques se emplea la fusión sensorial con la tecnología inercial para la mejora de los resultados. El bloque de generación de rutinas emplea los resultados obtenidos por ambos sistemas de localización con el objetivo de detectar y evaluar rutinas de comportamiento de personas.

Para ambos sistemas se realizan tres tipos de pruebas: el primer tipo de pruebas consiste en las pruebas de calibración, cuyo objetivo es configurar los parámetros del sistema con el fin de obtener los mejores resultados. El segundo tipo consiste en las pruebas validación, donde se determina si el sistema funciona correctamente dentro de un entorno real y, puede utilizarse para la evaluación de rutinas de comportamiento. Por último, el tercer tipo de pruebas consiste en la evaluación de rutinas de comportamiento a partir de la monitorización de uno o varios usuarios dentro de un entorno controlado.

Palabras Clave: Sistemas de localización, WiFi, *Bluetooth*, fusión sensorial, rutinas de comportamiento.

Índice

<i>Resumen</i>	<i>ix</i>
<i>Abstract</i>	<i>xii</i>
<i>Resumen extendido</i>	<i>xiv</i>
<i>Índice</i>	<i>xv</i>
<i>Índice de figuras</i>	<i>xix</i>
<i>Índice de tablas</i>	<i>xxi</i>
<i>Glosario de acrónimos y símbolos</i>	<i>xxiii</i>
1. Introducción	1
1.1 Contexto del Trabajo Fin de Máster	1
1.2 Objetivos	2
1.3 Estructura del trabajo	2
2. Conceptos teóricos	3
2.1 Métodos de Machine Learning	3
2.2 Sistemas de Localización	7
2.2.1 RF	8
2.2.2 Inercial	11
2.3 Fusión de datos/información	12
3. Descripción del material utilizado	17
3.1 Sistema WiFi	17
3.2 Sistema Bluetooth	18

4.	<i>Algoritmos propuestos.....</i>	22
4.1	<i>Localización simbólica con tecnología WiFi.....</i>	22
4.2	<i>Localización simbólica con tecnología Bluetooth.....</i>	29
4.3	<i>Generación de rutinas.....</i>	31
5.	<i>Resultados.....</i>	35
5.1	<i>Localización y generación de rutinas a partir de señales WiFi</i> <i>35</i>	
5.1.1	<i>Pruebas calibración.....</i>	35
5.1.2	<i>Pruebas de evaluación del sistema WiFi en un entorno</i> <i>real 42</i>	
5.1.3	<i>Pruebas de evaluación de rutina de comportamiento....</i>	47
5.2	<i>Localización y generación de rutinas a partir de señales</i> <i>Bluetooth.....</i>	51
5.2.1	<i>Pruebas de calibración del sistema.....</i>	51
5.2.2	<i>Pruebas de validación del sistema.....</i>	57
5.2.3	<i>Pruebas de evaluación de rutinas de comportamiento..</i>	59
6.	<i>Conclusiones y trabajos futuros.....</i>	64
7.	<i>Planos.....</i>	67
7.1	<i>Sistema WiFi</i>	67
7.1.1	<i>Localización</i>	67
7.1.2	<i>Generación de rutinas.....</i>	76
7.2	<i>Sistema Bluetooth.....</i>	77
7.2.1	<i>Localización</i>	77
7.2.2	<i>Generación de rutinas.....</i>	79

8.	<i>Pliego de condiciones</i>	82
	8.1 <i>Pliego de condiciones generales</i>	82
	8.2 <i>Pliego de condiciones particulares</i>	82
9.	<i>Presupuesto</i>	84
10.	<i>Bibliografía</i>	86

Índice de figuras

Figura 1. Ejemplo de árbol de decisión [8].	7
Figura 2. Precisión en función de la tecnología utilizada [12].	8
Figura 3. Receptor del sistema propuesto, NGIMU [28].	17
Figura 4. Emisor router propuesto de TP-LINK (TL-WR802N) [29].	18
Figura 5. Pantalla principal de la aplicación GetSensorData [31].	19
Figura 6. Formato de salida de la aplicación GetSensorData [31].	19
Figura 7. Pantalla de configuración de la aplicación GetSensorData [31].	20
Figura 8. Baliza IBKS 105 [32] (derecha) y baliza IBKS Plus [33] (izquierda).	21
Figura 9. Diagrama de bloques de los algoritmos utilizados para la localización WiFi.	22
Figura 10. Diagrama de bloques del algoritmo de detección de pasos.	24
Figura 11. Diagrama de bloques del algoritmo de cambio de planta.	26
Figura 12. Ángulo Pitch al realizar una transición por las escaleras.	26
Figura 13. Ángulo Pitch al realizar una transición por el ascensor.	26
Figura 14. Detección del tipo de cambio de planta a partir del número de ciclos de la señal.	27
Figura 15. Detección del tipo de cambio de planta a partir de la desviación estándar de la señal.	27
Figura 16. Diagrama de bloques del algoritmo de localización simbólica.	28
Figura 17. Diagrama de bloques del algoritmo de fusión.	29
Figura 18. Diagrama de bloques de los algoritmos desarrollados para el sistema Bluetooth.	29
Figura 19. Diagrama de bloques del algoritmo de fusión sensorial.	31
Figura 20. Diagrama de bloques de la generación de rutinas empleando la localización WiFi.	31
Figura 21. Diagrama de bloques de la generación de rutinas empleando la localización Bluetooth.	32
Figura 22. Generación del vector de referencia.	33
Figura 23. Plano del pasillo donde se realizan las pruebas de umbral. En rojo la posición del emisor y en azul el receptor y su recorrido.	36
Figura 24. Número de zancadas para una potencia de emisión baja.	37
Figura 25. Radios de cobertura de porcentaje de conexión al 100 % de aproximadamente 4.22 m para una potencia baja, 4.37 m para una potencia media y 4.44 para una potencia alta de emisión.	38
Figura 26. Resultados de 4 pruebas con el router colocado en la planta 2 y con potencia de emisión baja.	39
Figura 27. Resultados de 4 pruebas con ambos router configurados en potencia media.	40
Figura 28. Posición de los router en las pruebas B de interferencia en el hall de la 2ª planta del ala oeste de la EPS de la UAH.	41
Figura 29. Pruebas B de interferencia realizadas con los router colocados en las posiciones 1 y 3 del hall de la 2ª planta del ala oeste de la EPS de la UAH.	41

Figura 30. Nuevas posiciones de los router en las pruebas B de interferencia dentro del hall de la 2º planta del ala oeste de la EPS de la UAH.....	42
Figura 31. Ubicación de cada una de las estancias monitorizadas en el centro sociosanitario Las Palmeras.....	43
Figura 32. Número de pasos, porcentaje de conexión y presión frente al ground truth.....	44
Figura 33. Número de planta y habitación frente al ground truth.....	45
Figura 34. Datos representativos de la ocupación de las diferentes estancias a lo largo de las diferentes mañanas.....	46
Figura 35. Resultados de la localización del voluntario 1.....	48
Figura 36. Coeficiente de reproducibilidad para cada uno de los días del voluntario 1.....	49
Figura 37. Resultados de la localización simbólica del voluntario 2.....	49
Figura 38. Coeficientes de reproducibilidad del voluntario 2.....	50
Figura 39. Habitaciones del domicilio y posición de las balizas.....	51
Figura 40. Resultados de una de las pruebas con nivel de potencia de emisión de -8 dBm.....	52
Figura 41. Resultados de una de las pruebas de 20 segundos.....	53
Figura 42. Resultado de una de las pruebas con el receptor situado en la sujeción del brazo.....	54
Figura 43. Resultados de una de las pruebas de larga duración.....	56
Figura 44. Resultados del sistema utilizando el algoritmo de fusión sensorial.....	56
Figura 45. Mapa del nuevo entorno experimental y de la localización de las balizas.....	57
Figura 46. Matrices de confusión con KNN. Izquierda: sin considerar la aceleración; Derecha: considerando la aceleración.....	58
Figura 47. Resultados del algoritmo de fusión durante 10 días para tecnología BLE.....	59
Figura 48. Coeficiente de reproducibilidad para cada uno de los días.....	60
Figura 49. Coeficiente de similitud para cada uno de los días.....	61
Figura 50. Coeficiente de reproducibilidad para el nuevo vector de referencia.....	62
Figura 51. Coeficiente de similitud para el nuevo orden de referencia.....	63

Índice de tablas

Tabla 1. Número de zancadas y distancia de cada zancada para cada uno de los recorridos	36
Tabla 2. Distancia para cada uno de los modos de potencia.....	37
Tabla 3. Resultados de las pruebas A1.	39
Tabla 4. Resultados de las pruebas A2.	40
<i>Tabla 5. Resultados de las pruebas B.</i>	<i>41</i>
Tabla 6. Resultados obtenidos para las nuevas posiciones de los router.....	42
Tabla 7. Vector de referencia del voluntario 1.	48
Tabla 8. Vector de referencia voluntario 2	50
Tabla 9. Exactitud de los modelos a partir de los datos de entrenamiento.	52
Tabla 10. Exactitud de los modelos frente a diferentes niveles de potencia.	53
Tabla 11. Exactitud de las pruebas de tiempo.....	54
Tabla 12. Exactitud frente a la posición del receptor.	54
Tabla 13. Comparación de los modelos individuales frente al modelo completo. ...	55
Tabla 14. Exactitud de los clasificadores en las pruebas de validación del sistema.	58
Tabla 15. Métricas de la localización a nivel de habitación.....	58
Tabla 16. Vector de referencia.....	60
Tabla 17. Nuevo vector de referencia.	62
Tabla 18. Costes directos del proyecto.....	84
Tabla 19. Costes del material utilizado en el proyecto.....	84
Tabla 20. Costes indirectos del proyecto.....	84
Tabla 21. Costes totales del proyecto.....	85

Glosario de acrónimos y símbolos

IMU	Unidad de medida inercial (<i>Inertial Measurement Unit</i>).
ML	Aprendizaje automático (<i>Machine Learning</i>).
PCA	Análisis de componentes principales (<i>Principal Component Analysis</i>).
KNN	K vecinos más cercanos (<i>K Nearest Neighbours</i>).
SVM	Máquina de vector soporte (<i>Support Vector Machine</i>).
DT	Árbol de decisión (<i>Decision Tree</i>).
GNSS	Sistemas globales de navegación por satélite (<i>Global Navegation Satelite System</i>).
IPS	Sistema de posicionamiento en interiores (<i>Indoor Positioning System</i>).
RF	Radiofrecuencia.
VLC	Comunicación de luz visible (<i>Visible Light Communication</i>).
AoA	Ángulo de llegada (<i>Angle of Arrival</i>).
ToA	Tiempo de llegada (<i>Time of Arrival</i>).
CSI	Información del estado del canal (<i>Channel Status Information</i>).
RSSI	Potencia de la señal recibida por el receptor (<i>Receptor Signal Strength Information</i>).
EKF	Filtro de Kalman extendido (<i>Extenden Kalman Filter</i>).
SSID	Identificador de conjunto de servicios (<i>Service Set Identifier</i>).
BLE	Bluetooth de baja energía (<i>Bluetooth Low Energy</i>).
MAC	Identificador de control de acceso (<i>Media Access Control</i>).
ZCT	Técnica de cruce por cero (<i>Zero Cross Technique</i>).
FMM	Filtro de media móvil.
STD	Desviación estándar (<i>Standard Desviation</i>).
EPS	Escuela Politécnica Superior.

1. Introducción

1.1 Contexto del Trabajo Fin de Máster

Actualmente, una línea de investigación muy activa de los sistemas de localización y posicionamiento es la monitorización del comportamiento de personas o robots. La monitorización tiene objetivos muy diferentes como son la reducción del consumo de energía, el conocimiento de patrones de trabajo para mejorar la productividad, la determinación de cuáles son las estancias más visitadas dentro de un hogar y cuáles son las conexiones entre habitaciones más habituales, o la identificación de desviaciones en la rutina de comportamiento de personas mayores, entre muchos otros. Por ello la monitorización con el objetivo de obtener rutinas de comportamiento tiene multitud de aplicaciones en ámbitos muy diferentes, desde del ámbito de la eficiencia energética hasta llegar al ámbito de la salud.

La monitorización del comportamiento se realiza generalmente mediante el uso de sistemas de posicionamiento en interiores. Una de las tecnologías de base más utilizadas para los sistemas de posicionamiento en interiores es la radiofrecuencia. Utilizando como base la tecnología de radiofrecuencia se pueden conseguir desde precisiones en la localización centimétricas (*Ultrawideband*) hasta precisiones a nivel de habitación (*WiFi* y *Bluetooth*). La precisión del sistema está relacionada con el coste del sistema, por lo que este trabajo se va a centrar en el uso de la tecnología *WiFi* y *Bluetooth* al tratarse de tecnología de bajo coste.

Otra tecnología muy utilizada actualmente es la basada en acelerómetros como ocurre en el caso de los altímetros. Además, con el fin de mejorar los sistemas de posicionamiento basado en las tecnologías de radiofrecuencia se utilizan las unidades de medida inercial (IMU) para poder información sobre la actividad física y el movimiento. El uso de las tecnologías inerciales combinadas con las tecnologías de radiofrecuencia permite mejorar la precisión de la localización.

En cualquiera de los casos, parece que el uso de una única tecnología no puede proporcionar información precisa para generar patrones de comportamiento por lo que todas las soluciones actuales se basan en la fusión sensorial. En este sentido este trabajo propone la fusión de diferentes tecnologías (Radiofrecuencia e inercial) y que, realizando un análisis a largo plazo sobre una persona o robot móvil monitorizado, se pueda proporcionar información sobre las rutinas de comportamiento realizadas y los posibles cambios en éstas aplicando técnicas de aprendizaje automático.

1.2 Objetivos

El principal objetivo del proyecto es el desarrollo de los algoritmos basados en técnicas de aprendizaje automático para la detección y evaluación de rutinas de comportamiento empleando como herramienta la localización en interiores. Las tecnologías que se van a utilizar para la localización en interiores son la tecnología WiFi y *Bluetooth*. Para llegar a cumplir el objetivo principal de este trabajo, el proyecto se divide en una serie de objetivos parciales:

- Obtención de la localización contextual empleando la tecnología WiFi y *Bluetooth*.
- Fusión de resultados de localización con la IMU para mejorar la precisión de los resultados.
- Obtención de rutinas de comportamiento a partir de los resultados de localización obtenidos aplicando en algunos casos técnicas de *Machine Learning*.

1.3 Estructura del trabajo

Este documento está dividido en dos bloques principales, en el primero que está dividido en 6 capítulos se desarrolla la memoria del trabajo y el segundo que está dividido en 4 capítulos en el que se describen otros aspectos importantes del proyecto.

En el capítulo 1, dentro del que se incluye este apartado, se describe el contexto en el que se encuentra ubicado este trabajo, los objetivos y la estructura de este; En el capítulo 2 se desarrollan los conceptos teóricos necesarios para la realización del proyecto que incluyen técnicas de *Machine Learning*, los sistemas de localización empleando como tecnología base la radiofrecuencia y los sistemas inerciales; En el capítulo 3 se describe el *Hardware* que se utiliza en el proyecto; En el capítulo 4 se describen todos los algoritmos desarrollados para los sistemas de localización de este trabajo; En el capítulo 5 se evalúa tanto el *Hardware* utilizado como los algoritmos propuestos con diferentes pruebas experimentales. Por último, en el capítulo 6 se incluyen las principales conclusiones del trabajo realizado y el trabajo futuro que se propone.

Fuera de la memoria del trabajo se incluyen el capítulo 7 (Planos) en el que se adjuntan los fragmentos más importantes del código desarrollado, el capítulo 8 (Pliego de condiciones) en el que se incluyen las condiciones para el correcto funcionamiento del sistema, el capítulo 9 (Presupuesto) en el que se incluye el coste del trabajo y por último, el capítulo 10 en el que se muestra la bibliografía empleada.

2. Conceptos teóricos

En este apartado se van a desarrollar los conceptos teóricos necesarios para el desarrollo del proyecto.

2.1 Métodos de Machine Learning

Por definición, *Machine Learning* (Aprendizaje automático) es la ciencia que permite que las computadoras aprendan y actúen de forma similar a como lo haría un humano, para ello es necesario ir mejorando su aprendizaje a lo largo del tiempo de una forma autónoma, entrenando los algoritmos con datos e información en forma de observaciones e interacciones con el mundo real [1].

Las técnicas de *Machine Learning* (ML) ofrecen una manera eficiente de obtener conocimiento mediante la información contenida en los datos, mejorando de forma gradual el rendimiento de los modelos predictivos y tomando decisiones basadas en esos datos.

En *Machine Learning*, una de las alternativas de representación ampliamente utilizada son las matrices y notaciones vectoriales para referirnos a los datos:

- Cada una de las filas de la matriz es una muestra, observación o dato puntual.
- Cada columna es una característica de la muestra, observación o dato puntual anterior.
- La última columna es el objetivo, etiqueta o respuesta, el cual es el valor que se pretende predecir, si se dispone de él.

En la expresión (1) se muestra la estructura de las matrices de datos descrita previamente. N es el número de características X , m el número de filas de la matriz e Y la característica de cada una de las filas.

$$\begin{pmatrix} X_{11} & \cdots & X_{n1} & Y_1 \\ \vdots & \ddots & \vdots & \vdots \\ X_{1m} & \cdots & X_{nm} & Y_m \end{pmatrix} \quad (1)$$

Los distintos algoritmos basados en *Machine Learning* podrían agruparse en cuatro tipos diferentes: aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje profundo o *Deep Learning* y el aprendizaje por refuerzo. A continuación se describen brevemente cada uno de ellos:

- **Aprendizaje supervisado:** consiste en un tipo de modelos de Machine Learning que se entrenan con un conjunto de ejemplos en los que los resultados de salida son conocidos. Los modelos aprenden de esos resultados conocidos y realizan ajustes en sus parámetros internos, o hiperparámetros, para adaptarse a los datos de entrada.

Una vez el modelo es entrenado, y los hiperparámetros son coherentes con los datos de entrada y los resultados de dos datos de entrenamiento,

el modelo puede ser utilizado para realizar predicciones de nuevos datos. Dentro del aprendizaje supervisado se distinguen dos aplicaciones principales, la clasificación y la regresión:

- **Clasificación:** es un tipo de aprendizaje supervisado en la que el objetivo es realizar predicciones de los nuevos datos asignándoles una etiqueta discreta. Existen dos tipos de clasificación que son la clasificación binaria en la que las etiquetas pueden ser 1 o 0 y la clasificación multiclase en la que el número de etiquetas puede ser mayor.
 - **Regresión:** consiste en un tipo de aprendizaje en el que se tienen un número de variables predictoras (características) y una variable de respuesta que en este caso no es discreta, sino que es continua. La variable de respuesta consiste en una relación de las variables predictoras que proporcionen un resultado continuo.
- **Aprendizaje no supervisado:** consiste en el uso de datos sin etiquetar cuya estructura es desconocida teniendo como objetivo la extracción de información significativa mediante la exploración de la estructura de dichos datos. Dentro del aprendizaje no supervisado existen dos categorías principales, agrupamiento y reducción dimensional.
 - **Agrupamiento:** se utiliza para organizar la información en grupos sin tener conocimiento previo de su estructura. Cada grupo que se forma es un conjunto de objetos que se diferencia de los objetos de otros grupos. En la mayor parte de algoritmos de agrupamiento se debe determinar el número de grupos a formar como puede ser el algoritmo *k-medoids* [2] y el *k-means* [3]. En otros algoritmos como puede ser el *Affinity propagation clustering* [4] no es necesario determinar el número de grupos a formar.
 - **Reducción dimensional:** se utiliza para reducir el número de características de cada una de las observaciones. La reducción dimensional consiste en encontrar correlaciones entre las características, ya que alguna de estas puede explicarse parcialmente con otras. Un método de reducción dimensional muy utilizado es el *Principal Component Analysis* (PCA) [5].
 - **Aprendizaje profundo o *Deep Learning*:** es un subcampo del *Machine Learning* que utiliza una estructura jerárquica de redes neuronales artificiales, que se construyen de una forma similar a la estructura neuronal del cerebro humano. La primera capa de la red neuronal toma datos en bruto, los procesa, extrae información y los transfiere a la siguiente capa. Este proceso se repite hasta la capa final donde se obtiene la predicción.
 - **Aprendizaje por refuerzo:** se trata de un área del aprendizaje automático [6]. Su principal particularidad es que es capaz de funcionar sin grandes cantidades de datos de entrenamiento. Tan solo necesita una

serie de indicaciones para ir aprendiendo a partir de un proceso de prueba y error. A diferencia del aprendizaje supervisado basado en un conjunto de datos que le indica a la máquina qué debe hacer, aquí se utilizan recompensas para reforzar el comportamiento deseado.

En un sistema de aprendizaje por refuerzo, un agente explora un entorno desconocido y determina las acciones a llevar a cabo mediante prueba y error. El sistema será capaz de aprender por sí sólo obteniendo recompensas o penalizaciones, no sólo de manera inmediata sino buscando maximizar la recompensa a la larga.

En este trabajo nos vamos a centrar en los métodos de clasificación de aprendizaje supervisado que se van a utilizar para estimar la localización. Todos los modelos de clasificación que se van a utilizar son métodos no paramétricos. Un modelo paramétrico es un modelo que resume los datos de entrenamiento con un conjunto de parámetros de tamaño fijo (Independientemente del volumen de datos de entrenamiento utilizados). Los algoritmos paramétricos son aquellos que optimizan la función a una forma conocida por el contrario, los algoritmos no paramétricos no presuponen una forma concreta en el modelo a generar. Los métodos de clasificación que se van a utilizar son el *K-Nearest-Neighbours* (KNN), el *Support Vector Machine* (SVM) y el *Decision tree* (DT):

- **KNN:** el algoritmo KNN [7] es un tipo de método de clasificación supervisado cuya lógica es muy sencilla, ya que al entrar un nuevo vector se obtiene la distancia a cada uno de los vectores de entrenamiento almacenados. De las distancias calculadas se seleccionan las K más cercanas y en base a esto se realiza la clasificación.

Una de las medidas de distancia más utilizadas es la distancia euclídea ($d(\mathbf{x}_i, \mathbf{x}_j)$) [8], que se muestra en la ecuación (2), donde los vectores \mathbf{x}_i y \mathbf{x}_j son respectivamente el vector i incluido en los datos de entrenamiento y el vector j que se trata del vector de entrada al algoritmo respecto al cual se va a calcular la distancia; x_{ri} y x_{rj} se corresponden con la característica r de los vectores \mathbf{x}_i y \mathbf{x}_j ; y P representa el número de características de los vectores \mathbf{x}_i y \mathbf{x}_j .

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^P (x_{ri} - x_{rj})^2} \quad (2)$$

Además de la distancia euclídea también se pueden utilizar otro tipo de distancias como son la distancia Minkowski, la distancia Manhattan y la distancia Jaccard entre otras. Otro de los parámetros a seleccionar dentro del algoritmo es el valor de K , cuya mejor elección depende fundamentalmente de los datos. Valores grandes de K reducen el efecto del ruido en la clasificación, pero crean límites entre clases parecidas. El

valor óptimo de K puede seleccionarse mediante un proceso de optimización.

- **SVM:** el modelo SVM [8] es un modelo de aprendizaje supervisado que, partiendo de un conjunto de datos de entrenamiento etiquetados en diferentes clases, puede representar dichos datos en el espacio para tratar de separar las diferentes clases mediante un espacio lo más amplio posible, para que los nuevos datos puedan ser clasificados correctamente en función de su proximidad.

En el concepto de separación óptima es donde reside la característica fundamental de los modelos SVM. Este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia con los puntos que estén más cerca de él mismo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentran en otras categorías estarán a otro lado del hiperplano.

La separación entre clases se realiza mediante la representación por funciones núcleo o *Kernel*. Cuando las clases no son linealmente separables se pueden aplicar las funciones *Kernel* que definen una nueva dimensión para encontrar un hiperplano que separe a las distintas clases. La función *Kernel* [9] es una función que devuelve el resultado del producto escalar entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores. Gracias a los *Kernels* se puede conseguir la clasificación para cualquier número de dimensiones. Existen multitud de *Kernels* diferentes, entre los más utilizados destacan el *Kernel* lineal, el gaussiano, el cúbico y el cuadrático.

- **Decision tree:** el modelo DT o árbol de decisión [8] es un algoritmo de aprendizaje supervisado que se utiliza para clasificar utilizando particiones sucesivas. Son apropiados cuando hay un número elevado de datos, siendo una de sus ventajas su carácter descriptivo que permite entender las decisiones tomadas por el modelo.

Los árboles de decisión son un modelo que, partiendo de una base de datos de entrenamiento, crea diagramas de construcciones lógicas que nos ayudan a resolver problemas de clasificación. A esta técnica se la conoce como segmentación jerárquica, la cual se trata de una técnica explicativa y de descomposición que utiliza un proceso de división secuencial y descendiente que, partiendo de una variable dependiente, forma grupos que se definen mediante combinaciones de variables independientes.

Los árboles de decisión están formados por los siguientes elementos: nodos ramas y hojas. Los nodos son las variables de entrada, las ramas

representan los posibles valores de las variables de entrada y las hojas son los posibles valores de la variable de salida. Como primer elemento de un árbol de decisión se tiene el nodo raíz que representa la variable de mayor importancia en el proceso de clasificación. Un ejemplo de árbol de decisión en el que se clasifican diferentes tipos de flores, en función del pétalo y el sépalo se muestra en la figura 1.

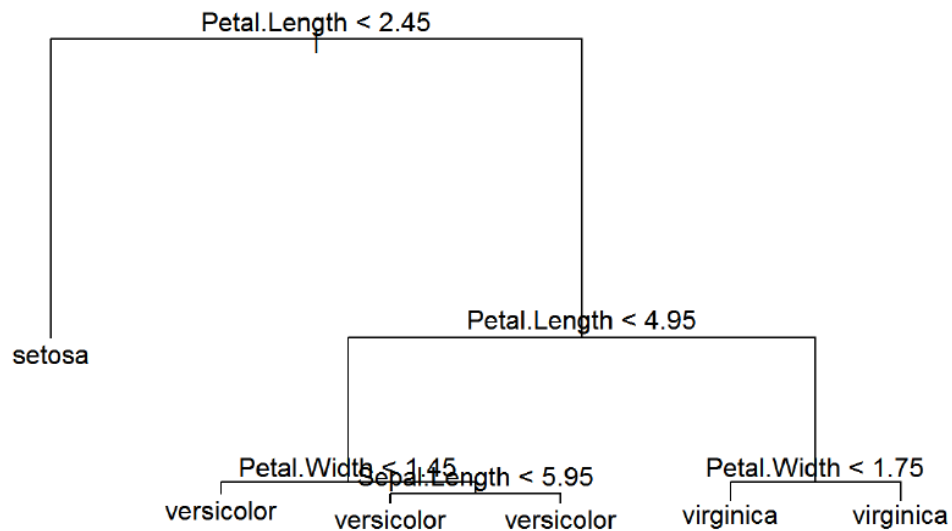


Figura 1. Ejemplo de árbol de decisión [8].

2.2 Sistemas de Localización

Un sistema de localización [10] es un protocolo automatizado que ubica cualquier objeto móvil tanto en un espacio físico concreto como en un tiempo y momento determinado. La ubicación sistemática y continuada del objeto móvil utilizando el mismo sistema de localización se denomina seguimiento o *tracking*.

En exteriores los sistemas de posicionamiento ya tienen un estándar exitoso con los Sistemas Globales de Navegación por Satélite (GNSS). Sin embargo, éste no es el caso en interiores. Los sistemas de posicionamiento en interiores se definen como [11] una red de dispositivos que permite localizar a través de un sistema inalámbrico objetos o personas dentro de un edificio.

El diseño de un Sistema de Posicionamiento en Interiores (IPS) depende en gran medida del contexto y se basa en tres componentes principales, la tecnología base del sistema, la técnica de localización empleada y el método de posicionamiento. Las tecnologías que se utilizan en estos sistemas son de diversa naturaleza, pueden ser tecnologías ópticas como la comunicación de luz visible (VLC), de radiofrecuencia como el WiFi, *Bluetooth* o el *Ultrawideband* (se utiliza para aplicaciones más precisas), de medición acústica e inercial. Este trabajo se centra en el uso de la tecnología de radiofrecuencia (RF) y de la tecnología

inercial. A continuación, se describen brevemente las técnicas de localización y los métodos más empleados de ambas tecnologías.

2.2.1 RF

La tecnología base de radiofrecuencia empleada depende de varios aspectos de diseño [12], las características del tipo de radiofrecuencia, la magnitud física observable, la privacidad del usuario, la localización individual o conjunta, la precisión obtenible y los métodos matemáticos de estimación.

La precisión en la localización va ligada a la tecnología utilizada, como se muestra en la figura 2.

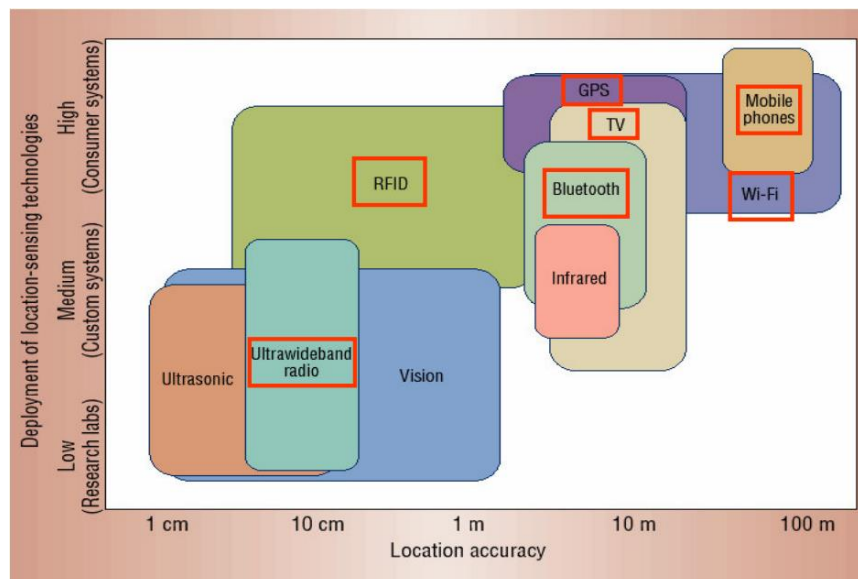


Figura 2. Precisión en función de la tecnología utilizada [12].

Las principales características de los sistemas de localización basados en RF son las siguientes:

- Son capaces de operar con o sin visión directa entre emisor y receptor.
- La influencia del entorno es importante (atenuación causada por obstáculos y personas entre otros).
- Resultan afectados por interferencias procedentes de otros sistemas de RF o entre ellos mismos.
- La predicción determinista del comportamiento de la señal RF es imposible en condiciones realistas.
- Los sistemas RF pueden beneficiarse de infraestructuras ya presentes.
- La diversidad espacial obtenida por combinar medidas procedes de diferentes señales RF aumenta la precisión.

A la vista de las características de los sistemas de localización basado en RF y los aspectos de diseño se va a escoger la tecnología o tecnologías bases para

nuestro sistema. Dentro de las especificaciones de diseño está la localización a nivel de habitación, por lo que la precisión estará comprendida en torno a unos pocos metros, además la tecnología escogida debe ser lo menos invasiva posible para garantizar la privacidad del usuario. Por tanto, las dos tecnologías que se van a utilizar como base de los sistemas de localización que se van a desarrollar en el trabajo son la tecnología WiFi y *Bluetooth*. Estas tecnologías cumplen con las condiciones de diseño descritas, además otra de las ventajas de estas tecnologías es su bajo coste.

Una vez conocidas las tecnologías que se van a utilizar, se van a desarrollar las técnicas más empleadas utilizando como base la tecnología RF. Las técnicas empleadas indican que medidas o información se procesa para calcular la posición. Las técnicas más empleadas son las siguientes:

- **Técnica del ángulo de llegada (AoA):** Utiliza como medidas la dirección y el ángulo de llegada una señal.
- **Técnica del tiempo de llegada (ToA):** Utiliza el tiempo transcurrido de una señal desde el transmisor al receptor.
- **Técnica de la información del estado del canal (CSI):** Utiliza como información las propiedades de un canal en un enlace de comunicación.
- **Potencia de la señal medida por el receptor (RSSI):** Se utiliza como información la potencia de la señal medida por el receptor.
- **Método *fingerprinting*:** Se utiliza como información el conjunto de RSSI procedente de múltiples emisores medida por el receptor.

En el caso del sistema WiFi la técnica que se va a emplear es la de la potencia de la señal medida por el receptor y en el caso del sistema *Bluetooth* el método escogido es el método *fingerprinting*.

El método de *fingerprint* [13] se basa en un vector de potencias RSSI compuesto por los valores de emisión de diferentes dispositivos o emisores asociado a una posición determinada. El método *fingerprinting* está dividido en dos fases, una fase de entrenamiento conocida como fase *offline* y una fase de posicionamiento conocida como fase *online*. En la fase de entrenamiento se generan los vectores de potencia RSSI que se asocian a posiciones conocidas. Esta fase se utiliza para construir la base de datos o mapa de radio que cubre el área de interés. La base de datos puede incluir además de los valores de potencia RSSI y su posición asociada, el tipo de dispositivo que se utiliza para realizar las mediciones o la orientación de la medida. La fase de posicionamiento o fase *online* consiste en medir vectores de potencia RSSI y compararlos con los datos almacenados en la fase de entrenamiento para generar una estimación de la posición.

Por último, para la estimación de la posición es necesario el uso de un método de posicionamiento en interiores. El método es el algoritmo que se va a

emplear para procesar los datos/mediciones o la información recopilada para el posicionamiento. Entre los métodos más utilizados para la localización en interiores se encuentran la triangulación y la multilateración, pero actualmente se emplean en mayor medida los algoritmos de *Machine Learning*. Además del empleo de métodos más complejos también se emplean métodos más sencillos como puede ser el método por proximidad.

El método por proximidad consiste en medir la potencia RSSI de diferentes dispositivos o emisores y estimar la localización en función del dispositivo cuya potencia RSSI sea la mayor. En el caso del sistema WiFi se va a utilizar este método y en el caso del sistema *Bluetooth* se van a utilizar los tres métodos de *Machine Learning* que se han explicado en el apartado anterior, el algoritmo KNN, el SVM y el DT. Por tanto, ya se han establecido tanto la tecnología, la técnica y los métodos que se van a utilizar para los sistemas de localización a desarrollar en este trabajo.

En la literatura se encuentran numerosos trabajos en los que se obtiene la localización contextual empleando tecnología WiFi y *Bluetooth*. En [14] se realiza la comparación de dos métodos de ML (KNN y SVM) utilizando como tecnología de base la tecnología WiFi y *Bluetooth*. En el caso del método KNN los mejores resultados se obtienen combinando ambas tecnologías consiguiendo una precisión del 75 % y, en el caso del método SVM los mejores resultados se obtienen utilizando la tecnología WiFi con una precisión del 73,7 %. Por otro lado, en [15] se presenta un sistema de localización simbólica semejante al que se va a desarrollar en este trabajo empleando ambas tecnologías. En el caso del sistema WiFi se va a emplear un método de ML conocido como *Random Forest* obteniendo precisiones por encima del 90 % en algunos usuarios. En el caso del sistema *Bluetooth* se plantean tres métodos diferentes, un método por proximidad, el método KNN realizando el entrenamiento de forma manual y el método KNN realizando el entrenamiento de forma automática. La mejor precisión se obtiene utilizando el método KNN cuyo entrenamiento se realiza de forma manual, aunque con los otros dos métodos también se obtienen buenos resultados.

Por tanto, usando métodos de proximidad o métodos basados en ML utilizando como base la tecnología *Bluetooth* o WiFi se obtienen buenos resultados en lo que a precisión en la localización simbólica se refiere por lo que su uso está justificado para el desarrollo de este trabajo. Además, para mejorar la precisión en los resultados se va a hacer uso de la tecnología inercial.

Uno de los objetivos principales de este trabajo es la obtención de rutinas de comportamiento a partir de la localización para identificar posibles desviaciones en la misma. En la literatura se encuentran diferentes trabajos en los que se extraen ciertas características a partir de la localización que

pueden ser interesantes para el establecimiento de rutinas. En [16] se emplea un sistema de localización basado en tecnología *Bluetooth* en el que se obtiene la localización a partir de varios modelos de ML (*Decision Tree*, *Random Forest* y KNN). De la localización obtenida se extraen la zona en la que se encuentra el usuario monitorizado cada minuto y el tiempo total en cada una de las habitaciones. Por otro lado, en [17], se emplea un sistema basado en *Bluetooth* utilizando un modelo de ML basado en *Random Forest*. En este caso a la localización obtenida se le asocia una actividad con la que se puede obtener un patrón de comportamiento del usuario monitorizado.

Por tanto, utilizando sistemas de localización basados en RF se pueden obtener características para poder determinar posibles rutinas de comportamiento y las posibles variaciones respecto de estas. En este trabajo a partir de la localización se pretende conocer el tiempo total en cada una de las estancias monitorizadas para determinar las rutinas de comportamiento. Además, en el sistema *Bluetooth* también se va a incorporar el orden en el que se visitan las diferentes habitaciones como forma complementaria de establecer las rutinas de comportamiento.

2.2.2 Inercial

Los sistemas inerciales están formados por un componente principal, la Unidad de Medida Inercial (IMU). Una IMU [18] es un dispositivo formado por una combinación de acelerómetros y giróscopos que permite medir las aceleraciones lineales y velocidades angulares de los tres ejes.

Los sistemas de navegación inercial son sistemas de navegación autónoma en la que las medidas proporcionadas por la IMU se emplean para rastrear la posición y orientación de un objeto con respecto a un punto inicial conocido.

Aunque el uso principal de los sistemas inerciales es la navegación inercial actualmente otro de sus usos se encuentra en la medición de la actividad física. A partir de los sensores inerciales se pueden medir la velocidad de la marcha o el número de pasos que se han realizado. El método que se utiliza para la detección del número de pasos o de la velocidad de la marcha depende de la posición de la IMU. Las posiciones más utilizadas para la detección de pasos son el muslo [19], el tobillo y el propio zapato, aunque las IMUs también se usan a menudo en la muñeca, cintura y en algunos casos colgado del pecho. En el caso del muslo los métodos utilizados se basan en algoritmos que identifiquen desplazamientos verticales [20], la medición de la aceleración en el eje z [21], la medición de la aceleración en los tres ejes [22] y, por último, el enfoque basado en el cálculo del ángulo de Euler *Pitch* [23]. Este último también se emplea en el caso de que la IMU se coloque en el tobillo [24].

En el caso del sistema WiFi se va a utilizar el método del ángulo de Euler *Pitch* para calcular el número de pasos colocando la IMU en el tobillo de usuario

monitorizado. Los ángulos de Euler (*Pitch*, *roll* y *yaw*) constituyen un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijo. En nuestro caso sólo se va a utilizar el ángulo *Pitch*, que representa la orientación respecto al eje de coordenadas y.

El método del ángulo de Euler *Pitch* se desarrolla en [23], este consiste en el cálculo del ángulo empleando un método de fusión sensorial a partir de las aceleraciones lineales y las velocidades angulares medidas por la IMU. Una vez estimado el ángulo se emplea la técnica de cruce por cero (ZCT) para poder estimar el número de pasos. La técnica de cruce por cero consiste en el cálculo de la amplitud de la señal obteniendo el valor máximo antes de que la señal cruce por cero y el valor mínimo una vez haya cruzado. En el caso de que la amplitud calculada se encuentre por encima de un umbral se determina que se ha dado un paso.

Por otro lado, en el caso del sistema *Bluetooth* no se va a calcular el número de pasos ya que el receptor no va a estar en una posición única y éstos no se pueden determinar con precisión, pero si se va a hacer uso de la medida de la aceleración para determinar si el usuario monitorizado se encuentra en movimiento.

Para ambos sistemas, la información proporcionada por la IMU se va a utilizar para corregir posibles errores en la localización a nivel de habitación, mejorando la precisión de los sistemas utilizando únicamente tecnología de radiofrecuencia.

2.3 Fusión de datos/información

La fusión de datos o fusión sensorial [27] es el proceso en el que se utiliza información de varios sensores diferentes para estimar el estado de un sistema dinámico. La estimación resultante es, en algunos sentidos, mejor de lo que sería si los sensores se utilizan individualmente. Además, la estimación resultante se puede obtener utilizando datos de diferentes tipos de sensores.

El planteamiento de un marco de fusión sensorial es el siguiente: el marco recibe mediciones de varios sensores, las fusiona y produce una estimación del estado que puede ser utilizada para varias aplicaciones. La fusión sensorial puede reducir el coste, el número de componentes implicados, al tiempo que aumenta la precisión y la confianza del sistema.

Los sistemas que utilizan métodos de fusión sensorial presentan una serie de ventajas respecto a los sistemas que utilizan un solo sensor. Los sistemas que utilizan un solo sensor suelen sufrir los siguientes problemas:

- **Privación de sensores:** la avería del sensor provoca una pérdida de percepción del entorno o elemento que se quiere analizar.

- **Cobertura espacial limitada:** un único sensor suele cubrir una región limitada.
- **Cobertura temporal limitada:** algunos sensores necesitan un tiempo de preparación determinado para realizar y transmitir una medición, lo que limita la frecuencia máxima de las mediciones.
- **Imprecisión:** las mediciones de los sensores individuales están limitadas a la precisión del sensor empleado.
- **Incertidumbre:** a diferencia de la imprecisión, depende del entorno o elemento observado y no del sensor. La incertidumbre surge cuando faltan características, cuando el sensor no puede medir todos los atributos relevantes del entorno o elemento o cuando la observación es ambigua.

En un sistema de fusión sensorial, se puede conseguir un sistema robusto frente a la falta de sensores utilizando sensores que proporcionen información superpuesta sobre el elemento o entorno analizado. Esto funciona con un conjunto de sensores del mismo tipo (homogéneos) o de diferente tipo (heterogéneos). De la fusión de datos de sensores heterogéneos o homogéneos se obtienen las siguientes ventajas:

- **Robustez y fiabilidad:** los conjuntos de sensores tienen una redundancia inherente, que permite al sistema proporcionar información incluso en caso de fallo de alguno de los sensores.
- **Cobertura espacial y temporal ampliada:** uno de los sensores puede realizar una medición donde otros no pueden.
- **Aumento de la confianza:** una medición de uno de los sensores es confirmada por las mediciones de otros sensores que cubren el mismo dominio.
- **Reducción de la ambigüedad:** la información procedente de varios sensores reduce el número de interpretaciones ambiguas de los valores medidos.
- **Robustez frente a interferencias:** al aumentar la dimensionalidad del espacio de medición, el sistema es menos vulnerable a las interferencias.
- **Mejora de la resolución:** cuando se combinan varias mediciones de sensores independientes que miden la misma propiedad la resolución obtenida es mayor que en el caso de una única medición de los sensores.

Los procesos de fusión sensorial se pueden clasificar en función del tipo de datos que se utilizan y en función del tipo de configuración de los sensores. En función del tipo de datos que se utiliza los procesos de fusión se clasifican en tres niveles diferentes:

- **Fusión de bajo nivel:** combina fuentes de datos brutos para producir nuevos datos que se esperan que sean más informativos que los sensores de entrada.

- **Fusión de medio nivel:** combina varias características como pueden ser bordes, texturas, posiciones, en un mapa de características que luego puede utilizarse para la segmentación y la detección. Generalmente este nivel de fusión se utiliza en el tratamiento de imágenes.
- **Fusión de alto nivel:** se basa en la combinación de diferentes decisiones. Los métodos de decisión que se utilizan incluyen la votación, los métodos de lógica difusa y los métodos estáticos entre otros.

Una vez conocidos los tipos de fusión sensorial en función del tipo de datos que se utilizan se va a realizar la clasificación en función del tipo de configuración de sensores. Los tipos de fusión sensorial en función de la configuración de los sensores son los siguientes:

- **Complementaria:** la configuración de sensores se denomina complementaria si los sensores no dependen directamente unos de otros, sino que pueden combinarse para obtener una información más completa acerca del entorno o elemento observado. Esto resuelve el carácter incompleto de los datos de los sensores.
- **Competitiva:** los sensores se configuran de forma competitiva si cada uno de ellos proporciona mediciones independientes de la misma propiedad. Se distinguen dos posibles configuraciones de sensores, la fusión de datos de diferentes sensores o la fusión de medidas de un solo sensor tomadas en diferentes instantes.
- **Cooperativa:** una red de sensores cooperativa utiliza la información proporcionada por dos sensores independientes para obtener información que no podría obtenerse de los sensores individuales.

Estas tres categorías que no son mutuamente excluyentes, sino que en muchas aplicaciones se implementa más de uno de estos tres tipos. En el caso del sistema WiFi que se va a desarrollar en este trabajo se van a realizar dos fusiones sensoriales diferentes. La primera consiste en la fusión de los datos del acelerómetro y giróscopo para la obtención del ángulo de Euler *Pitch* con el que se va a medir el número de pasos del usuario monitorizado. Este tipo de fusión consiste en una fusión de bajo nivel ya que se utilizan los datos brutos de acelerómetro y giróscopo, además también se trata de fusión complementaria y cooperativa al mismo tiempo. La fusión es complementaria ya que los datos proporcionados por acelerómetro y giróscopo no dependen unos de otros, pero puede utilizarse para obtener información más completa y es cooperativa ya que a partir de la información de ambos se obtiene el ángulo que no podría obtenerse de los sensores individuales.

La segunda consiste en la fusión de los datos de localización obtenidos por sistemas con la información de la actividad física con el fin de corregir posibles errores en la localización. Este tipo de fusión se trata de alto nivel ya que en función de ambas informaciones se tomará la decisión o no de corregir la localización, además se trata de una fusión complementaria. Este tipo de fusión

también se va a emplear en el sistema *Bluetooth* que se va a desarrollar en este trabajo siendo el objetivo el mismo que en el caso del sistema WiFi.

Para la primera fusión sensorial se utiliza el filtro de Kalman Extendido (EKF) desarrollado en [25]. El filtro de Kalman Extendido es una particularización del filtro de Kalman utilizada en sistemas no lineales. El EKF resuelve el problema de la estimación del estado generado por un sistema no lineal contaminado por ruido Gaussiano de media nula y varianza conocida, utilizando la expansión de la serie de Taylor que aproxima las ecuaciones no lineales de estado y de observación [26]. El EKF se trata de un proceso iterativo que está compuesto por dos etapas diferentes, la etapa de predicción y la etapa de corrección. En la etapa de predicción se realiza la estimación del estado actual a partir de una serie de ecuaciones y del estado anterior y en la etapa de corrección se realiza el ajuste del estado predicho mediante una observación del estado actual.

La etapa de predicción se encarga de calcular la estimación del estado y la matriz de covarianza del error a priori a partir de una serie de ecuaciones.

$$\hat{x}_{\bar{k}} = f(\hat{x}_{k-1}) \quad (3)$$

$$P_k^- = A_k * P_{k-1} * A_k^T + Q \quad (4)$$

$$A_k = \frac{d\hat{x}_{\bar{k}}}{dx_k} \quad (5)$$

$\hat{x}_{\bar{k}}$ se corresponde con la estimación del vector de estados que es función del estado anterior y P_k^- se corresponde con la matriz de covarianza del error a priori, que depende de su estado anterior, de la matriz A_k que es el jacobiano de la ecuación de transición del sistema y es la que se encarga de linealizar el sistema y Q que se trata de la matriz de covarianza de ruido de proceso.

La etapa de corrección se encarga de la corrección de la estimación calculada a priori en la etapa anterior mediante la observación del estado actual a partir de una serie de ecuaciones.

$$S = H * P_k^- * H^T + R \quad (6)$$

$$K = P_k^- * H^T * S^{-1} \quad (7)$$

$$\hat{x}_k = \hat{x}_{\bar{k}} + K * (z_k - h(\hat{x}_{\bar{k}})) \quad (8)$$

$$P_k = P_k^- - K * S * K^T \quad (9)$$

$$H = \frac{dh(\hat{x}_{\bar{k}})}{dx_k^-} \quad (10)$$

S se corresponde con la matriz intermedia que depende de H que es el jacobiano de la ecuación de observación (10), de la matriz de covarianza del error a priori

P_k^- y de R , que se trata de la matriz de covarianza del ruido de medida R . K se corresponde con la ganancia de Kalman, que depende de P_k^- , H y S . \hat{x}_k se corresponde con la estimación del estado a posteriori a partir de \hat{x}_{k-1} , K y de la diferencia entre el vector de estimación z_k y el vector de observación $h(\hat{x}_{k-1})$. Por último P_k se corresponde con la matriz de covarianza del error a posteriori, que depende de P_k^- , de S y de K .

3. Descripción del material utilizado

En este apartado se va a describir todo el material Hardware que ha sido necesario para la realización del trabajo, incluyendo los elementos que forman el sistema WiFi y los elementos que forman el sistema Bluetooth.

3.1 Sistema WiFi

Desde el punto de vista del Hardware, el sistema consta de dos elementos principales: los emisores WiFi y el receptor portátil que lleva el usuario. Los elementos utilizados se describen a continuación:

- **Receptor portátil:** el receptor escogido, al que denominaremos por su nombre comercial NGIMU [28], cumple con las especificaciones del sistema. La NGIMU es una plataforma de adquisición de datos compacta (Su tamaño es de 56×39×18 mm y su peso de 46 g) que combina sensores integrados con algoritmos de procesamiento de datos, además de un amplio rango de interfaces de comunicación, lo que es muy útil para aplicaciones en tiempo real y de almacenamiento de datos. La frecuencia de muestreo de los diferentes sensores de la NGIMU es configurable. Para esta aplicación, se utiliza el interfaz WiFi como un sistema receptor.

Entre los sensores de los que dispone el receptor destacan una IMU de 9 ejes formada por tres giróscopos, tres acelerómetros y tres magnetómetros; y otros sensores adicionales de presión, humedad y temperatura entre otros. Este receptor presenta dos puertos serie auxiliares, ocho entradas analógicas y un interfaz de comunicación WiFi, pudiendo trabajar como cliente o como punto de acceso. Una de las limitaciones que presenta es que al trabajar como cliente sólo puede conectarse a una determinada SSID. La NGIMU se muestra en la figura 3 junto con el gráfico del sistema de referencia propio que también puede verse grabado en la carcasa del receptor.

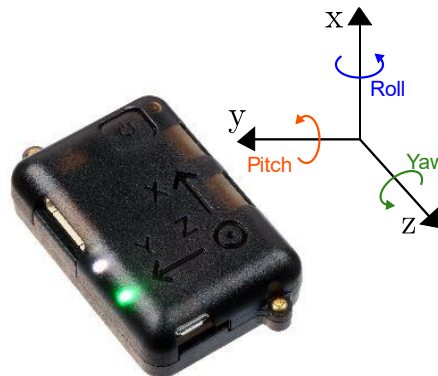


Figura 3. Receptor del sistema propuesto, NGIMU [28].

- **Emisor (*Router*):** como emisor del sistema se utiliza un *router* WiFi comercial de corto alcance de TP-LINK (TL-WR802N) [29]. Éste tiene una tasa de transferencia de 300 Mbps y una frecuencia de emisión que se encuentra entre los 2.4 y 2.4835 GHz. Su tamaño es pequeño (57x57x18 mm) lo que facilita su ubicación. Soporta 5 modos de funcionamiento: *router*, repetidor, cliente, punto de acceso y WISP. Al configurar un emisor en modo *router* debe realizarse con una única SSID (configurable) y contraseña que coincida con la definida en el receptor NGIMU. La potencia de emisión del emisor es configurable entre tres niveles diferentes, que son bajo, medio u alto. El emisor TL-WR802N se muestra en la figura 4.



Figura 4. Emisor router propuesto de TP-LINK (TL-WR802N) [29].

3.2 Sistema Bluetooth

Desde el punto de vista del Hardware el sistema *Bluetooth* está formado por dos elementos principales: los transmisores *Bluetooth* y un receptor portado por el usuario. Ambos elementos son sistemas comerciales, los cuales se describen a continuación:

- **Receptor portátil:** el receptor del sistema *Bluetooth* se trata del teléfono móvil BQ Aquaris Plus [30]. El receptor funciona como sistema multisensorial ya que dispone de IMU, sensor de proximidad y sensor de luz. Además, permite conectividad WiFi y *Bluetooth 4.2*, lo que es esencial para el sistema que se va a desarrollar. El sistema operativo que utiliza el receptor es el *Android Nougat 7.0*.

Para la captura de datos se emplea una aplicación específica, llamada *GetSensorData*, la cual se describe en [31]. La aplicación guarda en un mismo archivo la información de los sensores disponibles en el teléfono móvil. La pantalla principal de la aplicación muestra cuáles son los sensores disponibles y cuáles no, lo que se muestra en la figura 5. Además, también permite visualizar la información de los sensores en tiempo real y las características de estos.

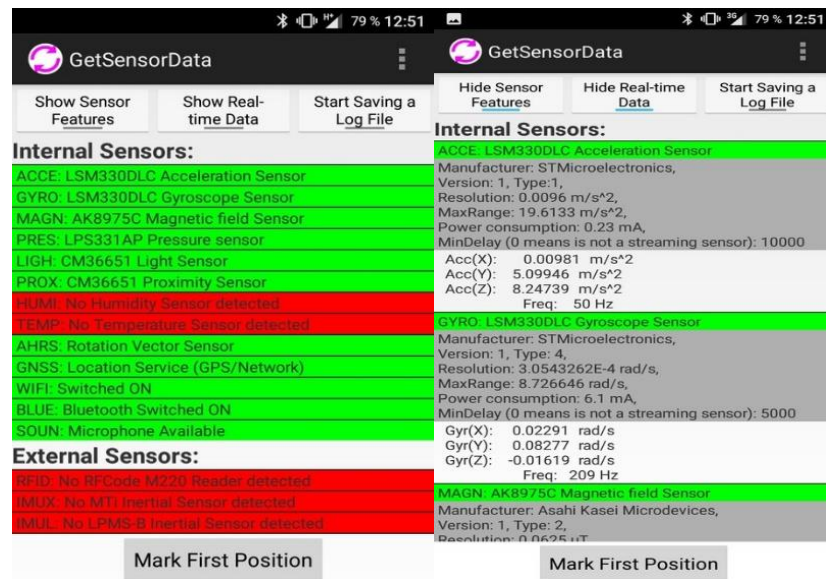


Figura 5. Pantalla principal de la aplicación GetSensorData [31].

Cada uno de los archivos de salida de la aplicación tiene formato txt y está estructurado en filas con diferentes tipos de datos. Cada fila recoge el dato de un sensor determinado en un instante de tiempo dado, lo cual se muestra en la figura 6.

```

GYRO;4.410;-0.03726;-0.16860;-0.17318;3
AHRS;4.410;28.7201;0.4508;177.5888;0.00141;0.24804;0.96855;-106
LIGH;4.410;4767.0;0
ACCE;4.411;-0.67995;4.85544;8.30309;3
MAGN;4.411;1.26000;-39.48000;-17.76000;3
PRES;4.419;948.5206;0
GYRO;4.420;-0.04948;-0.13347;-0.16463;3
AHRS;4.420;28.7140;0.5296;177.4918;0.00095;0.24800;0.96855;122
LIGH;4.420;4748.0;0
ACCE;4.421;-0.69911;4.90333;8.35098;3
MAGN;4.421;1.26000;-39.36000;-17.76000;3
GYRO;4.430;-0.06017;-0.09407;-0.13347;3
AHRS;4.430;28.6960;0.6096;177.4190;0.00043;0.24786;0.96858;-3
ACCE;4.432;-0.70868;4.88417;8.34140;3
MAGN;4.432;1.38000;-39.42000;-17.64000;3
LIGH;4.432;4748.0;0
WIFI;4.432;eduroam;00:0b:86:27:36:c0;-73
WIFI;4.432;portal-csic;00:0b:86:27:36:c1;-74
WIFI;4.432;eduroam;00:0b:86:27:32:e0;-65
WIFI;4.432;portal-csic;00:0b:86:27:32:e1;-65
WIFI;4.432;eduroam;00:0b:86:27:35:90;-82
WIFI;4.432;portal-csic;00:0b:86:27:35:91;-80
WIFI;4.432;WiFiArganda;00:0d:97:00:d2:55;-89
WIFI;4.432;eduroam;00:0b:86:27:35:80;-97
WIFI;4.432;portal-csic;00:0b:86:27:35:81;-95
GYRO;4.440;-0.06353;-0.05528;-0.08460;3
AHRS;4.440;28.6667;0.7015;177.3758;-0.00026;0.24763;0.96863;-68
LIGH;4.441;4748.0;0
    
```

Figura 6. Formato de salida de la aplicación GetSensorData [31].

La frecuencia de muestreo puede modificarse mediante la pantalla de configuración de la aplicación, lo que se muestra en la figura 7.

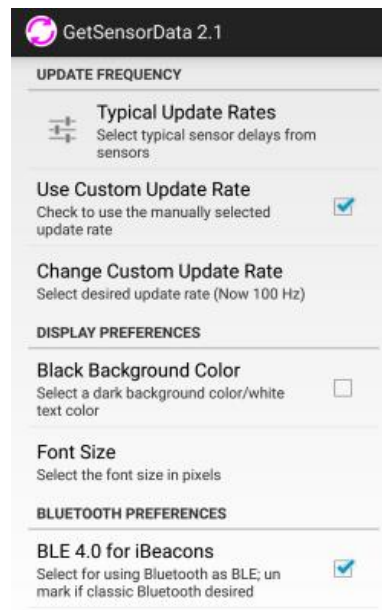


Figura 7. Pantalla de configuración de la aplicación *GetSensorData* [31].

La frecuencia de muestreo debe coincidir con las especificaciones de los sensores. La frecuencia de muestreo en el caso del interfaz Bluetooth es de 4.3 Hz y 99.4 Hz en el caso del acelerómetro para el receptor utilizado.

- **Transmisores Bluetooth (balizas):** se utilizan dos tipos de balizas *Bluetooth* fabricadas por *AccentSystem* (IBKS 105 [32] y IBKS PLUS [33]). La principal diferencia entre ellas se encuentra en el rango de emisión y la autonomía. En el caso de las IBKS Plus el rango se encuentra por encima de los 100 metros y su autonomía se encuentra por encima de los 104 meses, mientras que en el caso de las IBKS 105 el rango se encuentra por encima de los 50 metros y su autonomía se encuentra entre los 30 y 46 meses. Ambos dispositivos utilizan dos protocolos de comunicación diferentes, el protocolo *Eddystone* desarrollado por Google y el protocolo *Ibeacon* desarrollado por *Apple*. La potencia de transmisión es configurable (desde +4 dBm hasta -32 dBm) utilizando la aplicación proporcionada por *AccentSystem*.

La figura 8 muestra una imagen de ambas balizas, donde se puede observar su pequeño tamaño (en el caso de la baliza IBKS 105 su tamaño es de 11.3x52.6 mm y en el caso de la baliza IBKS PLUS su tamaño es de 84x84x24 mm). El tamaño es una importante ventaja para colocar las balizas discretamente dentro del entorno. Cada baliza se identifica mediante su dirección de control de acceso (dirección MAC).



Figura 8. Baliza IBKS 105 [\[32\]](#) (derecha) y baliza IBKS Plus [\[33\]](#) (izquierda).

4. Algoritmos propuestos

En este apartado se van a desarrollar los algoritmos que se han utilizado para la generación de rutinas a partir de la estimación de la localización. Inicialmente se presentarán los algoritmos desarrollados para una localización simbólica utilizando la tecnología WiFi (Localización WiFi). En el apartado siguiente se mostrará el equivalente haciendo uso de la tecnología *Bluetooth* (Localización *Bluetooth*). En el tercer y último apartado de este capítulo se van a incluir los algoritmos desarrollados para la generación de rutinas de comportamiento a partir de la localización obtenida por ambos sistemas.

4.1 Localización simbólica con tecnología WiFi

En este apartado se van a desarrollar los algoritmos que se utilizan para obtener la localización simbólica empleando tecnología WiFi. El diagrama de bloques que incluye la algoritmia del sistema de localización WiFi se muestra en la figura 9.

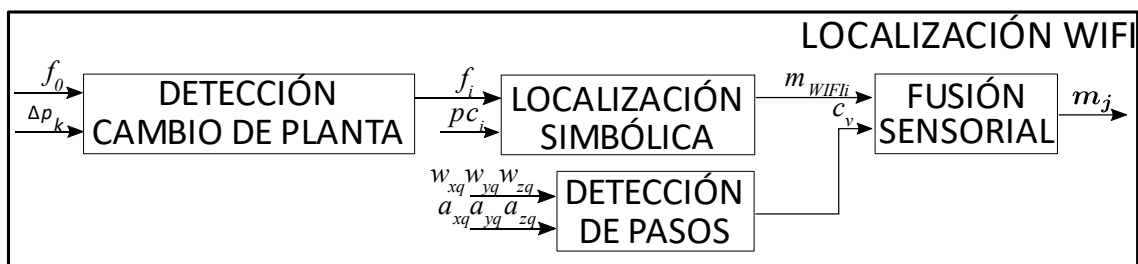


Figura 9. Diagrama de bloques de los algoritmos utilizados para la localización WiFi.

El bloque de localización WiFi está formado por tres etapas consecutivas: detección de cambio de planta, localización simbólica y detección de pasos y, por último, la fusión sensorial. Las entradas al algoritmo de detección de cambio de planta son la planta inicial (f_0) y el incremento de presión medido en el instante k (Δp_k) con el barómetro de la NGIMU. Las entradas del algoritmo de localización simbólica son la planta (f_i) y el porcentaje de conexión WiFi en el instante i (pc_i). Las entradas del algoritmo de detección de pasos son las velocidades angulares (w_q) y aceleraciones lineales (a_q) medidas en un instante q . La localización estimada por el algoritmo de localización simbólica (m_{WiFi}) y el número de pasos contado por el algoritmo de detección de pasos en una ventana de tiempo v (c_v) son las entradas del algoritmo de fusión sensorial. La salida del bloque de localización es un vector con la localización en cada una de las habitaciones monitorizadas para cada instante temporal i , a lo largo de un día j obtenido (m_j).

A continuación, se van a detallar todos los algoritmos que se han desarrollado para obtener la localización simbólica utilizando la tecnología WiFi como base incluidos en la figura 9.

- **Detección de pasos:** el algoritmo se emplea para conocer el número de pasos que se han realizado, para lo que es necesario conocer el ángulo de Euler *Pitch* (σ). La orientación de dicho ángulo sobre el receptor del sistema se muestra en la figura 3.

Los ángulos de Euler se estiman utilizando un filtro de Kalman Extendido (EKF) desarrollado previamente en el grupo de investigación y que se describe en [25]. El filtro está formado por dos etapas principales, una de predicción y otra de corrección. En la etapa de predicción se utilizan los valores de las velocidades angulares (w_{xq} w_{yq} w_{zq}) y en la etapa de corrección se utilizan los valores de las aceleraciones lineales (a_{xq} a_{yq} a_{zq}), proporcionados por la IMU. El índice q se corresponde con la inversa de frecuencia de muestreo de la IMU, que en este caso es de 100 Hz. La etapa de corrección del filtro solo se realiza en el caso de que el módulo de las tres componentes de la aceleración lineal sea aproximadamente el valor de la aceleración de la gravedad, que es el instante en el que el pie está apoyado.

Para el correcto funcionamiento del filtro es necesario definir los valores iniciales para los tres ángulos de Euler *roll*(Φ_0), *pitch*(σ_0) y *yaw*(θ_0). Los tres valores se inicializan a 0. Las ecuaciones de estimación de estado, de la matriz A_q y los valores de la matriz Q de la etapa de predicción del filtro se muestran a continuación. El valor de dt se corresponde con la inversa de la frecuencia de muestreo de la IMU.

$$\hat{x}_q^- = \hat{x}_{q-1} + dt * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (11)$$

$$x_1 = w_{xq-1} + w_{yq-1} * \sin(\Phi_{q-1}) * \tan(\sigma_{q-1}) + w_{zq-1} * \cos(\Phi_{q-1}) * \tan(\sigma_{q-1}) \quad (11.1)$$

$$x_2 = w_{yq-1} * \cos(\Phi_{q-1}) - w_{zq-1} * \sin(\Phi_{q-1}) \quad (11.2)$$

$$x_3 = w_{yq-1} * (\sin(\Phi_{q-1}) / \cos(\sigma_{q-1})) + w_{zq-1} * (\cos(\Phi_{q-1}) / \cos(\sigma_{q-1})) \quad (11.3)$$

$$A_q = \begin{pmatrix} A_{q11} & A_{q12} & 0 \\ A_{q21} & 0 & 0 \\ A_{q31} & A_{q32} & 0 \end{pmatrix} \quad (12)$$

$$A_{q11} = w_{yq-1} * \cos(\Phi_{q-1}) * \tan(\sigma_{q-1}) - w_{zq-1} * \sin(\Phi_{q-1}) * \tan(\sigma_{q-1}) \quad (12.1)$$

$$A_{q12} = w_{yq-1} * \sin(\Phi_{q-1}) * \sec(\sigma_{q-1})^2 + w_{zq-1} * \cos(\Phi_{q-1}) * \sec(\sigma_{q-1})^2 \quad (12.2)$$

$$A_{q21} = -w_{yq-1} * \sin(\Phi_{q-1}) - w_{zq-1} * \cos(\Phi_{q-1}) \quad (12.3)$$

$$A_{q31} = w_{yq-1} * \cos(\Phi_{q-1}) * \sec(\sigma_{q-1}) - w_{zq-1} * \sin(\Phi_{q-1}) * \sec(\sigma_{q-1}) \quad (12.4)$$

$$A_{q32} = w_{yq-1} * \sin(\Phi_{q-1}) * \sec(\sigma_{q-1}) * \tan(\sigma_{q-1}) + w_{zq-1} * \cos(\Phi_{q-1}) * \sec(\sigma_{q-1}) * \tan(\sigma_{q-1}) \quad (12.5)$$

$$Q = \begin{pmatrix} 0.001^2 & 0 & 0 \\ 0 & 0.001^2 & 0 \\ 0 & 0 & 0.001^2 \end{pmatrix} \quad (13)$$

La ecuación z_q de la etapa de corrección y los valores de las matrices R y H se muestran a continuación.

$$z_q = \begin{pmatrix} \tan^{-1}\left(\frac{a_{yq}}{a_{zq}}\right) \\ \tan^{-1}\left(-\frac{a_{xq}}{\sqrt{a_{yq}^2 + a_{zq}^2}}\right) \\ 0 \end{pmatrix} \quad (14)$$

$$R = \begin{pmatrix} 0.01^2 & 0 \\ 0 & 0.01^2 \end{pmatrix} \quad (15)$$

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (16)$$

Una vez estimado el ángulo *Pitch* a partir del EKF se puede determinar el número de pasos a partir de la técnica conocida como cruce por cero (ZCT). La técnica utilizada aprovecha que la señal *Pitch* es periódica a la hora de caminar y que al apoyar el pie su valor es 0, por lo que a la hora de realizar un paso la señal tiene que pasar por 0. Por ello se calcula la amplitud de la señal obteniendo el máximo y el mínimo de la señal antes y después de cruzar por 0. Para identificar que es un paso y que no se define un umbral de amplitud (u_a) con el que se compara la amplitud calculada. El valor de la amplitud (a_s) a comparar con el umbral se calcula cada vez que se produce un paso por cero (índice s), siendo el umbral que se utiliza $u_a=50^\circ$. El umbral empleado es un umbral empírico estimado en el grupo anteriormente. El diagrama de bloques del algoritmo se muestra en la figura 10.

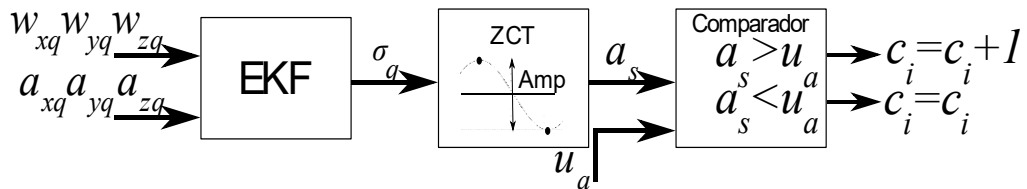


Figura 10. Diagrama de bloques del algoritmo de detección de pasos.

El número de pasos se calcula para ventanas temporales de un segundo (índice i) y su valor se almacena en el contador c_i . El valor del contador se reinicia para cada ventana temporal. Para nuestro caso se calcula el número de pasos por segundo.

- **Detección de cambio de planta:** el objetivo del algoritmo es conocer en que planta se encuentra el usuario monitorizado en cada instante i para lo que hay que detectar cuando cambia de planta dentro del edificio. Las entradas del algoritmo son las diferencias de presión atmosférica medidas entre el instante i y el instante i más k ($\Delta p_{(i:i+k)}$). El índice k se corresponde con un valor de 25 ya que se trata de un valor suficiente como para realizar el cambio de planta tanto por escaleras como por ascensor.

Para relacionar una variación de presión a un cambio de planta es necesaria una calibración previa. El valor u_1 se corresponde con el cambio de presión al subir o bajar una planta y el valor u_n se corresponde con el cambio de presión al subir o bajar las n plantas del edificio. Una vez realizada la calibración previa determinando los umbrales (u_1, \dots, u_n) e indicando la planta inicial (f_0), el algoritmo se encarga de buscar diferencias significativas de cambios de presión. Para la búsqueda de cambios significativos se utiliza el valor de la derivada, cuyo cálculo se incluye en (17).

$$dp(i) = \frac{\Delta p(i:i+k)}{k} \quad (17)$$

El algoritmo detecta un cambio significativo cuando se da la condición descrita en (18). U_{dp} se trata de un umbral empírico que se utiliza para poder detectar los cambios significativos de presión, el valor asociado a este umbral es de 0.5.

$$dp(i) > u_{dp} \ \&\& \ dp(i-1) < u_{dp} \ \&\& \ dp(i-2) < u_{dp} \quad (18)$$

En el momento que el algoritmo detecta un cambio de presión significativo este se compara con los umbrales definidos para conocer el número de plantas que se han subido o bajado. Si el incremento de presión es negativo significa que se ha subido, mientras que si el incremento de presión es positivo significa que se ha bajado. El valor de Δp se compara con los umbrales u_1 hasta u_n para determinar cuál es el número de plantas n_i que se han subido o bajado. El diagrama de bloques del algoritmo se muestra en la figura 11.

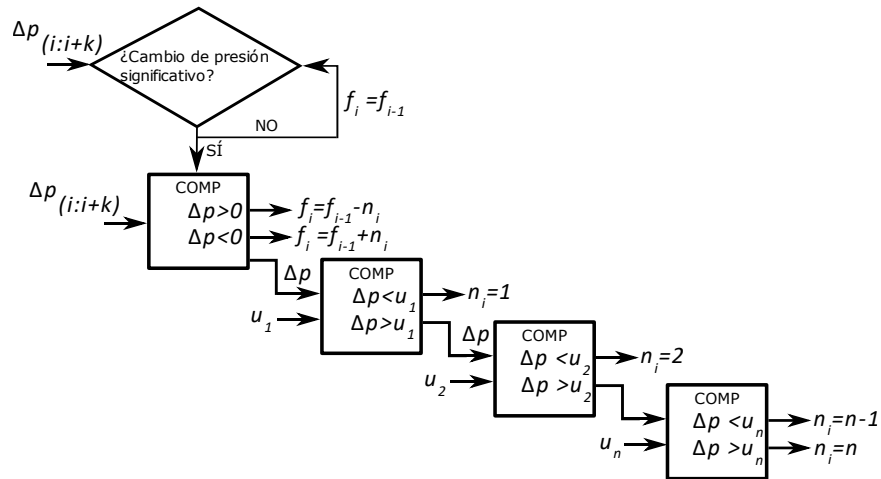


Figura 11. Diagrama de bloques del algoritmo de cambio de planta.

- Detección del tipo de cambio de planta:** el objetivo es conocer si la transición entre plantas se realiza por las escaleras o por el ascensor. Para detectar el tipo de transición se analiza el ángulo de Euler *Pitch* (σ) [34]. El ángulo *Pitch* a la hora de realizar una transición por las escaleras se muestra en la figura 12.

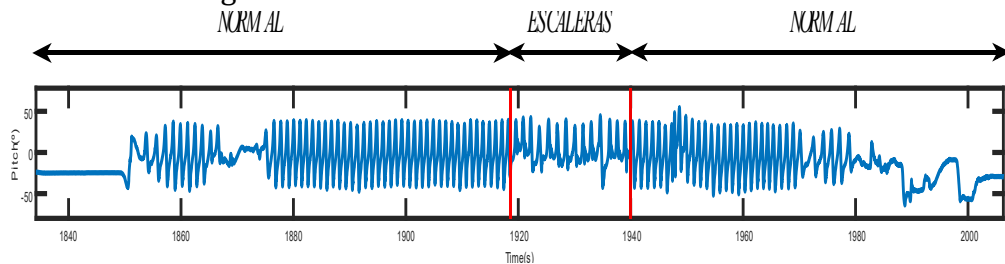


Figura 12. Ángulo *Pitch* al realizar una transición por las escaleras.

Al realizar la transición por las escaleras el ángulo de Euler *Pitch* experimenta un cambio de polaridad. El ángulo *Pitch* a la hora de realizar una transición por el ascensor se muestra en la figura 13.

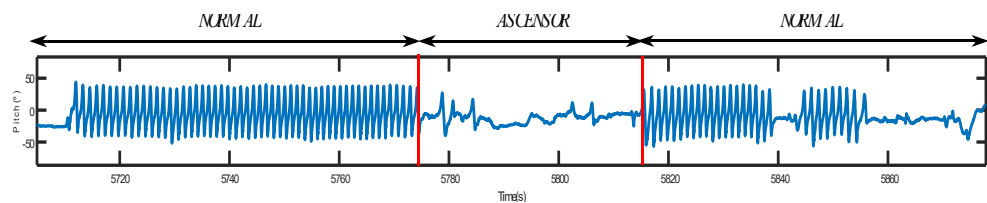


Figura 13. Ángulo *Pitch* al realizar una transición por el ascensor.

Al realizar la transición por el ascensor el ángulo de Euler se encuentra en torno a 0, con la aparición de ruido en la medida. Con las diferencias que se observan entre ambas figuras se puede diferenciar el tipo de transición realizada.

El algoritmo desarrollado permite clasificar el tipo de transición a partir de dos vías. La primera es analizar el cambio en la señal *Pitch* en función del tipo de transición. Para que el algoritmo sea capaz de identificar el

principio (t_1) y final de la transición (t_2) se utilizan las diferencias de presión medidas $\Delta p_{(i:i+k)}$. Una vez determinada la ventana de tiempos se calculan los ciclos de señal (c_c) en los que la señal *Pitch* ha cambiado la polaridad ($\sigma[t_1:t_2]$). Si el número de ciclos es mayor que 5 la transición se realiza por las escaleras y en el caso contrario la transición se realiza por el ascensor.

El umbral que se utiliza permite diferenciar el tipo de transición ya que en ocasiones al realizar la transición por el ascensor aparece un ruido en la señal que se comporta de manera similar a un ciclo de la señal realizando la transición por escaleras. El diagrama de bloques de la primera vía se muestra en la figura 14.

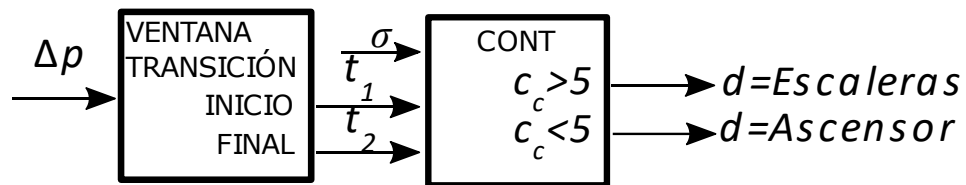


Figura 14. Detección del tipo de cambio de planta a partir del número de ciclos de la señal.

La segunda vía consiste en analizar la desviación estándar de la parte de la señal donde se realiza la transición ($\text{std}(\sigma[t_1:t_2])$). En el caso de que la transición se realice por el ascensor la desviación estándar de la señal es inferior que si la transición se realiza por las escaleras. Es necesario definir un umbral para poder diferenciar el tipo de transición, cuyo valor es 12. El umbral se define empíricamente. El diagrama de bloques de la segunda vía se muestra en la figura 15.

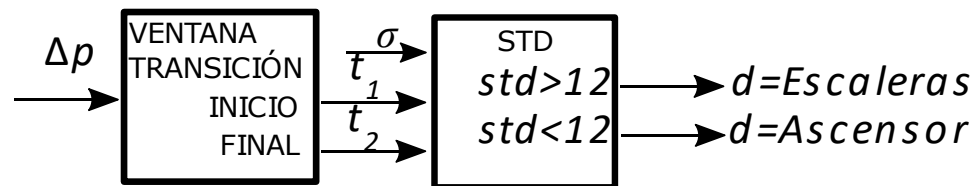


Figura 15. Detección del tipo de cambio de planta a partir de la desviación estándar de la señal.

- Localización simbólica:** el objetivo del algoritmo de localización simbólica es conocer la localización del paciente a nivel de habitación. Las estancias más frecuentadas, que son las más representativas para la actividad del usuario, se seleccionan para la monitorización. En las estancias seleccionadas se colocan los emisores del sistema configurados con la misma SSID previamente definidas en el receptor del sistema. Analizando el porcentaje de conexión (pci), que se compara con un umbral uc , se puede determinar la presencia o no en la sala. El umbral se determina teniendo en cuenta las características de la sala (tamaño, obstáculos entre emisor y receptor, etc). El índice i se corresponde con la

inversa de la frecuencia de muestreo del interfaz WiFi del receptor, que es de 1 Hz (1s). A cada una de las plantas f_i se le asocia una habitación m . El diagrama de bloques del algoritmo de localización simbólica se muestra en la figura 16, donde la primera etapa se corresponde con un Filtro de Media Móvil (FMM) para suavizar la señal de porcentaje de conexión (pcf_i). La segunda etapa se encarga de detectar si el porcentaje de conexión en el instante i se encuentra por encima o por debajo del umbral y por último, la tercera etapa se encarga de detectar en que habitación se encuentra el usuario.

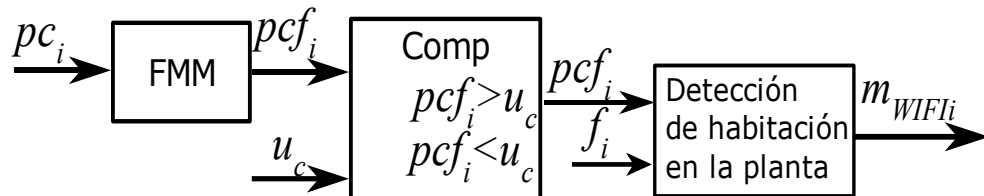


Figura 16. Diagrama de bloques del algoritmo de localización simbólica.

- Algoritmo de fusión sensorial:** la localización simbólica que emplea únicamente el porcentaje de conexión WiFi es poco precisa debido a posibles pérdidas de la señal. Por lo tanto, se propone mejorar los resultados fusionando diferentes sensores. La fusión de los resultados obtenidos (número de pasos, planta y localización simbólica a partir del porcentaje de conexión WiFi) se realiza comprobando la coherencia entre ellos y asignando prioridades, de esta forma se mejora la estimación de la localización.

El algoritmo de fusión se emplea cuando el porcentaje de conexión (pcf_i) se encuentra por debajo del umbral de conexión establecido (u_c), en ese momento se guarda el instante temporal t_i en el que la señal se encuentra por debajo del umbral de conexión y se busca el instante temporal t_f en el que la señal vuelve a estar por encima del umbral. Una vez conocidos t_i y t_f , se analiza el número de pasos de la ventana de tiempo v . La ventana de tiempo v se corresponde con el intervalo $[t_i - \Delta t : t_f + \Delta t]$, donde se utiliza Δt para conocer también los pasos antes y después de la desconexión ya que pueden tener relación directa con ella. El contador c_v acumula los pasos que se dan en la ventana de tiempo y en el caso de que sea inferior al umbral (u_p), se asigna la localización simbólica (m_i) en función de la planta (f_k) en la que se encuentre el usuario. El valor del Δt y del u_p dependen del usuario monitorizado (robusto o frágil). Por usuario robusto se entiende a aquella persona que es capaz de caminar sin ningún tipo de problema físico. El algoritmo de fusión sensorial no corrige la localización en el caso de que el porcentaje se encuentre por debajo de un segundo umbral de conexión u_{c2} . El diagrama de bloques del algoritmo de fusión se muestra en la figura 17.

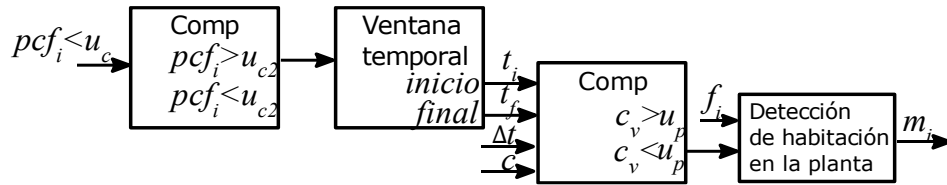


Figura 17. Diagrama de bloques del algoritmo de fusión.

A partir de los resultados del algoritmo de fusión se puede calcular el número de minutos en cada una de las R habitaciones analizadas (característica principal para la generación de rutinas) para cada uno de los días analizados, t_r^j , con $r=1\dots R$, y $j=1\dots J$, siendo J el número de días considerados. Entonces se puede obtener un vector de tiempo por habitación \mathbf{t}_m^j para cada día, el cual se muestra en (19).

$$\mathbf{t}_m^j = [t_1^j \quad \dots \quad t_R^j]_{1 \times R} \quad (19)$$

4.2 Localización simbólica con tecnología Bluetooth

En este apartado se van a desarrollar los algoritmos que se utilizan para obtener la localización simbólica empleando tecnología *Bluetooth*. El diagrama de bloques que incluye la algoritmia del sistema de localización *Bluetooth* se muestra en la figura 18.

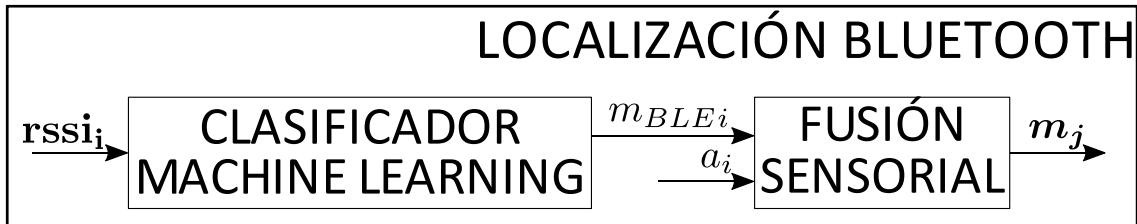


Figura 18. Diagrama de bloques de los algoritmos desarrollados para el sistema Bluetooth.

La localización Bluetooth se basa en la potencia BLE recibida y un clasificador de ML cuya salida se fusiona con la información de la aceleración de la IMU. La entrada del clasificador de ML es el vector de potencias *Bluetooth* medidas en un instante i (\mathbf{rssi}_i). Las entradas del algoritmo de fusión sensorial son la localización obtenida por el clasificador en el instante i (m_{BLEi}) y la aceleración medida en el instante i (a_i). La salida del bloque de localización es el vector de con la localización obtenida en cada una de las habitaciones monitorizadas para cada instante temporal i durante un día j (\mathbf{m}_j).

A continuación, se van a detallar todos los algoritmos que se han desarrollado para obtener la localización simbólica utilizando la tecnología *Bluetooth* como base incluidos en la figura 18.

- **Clasificador basado en Machine Learning:** el clasificador se utiliza para estimar la localización a nivel de habitación (m_{BLEi}) a través de la potencia *Bluetooth* recibida (RSSI) para cada instante de tiempo, \mathbf{rssi}_i . El

clasificador se construye a partir de la aplicación de *Matlab Classification Learner* [35] que permite entrenar modelos de aprendizaje supervisado. Específicamente se van a evaluar y comparar tres algoritmos diferentes: un modelo basado en un *decision tree* con un número máximo de ramas de 100, un modelo basado en el algoritmo SVM empleando un kernel lineal y por último un modelo basado en el algoritmo KNN, donde el número de vecinos $K=1$, utilizando la distancia Euclidiana. Los hiperparámetros son los que vienen por defecto a la hora de generar cualquiera de los tres modelos en la aplicación de *Matlab*. La aplicación permite modificar los parámetros por defecto de los modelos utilizados.

Para todos los modelos, la entrada es un conjunto de RSSI *Bluetooth* medidas en cada una de las habitaciones monitorizadas en las que se encuentra el receptor. En la fase de entrenamiento los modelos se crean a partir de las potencias *Bluetooth* medidas, las cuales están etiquetadas con la habitación en la que se realiza la medida. En la fase de posicionamiento, el modelo generado se utiliza para estimar la localización a nivel de habitación (m_{BLEi}) a partir de las potencias *Bluetooth* medidas ($rssi_i$).

- **Algoritmo de fusión sensorial:** las potencias *Bluetooth* recibidas tienen una gran variabilidad lo que puede provocar grandes errores en la localización. Para minimizar ese efecto, se propone fusionar los resultados del clasificador con los datos del acelerómetro medidos por el receptor del sistema. El diagrama de bloques del algoritmo se muestra en la figura 19.

Si el clasificador estima que hay un cambio de habitación, se analiza si hay movimiento significativo en una ventana de tiempo de 2 segundos, indicado por la señal del acelerómetro (a_i) cuando es superior a un umbral empírico $a_{th} = 4 \text{ m/s}^2$. Si existe movimiento significativo el cambio de habitación m_i estimado por el clasificador se considera como correcto ($m_i = m_{BLEi}$), pero, si no hay movimiento significativo el cambio de habitación estimado por el clasificador no se tiene en cuenta y se corrige por lo que se mantiene la ubicación del estado anterior ($m_i = m_{BLEi-1}$).

Al igual que en el caso del sistema de localización WiFi, a partir de los resultados del algoritmo de fusión sensorial se calcula el número de minutos en cada una de las R habitaciones analizadas para cada uno de los J días analizados. Con ello se puede volver a calcular el vector de tiempo \mathbf{t}_m^j (19) para cada uno de los días.

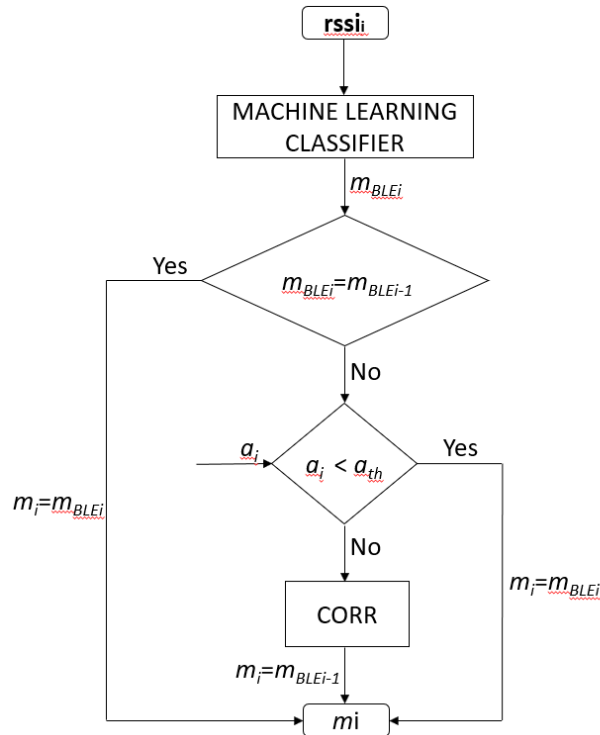


Figura 19. Diagrama de bloques del algoritmo de fusión sensorial.

4.3 Generación de rutinas

En este apartado se van a detallar todos los algoritmos que se han desarrollado para la generación de rutinas de comportamiento a partir de la localización obtenida por ambos sistemas. En el caso de este trabajo la característica principal para la generación de rutinas es el tiempo de paso en cada de las habitaciones monitorizadas.

El diagrama de bloques de la generación de rutinas empleando la localización WiFi se muestra en la figura 20. El bloque de rutinas de comportamiento WiFi se divide en dos bloques principales, en el primero se va a generar un vector de referencia que define los tiempos en cada habitación que se van a establecer como rutina. El segundo bloque estima el coeficiente de reproducibilidad diario para comprobar si se sigue o no la rutina generada por el vector de referencia. La entrada del algoritmo del vector de referencia es la matriz (T_m^j) que incluye los vectores de tiempo en cada una de las habitaciones para un día j (t_m^j) y las entradas del bloque del coeficiente de reproducibilidad son el vector de referencia t_{REF} y los vectores t_m^j .

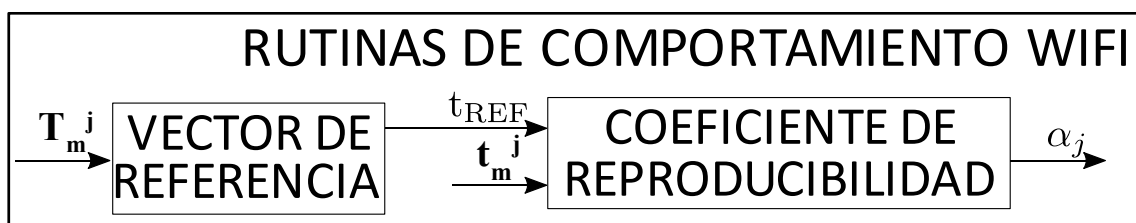


Figura 20. Diagrama de bloques de la generación de rutinas empleando la localización WiFi.

En el caso de la generación de rutinas empleando la localización *Bluetooth* además de emplearse como característica principal el tiempo en cada una de las habitaciones se emplea como característica complementaria el orden en el que se visitan las habitaciones. El diagrama de bloques de la generación de rutinas empleando la localización *Bluetooth* se muestra en la figura 21. El bloque del vector de referencia y del coeficiente de reproducibilidad se vuelve a utilizar igual que en el caso del WiFi, pero en este caso se añaden dos nuevos bloques que son el orden de referencia y el coeficiente de similitud. El primer bloque permite conocer cual es el orden de habitaciones más repetido que se va a establecer como rutina y el segundo bloque estima el coeficiente de similitud diario para comprobar si se sigue o no la rutina generada por el orden de referencia.

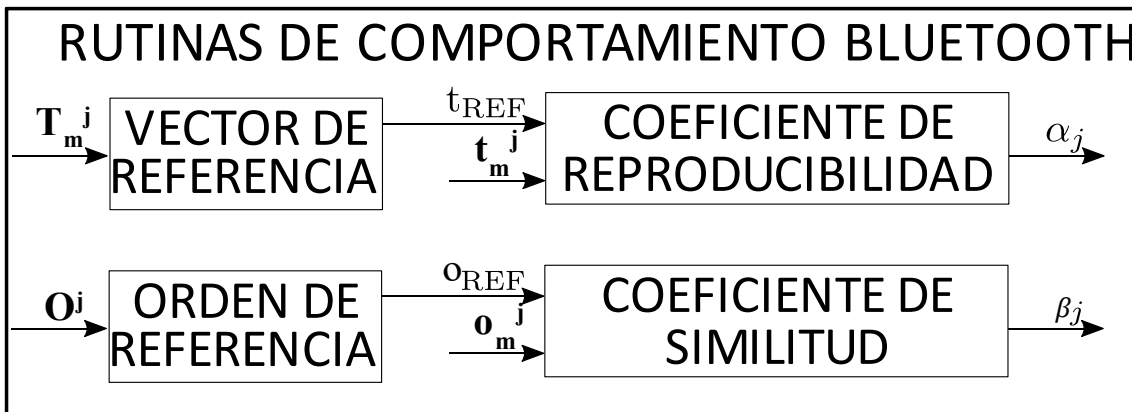


Figura 21. Diagrama de bloques de la generación de rutinas empleando la localización *Bluetooth*.

A continuación, se van a detallar todos los algoritmos que se han desarrollado para la generación de rutinas utilizando tanto la tecnología WiFi como la tecnología *Bluetooth*:

- Vector de referencia:** en el caso del sistema propuesto para la generación de rutinas uno de los métodos de análisis es el tiempo que pasa cada usuario en cada una de las habitaciones, y para R habitaciones se describe el vector de tiempo por habitación para cada día (\mathbf{t}_m^j) que define la ecuación (19). Analizando los vectores de tiempo para un conjunto de días, se puede generar un vector de referencia $\mathbf{t}_{mREF} \in \mathfrak{M}_{1 \times R}(\mathbb{R})$, que represente el tiempo medio que está diariamente el usuario en cada habitación.

Las entradas al algoritmo de generación de vector de referencia son los vectores de tiempo por habitación para J días, que se han estimado a partir de la localización con cualquiera de las dos tecnologías presentadas. A partir de los vectores de tiempo para J días se crean subconjuntos de D días (D seleccionado empíricamente) para obtener diferentes combinaciones de vectores \mathbf{t}_m^j , siendo el objetivo de esto buscar un subconjunto de vectores cuya desviación estándar sea la más pequeña para poder generar un vector de referencia que represente una determinada rutina. El número de combinaciones es $W=C_J^D$. Cada

subconjunto generado es una matriz $T_w \in \mathfrak{M}_{D \times R}(\mathbb{R}), \forall w = 1 \dots W$, siendo R el número de habitaciones y w el número máximo de combinaciones W .

Cada matriz T_w es independientemente analizada calculando la desviación estándar σ del tiempo que se pasa en cada una de las habitaciones R . El algoritmo selecciona los conjuntos w que proporcionan la menor desviación estándar, generando el vector σ_{\min} como se muestra en (20). σ_1^w se corresponde con la desviación estándar mínima para la habitación 1 que se encuentra en un conjunto w y respectivamente σ_R^w se corresponde con la desviación estándar mínima calculada para la habitación R que se encuentra en un conjunto w , pudiendo variar w desde 1 hasta el número máximo de combinaciones posibles W .

$$\sigma_{\min} = [\sigma_1^w \quad \dots \quad \sigma_R^w]_{1 \times R}, w \in [1, \dots, W] \quad (20)$$

Finalmente, el tiempo de referencia por habitación es el tiempo medio dentro del conjunto w que proporciona la menor σ_r^w , que se denomina como \bar{t}_r^w . La ecuación (21) muestra el vector de referencia y la figura 22 representa el diagrama de bloques del proceso descrito.

$$t_{mREF} = [\bar{t}_1^w \quad \dots \quad \bar{t}_R^w]_{1 \times R}, w \in [1, \dots, W] \quad (21)$$

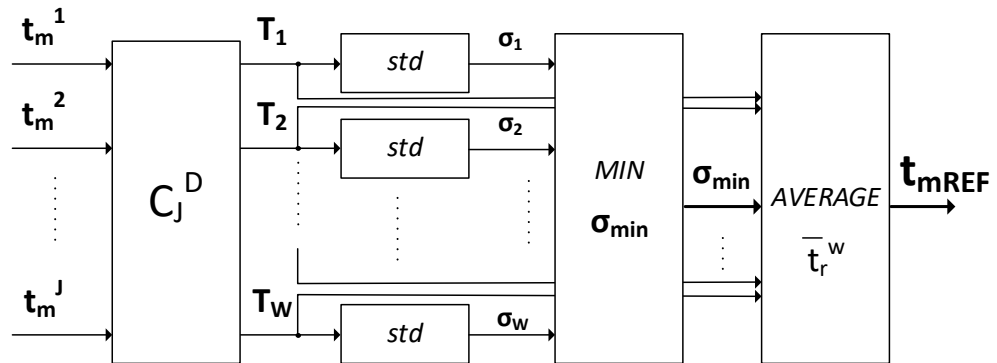


Figura 22. Generación del vector de referencia.

- **Coefficiente de reproducibilidad:** la cuantificación de la semejanza entre el vector de referencia generado t_{mREF} y el tiempo por habitación en un día analizado t_m^j se realiza mediante el coeficiente de reproducibilidad del día j , definido como α_j . El coeficiente se calcula mediante la ecuación (22).

$$\alpha_j = 1.96 \cdot std(t_m^j - t_{mREF}) \quad (22)$$

Donde $std(x)$ denota la desviación estándar de x . El factor 1.96 se utiliza para obtener un intervalo de confianza en la desviación del 95 %. Entonces, cuanto menor es el coeficiente, más se asemeja el día analizado al vector de referencia, lo que significa que el usuario está siguiendo un patrón calculado previamente.

- **Orden de referencia:** en el caso del sistema propuesto otro de los métodos de análisis es el orden en el que el usuario recorre las diferentes habitaciones y para J días se describe el vector \mathbf{o}^j que se muestra en (23). n se corresponde con el número de estancias diferentes que se visitan en un día J .

$$\mathbf{o}^j = [o_1^j \quad \dots \quad o_n^j]_{1 \times n} \quad (23)$$

Para el número total de días J se forma una matriz $\mathbf{O}^j \in \mathfrak{M}_{j \times n}(\mathbb{R})$, en la que cada una de las filas representa la actividad diaria con el vector \mathbf{o}^j . A partir de la matriz \mathbf{O}^j se puede conocer cuál es la rutina de referencia, que es el vector \mathbf{o}^j que más veces se repita dentro de esta. El orden de referencia se muestra en la ecuación (24).

$$\mathbf{o}_{\text{REF}} = [o_1 \quad \dots \quad o_n]_{1 \times n} \quad (24)$$

- **Coefficiente de similitud:** la cuantificación de la semejanza entre el orden de referencia \mathbf{o}_{REF} y el orden que se sigue en un día analizado \mathbf{o}^j se realiza mediante el coeficiente de similitud del día j , definido como β_j . El coeficiente de similitud se calcula a partir de la distancia *Levensthein* [36]. La distancia *Levensthein* entre dos vectores puede definirse como el número mínimo de operaciones (inserciones, eliminaciones o sustituciones) que son necesarios para cambiar un vector por otro. El algoritmo matemático de la distancia *Levensthein* [37] se incluye en la ecuación (25).

$$d(a_i, b_j) = \min \begin{cases} d(a_{i-1}, b_j) + c_I, \\ d(a_i, b_{j-1}) + c_B, \\ d(a_{i-1}, b_{j-1}) + c_S, \end{cases} \quad (25)$$

Se parte de dos cadenas a y b de longitudes n y m . El algoritmo calcula la distancia entre a y b usando una matriz como estructura de datos. La matriz tiene unas dimensiones de $n+1$ por $m+1$. En un paso genérico el algoritmo calcula la distancia $d(a [1...i], b [1...j])$ y para ello se apoya en las distancias de subcadenas más pequeñas. c_I , c_B y c_S son los costes de inserción, borrado y sustitución.

A partir de la distancia *Levensthein* se puede calcular el coeficiente de similitud. El coeficiente de similitud de cada uno de los días j se calcula como la diferencia con la unidad del cociente entre la distancia calculada y la longitud máxima del vector \mathbf{o}_{REF} o \mathbf{o}^j , como se muestra en la ecuación (26).

$$j = 1 - \frac{d(\mathbf{o}_{\text{REF}}, \mathbf{o}^j)}{\text{long}_{\text{max}}(\mathbf{o}_{\text{REF}}, \mathbf{o}^j)} \quad (26)$$

5. Resultados

Este capítulo recoge los resultados de las distintas pruebas que se han llevado a cabo a lo largo del presente TFM utilizando los distintos algoritmos desarrollados en el capítulo previo, con el objetivo de monitorizar a una persona e identificar si su comportamiento sigue una rutina o se aleja de ella. En la primera sección 5.1 se van a presentar los resultados obtenidos a partir de la tecnología WiFi y a continuación, en la sección 5.2, se van a presentar los resultados obtenidos a partir de la tecnología *Bluetooth*.

5.1 Localización y generación de rutinas a partir de señales WiFi

En este apartado se van a desarrollar las pruebas que se han realizado con el sistema WiFi. Dentro de las pruebas que se han realizado se incluyen las pruebas de calibración del sistema, las pruebas de evaluación del sistema de localización WiFi en un entorno real y, por último, la evaluación de las rutinas de comportamiento a partir del sistema propuesto.

5.1.1 Pruebas calibración

Las pruebas de calibración consisten en la realización de dos conjuntos de pruebas, las pruebas de umbral y las pruebas de interferencia.

- **Pruebas umbral:** el objetivo de las pruebas de umbral es fijar un umbral de conexión que cubra la mayor distancia posible a la hora de realizar la localización simbólica. En estas pruebas se va a determinar la distancia a la que se pierde el 100 % de conexión configurando un emisor del sistema en sus tres modos de potencia (baja, media y alta).

El entorno experimental escogido para realizar las pruebas es la Escuela Politécnica Superior (EPS) de la Universidad de Alcalá. Más concretamente, estas pruebas se realizan en el segundo pasillo de la segunda planta de la zona oeste de la escuela. El pasillo es un espacio diáfano en el que no hay ningún tipo de interferencia entre emisor y receptor (figura 23). La colocación del emisor (rojo) y la dirección del recorrido que se va a realizar se muestra en la figura 23. El receptor se coloca en el tobillo del usuario que realiza las pruebas.

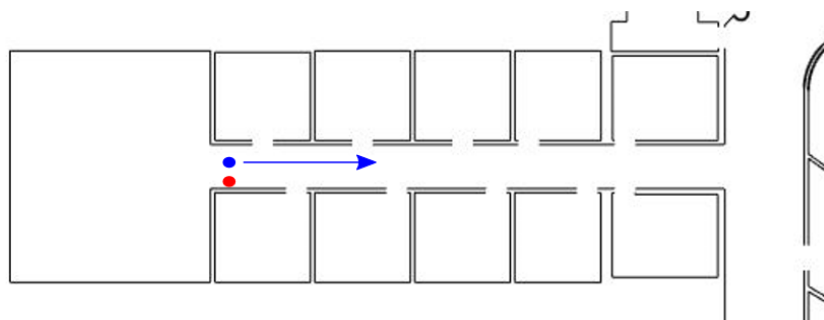


Figura 23. Plano del pasillo donde se realizan las pruebas de umbral. En rojo la posición del emisor y en azul el receptor y su recorrido.

Para conocer la distancia a la que el porcentaje de conexión baja del 100 % se va a emplear el número de zancadas que se han dado hasta ese momento, por lo que en primer lugar se debe conocer la distancia de cada zancada del portador del receptor. Para obtener la distancia de cada zancada se van a medir el número de zancadas que se dan recorriendo una distancia en línea recta de 13.6 metros. Para que la distancia de la zancada sea lo más precisa posible se va a realizar el mismo recorrido en 8 ocasiones, y la distancia final de zancada estimada será la media de todos los recorridos. Los resultados se muestran en la tabla 1, siendo la distancia final de zancada estimada de 1.09 metros.

Número de zancadas (pierna derecha)	Distancia de cada zancada(m)
11.5	1.18
12.5	1.08
12.5	1.08
13	1.04
12	1.13
13	1.04
12	1.13
13	1.04

Tabla 1. Número de zancadas y distancia de cada zancada para cada uno de los recorridos

Una vez conocida la distancia por zancada, se puede conocer la distancia a la que la conexión baja del 100 % en cada uno de los modos de configuración de potencia. Al número de zancadas de cada una de las pruebas se le debe sumar una ya que en todas las pruebas se comienza con la pierna izquierda adelantada. Para cada uno de los modos de configuración de potencia se realizan 8 pruebas. Los resultados de una de las pruebas configurando el emisor en potencia baja se muestra en la figura 24, donde en la parte superior se muestra la señal del ángulo de Euler pitch extraída de la unidad de medida inercial que se lleva en el tobillo, y en la parte inferior el porcentaje de conexión registrado en el receptor. Nótese como en la señal de pitch se pueden identificar claramente el número de zancadas antes de que el porcentaje de conexión descienda del 100 %.

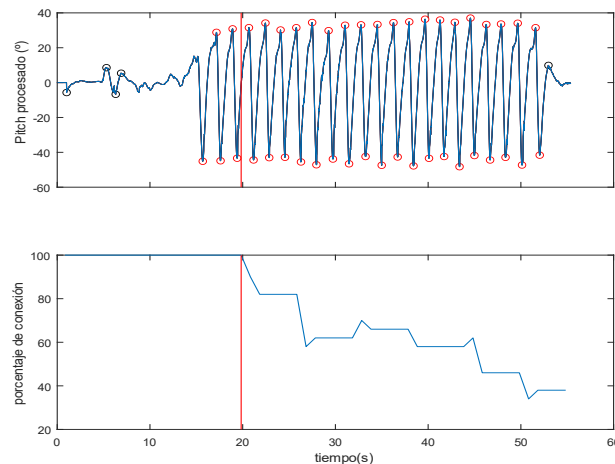


Figura 24. Número de zancadas para una potencia de emisión baja.

Los resultados de las pruebas se recogen en la tabla 2. La distancia media para cada uno de los modos de configuración de potencia es 4.22 m en el caso de la potencia baja, 4.37 m en el caso de la potencia media y 4.44 m en el caso de la potencia alta. El radio que cubre el 100 % de porcentaje de conexión para los tres modos de funcionamiento se muestra en la figura 25, que es aproximadamente de 4.22 m para una potencia baja, 4.37 m para una potencia media y 4.44 m para una potencia alta de emisión.

Potencia baja		Potencia media		Potencia alta	
Zancadas	Distancia(m)	Zancadas	Distancia(m)	Zancadas	Distancia(m)
3.5	3.83	3.5	3.83	3	3.28
4	4.37	4.5	4.92	4	4.37
4.5	4.92	3	3.28	6	6.56
4.5	4.92	1.5	1.64	3	3.28
0	0	5	5.47	5	5.47
5	5,47	7	7.66	3.5	3.83
3.5	3,83	4.5	4.92	3.5	3.83
2	2,18	3	3.28	4.5	4.92

Tabla 2. Distancia para cada uno de los modos de potencia.

Como se observa en la figura 25, la diferencia de distancia es mínima para los tres modos de potencia. La principal diferencia entre los tres modos se encuentra en la forma en la que la señal desciende del 100 % de conexión, que es más abrupta en el caso de la potencia baja. Como la distancia a la que el porcentaje de conexión baja del 100 % es pequeña para todos los modos de configuración de potencia es necesario utilizar un umbral menos restrictivo a la hora de realizar la localización simbólica con el sistema propuesto. Por tanto, se puede considerar que si el porcentaje de conexión se encuentra en torno al 75-80 % se puede considerar que el receptor se encuentra conectado al emisor

correspondiente. Una vez realizadas las pruebas de umbral se van a realizar las pruebas de interferencia.

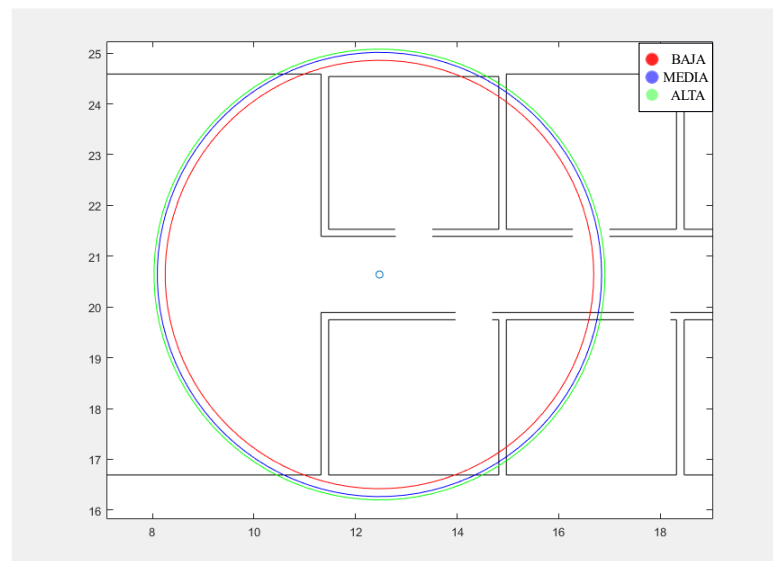


Figura 25. Radios de cobertura de porcentaje de conexión al 100 % de aproximadamente 4.22 m para una potencia baja, 4.37 m para una potencia media y 4.44 para una potencia alta de emisión.

- **Pruebas interferencia:** las pruebas de interferencia son consecuencia de las limitaciones del sistema receptor propuesto. La limitación consiste en que el receptor del sistema solo puede configurarse para conectarse a una determinada SSID por lo que todos los emisores del sistema deben estar configurados con la misma SSID y sus señales pueden verse interferidas. El objetivo de las pruebas de interferencia es conocer a qué distancia deben colocarse los *router* para que el receptor pueda conectarse a ellos sin problema, ya que en ocasiones, si la distancia no es suficiente entre emisores, el receptor sigue conectado al primero de los emisores y no es capaz de conectarse al segundo.

El entorno experimental escogido sigue siendo la Escuela Politécnica Superior de la Universidad de Alcalá. En este caso se va a utilizar el pasillo empleado en las pruebas de umbral, el pasillo 2 de la planta 3 de la zona oeste y el *hall* de la segunda planta de las zonas oeste y norte. Se van a realizar dos tipos de pruebas: las pruebas A consisten en determinar el alcance vertical de los emisores (entre dos plantas), y las pruebas B consisten en determinar el alcance horizontal de los emisores en un espacio diáfano como es el *hall*.

En las pruebas A se realiza el siguiente recorrido: se inicia en el segundo pasillo de la zona oeste de la segunda planta, se sube al segundo pasillo de la zona oeste de la tercera planta y después se vuelve a bajar al pasillo inicial. Dentro de las pruebas A se distinguen dos tipos de pruebas, las A1

donde solo se coloca un *router*, ya sea en la planta 2 o 3, y las A2 donde se coloca un *router* en cada una de las plantas.

En las pruebas A1 primero se coloca el *router* en la planta 2 (configurando el *router* en potencia baja, media y alta) y después en la planta 3. Para cada una de las potencias se realizan 8 pruebas. Con las pruebas A1 se quiere determinar si el receptor es capaz de conectarse al *router* en la planta en la que este no está colocado. Los resultados se muestran en la tabla 3, en la que se muestran el número de conexiones en la planta contraria y el porcentaje respecto al número de pruebas por potencia configurada (8 por potencia). Cuatro de los resultados configurando el *router* en potencia de emisión baja se muestra en la figura 26.

Potencia	Planta 2		Planta 3	
	Conexiones	Porcentaje (%)	Conexiones	Porcentaje (%)
Baja	5	62.5	3	37.5
Media	1	12.5	4	50.0
Alta	4	50.0	6	75.0

Tabla 3. Resultados de las pruebas A1.

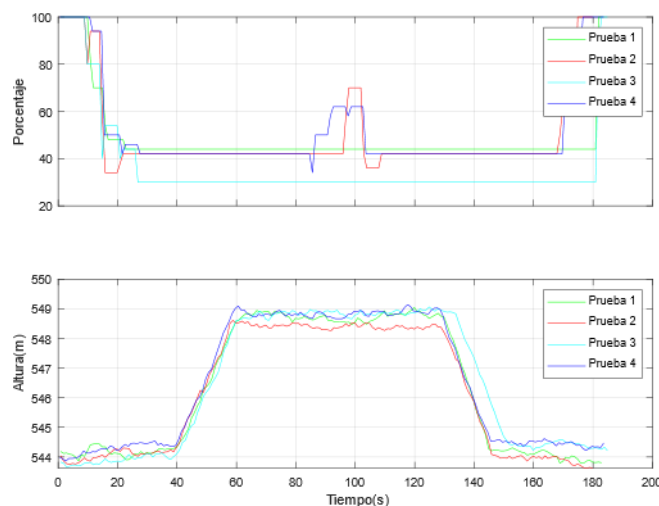


Figura 26. Resultados de 4 pruebas con el *router* colocado en la planta 2 y con potencia de emisión baja.

En las pruebas A1 se muestra que para cualquier modo de emisión el receptor se conecta al *router* estando en la planta contraria, lo que supone un problema para el sistema propuesto.

En las pruebas A2 se coloca un *router* en cada una de las plantas, realizando 8 pruebas para cada una de las combinaciones posibles de potencia de emisión. Con las pruebas A2 se quiere determinar si el receptor del sistema es capaz de conectarse a los *router* de ambas plantas. Los resultados se muestran en la tabla 4, en la que los errores de conexión representan el número de veces en las que el receptor no se ha conectado al *router* de la planta en la que se encuentra y el porcentaje representa el número de errores respecto al número total de conexiones posibles (3 por cada una de las pruebas y 24 para cada una de las configuraciones).

Además, cuatro de los resultados con ambos *router* configurados en potencia media se muestran en la figura 27.

Potencia		Errores de conexión	Porcentaje (%)
Planta 2	Planta 3		
Baja	Baja	1	4.16
Media	Baja	3	12.50
Alta	Baja	3	12.50
Baja	Media	2	8.33
Media	Media	0	0.00
Alta	Media	1	4.16
Baja	Alta	1	4.16
Media	Alta	1	4.16
Alta	Alta	2	8.33

Tabla 4. Resultados de las pruebas A2.

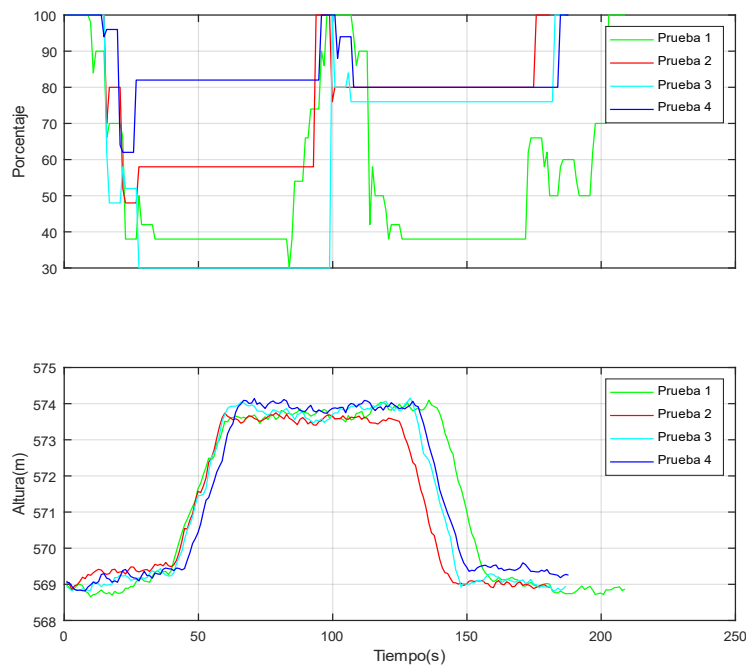


Figura 27. Resultados de 4 pruebas con ambos *router* configurados en potencia media.

A la vista de los resultados de la tabla se muestra que los errores de conexión son pequeños para cualquiera de las combinaciones, obteniéndose los mejores resultados cuando ambos *router* se configuran en potencia media. Por tanto, a partir de ahora todos los resultados son obtenidos con una configuración de potencia media de todos los *router*.

En las pruebas B los *router* se van a colocar en una sala diáfana, como se muestra en la figura 28, que se corresponde con el *hall* de la segunda planta del oeste de la EPS. En primer lugar, se van a probar dos distancias,

que son 16.8 metros en el caso del recorrido 1-2 y 24 metros en el caso del recorrido 1-3. En todas las pruebas se va a realizar el mismo recorrido, se parte del primer *router* hasta que se llega al segundo y después se vuelve al primero. Para cada una de las distancias se realizan 8 pruebas.

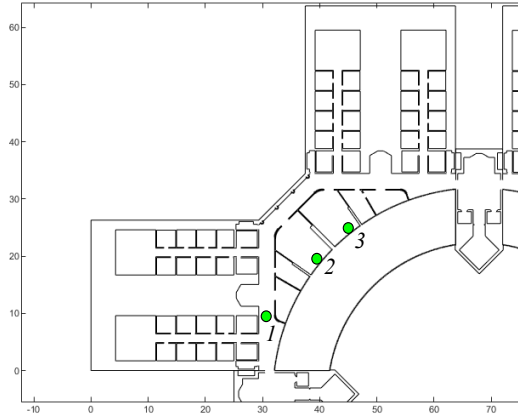


Figura 28. Posición de los *router* en las pruebas B de interferencia en el hall de la 2ª planta del ala oeste de la EPS de la UAH.

Los resultados de las pruebas B se recogen en la tabla 5. Los errores de conexión son el número de veces en las que el receptor no se conecta al *router* más cercano y el porcentaje representa el número de errores respecto al número total de conexiones (3 por cada prueba y 24 por distancia). 4 de los resultados con los *router* colocados en las posiciones 1 y 3 se muestran en la figura 29.

Posición <i>router</i>	Errores de conexión	Porcentaje (%)
1-2	4	16.67
1-3	5	20.08

Tabla 5. Resultados de las pruebas B.

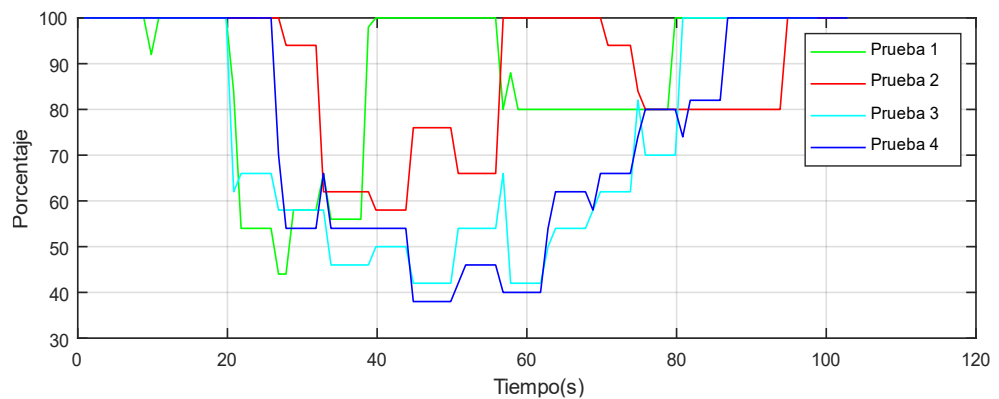


Figura 29. Pruebas B de interferencia realizadas con los *router* colocados en las posiciones 1 y 3 del hall de la 2ª planta del ala oeste de la EPS de la UAH.

Como los resultados obtenidos no han sido del todo satisfactorios se va a realizar el mismo tipo de prueba, pero para tres distancias más alejadas.

Los nuevos puntos escogidos se muestran en la figura 30. Los resultados obtenidos se muestran en la tabla 6.

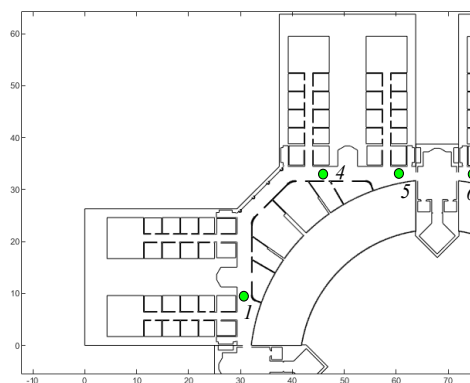


Figura 30. Nuevas posiciones de los router en las pruebas B de interferencia dentro del hall de la 2ª planta del ala oeste de la EPS de la UAH.

Posición router	Errores de conexión	Porcentaje (%)
1-4	4	16.67
1-5	4	16,67
1-6	0	0,00

Tabla 6. Resultados obtenidos para las nuevas posiciones de los router.

En las posiciones 1-4 y 1-5 se siguen repitiendo los errores de conexión ya que ambos se siguen encontrando en la misma sala, aunque a mayor distancia. En la posición 6 el router se encuentra en el hall de la zona norte del edificio, por lo que ya hay obstáculos entre ambos router lo que hace que los resultados mejoren. A partir de las pruebas B se determina que colocar ambos router dentro de una misma sala diáfana puede suponer un problema de interferencia entre ellos por lo que se opta por colocar los router en habitaciones diferentes.

5.1.2 Pruebas de evaluación del sistema WiFi en un entorno real

Una vez realizadas las pruebas de calibración en las que se han determinado el umbral de conexión a utilizar y la distancia a la que se deben encontrar los emisores del sistema para que no se vean interferidos entre ellos, se va a evaluar el sistema de localización WiFi dentro de un entorno real. Además, se va a evaluar la viabilidad del sistema para la posible detección de rutinas de comportamiento.

El entorno experimental escogido para la evaluación del sistema WiFi es el centro sociosanitario del grupo Albertia Las Palmeras, situado en Azuqueca de Henares, España. Los routers WiFi se colocan en estancias significativas para los pacientes del mismo centro, además estas estancias deben estar lo suficientemente alejadas para que no haya problemas de interferencia entre emisores. Las estancias escogidas son el comedor, el gimnasio y la habitación

de uno de los pacientes del centro. El receptor NGIMU se coloca en el tobillo del paciente. La figura 31 muestra esquemáticamente el plano y la ubicación en planta de cada una de las estancias escogidas en las que están colocados los *routers*.

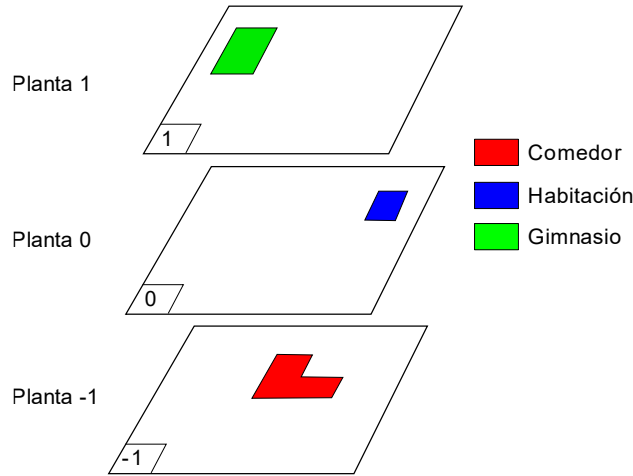


Figura 31. Ubicación de cada una de las estancias monitorizadas en el centro sociosanitario Las Palmeras.

Para el entorno escogido es necesario determinar los umbrales de presión para el algoritmo de detección de cambio de planta, el umbral de conexión para el algoritmo de localización simbólica, el umbral de pasos del algoritmo de fusión sensorial y, por último, la ventana de tiempo empleada en este mismo algoritmo. El valor del umbral de pasos y del Δt que se utilizan en el algoritmo de fusión sensorial dependen del grado de fragilidad de la persona monitorizada que revertirá directamente en su grado de movilidad.

El centro sociosanitario tiene 4 plantas, por tanto, los umbrales que se definen para el algoritmo de detección de cambio de planta se muestran en (27). U_1 , u_2 y u_3 son respectivamente las diferencias de presión que se producir para hacer el cambio entre 1, 2 y 3 plantas.

$$u_1 = 0.2 Pa; u_2 = 0.5 Pa; u_3 = 0.9 Pa; \quad (27)$$

Por otro lado, se debe definir el umbral de conexión u_c que utiliza el algoritmo de localización simbólica y, el umbral de conexión u_{c2} y el umbral de pasos u_p que utiliza el algoritmo de fusión sensorial. Los valores de estos umbrales se definen en (28).

$$u_c = 75 \%; u_{c2} = 45\%; u_p = 10 \text{ pasos}; \quad (28)$$

Por último, el Δt que se utiliza para determinar la ventana de tiempo a analizar por el algoritmo de fusión sensorial es de 300 s.

En el centro se han llevado a cabo dos tipos de pruebas que consistían en realizar recorridos por el centro pasando por las estancias definidas previamente. La primera serie (pruebas A) fue realizada por un voluntario de 25 años (robusto) un único día. La segunda serie (pruebas B) fue realizada por un paciente de 90 años (frágil) del centro sociosanitario durante 12 días

con el objetivo de validar el sistema propuesto. La voluntaria necesita andador para los desplazamientos por lo que se asegura que todos los desplazamientos entre plantas tendrán lugar por ascensor.

Las pruebas A, realizadas por un voluntario en un único día, se realizaron siguiendo un trazado preestablecido para así tener el *ground truth*. Como ejemplo se muestran los resultados de una de las pruebas en la figura 32, cuyo recorrido fue: habitación, comedor, habitación, gimnasio y habitación. Se puede observar el número de pasos por segundo, el porcentaje de conexión WiFi y la presión atmosférica medida. El *ground truth* se ha representado superpuesto por una línea negra. A partir de estos datos y los algoritmos propuestos, es posible tener información acerca de la localización del paciente: la planta y la habitación (si esta se identifica con un nodo WiFi).

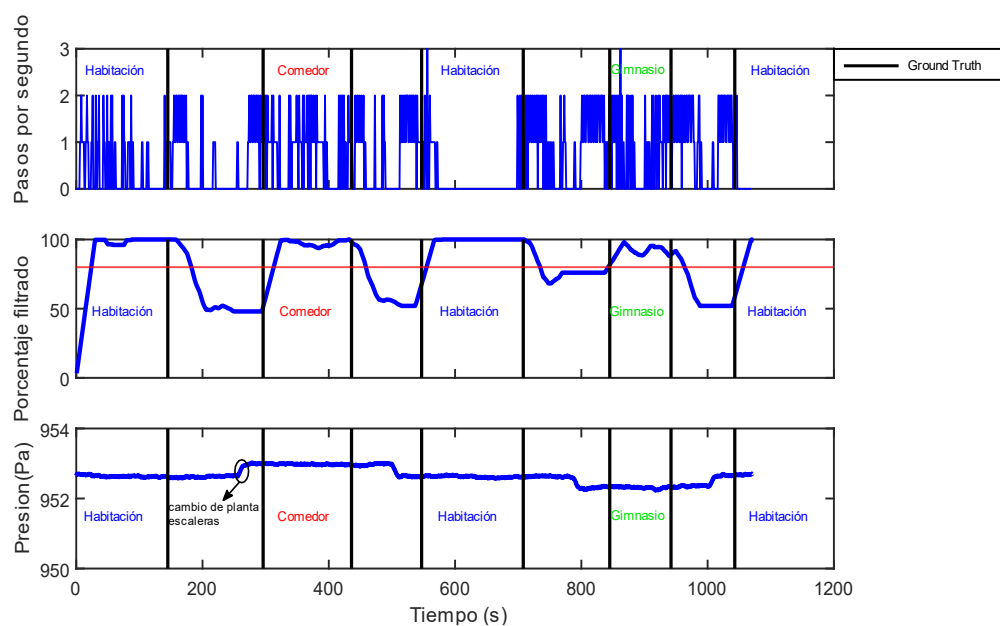


Figura 32. Número de pasos, porcentaje de conexión y presión frente al *ground truth*.

La figura 33 muestra la diferencia entre el *ground truth* y el resultado del sistema propuesto para los datos de la figura anterior. La diferencia se debe principalmente al tiempo que el receptor requiere para conectarse/desconectarse de los diferentes nodos WiFi, lo que provoca un retraso en la señal de localización como se muestra en la figura 33. El máximo retardo de sincronización es de 39 segundos.

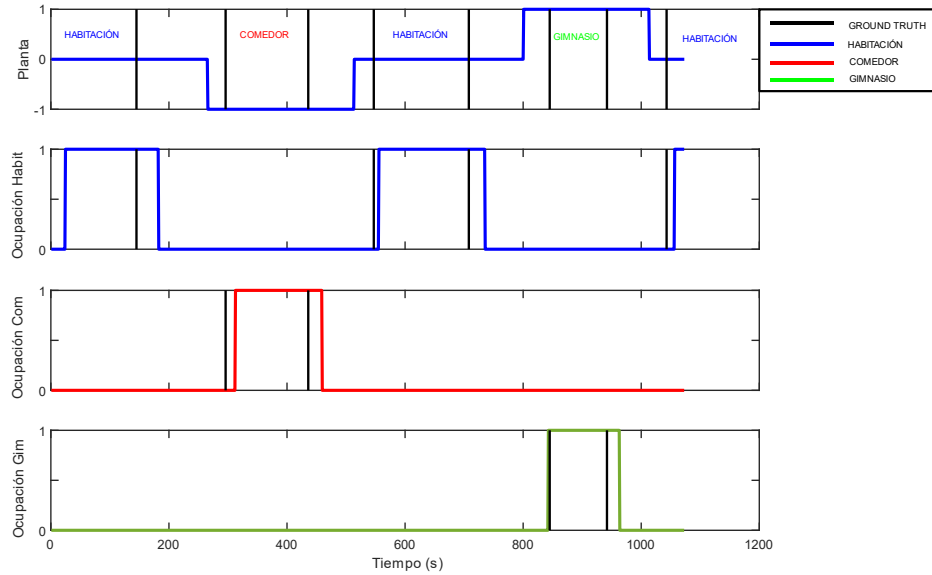


Figura 33. Número de planta y habitación frente al ground truth.

Para las pruebas A la tasa de acierto en la estimación de la localización simbólica es del 95,6 % estimada como la diferencia del *ground truth* con los resultados del sistema, sin considerar el retardo en las transiciones. Ese pequeño porcentaje de error se debe a que en ocasiones el receptor no es capaz de conectarse a alguno de los nodos WiFi. Por tanto, en estos casos se puede saber en qué planta se localiza el portador del receptor, pero no en qué habitación se encuentra. El alto porcentaje de acierto justifica su viabilidad para que el sistema pueda ser utilizado por un paciente del propio centro.

Las pruebas B se realizaron con un voluntario de la residencia durante 12 días para evaluar el uso del sistema para poder determinar posibles rutinas de comportamiento. Aunque las pruebas B se realizan durante 12 días, la figura 34 solo muestra una parte de los resultados representativos que se corresponden con 4 horas de actividades realizadas por la mañana correspondiéndose con 6 días diferentes.

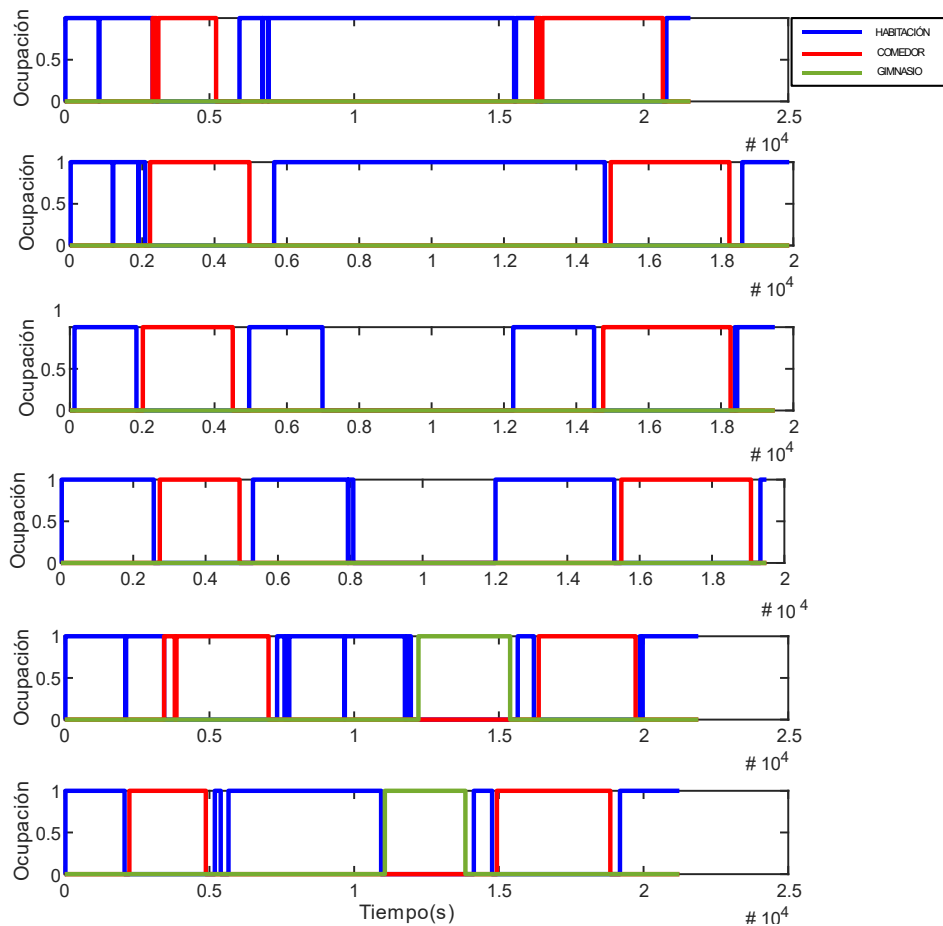


Figura 34. Datos representativos de la ocupación de las diferentes estancias a lo largo de las diferentes mañanas.

Las gráficas muestran la ocupación de las distintas estancias con nodo WiFi: habitación, comedor y gimnasio. A partir de la ocupación mostrada en la figura se podría elaborar una rutina de referencia, o comportamiento basal, para este paciente para luego poder comparar su actividad en el tiempo. Sin embargo, será imposible realizar una rutina detallada ya que solo al tener tres habitaciones monitorizadas habrá actividades que queden indeterminadas porque las salas donde se realizan no disponen de *router* identificativo. En este caso que se muestra en la figura 34, la secuencia típica seguida por la mañana que se puede asegurar es: habitación, comedor, habitación, actividad indeterminada, habitación, comedor y habitación.

5.1.3 Pruebas de evaluación de rutina de comportamiento

El objetivo de las pruebas de evaluación de rutinas de comportamiento es la monitorización de dos usuarios dentro de un mismo entorno para obtener posibles rutinas de comportamiento.

El entorno escogido para las pruebas de evaluación de rutina es la EPS, al igual que en el caso de las pruebas de calibración. En este caso los emisores del sistema se colocan en estancias significativas para la vida diaria de ambos voluntarios. Los *router* se colocan en el laboratorio del pasillo 2 de la segunda planta de la zona oeste (laboratorio 22, planta 2), el laboratorio del pasillo 1 de la zona oeste (laboratorio 31, planta 3) y en la cafetería del edificio (planta 0). Al igual que en el caso del centro sociosanitario, el receptor del sistema se coloca en el tobillo, además se deben determinar los umbrales para el algoritmo de detección de cambios de planta y para los algoritmos de localización simbólica y fusión sensorial.

La escuela de Ingeniería tiene 4 plantas, y tras unas pruebas los umbrales que se utilizan para la detección de cambios de planta son los que se muestran en (29).

$$u_1 = 0.2 Pa; u_2 = 0.7 Pa; u_3 = 1.3 Pa; \quad (29)$$

Por otro lado, se debe definir el umbral de conexión u_c que utiliza el algoritmo de localización simbólica el umbral de conexión u_{c2} y el umbral de pasos u_p que utiliza el algoritmo de fusión sensorial. En este caso el umbral de conexión u_{c2} también se emplea en el algoritmo de localización simbólica para una de las habitaciones, que es la cafetería. El umbral que se utiliza en la cafetería es diferente porque es una sala de gran tamaño y su valor debe ser más pequeño que en el caso de las demás habitaciones. Los valores de estos umbrales se definen en (30).

$$u_c = 75 \% ; u_{c2} = 50 \% ; u_p = 10 \text{ pasos}; \quad (30)$$

Por último, el Δt que se utiliza para determinar la ventana de tiempo a analizar por el algoritmo de fusión sensorial es de 60 s.

Las pruebas consisten en la monitorización de dos usuarios dentro del entorno experimental durante 4 horas cada día. En este caso sí que se va a determinar si la transición entre habitaciones se realiza por el ascensor o por las escaleras.

Como consecuencia de la crisis sanitario del COVID-19 al voluntario 1 únicamente se le pudo monitorizar durante 3 días y al voluntario 2 durante 4. Como alternativa se generan pruebas sintéticas a partir de las pruebas reales para llegar a conseguir 20 días para cada uno de los voluntarios y así poder emplear la algoritmia de generación de rutinas. La algoritmia de

generación de rutinas se basa en un único elemento, que es el tiempo que pasa el usuario en cada una de las habitaciones monitorizadas.

En la figura 35 se muestran 10 de los resultados de la monitorización del voluntario 1. En negro aparecen aquellos momentos en los que el sistema no es capaz de determinar la ubicación del usuario.

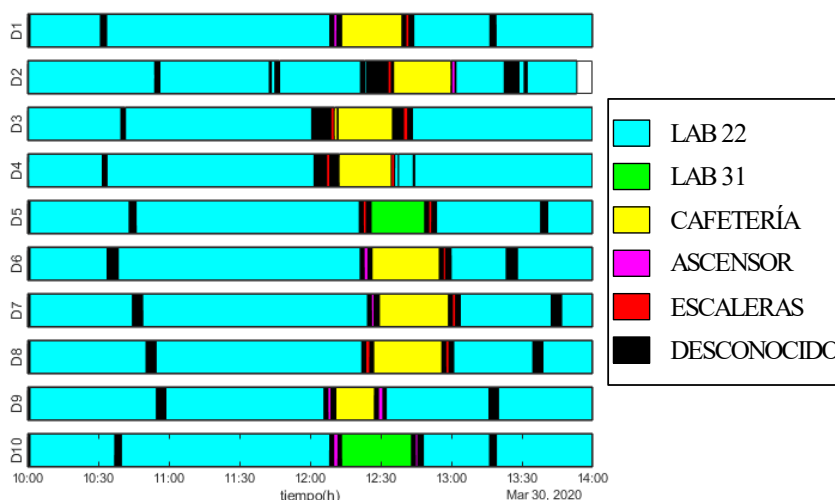


Figura 35. Resultados de la localización del voluntario 1.

A partir de los resultados se puede obtener el número de minutos para cada una de las habitaciones y así calcular el vector de referencia, t_{mref} . Para el cálculo del vector se consideran 20 días, y cada uno de los subconjuntos está compuesto por 8 días, lo que significa que se forman 125970 conjuntos de 8 días ($J=20, D=8, W=125970$). La tabla 7 muestra los valores del vector de referencia del voluntario 1.

Vector de referencia	
	Valor de referencia (min) ± STD
LAB 22	193,9 ± 2,5
LAB 31	0,0 ± 0,0
CAFETERÍA	24,7 ± 1,9
ASCENSOR	1,3 ± 0,1
ESCALERAS	1,3 ± 0,2
DESCONOCIDO	17,1 ± 0,32
$t_{mREF} = [193,9 \ 0,0 \ 24,7 \ 1,3 \ 1,3 \ 17,1]$	

Tabla 7. Vector de referencia del voluntario 1.

La evaluación de la actividad diaria se va a realizar mediante el uso del coeficiente de reproducibilidad diario (α_j). En la figura 36 se muestra el coeficiente de reproducibilidad para los 20 días monitorizados.

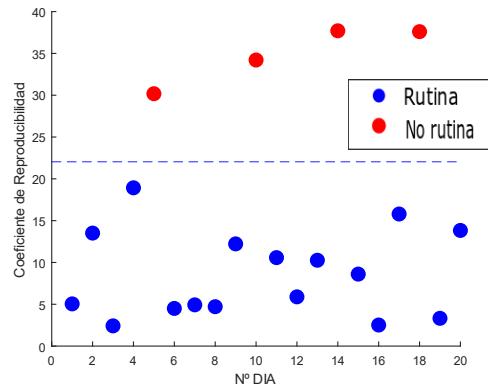


Figura 36. Coeficiente de reproducibilidad para cada uno de los días del voluntario 1.

En la figura es posible diferenciar los días en los que se sigue la rutina de referencia (azul) y los días en los que no se ha seguido (rojo). El criterio para considerar un día rutinario es que el coeficiente se encuentre por debajo de 22.4, que se corresponde con la desviación del 15 % respecto al tiempo total en cada habitación.

Con el voluntario 2 se sigue el mismo procedimiento. Diez de los resultados de la monitorización del voluntario 2 se muestran en la figura 37.

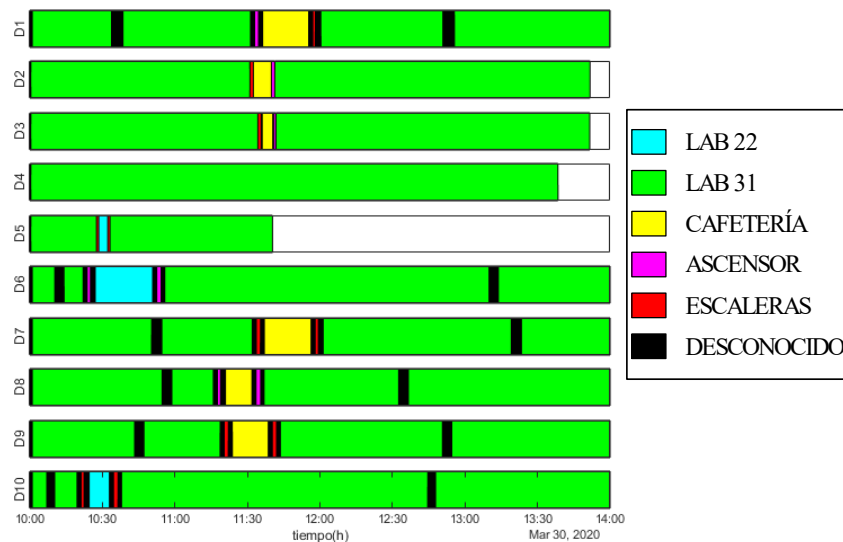


Figura 37. Resultados de la localización simbólica del voluntario 2.

El vector de referencia en el caso del voluntario 2 se muestra en la tabla 8 y el cálculo del coeficiente de reproducibilidad se incluye en la figura 38.

Vector de referencia	
	Valor de referencia (min) ± STD
LAB 22	0,0 ± 0,0
LAB 31	213,4 ± 1,7
CAFETERÍA	12,6 ± 2,2
ASCENSOR	1,6 ± 0,2
ESCALERAS	0,0 ± 0,0
DESCONOCIDO	12,9 ± 0,84
$t_{mREF} = [0,0 \ 213,4 \ 12,6 \ 1,6 \ 0,0 \ 12,9]$	

Tabla 8. Vector de referencia voluntario 2

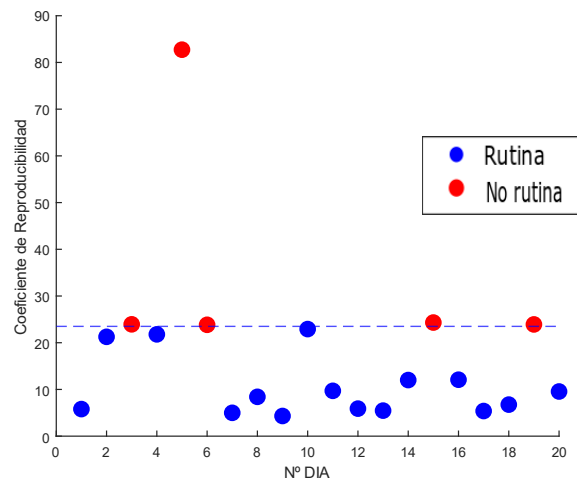


Figura 38. Coeficientes de reproducibilidad del voluntario 2.

En la figura es posible diferenciar los días en los que se sigue la rutina de referencia (azul) y los días en los que no se ha seguido (rojo). El criterio para considerar un día rutinario es que el coeficiente se encuentre por debajo de 25, que se corresponde con la desviación del 15 % respecto al tiempo total en cada habitación.

5.2 Localización y generación de rutinas a partir de señales *Bluetooth*

En este apartado se muestran los resultados de las pruebas que se han realizado con el sistema *Bluetooth*. Dentro de las pruebas que se han realizado se incluyen las pruebas de calibración del sistema, las pruebas de validación del sistema en un entorno real y las pruebas de evaluación de rutinas de comportamiento.

5.2.1 Pruebas de calibración del sistema.

El objetivo de las pruebas de calibración del sistema es determinar la potencia de emisión de los emisores con la que se consiguen los mejores resultados, determinar el tiempo mínimo que se debe estar en una habitación para que el sistema sea capaz de identificar que se ha estado en ella, evaluar diferentes posiciones del receptor para determinar en cual se consiguen mejores resultados y por último determinar el comportamiento del sistema en pruebas de larga duración. Por tanto, dentro de las pruebas de calibración se incluyen las pruebas de potencia, las pruebas de tiempo, las pruebas de posición y las pruebas de larga duración.

El entorno escogido es el domicilio del usuario que se va a monitorizar. El domicilio donde se van a realizar las pruebas consta de 6 habitaciones, que son la cocina, el comedor, el baño y tres habitaciones. Las habitaciones se diferencian como habitación 1,2 y 3. En cada una de las habitaciones se coloca una única baliza. El plano donde se muestran las habitaciones y la posición aproximada de las balizas se muestra en la figura 39.

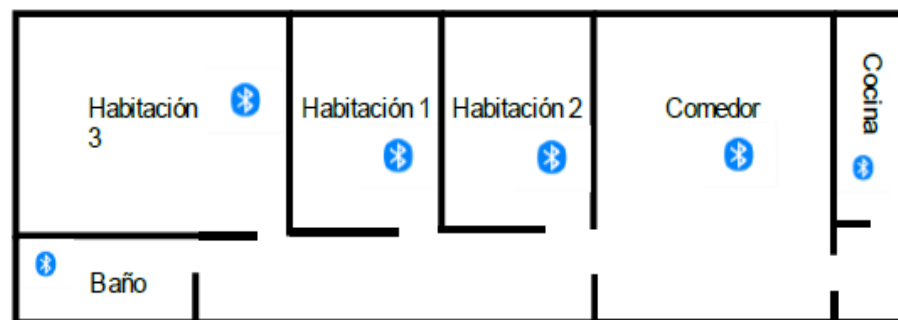


Figura 39. Habitaciones del domicilio y posición de las balizas

- Pruebas de potencia:** las pruebas de potencia consisten en evaluar la exactitud del sistema configurando la potencia de emisión de las balizas a diferentes niveles. Los niveles de potencia a los que se configuran las balizas son 0, -4, -8 y -12 dBm. Inicialmente es necesaria una fase de entrenamiento para cada una de las potencias de emisión, para ello se van a evaluar los modelos KNN, SVM y DT a partir de los datos de entrenamiento. El entrenamiento de los modelos consiste en tomar 5 medidas en 10 puntos diferentes de las habitaciones que se muestran en

la figura 39, por lo que para cada uno de los modelos se toman 300 medidas, 50 por habitación. La exactitud a la hora de crear los modelos con los datos de entrenamiento se muestra en la tabla 9.

Potencia	Exactitud (%)		
	SVM	DT	KNN
0	100	98,7	100
-4	100	99,3	100
-8	100	99,7	100
-12	100	98,7	100

Tabla 9. Exactitud de los modelos a partir de los datos de entrenamiento.

El modelo que se escoge para las pruebas de potencia es el modelo KNN. Una vez creado el modelo se realizan las pruebas de potencia, para cada una de las potencias se van a realizar tres pruebas iguales. El recorrido que se sigue en cada prueba es habitación 3, baño, habitación 1, habitación 2, baño y cocina. En cada una de las habitaciones el usuario monitorizado pasa 30 segundos, siendo la transición entre habitaciones de 10 segundos salvo la última transición que es de 20 segundos. Una de las pruebas realizadas con los emisores configurados a una potencia emisión de -8 dBm se muestra en la figura 40.

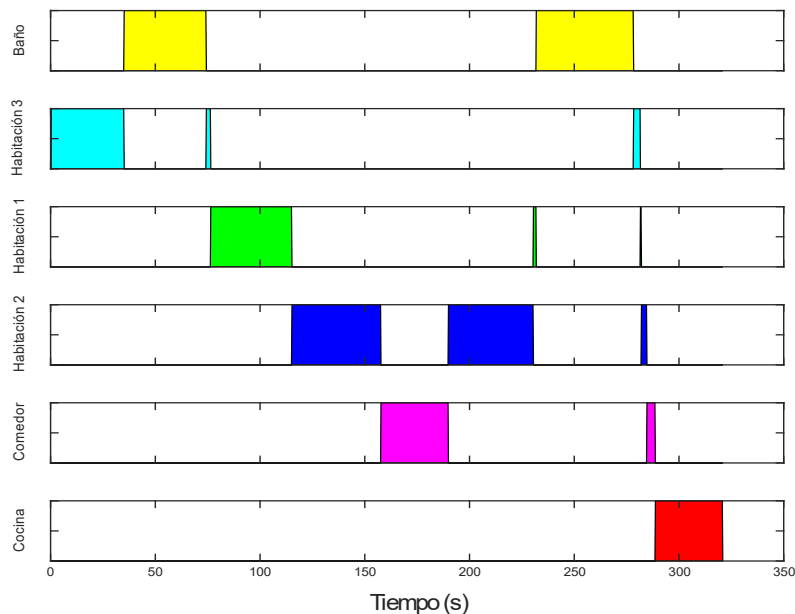


Figura 40. Resultados de una de las pruebas con nivel de potencia de emisión de -8 dBm.

La evaluación de cada una de las potencias de emisión se realiza mediante las matrices de confusión. Las matrices de confusión permiten evaluar la exactitud de los modelos que se utilizan. La exactitud de cada uno de los modelos (para diferentes potencias) se muestra en la tabla 10.

Potencia	Exactitud (%)
0	92,1
-4	99,2
-8	99,3
-12	99,2

Tabla 10. Exactitud de los modelos frente a diferentes niveles de potencia.

Los mejores resultados se obtienen configurando la potencia de las balizas a -8 dBm aunque con las potencias de -4 y -12 también se obtienen buenos resultados. Para las pruebas de tiempo y posición se va a utilizar el modelo KNN con el nivel de potencia de emisión de los emisores configurado a -8 dBm.

- Pruebas de tiempo:** en las pruebas de tiempo se va a comprobar cuál es el tiempo mínimo que el usuario debe encontrarse en cada una de las habitaciones para que el sistema sea capaz de identificar que ha estado en ella. En las pruebas de potencia el voluntario pasa en cada una de las estancias 30 segundos y la exactitud en la localización se encuentra por encima del 99 %, por tanto, se va a comprobar si con 20 y 10 segundos por habitación se consiguen resultados similares de exactitud. Se van a realizar tres pruebas por tiempo en cada de las habitaciones siendo el recorrido que se va a realizar habitación 2, habitación 1, baño, habitación 3, habitación 1, habitación 2, comedor y cocina. Las transiciones entre estancias son de 10 segundos. Uno de los resultados pasando 20 segundos en cada una de las estancias del recorrido se muestra en la figura 41.

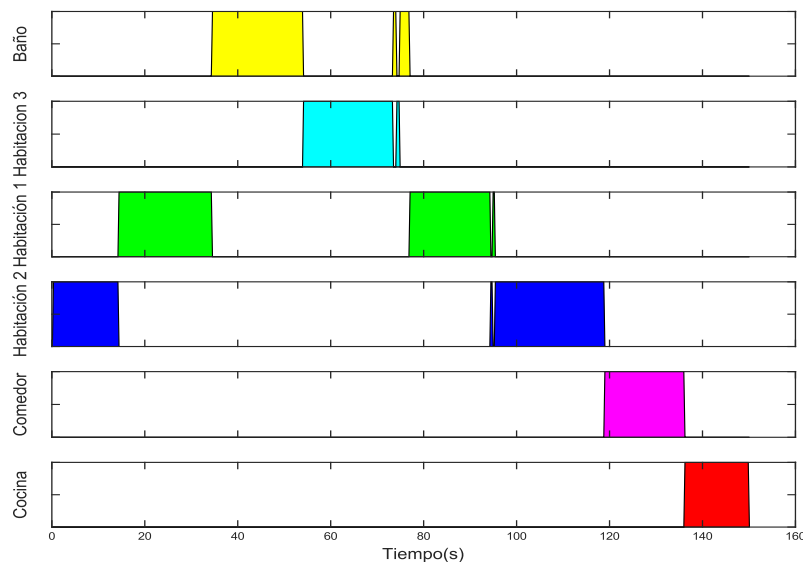


Figura 41. Resultados de una de las pruebas de 20 segundos.

La exactitud obtenida en las pruebas de tiempo se muestra en la tabla 11. A partir de los resultados de la tabla se demuestra que el sistema es capaz de identificar que el usuario se encuentra en una habitación incluso estando solo 10 segundos en ella.

Tiempo por habitación (s)	Exactitud (%)
20	97,8
10	98,5

Tabla 11. Exactitud de las pruebas de tiempo.

- Pruebas de posición:** las pruebas de posición consisten en evaluar el sistema para tres posiciones del receptor. Las posiciones que se van a utilizar son una posición fija dentro de cada estancia, el receptor colocado en el bolsillo y por último el receptor colocado en una sujeción en el brazo. En este caso el recorrido que se va a realizar es el mismo que se ha realizado en las pruebas de potencia. Uno de los resultados con el receptor colocado en la sujeción del brazo se muestra en la figura 42.

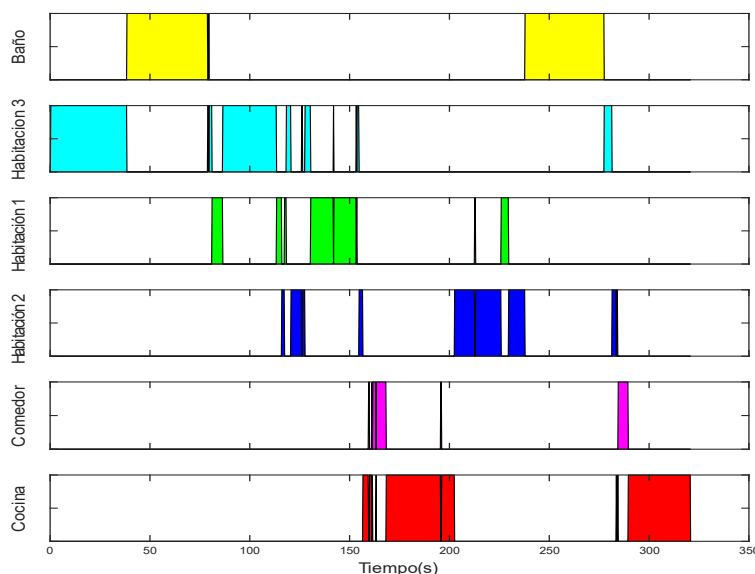


Figura 42. Resultado de una de las pruebas con el receptor situado en la sujeción del brazo.

A la vista de los resultados se muestra que la posición del receptor puede afectar considerablemente a la exactitud del sistema. La exactitud en función de la posición del receptor se recoge en la tabla 12.

Posición	Exactitud (%)
Fija	98,9
Bolsillo	83,4
Sujeción	66,5

Tabla 12. Exactitud frente a la posición del receptor.

A la vista de los resultados de la tabla se puede determinar que la posición del receptor afecta negativamente a la exactitud del sistema propuesto, bajando esta desde el 98,9 % hasta el 66,5 %. Esto se debe a que a la hora de entrenar el sistema no se han tenido en cuenta las distintas posiciones en las que se puede colocar el receptor, ya que todos los datos de entrenamiento se recogen en distintos puntos de las habitaciones con el receptor en la mano del usuario.

Para solucionar este problema se propone entrenar 5 modelos diferentes, cada uno de ellos se corresponde con una posición diferente del receptor. Además, se va a crear un modelo que incluya los 5 modelos individuales y se van a comparar los resultados. Los modelos que se van a crear son el modelo de mano, el de bolsillo delantero, el de bolsillo trasero, el de sujeción de brazo y el de sujeción de cuello.

Para generar los modelos es necesario realizar una fase de entrenamiento para cada uno de ellos. La fase de entrenamiento consiste en la medición de la potencia de las balizas bluetooth en cada una de las habitaciones durante 60 segundos. Una vez creados tanto los modelos individuales como el modelo completo se van a realizar tres pruebas para cada una de las posiciones del receptor. En todas las pruebas se sigue el mismo recorrido, que es cocina, comedor, habitación 2, habitación 1, habitación 2, habitación 3 y baño. En cada una de las habitaciones el usuario pasa 30 segundos y la transición entre ellas es de 10 segundos.

Para comprobar la exactitud de los modelos individuales y del modelo completo se van a volver a utilizar las matrices de confusión. Las exactitudes obtenidas en función de la posición del receptor se muestran en la tabla 13.

Posición	Exactitud Modelo (%)	
	Individual	Completo
Mano	97,9	97,9
Bolsillo delantero	97,1	97,3
Bolsillo trasero	88,8	87,8
Sujeción brazo	84,8	82,4
Sujeción cuello	98,4	98,4

Tabla 13. Comparación de los modelos individuales frente al modelo completo.

A la vista de los resultados de la tabla se justifica el uso del modelo completo ya que los resultados que se obtienen son similares. Con el uso de los nuevos modelos se ha conseguido mejorar la precisión del modelo más desfavorable (Sujeción brazo) del 66,5 % al 84,8 %.

- Pruebas de larga duración:** una vez comprobado el modelo completo en diferentes posiciones del receptor se va a comprobar el funcionamiento del sistema en pruebas de larga duración. El objetivo final del sistema propuesto es conseguir monitorizar al usuario durante su vida diaria. Se van a realizar tres pruebas de larga duración en las que el usuario sigue el mismo recorrido. El recorrido realizado es habitación 2 (30 minutos), baño (10 minutos), habitación 2 (20 minutos), cocina (20 minutos) y habitación 2 (40 minutos). El recorrido pretende simular 2 horas de la vida diaria del usuario monitorizado. Uno de los resultados de las pruebas de larga duración se muestra en la figura 43.

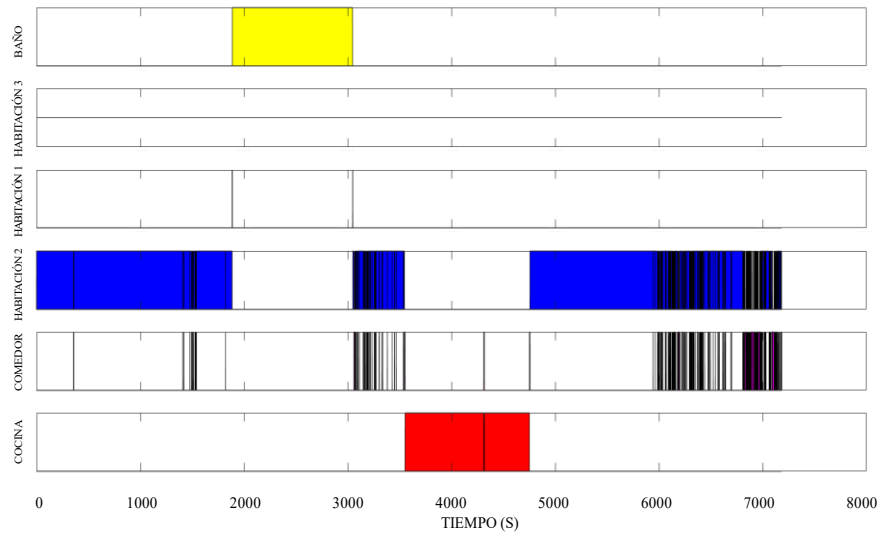


Figura 43. Resultados de una de las pruebas de larga duración.

En la figura se muestra como la variabilidad de la señal bluetooth puede afectar a la exactitud del sistema en pruebas de larga duración. Para la mejora de los resultados se propone el uso del algoritmo de fusión sensorial para corregir los posibles errores en el clasificador de ML. Los resultados de la figura anterior mejorados con el algoritmo de fusión sensorial se muestran en la figura 44.

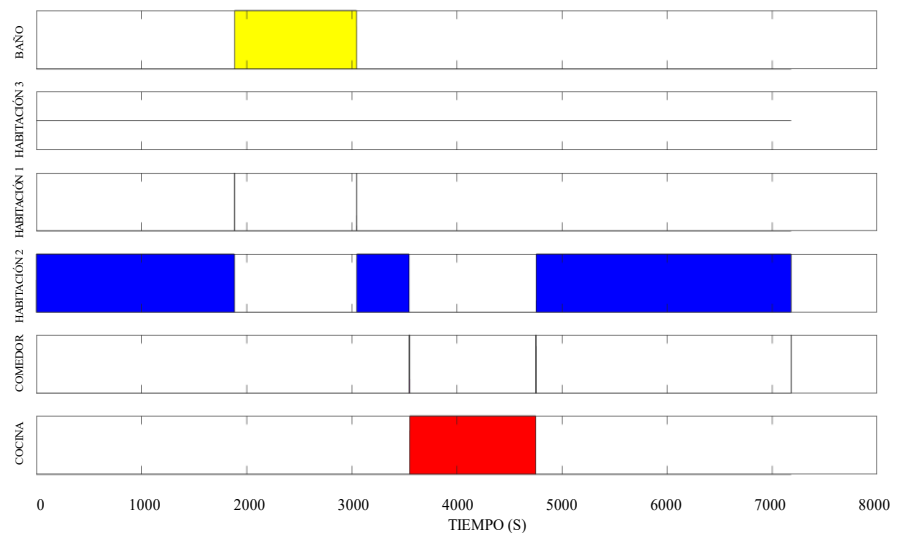


Figura 44. Resultados del sistema utilizando el algoritmo de fusión sensorial.

A la vista de la figura se observa como el uso del algoritmo de fusión sensorial mejora los resultados de localización. Para cuantificar esa mejora se ha comparado la exactitud del sistema utilizando o no el algoritmo de fusión sensorial. La exactitud sin utilizar el algoritmo de fusión es del 83,4 % y utilizando el algoritmo de fusión es del 91,2 % lo que justifica el uso del algoritmo de fusión sensorial para corregir los errores en las pruebas de larga duración.

5.2.2 Pruebas de validación del sistema.

El objetivo de las pruebas de validación es utilizar como referencia los resultados de las pruebas de calibración para validar el sistema en un entorno diferente al entorno inicial. Es necesario validar el sistema en este nuevo entorno ya que es en el que se van a evaluar las rutinas de comportamiento, por lo que la exactitud obtenida en las pruebas de validación debe ser alta para justificar su uso.

El nuevo entorno en el que se van a realizar las pruebas es la EPS de la Universidad de Alcalá, particularmente un pasillo con varias oficinas, siendo equivalentes cada oficina como una habitación de un hogar. En el entorno se seleccionan 6 habitaciones y en cada una de ellas se coloca una baliza. El nuevo entorno experimental se muestra en la figura 45.

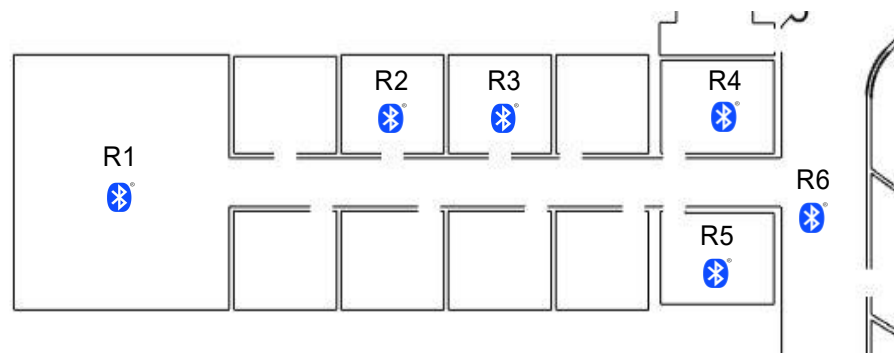


Figura 45. Mapa del nuevo entorno experimental y de la localización de las balizas.

En este entorno se van a realizar una serie de pruebas de validación previas al análisis de las rutinas ya que como se ha mencionado anteriormente la exactitud del sistema debe ser alta para que nos permita estimar las rutinas de comportamiento con gran precisión.

Previo a la validación del sistema es necesaria una fase de entrenamiento. El modelo se entrena en las 5 posiciones del receptor que se han considerado en las pruebas de calibración (mano, bolsillo delantero, bolsillo trasero, sujeción brazo y sujeción cuello). Para cada una de las posiciones se recogen datos en todas las habitaciones durante 1 minuto salvo en la habitación R1 en la que se recogen datos durante 2 minutos al tratarse de una habitación de mayor tamaño. Se van a evaluar tres modelos de ML diferentes para la estimación de la localización: DT, SVM y KNN.

Una vez creados los modelos se van a realizar las pruebas de validación de estos. Durante las pruebas de validación el receptor se coloca en puntos fijos de las habitaciones, como puede ser el bolsillo del usuario o en el mobiliario, mientras que durante las transiciones entre habitaciones el receptor se lleva en la mano. Se adquieren señales durante 5 horas, lo que significa que el número de muestras para evaluar los algoritmos es de 76943. La exactitud de los modelos se calcula a partir de las matrices de confusión utilizando o no

el algoritmo de fusión. La tabla 14 muestra la exactitud de cada modelo (ML + IMU significa que se ha utilizado el algoritmo de fusión).

Algoritmo	Exactitud (%)	
	DT	ML
	ML+IMU	94,0
SVM	ML	96,1
	ML+IMU	96,3
KNN	ML	96,0
	ML+IMU	96,3

Tabla 14. Exactitud de los clasificadores en las pruebas de validación del sistema.

El modelo KNN (ML+IMU) proporciona los mejores resultados, por lo que se estudia la sensibilidad, precisión, especificidad y exactitud del sistema de localización a nivel de habitación. La tabla 15 muestra las métricas de localización para las 6 habitaciones y la figura 46 representa la matriz de confusión para cada habitación, utilizando o no el algoritmo de fusión sensorial.

R	Sensibilidad	Precisión	Especificidad	Exactitud
1	97,5	99,3	99,7	99,1
2	98,0	91,1	97,9	97,9
3	92,8	92,5	98,4	97,4
4	76,5	93,4	98,8	95,0
5	93,8	94,5	98,8	97,9
6	90,4	71,1	96,5	95,9

Tabla 15. Métricas de la localización a nivel de habitación.

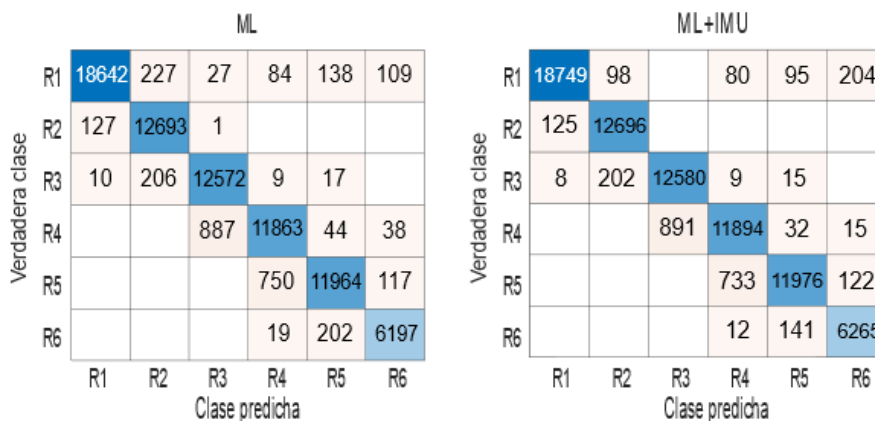


Figura 46. Matrices de confusión con KNN. Izquierda: sin considerar la aceleración; Derecha: considerando la aceleración.

Como se muestra en la tabla 15, todas las métricas son buenas, lo que justifica su uso para la evaluación de rutinas de comportamiento. Además, se vuelve a corroborar que utilizando el algoritmo de fusión los resultados obtenidos son mejores.

5.2.3 Pruebas de evaluación de rutinas de comportamiento.

Una vez validado el sistema para su uso en la estimación de rutinas de comportamiento, se van a realizar las pruebas de evaluación de rutinas de comportamiento.

Las pruebas de evaluación de rutinas se realizan con un voluntario en el mismo entorno experimental de las pruebas de validación durante 35 días. El voluntario lleva el receptor durante dos horas al día y diariamente visita alguna de las habitaciones balizadas. La figura 47 muestra los resultados de localización durante 10 días en el entorno de prueba empleando el algoritmo de fusión. El tiempo de localización en cada una de las habitaciones se representa con un color diferente, y las transiciones entre habitaciones se representan en negro.

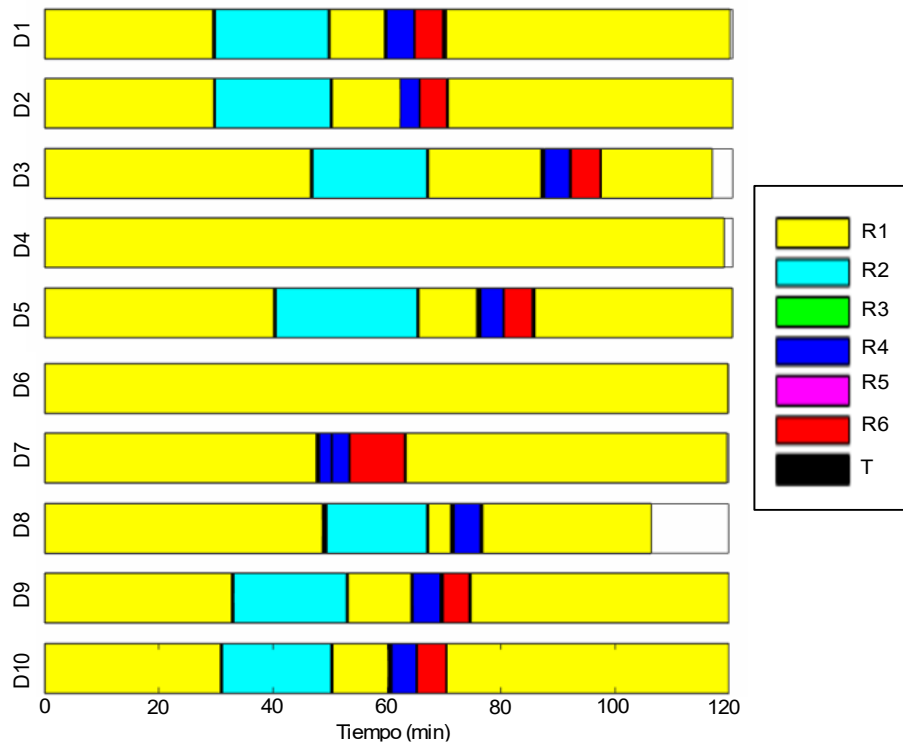


Figura 47. Resultados del algoritmo de fusión durante 10 días para tecnología BLE.

Para calcular el vector de referencia, t_{mREF} , se consideran 17 días, y cada uno de los conjuntos de días está compuesto por 8 días, lo que significa que se forman 24310 conjuntos de 8 días ($J=17$, $D=8$, $W=24310$). La tabla 16 muestra los valores del vector de referencia para cada una de las habitaciones.

Vector de referencia	
	Valor de referencia (min) ± STD
R 1	86,0 ± 2,6
R 2	19,3 ± 0,9
R 3	0,0 ± 0,0
R 4	4,7 ± 0,2
R 5	0,0 ± 0,0
R 6	4,8 ± 0,3
Transición	2,2 ± 0,1
$t_{mREF} = [86,0 \ 19,3 \ 0,0 \ 4,7 \ 0,0 \ 4,8]$	

Tabla 16. Vector de referencia.

Al igual que en el caso del vector de referencia, para el cálculo del orden de referencia, \mathbf{o}_{REF} , se consideran los mismos 17 días. El orden de referencia obtenido se muestra en (31).

$$\mathbf{o}_{REF} = [\mathbf{R1}, \mathbf{R2}, \mathbf{R1}, \mathbf{R4}, \mathbf{R6}, \mathbf{R1}] \tag{31}$$

La evaluación de la actividad diaria se va a realizar mediante el uso del coeficiente de reproducibilidad diario (α_j) y el coeficiente de similitud diario (β_j). En la figura 48 se muestra el coeficiente de reproducibilidad para los 35 días monitorizados.

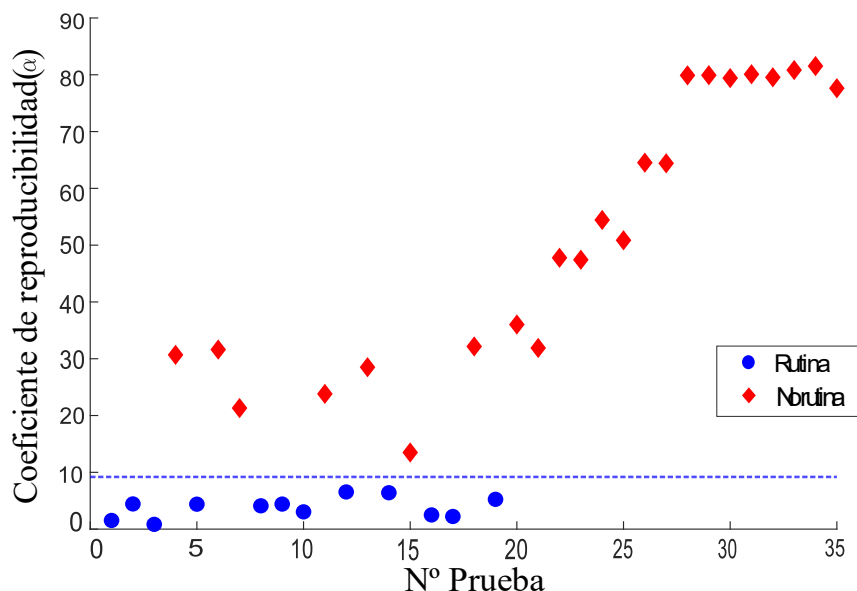


Figura 48. Coeficiente de reproducibilidad para cada uno de los días.

En la figura es posible diferenciar los días en los que se sigue la rutina de referencia (círculos azules) y los días en los que no se ha seguido (diamantes

rojos). El criterio para considerar un día rutinario es que el coeficiente se encuentre por debajo de 9, que se corresponde con la desviación del 15 % respecto al tiempo total en cada habitación.

La figura muestra como en los primeros 17 días el coeficiente de reproducibilidad tiene un valor pequeño y en la mayoría de los casos se encuentra por debajo de umbral de 9, lo que muestra la existencia de una rutina. Sin embargo, en ese período hay días que se encuentran fuera de la rutina establecida (diamantes rojos). Esta tendencia aumenta con el paso del tiempo observándose que a partir del día 25 el coeficiente vuelve a estabilizarse, pero con valores mucho más altos, mostrando un cambio completo de rutina.

En la figura 49 se muestra el coeficiente de similitud para los 35 días monitorizados. En la figura aparecen diferenciados los días en los que se cumple la rutina de tiempo (azul) y los que no (rojo).

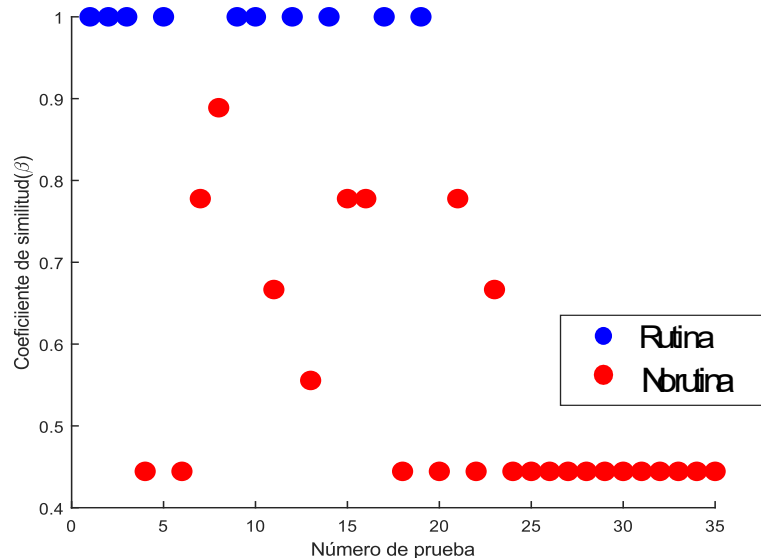


Figura 49. Coeficiente de similitud para cada uno de los días.

Se observa que los días en los que se sigue el vector de referencia también se sigue el orden de referencia ya que el coeficiente de similitud es igual a 1. En los demás casos el coeficiente se encuentra por debajo de 1. Al igual que en el caso del coeficiente de reproducibilidad se observa que el coeficiente a partir del día 25 se estabiliza, lo que confirma el cambio de rutina tanto de tiempo por cada una de las habitaciones como de orden que se ha seguido. En este punto se debe generar un nuevo vector y orden de referencia nuevos. El nuevo vector de referencia se muestra en la tabla 17 y el nuevo orden de referencia se muestra en (32).

Vector de referencia	
	Valor de referencia (min) ± STD
R 1	29,6 ± 0,9
R 2	0,0 ± 0,0
R 3	79,1 ± 1,4
R 4	0,0 ± 0,0
R 5	10,08 ± 0,9
R 6	0,0 ± 0,0
Transición	0,94 ± 0,1
$t2_{mREF} = [29,6 \ 0,0 \ 79,1 \ 0,0 \ 10,08 \ 0,0]$	

Tabla 17. Nuevo vector de referencia.

$$o2_{REF} = [R3, R5, R1, R3] \tag{32}$$

Los nuevos vector y orden de referencia se han creado manualmente, en trabajos futuros se pretende automatizar el sistema para que sea capaz de detectar nuevas rutinas de comportamiento y por tanto nuevos vectores y ordenes de referencia. A partir del nuevo vector y orden de referencia se pueden calcular los nuevos coeficientes de reproducibilidad y similitud, que se muestran en las figuras 50 y 51.

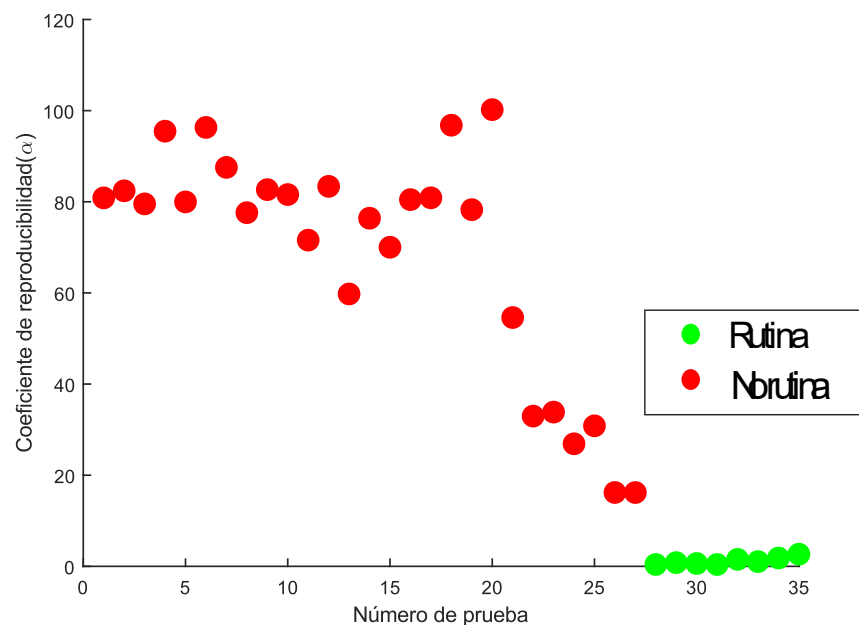


Figura 50. Coeficiente de reproducibilidad para el nuevo vector de referencia.

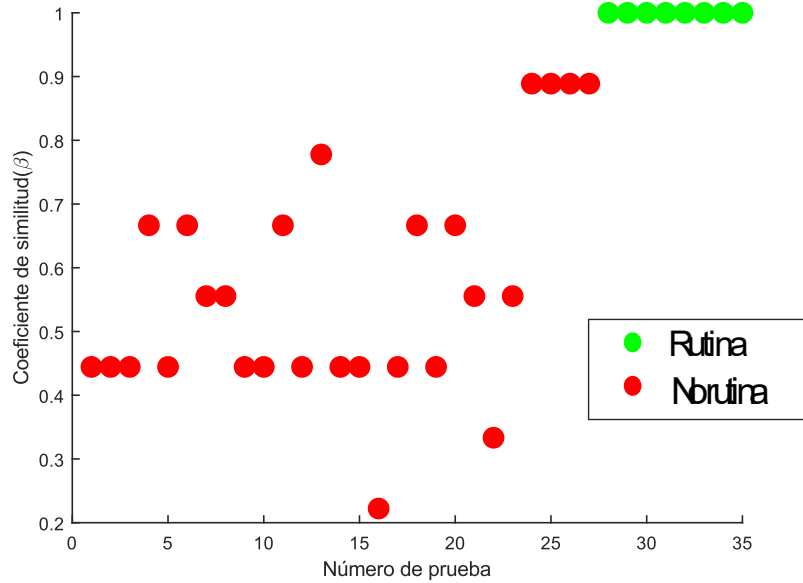


Figura 51. Coeficiente de similitud para el nuevo orden de referencia.

En la primera figura se muestra que cuando se cumple la segunda rutina establecida el coeficiente de reproducibilidad se encuentra en torno a 0, y en la segunda figura se muestra que cuando se sigue el orden de referencia de la segunda rutina el coeficiente de similitud vale 1.

A la vista de los resultados obtenidos la creación del vector y el orden de referencia y el posterior cálculo de los coeficientes de reproducibilidad y similitud permiten determinar cuando se siguen una o varias rutinas diferentes. La adición del orden de referencia y el coeficiente de similitud con respecto al sistema WiFi permiten determinar si además de cumplirse la rutina en cuanto al tiempo medio en cada una de las habitaciones monitorizadas, se cumple la rutina también en cuanto al orden de visita de las habitaciones.

6. Conclusiones y trabajos futuros

El desarrollo de este Trabajo de Fin de Máster ha permitido conocer los sistemas de localización basados en tecnologías de radiofrecuencia además del uso de la fusión sensorial con el objetivo de obtener mejores resultados. Las diferentes pruebas realizadas permiten estudiar la viabilidad de los sistemas WiFi y *Bluetooth* para la evaluación de rutinas de comportamiento.

Las pruebas del sistema WiFi se realizan en dos entornos diferentes. En el primer entorno, la Escuela Politécnica Superior de la Universidad de Alcalá, se realizan tres tipos de pruebas diferentes. En el primer tipo de pruebas, las pruebas de umbral, se determina la potencia de emisión en la que se configure el *router* no influye en la distancia en la que el porcentaje de conexión baja del 100 %, pero sí influye en la caída de la señal, la cual es más o menos abrupta en función del modo de conexión en el que se configura el *router*. En estas pruebas también se determina que un umbral característico para determinar que un usuario se encuentra en una determinada estancia se encuentra en torno al 75-80% de conectividad.

Las pruebas de interferencia permiten determinar si los *routers* que se encuentran en plantas diferentes pueden verse interferidos y provocar fallos de estimación de habitación en el sistema propuesto. Del mismo modo, si varios *router* se encuentran dentro de una sala diáfana, las señales pueden verse interferidas si no se encuentran a una distancia suficiente, provocando fallos en el sistema propuesto.

Por último, se realizan las pruebas para la evaluación de rutinas a partir del sistema propuesto. En estas pruebas se monitoriza a dos voluntarios durante su actividad diaria en el entorno experimental escogido. El uso del vector de referencia y del coeficiente de reproducibilidad permiten determinar qué días se encuentran dentro de la rutina y cuáles no.

En el segundo entorno, el centro sociosanitario Las Palmeras del grupo Albertia en Azuqueca, se realizan dos series de pruebas diferentes. Las pruebas A las realiza un voluntario robusto para evaluar la precisión del sistema dentro del nuevo entorno. La precisión obtenida es del 95,6 % lo que justifica su uso para un paciente de la propia residencia (voluntario frágil). Las pruebas B se realizan durante 12 días. A partir de los resultados del sistema se puede visualizar una posible rutina, pero no se puede establecer una rutina detallada ya que hay habitaciones a las que acude el paciente durante su vida diaria que no están monitorizadas.

Las pruebas del sistema *Bluetooth* también se realizan en dos entornos diferentes, el domicilio del usuario monitorizado y la EPS. En el primer entorno se demuestra la incidencia que tienen el nivel de potencia de emisión de las balizas, el tiempo que necesita el sistema para detectar que un usuario se encuentra en una determinada habitación y la posición del receptor sobre la

exactitud del sistema. De los tres elementos el que mayor incidencia tiene sobre la exactitud del sistema es la posición del receptor, teniendo que crear un nuevo modelo para mejorar la exactitud del sistema propuesto. En menor medida, la potencia también afecta a la exactitud del sistema obteniendo los mejores resultados con la potencia de emisión de -8 dBm. Por último, se comprueba que el sistema es capaz de detectar que el usuario se encuentra en una habitación estando 10 segundos en ella. Por otro lado, se realizan pruebas de larga duración y en ellas se comprueba cómo afecta la variabilidad de la señal *Bluetooth* a la localización. En ellas se justifica el uso de la aceleración para mejorar los resultados.

En el segundo entorno se valida el sistema propuesto en el primer entorno. La validación se realiza utilizando tres modelos de ML diferentes, KNN, SVM y DT, obteniendo los mejores resultados con el algoritmo KNN. Posteriormente se utilizan los resultados del algoritmo de fusión para evaluar la rutina de comportamiento de un usuario monitorizado durante 35 días. Al igual que en el caso del sistema WiFi, el uso del vector de referencia y del coeficiente de reproducibilidad permiten evaluar las rutinas de comportamiento, además en este caso se añade el orden de referencia que permite evaluar las rutinas añadiendo el orden en que se han visitado las diferentes habitaciones. Además, a partir de los coeficientes de reproducibilidad y de similitud se pueden establecer nuevas rutinas de comportamiento diferentes a la inicial si estos valores tienden a estabilizarse en el tiempo.

En ambos sistemas se consigue el objetivo de la evaluación de rutinas de comportamiento, pero sus principales diferencias se encuentran en los sistemas de localización. Respecto a la localización, el sistema WiFi se trata de un sistema más limitado que el sistema *Bluetooth*, esto se debe a que todos los routers WiFi deben estar configurados con la misma SSID por limitaciones del sistema receptor. Esto provoca que tengan que estar lo suficiente alejados y que sea imposible diferenciar dos routers si estos están colocados en la misma planta. En este aspecto el sistema de localización *Bluetooth* permite monitorizar un mayor número de habitaciones estando estas en la misma o en diferentes plantas.

Respecto a la fusión sensorial, la medición de la actividad física del usuario monitorizado, que es independiente de la tecnología de localización, es más precisa en el caso del sistema WiFi debido al uso de IMUs específicas (NGIMU) ubicadas en posiciones óptimas, como el tobillo.

De cada uno de los sistemas se ha presentado un trabajo a un congreso con su correspondiente publicación en los libros de *abstracts*. En el año 2020 se publicó el artículo “*Symbolic localization of institutionalized patients for detection of daily living activities*” [38] en el congreso internacional MeMeA (*International Symposium on Medical Measurements and Applications*) y en el año 2021 se publicó el artículo “*BLE-bases approach for detecting daily routines changes*” [39] en el mismo congreso.

Por tanto, ambos sistemas sirven como una primera aproximación para la evaluación de rutinas de comportamiento. En trabajos futuros se pretende incluir procedimientos estandarizados para la evaluación de estas, como puede ser el uso del análisis de componentes principales (PCA) y el uso del algoritmo de propagación por afinidad que pueden permitir detectar varias rutinas de manera automática. Además, se pretende incluir nuevas características de análisis de las rutinas además del tiempo por habitación y el orden en el que se visitan las habitaciones.

7. Planos

En este apartado se van a incluir alguna de las partes más importantes del código desarrollado sobre Matlab.

7.1 Sistema WiFi

En este apartado se van a incluir las partes de código que se han desarrollado para el sistema WiFi, tanto de la parte de localización como de la generación de rutinas.

7.1.1 Localización

- **Detección de pasos:** en este apartado se van a incluir las diferentes partes del algoritmo de detección de pasos, entre las que se incluye el filtro de Kalman y la técnica de cruce por cero.

- **EKF:**

```
function
[Rolle,Pitche,Yawe]=EKF(Time,AccX,AccY,AccZ,GyrX,GyrY,GyrZ,X,P,Q,R,H)

for i=2:1:length(AccX)
    dt=(Time(i)-Time(i-1));

    % Etapa de predicción: información relativa con el giroscopio.
    sinRoll=sin(X(1)); cosRoll=cos(X(1)); tanRoll=tan(X(1));
    secRoll=sec(X(1));
    sinPitch=sin(X(2)); cosPitch=cos(X(2)); tanPitch=tan(X(2));
    secPitch=sec(X(2));
    sinYaw=sin(X(3)); cosYaw=cos(X(3)); tanYaw=tan(X(3));
    secYaw=sec(X(3));

    X_(1,1)=X(1) + dt*( GyrX(i) + GyrY(i)*sinRoll*tanPitch +
    GyrZ(i)*cosRoll*tanPitch);
    X_(2,1)=X(2) + dt*( GyrY(i)*cosRoll -
    GyrZ(i)*sinRoll );
    X_(3,1)=X(3) + dt*( GyrY(i)*sinRoll/cosPitch +
    GyrZ(i)*cosRoll/cosPitch );

    A=[];

    A(1,1) = GyrY(i)*cosRoll*tanPitch - GyrZ(i)*sinRoll*tanPitch;
    A(1,2) = GyrY(i)*sinRoll*secPitch^2 + GyrZ(i)*cosRoll*secPitch^2;
    A(1,3) = 0;

    A(2,1) = -GyrY(i)*sinRoll - GyrZ(i)*cosRoll;
    A(2,2) = 0;
    A(2,3) = 0;

    A(3,1) = GyrY(i)*cosRoll*secPitch -
    GyrZ(i)*sinRoll*secPitch;
    A(3,2) = GyrY(i)*sinRoll*secPitch*tanPitch +
    GyrZ(i)*cosRoll*secPitch*tanPitch;
    A(3,3) = 0;
```

```

A = eye(3) + A*dt;

for j=1:3
if X_(j)>=pi
    X_(j)=X_(j)-2*pi;
end

if X_(j)<=-pi
    X_(j)=X_(j)+2*pi;
end
end

% Etapa de actualización Roll y Pitch
Acc=sqrt(AccX(i)^2+AccY(i)^2+AccZ(i)^2);

if ((abs(Acc-1)<0.01) && (1==1))
    % Vector de observaciones: roll y pitch con los acelerómetros
    Z(1,1)=atan(AccY(i)/AccZ(i));
    Z(2,1)=atan(-AccX(i)/sqrt(AccY(i)^2+AccZ(i)^2));
    % Vector de estimaciones de las observaciones: Estimaciones a
priori
    h_X_(1,1)=X_(1);
    h_X_(2,1)=X_(2);

    %Filtro EKF
    [X P]=Filtro_Kalman_Extendido(X_,P,A,H,Z,h_X_,Q,R); %incluye
las ecuaciones de la etapa de actualización del filtro
    %Si el sujeto está en movimiento actualizo con mi estado a priori
else
    X=X_;
end

Rolle(i,1)=X(1)*180/pi;
Pitche(i,1)=X(2)*180/pi;
Yawe(i,1)=X(3)*180/pi;

end
end

```

```

function [X,P] = Filtro_Kalman_Extendido(X_,P,A,H,Z,h_X_,Q,R)
%FILTRO_KALMAN_EXTENDIDO Summary of this function goes here
% Detailed explanation goes here

% X_: estado a priori
% P: Matriz de covarianza del error
% A: Jacobiano ecuación transición
% H: Jacobiano ecuación de medida
% Z: Observaciones
% h_X_: Estimación de las observaciones a partir del estado a priori
% Q: Matriz covarianza error ecuacion de transición
% R: Matriz covarianza error ecuación de medida

% Devuelve el estado actualizado X y la matriz de covarianza del error
P
%Ecuaciones de la etapa de actualización del filtro
P_=A*P*A' + Q;
S=H*P_*H' + R;
K=P_*H'*(S^(-1));
V=Z-h_X_;
X=X_+K*V;
P=P_- K*S*K';

```

end

o ZCT:

```
function [ARRAY,Pitche_procesado,amplitud]=Deteccion_pasos (Pitche,num)

up=0;
down=0;
semaforo_inicial=0;
zero=0;
maximo_procesado=0;
minimo_procesado=0;
contador=0;

for i=2:1:num
    if i>=101
        promediado(i,1)=mean(Pitche(i-100:i));
        Pitche_procesado(i,1)=Pitche(i,1)-promediado(i,1);

        if Pitche_procesado(i-1,1)<=0 && Pitche_procesado(i,1)>0
%cruce por cero, direccion ascendente
            if semaforo_inicial>=1
                zero=zero+1;
            else
                semaforo_inicial=1;
            end
            up=1;
            down=0;
        end

        if Pitche_procesado(i-1,1)>=0 && Pitche_procesado(i,1)<0
%cruce por cero, direccion descendente
            if semaforo_inicial>=1
                zero=zero+1;
            else
                semaforo_inicial=1;
            end
            down=1;
            up=0;
        end

        if up==1 && Pitche_procesado(i,1)>maximo_procesado
            maximo_procesado= Pitche_procesado(i,1);
            posmax_procesado=i;
        end

        if down==1 && Pitche_procesado(i,1)<minimo_procesado
            minimo_procesado= Pitche_procesado(i,1);
            posmin_procesado=i;
        end
        amplitud=maximo_procesado-minimo_procesado;

        if zero==2 %Dos cruce por cero
            if abs(amplitud)>50 && maximo_procesado>25 &&
minimo_procesado<-25
                if up==1
                    contador=contador+1;
                    ARRAY(contador,1)=maximo_procesado;
                    ARRAY(contador,2)=posmax_procesado;
                    ARRAY(contador,3)=1;
                    contador=contador+1;
                    ARRAY(contador,1)=minimo_procesado;
```



```

tipo_recorrido=12; %valor de la desviación estándar para diferenciar
un cambio de planta por escaleras o ascensor

N_muestras = length(BarometerhPa);
secs = 0;
k = 0; %Índice iterativo para hallar la
pendiente cada 250 muestras
pos = 1; %Posición del array de la pendiente

pos_hpa = 1;
delta_index = 1;

start_slope = 0;
end_slope = 0;

for i = 1:N_muestras

    secs = secs + 0.01;
    seconds(i) = secs;
    minutes(i) = secs/60;
    slope=2500; %Cálculo de la presión cada 25 s
    k = k + 1;
    if k > slope

        k = 0;
        m(pos) = (BarometerhPa(i) - BarometerhPa(i -
slope))/(minutes(i) - minutes(i - slope));

        if pos > 2
            if abs(m(pos)) > diff_m && abs(m(pos - 1)) < diff_m
%condiciones para identificar un cambio de presión significativo
                pressure(pos_hpa) = BarometerhPa(i - slope);
                posp(pos_hpa) = i-slope;
                start_slope = 1;
                pos_hpa = pos_hpa + 1;
            else if abs(m(pos)) < diff_m && abs(m(pos - 1)) > diff_m
&& start_slope
                pressure(pos_hpa) = BarometerhPa(i);
                posp(pos_hpa)= i;
                end_slope = 1;
                pos_hpa = pos_hpa + 1;
            end
        end
    end

    if start_slope && end_slope
        start_slope = 0;
        end_slope = 0;

        delta_pres(delta_index) = pressure(pos_hpa - 1) -
pressure(pos_hpa - 2);
        %detección del tipo de cambio de planta
        tipocambiodeplanta

        if abs(delta_pres(delta_index)) > delta_three
            if delta_pres(delta_index) > 0
                floor_number = floor(i - 1) - 3;
            else
                floor_number = floor(i - 1) + 3;
            end
        end
    end
end

```

```

        end
        else if abs(delta_pres(delta_index)) > delta_two
            if delta_pres(delta_index) > 0
                floor_number = floor(i - 1) - 2;
            else
                floor_number = floor(i - 1) + 2;
            end
        else if abs(delta_pres(delta_index)) > delta_one
            if delta_pres(delta_index) > 0
                floor_number = floor(i - 1) - 1;
            else
                floor_number = floor(i - 1) + 1;
            end
        end
        end
        end
        delta_index = delta_index + 1;
    end

    pos = pos + 1;

end

floor(i) = floor_number;

end

```

- **Detección de tipo de cambio de planta:**

```

%detección del tipo de cambio de planta
inicial=0;
final=0;
cogerinicial=1;
cogerfinal=1;
for z=1:length(ARRAY) %Se analiza la parte de la señal pitch
correspondiente al cambio de planta
    if ARRAY(z,2)>(posp(pos_hpa-2)-1000) && cogerinicial==1
        inicial=z;
        cogerinicial=0;
    end
    if ARRAY(z,2)>(posp(pos_hpa-1)-3000) && cogerfinal==1
        final=z;
        cogerfinal=0;
    end
end
noascensor=0;
ascensor=0;
contascensor=0;
contescaleras=0;
contnormal=0;
if ARRAY(inicial,1)>0 %detección del tipo de cambio de planta primera
via
    for k=inicial+1:2:final
        if ARRAY(k,3)==1
            contnormal=contnormal+1;
        elseif ARRAY(k,3)==2 %Cambio de polaridad en la señal Pitch
            contescaleras=contescaleras+1;
        end
    end
elseif ARRAY(inicial,1)<0

```



```

    for k=inicial+2:2:final
        if ARRAY(k,3)==1
            contnormal=contnormal+1;
        elseif ARRAY(k,3)==2 %cambio de polaridad en la señal pitch
            contescaleras=contescaleras+1;
        end
    end
end
if contescaleras<5
    ascensor=1;
end
if ascensor==0
    fprintf('El desplazamiento se ha realizado por las escaleras\n')
elseif ascensor==1
    fprintf('El desplazamiento se ha realizado por el ascensor\n')
end

%detección del tipo de cambio de planta segunda via
desviacionpitch(delta_index)=std(Pitche_procesado((posp(pos_hpa-
2)+100):(posp(pos_hpa-1)-4000)));
if desviacionpitch(delta_index) > tipo_recorrido
    escaleras_ascensor(delta_index)=0;
    fprintf('El desplazamiento se ha realizado por las escaleras\n')
elseif desviacionpitch(delta_index) < tipo_recorrido
    escaleras_ascensor(delta_index)=1;
    fprintf('El desplazamiento se ha realizado por el ascensor\n')
end
end

```

- Localización simbólica + fusión sensorial:** en este apartado se incluyen las partes de código del algoritmo de localización simbólica y del algoritmo de fusión sensorial.

```

%Algoritmo para localización simbólica
z=2;
planta(1)=floor(1);
for i=1:length(floor) %se conoce en que planta se está cada segundo
    if mod(i,100)==0
        planta(z)=floor(i);
        z=z+1;
    end
end
umbral=75; %Umbral de conexión
dt=60; %valor de la ventana de tiempo
%NO LOCALIZACIÓN
nocalizacion(1:length(y))=0;
%Transición ascensor
ascensor(1:length(y))=0;
%Transición escalera
escaleras(1:length(y))=0;
%VARIABLES CAFETERÍA
umbralodos=50; %Umbral especial para la cafetería
cafeteria(1:length(y))=0;
%VARIABLES LAB22
contador_lab22=0;
lab22(1:length(y))=0;
inicial_lab22=0;
a=1;
b=0;
k=1;
final_lab22=0;

```

```

%VARIABLES LAB31
contador_lab31=0;
inicial_lab31=0;
final_lab31=0;
c=1;
d=0;
umbral_pasos=10;
lab31(1:length(y))=0;
planta(length(planta):length(y))=floor(1);
for i=1:(length(y)-1)
    if planta(i)==0 %cafetería
        if(y(i)>=umbralDOS)
            cafeteria(i)=1;
            nlocalizacion(i)=0;
        elseif(y(i)<umbralDOS)
            cafeteria(i)=0;
            nlocalizacion(i)=1;
        end
    elseif planta(i)==2 %lab 22
        if(y(i)>=umbral)
            lab22(i)=1;
            nlocalizacion(i)=0;
        elseif(y(i)<umbral)
            lab22(i)=0;
            nlocalizacion(i)=1;
            if(a==1)
                inicial_lab22=i;
                inicial(k)=i;
                a=0;
                b=1;
            end
            if(y(i+1)>=umbral && b==1)
                final_lab22=i;
                final(k)=i;
                b=0;
            end
            if (a==0) && (b==0) %fusión sensorial
                if(inicial_lab22>100)
                    for j=inicial_lab22-dt:final_lab22+dt
                        if (j<length(pasos_ventana))
                            if (pasos_ventana(j)~=0)
                                contador_lab22=contador_lab22 + pasos_ventana(j);
                            end
                        end
                    end
                    if(contador_lab22<umbral_pasos)
                        lab22(inicial_lab22:final_lab22)=1;
                        nlocalizacion(inicial_lab22:final_lab22)=0;
                    end
                    contador_lab22=0;
                end
                a=1;
                b=0;
                k=k+1;
            end
        end

    end
    if(y(i)<umbralDOS)
        lab22(i)=0;
    end
end

```

```

        nlocalizacion(i)=1;
    end
elseif planta(i)==3 %lab 31
    if (y(i)>=umbral)
        lab31(i)=1;
        nlocalizacion(i)=0;
    elseif (y(i)<umbral)
        lab31(i)=0;
        nlocalizacion(i)=1;
        if (c==1)
            inicial_lab31=i;
            inicial(k)=i;
            c=0;
            d=1;
        end
        if (y(i+1)>=umbral && d==1)
            final_lab31=i;
            final(k)=i;
            d=0;
        end
        if (c==0) && (d==0) %fusion sensorial
            if (inicial_lab31>100)
                for j=inicial_lab31-dt:final_lab31+dt
                    if (j<length(pasos_ventana))
                        if (pasos_ventana(j)~=0)
                            contador_lab31=contador_lab31 + pasos_ventana(j);
                        end
                    end
                end
                if (contador_lab31<umbral_pasos)
                    lab31(inicial_lab31:final_lab31)=1;
                    nlocalizacion(inicial_lab31:final_lab31)=0;
                end
                contador_lab31=0;
            end
            c=1;
            d=0;
            k=k+1;
        end
    end

    end
    if (y(i)<umbraldos)
        lab31(i)=0;
        nlocalizacion(i)=1;
    end
end

end
cafeteria(length(y))=0;
lab22(length(y))=0;
lab31(length(y))=0;
nlocalizacion(length(y))=0;
oeste(length(y))=0;
nlocalizacion(1)=0;
delta_index=1;
%Se añade el tipo de transición
if pos_hpa>1
    posp=round(posp/100);
    for i=1:2:(length(posp)-1)
        cafeteria(posp(i):posp(i+1))=0;
    end
end

```

```

lab22 (posp(i) : posp(i+1))=0;
lab31 (posp(i) : posp(i+1))=0;
nolocalizacion (posp(i) : posp(i+1))=0;
oeste (posp(i) : posp(i+1))=0;
if escaleras_ascensor(delta_index)==0
escaleras (posp(i) : posp(i+1))=1;
delta_index=delta_index+1;
elseif escaleras_ascensor(delta_index)==1
ascensor (posp(i) : posp(i+1))=1;
delta_index=delta_index+1;
end
end
end

```

7.1.2 Generación de rutinas

- **Vector de referencia + coeficiente de reproducibilidad:** en este apartado se incluyen las partes de código correspondientes al cálculo del vector de referencia y del coeficiente de reproducibilidad.

```

%VECTOR DE HORAS OBTENIDOS A PARTIR DE LA LOCALIZACIÓN
v_horas(1,:)=horas_semana1;
v_horas(2,:)=horas_semana2;
v_horas(3,:)=horas_semana3;
v_horas(4,:)=horas_semana4;
v_horas(5,:)=horas_semana5;
v_horas(6,:)=horas_semana6;
v_horas(7,:)=horas_semana7;
v_horas(8,:)=horas_semana8;
v_horas(9,:)=horas_semana9;
v_horas(10,:)=horas_semana10;
v_horas(11,:)=horas_semana11;
v_horas(12,:)=horas_semana12;
v_horas(13,:)=horas_semana13;
v_horas(14,:)=horas_semana14;
v_horas(15,:)=horas_semana15;
v_horas(16,:)=horas_semana16;
v_horas(17,:)=horas_semana17;
v_horas(18,:)=horas_semana18;
v_horas(19,:)=horas_semana19;
v_horas(20,:)=horas_semana20;
v_min=v_horas*60; %vector de minutos

%CALCULO DEL VECTOR DE REFERENCIA
tamano=1:1:length(v_min);
grupos=8;%combinaciones de 8 elementos
combinaciones=nchoosek(tamano,grupos);
estados=6;%numero de habitaciones monitorizadas
for i=1:estados
desv=0;
v_desv=[];
    for j=1:length(combinaciones)
desv=std(v_min(combinaciones(j,:),i));
v_desv(j)=desv;
    end
[minimo,pos]=min(v_desv);
v_referencia(i)=mean(v_min(combinaciones(pos,:),i));
v_desvmin(i,:)=v_min(combinaciones(pos,:),i);

```

```
v_std(i)=std(v_min(combinaciones(pos,:),i));
end
cafeteria_ref=v_referencia(1)
lab22_ref=v_referencia(2)
lab31_ref=v_referencia(3)
ascensor_ref=v_referencia(4)
escaleras_ref=v_referencia(5)
nolocalizacion_ref=v_referencia(6)

%Coeficiente de reproducibilidad

for i=1:length(v_min)
repro(i)=1.96*std(v_min(i,:)-v_referencia);
end

t_a=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];
sz=140;
figure();
uno=scatter(t_a,repro,sz,'filled','b')
hold on
yline(10,'--b','linewidth',1);
xlabel('N° DIA')
ylabel('Coeficiente de Reproducibilidad')
```

7.2 Sistema Bluetooth

En este apartado se van a incluir las partes de código que se han desarrollado para el sistema *Bluetooth*, tanto de la parte de localización como de la generación de rutinas.

7.2.1 Localización

En este apartado se van a incluir los algoritmos que se han desarrollado para la localización mediante tecnología *Bluetooth*.

- **Clasificador basado en *Machine Learning*:**

```
%Colocación de los datos para el uso del clasificador de ML
baliza1=test_potencia(:,1);
baliza2=test_potencia(:,2);
baliza3=test_potencia(:,3);
baliza4=test_potencia(:,4);
baliza5=test_potencia(:,5);
baliza6=test_potencia(:,6);
test=table(baliza1,baliza2,baliza3,baliza4,baliza5,baliza6);
load('modelouni_completo.mat') %Se carga el modelo de ML (KNN,SVM o
DT)
prediccion=modelouni_completo.predictFcn(test); %Predicción del modelo
de ML
val=table(baliza1,baliza2,baliza3,baliza4,baliza5,baliza6,prediccion);
for i=1:length(prediccion)
a=string(prediccion(i,1));
prediccion_str(i)=a;
end
variable=0;
```

```

for i=1:length(prediccion_str)
if prediccion_str(i)=="Laboratorio 22"
lab22(i)=1;
d1(i)=0;
d2(i)=0;
b1(i)=0;
b2(i)=0;
hall(i)=0;
elseif prediccion_str(i)=="Despacho 1"
lab22(i)=0;
d1(i)=1;
d2(i)=0;
b1(i)=0;
b2(i)=0;
hall(i)=0;
elseif prediccion_str(i)=="Despacho 2"
lab22(i)=0;
d1(i)=0;
d2(i)=1;
b1(i)=0;
b2(i)=0;
hall(i)=0;
elseif prediccion_str(i)=="Baño 1"
lab22(i)=0;
d1(i)=0;
d2(i)=0;
b1(i)=1;
b2(i)=0;
hall(i)=0;
elseif prediccion_str(i)=="Baño 2"
lab22(i)=0;
d1(i)=0;
d2(i)=0;
b1(i)=0;
b2(i)=1;
hall(i)=0;
elseif prediccion_str(i)=="Hall"
lab22(i)=0;
d1(i)=0;
d2(i)=0;
b1(i)=0;
b2(i)=0;
hall(i)=1;
end
end
correccionbluetoothuni

```

- **Algoritmo de fusión sensorial:**

```

%Corrección mediante IMU
%datosacelerómetro
for k=1:length(pruebas_casa_ismael(:,1))
tipodato=pruebas_casa_ismael(k,1);
if (tipodato=="ACCE")
contador2=contador2+1;
datax=char(pruebas_casa_ismael(k,4));
datay=char(pruebas_casa_ismael(k,5));
dataz=char(pruebas_casa_ismael(k,6));
datatime=char(pruebas_casa_ismael(k,2));
ax(contador2)=str2double(datax);
ay(contador2)=str2double(datay);

```

```

az(contador2)=str2double(dataz);
ta(contador2)=str2double(datatime);
end
end
vector_corr=[];
for i=1:length(lab22)
if lab22(i)==1
vector_corr(i)=1;
elseif d1(i)==1
vector_corr(i)=2;
elseif d2(i)==1
vector_corr(i)=3;
elseif b1(i)==1
vector_corr(i)=4;
elseif b2(i)==1
vector_corr(i)=5;
elseif hall(i)==1
vector_corr(i)=6;
end
end
%CORRECCION
umbral=4; %UMBRAL MODIFICABLE
tiempo=1;
correccion=1;
cont=0;
for x=100:length(vector_corr)
if vector_corr(x)~=vector_corr(x-1)
tiempo_corr=v_tiempo(x);
for y=1:length(ta)
if ta(y)>tiempo_corr-tiempo && ta(y)<tiempo_corr+tiempo
if ax(y)>umbral %No corrección
correccion=0;
end
end

end
if correccion==1
vector_corr(x)=vector_corr(x-1)
elseif correccion==0
correccion=1;
end
end
end
end

```

7.2.2 Generación de rutinas

En este apartado se van a incluir los algoritmos que se han desarrollado para la generación de rutinas a partir de la localización *Bluetooth*. En este caso no se va a incluir el código del cálculo del vector de referencia ni del coeficiente de reproducibilidad ya que se utiliza el mismo código que en el caso del sistema WiFi.

- **Orden de referencia:**

```

%Secuencias de los días analizados
v_secuencia(1,:)=secuencia1;
v_secuencia(2,:)=secuencia2;

```

```

v_secuencia(3,:)=secuencia3;
v_secuencia(4,:)=secuencia4;
v_secuencia(5,:)=secuencia5;
v_secuencia(6,:)=secuencia6;
v_secuencia(7,:)=secuencia7;
v_secuencia(8,:)=secuencia8;
v_secuencia(9,:)=secuencia9;
v_secuencia(10,:)=secuencia10;
v_secuencia(11,:)=secuencia11;
v_secuencia(12,:)=secuencia12;
v_secuencia(13,:)=secuencia13;
v_secuencia(14,:)=secuencia14;
v_secuencia(15,:)=secuencia15;
v_secuencia(16,:)=secuencia16;
v_secuencia(17,:)=secuencia17;

for i=1:length(v_secuencia)
contador_secuencia=0;
    for j=1:length(v_secuencia)
        if isequal(v_secuencia(i,:),v_secuencia(j,:))
            contador_secuencia=contador_secuencia+1;
        end
    end
v_contadorsecuencia(i)=contador_secuencia;
end
val=max(v_contadorsecuencia);
t=1;
primervalor=0;
for i=1:length(v_secuencia)
if v_contadorsecuencia(i)==val
    if primervalor==0
        secuencia_ref(t,:)=v_secuencia(i,:);
        t=t+1;
        primervalor=1;
    elseif primervalor==1
        if t==2 && not(isequal(secuencia_ref(t-1,:),v_secuencia(i,:)))
            secuencia_ref(t,:)=v_secuencia(i,:);
            t=t+1;
        elseif t>2
            rep=0;
            for j=1:(t-1)
                if isequal(secuencia_ref(j,:),v_secuencia(i,:))
                    rep=rep+1;
                end
            end
            if rep==0
                secuencia_ref(t,:)=v_secuencia(i,:);
                end
            end
        end
    end
end
end
end
if t==2
fprintf('La secuencia más repetida es : \n\n');
fprintf('%s ', secuencia_ref);
fprintf('\n');
elseif t>2
fprintf('Las secuencias más repetidas son : \n\n')
for i=1:(t-1)
    %pintarsecuencia=secuencia_ref(i,:);
    fprintf('%s ', secuencia_ref(i,:));

```



```
fprintf('\n\n');  
end
```

- **Coefficiente de similitud:**

```
load('secuencia_referencia')  
secuencia_ref=quitaespacios(secuencia_ref);  
l1=length(secuencia_ref);  
for i=1:length(v_secuencia)  
a=quitaespacios(v_secuencia(i,:));  
l2=length(v_secuencia(i,:));  
Levensthein(i)=strdist(secuencia_ref,a);  
l=max(l1,l2);  
similitud(i)=(1-(Levensthein(i)/l));%calculo de la similitud que sirve  
para identificar la actividad que se está realizando  
end  
figure();  
scatter(dia,similitud,sz,'filled','r')  
xlabel('Número de prueba')  
ylabel('Coeficiente de similitud(\beta)')
```

8. Pliego de condiciones

8.1 Pliego de condiciones generales

El presente pliego de condiciones técnicas resume las características que se deben cumplir en el desarrollo del proyecto. El objetivo principal de este documento es utilizar la fusión de información obtenida a partir de dos sistemas de localización para la evaluación de rutinas de comportamiento, para lo que ha sido necesario:

- Conocimiento y comprensión de la teoría de los sistemas de localización basados en radiofrecuencia y la teoría que hay detrás de la fusión sensorial.
- Análisis del material *Hardware* necesario para el desarrollo de ambos sistemas.
- Análisis del funcionamiento de los sistemas de localización desarrollados (*Hardware + Software*) para la evaluación de rutinas.

El proyecto está formado por la siguiente documentación:

- Memoria.
- Planos.
- Pliego de condiciones.
- Presupuesto.
- Bibliografía.

Se entiende por documentación aquella que es de obligado cumplimiento.

8.2 Pliego de condiciones particulares

En el pliego de condiciones particulares se recogen los requerimientos *Software* y *Hardware* comerciales necesarios para la realización del trabajo.

- **Requerimientos Software:**
 - Matlab 2018-b: se trata de un *software* matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Mediante Matlab se ha desarrollado la algoritmia necesaria para el desarrollo del proyecto.
 - NGIMU GUI v1.15: software propietario de *IO Technologies*, se utiliza como interfaz para poder acceder a los datos proporcionados por la NGIMU y para la configuración de los parámetros de funcionamiento de esta.
 - Microsoft Office 365 v2018: *software* necesario para la elaboración de la memoria.
 - Sistema operativo: en todos los ordenadores en los que se ha desarrollado el proyecto el sistema operativo utilizado es el

Microsoft Windows 10, el cual fue desarrollado por *Microsoft* en 2014.

- **Requerimientos Hardware:**

- NGIMU: se trata de una IMU desarrollada por IO Technologies que combina sensores internos y algoritmos de procesamiento de datos con un amplio rango de interfaces de comunicación para crear una plataforma sensorial versátil enfocada tanto a las aplicaciones en tiempo real como al registro de datos.
- BQ Aquaris V Plus: se trata de un teléfono móvil desarrollado por BQ el cual se trata de un dispositivo que combina diferentes sensores con interfaces de comunicación *WiFi* y *Bluetooth* que permiten desarrollar aplicaciones tanto en tiempo real como para registro de datos.
- Router TP LINK TL-WR802N: el router TL-WR802N desarrollado por TP-LINK permite a un dispositivo conectarse a internet sin la necesidad de utilizar ningún cable.
- Balizas *Bluetooth* IBKS 105 y IBKS Plus: las balizas IBKS 105 y IBKS Plus son balizas desarrollados por *Accent System*. Las balizas son dispositivos que alertan a los dispositivos móviles de su presencia a través de *Bluetooth Low Energy* (BLE).
- PC portátil HP-HT89FRIM i3-7020U 2.3 GHz: dispositivo necesario desarrollado por HP para el uso de todos los requerimientos *software* del apartado anterior.

- **Otros requerimientos:**

A lo largo del desarrollo del TFM se han utilizado otras herramientas de trabajo como pueden ser las cintas para colocar la IMU en el tobillo o los medidores láser de distancia. Además, se han utilizado todos los documentos de referencia que aparecen incluidos en la bibliografía del documento.

9. Presupuesto

El presupuesto que se incluye en el documento incluye los costes debidos a las horas de trabajo como ingeniero (costes directos), los costes del material utilizado y los costes indirectos asociados al proyecto.

- **Costes directos del proyecto:**

El coste destinado al tiempo invertido en el proyecto, tanto por investigación, desarrollo de algoritmia, desarrollo de pruebas y el desarrollo de la documentación se recogen en la tabla 18. Se asume un coste estándar como personal de investigación de 10 euros por cada hora de trabajo.

Actividad	Tiempo empleado (horas)	Coste (€)
Investigación	170	1700
Desarrollo Algoritmia	200	2000
Desarrollo pruebas	400	4000
Desarrollo documentación	190	1900
	Tiempo total	Coste total
	960	9600

Tabla 18. Costes directos del proyecto.

- **Costes del material utilizado:**

En la tabla 19 se adjuntan los costes asociados al Hardware necesario para el desarrollo del proyecto.

Material	Precio Unitario (€)	Unidades	Precio total (€)	Periodo de Amortización (años)	Uso (meses)	Coste Amortización (€)
NGIMU	345.82	1	345.82	2	6	86.45
BQ Aquaris	249	1	249	2	6	62,25
TL-WR802N	21.63	10	216.3	2	6	54
IBKS 105	13	10	130	2	6	32,5
IBKS PLUS	22.50	10	225	2	6	56,25
HP-HT89FRIM	400	1	400	2	6	100
		TOTAL	1341.12 €		TOTAL	391,45

Tabla 19. Costes del material utilizado en el proyecto.

- **Costes indirectos del proyecto:**

En la tabla 20 se recogen los gastos indirectos del proyecto, que incluyen los gastos que suponen las licencias del *Software* empleado.

Licencia	Coste (€)	Periodo de amortización (años)	Uso (meses)	Coste amortización (€)
Matlab 2018b	2000	2	6	500
NGIMU GUI	0	2	6	0
Microsoft Office 2018	450	2	6	112,5
Windows 10	0	2	6	0
TOTAL	2450		TOTAL	612,5

Tabla 20. Costes indirectos del proyecto.

- **Costes totales del proyecto:**

En la tabla 21 se recogen los gastos totales del proyecto.

Concepto	Coste íntegro del proyecto (€)	Coste de amortización (€)
Costes de material	1341.12	391.45
Costes directos	9600	9600
Costes indirectos	2450	612.5
Total (sin IVA)	13391.12	10603.95
Total (IVA 21%)	16203.25	12830.77

Tabla 21. Costes totales del proyecto.

Por tanto, el importe total estimado del proyecto asciende a la cantidad de:

doce mil ochocientos treinta euros con setenta y siete céntimos

Alcalá de Henares, 22 de septiembre de 2021

Firmado: Ismael Miranda Gordo.

Máster Universitario en Ingeniería Industrial.

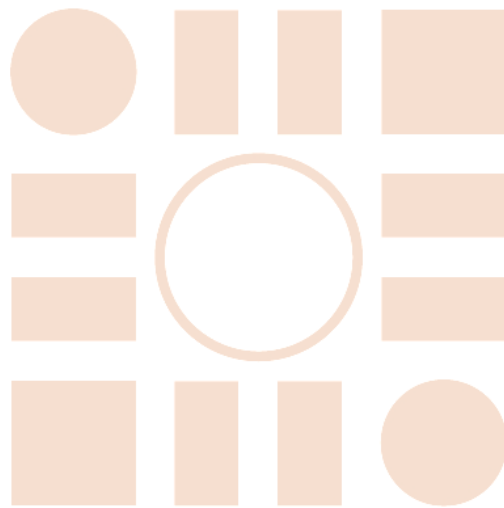
10. Bibliografía

- [1] Introducción al Machine Learning, <https://medium.com/datos-y-ciencia/introduccion-al-machine-learning-una-gu%C3%ADa-desde-cero-b696a2ead359>, último acceso 22/06/2021.
- [2] K means, <https://estrategiastrading.com/k-means/>, último acceso 29/07/2021.
- [3] K medoids, https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_426, último acceso 29/07/2021.
- [4] Affinity propagation clustering, <https://www.toptal.com/machine-learning/clustering-algorithms>, último acceso 29/07/2021.
- [5] Principal Component analysis, <https://www.estadistica.net/Master-Econometria/Componentes Principales.pdf>, último acceso 29/07/2021.
- [6] Aprendizaje por refuerzo, <https://www.kabel.es/aprendizaje-refuerzos/>, último acceso 29/07/2021.
- [7] Algoritmo KNN, <https://anderfernandez.com/blog/programar-knn-r/>, último acceso 22/06/2021.
- [8] Rpubs. Estadística y Machine Learning con R, <https://rpubs.com/PacoParra/293398>, último acceso 08/09/2021.
- [9] Máquinas de Vector Soporte, https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_su_pport_vector_machines, último acceso 25/06/2021.
- [10] Sistemas de localización, <https://www.agps.es/sistemas-de-localizacion/#:~:text=Un%20sistema%20de%20localizaci%C3%B3n%20es,se%20denomina%20Seguimiento%20o%20Tracking>, último acceso 25/06/2021.
- [11] Pascacio, Pável & Casteleyn, Sven & Torres-Sospedra, Joaquín & Lohan, Elena Simona & Nurmi, Jari. (2021). Collaborative Indoor Positioning Systems: A Systematic Review. Sensors. 21. 1002. 10.3390/s21031002.
- [12] Sistemas de localización en interiores basados en radiofrecuencia, <http://www.car.upm-csic.es/lopsi/static/publicaciones/docencia/Apuntes%20RF-LPS.pdf>, último acceso 25/06/2021.
- [13] E. Sansano, “Machine learning-based techniques for indoor localization and human activity recognition through wearable devices”, Tesis doctoral, 2020.
- [14] M. Terán, H. Carrillo, C. Parra. WLAN-BLE Based Indoor Positioning System using Machine Learning Cloud Services.
- [15] R. Montoliu, E. Sansano, A. Gascó, O. Belmonte. “Indoor Posiotioning for Monitoring Older Adults at Home: WiFi and BLE Technologies in Real Scenarios”

- [16] S. Sukreep, C. Nukoolkit y P Mongkolnam. “Indoor Position Detection Using Smartwatch and Beacons”. *Sensors and materials*, 2020.
- [17] K. Komai, M. Fujimoto, Y. Arakawa, H. Suwa, Y. Kashimoto and K. Yasumoto, “Elderly person monitoring in day care center using Bluetooth Low Energy”, 10th International Symposium on Medical Information and Communication Technology (ISMICT), 2016
- [18] Oliver J. Woodman, *An introduction to inertial navigation*. University of Cambridge, 2007. <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>
- [19] D. Gualda, E. Díaz, J. J. García, M. C. Pérez, J. Ureña, R. Cervigón. *Patient Activity Monitoring for Smartphones base on a PDR Algorithm*. IPIN, 2017.
- [20] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar, “Strap-Down Pedestrian Dead-Reckoning System”, *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-7, 2011
- [21] U. Steinhoff and B. Schiele, “Dead Reckoning from the Pocket-An Experimental Study,” *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 162–170, 2010
- [22] D. Gusenbauer, C. Isert, and J. Krosche, “Self-Contained Indoor Positioning on off the Shelf Mobile”, *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-9, 2010.
- [23] E. Munoz Diaz and A. L. Mendiguchia Gonzalez, “Step Detector and Step Length Estimator for an Inertial Pocket Navigation System”, *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014.
- [24] D. Gualda, J. J. García, A. Jiménez, A. Rocandio. “Uso de una Unidad de Medida Inercial para la Ayuda al Diagnóstico de la Fragilidad”, *Actas de la XXXIX Jornadas de Automática*, 2018.
- [25] D. Gualda, E. Díaz, J. J. García, M. C. Pérez, J. Ureña, R. Cervigón. “Patient Activity Monitoring for Smartphones based on a PDR Algorithm”, IPIN, 2017.
- [26] Filtro de Kalman extendido, <https://www.redalyc.org/pdf/707/70732640002.pdf>, último acceso 01/09/2021.
- [27] Diego Galar, Uday Kumar. *eMaintenance, Essential Electronic Tools for Efficiency*. 2017.
- [28] NGIMU, <https://x-io.co.uk/ngimu/>, último acceso 14/07/2021.
- [29] TP-LINK TL-WR802N, <https://www.tp-link.com/es/homenetworking/wifi-router/tl-wr802n/>. último acceso 14/07/2021.
- [30] BQ Aquaris V Plus, <https://www.phonehouse.es/movil/bq/aquaris-v-plus-32gb-3gb-ram.html?tipo=libre>, Último acceso 14/07/2021.
- [31] Aplicación GetSensorData , <https://lopsi.weebly.com/downloads.html> , Último acceso 14/07/2021.

- [32] Baliza: IBKS 105, <https://accent-systems.com/es/producto/ibks-105/> , Último acceso 14/07/2021.
- [33] Baliza: IBKS Plus, <https://accent-systems.com/es/producto/ibks-plus/>, Último acceso 14/07/2021.
- [34] E. Muñoz Díaz, “Inertial Pocket Navigation System: Unaided 3D Positioning,” *Sensors*, vol. 15, pp. 9156–9178, 2015.
- [35] Classification Learner Matlab, <https://es.mathworks.com/help/stats/classificationlearner-app.html>, Último acceso 08/09/2021.
- [36] Distancia de Levenshtein, <https://www.analyticslane.com/2020/06/17/la-distancia-de-levenshtein/> , Último acceso 19/07/2021.
- [37] Cálculo de la similitud, INVE_MEM_2013_181141.pdf (upm.es), Último acceso 19/07/2021
- [38] I. Miranda, D. Gualda, J. J. García, A. Jiménez, S. García, “Symbolic localization of institutionalized patients for detection of daily living activities”, MeMea, 2020.
- [39] I. Miranda, D. Gualda, J. J. García, A. Jiménez, S. García, S. Lluva, “BLE-bases approach for detecting daily routines changes”, MeMea, 2021.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá