



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Sistema de localización de personal a bordo basado en técnicas
de aprendizaje automático y Bluetooth Low Energy*

Grado en Ingeniería Mecánica

ALUMNO: Pedro Javier González Negro

DIRECTORES: Belén Barragáns Martínez
Norberto Fernández García

CURSO ACADÉMICO: 2017-2018

Universida_deVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Sistema de localización de personal a bordo basado en técnicas
de aprendizaje automático y Bluetooth Low Energy*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Conocer la posición del personal a bordo de buques de la Armada es relevante para situaciones de emergencia donde domina la confusión, alertas en circunstancias de “hombre al agua”, control de sistemas de CO2 en compartimentos transitables o restricción del tráfico en determinadas zonas del buque, entre otras aplicaciones. Este Trabajo Fin de Grado tiene como objetivo la búsqueda de una solución a dicho problema, implementando un sistema de radiobaliza basado en el ordenador de placa base Raspberry Pi y la tecnología inalámbrica *Bluetooth Low Energy* (BLE), utilizando la información proporcionada por las radiobalizas para entrenar un sistema de aprendizaje automático basado en máquinas de vector soporte. Este sistema de radiobaliza se instaló en distintos entornos controlados, realizando las pruebas iniciales en la segunda planta del cuartel de alumnos “Marqués de la Victoria” y en los seminarios del CUD, para finalmente validar que el sistema es capaz de identificar el compartimento o sala en la que se encuentra el usuario, en el patrullero “Tabarca”.

PALABRAS CLAVE

Raspberry pi, *Bluetooth Low Energy*, aprendizaje automático, máquinas de vector soporte, localización en interiores.

AGRADECIMIENTOS

Quisiera agradecer a mi familia, amigos y compañeros por su apoyo incondicional y confianza.

A mis tutores, Belén Barragáns Martínez y Norberto Fernández García, por su atención, apoyo y gran ayuda prestada para pulir este trabajo.

Al AN Ione Camacho Lorenzo y al AN Juan Francisco Farrona Marín por su ayuda, experiencia compartida e interés mostrado.

Por último, al Sr. Comandante del patrullero “Tabarca”, el TN Manuel Corral Iranzo, y a su dotación, por su atención, disponibilidad y trato prestado.

CONTENIDO

Contenido	1
Índice de Figuras	5
Índice de Tablas.....	9
1 Introducción y objetivos	11
1.1 Motivación	11
1.2 Objetivos del trabajo	12
1.3 Organización de la memoria	12
2 Estado del arte	15
2.1 Sistema global de navegación por satélite	15
2.2 Tecnologías para localización y posicionamiento en interiores.....	16
2.2.1 WiFi. Protocolo 802.11.....	16
2.2.2 Radio de banda ultra ancha (UWB, <i>Ultra Wide Band</i>)	16
2.2.3 Identificación por radiofrecuencia (RFID, <i>Radio Frequency Identification</i>).....	17
2.2.4 Localización y posicionamiento utilizando telefonía móvil (GSM, <i>Global System for Mobile</i>)	17
2.2.5 ZigBee. Protocolo IEEE 802.15.4	18
2.2.6 Bluetooth.....	18
2.2.7 Localización y mapeado simultáneo.....	20
2.2.8 Infrarrojo.....	20
2.2.9 WiMax	20
2.3 Técnicas de medición de posicionamiento.....	21
2.3.1 Triangulación.....	21
2.3.2 Trilateración.....	21
2.3.3 Medida del nivel de señal recibida	22
2.3.4 Medida de transferencia de datos	22
2.3.5 Medida de conexión a partir de división del espacio.....	23
2.3.6 Métodos KNN.....	23
2.3.7 Métodos bayesianos.....	24
2.3.8 <i>Support Vector Machine</i>	24
2.3.9 Métodos basados en redes neuronales	24
2.4 Aprendizaje automático	25
2.4.1 Tipos de tareas	26
2.4.2 Tipos de algoritmos	27
2.4.3 Modelos de clasificación	27

2.4.4 Aplicaciones	27
2.5 Máquinas de vector soporte	27
2.5.1 Función Kernel	29
2.6 Actividad empresarial y académica en el área de los sistemas de localización en interiores	31
2.6.1 Ekahau	32
2.6.2 AeroScout	32
2.6.3 PanGo Networks	32
2.6.4 Telefónica I+D. Localización en entornos WiFi	33
2.6.5 Sistemas de localización de la Universidad de Trento basado en redes neuronales	33
2.6.6 NAPT system	34
2.6.7 Sistema de Localización de Personal desarrollado por FABAs	34
2.6.8 <i>Real-time Location on Board Ships</i> (Omnisense)	35
2.6.9 <i>On Board Personnel Tracking System</i> (Martec)	35
2.6.10 iBeacon (Apple)	35
2.6.11 Eddystone (Google)	36
2.6.12 Estimote	37
2.6.13 Situm	39
3 Desarrollo del TFG: sistema de localización	41
3.1 Recursos empleados	41
3.1.1 Hardware	41
3.1.2 Software	44
3.2 Configuración	44
3.2.1 Instalación del sistema operativo Raspbian y funcionamiento en modo BLE	44
3.2.2 Operación en modo radiobaliza iBeacon	45
3.2.3 Operación en modo escáner	47
3.3 Fase de experimentación	48
3.3.1 Experimento 1	48
3.3.2 Experimento 2	56
3.3.3 Experimento 3	57
3.3.4 Experimento 4	60
3.3.5 Experimento 5	63
3.3.6 Experimento 6	64
4 Prueba y Validación a bordo del patrullero “Tabarca”	73
4.1 Prueba 1	74
4.2 Prueba 2	78
5 Conclusiones y líneas futuras	81

6 Bibliografía.....	83
Anexo I: Glosario	89
Anexo II: Script para el modo escáner: Formato SVM ^{light}	91
Anexo III: Script para el modo escáner: formato para los primeros experimentos.....	93
Anexo IV: Script para análisis SVM: <i>Experimento 6</i>	95
Anexo V: Script para análisis SVM: VALIDACIÓN EN EL PATRULLERO “TABARCA”	97
Anexo VI: Resultados de la fase de experimentación. <i>Experimento 1</i>	99
Anexo VII: Resultados de la fase de experimentación. <i>Experimento 2</i>	109
Anexo VIII: Resultados de la fase de experimentación. <i>Experimento 4</i>	113

ÍNDICE DE FIGURAS

Figura 2-1 Comparación del uso de ancho de banda de diferentes tecnologías. Fuente: [2]	17
Figura 2-2 Ejemplo aplicación tecnología ZigBee. Fuente: [7]	18
Figura 2-3 Ejemplo de mapeado a partir de la técnica SLAM. Fuente [11]	20
Figura 2-4 Esquema descriptivo del método AOA. Fuente [13].....	21
Figura 2-5 Esquema descriptivo del método de trilateración. Fuente: [13]	21
Figura 2-6 Esquema descriptivo del método de medida por nivel de señal recibida. Fuente: [13] .	22
Figura 2-7 Esquema representativo del método BER. Fuente: [2].....	22
Figura 2-8 Esquema representativo del método basado en medida de conexión. Fuente: [2]	23
Figura 2-9 Modelo explicativo para algoritmos KNN. Fuente: [14].....	24
Figura 2-10 Ejemplos representativos de clases de problemas que resuelve el <i>machine learning</i> . Fuente: [15].....	26
Figura 2-11 Ejemplo representación modelo SVM. Fuente: [21]	28
Figura 2-12 Ejemplo extensión a reglas no lineales. Fuente: [22]	29
Figura 2-13 Polinomial. Fuente: [23]	30
Figura 2-14 Perceptrón. Fuente: [23]	30
Figura 2-15 Función de base radial Gaussiana. Fuente [23]	30
Figura 2-16 Obtención de rutas de carritos de un supermercado. Fuente [2]	31
Figura 2-17 Etiquetas WiFi del sistema Ekahau. Fuente [3].....	32
Figura 2-18 Distribución del sistema de localización de personal de los BAM. Fuente: [1].....	34
Figura 2-19 Proveedores de servicios de ubicación que desarrollan Eddystone. Fuente: [43]	36
Figura 2-20 Fabricantes de Beacons que desarrollan Eddystone. Fuente: [43]	37
Figura 2-21 Ejemplo aplicación <i>Estimote Indoor Location SDK</i> y dispositivos <i>Location Beacons</i> . Fuente: [44].....	38
Figura 2-22 Ejemplo de aplicaciones Situm en compañías de seguridad. Fuente: [45].....	40
Figura 3-1 Componentes de la Raspberry Pi 3, Modelo B. Fuente: [48]	42
Figura 3-2 Ejemplo de ejecución del comando <i>LE Set Advertising Data</i> en la RPi2	46
Figura 3-3 Ejemplo de ejecución del comando <i>LE Set Advertising Parameters</i> en la RPi2.....	46
Figura 3-4 Ejemplo de ejecución del comando <i>LE Set Advertising Enable</i> en la RPi2	47
Figura 3-5 Datos volcados tras ejecutar el script <i>scansvm.sh</i>	47
Figura 3-6 Formato de salida tras ejecutar el script <i>scan.sh</i>	48
Figura 3-7 Zona central, segunda planta, Cuartel “Marqués de la Victoria”. Disposición <i>Experimento 1</i>	48
Figura 3-8 Posición horizontal (izquierda), posición vertical (derecha)	50
Figura 3-9 Ejemplo visualización de la aplicación <i>Android BLE Scanner & Recorder</i>	50

Figura 3-10 Ejemplo archivo Excel generado por la aplicación <i>Android BLE Scanner & Recorder</i>	51
Figura 3-11 <i>Experimento 1</i> , Prueba 5	51
Figura 3-12 <i>Experimento 1</i> , Prueba 2	52
Figura 3-13 <i>Experimento 1</i> , Prueba 3	52
Figura 3-14 <i>Experimento 1</i> , prueba 5. Escáner móvil	53
Figura 3-15 <i>Experimento 1</i> , prueba 4. Escáner RPi1	53
Figura 3-16 <i>Experimento 1</i> , Prueba 6	54
Figura 3-17 <i>Experimento 1</i> , Prueba 7	54
Figura 3-18 <i>Experimento 1</i> , Prueba 7. Periodo de 100 ms	54
Figura 3-19 <i>Experimento 1</i> , Prueba 8. Periodo de 100 ms (eliminando 4 de cada 5 muestras)	55
Figura 3-20 <i>Experimento 1</i> , Prueba 9. Periodo de 500 ms	55
Figura 3-21 <i>Experimento 1</i> , Prueba 10	55
Figura 3-22 Seminarios del CUD. Disposición <i>Experimento 2</i> , Prueba 1	56
Figura 3-23 Seminarios del CUD. Disposición <i>Experimento 2</i> , Pruebas 2, 3 y 4	57
Figura 3-24 Zona central, segunda planta, Cuartel “Marqués de la Victoria”. Disposición <i>Experimento 3</i>	58
Figura 3-25 Resultados del <i>Experimento 3</i>	59
Figura 3-26 Gráfica representativa del efecto de atenuación de las paredes. Ver Figura 3-28	60
Figura 3-27 Gráfica representativa tras colocar las RPi5 y RPi2 como se muestra en la Figura 3-28	61
Figura 3-28 Disposición, escenario y resultados del <i>Experimento 5</i>	62
Figura 3-29 Disposición, escenario y resultados del <i>Experimento 5</i> , Prueba 1	63
Figura 3-30 Disposición escenario y resultados del <i>Experimento 5</i> , Prueba 2	64
Figura 3-31 Ejemplo de ejecución del módulo de aprendizaje	67
Figura 3-32 Ejemplo de ejecución del módulo de clasificación	67
Figura 3-33 Disposición <i>Experimento 6</i>	68
Figura 3-34 Captura de un segmento del archivo <i>RESULTADOS.txt</i>	70
Figura 3-35 Resultados de la precisión del conjunto de clasificación. <i>Experimento 6</i>	71
Figura 3-36 Leyenda con escala de colores	71
Figura 4-1 Cubierta puente. Prueba 1	75
Figura 4-2 Cubierta castillo. Prueba 1	75
Figura 4-3 Cubierta principal. Prueba 1	75
Figura 4-4 Cubierta baja. Prueba 1	75
Figura 4-5 Leyenda con escala de colores	77
Figura 4-6 Cubierta puente. Prueba 2	78
Figura 4-7 Cubierta castillo. Prueba 2	78

Figura 4-8 Cubierta principal. Prueba 2	78
Figura 4-9 Cubierta baja. Prueba 2.....	78
Figura 4-10 Leyenda con escala de colores.....	80

ÍNDICE DE TABLAS

Tabla 2-1 Ejemplo de vectores de prueba de una base de datos para uso de KNN. Fuente: [3].....	23
Tabla 2-2 Características Beacons Estimote. Fuente: [44].....	39
Tabla 3-1 Conversión de intervalos de repetición.....	46
Tabla 4-1 Resultados prueba 1. Kernel lineal	76
Tabla 4-2 Resultados prueba 1. Kernel radial (gamma 0,001).....	76
Tabla 4-3 Resultados prueba 1. Kernel radial (gamma 0,01).....	76
Tabla 4-4 Resultados prueba 1. Kernel radial (gamma 0,1).....	76
Tabla 4-5 Resultados prueba 1. Kernel radial (gamma 1).....	77
Tabla 4-6 Resultados prueba 1. Kernel radial (gamma 10).....	77
Tabla 4-7 Resultados prueba 2. Kernel lineal	79
Tabla 4-8 Resultados prueba 2. Kernel radial (gamma 0,001).....	79
Tabla 4-9 Resultados prueba 2. Kernel radial (gamma 0,01).....	79
Tabla 4-10 Resultados prueba 2. Kernel radial (gamma 0,1).....	79
Tabla 4-11 Resultados prueba 2. Kernel radial (gamma 1).....	80
Tabla 4-12 Resultados prueba 2. Kernel radial (gamma 10).....	80

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Motivación

La motivación de este Trabajo Fin de Grado surge de la necesidad de conocer la localización del personal a bordo de buques de la Armada, especialmente en determinados entornos, donde esta información sería de gran ayuda. Entre la gran variedad de aplicaciones que nos aportaría dicho sistema, destacaríamos:

- Zafarrancho de combate: En este caso, la localización del personal a bordo sería una herramienta muy útil para los trozos de seguridad interior, que tendrían claro en todo momento dónde se encuentra el personal que compone dichos trozos y su disponibilidad. También nos permitiría comprobar si algún miembro de la dotación ha quedado atrapado en algún compartimento o sala, en caso de un incendio o una inundación, sin tener que acceder físicamente a ella.
- “Hombre al agua”: Cuando se produce un “hombre al agua” y ningún miembro de la dotación se ha dado cuenta de esta situación, la probabilidad del salvar al náufrago es prácticamente nula. Disponer de un sistema de localización en el buque que nos permita alertar cuando se pierde la señal de un usuario de este sistema, nos permitiría alertar de situaciones de “hombre al agua” en el momento que se produce y salvar vidas.
- Compartimentos con sistemas contraincendios de CO₂: Con un sistema de localización, se podría desactivar automáticamente el sistema CO₂ en el caso de que algún miembro de la dotación se encontrase en el interior de dicho compartimento o sala.
- Zonas restringidas: El tránsito nocturno por exteriores o con mal tiempo está prohibido, salvo en situaciones que se requieran. Este sistema alertaría de los casos en los que no se esté respetando dicha norma. Por otro lado, también nos permitiría alertar en caso de que un miembro de la dotación accediera a una zona restringida, sin tener la autorización pertinente para ello.
- Botón de socorro: Nos permitiría conocer de manera instantánea la localización de un miembro de la dotación que se encontrase en peligro.

Teniendo en cuenta la gran variedad de aplicaciones que aportaría este sistema, algunas de ellas incluso para salvar vidas, en este TFG se plantea una solución que nos permite implementar este sistema de localización de personal a bordo en los buques de la Armada.

Debido a que la localización en interiores es un ámbito sin una solución tan aceptada y extendida como es el GPS para el posicionamiento en exteriores, hemos tenido que decantarnos por una de las muchas posibilidades que se han probado en esta área. Finalmente, se optó por implementar un sistema basado en el ordenador de placa reducida Raspberry Pi (RPi), operando como radiobaliza gracias a la tecnología *Bluetooth Low Energy*, e integrando un sistema de aprendizaje de máquina basado en máquinas de vector soporte para cumplir con el objetivo principal de este TFG: conseguir identificar el compartimento o sala en la que se encuentra un usuario de este sistema. Aunque lo mencionaremos también en el Capítulo 2, adelantamos aquí que existe un TFG previo [1] cuyo objetivo era obtener un sistema de control de personal a bordo empleando placas RPi y tecnología *Bluetooth Low Energy*. En este TFG se pretende ser más ambicioso y, partiendo de esas mismas tecnologías y apoyándonos en las técnicas de aprendizaje basado en máquinas de vector soporte, evolucionar de un sistema de control a un sistema de localización.

1.2 Objetivos del trabajo

Los objetivos de este Trabajo Fin de Grado se enumeran con detalle a continuación:

- Desarrollar un sistema de localización de personal a bordo de buques de la Armada, a partir de radiobalizas, basado en la placa Raspberry Pi y la tecnología inalámbrica *Bluetooth Low Energy* (BLE).
- Prueba de dicho sistema en entornos controlados, como los seminarios del CUD o las camaretas del cuartel de alumnos “Marqués de la Victoria”.
- Validación del sistema en un entorno militar, como es el patrullero “Tabarca”.
- Usar el número mínimo de radiobalizas imprescindibles para poder situarnos correctamente, es decir, no tener la necesidad de usar una radiobaliza por compartimento o sala.
- Por último, nos interesa que los criterios para colocar las radiobalizas fuesen suficientemente sencillos, de manera que, en la instalación de este sistema, tuviesen numerosas posibilidades a la hora de disponer las radiobalizas.

1.3 Organización de la memoria

La memoria de este TFG está dividida en cinco capítulos, complementados con ocho anexos.

En el presente capítulo se describe la motivación del trabajo, los objetivos marcados y la organización de la memoria.

En el Capítulo 2 desarrollamos el estado del arte. Se introducen los sistemas de localización, comentando primero los sistemas de localización en exteriores, para centrarnos, después, en las tecnologías enfocadas al posicionamiento en interiores. Aquí se explica la tecnología *Bluetooth* utilizada en nuestro sistema de localización. Posteriormente, describimos las diferentes técnicas de medición de posicionamiento, dedicando un apartado entero para el aprendizaje automático y otro para máquinas de vector soporte, áreas donde se encuadra la técnica de medición de posicionamiento que hemos utilizado en este TFG. Por último, se explica la actividad industrial desarrollada en el ámbito de la localización en interiores.

El Capítulo 3 se divide en tres partes. En primer lugar, se describe el software y hardware empleado. A continuación, se explican los pasos iniciales tomados en este trabajo, la configuración y puesta en marcha de las placas Raspberry Pi para poder operar con ellas. Finalmente, se detalla los experimentos realizados, previos a la validación del sistema. Se realiza una primera fase (los primeros cinco experimentos), donde se estudia si la señal y las condiciones del entorno son favorables para implementar las técnicas propuestas por nuestro sistema. En la segunda fase (último experimento) se

realiza una prueba en la que se hace uso de las librerías SVM, describiendo todos los pasos para su instalación, configuración y empleo.

Este apartado de experimentación está compuesto de diferentes pruebas que se estructuran de la siguiente manera:

1. Descripción del escenario o entorno controlado.
2. Explicación de los pasos a seguir para la toma de datos y representación de los resultados.
3. Análisis de resultados: se detallan las observaciones y conclusiones obtenidas de los resultados.

En el Capítulo 4 se muestra la validación y prueba de este sistema realizado en un entorno militar y marino, el patrullero “Tabarca”. Se presentan las dos pruebas realizadas, siguiendo la misma estructura que en la fase de experimentación.

En el Capítulo 5 se presentan las conclusiones obtenidas y las líneas futuras de investigación propuestas.

En la bibliografía se enumeran los recursos consultados para la realización del TFG, donde quedan reflejadas todas las referencias de la memoria.

Finalmente, se presentan los anexos, donde se adjuntan:

- Glosario de siglas. El significado de siglas utilizadas en este TFG se detalla la primera vez que aparecen en la memoria. Este glosario tiene como objetivo facilitar el trabajo al lector, evitando que tenga que buscar la primera vez que se menciona la sigla que quiera consultar para conocer su significado.
- Conjunto de scripts elaborados para el desarrollo del trabajo.
- Todos los resultados, ordenados y clasificados, de los dos primeros experimentos del Capítulo 3.

2 ESTADO DEL ARTE

Las tecnologías enfocadas al desarrollo de localización en interiores se encuentran en plena expansión debido al reciente interés y a las deficiencias de la geolocalización vía satélite en zonas de sombra y lugares donde se produce la pérdida o atenuación de señal de los satélites, como en interiores de edificios o a bordo de buques.

En el presente capítulo comenzaremos explicando los sistemas de localización en exteriores o de geolocalización, para entender por qué estas tecnologías no son aplicables en la localización en interiores. A continuación, se expone un repaso a las tecnologías que han tenido éxito en el desarrollo de sistemas de localización en interiores. En el siguiente apartado, se describen las técnicas de medición de posicionamiento, centrándonos en los métodos de aprendizaje automático basados en máquinas de vector soporte. Finalmente, desglosamos los principales proyectos y empresas que desarrollan o han desarrollado trabajos dentro de este ámbito, incluido el sistema puesto en funcionamiento en los Buques de Acción Marítima de la Armada Española.

2.1 Sistema global de navegación por satélite

Estos sistemas son la solución más extendida para obtener localización rápida y precisa en cualquier punto del globo. Los países o comunidades que desarrollan este tipo de sistema son EE.UU. (GPS), Rusia (GLONASS), China (BEIDU) y Europa (GALILEO).

Un sistema global de navegación por satélite está formado por una constelación artificial de al menos 24 satélites que emiten continuamente señales donde indican su posición y tiempo de envío de la señal. Cualquier dispositivo que sea capaz de recibir e interpretar la señal de al menos cuatro satélites (si el número es mayor obtendrá un resultado más preciso) consigue calcular por triangulación su posición absoluta en coordenadas geográficas de latitud, longitud y altitud. La distancia entre el satélite y receptor se conoce gracias al retardo de la señal.

Dependiendo del sistema global de navegación por satélite, del uso (militar o civil), de las características técnicas del dispositivo receptor y otros parámetros condicionantes hoy en día se puede obtener una posición con una precisión inferior a un metro de error. Es importante saber que la precisión es un parámetro controlable, permitiendo la máxima precisión a las instituciones y aplicaciones militares del propio país que desarrolla este sistema, introduciendo un margen de error para aplicaciones civiles.

Sin embargo, en determinados entornos exteriores, y en la mayoría de los interiores, el posicionamiento por satélite no es operativo por las siguientes condiciones:

- Las señales de los satélites no tienen la potencia suficiente para atravesar la mayoría de los obstáculos. Se produce una atenuación importante de la señal debido a que los satélites se encuentran a gran distancia de los receptores y su diseño, limitado por el peso y el tamaño, no les permite portar grandes antenas de frecuencias inferiores que tienen mayores alcances.
- Además, necesitan línea de visión directa (LoS, *Line of Sight*), por lo que, aunque nos encontremos en espacios abiertos, como puede ser la calle de una ciudad, los edificios altos obstaculizarían la línea de visión directa de algunos satélites, impidiendo que nuestro receptor pueda establecer conexión con al menos cuatro satélites.
- Por último, los sistemas de localización en interiores están pensados para funcionar en entornos locales, por lo que es preferible el uso de una referencia relativa, a partir de zonas de interés y no de una referencia absoluta como la que utilizan los sistemas de navegación por satélite, que a partir de un punto de referencia *datum* establecen un sistema de referencia cartesiano X, Y, Z.

2.2 Tecnologías para localización y posicionamiento en interiores

Los sistemas de localización en entornos locales (*LPS, Local Positioning Systems*) son sistemas alternativos a los sistemas de navegación por GPS para la localización en interiores. Actualmente, las tecnologías basadas en radiofrecuencia representan la cúspide de estos sistemas. A continuación, se detallarán los sistemas de localización en interiores [2].

2.2.1 WiFi. Protocolo 802.11

El protocolo 802.11 suele seguir un modelo de comunicación típicamente centralizado. De esta manera, las redes suelen estar formadas por uno o varios puntos de acceso (APs: *Access Points*) y numerosos clientes conectados a uno de los APs, representados por tarjetas de red inalámbrica. En un sistema de localización, donde se utilice el protocolo 802.11, los puntos de accesos operarían como radiobalizas, emitiendo periódicamente tramas de datos, que serían recibidas por las tarjetas de red inalámbrica [3].

Los protocolos 802.11b y 802.11g emiten en la franja de frecuencia cercana a los 2,4 GHz, subdividido en 14 canales, separados por 5 MHz. Cada país y zona geográfica aplica sus propias restricciones al número de canales disponibles, en Europa, por ejemplo, disponemos de 13 canales. WiFi utiliza el mecanismo de CSMA-CA para eliminar las interferencias entre APs, reduciendo la probabilidad de que se transmita a la vez en el mismo canal por diferentes puntos de acceso.

2.2.2 Radio de banda ultra ancha (UWB, *Ultra Wide Band*)

Esta tecnología se empezó a usar en el ámbito militar, en radares y comunicaciones seguras. A partir de 1990 se licenció para usos civiles. En la Figura 2-1 se muestra una gráfica representativa del uso del ancho de banda por las tecnologías de banda estrecha, banda ancha y banda untra-ancha (UWB). *Ultra Wide Band* presenta grandes ventajas que la convierten en una buena candidata para la localización en interiores:

- Con respecto a otras tecnologías de radiofrecuencia, esta cuenta con una gran capacidad de detección, propagación en interiores y penetración en materiales presentes en edificios.
- Sufre menos interferencias por su baja densidad espectral.
- Su elevado ancho de banda permite una gran resolución en la medida de los retardos, con un margen de error aproximado de 1 ns.
- Precisión en localización en interiores de hasta un metro.

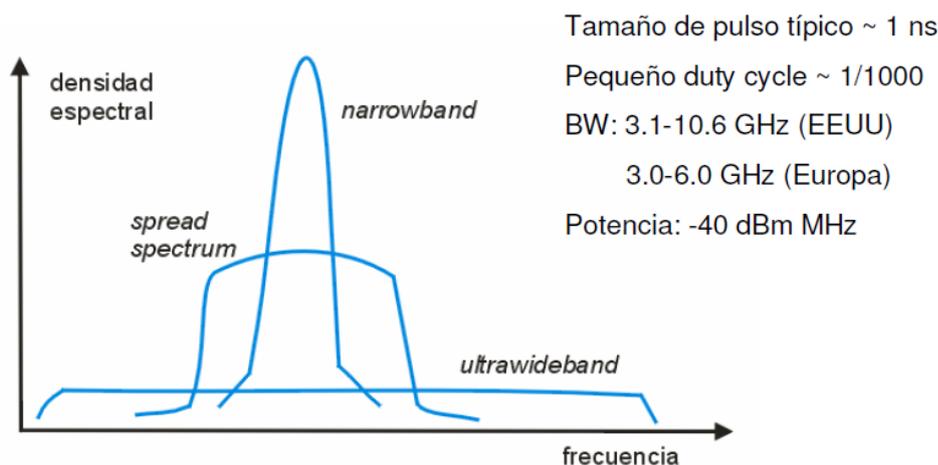


Figura 2-1 Comparación del uso de ancho de banda de diferentes tecnologías. Fuente: [2]

Entre las desventajas principales, podemos destacar:

- Necesidad de filtro adaptado para la detección de llegada de pulsos.
- Requiere de una sincronización precisa entre los receptores.
- La estimación de distancia se limita al cálculo de tiempo de llegada, que se explica en el apartado 2.3, siendo casi inservibles otras técnicas como la de triangulación.
- Esta tecnología se encuentra en proceso de estandarización.

2.2.3 Identificación por radiofrecuencia (RFID, Radio Frequency Identification)

La tecnología RFID es un sistema de identificación basado en etiquetas de radiofrecuencia que contienen una antena que recibe una señal por RF de un transmisor y devuelve una señal al ser excitada por la señal que le llega de este transmisor. Para situarse una persona mediante este sistema, portaría el transmisor que excita a las etiquetas y a la vez lee la señal que devuelven éstas, siendo capaz de identificar qué etiquetas tiene cerca y a partir de esta información, conocer su localización [3].

Cricket [4], un sistema desarrollado por un grupo de ingenieros del Instituto de Tecnología de Massachussets consiguió tener una precisión de 2 centímetros. Es por ello por lo que ha sido implementado en otros sistemas, como control de robots, seguimientos de objetos o aplicaciones *context-aware* (en las que la localización forma una parte esencial).

El inconveniente de la tecnología RFID es que, al necesitar una gran cantidad de lectores RFID para una correcta localización (el alcance de la señal es muy reducido), se convierte en un sistema caro de implementar. Es por ello por lo que se opta por otras tecnologías en este TFG.

2.2.4 Localización y posicionamiento utilizando telefonía móvil (GSM, Global System for Mobile)

La ventaja de esta tecnología es que no necesitaría ningún hardware adicional, se aprovecharía de las tecnologías GSM ya instaladas. Actualmente, existen operadores de telefonía móvil que ofrecen la opción de localización vía móvil a sus clientes. Su principal desventaja es la falta de precisión, ya que habitualmente, este tipo de tecnología no puede dar precisiones mayores de 50 metros, por lo que no serviría para interiores. Esto es debido a que el cálculo de la localización se basa en los siguientes dos métodos posibles:

- *Cell ID*: Detección de la célula a la que está conectado el móvil. La precisión depende del tamaño de célula. En zonas urbanas la precisión es de decenas de metros, mientras que en zonas rurales la precisión disminuye.
- *E-OTD (Enhanced Observed Time Difference)*: En español, diferencia de tiempos mejorada, es un método por el cual el móvil mide la diferencia de tiempos de vuelo entre señales transmitidas a varias estaciones base. La precisión de esta técnica se encuentra entre los 50 y 200 metros.

En [5] podemos encontrar un sistema experimental basado en la medida de la intensidad de señal de 6 estaciones base y 30 canales GSM (este sistema no permite ser utilizado para la comunicación) mediante un módem GSM. Utiliza el método de *k-vecinos* más cercanos, necesita un *fingerprint* denso del área y finalmente, se puede obtener una precisión con 2,5 m de mediana de error.

2.2.5 ZigBee. Protocolo IEEE 802.15.4

Esta tecnología se ha diseñado para ser el estándar de las comunicaciones entre las redes inalámbricas de sensores (*WSN, Wireless Sensor Network*) [6]. En la Figura 2-2 podemos ver un ejemplo de distribución de tecnología ZigBee dentro de salón. Entre sus principales características, podemos destacar: bajo consumo energético, bajo coste y transmisión de datos a baja velocidad. Por otro lado, su ancho de banda y su alcance es más reducido que en tecnologías WiFi, por ejemplo, y la señal fluctúa de igual manera cuando cambian las condiciones de la planta, o con el simple movimiento de las personas.

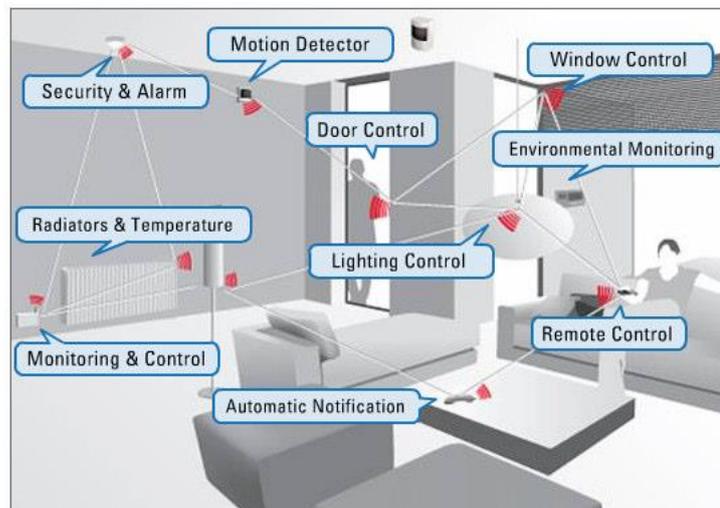


Figura 2-2 Ejemplo aplicación tecnología ZigBee. Fuente: [7]

2.2.6 Bluetooth

Bluetooth [8] es un sistema de transmisión de datos, inalámbrico, entre dispositivos a corta distancia. La banda de frecuencia se mueve en un rango que oscila entre los 2,4 y 2,48 GHz. Ericsson fue la primera empresa en investigar sobre sistemas de comunicación entre dispositivos basados en una interfaz de bajo consumo y coste, destinada a la transmisión y recepción de datos entre teléfonos móviles y otros dispositivos. En 1999, dicha empresa se juntó con Intel, Nokia, Toshiba e IBM para crear el SIG de Bluetooth (*Special Interest Group*) a las que se les unió Microsoft, 3COM, Motorola y Lucent para desarrollar esta tecnología.

A continuación, se detallan las diferentes versiones de Bluetooth:

- Bluetooth v1.0 y v1.0B: Estas primeras versiones de Bluetooth, prácticamente obsoletas en la actualidad, dieron problemas de compatibilidad entre compañías y

prescindían de privacidad, ya que, en cada transmisión de datos, enviaban nuestra dirección privada de dispositivo Bluetooth. La tasa de transmisión giraba en torno a los 721 kbps.

- Bluetooth v1.1: Utiliza el estándar IEEE 802.15.1-2002. Corrige los errores, que hemos indicado, de las versiones anteriores; se añaden y soportan los canales no encriptados; y se le añade el indicador de la señal, RSSI.
- Bluetooth v1.2: Usa el nuevo protocolo estándar IEEE 802.15.1-2005; mejora la velocidad de conexión y transferencia de datos; se le añade la función de detección de otros dispositivos bluetooth en el radio de actuación; mejora notablemente la calidad de audio; se le añade HCI (*Host Controller Interface*), control de flujo y modos de retransmisión L2CAP (*Logical Link Control and Adaptation Protocol*).
- Bluetooth v2.0 + EDR (*Enhanced Data Rate*): EDR es una característica opcional de transmisión de datos a mayor velocidad. La v2.0 tiene una transferencia máxima de datos de 3Mb/s, siendo su tasa real máxima 2,1Mb/s.
- Bluetooth v2.1 + EDR: Mejoras importantes en cuanto a la seguridad de datos y consumo de energía. En relación con la seguridad, se le añade SSP (*Secure Simple Pairing*) que permite mejorar el filtrado de datos.
- Bluetooth v3.0 + HS: Supone un gran salto en la velocidad de transmisión de datos, 24Mb/s. Como novedad, esta versión incluye la *Alternativa MAC / PHY*, la cual le permite hacer uso del WiFi para el envío y recepción de grandes paquetes de datos, utilizando el estándar 802.11 de alta velocidad.
- Bluetooth v4.0: Lanzada en el año 2010, aumenta la velocidad de transmisión y recepción de datos a 32Mb/s. Esta versión mejora la combinación del Bluetooth clásico con la conexión inalámbrica vía WiFi. Añade la tecnología BLE (*Bluetooth Low Energy*), lo que genera un consumo menor de energía, algo muy útil para aquellos dispositivos diseñados para usos prolongados. Los dispositivos que puedan interactuar tanto con dispositivos Bluetooth clásicos como con dispositivos BLE, son considerados como *Bluetooth Smart Ready*.
- Bluetooth v5.0: Lanzado oficialmente en verano de 2016; dobla la velocidad de transmisión de datos, con respecto a la versión anterior, el ancho de banda se multiplica por ocho, mejorando la transmisión y recepción de gran cantidad de información de manera instantánea, rápida y estable; y el alcance es cuatro veces mayor. Esta versión también ha conseguido implementar un nuevo sistema para evitar interferencias con otros dispositivos inalámbricos. Por otro lado, Bluetooth v5.0 soporta BLE, de esta manera se reduce el impacto que el aumento de velocidad y alcance tienen sobre el consumo de energía.

Actualmente podemos diferenciar cuatro clases de dispositivos con módulos Bluetooth, clasificados de la siguiente manera:

- Clase 1: Alcance de unos 100 m y un consumo de potencia de unos 100 mW.
- Clase 2: Alcance de hasta 20 m y una potencia requerida de 2,5 mW.
- Clase 3: Opera con un alcance de 1 m y una potencia de 1 mW.
- Clase 4: Su alcance no supera los 0,5 m y consume una potencia de 0,5 mW.

Para aquellos lectores que deseen más información sobre el protocolo Bluetooth, podemos recomendar la consulta del estado del arte del TFG [1], trabajo que sirve de punto de partida a este TFG y donde se emplea la tecnología BLE para un sistema de control de personal a bordo.

2.2.7 Localización y mapeado simultáneo

La tecnología SLAM (*Simultaneous Location and Mapping*) [9], a diferencia de la mayoría de los sistemas actuales de localización, no utiliza tecnología RF, sino que funcionan mediante exploración visual. Sin embargo, debe quedar claro que la misión principal de la tecnología SLAM es el posicionamiento de la propia máquina en un a partir de la interpretación visual que hace del mismo, siendo la localización una posibilidad a mayores a partir de los recursos que ofrece el sistema de posicionamiento, de tal manera que, cuando la máquina informa de su posición, esta información se puede interpretar y obtener su localización.

La técnica SLAM utilizada por las máquinas consiste en realizar una exploración del entorno, extrayendo los puntos más característicos de éste, para ir creando simultáneamente su propio mapa interno, como se puede observar en el ejemplo de la Figura 2-3, mediante el algoritmo de visión por computador SURF (*Speeded-Up Robust Features*) [10].

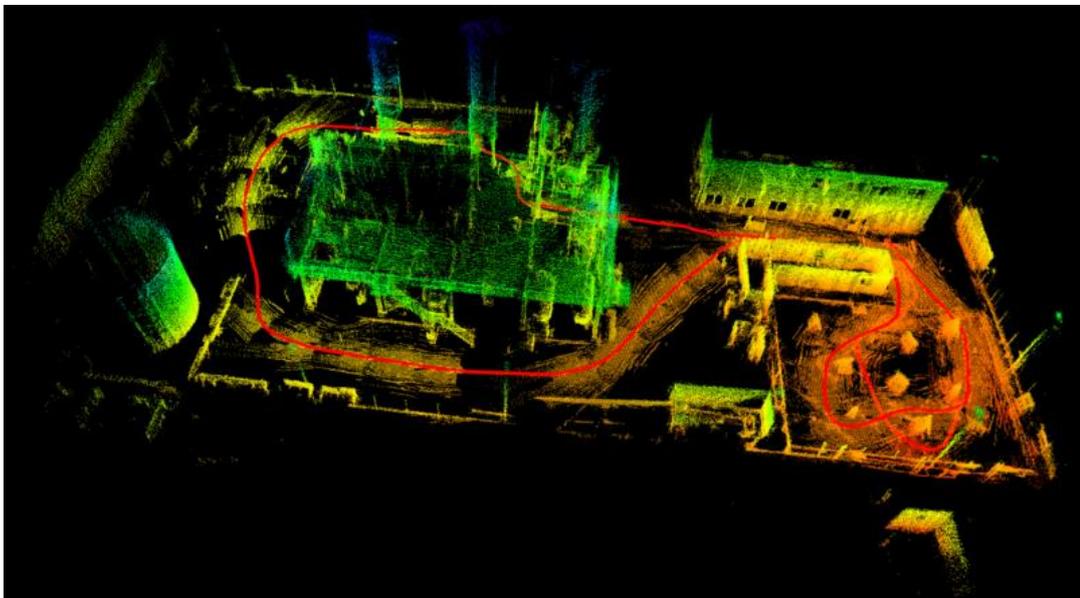


Figura 2-3 Ejemplo de mapeado a partir de la técnica SLAM. Fuente [11]

A continuación, se presentan otras técnicas de menor interés. Se trata de tecnologías en las que, tras haber intentado desarrollar sistemas LPS a partir de las mismas, se han encontrado grandes carencias.

2.2.8 Infrarrojo

El problema de esta tecnología principalmente es el alcance, que es de pocos metros, ya que utiliza frecuencias de transmisión mucho más altas que otras tecnologías como Bluetooth. Además, al producirse una atenuación de la señal muy alta, sería imposible recibir la señal de las balizas si no fuera por línea directa.

Sin embargo, existen proyectos, sobre todo basados en detectar y controlar el paso de personal por puntos específicos de una zona local. Un proyecto llamado WIPS (*Wireless Indoor Positioning System*) [12] utiliza la localización por infrarrojos en dispositivos beacons y *SmartBadges*.

2.2.9 WiMax

El protocolo IEEE 802.16 o WiMax es un conjunto de estándares inalámbricos de banda ancha fija para las redes de acceso metropolitanas. El gran inconveniente de esta tecnología, para ser implantada

en un sistema de localización en interiores, es que está pensada para la intercomunicación de áreas muy extensas. Puede llegar a transmitir hasta 70Mbps y sus alcances son de casi 50km.

2.3 Técnicas de medición de posicionamiento

A continuación, se especifican las técnicas de medición apoyadas en algoritmos geométricos.

2.3.1 Triangulación

Es una técnica basada en la medida de la dirección de llegada de la señal (AOA, *Angle of arrival*) como se puede observar en la Figura 2-4.

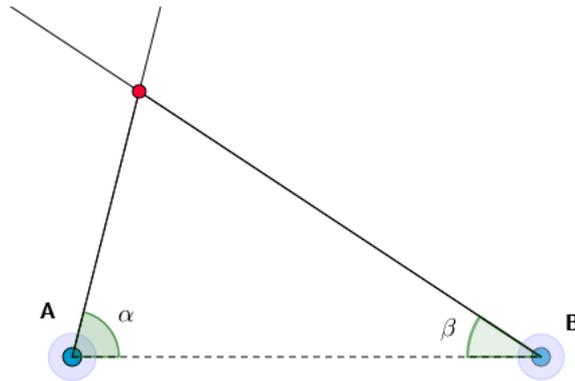


Figura 2-4 Esquema descriptivo del método AOA. Fuente [13]

Gracias a esta técnica no es necesario sincronizar las balizas con el objeto a localizar. Son necesarias antenas direccionales o arrays con medida de diferencia de fase. El resultado geométrico se obtiene a partir de intersecciones de rectas o planos.

2.3.2 Trilateración

Esta técnica está basada en la medida de rangos o pseudorangos. Al igual que la técnica de triangulación, se basa en geometría de triángulos, pero en lugar de utilizar medidas de ángulos y al menos una distancia conocida, la trilateración utiliza la distancia conocida entre el sujeto y cada punto de referencia. En la Figura 2-5 podemos obtener una visión representativa de este método.

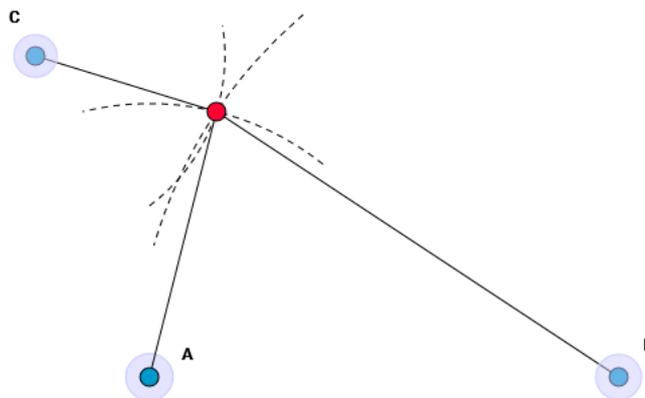


Figura 2-5 Esquema descriptivo del método de trilateración. Fuente: [13]

En este caso, entre los posibles parámetros observables tenemos:

- Tiempo de llegada o de propagación (TOA/TOF, *Time of Arrival/Flight*). Debe haber sincronización entre la baliza y el objeto a localizar. El resultado geométrico se obtiene a partir de intersecciones de esferas.

- Diferencia de tiempo de llegada (DTOA, *Differential Time of Arrival*). Es necesario sincronía entre balizas, pero no con el objeto a localizar. El resultado geométrico se obtiene a partir de intersecciones de hiperboloides.
- Tiempo de propagación de ida y vuelta de la señal (RTOF, *Round-trip Time of Flight*). No es necesaria sincronía entre ningún dispositivo. El resultado geométrico se obtiene a partir de intersecciones de esferas.

2.3.3 Medida del nivel de señal recibida

Es una medida basada en la medida de fuerza de señal (RSSI, *Received Signal Strength Indicator*). En la Figura 2-6 podemos apreciar como la intensidad de la señal recibida es inversamente proporcional a la distancia entre ésta y el objeto a localizar.

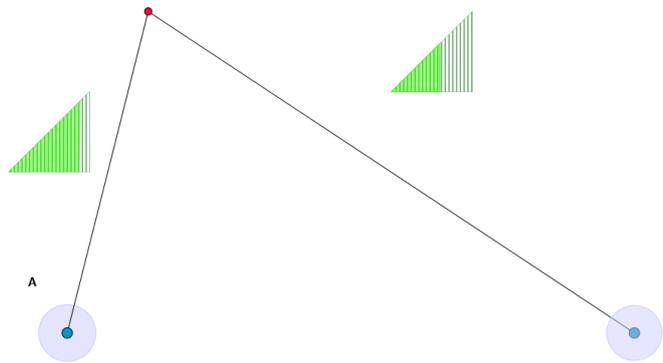


Figura 2-6 Esquema descriptivo del método de medida por nivel de señal recibida. Fuente: [13]

Los niveles de señal medidos pueden ser de tecnologías WiFi, Bluetooth, GSM, RFID, etc. No requieren línea de visión directa entre las balizas y el objeto o dispositivo a localizar, aunque el nivel de señal recibida se ve afectado por el impacto de las paredes como interferencia y esto se debe tener en cuenta para obtener una buena solución.

2.3.4 Medida de transferencia de datos

Dependiendo del parámetro observable, podemos obtener la medición a partir de:

- Calidad de enlace (LQI, *Link Quality Indicator*)
- Tasa de errores de comunicación (BER, *Bit Error Rate*)

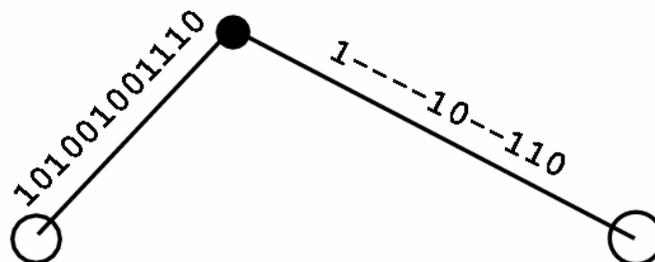


Figura 2-7 Esquema representativo del método BER. Fuente: [2]

Se apoyan en sistemas de comunicación de datos. Como se observa en la Figura 2-7, la tasa de error es inversamente proporcional a la distancia.

2.3.5 Medida de conexión a partir de división del espacio

Esta técnica está basada en la conectividad a nodos pertenecientes a un área o una celda determinada. Todos los nodos tienen la misma cobertura, de manera que dependiendo del nodo del que recibamos la señal, nos encontraremos en un área u otra. En la Figura 2-8 podemos visualizar una sencilla representación de este método.

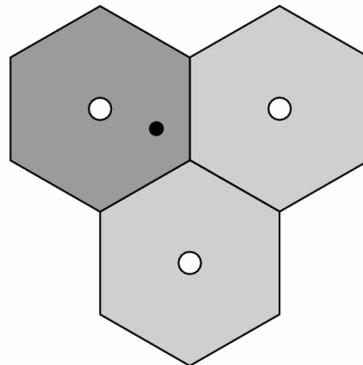


Figura 2-8 Esquema representativo del método basado en medida de conexión. Fuente: [2]

Aparte de las técnicas de medición de posicionamiento a partir de algoritmos geométricos, han surgido nuevas técnicas con el desarrollo del aprendizaje automático, basadas en técnicas de reconocimiento de patrones. En este apartado, explicaremos brevemente dichas técnicas. Al ser una de ellas (máquinas de vector soporte), la empleada en este TFG, se dedicarán los apartados 2.4 y 2.5 para explicar con más detalle qué es el aprendizaje automático y las máquinas de vector soporte.

Las técnicas de reconocimiento de patrones son métodos de aprendizaje automático, que pueden ser empleados (entre otras muchas aplicaciones) para estimar la localización relacionando la potencia recibida de varias balizas en un instante determinado, con una base de datos procedente de un mapeo de muestras obtenido en una calibración. De esa forma evitamos utilizar algoritmos geométricos.

2.3.6 Métodos KNN

KNN (*K-Nearest Neighbors*) es un método de clasificación supervisado, que también se puede utilizar en regresión y que sirve para estimar una función de densidad $F(x/C_j)$ que predice el valor de x para la Clase C_j [14].

Se apoya en la señal recibida (RSSI) de los puntos de acceso, siendo necesario que los radios de cobertura de todos ellos se solapen. Gracias a estos algoritmos no se necesita establecer una relación lineal entre la RSSI recibida y la distancia al punto de acceso. Tan solo nos interesa asegurar que, si el AP (*Access Point*) está cerca, la potencia recibida será mayor que si está lejos. En base a esto, se crea una base de datos a partir de localizaciones, que serán el conjunto de *vectores de prueba*, formado cada vector por una coordenada en el espacio y la RSSI correspondiente de cada AP [3]. En la Tabla 2-1 se muestra un ejemplo de estos datos recogidos en una tabla.

Localización	x (m)	y (m)	z (m)	RSS ₁ (dBm)	RSS ₂ (dBm)
1	0	0	0	-28	-88
2	5	0	0	-35	-70
3	10	0	0	-51	-62
4	0	5	0	-41	-85
5	5	5	0	-48	-68
6	10	5	0	-60	-50
7	10	10	0	-72	-30

Tabla 2-1 Ejemplo de vectores de prueba de una base de datos para uso de KNN. Fuente: [3]

Una vez implementado el sistema, el usuario se localizará comparando la señal recibida (*vector potencia x*) con la base de datos. Esta comparación, base de los algoritmos KNN, busca la clase más frecuente de sus K vecinos más próximos. En la Figura 2-9 podemos ver cómo funciona KNN de una forma representativa. El círculo verde representaría el *vector potencia* obtenida en un instante determinado. El resto serían los *vectores de prueba* de la base de datos. Si $k=3$, la clasificación del *vector potencia* pertenecerá a la clase representada con un triángulo rojo. Si $k=5$, pertenecerá a la clase representada con el cuadro azul.

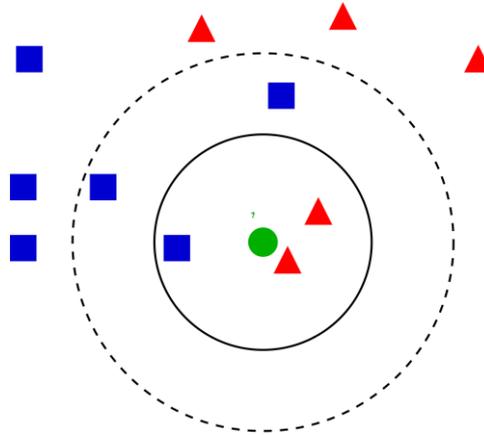


Figura 2-9 Modelo explicativo para algoritmos KNN. Fuente: [14]

2.3.7 Métodos bayesianos

Estos métodos, también denominados *Nibble*, se basan en la construcción de una red bayesiana, como su propio nombre indica, durante la fase de calibración o entrenamiento. El primer paso necesario comprendería que el usuario determinase los puntos de interés, para posteriormente entrenar el sistema durante el tiempo que se crea necesario, durante el cual se capturan muestras de RSSI de los AP.

Los métodos bayesianos son una mejora de los métodos KNN, ya que son más inmunes a los cambios intrínsecos de la señal WiFi, causados por los efectos de multitrayecto. Además, al ser un sistema basado en aprendizaje, no tiene en cuenta la modelización de la señal WiFi. Esto se consigue gracias a que, en la fase de entrenamiento, primero se definen las probabilidades de estar situados en un lugar determinado $p(L)$, y posteriormente, las probabilidades condicionadas de obtener una determinada RSSI de un AP, estando en una localización determinada $p(E/L)$ [3].

2.3.8 Support Vector Machine

Las máquinas de vector soporte (SVM, *Support Vector Machine*) son un método supervisado de clasificación, aunque también puede utilizarse en regresión. Su objetivo principal es realizar una separación de clases, de un conjunto de datos, optimizando la anchura del margen que separa ambas clases de datos. A estos márgenes se les denominan *vectores soporte* [14].

Como ya ha sido puntualizado, este método de posicionamiento a partir de reconocimiento de patrones es el método escogido en nuestro TFG y será desarrollado con más detalle en el apartado 2.5.

2.3.9 Métodos basados en redes neuronales

Las redes neuronales son una buena herramienta para la estimación de la localización, ya que también se basan en la clasificación de conjuntos de datos. Las redes neuronales utilizan el método de perceptrón multicapa. Enfocando este método al ámbito de la localización, las RSSIs de los puntos de acceso serían las entradas y las probabilidades de estar en cada una de las localizaciones, serían las salidas.

A diferencia de los métodos anteriores, esta técnica requiere mucho tiempo de entrenamiento y una cantidad elevada de muestras de entrenamiento. Otro de los inconvenientes es que corre el riesgo de sobreajustarse (*overfitting*: efecto de sobreentrenar un algoritmo a partir de los datos de entrenamiento, quedando ajustado a unas características muy específicas de esos datos, de manera que, al introducir nuevos datos, arrojará errores considerables al comparar el resultado de la clasificación con el resultado esperado), prolongándose en el tiempo la planificación del número de pesos a ajustar y cuantos casos de prueba se van a usar.

2.4 Aprendizaje automático

El aprendizaje de máquina o aprendizaje automático (en inglés, *Machine Learning*) es un campo de las ciencias de la computación y rama de la inteligencia artificial cuyo fin es desarrollar técnicas que permitan a las computadoras aprender sin estar explícitamente programadas. A partir de bases de datos, pone en marcha algoritmos, que posibilitan a la computadora realizar correlaciones entre variables, clasificar grandes cantidades de datos y detectar diferencias y errores entre ellos. En resumen, trata de obtener análisis predictivos con fines precisos y establecer correlaciones entre varios sucesos.

El aprendizaje automático surge como resultado del desarrollo de la inteligencia artificial. Ya en los primeros días de la inteligencia artificial como disciplina académica, algunos investigadores se interesaron en hacer que las máquinas aprendiesen por sí solas. Trataron de resolver el problema con diversos métodos simbólicos, así como los que ellos llamaron 'redes neuronales' que eran en general perceptrones y otros modelos básicamente basados en técnicas estadísticas, como los modelos lineales generalizados.

Pero para ello es necesario definir antes el concepto de *aprendizaje*. Dicho concepto que parece cotidiano e intuitivo, a la hora de llevarlo a un punto de vista computacional acaba teniendo diversas definiciones relacionadas cada una con contextos muy concretos y posiblemente difieren de lo que intuitivamente conocemos como *aprendizaje*.

Una definición general e intuitiva de aprendizaje la encontramos en [15]: “proceso a través del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación”. De lo anterior podríamos destacar, a la hora de llevar la definición al campo computacional, el hecho de que el aprendizaje debe producirse a partir de la experiencia del entorno y no de las habilidades o conocimiento que sean innatos en el individuo o que se adquieran como resultado del crecimiento natural de éste. Por lo que podemos considerar aprendizaje a aquello que la máquina pueda aprender a partir de la experiencia y no gracias al reconocimiento de patrones anteriormente programados. Resumiendo, podemos decir que la manera en que la máquina aprende es por medio de objetos con los que entrenarse (ejemplos) para luego aplicar los patrones que haya reconocido sobre otros objetos distintos.

Podemos decir que es un proceso de inducción del conocimiento, ya que es un método que nos permite obtener un enunciado general a partir de enunciados que describen casos particulares. Si se han observado todos los casos particulares, la inducción se considera completa y la generalización se da por válida. Sin embargo, es imposible obtener una inducción completa en la mayoría de los casos, quedando el enunciado sometido a cierto grado de incertidumbre de tal forma que no se puede justificar empíricamente ni considerar como un esquema de inferencia formalmente válido.

Son numerosos los casos en los que el campo de actuación del aprendizaje de máquina se solapa con el de *data mining* o *estadística computacional*, debido a que ambas disciplinas están enfocadas al análisis de datos. No obstante, el aprendizaje de máquina está más centrado en el estudio de la complejidad computacional de los problemas con el propósito de hacerlos realizables desde una perspectiva práctica y no únicamente teórica. Gran parte de la investigación realizada en aprendizaje

de máquina se centra en diseñar soluciones factibles a problemas de la clase *NP-hard*. El aprendizaje de máquina se puede ver como un intento de automatizar algunas partes del método científico mediante métodos computacionales.

2.4.1 Tipos de tareas

Dentro de lo que llamamos aprendizaje automático hay una gran cantidad de problemas cuya diferencia principal recae en el tipo de objetos que intentan predecir. Algunas de las tareas habituales que intentan resolver estos problemas son:

2.4.1.1 Regresión

Su objetivo es predecir un valor real. A un nivel muy básico, consiste en intentar extraer conocimiento sobre algunas propiedades no observadas de un objeto basándose en las propiedades que sí han sido observadas de ese mismo objeto (o en otros objetos similares). Por ejemplo, predecir las ventas del mes que viene a partir del comportamiento de las ventas que está almacenado (pasado), predecir la nota de corte de unas oposiciones basándose en las notas obtenidas en la prueba de selectividad ese año o predecir si un determinado producto le va a gustar a un cliente basándonos en las valoraciones que ese mismo cliente ha dado a otros productos que sí ha probado.

2.4.1.2 Clasificación (binaria o multiclase)

Tiene como objetivo predecir la clasificación de objetos sobre un conjunto de clases prefijadas. Por ejemplo, clasificar una determinada noticia en la categoría de entretenimiento, deportes, política, etc. La clasificación sería binaria si solo se permiten dos posibles clases y multiclase si se permiten más de dos como se puede observar en el segundo modelo de la Figura 2-10.

2.4.1.3 Ranking

Su objetivo es predecir el orden óptimo de un conjunto de objetos según un orden de relevancia predefinido. Por ejemplo, tras la búsqueda realizada por un usuario, el buscador intenta seguir un orden óptimo a la hora de devolver los recursos.

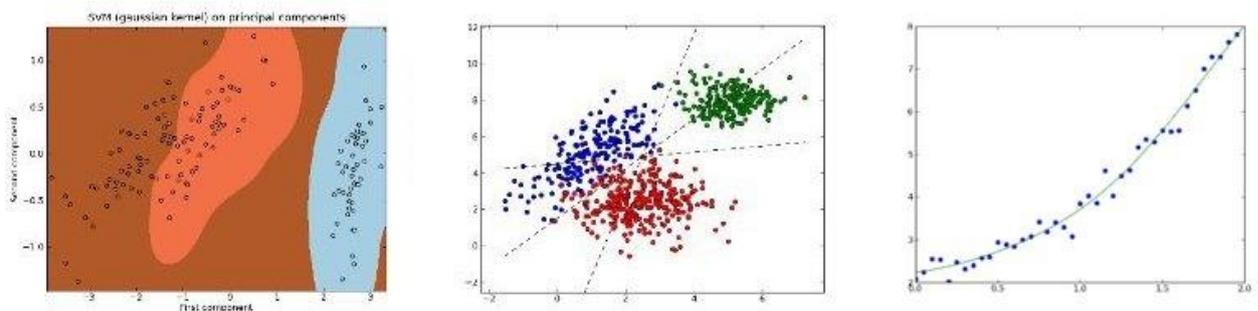


Figura 2-10 Ejemplos representativos de clases de problemas que resuelve el *machine learning*. Fuente: [15]

Cuando se afronta un nuevo problema de aprendizaje de máquina, habitualmente lo primero que se hace es encuadrarlo en alguna de las clases anteriores. De esta forma, dependiendo de cómo se clasifique, podremos medir el error cometido entre la predicción y la realidad; aunque podríamos decir que, generalmente, necesitaremos reducir la representación del problema a un espacio en el que tengamos definida una medida.

2.4.1.4 Agrupamiento (*clustering*)

Es un método de aprendizaje no supervisado, muy utilizado en el análisis estadístico de datos, que tiene como objetivo agrupar y clasificar una serie de observaciones o vectores en subgrupos de acuerdo con un criterio. Estos criterios se guían generalmente por una similitud específica y una similitud entre los miembros de un grupo; y la separación entre los diferentes grupos.

2.4.2 Tipos de algoritmos

Como se mencionó en la definición de aprendizaje de máquina, se ponen en marcha algoritmos a partir de bases de datos. Dependiendo del tipo de salida que se produzca y de cómo se aborde el tratamiento de los ejemplos, los algoritmos de aprendizaje de máquina se agrupan tal como se indica en las siguientes secciones:

2.4.2.1 Aprendizaje supervisado

Se genera una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Este tipo de técnicas de aprendizaje se pueden utilizar, por ejemplo, en el problema de clasificación, en el cual el aprendizaje de máquina trata de etiquetar (clasificar) una serie de instancias utilizando una entre varias categorías (clases). La base del conocimiento del sistema está formada por ejemplos etiquetados a priori, de los cuales sabemos su clasificación correcta. Este tipo de aprendizaje resulta muy útil, por ejemplo, en problemas de investigación biológica.

2.4.2.2 Aprendizaje no supervisado

Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formados únicamente por entradas al sistema. No se conoce ninguna información sobre las categorías o clasificación correcta de esos ejemplos. Por lo tanto, en este caso, se busca que el sistema consiga reconocer patrones con el objetivo de poder etiquetar las nuevas entradas.

2.4.2.3 Aprendizaje semisupervisado

Combina los dos algoritmos anteriores. Tiene en cuenta ejemplos clasificados y no clasificados.

2.4.3 Modelos de clasificación

Entre los diferentes modelos de clasificación podemos destacar los siguientes: árboles de decisión, reglas de asociación, algoritmos genéticos, redes neuronales artificiales, máquinas de vector soporte y redes bayesianas. Algunos de estos modelos se han explicado en el apartado 2.3, dándose una breve descripción, enfocada a la medición del posicionamiento. Nuestro sistema de localización está basado en uno de los modelos nombrados, máquinas de vector soporte. Por este motivo, se dedica el apartado 2.5 exclusivamente a desarrollar este modelo.

2.4.4 Aplicaciones

En la actualidad, podemos encontrar una gran variedad de aplicaciones del aprendizaje automático como diagnósticos médicos [16], análisis del mercado de valores [17], reconocimiento del habla [18] y del lenguaje escrito, motores de búsqueda, detección de fraude en el uso de tarjetas de crédito [16], clasificación de secuencias de ADN, minería de datos, Big data, previsiones de series temporales, juegos y robótica, entre otros [19].

2.5 Máquinas de vector soporte

Las máquinas de vectores soporte (SVM, Support Vector Machine) son métodos de aprendizaje automático que están propiamente relacionados con problemas de clasificación y regresión. Estos métodos usan una serie de algoritmos de aprendizaje supervisado que buscan obtener el límite que separa diferentes clases con el mayor margen posible. Estos algoritmos fueron desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T [20].

SVM se apoya sobre un conjunto de ejemplos de entrenamiento (muestras) los cuales podemos clasificar en clases o categorías y entrenar a la máquina a través de los algoritmos de SVM para que al aportar una nueva muestra prediga la clase o categoría a la que pertenece. Para ello se debe hallar el límite, comentado anteriormente, que separe las muestras de entrenamiento de diferentes clases con el mayor margen posible como se muestra en el ejemplo de la Figura 2-11.

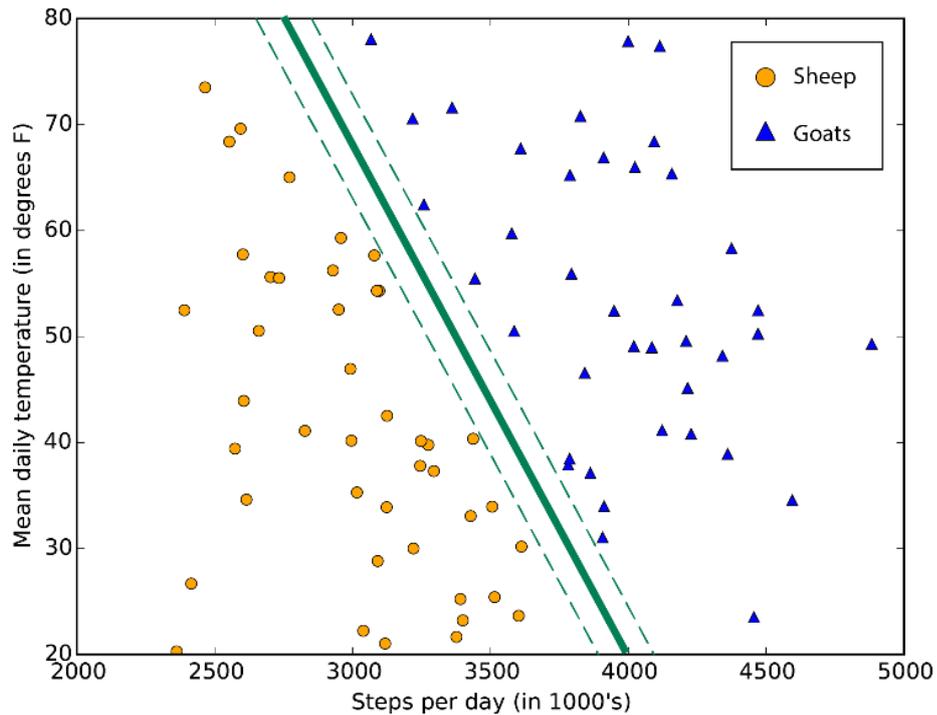


Figura 2-11 Ejemplo representación modelo SVM. Fuente: [21]

En la mayoría de los métodos de clasificación supervisada, como SVM, las muestras son vectores p -dimensionales, los cuales son representados como puntos en el espacio, cuyas coordenadas coinciden con las dimensiones del vector correspondiente. Como se indicó en el párrafo anterior, los puntos quedan separados en clases. Esta separación se realiza mediante un hiperplano de separación, definido como vector soporte, entre los dos puntos más cercanos de diferente clase. Se busca la mayor distancia posible entre estos dos puntos, es por ello por lo que a las SVM se les conoce también como *clasificadores de margen máximo*.

Una vez que tenemos nuestro modelo SVM con las correspondientes divisiones en clases, cuando el sistema introduzca un dato o vector nuevo, éste podrá ser clasificado directamente en una u otra clase dependiendo del lado del hiperplano en el que se encuentre.

Es posible encontrar los datos de entrada clasificados en dos categorías fácilmente separables por una recta, pero, si esto no es posible, como es el caso de la Figura 2-12, las muestras pueden ser previamente proyectadas a un espacio de dimensionalidad superior. Este tipo de proyecciones dependen del algoritmo o función que utilizemos. Nuestro objetivo es encontrar el algoritmo o función óptima, que mejor separe nuestras clases.

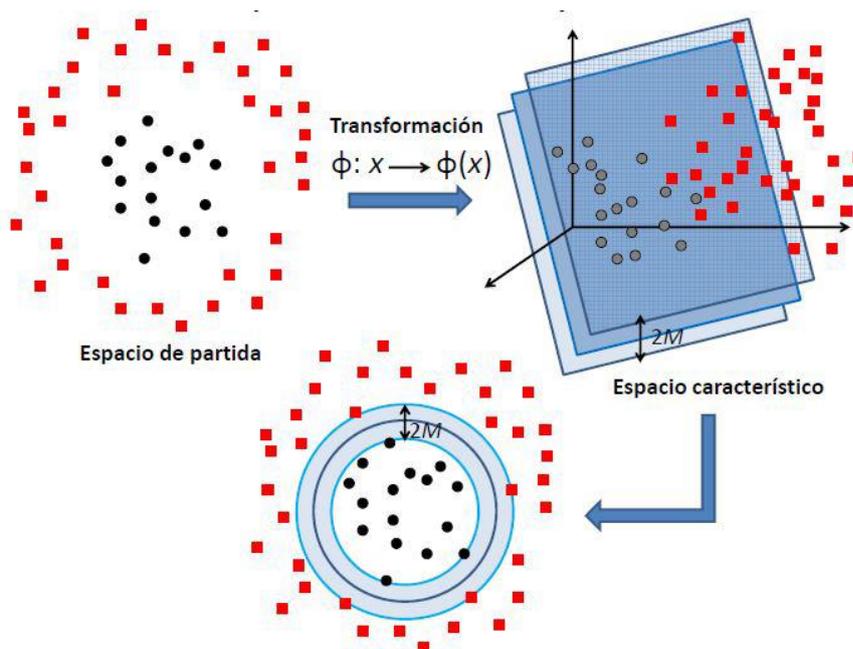


Figura 2-12 Ejemplo extensión a reglas no lineales. Fuente: [22]

Entre los conceptos básicos de SVM, utilizamos *atributo* para nombrar a la variable predictora y *característica* para nombrar a un atributo transformado que es usado para definir el hiperplano. La *selección de características* es un proceso de elección de la representación más adecuada del universo estudiado. *Vector soporte*, como se nombró anteriormente, es el vector obtenido a partir de los puntos más cercanos al hiperplano.

Los algoritmos SVM pertenecen a la familia de los clasificadores lineales. Además, sus modelos están estrechamente relacionados con las redes neuronales. En el caso de usar una función Kernel (que transforman a espacios de dimensionalidad muy superior), resultan un método de entrenamiento alternativo para clasificadores polinomiales, funciones de base radial y perceptrón multicapa.

2.5.1 Función Kernel

En los modelos de SVM la manera más sencilla de separar las categorías es mediante una línea recta, un plano recto o un hiperplano N-dimensional. Sin embargo, no siempre se presentan los casos ideales para que se pueda separar de esta manera, ya que se pueden dar más de dos variables predictoras; clasificación en más de dos categorías; casos donde los conjuntos de datos no pueden ser completamente separados; o son necesarias curvas no lineales de separación.

Estos casos dificultan la resolución del aprendizaje de máquina lineal. La representación mediante funciones Kernel presenta una solución a este problema. Gracias a estas funciones podemos proyectar la información a un espacio de características de mayor dimensionalidad, el cual consigue mejorar la capacidad computacional del aprendizaje de máquina lineal.

Existen distintos tipos de funciones Kernel, entre las que podemos destacar: lineal, polinomial, cuadrática, radial Gaussiana, Sigmoide y perceptrón. A continuación, se muestran algunos ejemplos:

- Polinomial-homogénea: $K(x_i, x_j) = (x_i * x_j)^n$. En la Figura 2-13 se muestra un ejemplo gráfico de esta función.

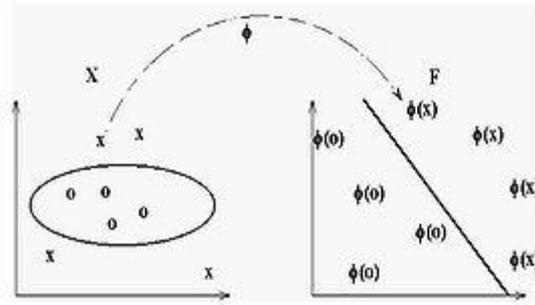


Figura 2-13 Polinomial. Fuente: [23]

- Perceptrón: $K(x_i, x_j) = \|x_i - x_j\|$. La Figura 2-14 muestra un ejemplo de la función perceptrón.

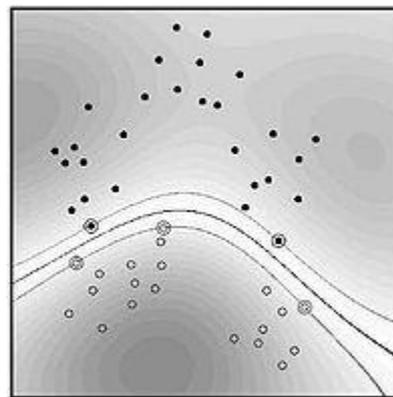


Figura 2-14 Perceptrón. Fuente: [23]

- Función de base radial Gaussiana: La separación se realiza a través de un hiperplano en el espacio transformado: $K(x_i, x_j) = e^{-(x_i - x_j)^2 / 2 * \sigma^2}$. En la Figura 2-15 se observa un ejemplo de esta función variando el valor de sigma (A, sigma = 20; B, sigma = 1; C, sigma = 0,05).

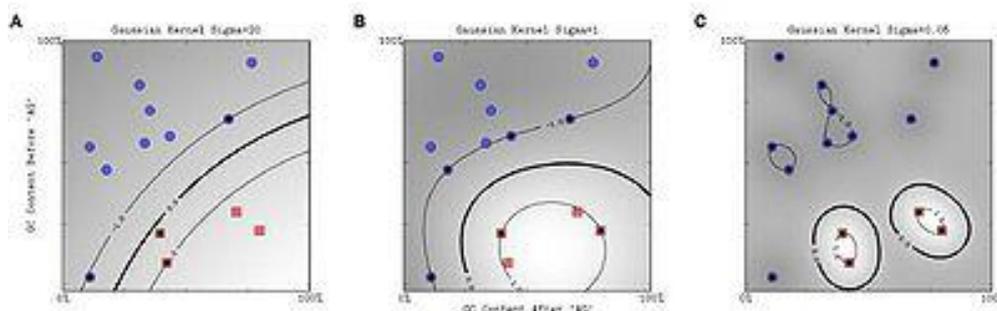


Figura 2-15 Función de base radial Gaussiana. Fuente [23]

SVM ha sido implementado en diversos campos. Entre las diversas aplicaciones encontramos: fusión de información acústica e idiolectal para tareas de reconocimiento de locutor en habla conversacional, sistemas de reconocimiento facial para control de acceso automático, modelos de recuperación de información, clasificación de imágenes biomédicas aplicadas a quemaduras [24].

2.6 Actividad empresarial y académica en el área de los sistemas de localización en interiores

Actualmente, los sistemas de localización en interiores están en pleno auge. Con el desarrollo de estos sistemas, van apareciendo nuevas aplicaciones y cada vez son más las empresas en el mundo que empiezan a utilizar esta tecnología. Entre la gran variedad de aplicaciones, a continuación, se citan algunos ejemplos:

- La localización exacta en tiempo real (*RTLS, Real Time Location System*) de personal, carretillas y otros elementos en almacenes [25]. Algunos de estos sistemas pueden llegar a complementar otros, como los sistemas AGV (*Automated Guided Vehicles*) de carretillas de palet autoguiadas.
- Navegación de drones a partir de la técnica SLAM y localización [26]. Estos drones integran un algoritmo avanzado de SLAM que les permite cartografiar en 3D y simultáneamente localizarse en el espacio. Este algoritmo se alimenta de toda la información de los sensores a bordo para calcular, en tiempo real, la posición en el espacio, sin GPS, y para cartografiar el entorno.
- Monitorización de una residencia de ancianos, discapacitados, invidentes o personas con problemas de movilidad [27]. Esta aplicación permite aumentar la independencia de estas personas y reducción del personal dedicado exclusivamente a vigilancia.
- Seguimiento de carritos en un supermercado. Este sistema trata de seguir las trayectorias de los clientes en un supermercado para el estudio de la disposición de productos, determinación de pautas de compra y estudios de mercado. Es un sistema anónimo, discreto y barato [2]. En la Figura 2-16 se muestra un ejemplo de la captura del seguimiento de los carritos de un supermercado.

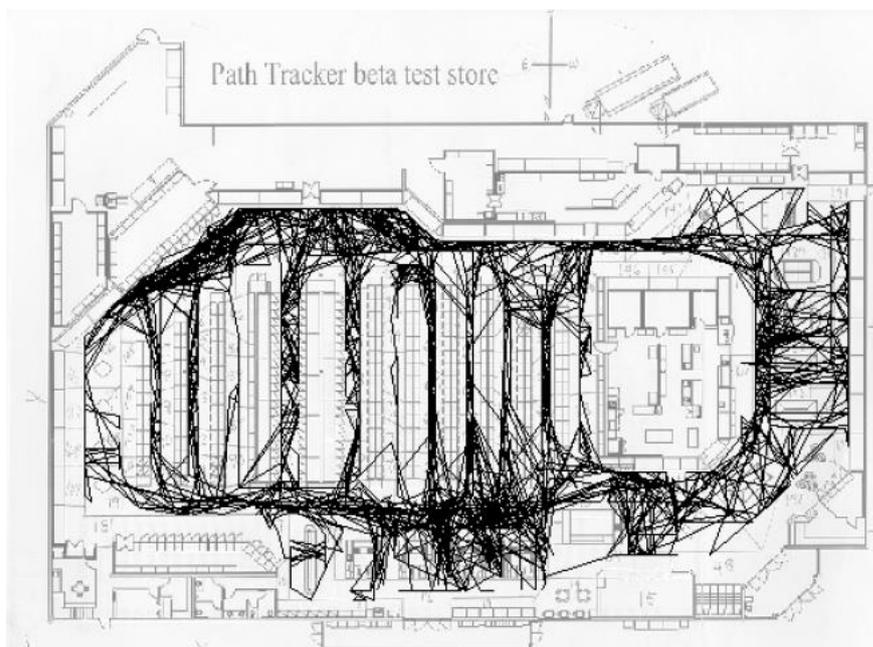


Figura 2-16 Obtención de rutas de carritos de un supermercado. Fuente [2]

Se presenta, a continuación, una revisión de los trabajos en este ámbito desarrollados en el entorno empresarial y académico.

2.6.1 Ekahau

La empresa Ekahau, Inc. [28] se dedica a la implantar redes WiFi que permiten incorporar un sistema de localización. El sistema llamado *Ekahau Positioning Engine* fue de los primeros en el mundo comercial en presentar un método que permitía el uso íntegro de aplicaciones (en cuanto a la implantación de balizas), necesitando únicamente la RSSI de los puntos de acceso del entorno. El personal o los recursos que se quieren localizar debían llevar unas etiquetas WiFi como las que se muestran en la Figura 2-17.



Figura 2-17 Etiquetas WiFi del sistema Ekahau. Fuente [3]

Más adelante, Ekahau, Inc. introduce nuevos sistemas con algunas mejoras, sobre todo a nivel de administración de la información proporcionada, como es el sistema de localización en tiempo real Ekahau (*Ekahau Real-Time Location System*) que monitoriza e informa en tiempo real de las ubicaciones de los recursos rastreados. Al igual que el anterior, este sistema no requiere el despliegue de antenas de radiofrecuencia o receptores costosos, sino que aprovecha los puntos de acceso de WiFi estándar como sensores. Todo esto se traduce en una implementación más rápida y una mejor relación coste-efectividad [29].

2.6.2 AeroScout

La tecnología AeroScout [30] proporciona una ubicación en tiempo real para los recursos y el personal, con el objetivo de mejorar la seguridad, agilizar las operaciones y aumentar la productividad. Ha sido probada en la industria y ha sido puesta en funcionamiento en diferentes países. Este sistema aprovecha las tecnologías WiFi y RFID implementadas con técnicas de nivel de señal recibida (RSSI) y de localización de tiempo de llegada (*ToA*) [3].

En el año 2005, Extronics y AeroScout se unieron como socios estratégicos para desarrollar productos con aplicaciones en áreas peligrosas, así como también, implementaciones inalámbricas certificadas en áreas peligrosas [31].

El uso de más de una tecnología permite a este sistema tener una mayor aplicabilidad que los sistemas de localización tradicionales [3].

2.6.3 PanGo Networks

PanGo Networks, Inc. [32] fue fundada en 1999 con sede en Framingham, Massachusetts y fue adquirida por InnerWireless, Inc. en marzo de 2007.

Esta corporación ofrece una plataforma de administración de ubicación inalámbrica y aplicaciones de seguimiento de recursos para los mercados empresariales (*PanOS Platform 4.0*). Esta plataforma de gestión de ubicación permite habilitar, administrar e integrar la ubicación y la información relacionada con la movilidad de los dispositivos a través del dispositivo *PanGo Locator* y etiquetas *PanGo Active RFID* [32].

PanGo Locator es una solución de seguimiento con identificación por radiofrecuencia, RFID, basada en WiFi y UWB que proporciona visibilidad de recursos empresariales. Las etiquetas *PanGo*

Active RFID permiten a los usuarios rastrear recursos que no están habilitados para conexión inalámbrica.

La compañía ofrece soluciones para industrias, incluidas la sanidad, la cadena de suministro, la empresa el comercio minorista y el gobierno [32].

El gran logro de este sistema de localización es que emplea varias capas sensoras (WiFi, UWB, RFID pasivo). El empleo simultáneo de diferentes sensores permite integrar sistemas con diferentes distribuciones de error, por lo tanto, se consigue reducir el error cometido en las estimaciones e incrementar la tasa de acierto [3].

2.6.4 Telefónica I+D. Localización en entornos WiFi

Este proyecto persigue los mismos principios y objetivos en los que se basan otros sistemas como los de *Ekahau*, es decir, a partir de la RSSI de puntos de acceso WiFi obtenemos la localización de un recurso o personal.

Para ello, este proyecto, desarrollado por el programa de I+D de Telefónica [33], utiliza el algoritmo de Monte Carlo. Este algoritmo tiene en cuenta otros aspectos, a parte de la potencia recibida, como la información que proviene de la arquitectura del edificio y el perfil del usuario. Además, es implementado con el algoritmo recursivo SIR (*Sampling Importance Resampling*) que tiene en cuenta datos dinámicos relativos al movimiento del usuario [3].

Toda la información se apoya en un plano del edificio donde se quiera obtener el entorno de localización interior, en formato BMP. La calibración inicial del sistema es experimental y se realiza midiendo la RSSI de cada punto de acceso en *células* divisorias de un metro cuadrado. Estas mediciones son evaluadas posteriormente por Matlab para comprobar que se puede distinguir entre las diferentes células. Si no se distingue bien, se deben mover algunos puntos de acceso WiFi y se vuelven a realizar mediciones de la RSSI [3].

Uno de los inconvenientes de este sistema es que debe realizarse la calibración de nuevo cada vez que se introduce un cambio en la planta y el proceso de calibración es complejo y requiere un tiempo considerable.

2.6.5 Sistemas de localización de la Universidad de Trento basado en redes neuronales

En este sistema [34] se opta por no hacer una modelización de la señal, es decir, no es necesario que conozca el entorno donde se va a poner a funcionar el sistema, ni la localización de los puntos de acceso. Además, no se precisa el uso hardware adicional, solo se necesita obtener la RSSI de los puntos de acceso.

El proceso de calibración es experimental. Se toman diferentes medidas de RSSI a lo largo del día para cubrir diferentes momentos donde la actividad y la contaminación por radiofrecuencia son diferentes.

El grupo investigador probó este sistema en una planta, de 625m², de la misma universidad. Se colocaron tres puntos de acceso completamente iguales en los extremos de la planta, facilitando la separación de las muestras recibidas. Además, las tarjetas de red WiFi, utilizadas para la calibración, eran del mismo modelo y fabricante.

Se empleó un perceptrón multicapa con tres entradas (los tres puntos de acceso) y dos salidas (la posición en dos dimensiones). Se probó a emplear diferentes capas intermedias de 4, 8 ó 16 neuronas y se llegó a la conclusión de que el empleo de muchas neuronas, 16 en su caso, era contraproducente porque se produce enseguida *overfitting*. La diferencia entre usar 4 y 8 era prácticamente imperceptible.

La localización se mostró en términos de coordenadas cartesianas, expresadas en metros. El error final que se calculó, tras la puesta en marcha de este sistema, fue de 1,5 m. Dicho error, aparentemente,

no es muy elevado, pero teniendo en cuenta que las condiciones eran muy favorables, no nos garantiza una buena funcionalidad en caso de implementarlo en lugares donde no nos hayamos preocupado de colocar los puntos de acceso de manera óptima para nuestro sistema de localización, ni de utilizar puntos de accesos y receptores del mismo modelo y fabricante.

2.6.6 NAPT system

El sistema *Naval Automated Personnel Tracking* (NAPT) [35] fue desarrollado por la compañía australiana *Blue Glue Pty Ltd*, con el apoyo del programa CTD (*Capability and Technology Demonstrator*). El DSTO (*Defence Science and Technology Group*) y la Armada Americana realizaron una demostración a bordo del buque de apoyo anfibio fuera de servicio HMAS Kanimbla.

Esta tecnología permite saber qué personas están a bordo en un momento dado y su ubicación en el buque a través del uso de etiquetas transmisoras de radio portátiles. Cada 1,5 segundos, la etiqueta transmite un paquete de datos que identifica al usuario de la etiqueta. Los receptores de radio instalados a lo largo de todo el buque detectan esta señal y en función de la intensidad de señal que se reciba, el sistema puede determinar donde se sitúa esta persona a bordo. Las etiquetas están programadas para no transmitir en el mismo momento e interferirse entre sí. El sistema también utiliza rayos láser e infrarrojos en dispositivos para alertar de accesos a zonas de peligro.

En el transcurso de los ensayos realizados en el HMAS Kanimbla se demostró que la tecnología NAPT funcionaba correctamente, cumpliendo con todas las medidas de rendimiento puestas como objetivo por CTD [36].

2.6.7 Sistema de Localización de Personal desarrollado por FABA

FABA (*Fábrica de Artillería de Bazán*), actualmente denominada como Navantia Sistemas [37], es una empresa situada en San Fernando que construye y desarrolla sistemas de combate y control de plataforma para barcos militares y sistemas de control para barcos civiles. Entre sus proyectos, FABA ha desarrollado e implantado en los Buques de Acción Marítima (BAM) un sistema de localización de personal a partir de tecnología RFID.

Este sistema permite controlar el acceso por personal embarcado a ciertas zonas del buque, donde se han colocado lectores de tarjetas RFID activas, como zonas de acceso, cubiertas y locales, enviando la información recibida a *switches* que centralizan y distribuyen las señales a las subestaciones del sistema de localización más cercanas. Los miembros de la dotación del buque deben portar sus tarjetas con identificador único [1]. En la Figura 2-18 se muestra un esquema de la infraestructura de este sistema.

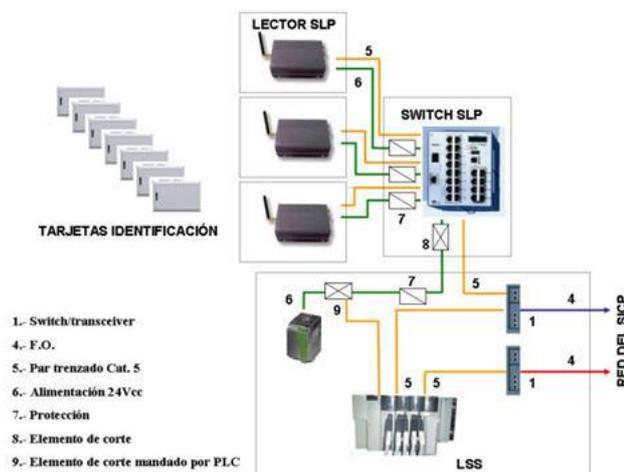


Figura 2-18 Distribución del sistema de localización de personal de los BAM. Fuente: [1]

El Sistema Integrado de Control de Plataforma (SICP), también desarrollado por FABA, integra el sistema de localización del personal. Desde este sistema podemos visualizar y controlar el acceso a zonas restringidas, el control de embarco, desembarco y la localización de personal en tiempo real por compartimentos o zonas. Entre los principales inconvenientes de este sistema encontramos la gran cantidad de falsas alarmas en el sistema SICP, la necesidad de reconfiguración de los lectores con cambios en la estructura del buque o la poca autonomía de las tarjetas RFID y la imposibilidad de intercambiar baterías, una vez agotadas, debido a su estructura estanca [1].

2.6.8 *Real-time Location on Board Ships (Omnisense)*

Omnisense [38] es una empresa norteamericana especializada en soluciones de monitorización remota en tiempo real. Ha desarrollado una tecnología de seguimiento que puede ubicar la situación del personal a bordo de un buque durante emergencias. Permite que procedimientos como la evacuación de personal en compartimentos donde se active el sistema de CO2 se realicen de manera efectiva en tiempo real y se tenga en cuenta todo el personal rastreado, mejorando tanto la calidad como la velocidad de la toma de decisiones [39].

En una emergencia en la mar, proporciona al Oficial de Seguridad, al Supervisor Operativo de Equipos y Sistemas (SOES) y demás personal encargado de la seguridad del buque, la garantía de que las normas de seguridad se están aplicando correctamente y una visión general de la localización del personal de seguridad interior y del resto del buque.

El sistema integra funciones avanzadas que permiten al personal que asiste, dirigirse instantáneamente a la ubicación correcta cuando el personal que está en peligro pulsa el botón de pánico o se activa una alerta automática de actividad anormal (por ejemplo, hombres al agua).

Se ha instalado con éxito en salas de máquinas con mamparos de metal, fábricas, almacenes, oficinas y túneles.

2.6.9 *On Board Personnel Tracking System (Martec)*

Martec (*Marine&Technologies*) [40] es un grupo empresarial privado italiana que opera en soluciones de defensa e integración de seguridad para la Armada italiana, cruceros y Mega Yates desde 1994.

Entre sus productos desarrollados, Martec cuenta con un sistema de localización de personal a bordo llamado *On Board Personnel Tracking System (OPTS)*. Este sistema está basado en tecnologías RFID activas, para monitorizar al personal en tiempo real dentro de zonas peligrosas del buque. Todo el personal a bordo tiene una etiqueta RFID personal que se detecta automáticamente por las antenas ubicadas en zonas vigiladas del barco [40].

OPTS tiene una interfaz directa con otros sistemas desarrollados por la propia empresa que permite proporcionar información e identidad del personal que porta la etiqueta RFID y su localización a bordo.

2.6.10 *iBeacon (Apple)*

iBeacon [41] es un proyecto desarrollado por *Apple Inc.* basado en un sistema de posicionamiento en interiores que utiliza tecnología BLE que se puede encontrar en dispositivos Bluetooth 4.0 que soporten modo dual. Los dispositivos con tecnologías *iBeacon* funcionan como balizas, las cuales permiten establecer una región alrededor, para que los dispositivos *iOS* puedan determinar cuándo han ingresado o abandonado esa región, junto con la estimación de la proximidad a la baliza. Es posible el envío de notificaciones *push* entre dispositivos.

Las balizas *iBeacon* trabajan como un sensor inalámbrico de posición/contexto en un entorno inteligente, como podría ser un almacén de un centro comercial o un supermercado, permitiendo obtener la localización del dispositivo. Utilizan *Bluetooth Low Energy Proximity Sensing* para

transmitir un identificador único universal, capturado por una aplicación o sistema operativo compatible y que puede ser transformado en una localización física o hacer que el dispositivo genere una acción de interés para el usuario (ofertas de productos, guía dentro de la tienda, etc.) [1].

Apple trabaja con otras plataformas, como *Estimote*, para optimizar el trabajo en el desarrollo de aplicaciones y hardware para el usuario de iBeacon.

2.6.11 Eddystone (Google)

Esta tecnología fue presentada por Google en 2015. Se trata de un nuevo formato de baliza Bluetooth Low Energy, disponible bajo código abierto. Es un formato independiente de la plataforma y al igual que *iBeacon*, es compatible con dispositivos que dispongan de BLE con Bluetooth 4.0 que soporten modo dual. Eddystone puede ser detectado tanto por dispositivos Android como iOS. Asimismo, permite al usuario desarrollar aplicaciones para el móvil que trabajen con estas balizas para ofrecer contenido en base a la ubicación de los teléfonos inteligentes de los usuarios. También nos ofrece una API de cercanía y una de proximidad [42].

Eddystone aparece como un rival directo de la tecnología iBeacon lanzada por Apple, que tratamos en el apartado anterior. Esta tecnología cuenta con la ventaja de que el 80% de los teléfonos inteligentes utilizan el sistema operativo de Google, Android. Además, Eddystone puede ofrecer una estrecha integración con aplicaciones propias como Google Chrome y Maps, entre otras [42].

Google trabaja con proveedores de toda la industria de balizas para fomentar la interoperabilidad a través de la plataforma Google Beacons en Android, iOS y Google Cloud Platform. Existen dos tipos de categorías de proveedores que proporcionan asistencia con la instalación, configuración y mantenimiento de las balizas. Por un lado, están los proveedores de servicios de ubicación que ofrecen servicios en la instalación de software basados en localización de interiores. Algunos ejemplos de estos proveedores se muestran en la Figura 2-19.



Figura 2-19 Proveedores de servicios de ubicación que desarrollan Eddystone. Fuente: [43]

Por otro lado, los fabricantes de Beacons ofrecen hardware de beacons que constituye la base de la funcionalidad relacionada con la ubicación y proximidad. Los fabricantes más importantes se muestran en la Figura 2-20.



Figura 2-20 Fabricantes de Beacons que desarrollan Eddystone. Fuente: [43]

En el siguiente subapartado, 2.6.12, se expondrán los recursos que ofrece uno de los fabricantes más importantes de Estimote, que trabaja tanto con Eddystone como con iBeacon, para reparar en las posibilidades que nos ofrece trabajar con una de estas plataformas, demostrando que sería útil trabajar con ellas en el caso de querer desarrollar un proyecto como el que se plantea en este trabajo.

2.6.12 Estimote

Estimote proporciona kits de desarrollo de software (SDK, *Software Development Kits*) completos para que el usuario pueda comenzar a crear aplicaciones compatibles con los dispositivos desarrollados por Estimote. Estos kits, también van equipados con herramientas de software e interfaces de programación de aplicaciones (API, *Application Programming Interface*) que permiten al usuario distribuir sus aplicaciones de forma segura y escalable [44].

Estimote Proximity SDK es la plataforma que permite la creación de software basado en proximidad a partir de diferentes modelos de balizas. No obstante, Estimote ha desarrollado sus propias balizas, *Estimote Proximity Beacons*, con los que es más sencilla de trabajar a la hora de crear aplicaciones con *Estimote Proximity SDK*. Estas balizas tienen una vida útil de aproximadamente dos años. Si se desea mayor duración de la batería, se pueden utilizar los *Location Beacons* para aplicaciones de proximidad [44].

Por otro lado, *Estimote Indoor Location SDK* es la plataforma basada en la localización de personas, búsqueda de rutas y seguimiento de asistencia, con la que es posible ubicar a sus usuarios de forma precisa, ver su posición en tiempo real en un plano, recopilar datos de asistencia o entregar instrucciones para encontrar una ruta. Para ello, Estimote ofrece los *Location Beacons*, que son otro modelo de balizas como las que podemos observar en la parte inferior izquierda de la Figura 2-21. También existe la posibilidad de utilizar los *Location Beacons* con un chip UWB (*Ultra Wideband System*) que permite mapear el entorno automáticamente.



Figura 2-21 Ejemplo aplicación Estimote Indoor Location SDK y dispositivos Location Beacons. Fuente: [44]

Estimote SDK con protocolo telemétrico es otra modalidad de Estimote que permite realizar un seguimiento de sus recursos a medida que se mueven de un lugar a otro y recopilar datos adicionales del sensor, como su temperatura o vibraciones. Para ello existen etiquetas Estimote que están optimizadas para objetos más pequeños y duran de seis a doce meses. Para rastrear objetos más grandes o cuando se necesite mayor duración de la batería, las *Proximity Beacons* son compatibles.

Por último, *Estimote Mirror SDK* ofrece señalización digital personalizada y pantallas digitales interactivas. Permite, además, escanear y buscar señales de los teléfonos de las personas y mostrar el contenido contextual en pantalla.

Una de las ventajas de Estimote es la variedad de Beacons que presenta al mercado disponibles bajo código abierto, que pueden ser muy útiles a la hora de desarrollar un proyecto de localización en interiores. A continuación, se resumen las principales diferencias entre las balizas que ofrece Estimote. Todas ellas aprovechan el mismo firmware inteligente, pero su duración, forma o detalles difieren según su aplicación.

	<i>Location UWB Beacon</i>	<i>Location Beacon</i>	<i>Proximity Beacon</i>	<i>Sticker Beacon</i>	<i>Video Beacon</i>
					
Duración de la batería	5 años	5 años	2 años	1 año	Sin batería
Alcance	200 m	200 m	70 m	7 m	10 m
Espesor	27 mm	24 mm	17 mm	6 mm	14 mm
Paquetes iBeacon™ o Eddystone™	8 al mismo tiempo	8 al mismo tiempo	1 al mismo tiempo	1 al mismo tiempo	2 al mismo tiempo
Paquetes adicionales	Conectividad, telemetría, usuario definido	Conectividad, telemetría, usuario definido	Conectividad, telemetría	Conectividad, telemetría	Conectividad, telemetría, usuario definido
Sensores incorporados	Movimiento, temperatura, luz ambiental, presión	Movimiento, temperatura, luz ambiental, magnetómetro, presión	Movimiento, temperatura	Movimiento, temperatura	N/A
Modo de ubicación interior	Automático	Mapeo manual	N/A	Rastreo de etiquetas	N/A
Tecnología adicional	Redes en malla, GPIO, RTC, LED RGB, EEPROM de 1 Mb, NFC programable	Redes en malla, GPIO, RTC, LED RGB, EEPROM de 1 Mb, NFC programable	NFC programable	N/A	WiFi, HDMI, USB, 1GB de eMMC

Tabla 2-2 Características Beacons Estimote. Fuente: [44]

2.6.13 Situm

Situm [45] es una empresa española que ha sido fundada por tres doctores en Ingeniería y que ha desarrollado una tecnología de localización y navegación, resultado de siete años de investigación. A diferencia de las tecnologías vistas hasta ahora, que se centran prácticamente a los dispositivos BLE, Situm aprovecha todos los sensores que lleva cualquier móvil inteligente actual, vía MSDF (*Multi-sensor Smart Data Fusion*): GPS, WiFi, BLE, detectores inerciales (acelerómetro, giróscopo), detectores magnéticos, etc. Situm se sitúa entre las plataformas de localización en interiores del mundo con mayor precisión (0,5-3m) [46], menor tiempo de despliegue y bajo coste (apenas requiere infraestructura adicional). Se considera una de las cinco *startups* más prometedoras para aplicaciones móviles en España según *South Summit* y es la empresa española líder y más premiada en localización en interiores.

Situm tiene su tecnología instalada en más de 1600 edificios en todo el mundo, entre los que se encuentran 25 hospitales y centros médicos en Galicia, centros comerciales y sedes corporativas. El Hospital Universitario Lucus Augusti es uno de los 25 Hospitales donde acaban de desplegar su

tecnología para guiar a pacientes y visitantes. Este edificio cuenta con más de 160.000 m² de superficie y para calibrar todo el complejo tan solo se necesitaron 48 horas y una semana para tener listos los 25 hospitales y centros médicos [46].

Para entender el gran interés que despierta la tecnología Situm, podemos destacar que la empresa está cerrando acuerdos para trabajar en proyectos que abarcan, por ejemplo, los Juegos Olímpicos 2020 de Tokyo, fabricantes mundiales de *smartglasses*, plantas de fabricación de automoción, aeropuertos en Latinoamérica y centros comerciales en Centroamérica y Estados Unidos [46].

En el ámbito de la seguridad, Situm desarrolla métodos de posicionamiento en interiores, identificación y monitorización del personal de seguridad, incluyendo otras aplicaciones tales como: alerta de *hombre caído*, botón de pánico y asignación de tareas geolocalizadas. La Figura 2-22 muestra un ejemplo del uso de Situm en compañías de seguridad [45].



Figura 2-22 Ejemplo de aplicaciones Situm en compañías de seguridad. Fuente: [45]

La tecnología Situm desplegada en edificios públicos multitudinarios como aeropuertos y estaciones de tren o autobús permite guiar al usuario desde su posición hasta los principales puntos de interés y provee de información valiosa que puede ayudar a los gestores de los edificios a monitorizar servicios clave como la localización de pasajeros con movilidad reducida e incrementa la explotación de sus áreas comerciales [45].

En el área industrial, Situm proporciona datos en tiempo real de la interacción de activos móviles con otros elementos (carretillas, AGVs y otros vehículos logísticos).

Además, provee a grandes infraestructuras dedicadas a eventos masivos como encuentros deportivos, espectáculos o congresos, como la Fira de Barcelona [45].

3 DESARROLLO DEL TFG: SISTEMA DE LOCALIZACIÓN

En este capítulo se describirán los recursos empleados en el TFG, se explicarán los pasos a seguir para la configuración de las RPi en modo iBeacon y se expondrán todos los experimentos realizados en el Cuartel “Marqués de la Victoria” y en los seminarios de CUD, antes de pasar a validar y probar el sistema en un escenario más acorde con nuestros objetivos, el patrullero “Tabarca”.

3.1 Recursos empleados

3.1.1 Hardware

El hardware principal empleado en este trabajo es el computador de placa reducida Raspberry Pi [47], desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de las ciencias de la computación. Para este TFG se dispuso al principio de tres Raspberry Pi 3 modelo B y, posteriormente, se adquirieron dos más de la versión anterior, Raspberry Pi 2 modelo B.

Para controlar y detallar de forma clara el manejo de las diferentes RPi, se decidió asignarle un número a cada una para nombrarlas y tenerlas diferenciadas en todo momento. La nomenclatura es la siguiente:

- RPi1, RPi2 y RPi3 tercera generación, modelo B.
- RPi4 y RPi5 segunda generación, modelo B.

Para los propósitos de este TFG, la principal diferencia entre una generación y otra es que la tercera generación viene con una antena WiFi y Bluetooth integrada y la segunda generación no, para lo cual fue necesario el uso de adaptadores USB.

A continuación, se detallan las especificaciones de los diferentes modelos utilizados y el resto del hardware necesario para el TFG.

- 1) 3 x Placas Raspberry Pi 3 modelo B [48]:
 - SoC: Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB)
 - CPU: 1,2 GHz 64-bit quad-core ARMv8
 - SDRAM: 1GB LPDDR2 (Compartidos con GPU)
 - GPU:
 - Dual Core VideoCore IV, Coprocesador Multimedia. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264/MPEG-4 AVC de alto perfil de decodificación.

- Capaz de 1 Gpixel / s, 1.5 Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
- Conexiones:
 - USB 4 x Conector USB 2.0
 - Conector micro USB para alimentación.
 - Salida de vídeo:
 - HDMI rev 1.3 y 1.4
 - RCA compuesto (PAL y NTSC)
 - Salida de audio:
 - Conector jack de 3,5 mm de salida de audio, HDMI
 - Conector GPIO:
 - 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira.
 - Proporcionar 27 pines GPIO.
 - Conector de cámara de 15 pines.
 - Pantalla de visualización: Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
 - Slot MicroSD para tarjetas
- Conectividad:
 - 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)
 - 10/1000 Ethernet (RJ-45) via hub USB.
- Consumo energético: 800 mA, (4 W)
- Sistemas operativos soportados:
 - GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM.
 - RISC OS.

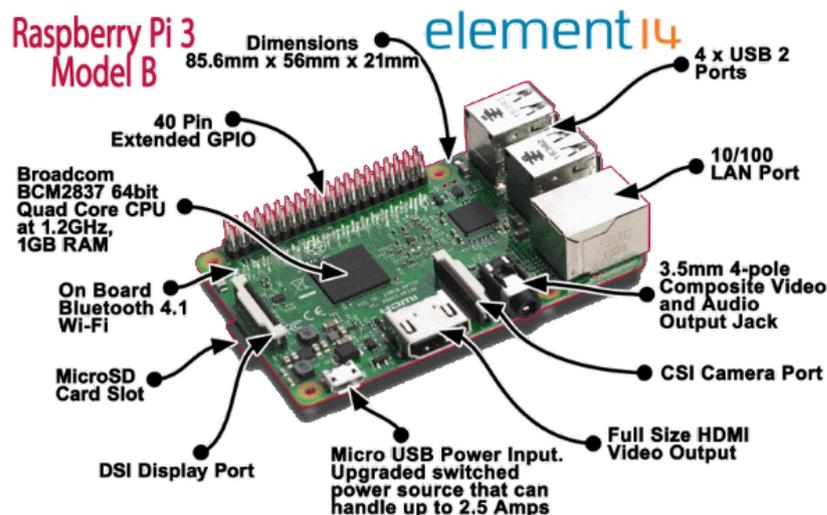


Figura 3-1 Componentes de la Raspberry Pi 3, Modelo B. Fuente: [48]

- 2) 2 x Placas Raspberry Pi 2 modelo B:
 - SoC: Broadcom BCM2836 (CPU + GPU + DSP + SDRAM + Puerto USB)
 - CPU: 900 MHz quad-core ARM Cortex A7
 - SDRAM: 1GB LPDDR2 (Compartidos con GPU)
 - GPU:
 - Dual Core VideoCore IV, Coprocesador Multimedia. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264/MPEG-4 AVC de alto perfil de decodificación.
 - Capaz de 1 Gpixel / s, 1.5 Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
 - Conexiones:
 - USB 4 x Conector USB 2.0
 - Conector micro USB para alimentación.
 - Salida de vídeo:
 - HDMI rev 1.3 y 1.4
 - RCA compuesto (PAL y NTSC)
 - Salida de audio:
 - Conector jack de 3,5 mm de salida de audio, HDMI
 - Conector GPIO:
 - 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira.
 - Proporcionar 27 pines GPIO.
 - Conector de cámara de 15 pines.
 - Pantalla de visualización: Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
 - Slot MicroSD para tarjetas
 - Conectividad de red:
 - 10/1000 Ethernet (RJ-45) via hub USB.
 - Consumo energético: 800 mA, (4 W)
 - Sistemas operativos soportados:
 - GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM.
 - RISC OS.
- 3) 5 x Tarjeta MicroSD de 16 GB
- 4) 2 x Adaptador nano USB Wireless 802.11.b/g/n
- 5) 2 x Adaptador USB Bluetooth CSR 4.0 serie BTA-CSR4B5.
- 6) 3 x Batería portátil externa
- 7) 5 x Cable de alimentación 5V 2000mA por micro USB
- 8) Monitor de ordenador con entrada HDMI
- 9) Teclado por cable USB
- 10) Ratón por cable USB
- 11) Cable HDMI
- 12) Ordenador portátil Asus para el manejo de las RPis por control remoto.

- 13) Samsung Galaxy S6 Edge para la utilización de aplicaciones Android compatibles con la tecnología BLE e iBeacon.

3.1.2 Software

- 1) Sistema operativo Raspbian Wheezy [47] para Raspberry Pi.
- 2) RealVNC [49]. Mediante esta aplicación se consigue tener control remoto de las RPi1, RPi2, RPi3 y RPi4. La ventaja de esta aplicación sobre Putty, es que nos permite acceder a la interfaz gráfica.
- 3) Putty [50] (Cliente SSH). Gracias a esta aplicación, se tiene control remoto de la RPi5, ya que la versión gratuita de la aplicación RealVNC no nos permitía integrar más de cuatro RPis.
- 4) Geany [51] (Editor de líneas de código de programación, que viene instalado por defecto en el sistema operativo Linux).
- 5) Notepad++ [52] (Editor de líneas de código de programación).
- 6) *BLE Scanner & Recorder* [53] (aplicación Android). Entre la gran variedad de aplicaciones Android, que sirven de escáner de señal BLE, ésta era de las pocas aplicaciones que permitía guardar los datos que se iban escaneando y la única que los volcaba en el formato deseado.
- 7) SVM^{light} [54] (Implementación de *Support Vector Machine* en C). El empleo de software se explicará con más detalle en el subapartado 3.3.6.1.

3.2 Configuración

Tras conocer los recursos utilizados en el trabajo, procedemos a explicar los pasos seguidos para la implementación del sistema, a partir de las RPis disponibles para este trabajo. Nuestro objetivo es configurar cuatro RPis (RPi2, RPi3, RPi4 y RPi5) para que actúen de radiobalizas y una RPi (RPi1) para que actúe en modo escáner.

3.2.1 Instalación del sistema operativo Raspbian y funcionamiento en modo BLE

Para poder operar con las RPis es necesaria la instalación de un SO (*Sistema Operativo*). Al no disponer de memoria no volátil interna propia, se instaló el sistema operativo en tarjetas MicroSD. Posteriormente se realizaron las configuraciones necesarias para poder operar con tecnología BLE.

En primer lugar, se formatearon las tarjetas MicroSD y se instaló Noobs [47] (versión 2.4.5), un instalador de sistemas operativos que ofrece la página oficial de Raspberry Pi [47] con el que se instaló Raspbian, el sistema operativo oficial de la Fundación Raspberry.

Una vez instalado el sistema operativo, se realizaron las configuraciones básicas, como cambio de usuario y contraseña; configuración de fecha y hora; permitir el acceso remoto a través de RealVNC y Putty. Después de esto, se conectaron las RPis a Internet mediante WiFi y se activó el Bluetooth para comprobar su funcionamiento. Para ello, hubo que conectar los adaptadores USB en las RPi4 y RPi5.

Para garantizar la comunicación de la placa RPi con el módulo Bluetooth y con el resto de los dispositivos conectados a la Raspberry Pi a través de los conectores USB, se procedió a la instalación de controladores de dispositivos USB. A continuación, se detallan las librerías y su utilidad:

- *libusb-dev*: Permite el acceso a los dispositivos que conectemos a los puertos USB.
- *libbus-1-dev*: Es un sistema de mensajes bus para comunicación entre aplicaciones.
- *libglib2.0-dev*: Con esta librería se implementan funciones del sistema.

- *libudev-dev*: Permite acceder y realizar consultas en la base de datos de los dispositivos conectados a los puertos USB.
- *libical-dev*: Con esta librería se implementa *iCalendar* para el intercambio de datos de calendario.
- *libreadline-dev*: Contiene un conjunto de funciones enfocadas al uso de aplicaciones que permiten editar líneas de comandos.

Tras la instalación de estas librerías, se instaló una pila de protocolo Bluetooth para Linux, BlueZ, que permite la comunicación entre dispositivos utilizando el estándar BLE. BlueZ es una pila de protocolos de código abierto de Bluetooth oficial de Linux, certificada por SIG (*Special Interest Group*). La versión instalada fue la 5.48, tomada del distribuidor *Linux Kernel Organization* [55].

Para la configuración inicial de las RPis fue preciso utilizar el monitor de ordenador, el teclado y el ratón. Tras la instalación del sistema operativo y su configuración, se descargó la aplicación RealVNC en las RPi1, RPi2, RPi3, RPi4 y en el portátil utilizado para el manejo de estas. Con RealVNC se pudo tener acceso remoto a la interfaz gráfica de las RPis. No se pudo utilizar dicho programa para conectar la RPi5, ya que la versión gratuita solo permitía conectar 5 dispositivos como máximo. Para controlar la RPi5 por control remoto se utilizó la aplicación Putty, con lo que solo se pudo tener acceso a su interfaz de línea de comandos.

3.2.2 Operación en modo radiobaliza iBeacon

El objetivo de esta etapa es la configuración del modo Bluetooth para la transmisión de datos de las RPi2, RPi3, RPi4 y RPi5 para operar en modo iBeacon y actuar de radiobalizas. Esta transmisión de datos será unidireccional, es decir, las balizas únicamente realizan transmisión de una trama de datos o *Payload Data Unit* (PDU), periódicamente, y el escáner solo recibe, sin establecerse conexión entre los dispositivos. Este tipo de comunicación es conocida con LEAd (*Low Energy Advertising*).

La PDU permite la localización e identificación de las balizas y su transmisión se realiza en tres canales de RF que BLE tiene predeterminado para LEAd, de los cuarenta en total de los que dispone. Los canales para la transmisión de PDU y sus frecuencias son los siguientes:

- Canal 37: 2401 MHz
- Canal 38: 2426 MHz
- Canal 39: 2480 MHz

La periodicidad de su transmisión es configurable entre los valores de 100 ms y 10,24 s (múltiplos de 0,625 ms). A la periodicidad seleccionada se le aplican retardos aleatorios de 10 ms en cada pulso con el objetivo de evitar interferencias entre radiobalizas transmitiendo a la vez en el mismo entorno.

El modo correcto de trabajo del módulo Bluetooth y sus controladores debe ser *UP RUNNING* y no *DOWN*. Para comprobar el estado del modo de trabajo, podemos ejecutar el comando *hciconfig* y para activar el modo *UP RUNNING* debemos ejecutar los comandos *sudo hciconfig hci0 up* y *sudo hciconfig hci0 noscan*.

Para la activación del modo LEAd se operó con los perfiles de BlueZ a partir de la interfaz de control de Hardware HCI. A continuación, se indican los tres comandos utilizados y su función dentro del modo LEAd:

- ***LE Set Advertising Data***: Con este comando ajustamos los datos de carga útil del PDU (*Protocol Data Unit*). El formato utilizado es el que podemos observar en la Figura 3-2, donde, <1E 02> es la cabecera, <01...15> es el *AD Address* y el *AD type*, <23 B0...3E 55>

es el UUID, <00 02> es *major*, el siguiente <00 02> es *minor* y <CB> es donde indicamos la potencia con la que va a transmitir la RPi.

```

pi@raspberrypigleznegro2:~ $ sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A
1A FF 4C 00 02 15 23 B0 AE 23 90 F9 4E 49 95 2A 10 8B E3 1E 3E 55 00 02 00 02
CB 00
< HCI Command: ogf 0x08, ocf 0x0008, plen 32
 1E 02 01 1A 1A FF 4C 00 02 15 23 B0 AE 23 90 F9 4E 49 95 2A
 10 8B E3 1E 3E 55 00 02 00 02 CB 00
> HCI Event: 0x0e plen 4
 01 08 20 00
    
```

Figura 3-2 Ejemplo de ejecución del comando *LE Set Advertising Data* en la RPi2

La cabecera, el *AD Address* y el *AD type* son siempre los mismos para la configuración de las RPi. El UUID se genera ejecutando el comando *sudo uuidgen*. Al *major* y *minor* le hemos asignado el mismo número que le corresponde a la RPi que va a transmitir, en el caso de la Figura 3-2, al ser la RPi2, se le asigna el valor <00 02>. Para el valor de la potencia transmitida, ponemos siempre el predeterminado, que corresponde con el valor real de transmisión, el cuál no se puede modificar.

- **LE Set Advertising Parameters:** Aquí indicamos el intervalo de transmisión de las tramas, el canal utilizado para la transmisión y el modo de transmisión (conectable o no conectable). El formato utilizado es el que se ha seguido en la Figura 3-3, donde el primer <20 03> indica el intervalo mínimo en ms expresado en hexadecimal, el siguiente <20 03> el intervalo máximo, el siguiente <00> indica que se activa el modo conectable (03 activa el modo no conectable), <07> indica el canal en uso.

```

pi@raspberrypigleznegro2:~ $ sudo hcitool -i hci0 cmd 0x08 0x0006 20 03 20 03
00 00 00 00 00 00 00 00 00 00 07 00
< HCI Command: ogf 0x08, ocf 0x0006, plen 15
 20 03 20 03 00 00 00 00 00 00 00 00 00 07 00
> HCI Event: 0x0e plen 4
 01 06 20 0C
    
```

Figura 3-3 Ejemplo de ejecución del comando *LE Set Advertising Parameters* en la RPi2

A la hora de configurar nuestras RPi, pondremos siempre los mismos intervalos, mínimo y máximo, introduciendo los valores que corresponda para cada prueba o experimento, siguiendo la conversión que se muestra en la Tabla 3-1. El resto de los valores permanecerán siempre igual.

<i>100 ms</i>	A0 00
<i>500 ms</i>	20 03
<i>1000 ms</i>	40 06
<i>1500 ms</i>	60 09
<i>2000 ms</i>	80 0C
<i>2500 ms</i>	A0 0F
<i>3000 ms</i>	C0 12

Tabla 3-1 Conversión de intervalos de repetición

- **LE Set Advertising Enable:** Este comando es utilizado para iniciar o interrumpir la transmisión del sistema *iBeacon* en modo LEAd. El formato se muestra en la Figura 3-4, donde el valor <01> indica activación. Si quisiéramos desactivar este modo, introduciríamos <00> sustituyendo a este valor.

```
pi@raspberrypigleznegro2:~ $ sudo hcitool -i hci0 cmd 0x08 0x000a 01
< HCI Command: ogf 0x08, ocf 0x000a, plen 1
 01
> HCI Event: 0x0e plen 4
 01 0A 20 00
```

Figura 3-4 Ejemplo de ejecución del comando *LE Set Advertising Enable* en la RPi2

Una vez realizados todos estos pasos y ejecutado todos los comandos indicados en cada una de las RPis, estas se encuentran transmitiendo como radiobalizas.

3.2.3 Operación en modo escáner

La finalidad de esta última etapa de configuración es conseguir que la RPi1 opere en modo escáner BLE, asumiendo las funciones de un dispositivo *Bluetooth Smart Ready* y recopilar la información necesaria. Para ello es necesario activar la opción *lescan*, de las acciones disponibles en la pila de protocolos *BlueZ*, ejecutando el comando *hcitool lescan*. Una vez ejecutado dicho comando, la RPi1 empieza a recibir las PDU transmitidas por las otras RPis operando como radiobalizas.

El objetivo final de funcionamiento de la RPi1 es recopilar la información de la RSSI de cada baliza, con una periodicidad de un segundo y en un formato apropiado para su posterior análisis e interpretación al trabajar con las librerías de SVM. Para ello fue necesaria la creación de un script (*scansvm.sh*), el cual se muestra en el Anexo II y que devuelve la información que se muestra en la Figura 2-3. En la primera columna siempre se imprime un +1 (su función se explica en el apartado 3.3.6.3), en la segunda columna se representa la media de los valores de RSSI recibidos en el último segundo de la RPi2, en la siguiente columna de la RPi3 y así sucesivamente.

```
+1 2:-63 3:-100 4:-72 5:-100
+1 2:-62 3:-100 4:-67 5:-77
+1 2:-63 3:-100 4:-68 5:-100
+1 2:-60 3:-100 4:-65 5:-100
+1 2:-65 3:-100 4:-71 5:-100
+1 2:-60 3:-87 4:-71 5:-76
+1 2:-64 3:-75 4:-71 5:-76
+1 2:-62 3:-100 4:-64 5:-100
+1 2:-59 3:-100 4:-76 5:-85
+1 2:-64 3:-100 4:-79 5:-100
+1 2:-59 3:-77 4:-73 5:-68
+1 2:-62 3:-100 4:-74 5:-100
+1 2:-60 3:-100 4:-73 5:-87
+1 2:-63 3:-89 4:-73 5:-76
+1 2:-63 3:-100 4:-71 5:-100
+1 2:-62 3:-100 4:-68 5:-100
+1 2:-61 3:-100 4:-68 5:-100
+1 2:-59 3:-90 4:-69 5:-100
+1 2:-61 3:-100 4:-71 5:-80
```

Figura 3-5 Datos volcados tras ejecutar el script *scansvm.sh*

Hay que destacar que se le asigna el valor de RSSI -100 a las balizas cuando no se recibe ningún paquete de ella. Se ha escogido este valor porque tras realizar todas las pruebas previas se ha comprobado que es lo suficientemente bajo como para que ninguna baliza sea capaz de recibir un paquete con esta intensidad de señal.

En la fase de experimentación no fue necesario utilizar el formato de salida que se muestra en la Figura 3-5, por lo que se utilizó el script *scan.sh* (Anexo III) para obtener el formato mostrado en la Figura 3-6, más sencillo a la hora de importar a Excel los datos y representar las gráficas.

```

3 19:19:55 -56
2 19:19:55 -39
3 19:19:56 -70
3 19:19:56 -55
3 19:19:56 -58
2 19:19:56 -44
2 19:19:56 -54
3 19:19:56 -71
    
```

Figura 3-6 Formato de salida tras ejecutar el script *scan.sh*

3.3 Fase de experimentación

Una vez finalizada la configuración de las RPis para operar en modo baliza y escáner, se realizan una serie de experimentos donde se estudia el comportamiento de la señal Bluetooth, tanto en términos generales, como variando diferentes parámetros modificables. Este apartado se dividirá en los diferentes experimentos e incluirán las diferentes pruebas realizadas en cada uno. Se avanzará desde lo más sencillo (estudiando la fluctuación de la señal) a lo más complejo (manipulación de los datos con SVM).

3.3.1 Experimento 1

El primer experimento se realizó en el hall de la segunda planta del cuartel de alumnos “Marqués de la Victoria” (extensión de 80 m²), como se puede observar en la Figura 3-7. El escenario se estableció de la siguiente manera:

- Se colocaron dos RPis en los extremos de dicha sala, que funcionarían de radiobalizas. Su posición queda marcada en la Figura 3-7 con los símbolos en color azul (*POS1*) y naranja (*POS5*).
- Se establecieron puntos fijos (*POS1*, *POS2*, *POS3*, *POS4* y *POS5*) donde se tomarían datos en estático o por los que se pasaría, al tomar los datos en movimiento.

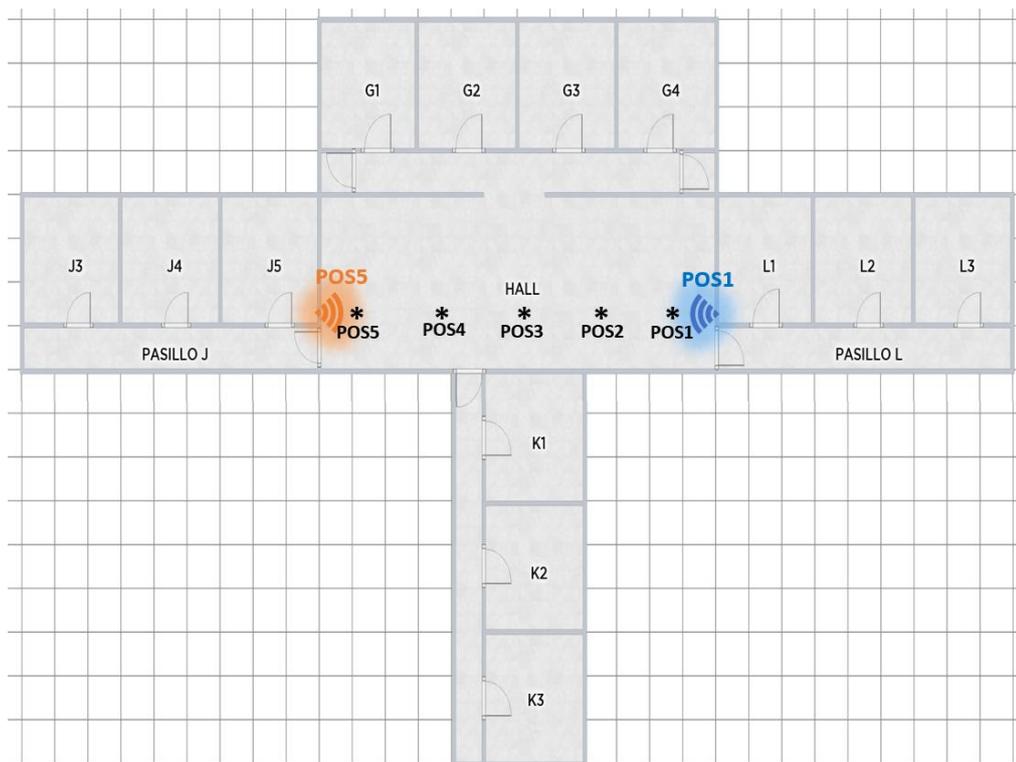


Figura 3-7 Zona central, segunda planta, Cuartel “Marqués de la Victoria”. Disposición *Experimento 1*

Se realizaron un total de diez pruebas. En todas ellas se siguió el mismo modelo a la hora de tomar las medidas y representar los resultados. Las medidas se tomaron de la siguiente manera:

- Por cada posición marcada en la Figura 3-7 (*POS1*, *POS2*, *POS3*, *POS4* y *POS5*) se tomaron datos durante 15 segundos fijos en cada posición, ejecutando el script *scan.sh* en la RPi1. Esta RPi se colocaba aproximadamente a un metro de altura y en posición horizontal.
- A continuación, se realiza un recorrido entre *POS1* y *POS5*, empezando a medio metro de la RPi más cercana a ese punto y acabando a medio metro también, de la RPi opuesta. La duración del recorrido oscila entre 30 y 40 segundos. A las diferentes gráficas, donde se representan los resultados de este recorrido, se les nombra *Desplazamiento lento*.
- Por último, se realiza el mismo recorrido, pero a mayor velocidad, con una duración entre 15 y 20 segundos. A los resultados de este recorrido se les nombra *Desplazamiento rápido*.

Una vez generados todos los ficheros de datos necesarios con el script, se pasan los datos a Excel para ser representados en gráficas. En el Anexo VI se exponen todas estas gráficas. Al principio de cada prueba hay una tabla con la información que la caracteriza. Esta información se define más adelante, cuando hablamos de los diferentes parámetros que se comparan en el escenario. En estas tablas hay cierta información que se encuentra subrayada en verde o en azul. Esto lo que nos quiere decir es que esa información es la que caracteriza a esa prueba y la que se quiere comparar con

En cada prueba, obtenemos una gráfica por cada posición fija, una del *Desplazamiento lento* y otra del *Desplazamiento rápido*. En el eje de abscisas representamos el tiempo y en el eje de ordenadas los dBm de RSSI. Quedando explicada aquí la representación de las gráficas, se elimina de dichas gráficas la información de los ejes para quedar todo visualmente más claro y menos cargado. Además, con las herramientas de Excel, generaremos una línea de tendencia para cada radiobaliza.

Los diferentes parámetros que se comparan en este escenario son los siguientes:

- Posición de la RPi: **Horizontal** y **vertical**. Se probaron diferentes posiciones de la antena para ver el efecto que esto tenía sobre la RSSI recibida. En la Figura 3-8 se muestran las dos posiciones diferentes de las RPis. Además, hay que tener en cuenta a la hora de consultar los diferentes planos de los experimentos realizados en el TFG, que la posición del símbolo amarillo indica también hacia donde apunta la RPis, es decir, sabiendo si la posición es horizontal o vertical y mirando este símbolo en los planos, podemos conocer su posición exacta.



Figura 3-8 Posición horizontal (izquierda), posición vertical (derecha)

- Altura de las radiobalizas: altura **mayor** y altura **menor**. Para la altura menor se colocaron las RPis sobre una mesa de un metro de altura aproximadamente y para la altura mayor se colocaron dos metros por encima de la altura de la mesa.
- Periodo de transmisión de paquetes: **100 ms, 500 ms y 100 ms mostrando 1 de cada 5 muestras**.
- Escáner: **RPi1 y móvil**. Se realizaron algunas pruebas con un móvil *Samsung Galaxy S6 Edge*, utilizando la aplicación *Android BLE Scanner & Recorder*, con el objetivo de estudiar si influía el dispositivo con el que recibíamos los datos. En la Figura 3-9 se muestra una captura de pantalla de la información en tiempo real que muestra la aplicación Android mientras está escaneando y guardando la información en un archivo Excel.

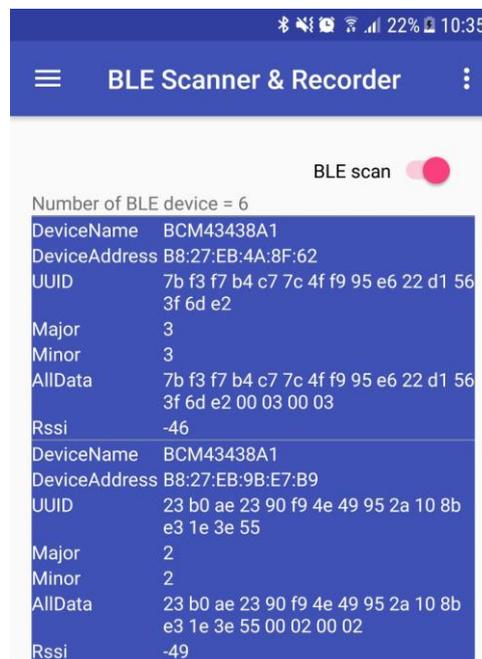


Figura 3-9 Ejemplo visualización de la aplicación *Android BLE Scanner & Recorder*

En la Figura 3-10 se muestra el formato en el que la aplicación devuelve los datos capturados tras realizar el escáner. A la hora de trabajar con los datos, solo nos interesarán las columnas *DataTime*, *Major* y *RSSI*.

DateTime	DeviceName	DeviceAddress	UUID	Major	Minor	ScanRecord	RSSI
2018-01-29 18:32:20	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-46
2018-01-29 18:32:20	BCM43438A1	B8:27:EB:4A:8F:62	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2	3	3	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2 00 03 00 03	-68
2018-01-29 18:32:21	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-45
2018-01-29 18:32:22	BCM43438A1	B8:27:EB:4A:8F:62	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2	3	3	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2 00 03 00 03	-71
2018-01-29 18:32:23	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-45
2018-01-29 18:32:23	BCM43438A1	B8:27:EB:4A:8F:62	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2	3	3	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2 00 03 00 03	-71
2018-01-29 18:32:24	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-54
2018-01-29 18:32:24	BCM43438A1	B8:27:EB:4A:8F:62	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2	3	3	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2 00 03 00 03	-62
2018-01-29 18:32:25	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-67
2018-01-29 18:32:26	BCM43438A1	B8:27:EB:4A:8F:62	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2	3	3	7b f3 f7 b4 c7 7c 4f f9 95 e6 22 d1 56 3f 6d e2 00 03 00 03	-62
2018-01-29 18:32:26	BCM43438A1	B8:27:EB:9B:E7:B9	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55	2	2	23 b0 ae 23 90 f9 4e 49 95 2a 10 8b e3 1e 3e 55 00 02 00 02	-62

Figura 3-10 Ejemplo archivo Excel generado por la aplicación *Android BLE Scanner & Recorder*

- Radiobalizas: **RPi2 y RPi3; RPi4 y RPi5**. Con las balizas se hacen dos comparaciones diferentes:
 - Primero se intercambian las mismas RPis de posición (RPi2 y RPi3) para comprobar que ninguna transmite a mayor potencia que la otra.
 - Después se sustituyen las RPi2 y RPi3 por las RPi4 y RPi5 para ver cómo influye utilizar RPis con antenas ya integradas o RPis con módulos USB.

Antes de pasar a analizar los resultados, debemos incidir en lo siguiente:

- Teniendo en cuenta dónde están colocadas las posiciones fijas, en las gráficas, las líneas de tendencia tendrían la separación mayor en POS1 y POS5, mientras que, en POS2 y POS4 la separación sería menor y, finalmente, en POS3 las líneas de tendencia deberían coincidir, ya que nos encontramos a la misma distancia de una radiobaliza que de otra.
- Por otro lado, en las gráficas de los recorridos, las líneas de tendencia deberían cruzarse en el medio, por el mismo motivo.

Los resultados y conclusiones obtenidas a partir de las gráficas son las siguientes:

- La fluctuación de la señal bluetooth es bastante alta. Como se puede observar en el ejemplo de la Figura 2-13, las líneas de tendencia siguen una trayectoria bastante coherente, ya que mientras nos acercamos a una RPi, nos estamos alejando de otra, y en el centro de la trayectoria nos encontramos a la misma distancia de ambas. Sin embargo, hay numerosas muestras que se alejan tanto de esta línea de tendencia, que podrían ser RSSI que perteneciesen a la otra RPi. Esto podría afectar bastante a la precisión de nuestro sistema. En el *Experimento 3* se realizará una pequeña demostración que nos puede hacer ver cuánto nos afecta esta fluctuación en cuánto a un mapeo de muestras con vistas a ser clasificadas con SVM.

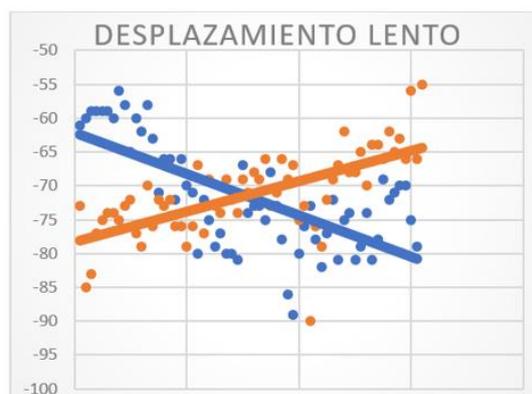


Figura 3-11 *Experimento 1, Prueba 5*

- Posición de las RPis: Centrándonos en las pruebas 1 y 2, donde la única variable es la posición, pero también teniendo en cuenta el resto de las pruebas, no se detecta ningún efecto notable que sea consecuencia de la posición. A la hora de colocarlas en otros experimentos, nos será indiferente.
- Altura: Al igual que con la posición, no se le puede asignar ninguna consecuencia clara al hecho de colocar las RPis a diferente altura. No obstante, se aprecia una mayor RSSI en los desplazamientos cuando la altura es menor (la RPi que actúa de escáner se encuentra a la misma altura que las radiobalizas) como se observa en la Figura 3-12 y Figura 3-13.

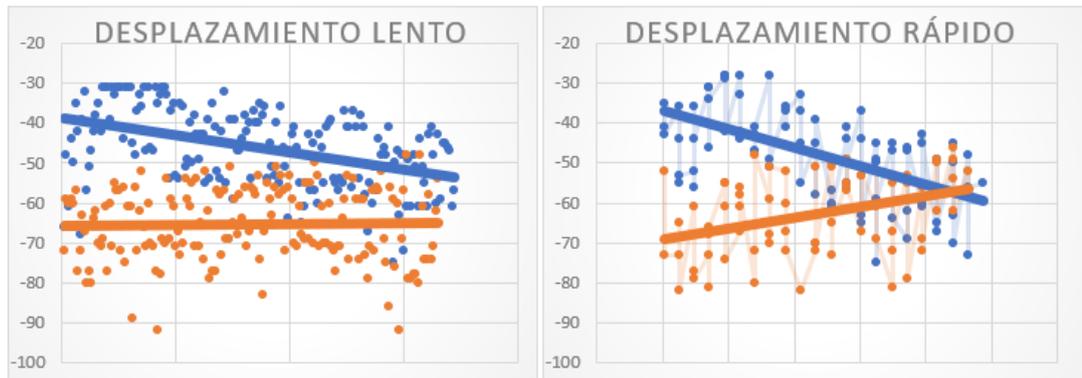


Figura 3-12 Experimento 1, Prueba 2

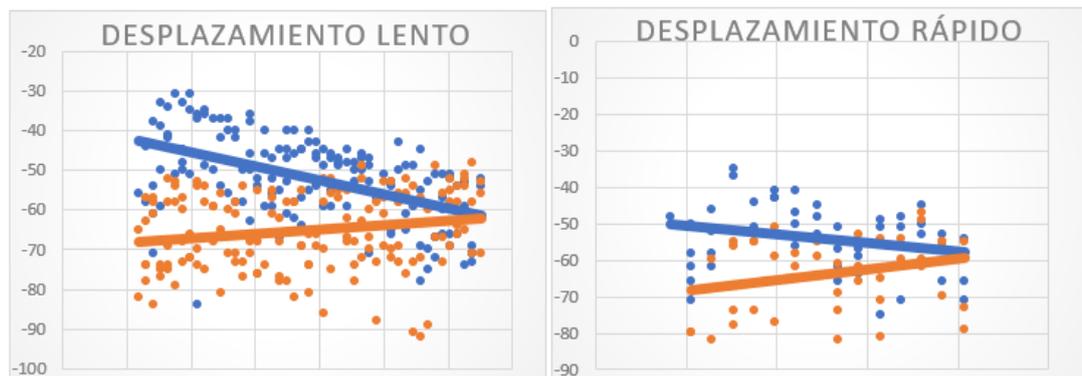


Figura 3-13 Experimento 1, Prueba 3

El hecho de que la posición de la antena afecte a la RSSI recibida, nos es indiferente al utilizar SVM como técnica de localización. Como se ha explicado en el capítulo 2 de este trabajo, SVM no se incluye entre las técnicas de medición de posicionamiento a partir de algoritmos geométricos, sino a partir de reconocimiento de patrones. Por lo tanto, con SVM no tenemos la necesidad de sacar una relación entre la RSSI y la distancia a la que nos encontremos de la radiobaliza. En nuestro caso, lo importante es que, independientemente del valor de la RSSI recibida en las diferentes zonas a localizar, ésta se mantenga siempre igual, es decir, que siga siempre el mismo patrón y comportamiento a lo largo del tiempo. Como conclusión, podemos decir que la altura a la que coloquemos las radiobalizas nos será indiferente.

- Escáner: En las gráficas del Anexo VI, en las pruebas 4 y 5 se utilizan diferentes escáneres al realizar las pruebas. En la prueba 5, donde se utiliza el móvil y la aplicación que se ha indicado anteriormente, en general se aprecia menos fluctuación y la RSSI es menor. En la Figura 3-14 y la Figura 3-15 se aprecia claramente lo que se acaba de describir. Asimismo,

en el *Experimento 2* se vuelve a demostrar que los resultados obtenidos con el móvil son más estables.

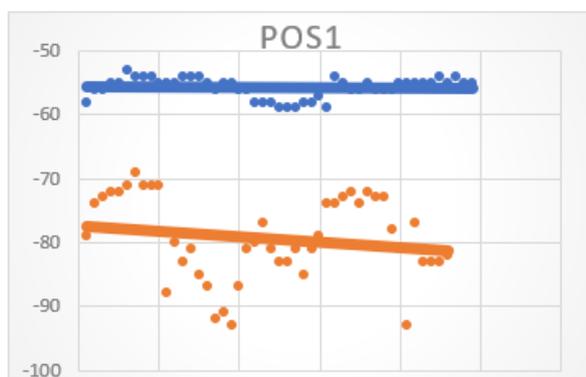


Figura 3-14 *Experimento 1*, prueba 5. Escáner móvil

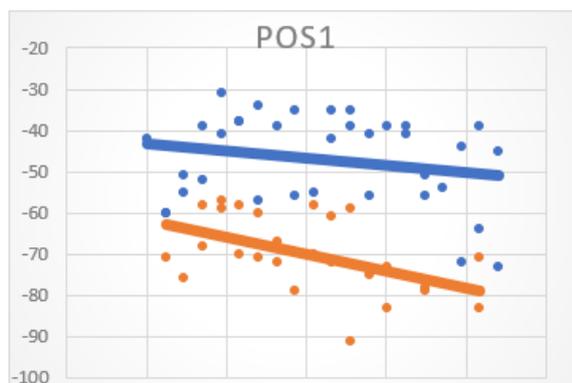


Figura 3-15 *Experimento 1*, prueba 4. Escáner RPi1

Como conclusión de estas dos pruebas, podemos afirmar que los resultados del escaneo se ven influidos por el dispositivo que estemos utilizando como escáner. Por lo tanto, si en un sistema de localización en interiores utilizamos escáneres diferentes, vamos a perder precisión. Lo ideal sería utilizar siempre el mismo modelo de dispositivo, aunque se podría estudiar cuánto difieren los resultados en la práctica al utilizar escáneres diferentes. En nuestro TFG, seguiremos utilizando la RPi1 como escáner, ya que el objetivo del TFG es implementar el sistema de localización en interiores con estos dispositivos.

- Tras analizar los resultados de las gráficas obtenidas en las pruebas 2 y 3, se observó lo siguiente:
 - Como podemos ver en la Figura 3-12 y en la Figura 3-13, en los desplazamientos las líneas de tendencia no se cruzan, de manera que la RPi2 podría estar transmitiendo con mayor potencia.
 - En la *POS3*, la RSSI de la RPi2 era mayor que la de la RPi3, por lo tanto, se vuelve a confirmar que la RPi2 podría estar transmitiendo con mayor potencia.

Con los resultados obtenidos en las pruebas 6 y 7, se puede concluir que esta hipótesis es incorrecta. Cuando se intercambian las posiciones de las RPi2 (azul en la Figura 3-16 y Figura 3-17) y RPi3 (naranja en la Figura 3-16 y Figura 3-17) pasa a ser la RPi3 la que parece transmitir con mayor potencia. Este fenómeno no parece estar causado porque una RPi transmita con más potencia que otra, sino parece más bien efecto del entorno y otras

causas externas, como interferencia con otros equipos que operan en la banda de Bluetooth (ISM, *Industrial, Scientific and Medical*).

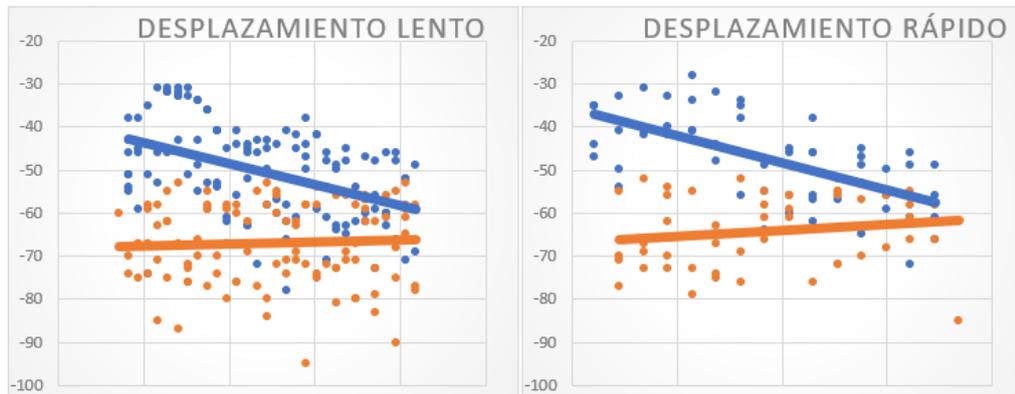


Figura 3-16 Experimento 1, Prueba 6

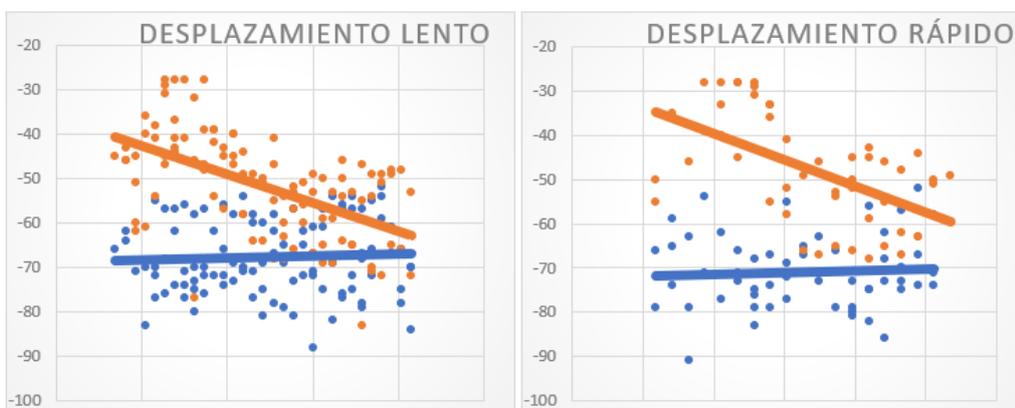


Figura 3-17 Experimento 1, Prueba 7

- Si repasamos todas las gráficas de los desplazamientos del Anexo VI, nos daremos cuenta de que cuando las radiobalizas transmiten con un periodo de 100 ms, las líneas de tendencia no se cruzan, pero si transmiten 500 ms, sí se cruzan. Sin embargo, en las posiciones fijas no se aprecia nada relevante al cambiar esta configuración de transmisión. Para verificar que esto es así se realizó la prueba 8, en la cual, las radiobalizas transmitían cada 100 ms, pero al representar los resultados en las gráficas, se obviaba cuatro de cada cinco muestras. El resultado fue el mismo, las líneas de tendencia no se cruzaban. De esa manera, se concluyó que, cuando realicemos experimentos en los que existan desplazamientos, las radiobalizas deben transmitir a 500 ms para obtener resultados coherentes. En la Figura 3-18, Figura 3-19 y Figura 3-20 se muestra el patrón seguido por la RSSI, variando la periodicidad de transmisión de las radiobalizas.

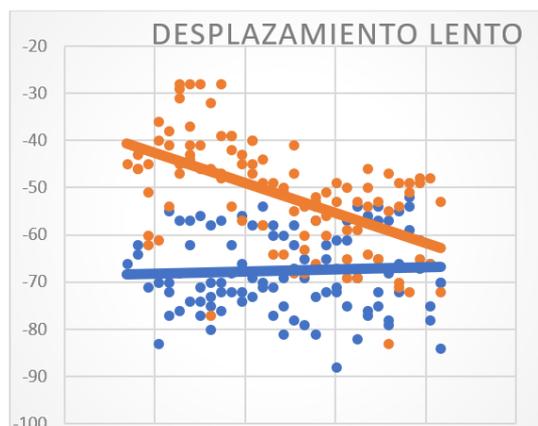


Figura 3-18 Experimento 1, Prueba 7. Periodo de 100 ms



Figura 3-19 Experimento 1, Prueba 8. Periodo de 100 ms (eliminando 4 de cada 5 muestras)

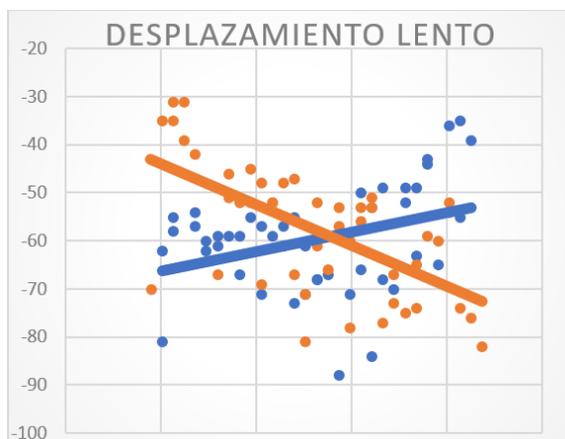


Figura 3-20 Experimento 1, Prueba 9. Periodo de 500 ms

- Por último, se comprobó que los diferentes modelos de RPis, los cuales funcionaban con antenas diferentes, ofreciesen resultados parecidos. En las pruebas 9 y 10, donde las únicas variables son las radiobalizas utilizadas, no se destaca ninguna diferencia aparente. En la Figura 2-19 y Figura 2-20 se pone de ejemplo la gráfica resultante del recorrido con desplazamiento lento de cada una de estas pruebas.

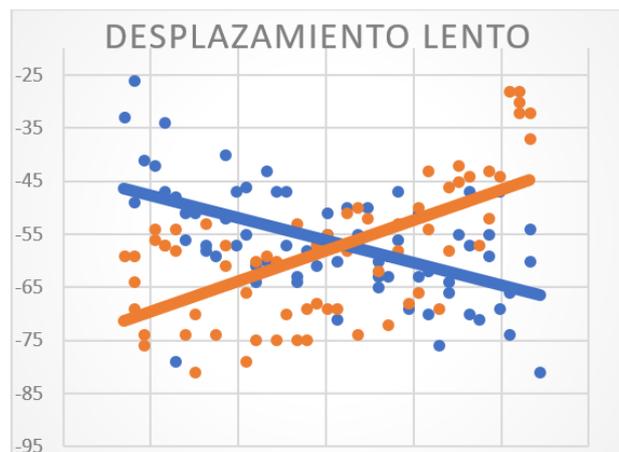


Figura 3-21 Experimento 1, Prueba 10

3.3.2 Experimento 2

Tras haber realizado numerosas pruebas en el mismo lugar, hall del Cuartel “Marqués de la Victoria”, se decide cambiar de escenario. Para ello se escogieron los seminarios del CUD, ya que las características de este lugar difieren de las del Cuartel “Marqués de la Victoria” en:

- Las paredes. Aunque son del mismo material (pladur), el aislante y el grosor pueden ser diferentes.
- La disposición de otros dispositivos que emiten ondas electromagnéticas, como routers, por ejemplo, es diferente.
- Los seminarios del CUD y el pasillo central que los separa presentan formas y dimensiones diferentes.

Por otro lado, se quiere observar también el efecto que produce el funcionamiento de otras tecnologías inalámbricas en los dispositivos con los que se trabaja, mientras se toman muestras.

Este experimento consta de cuatro pruebas diferentes. El primer objetivo del experimento será abordado con la *Prueba 1* y el segundo objetivo con las *Pruebas 2, 3 y 4*. A la hora de tomar las medidas, se siguió el mismo modelo del *Experimento 1*. La diferencia es que en la *Prueba 1* solo se toman tres puntos de referencia en vez de 5, como se observa en la Figura 3-22. A continuación, se explicará cada prueba y se comentarán los resultados obtenidos.

3.3.2.1 Prueba 1

En la Figura 3-22 se muestra la disposición de las RPIs utilizadas como radiobalizas y los puntos de referencia que marcan el recorrido de la prueba. En el Anexo VII se encuentran los resultados representados en gráficas, siguiendo el mismo modelo que en el *Experimento 1*.

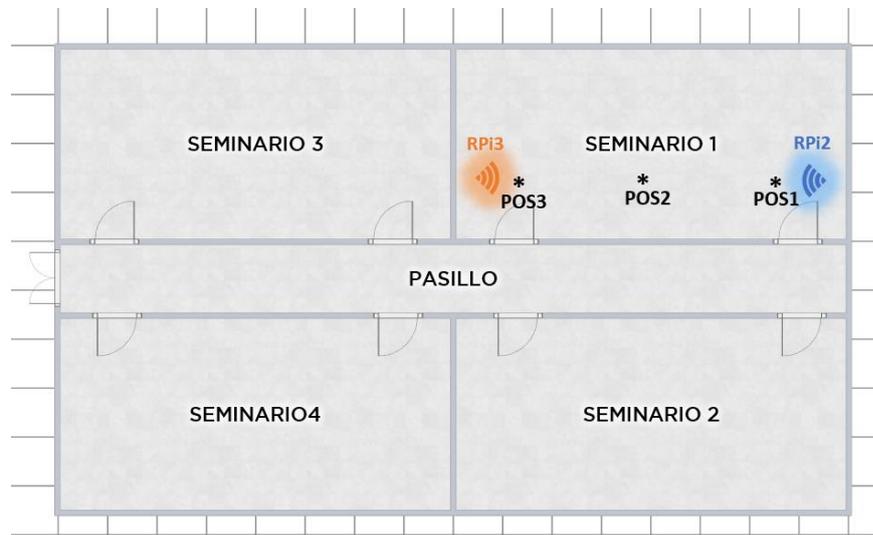


Figura 3-22 Seminarios del CUD. Disposición Experimento 2, Prueba 1

Tras comparar las gráficas de estas pruebas, con las del *Experimento 1*, podemos concluir que la señal Bluetooth sigue el mismo comportamiento en este tipo de pruebas. La verdadera influencia de las paredes en nuestro sistema se verá reflejada en los siguientes experimentos, concretamente en el *Experimento 5*.

3.3.2.2 Pruebas 2, 3 y 4

La Figura 3-23 muestra la disposición de las RPIs operando como radiobalizas y los puntos de referencia escogidos. Al igual que en la prueba 1 de este experimento, podemos encontrar el resto de los resultados en el Anexo VII.

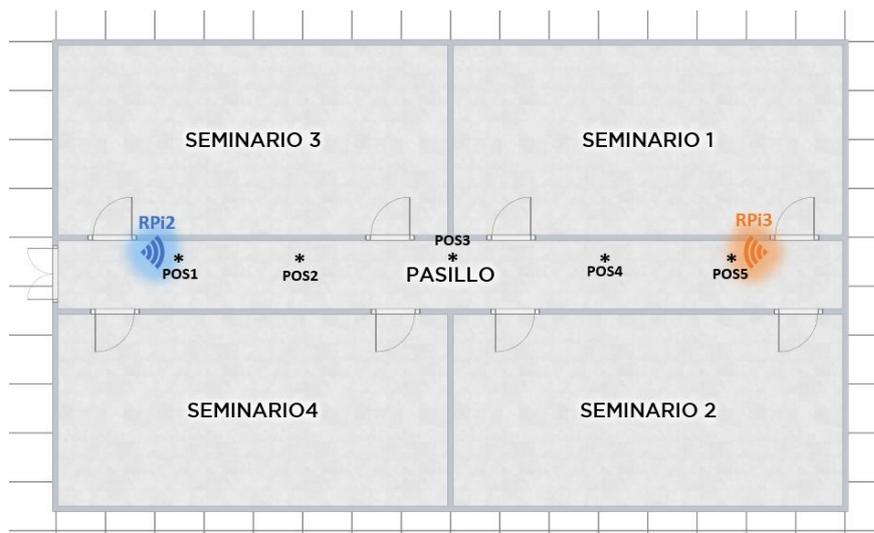


Figura 3-23 Seminarios del CUD. Disposición *Experimento 2*, Pruebas 2, 3 y 4

Con los resultados obtenidos en estas pruebas, se tiene como conclusión lo siguiente:

- Se vuelve a demostrar que el uso del móvil como escáner devuelve resultados más precisos y una menor fluctuación de la señal. Si comparamos los resultados de las pruebas 3 y 4, de este experimento, con las de la prueba 5 del *Experimento 1*, observaremos que la fluctuación en el *Experimento 2* es menor. Esto se debe a que, en el *Experimento 1* el móvil tenía el WiFi y los datos activados y en el *Experimento 2* no.
- También se probó a realizar un escáner con el móvil mientras se tenía activada la opción de compartición de datos del móvil. En este modo, el móvil perdía más de la mitad de las muestras.
- No se pudo probar a desactivar el WiFi de la RPi1 cuando operaba en modo escáner, ya que era necesario para acceder a ella por control remoto.
- Según los resultados obtenidos en las pruebas 3 y 4, el hecho de que las radiobalizas tengan el WiFi activado o no, no afecta a los resultados. De esta manera, cuando los dispositivos que transmitiesen (las radiobalizas) tienen activado el WiFi, no se aprecian efectos negativos, mientras que si son los dispositivos que reciben (los escáneres) los que tienen el WiFi activado, sí que afecta negativamente a los resultados.

3.3.3 Experimento 3

Al igual que el *Experimento 1*, el *Experimento 3* se realizó en el Cuartel “Marqués de la Victoria”. Si las pruebas anteriores se centraban en términos más generales, cómo la colocación de los dispositivos o el comportamiento de la señal Bluetooth; este experimento va más enfocado a la futura manipulación de las muestras con librerías de SVM^{light} que se hará en los experimentos posteriores.

Tras haber observado en los *Experimentos 1* y *2* que el valor de RSSI recibido fluctúa bastante en posiciones fijas, surge la necesidad de comprobar cuánto nos afecta esto, para la futura clasificación con SVM que hagamos, de manera que aseguremos que podemos seguir trabajando en el TFG con las RPi y el protocolo Bluetooth para obtener un sistema de localización en interiores. Para ello se ha diseñado el siguiente escenario:

- Se coloca la RPi1 y RPi2 en el hall del Cuartel “Marqués de la Victoria”, en la disposición observada en la Figura 3-24, con la altura menor y posición horizontal descritas en el *Experimento 1*. Dichas RPis operarán como radiobalizas, transmitiendo paquetes con una

periodo de 100 ms. Las dos RPi3 estaban alimentadas por un cable que las conectaba a la red eléctrica.

- Las mediciones se realizan en las camaretas o zonas marcadas con un símbolo diferente para cada una (*G1, G4, HALL, L1 y K3*). El conjunto de datos obtenidos en cada sala formará lo que en SVM se conoce como *clase*. Estas clases son representadas en una gráfica de dos dimensiones, donde una dimensión será la RSSI de la RPi2 y la otra de la RPi3. El fin último de este experimento es ver si las clases se agrupan y son fácilmente diferenciables en la gráfica, es decir, que se podrían separar fácilmente por un *vector*.

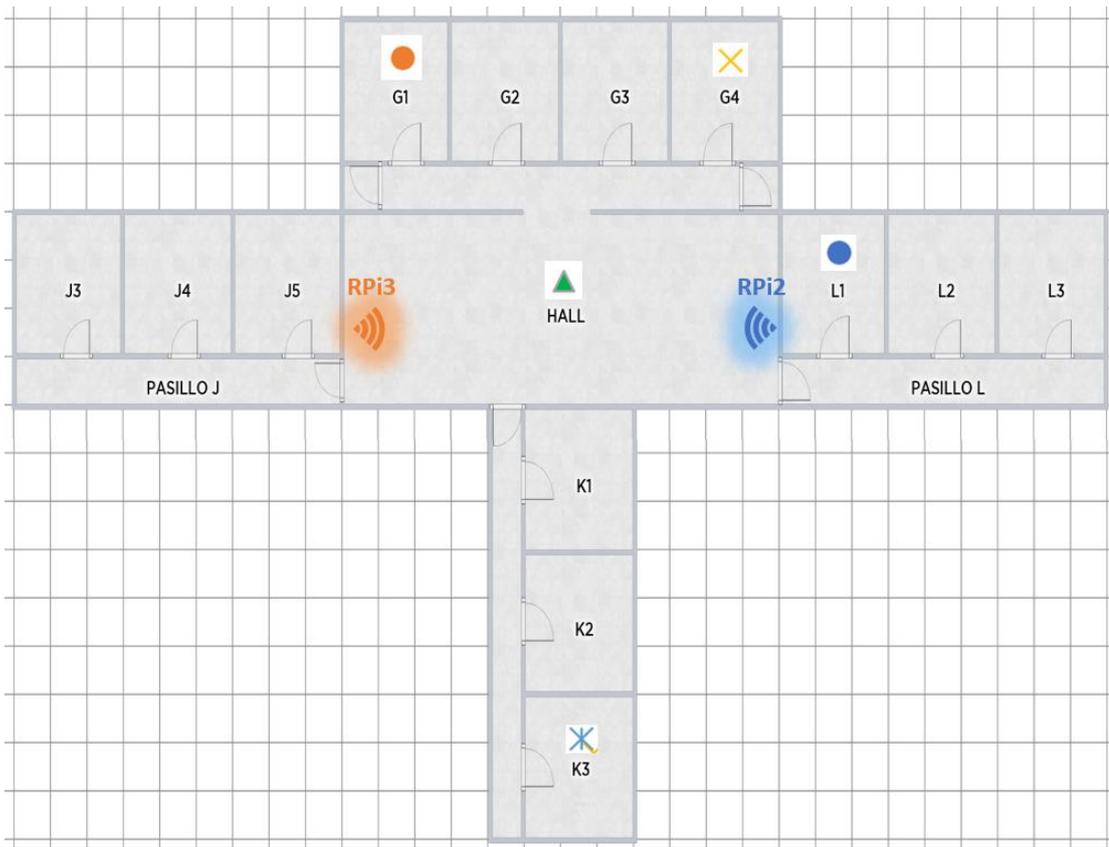


Figura 3-24 Zona central, segunda planta, Cuartel “Marqués de la Victoria”. Disposición Experimento 3

El procedimiento que se utilizó para la toma de datos y su representación fue el siguiente:

- Se tomaron datos durante un minuto con la RPi1, ejecutando el mismo script que en los anteriores experimentos (*scan.sh*), cubriendo todos los espacios del plano y la altura entre uno y dos metros con respecto al suelo.
- Los datos se pasaron a un archivo Excel, donde se realizó la media de las muestras capturadas en el mismo segundo.
- Estas medias se representaron en una gráfica a partir de dos coordenadas, una la RSSI de la RPi2 y la otra la RSSI de la RPi3. En la Figura 3-25 se pueden ver los resultados representados en la gráfica.

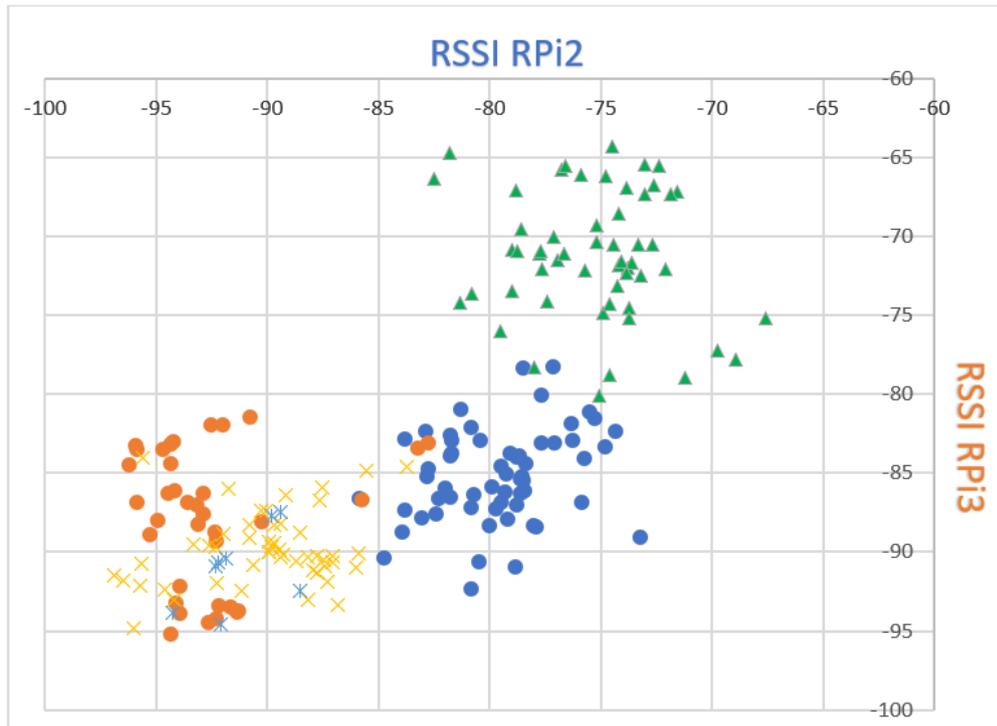


Figura 3-25 Resultados del Experimento 3

Tras obtener la gráfica, las observaciones y conclusiones fueron las siguientes:

- A pesar de la fluctuación de la señal mostrada en las gráficas de los Experimentos 1 y 2, y de los efectos negativos que puedan afectar a la señal (ruido, refracción, multitrayecto, reflexión, etc.) se obtiene una buena separación entre clases, al menos, entre las que pertenecen a *G1*, *L1* y *HALL*.
- Las camaretas *G1* y *G4* tienen algunas muestras que se mezclan. Esto puede ser debido a que la distancia de cada camareta con respecto a las dos radiobalizas no difiere mucho, y como además están colocadas simétricamente con respecto a las radiobalizas, estas dos distancias se repiten para ambas camaretas. Una manera de evitar esta situación sería, por ejemplo, colocando la *RPi3* en la esquina inferior izquierda del *HALL* y la *RPi2* en la superior derecha. De esa forma, las muestras de la *G1* se desplazarían hacia abajo y a la izquierda en la gráfica y las de la *G4* hacia arriba a la derecha.
- Las muestras de la camareta *K3* se mezclan con las camaretas *G1* y *G4*. Esto tiene sentido, ya que la disposición de *K3* con respecto a las balizas es prácticamente la misma que el centro entre la *G1* y *G4*, pero en el lado opuesto.
- Por otro lado, hay que destacar que estamos utilizando solamente dos radiobalizas. En el momento que utilizemos una tercera y la coloquemos, por ejemplo, en la *K1*, estamos ganando una tercera dimensión. Con esta tercera dimensión, las muestras de las clases que se superpongan en el plano se pueden separar en el espacio. De este hecho, sacamos como conclusión, que cuanto más radiobalizas coloquemos, más fácil va a ser separar las clases. En el caso límite esto nos llevaría a colocar una radiobaliza en cada espacio a diferenciar (sala, camareta, hall o pasillo) sin embargo queremos evitar este escenario por su coste en recursos.

3.3.4 Experimento 4

El *Experimento 3* fue una forma rápida de visualizar un mapeo sencillo de muestras y comprobar que se diferenciaban las clases, con solo dos radiobalizas y cinco espacios. En este experimento, se pretende abarcar todos los espacios que se pueda con cuatro radiobalizas y comprobar si la RSSI de las radiobalizas recibidas en cada uno de esos locales, nos permite diferenciar entre ellos, para después poder clasificar correctamente con SVM. Al utilizar cuatro balizas, no podemos representar en cuatro dimensiones los resultados como los hemos representado en el *Experimento 3*. Por este motivo se usa una representación de diagramas de barras. Cada barra representa el nivel de RSSI promediado de cada baliza en cada compartimento.

La descripción del escenario es la siguiente:

- Se colocaron las cuatro RPis disponibles para operar en modo radiobaliza en camaretas diferentes (*G1*, *G4*, *K1* y *K3*). En la Figura 3-28 se puede ver la posición exacta de las balizas. Se eligió esta disposición de balizas por las siguientes razones:
 - Se buscó una disposición en la que las radiobalizas se colocasen en los extremos del escenario. De esa forma, los vectores distancia que unen esas radiobalizas con cada una de las camaretas están mejor diferenciados.
 - Se optó por colocar las radiobalizas dentro de las camaretas y no en los pasillos o en el hall. De esta forma se juega con la atenuación producida por las paredes que dividen las camaretas. Como se puede ver en la Figura 3-26 de manera esquemática, las paredes ayudan a diferenciar la RSSI de la RPi5 (ubicada en *G1*) recibida en cada una de las camaretas.

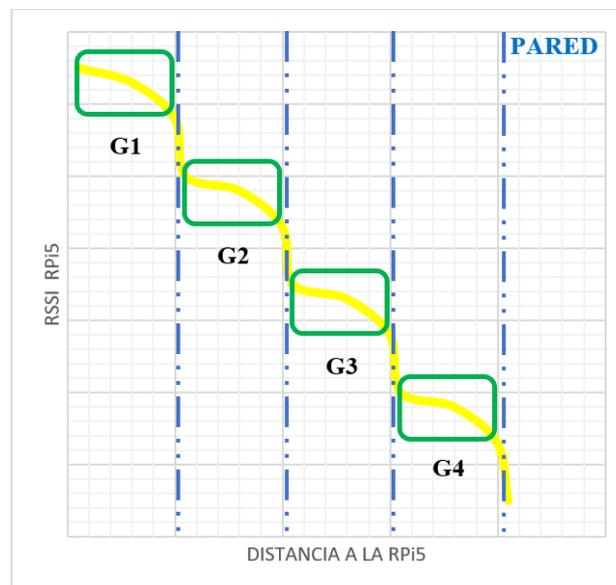


Figura 3-26 Gráfica representativa del efecto de atenuación de las paredes. Ver Figura 3-28

- Se pensó que, para cubrir un conjunto de camaretas contiguas, el hecho de colocar una radiobaliza en la primera camareta y otra en la última sería la forma óptima de diferenciar mejor entre esas camaretas. En la Figura 3-26 queda reflejado el efecto de colocar las radiobalizas de esta forma, verificándose en los diagramas de barra de la Figura 3-28.

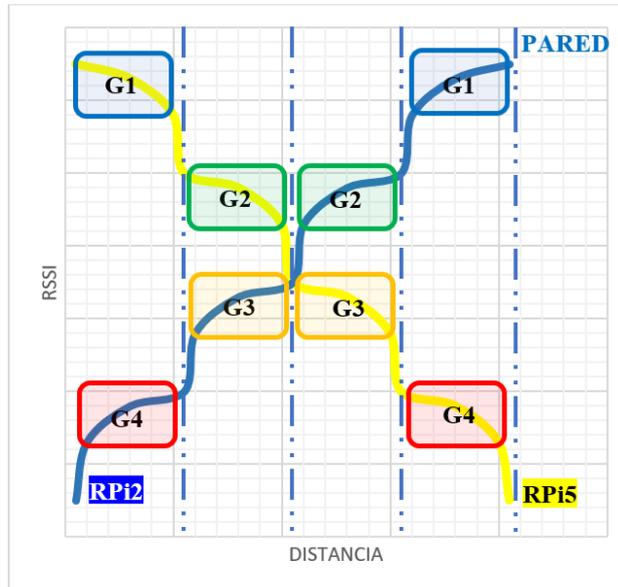


Figura 3-27 Gráfica representativa tras colocar las RPi5 y RPi2 como se muestra en la Figura 3-28

- Todas las radiobalizas se colocaron en horizontal, pero a diferente altura:
 - RPi2: Se colocó a una altura de 2 m, en la parte inferior de una ventana.
 - RPi3: Se dejó colocada en la parte superior de los armarios, aproximadamente a una altura de 2,5 m. La RPi2 y RPi3 estaban alimentadas por una batería externa.
 - RPi4 y RPi5: Se dejaron apoyadas sobre una estantería a 1,5m. de altura. Ambas estaban conectadas a la red eléctrica.
- Las RPis que operan en modo radiobalizas se ponen a transmitir paquetes con una periodo de 100 ms.
- Para este experimento, se estudian un total de 11 camaretas, 4 pasillos y el hall; es decir, todos los locales presentes en el plano de la Figura 3-28, excepto los dos cuartos del *PASILLO G* (por su pequeño tamaño) y las camaretas *J3* y *L3* (donde no llega apenas señal de las radiobalizas).

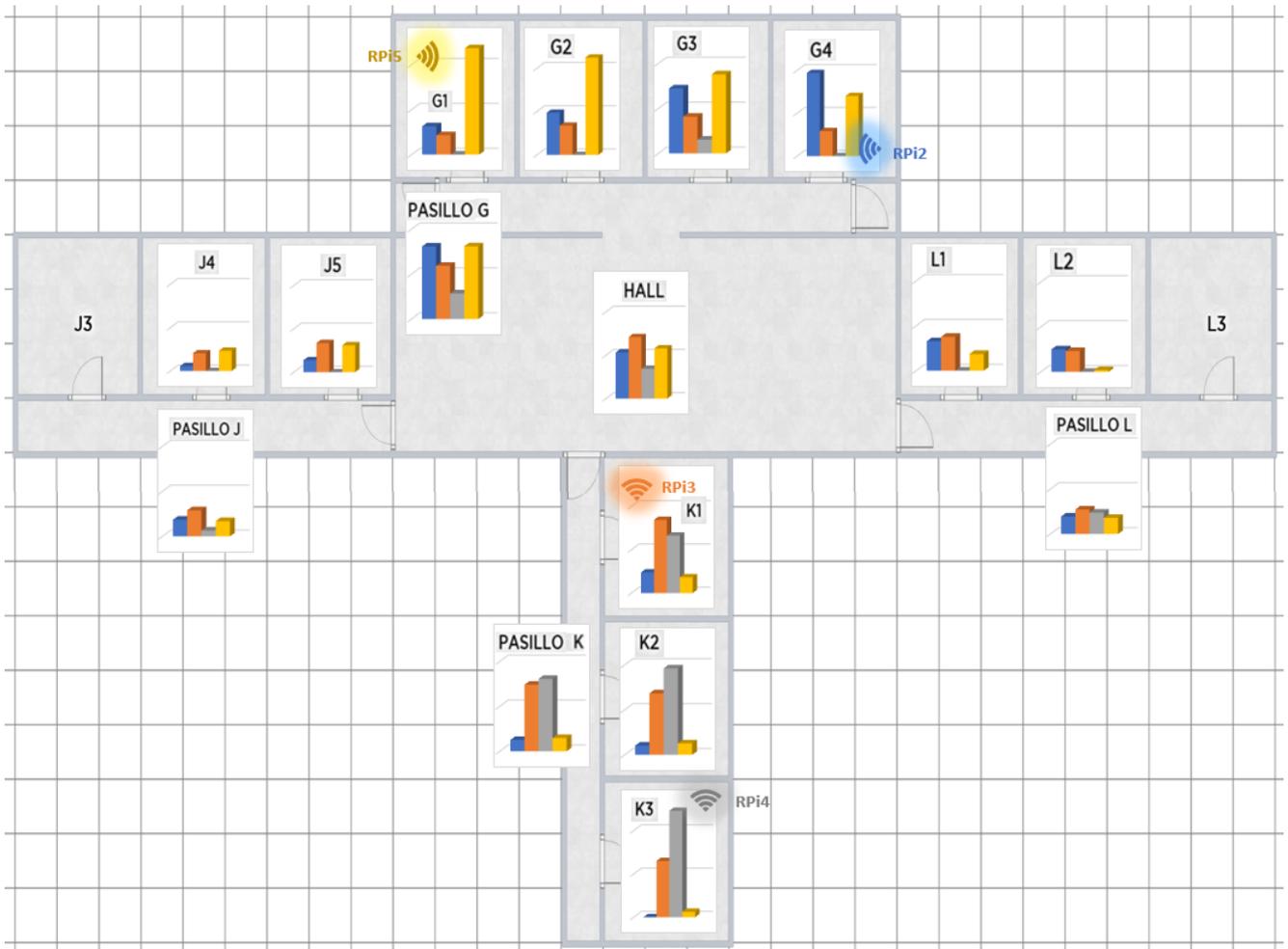


Figura 3-28 Disposición, escenario y resultados del *Experimento 5*

A continuación, se describen los pasos para la toma de datos y representación de los resultados:

- Para la toma de datos por camareta, se sigue el mismo procedimiento que en el *Experimento 3*.
- Los datos son pasados al Excel, donde se realiza la media de todos los valores de RSSI de cada radiobaliza. Los resultados se pueden ver en el Anexo VIII. A aquellas camaretas donde no se recibe la señal de alguna radiobaliza, se les asigna el valor de -90dBm. En los resultados del Anexo VIII está marcado en rojo. Posteriormente, en el capítulo destinado a la validación del sistema, este valor se define como -100 dBm, ya que se considera que el margen debe ser mayor.
- A partir de estos resultados, se representan diagramas de barras donde se le asigna una barra a cada RPi, con el mismo color con el que se representa la RPi en la Figura 3-28, es decir, de izquierda a derecha con el siguiente orden: *RPi2*, *RPi3*, *RPi4* y *RPi5*. El valor mínimo, de estos diagramas, es -90 dBm y el máximo -50 dBm. Posteriormente, se representan en el plano del escenario, sobre el espacio o sala que le corresponde, como se puede ver en la Figura 3-28.

A partir de los resultados obtenidos, se concluyó lo siguiente:

- Las *camaretas G* y *K* se diferencian bastante bien entre ellas, por lo que la disposición de las radiobalizas ha beneficiado la distinción de estas camaretas.

- Las *camaretas L* son fácilmente diferenciables del resto de camaretas, pero no entre sí. En el caso de las *camaretas J* ocurre lo mismo.
- El *Hall* sigue un patrón que le permite ser bien diferenciado del resto. Sin embargo, al abarcar tanto espacio, no podemos garantizar que no haya zonas conflictivas, donde no se pueda distinguir bien la localización, ya que nosotros hemos representado la media de todas las zonas que se pueden cubrir.
- En el caso de los pasillos, hay algunos que podemos confundir con camaretas, es el caso, por ejemplo, del *PASILLO G* y la camareta *G3* o el *PASILLO K* y la camareta *K2*. Por otro lado, los pasillos presentan una geometría muy alargada, por lo tanto, puede que ocurra el mismo problema que tenemos con el *HALL*, que hay puntos conflictivos que no conozcamos. Teniendo en cuenta este problema, podemos obtener como conclusión que los pasillos y las zonas que abarcan gran extensión son locales conflictivos.
- Si no tenemos en cuenta los pasillos y las camaretas *J4* y *L2* hemos sido capaces de abarcar 11 locales o zonas con 4 balizas.

3.3.5 Experimento 5

El objetivo de este experimento es comprobar el efecto que tiene el tipo de pared del lugar donde estamos instalando el sistema de localización, sobre las mediciones y los resultados. Para ello, volvemos a plantear el mismo escenario que en el *Experimento 2*, los seminarios del CUD, siguiendo ahora el modelo del *Experimento 4* y comparando resultados con éste.

Al repetirse los mismos procedimientos que en el *Experimento 4*, no se van a volver a explicar con detalle. Las características principales del escenario vienen indicadas en las tablas de la Figura 3-29 y Figura 3-30. Para más detalle, leer la descripción del escenario, la toma de datos y representación de los resultados del *Experimento 4*.

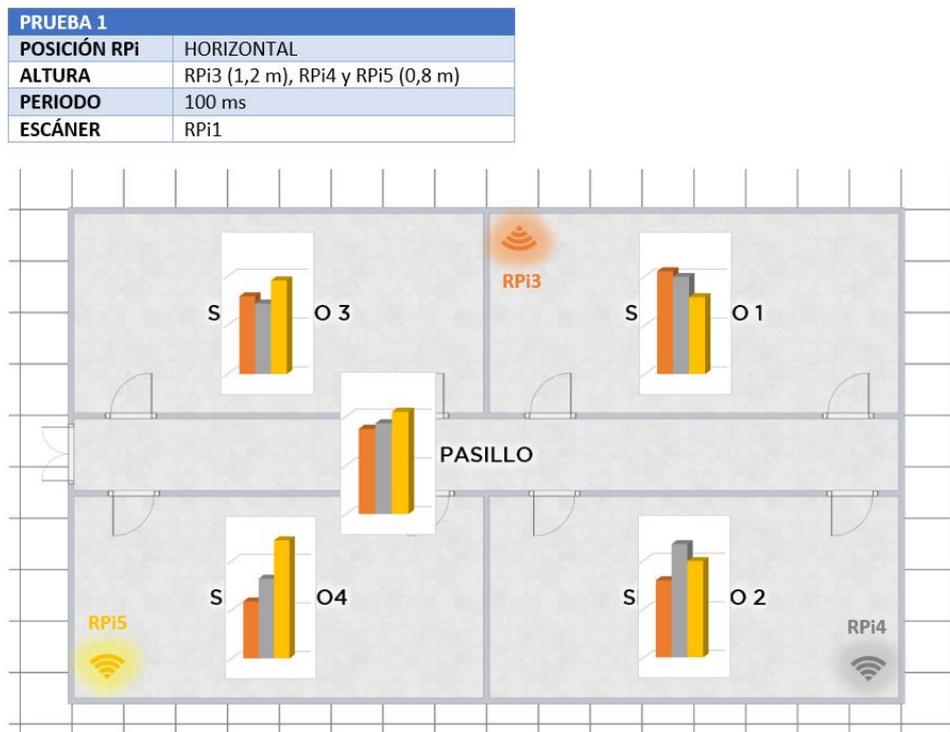


Figura 3-29 Disposición, escenario y resultados del *Experimento 5*, Prueba 1

PRUEBA 2	
POSICIÓN RPi	HORIZONTAL
ALTURA	RPi3 y RPi4 (0,8 m), RPi5 (1,5 m)
PERIODO	100 ms
ESCÁNER	RPi1

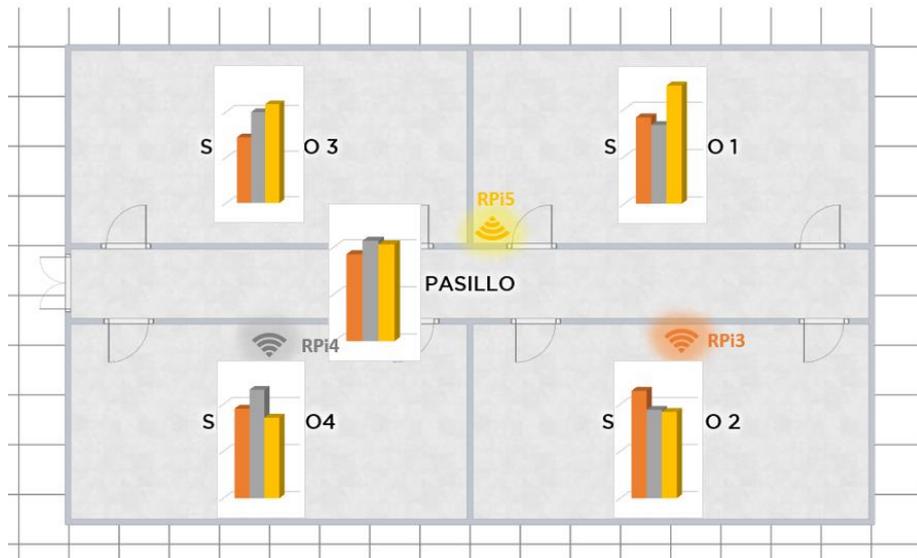


Figura 3-30 Disposición escenario y resultados del Experimento 5, Prueba 2

Las conclusiones de los resultados del Experimento 5 son las siguientes:

- Como podemos apreciar en los diagramas de barra, en este escenario es mucho más difícil diferenciar entre locales. Hay que tener en cuenta el fenómeno de la fluctuación, el cual no nos permite tener la precisión suficiente para diferenciar entre un seminario y otro.
- Probando diferentes disposiciones de radiobalizas no conseguimos resolver este problema. La conclusión que podemos sacar de este experimento es que la composición de las paredes es uno de los factores más importantes a la hora de implementar un sistema de localización en interiores. En el caso del “Tabarca” no tendremos ese problema, ya que la gran mayoría de los compartimentos están separados por chapas de acero.

3.3.6 Experimento 6

Este experimento tiene como finalidad cerrar esta fase realizando una prueba sencilla con el software SVM^{light} [56], de manera que quede explicada su utilidad y desarrollemos un modelo de trabajo, para aplicarlo en el patrullero “Tabarca”, para la validación del sistema.

El objetivo de este experimento y de los que se realicen en el “Tabarca” será analizar los resultados de clasificación de SVM^{light} en estos escenarios.

3.3.6.1 Descripción de SVM^{light}

Como se ha expuesto en los objetivos, este TFG tiene como finalidad probar un sistema de localización en interiores basado en SVM. Para ello se ha optado por utilizar las herramientas aportadas por Thorsten Joachims en su desarrollo de SVM^{light}. Se utiliza la última versión disponible, 6.02 con fecha de última actualización 14 de agosto de 2008. Se puede consultar su trabajo en [56]. A continuación, presentamos una breve descripción de SVM^{light}.

SVM^{light} es una implementación *Support Vector Machine* de Vapnik [57] en C, para el problema de reconocimiento de patrones, de regresión y de aprendizaje de una función de clasificación. Los algoritmos de optimización utilizados se describen en [58] y [59].

El código se ha probado en una amplia gama de problemas, incluida la clasificación de textos [60], [61], tareas de reconocimiento de imágenes, aplicaciones de búsqueda, sistemas de recomendación, bioinformática y aplicaciones médicas.

En la página web [56] se encuentra disponible la implementación, que se desarrolló en Solaris 2.5 con gcc, y la compilación en SunOS 3.1.4, Solaris 2.7, Linux (la que utilizaremos para este trabajo), IRIX, Windows NT y Mac.

A continuación, se recopilan las principales características y funciones del programa:

- Algoritmo de optimización rápida.
- Resuelve problemas de clasificación y regresión. Usar SVM^{struct} [54] para salidas multivariadas y estructuradas.
- El software proporciona métodos para evaluar el rendimiento de generación. Calcula estimaciones de tasa de error, de precisión y de recuperación (*recall*) [58] [62].
- La versión 6.02 incluye un nuevo algoritmo para aprender funciones de *ranking* [63]. El objetivo de este algoritmo es aprender una función a partir de ejemplos preferentes, de modo que ordene un nuevo conjunto de muestras con la mayor precisión posible. Este algoritmo se utiliza, por ejemplo, en el aprendizaje de funciones de recuperación en el motor de búsqueda *STRIVER* [64]. Se barajó la posibilidad de utilizar este algoritmo para nuestro propósito en el TFG si el modo de clasificación binario (predeterminado), no diese buenos resultados.
- La ejecución de los algoritmos tiene requisitos de memoria escalable, con lo que se puede manejar problemas de muchos miles de vectores de soporte de manera eficiente.
- Maneja varios cientos de miles de ejemplos de entrenamiento.
- Admite funciones estándar de Kernel y permite definir funciones propias.
- Usa representación de vector disperso.

3.3.6.2 Instalación de SVM^{light}

Los pasos seguidos para la instalación de SVM^{light} son los siguientes:

- Se descarga el archivo *svm_light.tar.gz* de [56] y se crea un directorio que se llame *svm_light*, descomprimiendo el archivo anterior en este directorio.
- Se ejecuta el comando *make* o *make all* para compilar el sistema y crear los dos ejecutables:
 - *svm_learn* (módulo de aprendizaje o entrenamiento)
 - *svm_classify* (módulo de clasificación o evaluación)

3.3.6.3 Utilización de SVM^{light}

Como se ha podido ver en la explicación de la instalación, SVM^{light} consiste en un módulo de aprendizaje y otro de clasificación. El módulo de clasificación se utiliza para aplicar el modelo aprendido, con el módulo de aprendizaje, a nuevos ejemplos. El formato del comando para ejecutar el módulo de aprendizaje es el siguiente:

svm_learn [opciones] datos_de_entrenamiento archivo_modelo

Las diferentes opciones se pueden consultar en [56]. En este apartado solamente se explicarán aquellas que se utilizarán en este TFG:

- Opciones de Kernel:

- -t int (tipo de función Kernel).
 - 0: Función lineal (predeterminado).
 - 2: Función de base radial $e^{-\gamma*\|a-b\|^2}$
- -g float
 - Parámetro gamma en función de base radial.

En el apéndice de [58] se proporciona una descripción más detallada de los parámetros y cómo se vinculan con los algoritmos respectivos.

El archivo de entrada *datos_de_entrenamiento* contiene las muestras de ejemplo que se van a utilizar para entrenar. Centrándonos en nuestro trabajo, se deberá crear un archivo de datos de entrenamiento por cada camareta o compartimento que se quiera tener en cuenta en la localización.

Las primeras líneas pueden contener comentarios y se ignoran si comienzan con #. Cada una de las siguientes líneas representan un ejemplo de entrenamiento y tienen el siguiente formato:

```
<line>. =. <target> <feature>:<value> <feature>:<value> ... <feature>:<value> # <info>
<target>. =. +1 | -1 | 0 | <float>
<feature>. =. <integer> | "qid"
<value>. =. <float>
<info>. =. <string>
```

El formato de la Figura 3-5, que muestra el archivo que vuelca nuestro escáner, cumple con este formato. El valor objetivo *<target>* y cada uno de los conjuntos de características *<feature>* y valor *<value>* (ejemplo: 2:-53) están separados por un espacio. Los conjuntos de característica y valor deben ordenarse aumentando el número de la función característica, como se puede ver en la Figura 3-5. Las características con valor cero se pueden omitir (ejemplo: 9:0). La cadena *<info>* se puede usar para pasar la información adicional al Kernel.

En el modo de clasificación binaria, el valor objetivo denota la clase del ejemplo:

- +1 Valor positivo. Se corresponde con la clase o categoría que se quiere clasificar.
- -1 Valor negativo. Corresponde con el resto de las clases o categorías.

Es decir, cuando creamos los archivos *datos_de_entrenamiento* por cada camareta o compartimento, las muestras que correspondan a dicha camareta o compartimento deberán llevar un +1 y las del resto un -1.

Para más información sobre el modo de regresión y el modo *ranking* en el módulo de aprendizaje, dirigirse a [56]. En la Figura 3-31 se muestra un ejemplo de lo que nos devolvería la ejecución del módulo de aprendizaje.

```

Terminal
alumno@ubuntu:~/svm_light$ ./svm_learn prueba01/datosG3.txt prueba01/modeloG3.txt
Scanning examples...done
Reading examples into memory...100..200..300..400..500..600..700..800..900..1000..1100
..1200..1300..1400..1500..1600..1700..1800..1900..2000..OK. (2000 examples read)
Setting default regularization parameter C=0.0001
Optimizing.....
.....
.....
.....
.....done. (425 iterations)
Optimization finished (179 misclassified, maxdiff=0.00085).
Runtime in cpu-seconds: 0.06
Number of SV: 1066 (including 1058 at upper bound)
L1 loss: loss=603.29269
Norm of weight vector: |w|=0.15413
Norm of longest example vector: |x|=173.15311
Estimated VCdim of classifier: VCdim<=713.23848
Computing XiAlpha-estimates...done
Runtime for XiAlpha-estimates in cpu-seconds: 0.00
XiAlpha-estimate of the error: error<=53.05% (rho=1.00,depth=0)
XiAlpha-estimate of the recall: recall=>46.90% (rho=1.00,depth=0)
XiAlpha-estimate of the precision: precision=>46.95% (rho=1.00,depth=0)
Number of kernel evaluations: 27483
Writing model file...done
alumno@ubuntu:~/svm_light$
    
```

Figura 3-31 Ejemplo de ejecución del módulo de aprendizaje

En todos los modos (binario, regresión y *ranking*), el resultado de *svm_learn* es el modelo que se aprende de *datos_de_entrenamiento*. El modelo se guarda en *archivo_modelo*. Para hacer predicciones sobre los ejemplos de prueba, *svm_classify* lee este archivo. El formato de *svm_classify* es el siguiente:

svm_classify [opciones] datos_de_evaluación archivo_modelo archivo_de_salida

Donde las opciones del comando se pueden consultar en la referencia [56].

Los datos de evaluación se dan en el mismo formato que los datos de entrenamiento (normalmente con 0 como valor objetivo, aunque no es necesario). Para todas las muestras en *datos_de_evaluación*, los valores pronosticados se escriben en *archivo_de_salida*. Hay una línea por cada muestra en *archivo_de_salida* que contiene el valor de la función de decisión para esa muestra. Para la clasificación binaria, el signo de este valor determina la clase predicha.

En la Figura 3-32 se muestra un ejemplo de lo que nos devolvería la ejecución del módulo de clasificación.

```

Terminal
alumno@ubuntu:~/svm_light$ ./svm_classify prueba01/capturaG3.txt prueba01/modeloG3.txt prueba01/salidaG3.txt
Reading model...OK. (1066 support vectors read)
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Accuracy on test set: 90.00% (180 correct, 20 incorrect, 200 total)
Precision/recall on test set: 100.00%/90.00%
alumno@ubuntu:~/svm_light$
    
```

Figura 3-32 Ejemplo de ejecución del módulo de clasificación

3.3.6.4 Descripción del escenario del *Experimento 6*

El escenario del *Experimento 6* queda definido de la siguiente manera:

- Se ubicaron las cuatro RPis operando en modo radiobaliza, en la disposición mostrada en la Figura 3-33:
 - RPi2 y RPi3: Se dejaron situadas sobre una estantería a 1,5m. de altura. Ambas estaban conectadas a la red eléctrica.

- RPi4: Se colocó a una altura de 0.6 m, sobre una mesilla, alimentada por un cable conectado a la red eléctrica.
- RPi5: Se dejó apoyada en el suelo, alimentada por una batería externa.
- Dichas RPis se colocaron en posición horizontal, transmitiendo con un periodo de 500 ms.
- Las camaretas G2 y G3 marcadas en la Figura 3-33 serán las zonas que analicemos en este experimento.

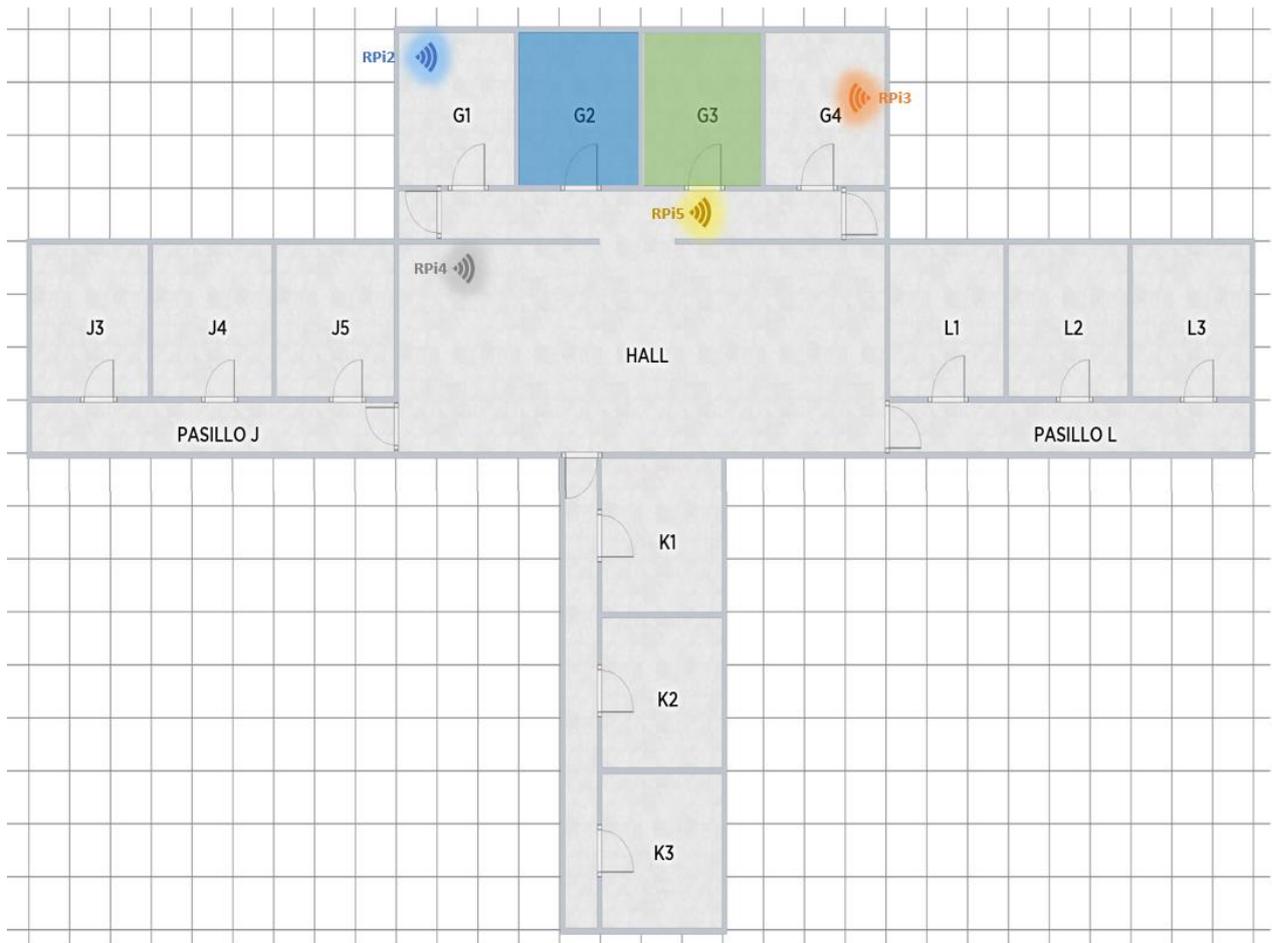


Figura 3-33 Disposición Experimento 6

El procedimiento que se utilizó para tomar muestras fue el siguiente:

- Se tomaron datos durante aproximadamente 21 minutos con la RPi1, ejecutando el script `scansvm.sh`, que devuelve los datos en el formato de la Figura 3-5. Mientras se estaba escaneando, nos desplazábamos por toda la camareta, cubriendo todos los espacios del plano, y la altura entre uno y dos metros con respecto al suelo. Es importante recordar que se imprime una muestra o línea por segundo, por lo que tendremos aproximadamente 1260 líneas por cada camareta.
- La extensión establecida para los archivos de salida del script `scansvm.sh` es “.ori”, lo que nos servirá para identificar estos archivos en el resto de los scripts:
 - `./scansvm.sh > G2.ori`

- Se comprueba que no haya ningún error en los archivos y se borran algunas líneas iniciales y finales. De esa forma, descartamos posibles errores que se pueden dar al inicio y al terminar el proceso de captura.
- De las muestras tomadas por camareta, 1000 irán destinadas para el aprendizaje y 200 diferentes para la clasificación.

Tras obtener los archivos indicados, se guardan en un directorio donde también hayamos guardado los ejecutables *svm_learn* y *svm_classify*, y el script *apreneval01.sh* que se muestra en el Anexo IV. Una vez se tengan todos estos archivos en el directorio y compilados, se accede al directorio con el terminal y se ejecuta el script:

```
./apreneval01.sh > R.txt
```

A continuación, se explica paso a paso el funcionamiento de dicho script:

1. Se editan los ficheros *.ori*:
 - a. Se desordenan aleatoriamente los datos. Al movernos por la camareta, si dividimos el archivo en dos para datos de entrenamiento y de evaluación, puede que haya zonas que no queden reflejadas en estos archivos. Para ello se ejecuta:
 - i. *shuf \$COM.ori > \$COM.dat* (*\$COM* y *\$NUM* son dos variables utilizadas referirnos a los distintos compartimentos (*G2* y *G3*, en este caso) dentro de un bucle *for*).
 - b. Nos quedamos con las 1200 primeras muestras, eliminando el resto:
 - i. *sed -i '1201,\$d' \$COM.dat*
2. Se crean los archivos *datos_de_entrenamiento*:
 - a. Se crea un archivo por cada camareta con +1.
 - i. *sed '1001,1200d' \$COM.dat > \$COM.pos.temp*
 - b. Se crea un archivo por cada camareta con -1.
 - i. *sed 's/+1/-1/' \$COM.pos.temp > \$COM.neg.temp*
 - c. Se crean finalmente *datos_de_entrenamiento* juntando las muestras de la camareta que queramos entrenar con el archivo *.pos.temp* y el resto de las camaretas, el archivo *.neg.temp*.
 - i. *cat G2.pos.temp G3.neg.temp > G2.entrena*
 - ii. *cat G3.pos.temp G2.neg.temp > G3.entrena*
3. Se crean los archivos *datos_de_evaluacion*:
 - a. Con el siguiente comando nos quedamos con las 200 últimas muestras con +1:
 - i. *sed '1,1000d' \$COM.dat > \$COM.evalua*
4. Se ejecuta el módulo de aprendizaje:
 - a. Uno para Kernel lineal:
 - i. *./svm_learn \$NUM.entrena \$NUM-LINEAL.model*
 - b. Cinco diferentes para Kernel radial, variando gamma. Al no tener claro qué valor de gamma es el óptimo para nuestras capturas, se prueban gamma = 0,001; 0,01; 0,1; 1 y 10:
 - i. *./svm_learn -t 2 -g \$gamma \$NUM.entrena \$NUM-RADIAL-\$gamma.model*
5. Se ejecuta el módulo de clasificación:

- a. Se comparan los *datos_de_evaluación* con cada uno de los modelos obtenidos a partir de los *datos_de_aprendizaje*:
 - i. `./svm_classify $PRU.evalua $NUM-LINEAL.model $PRU-$NUM-LINEAL.temp > $PRU-$NUM-LINEAL.result`
 - ii. `./svm_classify $PRU.evalua $NUM-RADIAL-$gamma.model $PRU-$NUM-RADIAL-$gamma.temp > $PRU-$NUM-RADIAL-$gamma.result`
6. Los resultados del script se imprimen en *R.txt*. Como vienen muchos datos repetidos, con el siguiente comando borramos las líneas duplicadas:
 - i. `cat R.txt | sort | uniq > RESULTADOS.txt`

En la Figura 3-34 se muestra un ejemplo de lo que visualizamos al abrir el archivo *RESULTADOS.txt*. Los resultados que se consideran correctos son aquellos que han salido positivo tras ejecutar el módulo de clasificación y los incorrectos, negativos. El hecho de que un resultado salga positivo quiere decir que la muestra que se ha comparado, de los datos de evaluación, queda dentro del espacio delimitado por el hiperplano (vector soporte), mientras que, si sale negativo, quiere decir que se encuentra fuera. En valor absoluto, cuanto mayor sea el número, más alejado queda del hiperplano. Por lo tanto, la precisión del conjunto de evaluación indica el porcentaje de muestras que se consideran de la clase perteneciente al modelo con la que se la está comparando.

```
G2-G2-LINEAL.result:Accuracy on test set: 91.50% (183 correct, 17 incorrect, 200 total)
G2-G2-RADIAL-0.001.result:Accuracy on test set: 92.50% (185 correct, 15 incorrect, 200 total)
G2-G2-RADIAL-0.01.result:Accuracy on test set: 94.50% (189 correct, 11 incorrect, 200 total)
G2-G2-RADIAL-0.1.result:Accuracy on test set: 92.00% (184 correct, 16 incorrect, 200 total)
G2-G2-RADIAL-10.result:Accuracy on test set: 98.50% (197 correct, 3 incorrect, 200 total)
G2-G2-RADIAL-1.result:Accuracy on test set: 94.50% (189 correct, 11 incorrect, 200 total)
G2-G3-LINEAL.result:Accuracy on test set: 8.50% (17 correct, 183 incorrect, 200 total)
G2-G3-RADIAL-0.001.result:Accuracy on test set: 7.50% (15 correct, 185 incorrect, 200 total)
G2-G3-RADIAL-0.01.result:Accuracy on test set: 5.50% (11 correct, 189 incorrect, 200 total)
G2-G3-RADIAL-0.1.result:Accuracy on test set: 8.00% (16 correct, 184 incorrect, 200 total)
```

Figura 3-34 Captura de un segmento del archivo *RESULTADOS.txt*

Los datos obtenidos en *RESULTADOS.txt* se han representado en las tablas de la Figura 3-35 de la siguiente manera:

- En la primera columna se indica a qué camareta pertenecen los datos de evaluación de esa fila.
- En la primera fila se indica el modelo (generado con los datos de entrenamiento) de la camareta con la que estamos comparando o clasificando.
- El título de la tabla indica qué opción de Kernel se ha escogido para el entrenamiento y generación de los modelos.
- Los valores indicados en la tabla están expresados en porcentajes.
- En la Figura 3-36 se muestra la leyenda de la escala de colores empleada en los resultados. Los valores donde se comparan los datos de evaluación y los datos de entrenamiento de la misma camareta están marcados con la gama de color verde y el resto con la gama de color gris.
- Para saber si obtenemos buenos resultados, lo ideal sería tener 100% en las celdas verdes y 0% en las celdas grises.

Kernel lineal:

	G2	G3
G2	91,50	8,50
G3	10,00	90,00

Kernel radial (gamma 0,001):

	G2	G3
G2	92,50	7,50
G3	9,50	90,50

Kernel radial (gamma 0,01):

	G2	G3
G2	94,50	5,50
G3	10,50	89,50

Kernel radial (gamma 0,1):

	G2	G3
G2	92,00	8,00
G3	9,50	90,50

Kernel radial (gamma 1):

	G2	G3
G2	94,50	5,50
G3	62,00	17,50

Kernel radial (gamma 10):

	G2	G3
G2	98,50	1,50
G3	82,50	38,00

Figura 3-35 Resultados de la precisión del conjunto de clasificación. Experimento 6

> 20	90-100
10-20	80-89
4-9	60-79
1-3	0-59
0	

Figura 3-36 Leyenda con escala de colores

A partir de estos resultados, podemos decir lo siguiente:

- La precisión del conjunto de clasificación en general es buena. Excepto en los casos donde *gamma* es igual a 1 o a 10, la precisión se encuentra en torno al 90%.
- Se puede considerar que el mejor resultado se obtiene cuando se usa la opción de Kernel radial, con *gamma* igual a 0,01, ya que la suma de errores tiene el valor mínimo, con respecto al resto de resultados.

4 PRUEBA Y VALIDACIÓN A BORDO DEL PATRULLERO “TABARCA”

En el capítulo anterior hemos ido avanzando desde la configuración de los dispositivos, al fin último que nos atañe, la obtención de la localización a partir de algoritmos SVM. Una vez detallados todos los pasos a seguir para ello y de realizar una pequeña prueba que nos mostrase la eficiencia de este sistema (*Experimento 6*), en este capítulo validaremos nuestro sistema de localización a bordo del patrullero “Tabarca”, como se marca en los objetivos de este TFG.

Para la validación del sistema en este nuevo escenario se va a seguir el mismo modelo aplicado en el *Experimento 6*. Nuestro objetivo en este caso es abarcar el mayor número de compartimentos que el tiempo y el número de radiobalizas nos permitan. Hay que tener en cuenta que, por cada local o compartimento que queramos reconocer con nuestro sistema, necesitaremos alrededor de 20 minutos; y que con las cinco RPis de las que disponíamos para realizar el TFG, no podíamos cubrir todo el buque. Es por ello por lo que nos hemos centrado en una zona del barco. Teniendo en cuenta estas limitaciones, se ha intentado cubrir compartimentos situados en diferentes cubiertas, algo que hasta ahora no se había tenido en cuenta (siempre habíamos trabajado en la misma planta). En total se cubren 11 compartimentos situados en 4 cubiertas diferentes.

Para entender mejor cómo se implementa un sistema de localización en interiores en un edificio, buque o cualquier otro recinto cubierto y compartimentado, debemos diferenciar bien entre la fase de calibración y la de validación y puesta en marcha. A continuación, se describen los pasos para cada una de las fases, pasos que se han considerado necesarios para cualquier sistema de localización en interiores similar:

- Fase de calibración:
 - Estudio del entorno (Dimensiones, compartimentación, cobertura, contaminación por RF, material de construcción, etc.)
 - Determinar la colocación de las radiobalizas y hacer un estudio rápido de la señal recibida en los diferentes compartimentos. Realizar las modificaciones necesarias en la colocación de las radiobalizas.
 - Se toman muestras de la señal recibida por cada compartimento que queramos que nuestro sistema reconozca.
 - Se realiza una base de datos a partir de los ficheros generados.
- Fase de validación y puesta en marcha:

- Uso de un software o ejecutable, apoyado en la base de datos obtenida en la fase de calibración, que nos devuelve la posición. El conjunto de todo ello debe conformar un sistema práctico y entendible para su futuro usuario.
- Se toman muestras en los diferentes compartimentos y se introducen en el sistema para comprobar la eficiencia de éste.

Estos pasos describen de forma general la idea básica de cómo implementar el sistema de localización. A continuación, se detallarán los pasos seguidos en la validación de este trabajo, acorde con los pasos anteriores:

- El primer día se realizó un reconocimiento de los compartimentos del buque, tanto, realizando un recorrido por todo el buque, como consultando los planos del destino de seguridad interior. Al no disponer de planos en formato digital, hubo que diseñar los nuestros propios de forma esquemática (no están realizados a escala), a partir de los planos del destino de seguridad interior. Los planos que en este capítulo se muestran del “Tabarca” sirven para situarnos y tener una visión de las zonas en las que estamos trabajando.
- Se determinó la posición de las radiobalizas. Para ello se siguieron los criterios aprendidos en la fase de experimentación. Por otro lado, para demostrar que podemos escoger diferentes distribuciones de balizas y que el sistema sigue dando buenos resultados, se realizaron dos pruebas diferentes donde cambiamos la distribución de éstas. Una vez colocadas las radiobalizas, se realizó un reconocimiento, con el móvil funcionando de escáner, comprobando la cobertura de las diferentes balizas en todos los compartimentos señalados.
- Posteriormente, se procedió a la obtención de muestras, compartimento por compartimento. Para ello, se siguió el mismo procedimiento que en el *Experimento 6* con alguna pequeña modificación: en lugar de tomar aproximadamente 1200 muestras, se tomaron 1000, de las cuales 900 fueron destinadas a la fase de calibración como datos de entrenamiento y 100 fueron utilizadas en la fase de validación como datos de evaluación. De esta manera, ahorramos tiempo realizando dos pasadas para tomar muestras.
- Una vez se tienen los ficheros con las muestras, se procesan dichas muestras de la misma manera que se hizo en el *Experimento 6*. Para ello hubo que modificar el script utilizado en este experimento, de manera que tuviera en cuenta que, ahora, el número de muestras era de 1000 y que ya no teníamos 2 compartimentos diferentes, sino 11. En el Anexo V se puede consultar el script utilizado (*apreneval02.sh*).
- Igualmente, los resultados obtenidos han sido representados del mismo modo que en el *Experimento 6*, solo que ahora comparamos muchos más compartimentos.

4.1 Prueba 1

A continuación, se muestran en la Figura 4-1, Figura 4-2, Figura 4-3 y Figura 4-4 las diferentes cubiertas del “Tabarca” donde hemos trabajado. Se le ha dado color a aquellos compartimentos que hemos tenido en cuenta a la hora de validar el sistema de localización. A dichos compartimentos se les ha asignado una numeración propia para facilitar el manejo de datos y ficheros. Se les nombra con las letras *COM* y seguidamente, un número del *01* al *11*. En tablas donde se muestran los resultados hemos utilizado la misma nomenclatura para nombrar los compartimentos. Las radiobalizas se han colocado lo más separadas posibles.

En la Tabla 4-1, Tabla 4-2, Tabla 4-3, Tabla 4-4, Tabla 4-5 y Tabla 4-6, se muestran los resultados de la evaluación. Lo ideal sería tener 100% en las celdas verdes (la diagonal que va desde la primera celda a la última) y 0% en las celdas grises (el resto de las celdas).

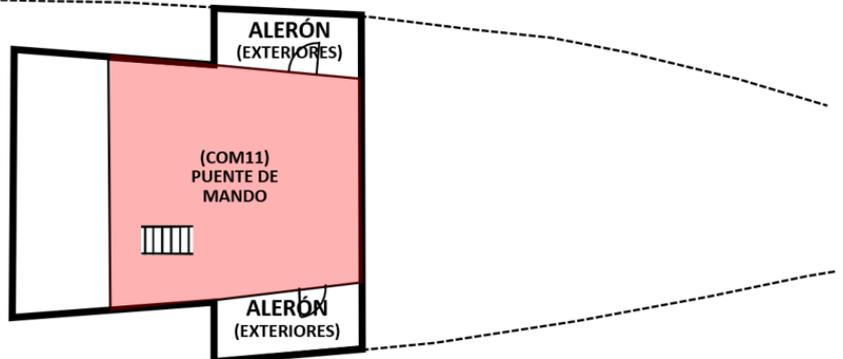


Figura 4-1 Cubierta puente. Prueba 1

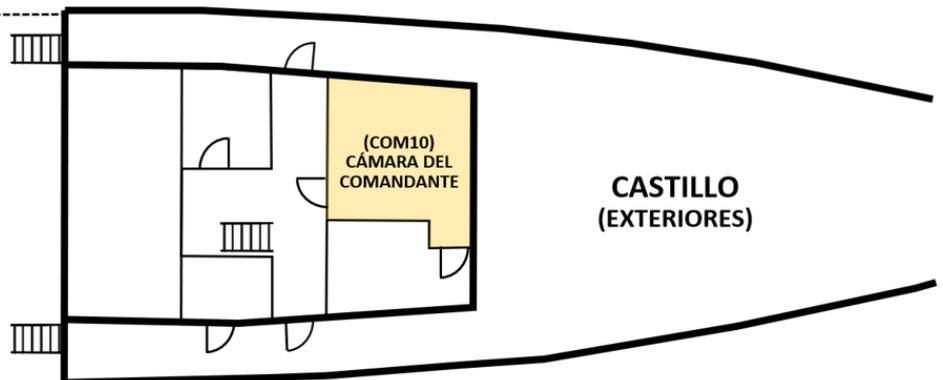


Figura 4-2 Cubierta castillo. Prueba 1

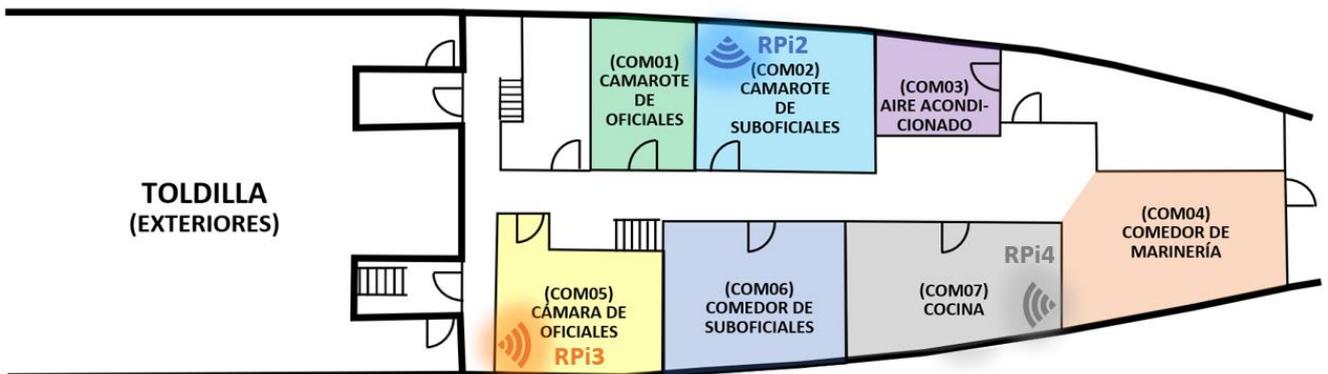


Figura 4-3 Cubierta principal. Prueba 1

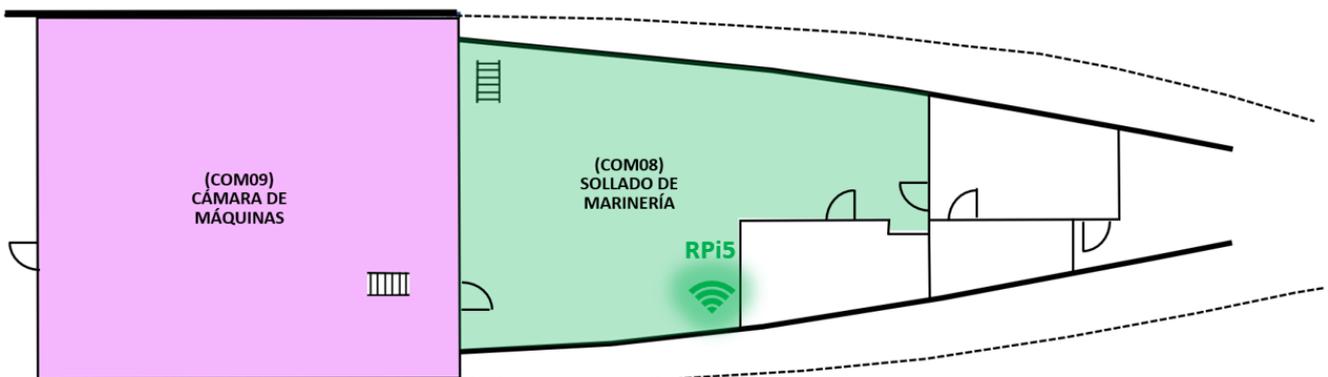


Figura 4-4 Cubierta baja. Prueba 1

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	0	0	0	0	0	0	0	0	0	0	0
COM02	0	91	0	0	0	8	0	0	0	0	0
COM03	0	0	0	0	0	0	0	0	0	0	0
COM04	0	0	0	0	0	0	1	0	0	0	0
COM05	0	0	0	0	56	0	0	0	0	0	0
COM06	0	0	0	0	0	18	1	0	0	0	0
COM07	0	0	0	0	0	0	99	0	0	0	0
COM08	1	0	0	0	0	0	0	62	2	0	0
COM09	0	0	0	0	0	0	0	0	84	0	0
COM10	0	0	0	0	0	0	0	0	0	0	0
COM11	0	0	0	0	0	0	0	0	5	0	0

Tabla 4-1 Resultados prueba 1. Kernel lineal

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	85	0	0	3	0	0	0	0	0	2	0
COM02	0	92	0	0	0	8	0	0	0	0	0
COM03	0	0	95	2	0	0	0	0	0	0	1
COM04	2	0	1	93	0	0	2	0	0	0	0
COM05	12	0	4	5	68	0	0	0	0	1	0
COM06	4	1	0	1	0	93	0	0	0	0	0
COM07	0	0	0	0	0	0	99	0	0	0	0
COM08	0	0	11	1	0	0	0	71	0	0	1
COM09	0	0	0	0	2	0	0	0	92	0	4
COM10	0	0	0	0	0	0	0	0	0	86	9
COM11	0	0	8	0	0	0	0	0	7	8	70

Tabla 4-2 Resultados prueba 1. Kernel radial (gamma 0,001)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	90	0	0	2	0	0	0	0	0	1	0
COM02	0	93	0	0	0	4	0	0	0	0	0
COM03	0	0	88	1	0	0	0	1	0	0	2
COM04	5	0	0	87	0	0	2	0	0	0	0
COM05	12	0	2	4	73	0	0	0	0	0	0
COM06	3	1	0	1	0	91	0	0	0	0	0
COM07	0	0	0	0	0	0	99	0	0	0	0
COM08	0	0	9	0	0	0	0	80	0	1	0
COM09	0	0	0	0	2	0	0	0	94	0	2
COM10	0	0	0	0	0	0	0	0	0	86	9
COM11	0	0	5	0	0	0	0	1	6	7	71

Tabla 4-3 Resultados prueba 1. Kernel radial (gamma 0,01)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	77	0	0	2	1	0	0	0	0	0	0
COM02	0	93	0	0	0	1	0	0	0	0	0
COM03	0	0	86	1	0	0	0	1	0	0	2
COM04	4	0	0	85	0	0	2	0	0	0	0
COM05	12	0	1	4	54	0	0	0	0	0	0
COM06	3	1	0	1	0	76	0	0	0	0	0
COM07	0	0	0	0	0	0	97	0	0	0	0
COM08	0	0	8	1	0	0	0	60	0	1	0
COM09	0	0	0	0	1	0	0	0	76	0	1
COM10	0	0	0	0	0	0	0	0	0	82	3
COM11	0	0	3	0	0	0	0	0	1	4	54

Tabla 4-4 Resultados prueba 1. Kernel radial (gamma 0,1)

SISTEMA DE LOCALIZACIÓN DE PERSONAL A BORDO BASADO EN
TÉCNICAS DE APRENDIZAJE AUTOMÁTICO Y BLUETOOTH LOW ENERGY

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	47	0	0	2	0	0	0	0	0	0	0
COM02	0	80	0	0	0	1	0	0	0	0	0
COM03	0	0	52	0	0	0	0	1	0	0	0
COM04	5	0	0	78	0	0	2	0	0	0	0
COM05	9	0	0	1	1	0	0	0	0	0	0
COM06	2	0	0	1	0	37	0	0	0	0	0
COM07	0	0	0	0	0	0	90	0	0	0	0
COM08	0	0	6	0	1	0	0	0	0	0	0
COM09	0	0	0	0	0	0	0	0	9	0	0
COM10	0	0	0	0	0	0	0	0	0	0	0
COM11	0	0	0	0	0	0	0	0	0	0	1

Tabla 4-5 Resultados prueba 1. Kernel radial (gamma 1)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	38	0	0	3	2	0	0	0	0	0	0
COM02	0	63	0	0	0	0	0	0	0	0	0
COM03	0	0	37	0	0	0	0	1	0	0	0
COM04	3	0	0	75	0	0	1	0	0	0	0
COM05	8	0	0	0	1	0	0	0	0	0	0
COM06	1	0	0	1	0	28	0	0	0	0	0
COM07	0	0	0	0	0	0	80	0	0	0	0
COM08	0	0	4	0	1	0	0	0	0	0	0
COM09	0	0	0	0	0	0	0	0	3	0	0
COM10	0	0	0	0	0	0	0	0	0	0	0
COM11	0	0	0	0	0	0	0	0	0	0	0

Tabla 4-6 Resultados prueba 1. Kernel radial (gamma 10)

> 20	90-100
10-20	80-89
4-9	60-79
1-3	0-59
0	

Figura 4-5 Leyenda con escala de colores

Como conclusión de los resultados obtenidos, podemos destacar lo siguiente:

- Cuando utilizamos Kernel lineal, obtenemos resultados bastante malos, ya que cuando estamos comparando los datos de evaluación con los de aprendizaje del mismo compartimento, obtenemos un 0% de aciertos en muchos de los casos. Con Kernel radial ocurre lo mismo cuando se le dan a *gamma* los valores más altos, 1 y 10. Cuando *gamma* vale 0,1, la precisión de acierto no baja del 50%, pero tampoco obtenemos una buena precisión. Los mejores resultados se obtienen cuando *gamma* vale 0,01 y 0,001, donde la precisión más baja es del 70%.
- Por otro lado, los casos contrarios donde el porcentaje obtenido debería haber sido del 0%, ya que los datos de evaluación se comparan con el modelo de otro compartimento diferente, tan solo llega al 12% en los casos en los que *gamma* vale 0,01 y 0,001.
- Teniendo en cuenta estos análisis de los resultados, a la hora de implementar nuestro sistema de localización en interiores, nuestra base de datos debería tener los modelos generados en el módulo de aprendizaje donde se ha aplicado Kernel radial, con *gamma* igual a 0,01 y 0,001.

4.2 Prueba 2

A continuación, se muestra la disposición de las radiobalizas en la prueba 2 y los resultados obtenidos con esta nueva disposición.

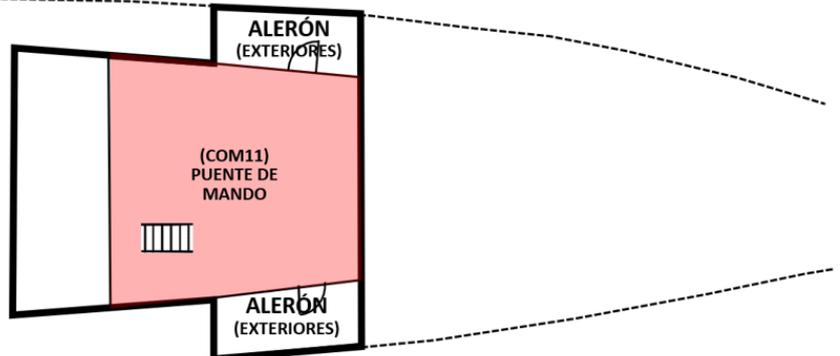


Figura 4-6 Cubierta puente. Prueba 2

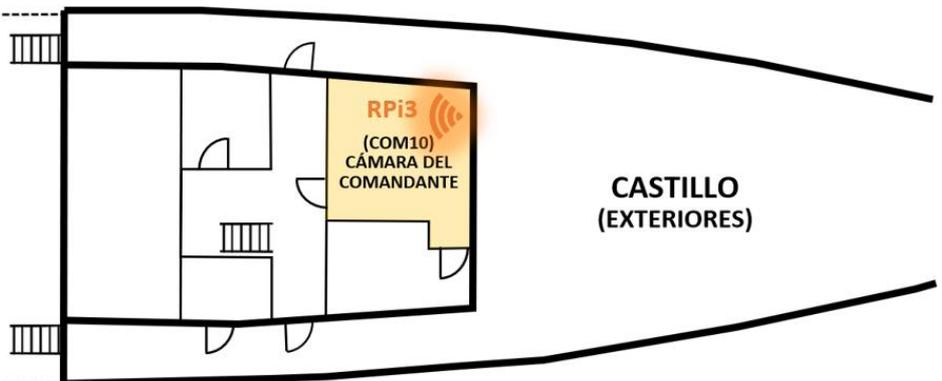


Figura 4-7 Cubierta castillo. Prueba 2

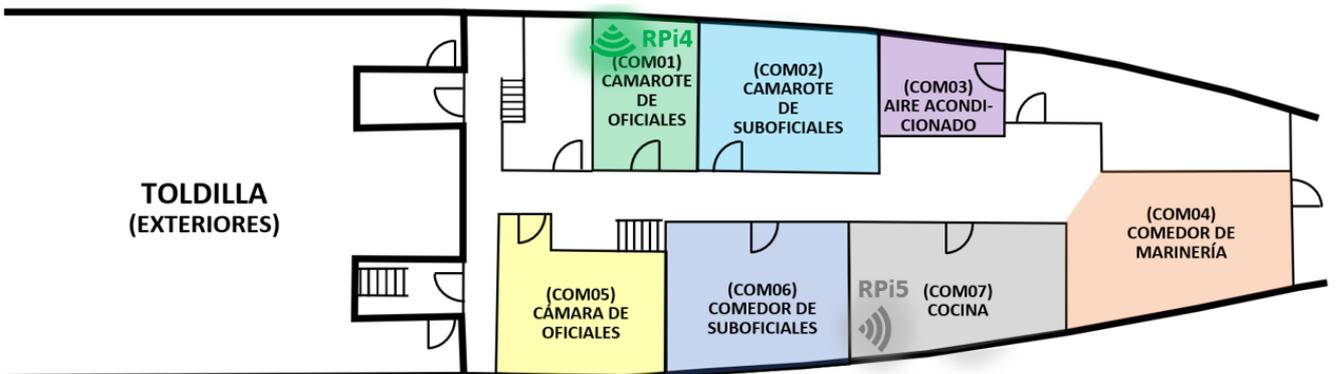


Figura 4-8 Cubierta principal. Prueba 2

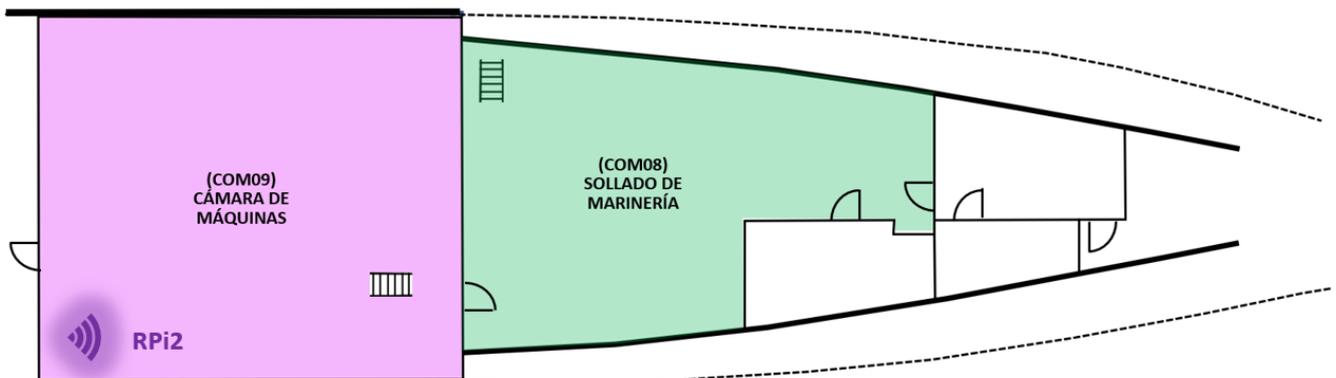


Figura 4-9 Cubierta baja. Prueba 2

SISTEMA DE LOCALIZACIÓN DE PERSONAL A BORDO BASADO EN
TÉCNICAS DE APRENDIZAJE AUTOMÁTICO Y BLUETOOTH LOW ENERGY

Esta nueva distribución empleada en la *prueba 2* difiere de la distribución de la *prueba 1* en que las radiobalizas están más separadas entre sí, localizadas en tres cubiertas diferentes, mientras que en la *prueba 1* solo se localizaban en dos. Veremos si esto afecta a los resultados en las siguientes tablas.

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	99	6	0	0	0	0	0	0	0	0	0
COM02	0	0	0	0	0	0	0	0	0	0	0
COM03	0	0	0	0	0	0	0	0	0	0	0
COM04	0	0	0	0	0	0	1	0	0	0	0
COM05	0	0	0	0	0	0	0	0	0	0	0
COM06	0	0	0	0	0	0	0	0	0	0	0
COM07	0	0	0	0	0	0	98	0	0	0	0
COM08	0	0	0	0	0	0	0	0	0	0	0
COM09	0	0	0	0	0	0	0	0	100	0	0
COM10	0	0	0	0	0	0	0	0	0	100	0
COM11	0	0	0	0	0	0	0	0	0	0	0

Tabla 4-7 Resultados prueba 2. Kernel lineal

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	100	1	0	0	0	0	0	0	0	0	0
COM02	0	99	0	0	0	1	0	0	0	0	0
COM03	0	0	99	1	0	0	0	0	0	0	0
COM04	0	0	19	81	0	1	1	0	0	0	0
COM05	0	0	2	0	77	0	0	18	0	0	0
COM06	0	0	3	0	0	94	0	0	0	0	0
COM07	0	0	0	3	0	0	98	0	0	0	0
COM08	0	0	0	0	14	0	0	88	0	0	0
COM09	0	0	0	0	0	0	0	0	100	0	0
COM10	0	0	0	0	0	0	0	0	0	100	0
COM11	0	0	0	0	0	0	0	0	0	0	96

Tabla 4-8 Resultados prueba 2. Kernel radial (gamma 0,001)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	99	0	0	0	0	0	0	0	0	0	0
COM02	0	99	0	0	0	1	0	0	0	0	0
COM03	0	0	98	1	0	0	0	0	0	0	0
COM04	0	0	17	79	0	1	1	0	0	0	0
COM05	0	0	0	0	80	0	0	18	0	0	0
COM06	0	0	2	0	0	95	0	0	0	0	0
COM07	0	0	0	2	0	0	97	0	0	0	0
COM08	0	0	0	0	11	0	0	89	0	0	0
COM09	0	0	0	0	0	0	0	0	100	0	0
COM10	0	0	0	0	0	0	0	0	0	100	0
COM11	0	0	0	0	0	0	0	0	0	0	96

Tabla 4-9 Resultados prueba 2. Kernel radial (gamma 0,01)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	88	0	0	0	0	0	0	0	0	0	0
COM02	0	95	0	0	0	0	0	0	0	0	0
COM03	0	0	97	0	0	0	0	0	0	0	0
COM04	0	0	18	79	0	1	1	0	0	0	0
COM05	0	0	1	0	74	0	0	18	0	0	0
COM06	0	0	2	0	0	83	0	0	0	0	0
COM07	0	0	0	2	0	0	96	0	0	0	0
COM08	0	0	0	0	10	0	0	83	0	0	0
COM09	0	0	0	0	0	0	0	0	97	0	0
COM10	0	0	0	0	0	0	0	0	0	97	0
COM11	0	0	0	0	0	0	0	0	0	0	90

Tabla 4-10 Resultados prueba 2. Kernel radial (gamma 0,1)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	17	0	0	0	0	0	0	0	0	0	0
COM02	0	81	0	0	0	0	0	0	0	0	0
COM03	0	0	92	0	0	0	0	0	0	0	0
COM04	0	0	16	67	0	0	1	0	0	0	0
COM05	0	0	0	0	20	0	0	7	0	0	0
COM06	0	0	2	0	0	0	0	0	0	0	0
COM07	0	0	0	2	0	0	66	0	0	0	0
COM08	0	0	0	0	5	0	0	27	0	0	0
COM09	0	0	0	0	0	0	0	0	79	0	0
COM10	0	0	0	0	0	0	0	0	0	59	0
COM11	0	0	0	0	0	0	0	0	0	0	25

Tabla 4-11 Resultados prueba 2. Kernel radial (gamma 1)

	COM01	COM02	COM03	COM04	COM05	COM06	COM07	COM08	COM09	COM10	COM11
COM01	5	0	0	0	0	0	0	0	0	0	0
COM02	0	77	0	0	0	0	0	0	0	0	0
COM03	0	0	83	0	0	0	0	0	0	0	0
COM04	0	0	12	48	0	0	0	0	0	0	0
COM05	0	0	0	0	3	0	0	1	0	0	0
COM06	0	0	1	0	0	0	0	0	0	0	0
COM07	0	0	0	2	0	0	55	0	0	0	0
COM08	0	0	0	0	1	0	0	7	0	0	0
COM09	0	0	0	0	0	0	0	0	73	0	0
COM10	0	0	0	0	0	0	0	0	0	34	0
COM11	0	0	0	0	0	0	0	0	0	0	6

Tabla 4-12 Resultados prueba 2. Kernel radial (gamma 10)

> 20	90-100
10-20	80-89
4-9	60-79
1-3	0-59
0	

Figura 4-10 Leyenda con escala de colores

Tras los datos mostrados en estas tablas, las conclusiones son las siguientes:

- Kernel lineal y Kernel radial con valores de *gamma* de 1 y 10 vuelven a dar malos resultados. Por otro lado, esta vez, cuando *gamma* tiene el valor de 0,1 se obtienen buenos resultados.
- En los casos donde *gamma* vale 0,01 y 0,001 obtenemos, incluso, mejores resultados que en la *prueba 1*, siendo el 77% el valor de precisión más bajo. Este valor mínimo coincide en ambas pruebas con el *COM05*, la cámara de oficiales, por lo que podemos afirmar que este compartimento es el más conflictivo.
- Con esta nueva disposición, al colocar la RPi3 en la cámara del comandante, los compartimentos *COM10* y *COM11* se distinguen con mayor precisión.
- Con respecto al sollado de marinería, también se obtiene que la precisión no es de las mejores en ambas pruebas. Como se concluyó en el *Experimento 4*, las zonas de mayor extensión pueden dar problemas.
- En esta prueba, el valor máximo de precisión en los casos donde se comparan compartimentos diferentes asciende al 18%. Vuelve a coincidir que en la cámara de oficiales y el sollado de marinería los compartimentos que dan problemas, ya que lo que nos viene a decir ese resultado, es que, al comparar los datos de evaluación de la cámara de oficiales con el sollado de marinería, el 18% nos afirma que estamos en este último.

5 CONCLUSIONES Y LÍNEAS FUTURAS

Tras la realización del estudio y validación de la implementación del sistema de localización de personal a bordo de buques de la Armada, objeto y motivación de este Trabajo Fin de Grado, a continuación, se detallan las conclusiones obtenidas:

- Según los resultados obtenidos en el capítulo de validación y a pesar de la fluctuación de la señal Bluetooth que podemos apreciar en los experimentos previos a la validación, se puede afirmar que la tecnología *Bluetooth Low Energy* es viable para el desarrollo de sistemas de localización en interiores. Además, cuenta con la ventaja de ser una tecnología de bajo coste y consumo energético.
- La placa Raspberry Pi es una buena herramienta de estudio y ha demostrado ser bastante versátil a la hora de trabajar con ella en este TFG, sin presentar limitaciones importantes. Sin embargo, algunos factores no la hacen apta para el uso que queremos darle. Por un lado, no nos permite aprovechar el bajo consumo energético que nos presenta la tecnología *Bluetooth Low Energy*, ya que, como se comprobó al realizar los experimentos, cualquiera de las baterías externas (de 10000 mAh aproximadamente) con las que trabajamos, se consumían en menos de 12 horas. Las placas que trabajen en modo radiobalizas se pueden conectar a la instalación eléctrica del buque, sin embargo, esta situación resulta inviable para las placas que debería portar el personal, operando en modo escáner. Por otro lado, el tamaño de la propia placa, sumado al de la batería externa, hacen de la Raspberry Pi un dispositivo poco recomendable para implementar en este tipo de sistemas.
- No obstante, *Bluetooth Low Energy* es una tecnología bastante universal y extendida, por lo que se pueden encontrar fáciles soluciones en el mercado para implementar el sistema planteado en este TFG en otro tipo de dispositivos más acordes con su finalidad, como los que presentamos en el capítulo 2, cuando hablamos de la compañía Estimote. Al mismo tiempo, estos dispositivos presentan un menor coste, lo que supone una gran ventaja, si tenemos en cuenta que habría que disponer, como mínimo, del mismo número de dispositivos que de miembros de la dotación.
- El uso de aprendizaje automático basado en máquinas de vector soporte es perfectamente válido para entrenar el sistema, a partir de muestras de RSSI recibidas de diferentes balizas, y obtener la localización del personal a partir de muestras nuevas.
- A pesar de haber demostrado con los experimentos en los seminarios del CUD, que existen entornos donde resulta difícil diferenciar las salas que determinan el sistema de

localización, esta situación no se da en el patrullero “Tabarca”, verificado que la estructura que conforma la compartimentación del buque facilita esta diferenciación.

- Tras haber comprobado la eficacia del sistema para diferenciar once salas o compartimentos a partir de cuatro radiobalizas, se puede concluir que hemos cumplido con nuestro objetivo de no tener que emplear una radiobaliza por sala o compartimento.
- Otros de los objetivos propuestos en este TFG era conseguir disponer las radiobalizas en los diferentes entornos controlados, con el menor número de limitaciones a la hora de escoger la disposición física de cada una de ellas. Dicho objetivo queda cumplido al obtener resultados positivos en las dos pruebas realizadas en el patrullero “Tabarca”.

Acorde con los problemas planteados en las conclusiones y en vistas a proponer mejoras, alternativas y trabajos que extiendan el alcance de este TFG, se plantean las siguientes líneas futuras:

- En base al problema que presenta el tamaño de las placas Raspberry Pi, se propone implementar este sistema en un hardware diferente que posea las cualidades oportunas para ser portado por el personal de un buque de la Armada.
- Desarrollar un software que permita gestionar la información aportada por este sistema de radiobalizas. Este software, además de mostrarnos visualmente la información sobre la localización del personal, debería gestionarla de tal manera que nos permita utilizarla para las aplicaciones que se desarrollan cuando describimos la motivación de este trabajo.
- Probar otras técnicas de medición de posicionamiento basadas en aprendizaje automático, como, por ejemplo, métodos KNN, bayesianos o redes neuronales, y realizar una comparación con SVM.
- Refinamiento de localización: Los sistemas basados en balizas no conseguirán tener un rendimiento eficiente si no gestionan bien los errores causados por los multitrayectos, difracción, reflexiones de la señal o diferentes ganancias de antenas. Los algoritmos de refinamiento se centran en el comportamiento de la señal. Entre los métodos más usados se encuentran: Filtro de partículas, filtros de Kalman, Algoritmos de Monte Carlo o Modelos Ocultos de Markov. Se propone como línea futura estudiar la posible implementación de estos algoritmos de refinamiento en nuestro sistema de radiobalizas.
- *Bluetooth* v5.0: Según [65], esta nueva versión de Bluetooth contará con nuevas funcionalidades de ubicación y navegación, que permitirán impulsar el desarrollo y la adopción de sistemas de balizas y servicios basados en la localización. Se plantea estudiar esta nueva versión de *Bluetooth* como posible mejora.
- Estudiar los efectos de los dispositivos inalámbricos del entorno sobre el resultado de la localización, cambiando la disposición o incorporar nuevos dispositivos en el entorno controlado donde hemos instalado un sistema basado en radiobalizas.
- Investigar y estudiar la posibilidad de desarrollar un sistema de localización similar al que se propone en este TFG, basado en la señal recibida de los puntos de acceso WiFi, ya instalados en el entorno controlado.

6 BIBLIOGRAFÍA

- [1] Ione Camacho Lorenzo, *Sistema de control de personal a bordo: Implementación de un sistema iBeacon-BLE basado en Raspberry Pi*, Trabajo Fin de Grado. Centro Universitario de la Defensa en la Escuela Naval Militar. Marín.: Universidad de Vigo, 2015.
- [2] Fernando Seco Granja, *Tema 3.3: Sistemas de localización en interiores basados en radiofrecuencia*, Departamento de Electrónica. Universidad de Alcalá..
- [3] Luis Díaz-Ambrona Tabernilla, *Sistema de localización en interiores*, Madrid: Universidad Politécnica de Madrid, 2008.
- [4] «*The Cricket Indoor Location System*» [En línea]. Available: <http://cricket.csail.mit.edu/>. [Último acceso: 4 febrero 2018].
- [5] Veljo Otsason; Alex Varshavsky; Anthony LaMarca and Eyal de Lara, *Accurate gsm indoor location*, Berlín, Heidelberg: Springer (ISBN: 978-3-540-28760-5), 2005.
- [6] Shashank Tadakamadla, *Indoor Local Positioning System For ZigBee, Based On RSSI*, Mid Sweden University. *The Department of Information Technology and Media* (ITM), 2006.
- [7] *Texas Instruments Incorporated*, «*Texas Instruments*» [En línea]. Available: <http://www.ti.com/wireless-connectivity/simplelink-solutions/zigbee/overview.html>. [Último acceso: 24 febrero 2018].
- [8] «*Bluetooth Technology Website*» [En línea]. Available: <https://www.bluetooth.com/>. [Último acceso: 2018 marzo 1].
- [9] Francisco Munguía-Alcalá Rodrigo, «SLAM con mediciones angulares: métodos por triangulación estocástica» *Revista Ingeniería, Investigación y Tecnología*, vol. 14, nº 2, 2013.
- [10] ARVC UMH, «Automatización, Robótica y Visión por computador,» 30 Enero 2013. [En línea]. Available: <http://arvc.umh.es/>.
- [11] *Robotics & Computer Vision Engineer*, PhD, «Luigi Freda,» [En línea]. Available: <http://www.luigifreda.com>. [Último acceso: 2018 marzo 4].
- [12] Teruaki Kitasuka, Tsuneo NAKANISHI and Akira Fukuda, «*Design of WiPS: WLAN-Based Indoor Positioning System*» *Korea Multimedia Society*, vol. 7, nº 4, pp. 15-29, 2003.

- [13] Jesús Piedra Vega, Sistema de posicionamiento móvil en interiores vía WiFi, Universidad Oberta de Cataluña, 2016.
- [14] Miguel Cárdenas Montes, *K-Nearest Neighbors. Gráficas, estadísticas y minería de datos con python*, Madrid: Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, 2015.
- [15] Fernando Sancho Caparrini, «Dpto. de Ciencia de la Computación e Inteligencia Artificial. Universidad de Sevilla,» [En línea]. Available: <http://www.cs.us.es/~fsancho/?e=75>. [Último acceso: 2018 febrero 21].
- [16] «Aprendizaje automático: cómo un algoritmo ayuda a un médico o a un banco,» *BBVAOPEN4U* (<https://bbvaopen4u.com/es/actualidad/aprendizaje-automatico-como-un-algoritmo-ayuda-un-medico-o-un-banco>), 8 abril 2015.
- [17] Alberto Martín, «Cuando la máquina es mejor que el humano: *machine learning* y *trading financiero*» *Hipertextual* (<https://hipertextual.com/2016/08/machine-learning-trading>), 2 Agosto 2016.
- [18] Beatriz Guillén, «Así trabajan las mentes detrás de la inteligencia artificial,» *El País* (https://elpais.com/tecnologia/2016/06/14/actualidad/1465922089_437599.html), 21 junio 2016.
- [19] Virginia Frías García, «¿Qué es *Machine Learning* y qué aplicaciones tiene en nuestro día a día?» *Intelygenz* (<https://www.intelygenz.es/que-es-machine-learning-y-que-aplicaciones-tiene-dia-a-dia/>), 19 mayo 2016.
- [20] Luis González Abril, *Modelos de Clasificación basados en Máquinas de Vectores*, Sevilla: Departamento de Economía Aplicada I, Universidad de Sevilla, 2014.
- [21] Microsoft Azure, «Microsoft Azure» 18 Diciembre 2017. [En línea]. Available: <https://docs.microsoft.com/es-es/azure/machine-learning/studio/algorithm-choice>. [Último acceso: 23 febrero 2018].
- [22] José Ramón Berrendero, *Tema 6. Extensiones y aplicaciones (Máquinas de vectores soporte, SVM)*, Departamento de Matemáticas. Universidad de Madrid..
- [23] Carlos Alonso González, *SVM: Máquinas de Vectores Soporte*, Universidad de Valladolid. Departamento de informática. Grupo de Sistemas Inteligentes..
- [24] Cristina Martínez Ruedas, *Detección Multiusuario para DS-CDMA basado en SVM*, Universidad de Sevilla.
- [25] «tagingenieros,» 13 Marzo 2017. [En línea]. Available: <http://www.tagingenieros.com/?r=es/noticia/138>. [Último acceso: 12 febrero 2018].
- [26] Álvaro Andrés Saltos Vásquez, *Desarrollo de un sistema de SLAM para el robot AR.Drone en entorno ROS*, Madrid: Universidad de Alcalá, Escuela Politécnica Superior, 2015.
- [27] Ávaro Marco; Roberto Casas; Jorge Falcó; Héctor Gracia; José Ignacio Artigas; Armando Roy, «Location-based services for elderly and disabled people,» *Journal Computer Communications (COMPUT COMMUN)*, vol. 31, pp. 1055-1066, 2008.
- [28] «Ekahau. Wireless design,» [En línea]. Available: <https://www.ekahau.com/es>. [Último acceso: 2018 marzo 2].
- [29] Ekahau, Inc., «*RFID Solutions Online*» [En línea]. Available: <https://www.rfidsolutionsonline.com/doc/ekahau-real-time-location-system-0001>. [Último acceso:

23 febrero 2018].

- [30] Aeroscout, «Aeroscout Industrial» 2 marzo 2018. [En línea]. Available: <http://www.aeroscout.com/>.
- [31] «Extronics» [En línea]. Available: <http://www.extronics.com/aeroscout/>. [Último acceso: 25 enero 2018].
- [32] «Crunchbase» [En línea]. Available: <https://www.crunchbase.com/organization/pango-networks>. [Último acceso: 2018 marzo 3].
- [33] Ismael Cortázar Múgica; Pilar Sanz Velasco, «Localización en entornos WiFi,» *Comunicaciones de Telefónica I+D*, n° 36, pp. 47-62, 2005.
- [34] Roberto Battiti; Nhat Thang Le; Alessandro Villani, *Location-aware computing: a neural network model for determining location in wireless LANs.*, Trento: Universidad de Trento, 2002.
- [35] Blue Glue Pty. Ltd., «Blueglue» [En línea]. Available: http://www.blueglue.com.au/products_2/personnel_tracking.html. [Último acceso: 15 febrero 2018].
- [36] Australian Government, Department of Defence, «DST Group. Department of Defence, Science and Technology» [En línea]. Available: <https://www.dst.defence.gov.au/ctd-success-story/naval-automated-personnel-tracking>. [Último acceso: 2018 febrero 10].
- [37] «Navantia» [En línea]. Available: <http://www.navantia.es/>. [Último acceso: 4 febrero 2018].
- [38] «Omnisense LLC» [En línea]. Available: <https://www.omnisense.com/>. [Último acceso: 5 febrero 2018].
- [39] «Omnisense,» [En línea]. Available: <http://www.omnisense.co.uk/defence.html>. [Último acceso: 12 febrero 2018].
- [40] «Martec (Marine&Technologies). Integrated safety solutions,» [En línea]. Available: <http://www.martec.it/>. [Último acceso: 17 febrero 2018].
- [41] Apple Inc., «Developer. iBeacon,» [En línea]. Available: <https://developer.apple.com/ibeacon/>. [Último acceso: 2018 marzo 3].
- [42] Using Beacons, «Using Beacons» [En línea]. Available: <http://www.usingbeacons.com/eddystone-de-google-nuevas-posibilidades-para-la-tecnologia-beacon/>. [Último acceso: 13 febrero 2018].
- [43] *Creative Commons Attribution 3.0 License*; Apache 2.0 License, «Google Beacon Platform,» 6 Noviembre 2017. [En línea]. Available: <https://developers.google.com/beacons/eddystone>. [Último acceso: 15 febrero 2018].
- [44] «Estimote» [En línea]. Available: <https://estimote.com/products/>. [Último acceso: 5 marzo 2018].
- [45] *Situm Indoor positioning*, «Situm,» [En línea]. Available: <https://situm.es>. [Último acceso: 12 febrero 2018].
- [46] *Situm technologies*, S.L., «Cluster TIC Galicia,» [En línea]. Available: <http://www.clusterticgalicia.com/directorio.php?idioma=es&id=199&sec=58>. [Último acceso: 2018 febrero 23].

- [47] Raspberrypi, «*Raspberrypi. Teach, Learn, and Make with Raspberry Pi*» [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: 3 marzo 2018].
- [48] Raspberryshop, «Raspberryshop,» [En línea]. Available: <https://www.raspberryshop.es/hardware-raspberry-pi.php>. [Último acceso: 27 febrero 2018].
- [49] «Real VNC,» [En línea]. Available: <https://www.realvnc.com/es/>. [Último acceso: 2 febrero 2018].
- [50] «Putty» [En línea]. Available: <https://www.putty.org/>. [Último acceso: 7 febrero 2018].
- [51] «Geany» [En línea]. Available: <https://www.geany.org/>. [Último acceso: 2018 enero 23].
- [52] «Notepad++» [En línea]. Available: <https://notepad-plus-plus.org/download/v7.5.5.html>. [Último acceso: 4 marzo 2018].
- [53] «*Google Play*» [En línea]. Available: https://play.google.com/store/apps/details?id=com.yoo.taek.blescanner_recorder&hl=es_419. [Último acceso: 17 enero 2018].
- [54] Thorsten Joachims, «*SVM struct. Support Vector Machine for Complex Outputs*» 14 Agosto 2008. [En línea]. Available: http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html. [Último acceso: 2 marzo 2018].
- [55] *Linux Kernel Organization*, «*The Linux Kernel Archives*» [En línea]. Available: <https://www.kernel.org/category/about.html>. [Último acceso: 1 marzo 2018].
- [56] Thorsten Joachims, «*SVM light Support Vector Machine*» 14 Agosto 2008. [En línea]. Available: <http://svmlight.joachims.org/>. [Último acceso: 5 marzo 2018].
- [57] Vladimir Naumovich Vapnik, *The Nature of Statistical Learning Theory*, Segunda ed., New York: Springer-Verlag (ISBN: 978-0-387-98780-4), 1995.
- [58] Thorsten Joachims, *Learning to Classify Text Using Support Vector Machines*, S. US, Ed., Springer (ISBN: 978-0-7923-7679-8), 2002.
- [59] Thorsten Joachims, *Making large-Scale SVM Learning Practical*, Cambridge, USA: MIT Press, 1998.
- [60] Thorsten Joachims, «*Transductive inference for text classification using Support Vector Machine*» de *ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning*, San Francisco, 1999.
- [61] Thorsten Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, Berlin, Heidelberg: Springer (ISBN: 978-3-540-64417-0), 1998.
- [62] Thorsten Joachims, «*Estimating the Generalization Performance of a SVM Efficiently*» *Morgan Kaufmann Publishers Inc. San Francisco, CA, USA*, pp. 431-438, 2000.
- [63] Thorsten Joachims, «*Optimizing Search Engines Using Clickthrough Data*» de *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada, julio, 2002.
- [64] Cornell University. *Department of Computer Science*, «*Learning Retrieval Functions from Implicit Feedback*» [En línea]. Available: <http://www.cs.cornell.edu/people/tj/career/>. [Último acceso: 1 Marzo 2018].

- [65] «5 ventajas de Bluetooth 5, la última versión de la popular tecnología inalámbrica,» *BBC* (<http://www.bbc.com/mundo/noticias-36517394>), 13 Junio 2016.
- [66] Sedar Yegulalp, «InfoWorld» 5 Octubre 2017. [En línea]. Available: <https://www.infoworld.com/article/2853707/machine-learning/11-open-source-tools-machine-learning.html>.
- [67] Jean-François Puget, «*Inside Machine Learning*» 16 Marzo 2017. [En línea]. Available: <https://medium.com/inside-machine-learning/the-most-popular-language-for-machine-learning-is-46e2084e851b>.
- [68] Chia-Hua Ho; Chih-Jen Lin, «*Large-scale Linear Support Vector Regression*» *Journal of Machine Learning Research*, 2012.

ANEXO I: GLOSARIO

AGV: *Automated Guided Vehicles*
ANN: *Artificial Neural Network*
AOA: *Angle of Arrival*
AP: *Access Point*
API: *Application Programming Interface*
BAM: *Buque de Acción Marítima*
BER: *Bit Error Rate*
CART: *Classification and Regression Trees*
DSTO: *Defence Science and Technology Group*
DTOA: *Differential Time of Arrival*
EDR: *Enhanced Data Rate*
E-OTD: *Enhanced Observed Time Difference*
FABA: *Fábrica de Artillería de Bazán*
GSM: *Global System for Mobile*
HCI: *Host Controller Interface*
L2CAP: *Logical Link Control and Adaptation Protocol*
LEAd: *Low Energy Advertising*
LoS: *Line of Sight*
LPS: *Local Positioning Systems*
LQI: *Link Quality Indicator*
MSDF: *Multi-sensor Smart Data Fusion*
NAPT: *Naval Automated Personnel Tracking*
OPTS: *On Board Personnel Tracking System*
PDU: *Payload Data Unit*
PDU: *Protocol Data Unit*
RF: *Radio Frequency*
RFID: *Radio Frequency Identification*
RPI: *Raspberry Pi*
RSSI: *Received Signal Strength Indicator*
RTLS: *Real Time Location System*
RTOF: *Round-trip Time of Flight*
SDK: *Software Development Kits*
SICP: *Sistema Integrado de Control de Plataforma*
SIG: *Special Interest Group*
SIR: *Sampling Importance Resampling*
SLAM: *Simultaneous Location and Mapping*
SOES: *Supervisor Operativo de Equipos y Sistemas*
SO: *Sistema Operativo*
SSP: *Secure Simple Pairing*
SURF: *Speeded-Up Robust Features*
SVM: *Support Vector Machine*
SVR: *Support Vector Regression*
TOA: *Time of Arrival*
TOF: *Time of Flight*
TSVM: *Transductive Support Vector Machines*
UWB: *Ultra Wide Band*
WIPS: *Wireless Indoor Positioning System*
WSN: *Wireless Sensor Network*

ANEXO II: SCRIPT PARA EL MODO ESCÁNER: FORMATO SVM^{LIGHT}

En este anexo mostramos el script *scansvm.sh* utilizado para volcar los datos de RSSI de cada baliza en el formato necesario, para posteriormente poder manejar esa información con SVM^{light}. En la Figura 3-5 se muestra un ejemplo de la salida tras ejecutar el script.

```
#!/bin/bash
# Este script sirve para imprimir el promedio de la RSSI de cada radiobaliza,
# en el formato normalizado para las librerías SVMlight.
if [[ $1 == "parse" ]]; then
    packet=""
    capturing=""
    count=0
    ULTDATE=`date +%T`
    function cero() {
        let C2=0
        let C3=0
        let C4=0
        let C5=0
        let RSSI2=0
        let RSSI3=0
        let RSSI4=0
        let RSSI5=0
    }
    function acu(){
        if [ $MAJOR = 2 ]; then
            let RSSI2=$RSSI2+$RSSI
            let C2=$C2+1
        else
            if [ $MAJOR = 3 ]; then
                let RSSI3=$RSSI3+$RSSI
                let C3=$C3+1
            else
                if [ $MAJOR = 4 ]; then
                    let RSSI4=$RSSI4+$RSSI
                    let C4=$C4+1
                else
                    let RSSI5=$RSSI5+$RSSI
                    let C5=$C5+1
                fi
            fi
        fi
    }
    zero
    while read line
    do
        count=$((count + 1))
        if [ "$capturing" ]; then
            if [[ $line =~ ^[0-9a-fA-F]{2}\ [0-9a-fA-F] ]]; then
                packet="$packet $line"
            else
                if [[ $packet =~ ^04\ 3E\ 2A\ 02\ 01\ \.{26}\ 02\ 01\ \.{14}\ 02\ 15 ]]; then
                    MAJOR=`echo $packet | sed 's/^\.\{117\}\(\.\{5\}\)\.*$/\1/'`
                    RSSI=`echo $packet | sed 's/^\.\{132\}\(\.\{2\}\)\.*$/\1/'`
                fi
            fi
        fi
    done
fi
```

```

MAJOR=`echo $MAJOR | sed 's/\ //g'`
MAJOR=`echo "ibase=16; $MAJOR" | bc`
RSSI=`echo "ibase=16; $RSSI" | bc`
RSSI=$((RSSI - 256))
DATE=`date +%T`
if [ $DATE = $ULTDATE ]; then
    acu
else
    if [ $C2 = 0 ];then
        let RSSI2=-100
    else
        let RSSI2=$((RSSI2/$C2))
    fi
    if [ $C3 = 0 ];then
        let RSSI3=-100
    else
        let RSSI3=$((RSSI3/$C3))
    fi
    if [ $C4 = 0 ];then
        let RSSI4=-100
    else
        let RSSI4=$((RSSI4/$C4))
    fi
    if [ $C5 = 0 ];then
        let RSSI5=-100
    else
        let RSSI5=$((RSSI5/$C5))
    fi
    echo "+1 2:$RSSI2 3:$RSSI3 4:$RSSI4 5:$RSSI5"
cero
ULTDATE=$DATE
acu
fi
fi
capturing=""
packet=""
fi
fi
if [ ! "$capturing" ]; then
    if [[ $line =~ ^\> ]]; then
        packet=`echo $line | sed 's/^\>.\(.*$\)/\1/'`
        capturing=1
    fi
fi
done
else
    sudo hcitool lescan --passive --duplicates 1>/dev/null &
    if [ "$(pidof hcitool)" ]; then
        sudo hcidump --raw | ./$0 parse $1
    fi
fi
fi

```

ANEXO III: SCRIPT PARA EL MODO ESCÁNER: FORMATO PARA LOS PRIMEROS EXPERIMENTOS

En este anexo mostramos el script *scan.sh* utilizado para volcar los datos de RSSI de cada baliza según los vaya recibiendo la RPi1 operando en modo escáner. En la Figura 3-6 se muestra un ejemplo de la salida tras ejecutar el script.

```
#!/bin/bash
# Este script sirve para imprimir la RSSI de las tramas de las radiobalizas
que le van llegando al escáner.
if [[ $1 == "parse" ]]; then
    packet=""
    capturing=""
    count=0
    while read line
    do
        count=$((count + 1))
        if [ "$capturing" ]; then
            if [[ $line =~ ^[0-9a-fA-F]{2}\ [0-9a-fA-F] ]]; then
                packet="$packet $line"
            else
                if [[ $packet =~ ^04\ 3E\ 2A\ 02\ 01\ \.{26}\ 02\ 01\ \.{14}\ 02\ 15 ]]; then
                    then
                        MAJOR=`echo $packet | sed 's/^\.{117}\(. \{5\}\)\.*$/\1/'`
                        RSSI=`echo $packet | sed 's/^\.{132}\(. \{2\}\)\.*$/\1/'`
                        MAJOR=`echo $MAJOR | sed 's/\ //g'`
                        MAJOR=`echo "ibase=16; $MAJOR" | bc`
                        RSSI=`echo "ibase=16; $RSSI" | bc`
                        RSSI=$((RSSI - 256))
                        DATE=`date +%T`
                        if [[ $2 == "-b" ]]; then
                            echo "$MAJOR $DATE $RSSI"
                        else
                            echo "$MAJOR $DATE $RSSI"
                        fi
                    fi
                    capturing=""
                    packet=""
                fi
            fi
            if [ ! "$capturing" ]; then
                if [[ $line =~ ^\> ]]; then
                    packet=`echo $line | sed 's/^\>.\(.*\)/\1/'`
                    capturing=1
                fi
            fi
        done
    else
        sudo hcitool lescan --passive --duplicates 1>/dev/null &
        if [ "$(pidof hcitool)" ]; then
            sudo hcidump --raw | ./$0 parse $1
        fi
    fi
fi
```


ANEXO IV: SCRIPT PARA ANÁLISIS SVM: *EXPERIMENTO 6*

En este anexo mostramos el script *apreneval01.sh*.

```
#!/bin/bash

#### Este script sirve para entrenar y generar los modelos de cada compartimento a partir de
las capturas que se hayan realizado en dichos compartimento y posteriormente evaluar
compartimento por compartimento con los diferentes modelos sacados a partir del entrenamiento y
mostrar los resultados.

##### Configurar

GAMMAS="0.001 0.01 0.1 1 10"
COMS=`ls *.ori | cut -d. -f1 | sort -u`

##### Desordenar ficheros originales
##### Quedarme con las 1200 primeras líneas
##### Crear un fichero para el estrenamiento con las primeras 1000 líneas, con (+1)
##### Crear un fichero para el estrenamiento con las primeras 1000 líneas, con (-1)
##### Crear un fichero para la evaluación con las últimas 200 líneas, con (+1)

for COM in $COMS
do
shuf $COM.ori > $COM.dat
sed -i '1201,$d' $COM.dat
sed '1001,1200d' $COM.dat > $COM.pos.temp
sed 's/+1/-1/' $COM.pos.temp > $COM.neg.temp
sed '1,1000d' $COM.dat > $COM.evalua
done

##### Crear el fichero completo de entrenamiento, con los datos (+1) del compartimento que
queremos entrenar y los datos (-1) del resto de compartimentos. Esta parte del script se debe
modificar, dependiendo del número de compartimentos que queramos entrenar y evaluar

cat G2.pos.temp G3.neg.temp > G2.entrena
cat G3.pos.temp G2.neg.temp > G3.entrena
```

```

##### Entrenar

for NUM in $COMS
do
./svm_learn $NUM.entrena $NUM-LINEAL.model
done

for NUM in $COMS
do
for gamma in $GAMMAS
do
./svm_learn -t 2 -g $gamma $NUM.entrena $NUM-RADIAL-$gamma.model
done
done

##### Clasificar

for PRU in $COMS
do
for NUM in $COMS
do
./svm_classify $PRU.evalua $NUM-LINEAL.model $PRU-$NUM-LINEAL.temp > $PRU-$NUM-LINEAL.result
done
done

for PRU in $COMS
do
for NUM in $COMS
do
for gamma in $GAMMAS
do
./svm_classify $PRU.evalua $NUM-RADIAL-$gamma.model $PRU-$NUM-RADIAL-$gamma.temp > $PRU-$NUM-
RADIAL-$gamma.result
done
done
done

##### Imprimir resultados

for PRU in $COMS
do
for NUM in $COMS
do
fgrep Accuracy *.result | grep "LINEAL."
done
done

for PRU in $COMS
do
for NUM in $COMS
do
for gamma in $GAMMAS
do
fgrep Accuracy *.result | grep "RADIAL-$gamma."
done
done
done

#### Limpiar los ficheros temporales que hemos creado

rm *.temp

```

ANEXO V: SCRIPT PARA ANÁLISIS SVM: VALIDACIÓN EN EL PATRULLERO “TABARCA”

En este anexo mostramos el script *apreneval02.sh*.

```
#!/bin/bash

#### Este script sirve para entrenar y generar los modelos de cada compartimento a
partir de las capturas que se hayan realizado en dichos compartimento y posteriormente
evaluar compartimento por compartimento con los diferentes modelos sacados a partir del
entrenamiento y mostrar los resultados.

##### Configurar

GAMMAS="0.001 0.01 0.1 1 10"
COMS=`ls *.ori | cut -d. -f1 | sort -u`

##### Desordenar ficheros originales
##### Quedarme con las 1000 primeras líneas
##### Crear un fichero para el estrenamiento con las primeras 900 líneas, con (+1)
##### Crear un fichero para el estrenamiento con las primeras 900 líneas, con (-1)
##### Crear un fichero para la evaluación con las últimas 100 líneas, con (+1)

for COM in $COMS
do
shuf $COM.ori > $COM.dat
sed -i '1001,$d' $COM.dat
sed '901,1000d' $COM.dat > $COM.pos.temp
sed 's/+1/-1/' $COM.pos.temp > $COM.neg.temp
sed '1,900d' $COM.dat > $COM.evalua
done

##### Crear el fichero completo de entrenamiento, con los datos (+1) del compartimento
que queremos entrenar y los datos (-1) del resto de compartimentos. Esta parte del
script se debe modificar, dependiendo del número de compartimentos que queramos
entrenar y evaluar

cat COM01.pos.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM01.entrena
cat COM01.neg.temp COM02.pos.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM02.entrena
cat COM01.neg.temp COM02.neg.temp COM03.pos.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM03.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.pos.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM04.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.pos.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM05.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.pos.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM06.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.pos.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM07.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.pos.temp COM09.neg.temp COM10.neg.temp
COM11.neg.temp > COM08.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.pos.temp COM10.neg.temp
COM11.neg.temp > COM09.entrena
```

```

cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.pos.temp
COM11.neg.temp > COM10.entrena
cat COM01.neg.temp COM02.neg.temp COM03.neg.temp COM04.neg.temp COM05.neg.temp
COM06.neg.temp COM07.neg.temp COM08.neg.temp COM09.neg.temp COM10.neg.temp
COM11.pos.temp > COM11.entrena

#### Entrenar

for NUM in $COMS
do
./svm_learn $NUM.entrena $NUM-LINEAL.model
done

for NUM in $COMS
do
for gamma in $GAMMAS
do
./svm_learn -t 2 -g $gamma $NUM.entrena $NUM-RADIAL-$gamma.model
done
done

#### Clasificar

for PRU in $COMS
do
for NUM in $COMS
do
./svm_classify $PRU.evalua $NUM-LINEAL.model $PRU-$NUM-LINEAL.temp > $PRU-$NUM-
LINEAL.result
done
done

for PRU in $COMS
do
for NUM in $COMS
do
for gamma in $GAMMAS
do
./svm_classify $PRU.evalua $NUM-RADIAL-$gamma.model $PRU-$NUM-RADIAL-$gamma.temp > $PRU-
$NUM-RADIAL-$gamma.result
done
done
done

#### Imprimir resultados

for PRU in $COMS
do
for NUM in $COMS
do
fgrep Accuracy *.result | grep "LINEAL."
done
done

for PRU in $COMS
do
for NUM in $COMS
do
for gamma in $GAMMAS
do
fgrep Accuracy *.result | grep "RADIAL-$gamma."
done
done
done

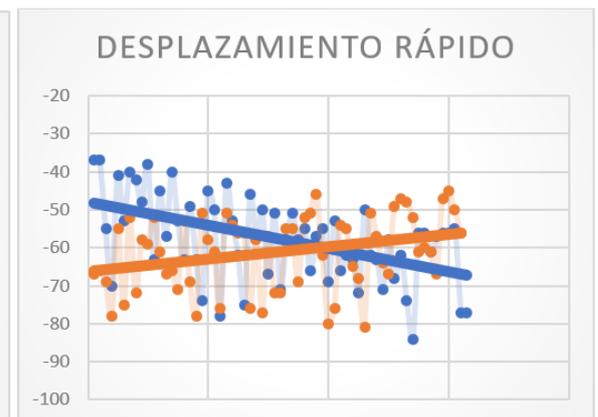
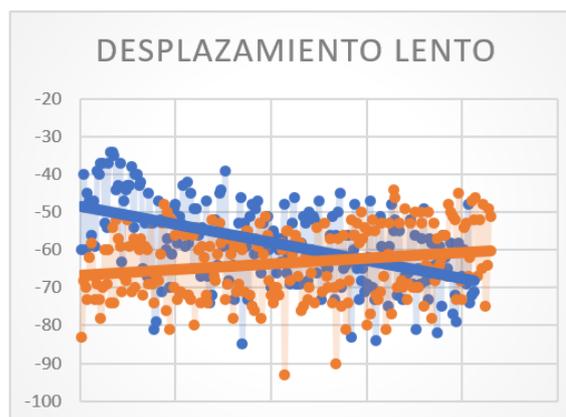
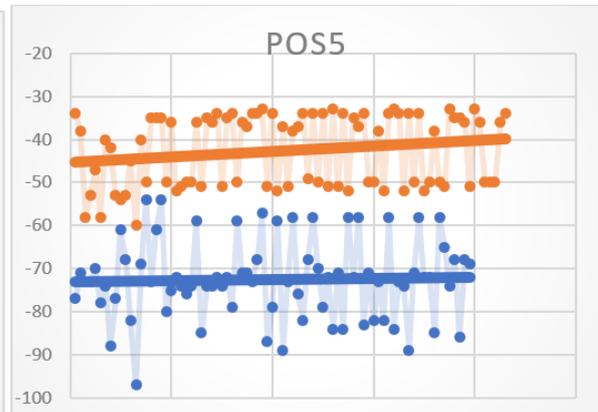
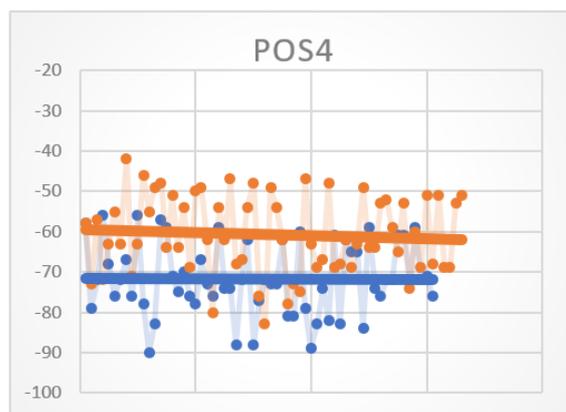
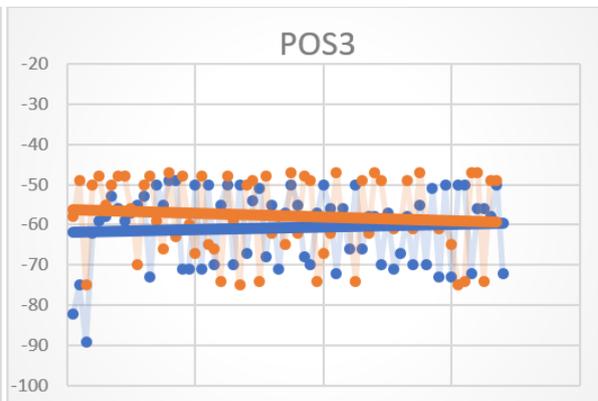
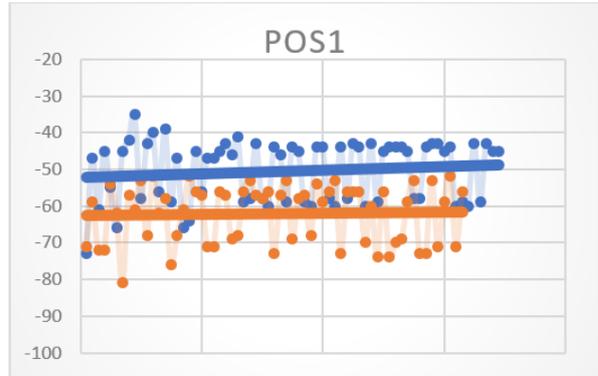
#### Limpiar los ficheros temporales que hemos creado

rm *.temp

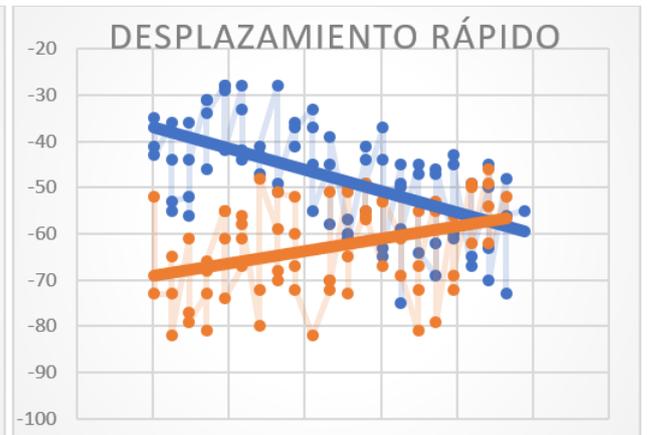
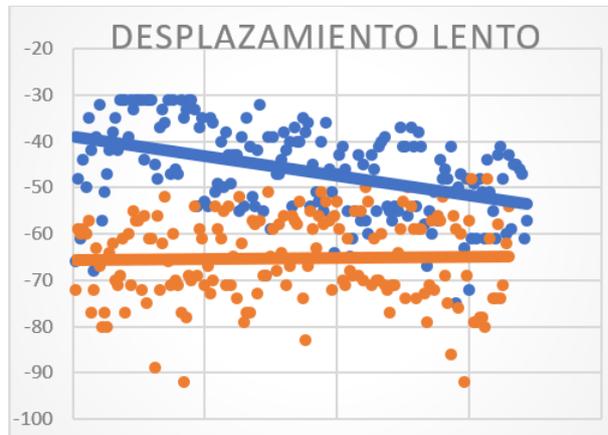
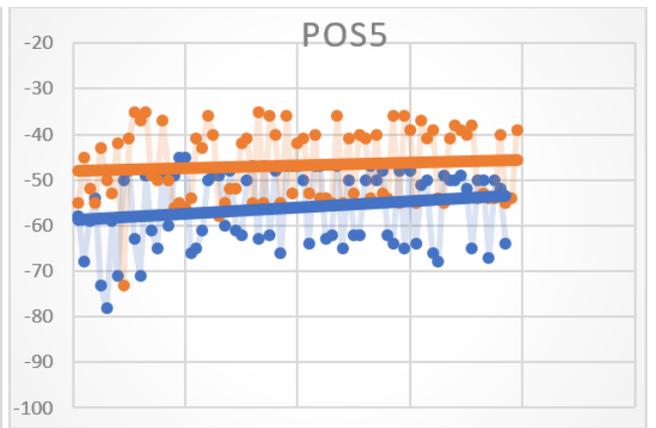
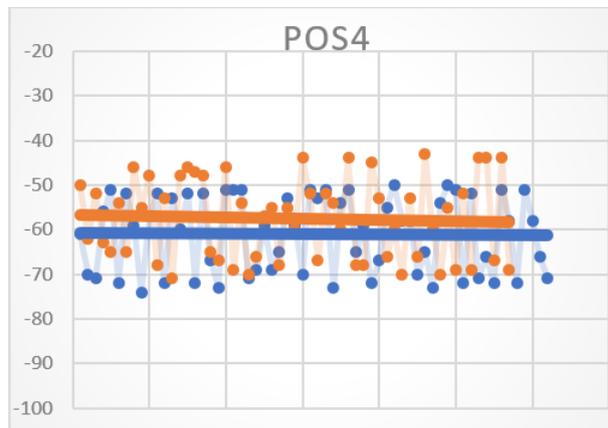
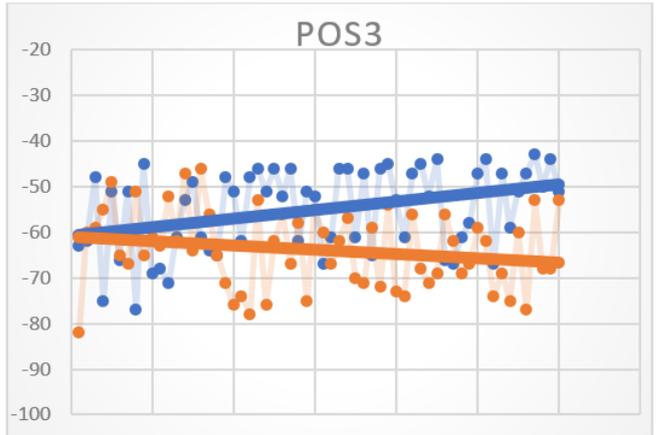
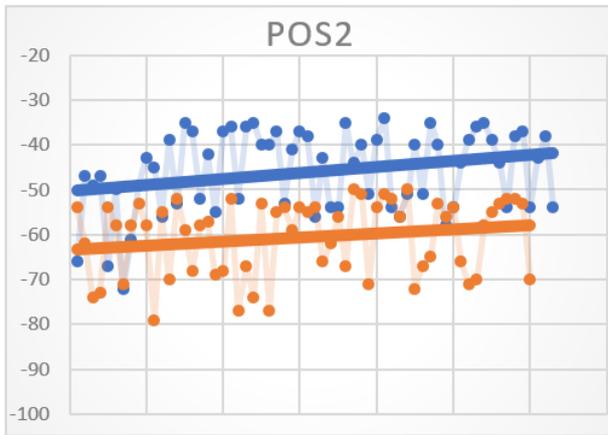
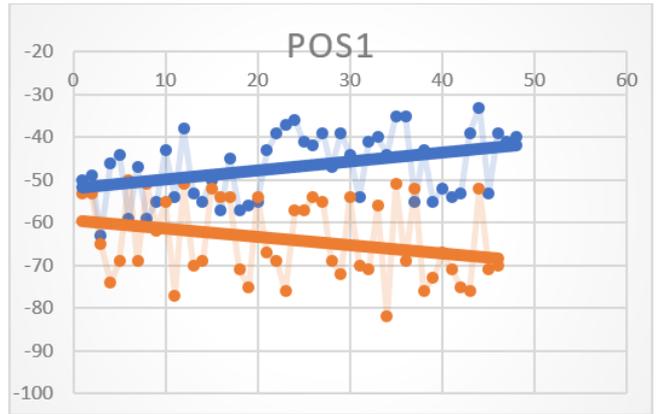
```

ANEXO VI: RESULTADOS DE LA FASE DE EXPERIMENTACIÓN. *EXPERIMENTO 1*

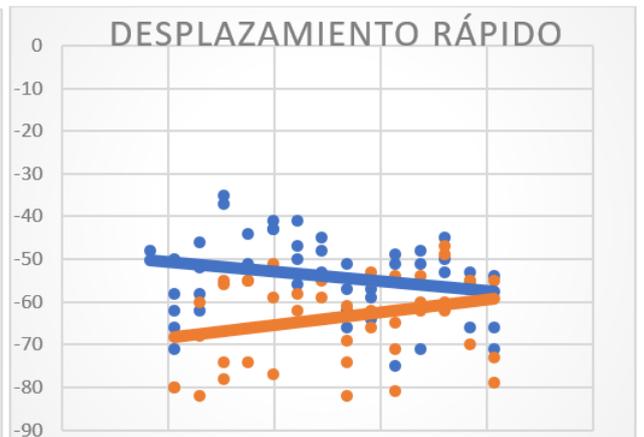
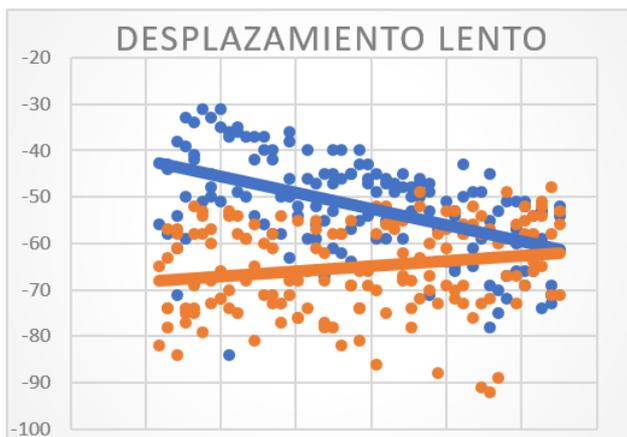
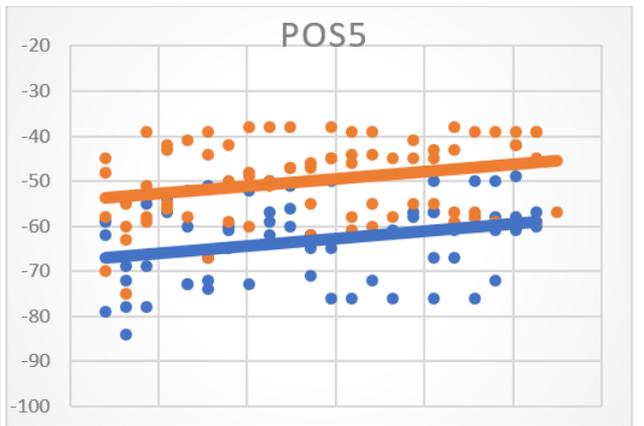
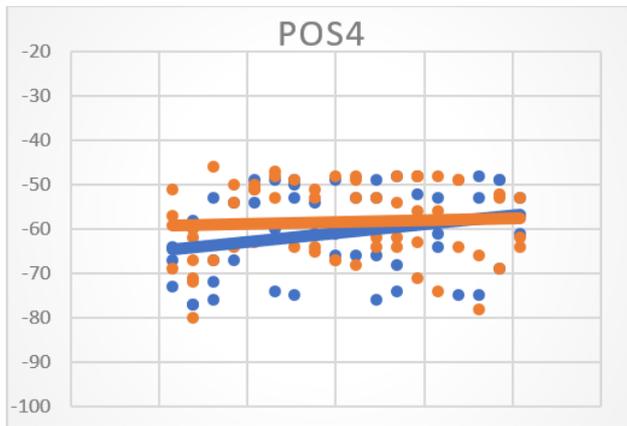
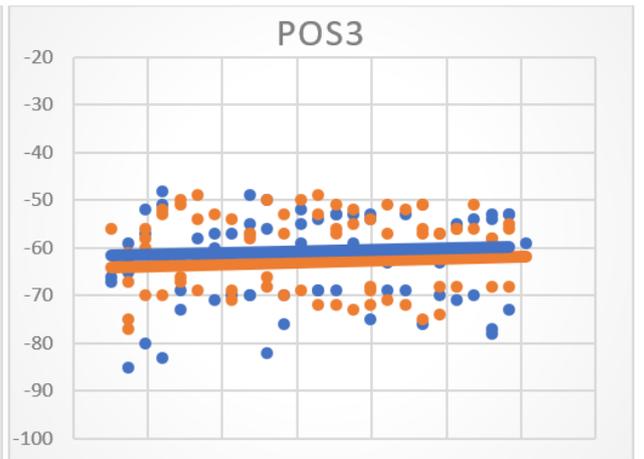
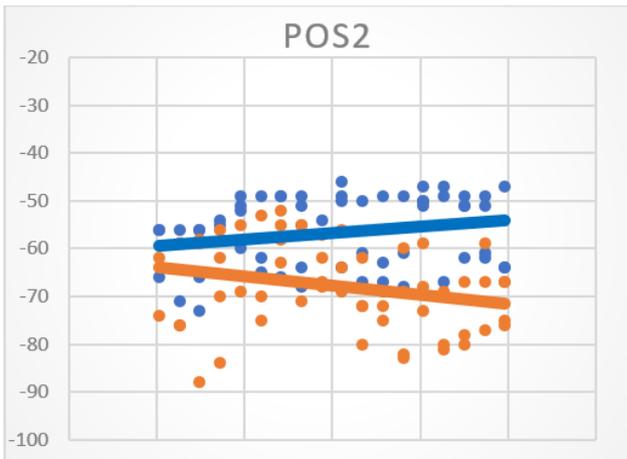
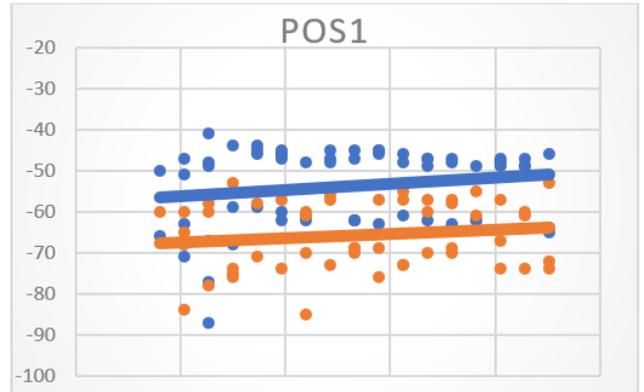
PRUEBA 1	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	100 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS5 (NARANJA)	RPi3



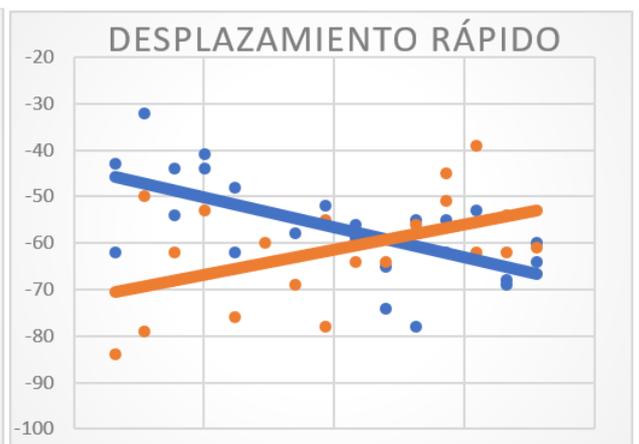
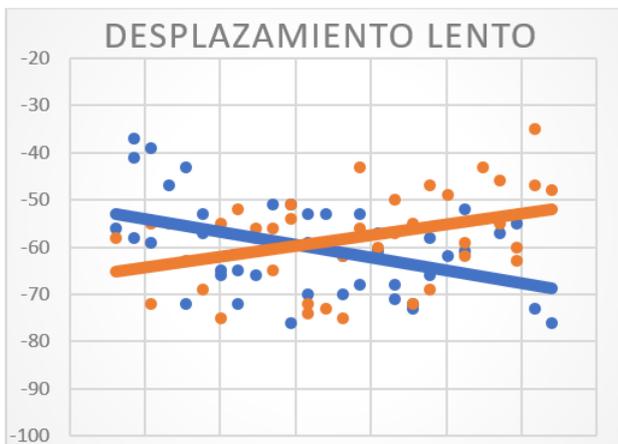
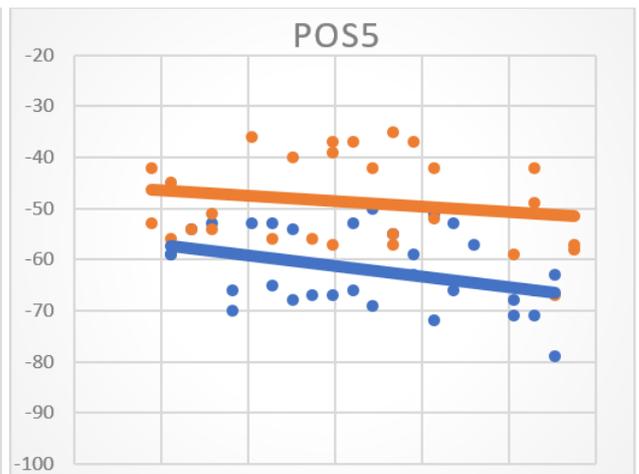
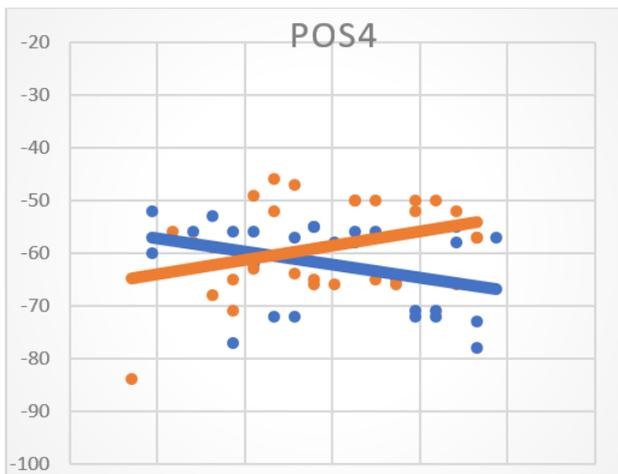
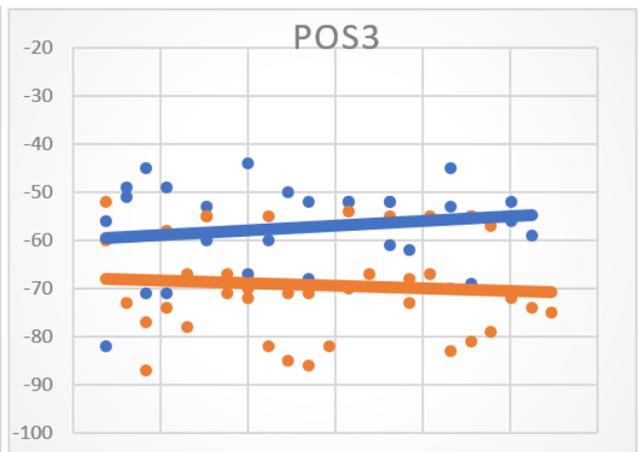
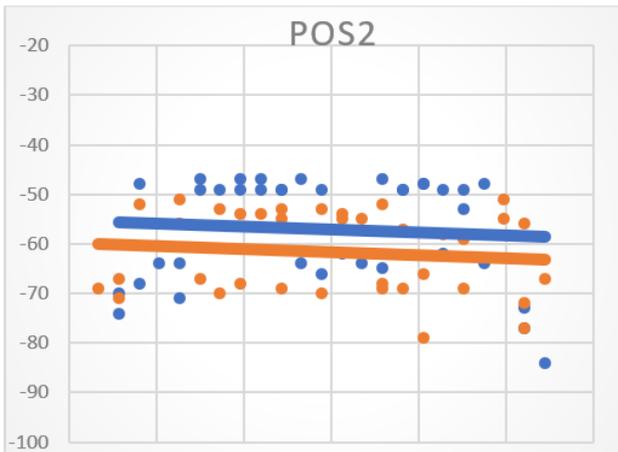
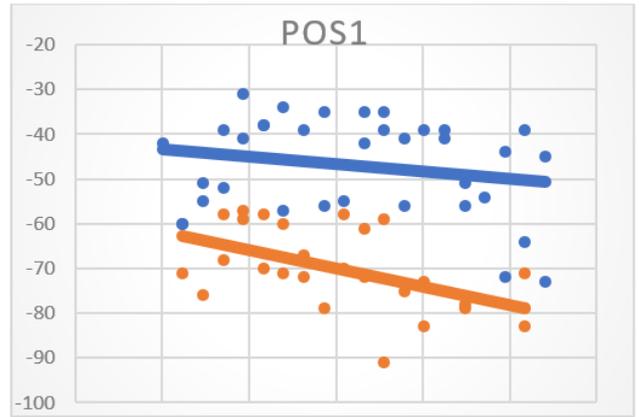
PRUEBA 2	
POSICIÓN RPi	VERTICAL
ALTURA	MENOR
PERIODO	100 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS5 (NARANJA)	RPi3



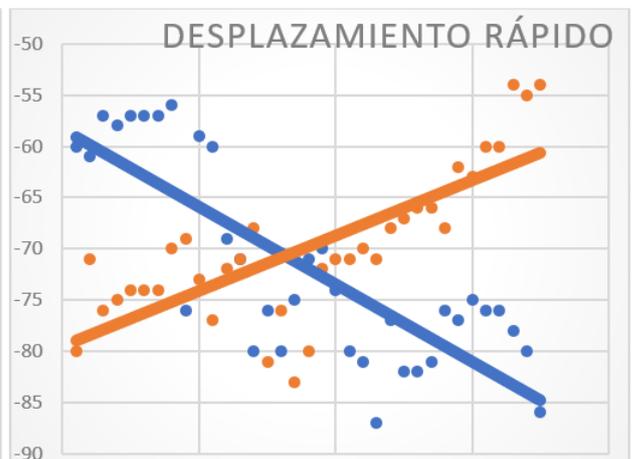
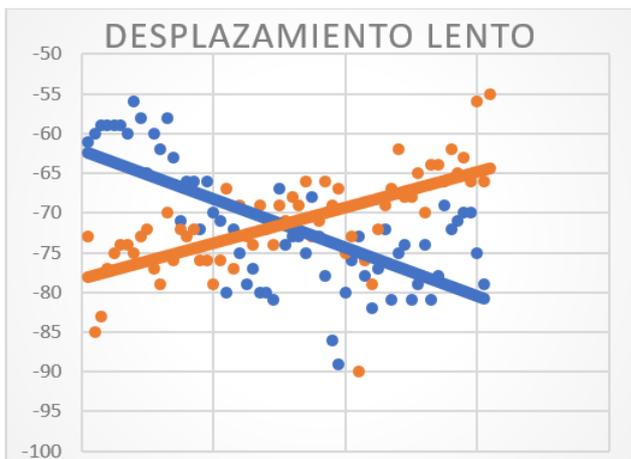
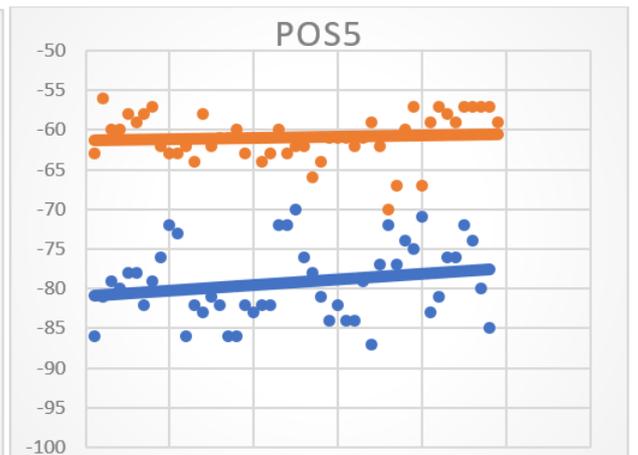
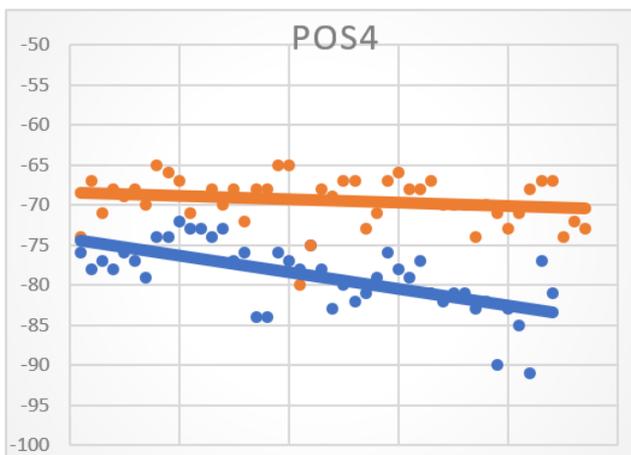
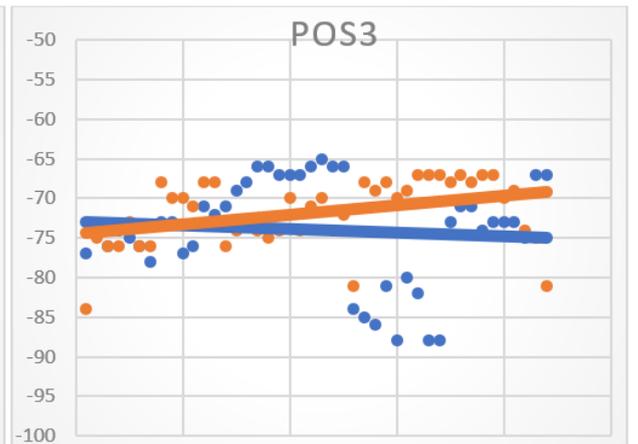
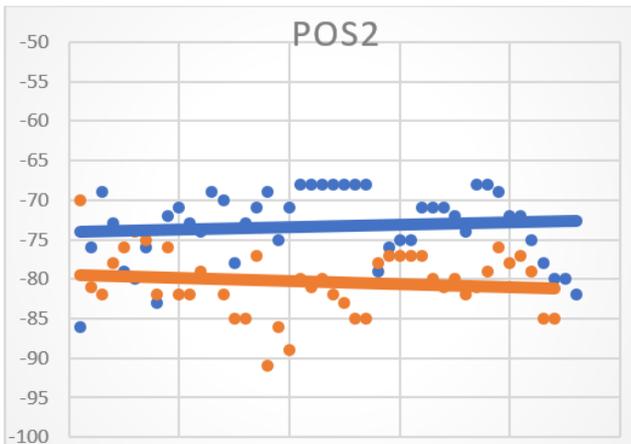
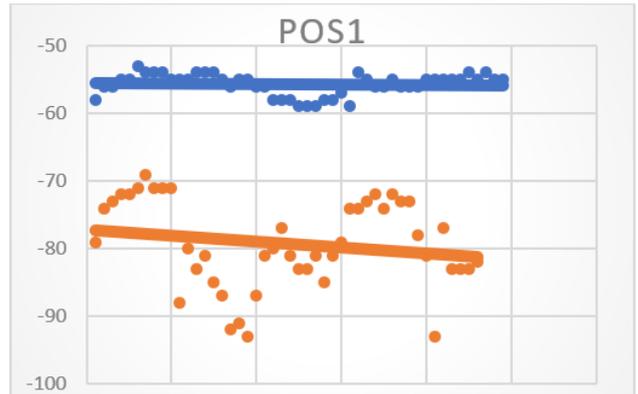
PRUEBA 3	
POSICIÓN RPi	VERTICAL
ALTURA	MAYOR
PERIODO	100 ms
SCANNER	RPi1
POS1 (AZUL)	RPi2
POS5 (NARANJA)	RPi3



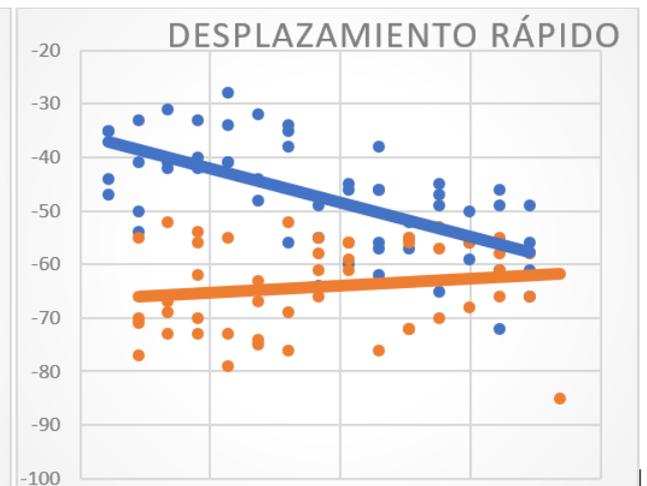
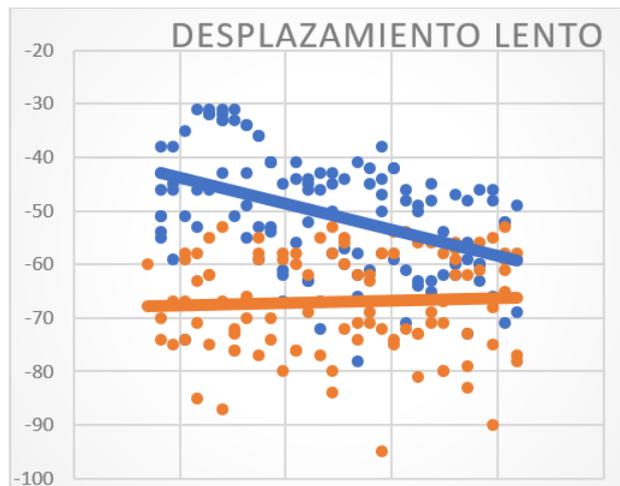
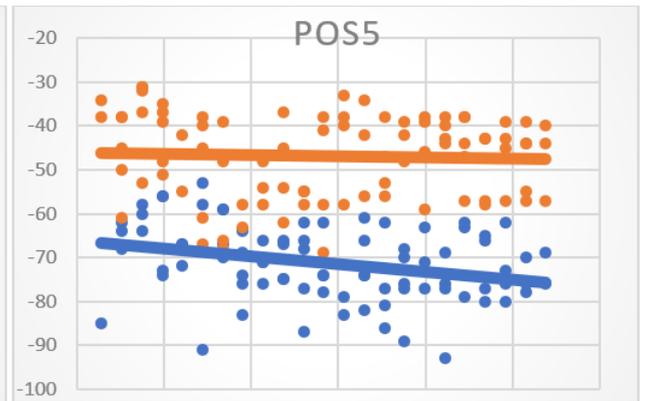
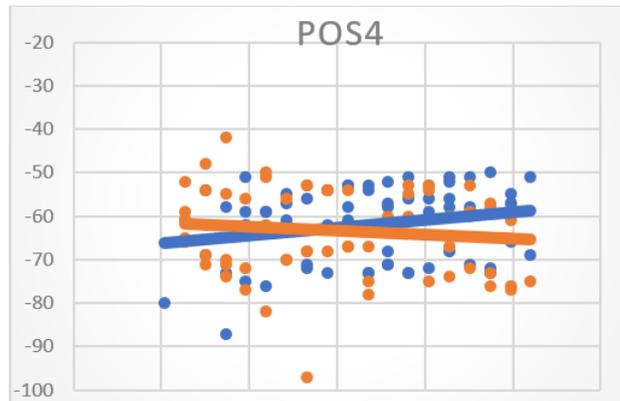
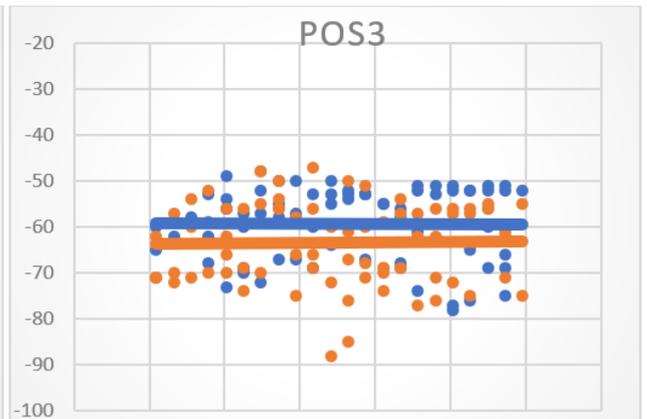
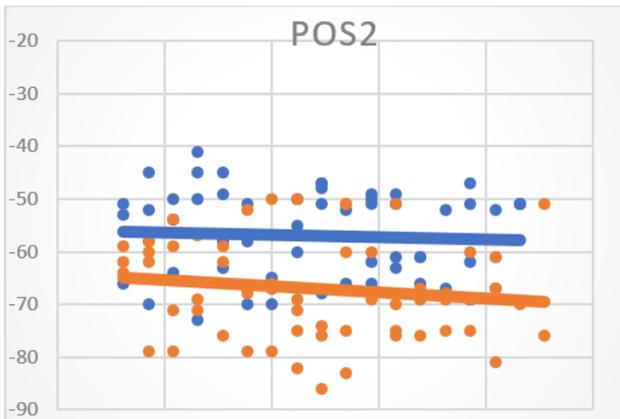
PRUEBA 4	
POSICIÓN RPi	VERTICAL
ALTURA	MAYOR
PERIODO	500 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS5 (NARANJA)	RPi3



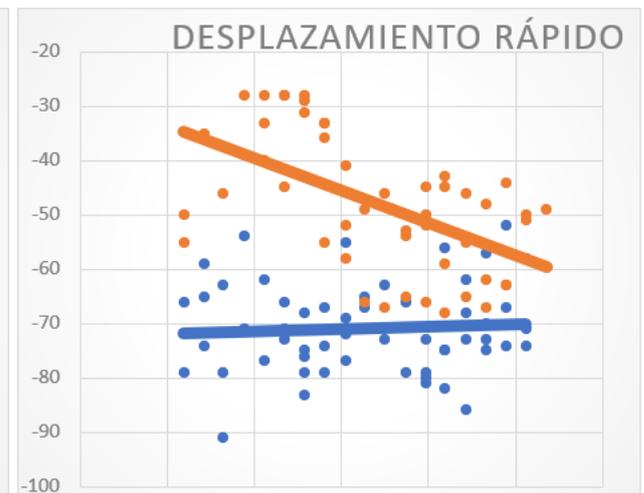
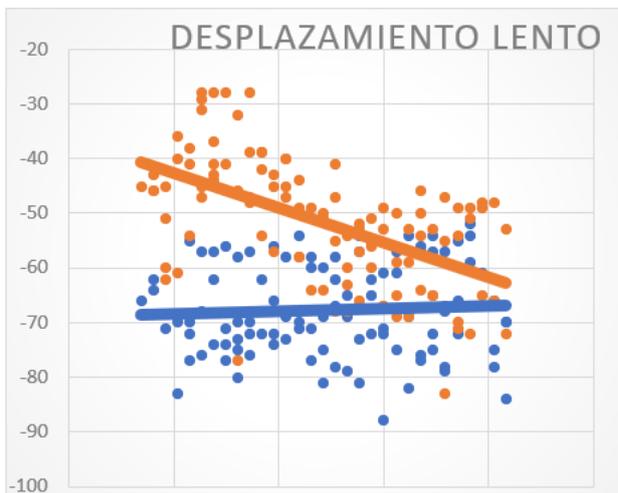
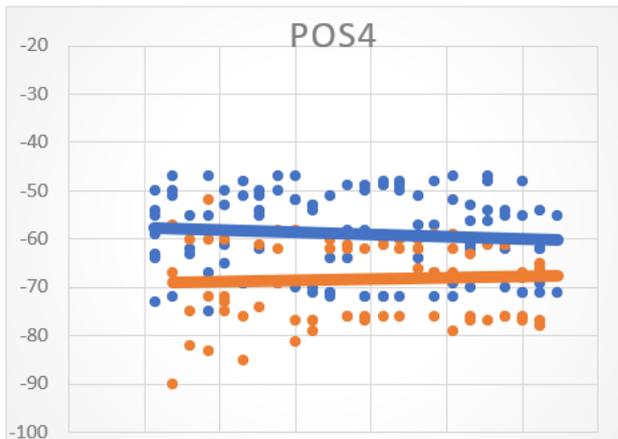
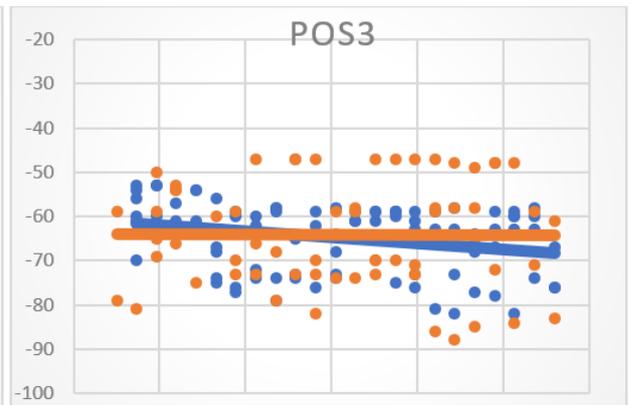
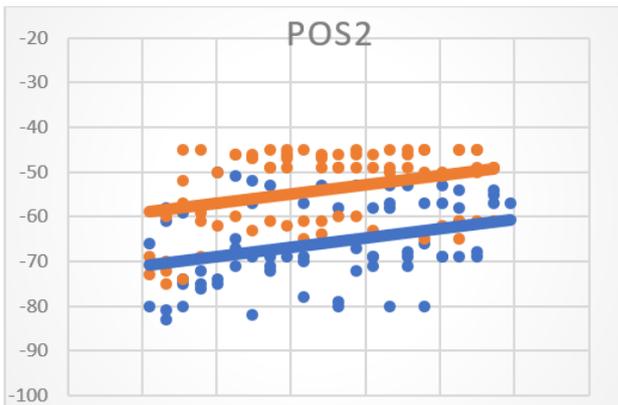
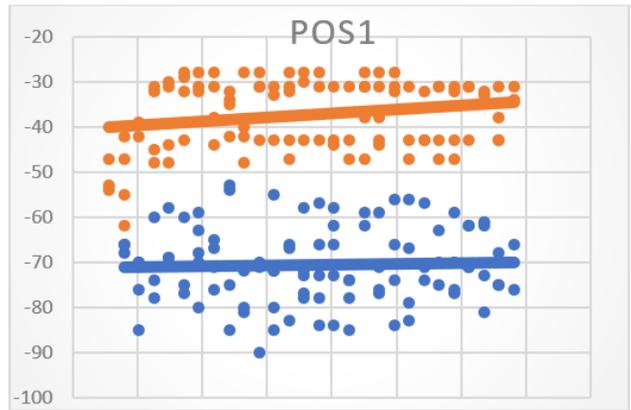
PRUEBA 5	
POSICIÓN RPI	VERTICAL
ALTURA	MAYOR
PERIODO	500 ms
ESCÁNER	MÓVIL
POS1 (AZUL)	RPI2
POS5 (NARANJA)	RPI3



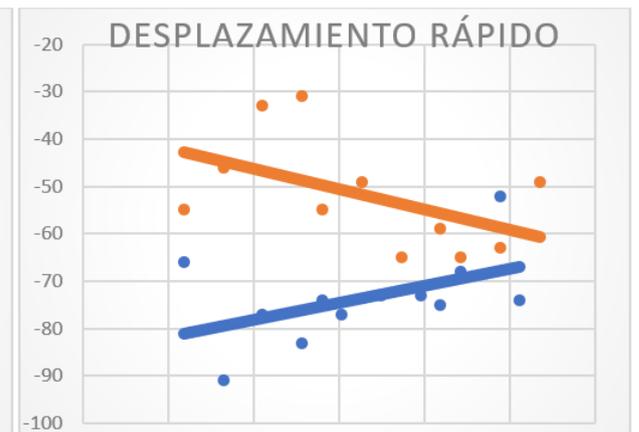
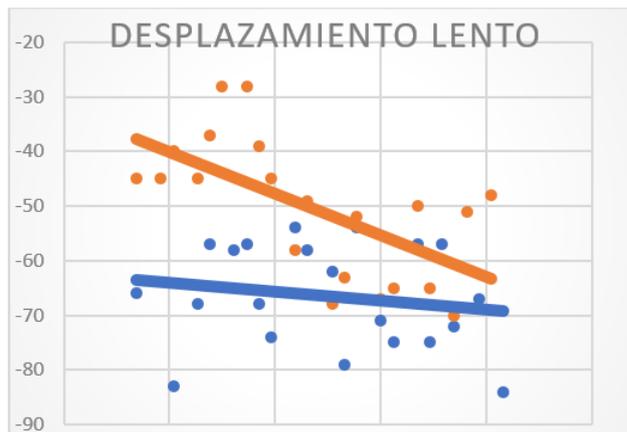
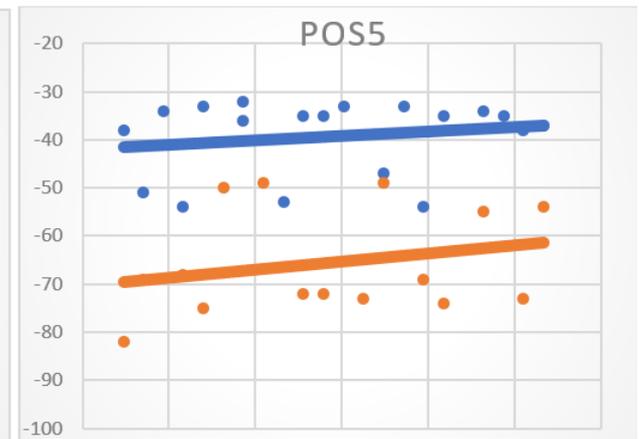
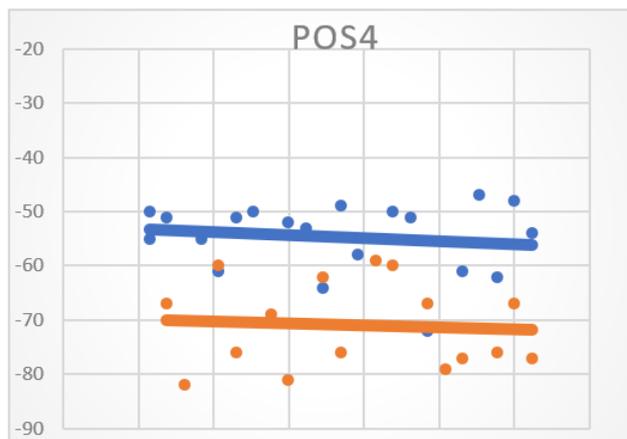
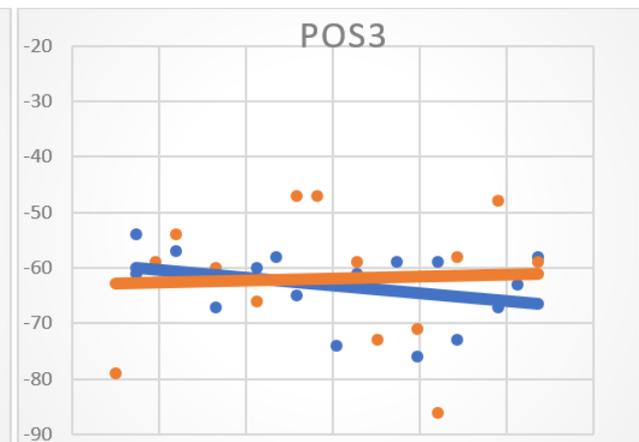
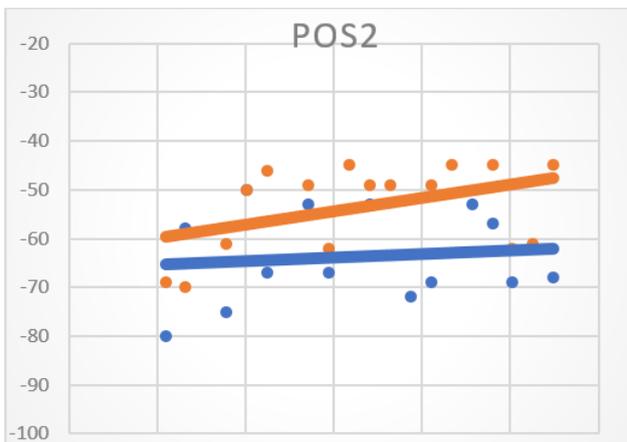
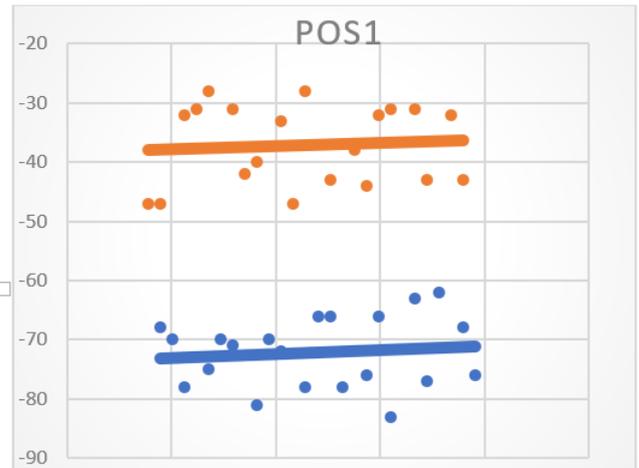
PRUEBA 6	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	100 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS5 (NARANJA)	RPi3



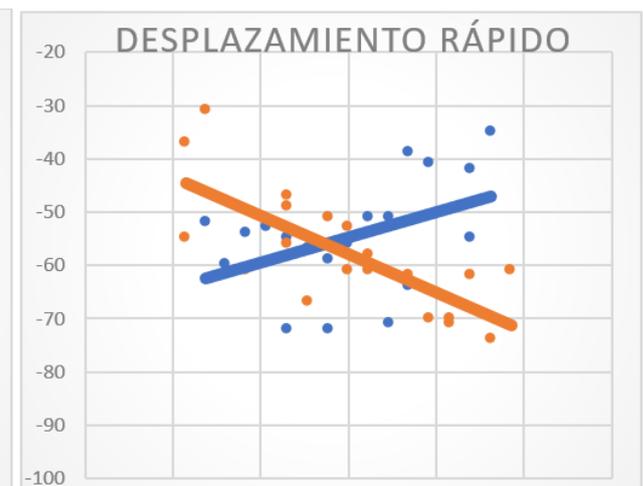
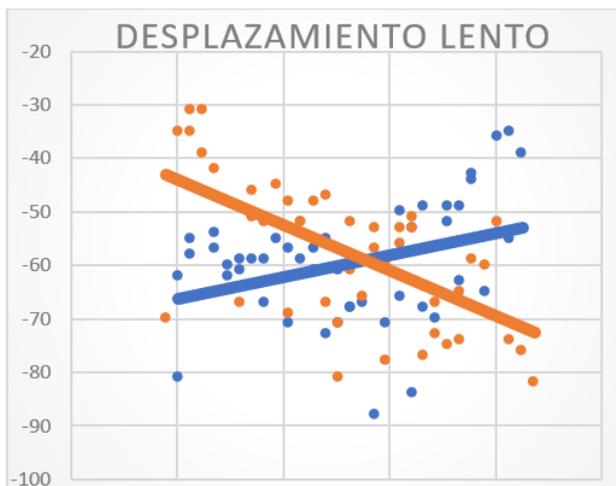
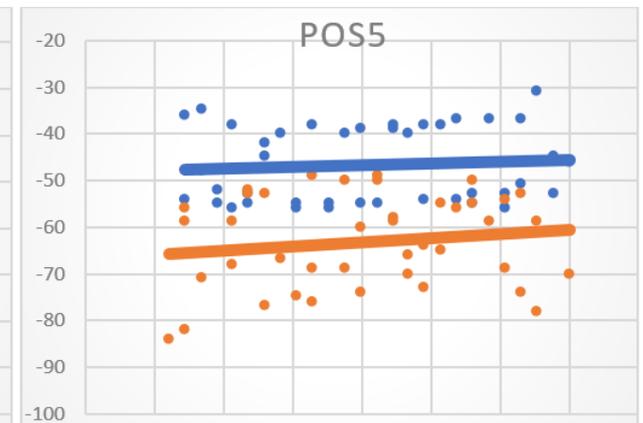
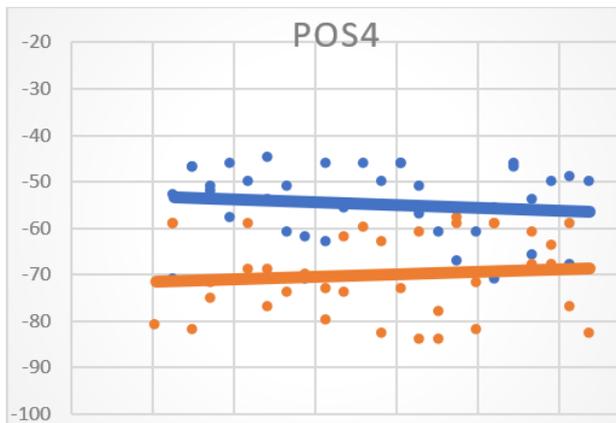
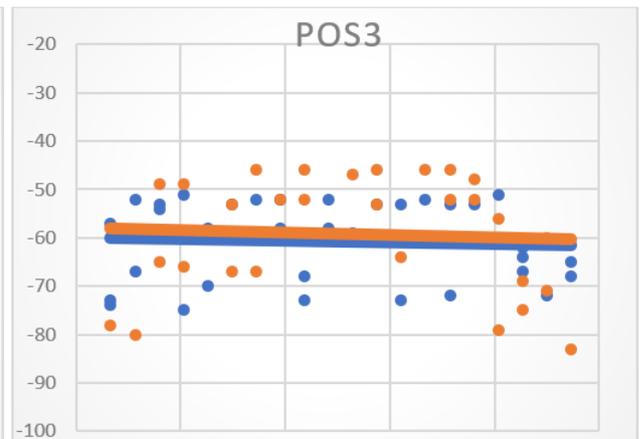
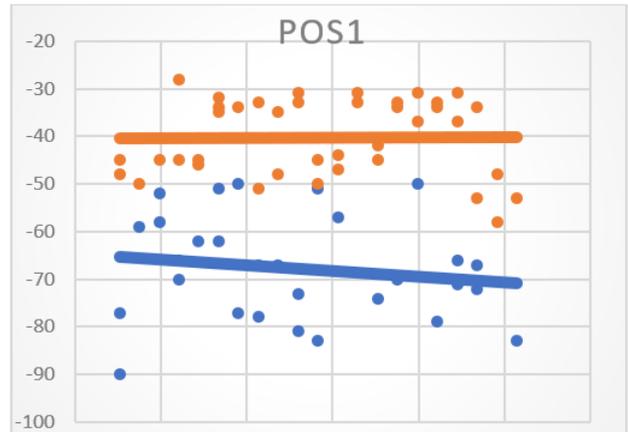
PRUEBA 7	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	100 ms
ESCÁNER	RPi1
POS1 (NARANJA)	RPi3
POS5 (AZUL)	RPi2



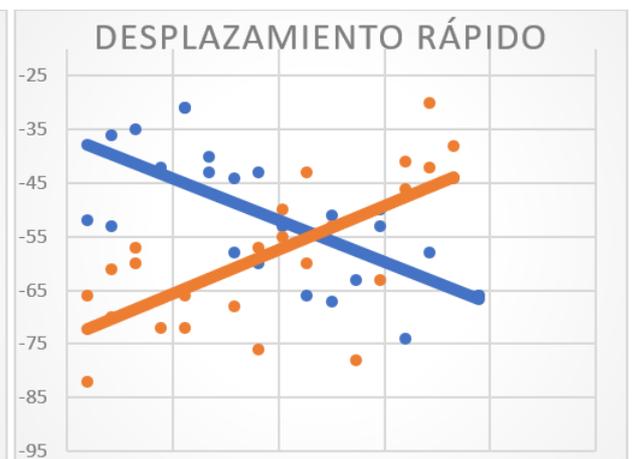
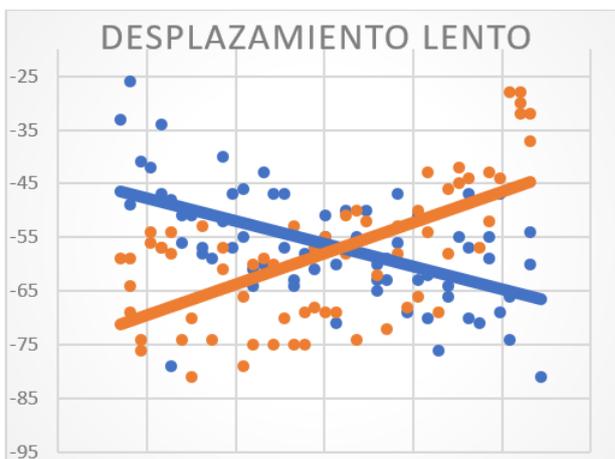
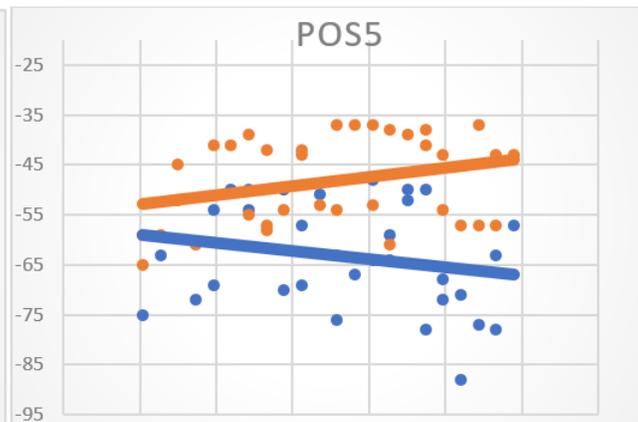
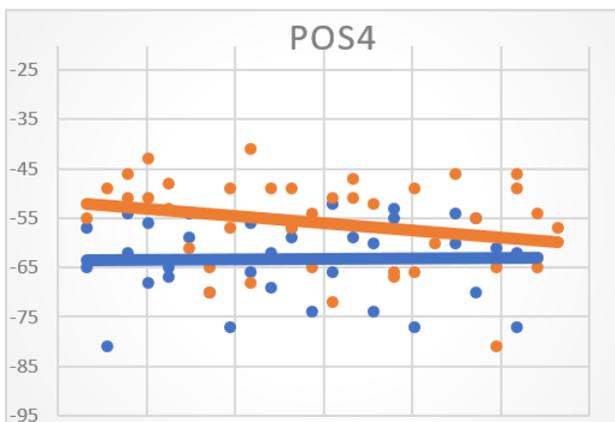
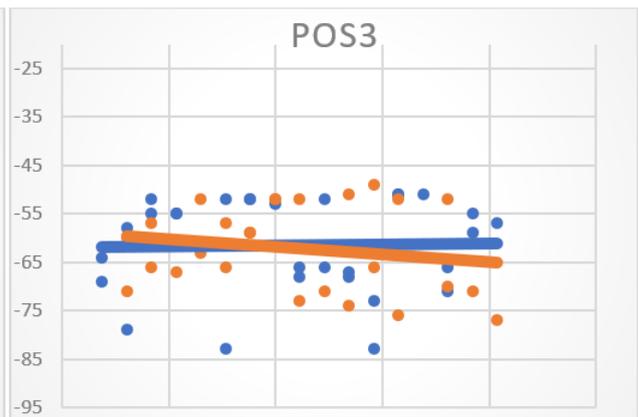
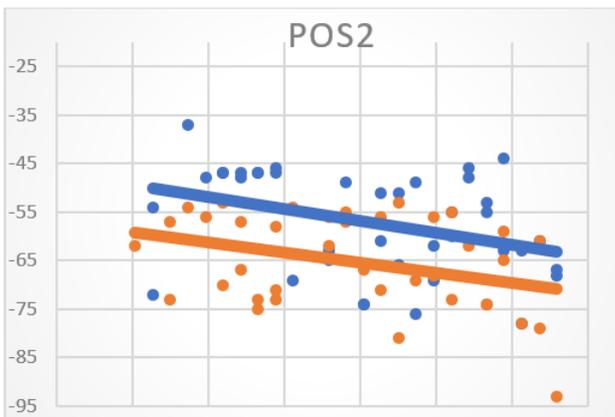
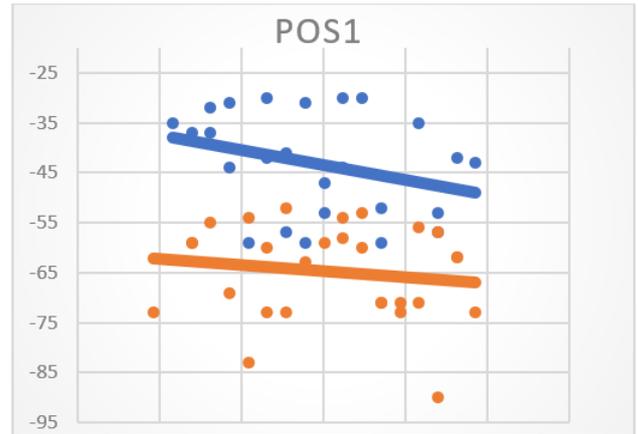
PRUEBA 8	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	Configurado en 100 ms (Pero se descartan 4 de cada 5 muestras)
ESCÁNER	RPi1
POS1 (NARANJA)	RPi3
POS5 (AZUL)	RPi2



PRUEBA 9	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	RPi1
POS1 (NARANJA)	RPi3
POS5 (AZUL)	RPi2

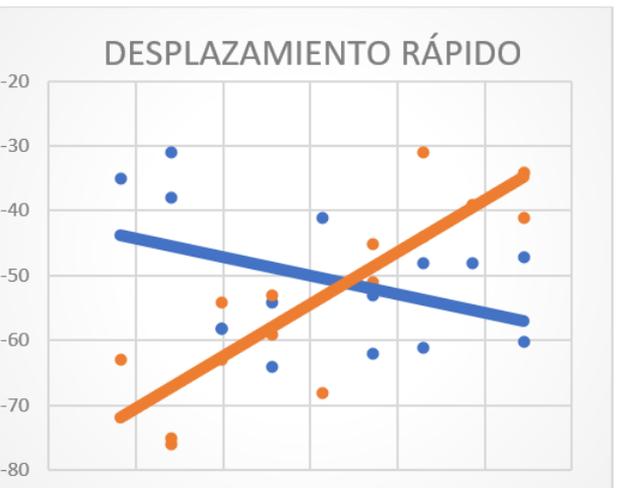
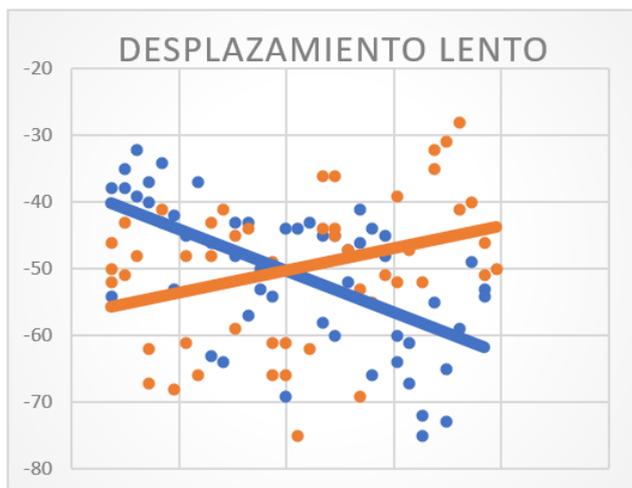
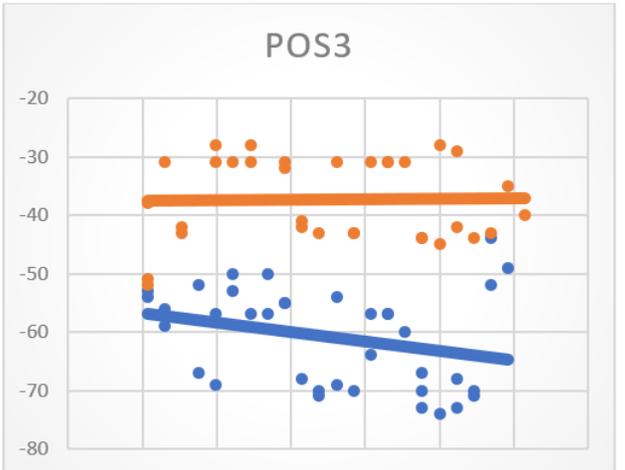
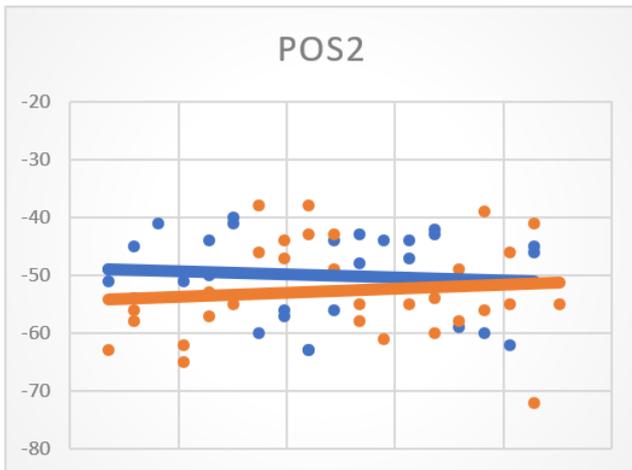
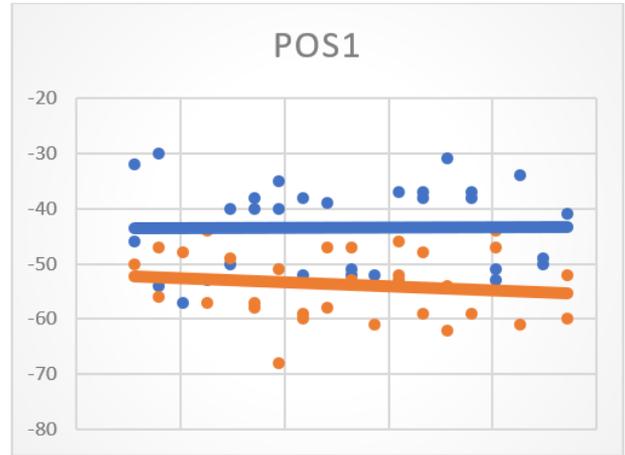


PRUEBA 10	
POSICIÓN RPi	HORIZONTAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi4
POS5 (NARANJA)	RPi5

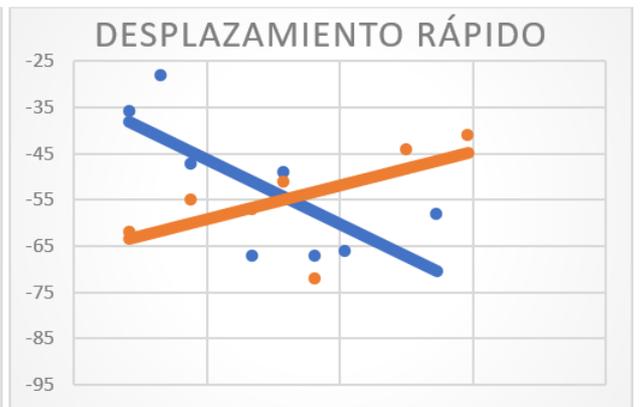
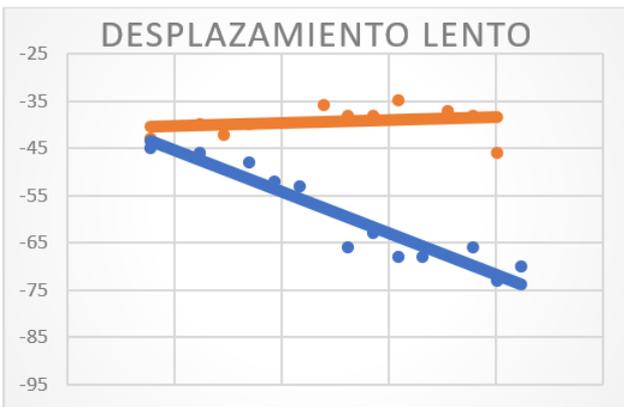
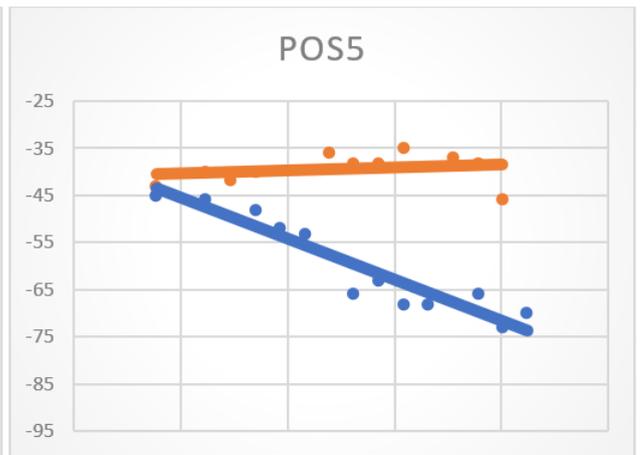
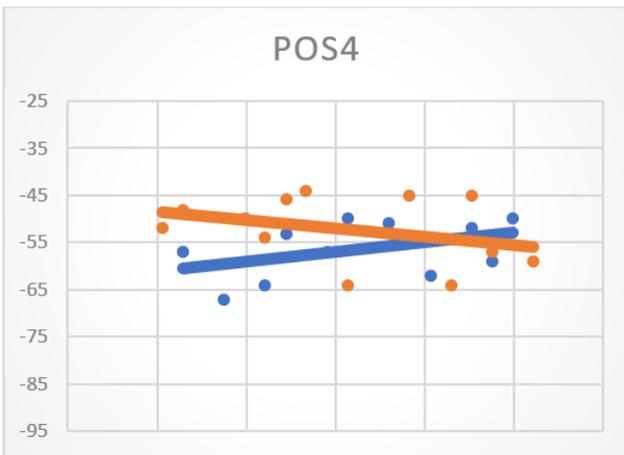
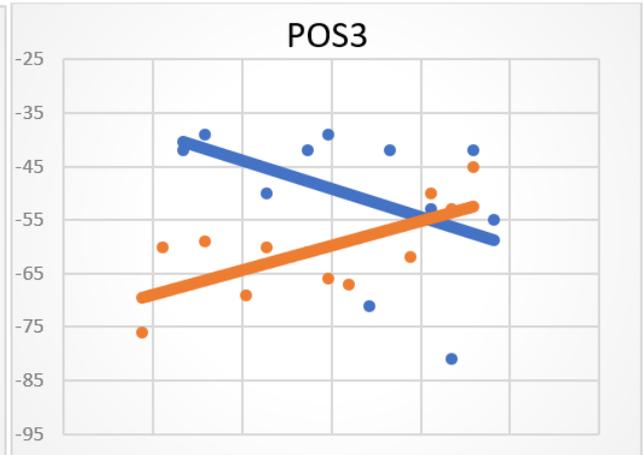


ANEXO VII: RESULTADOS DE LA FASE DE EXPERIMENTACIÓN. *EXPERIMENTO 2*

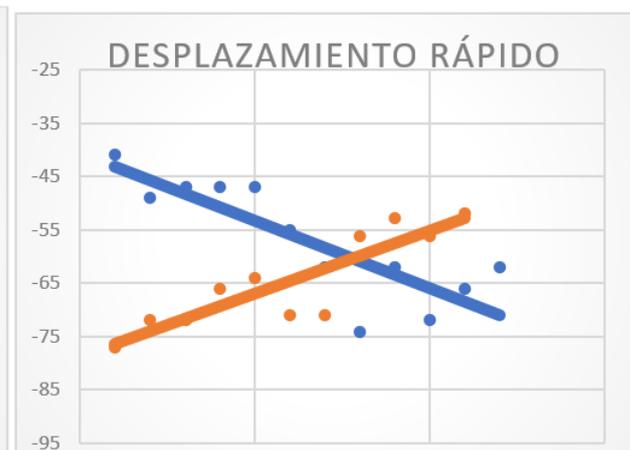
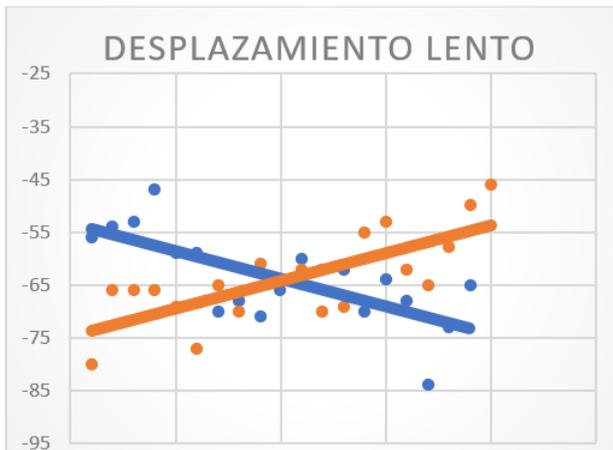
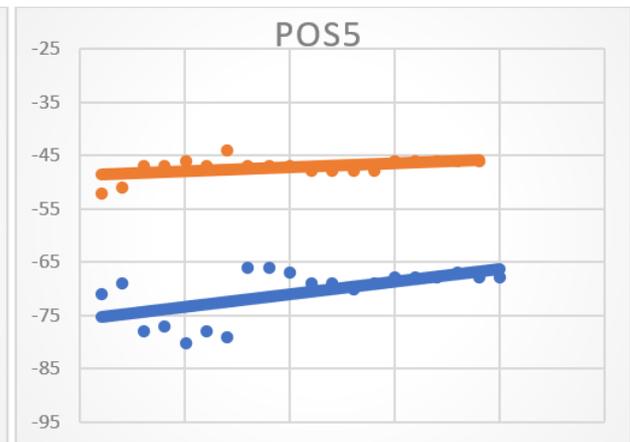
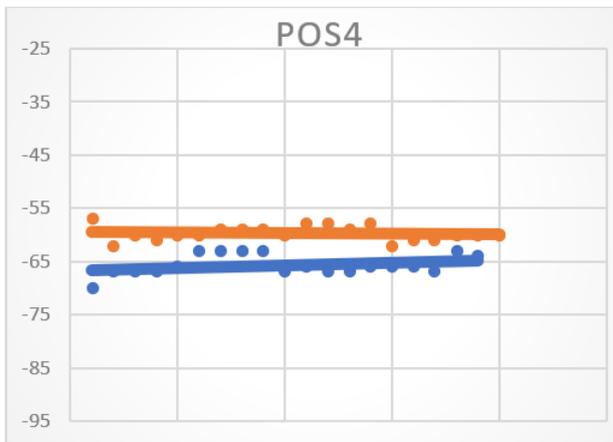
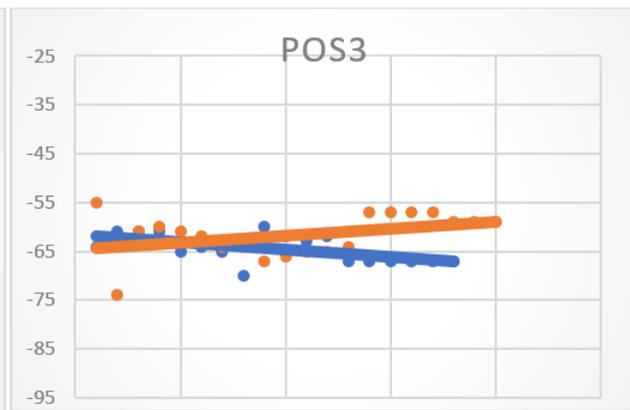
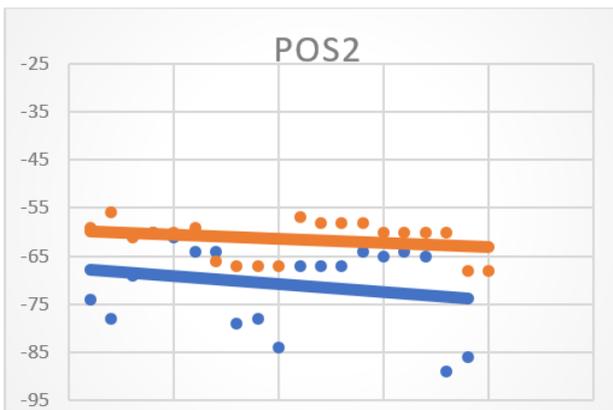
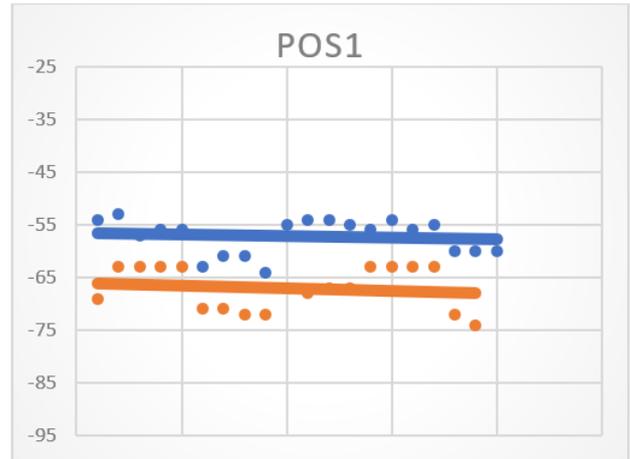
PRUEBA 1	
LOCALIZACIÓN	SEMINARIO 1
POSICIÓN RPi	VERTICAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS3 (NARANJA)	RPi3



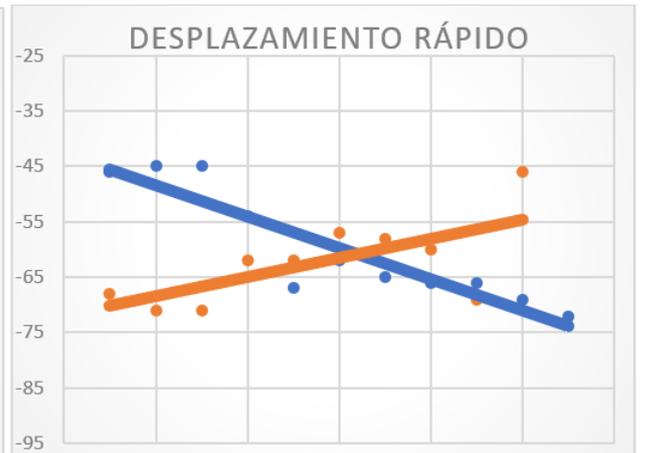
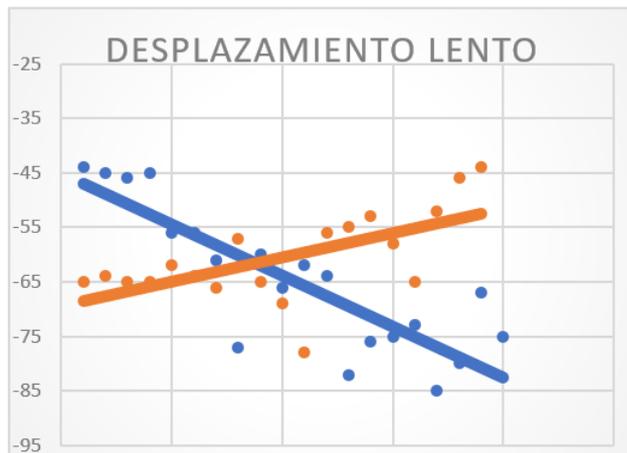
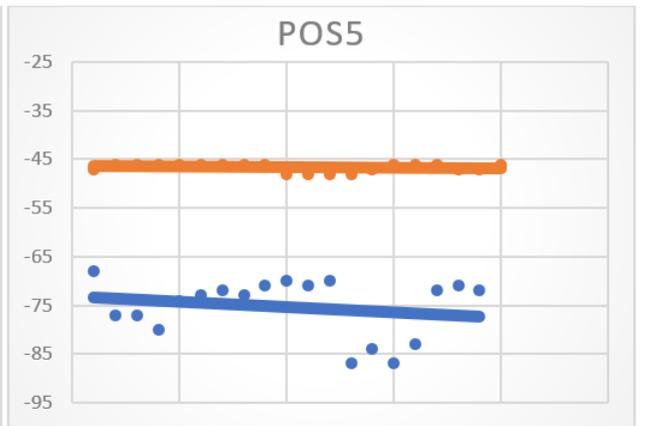
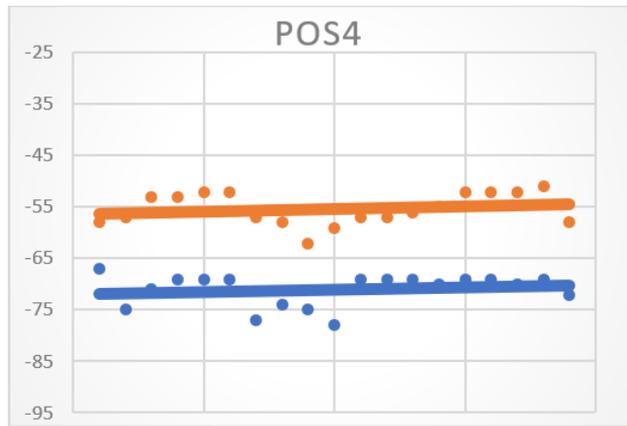
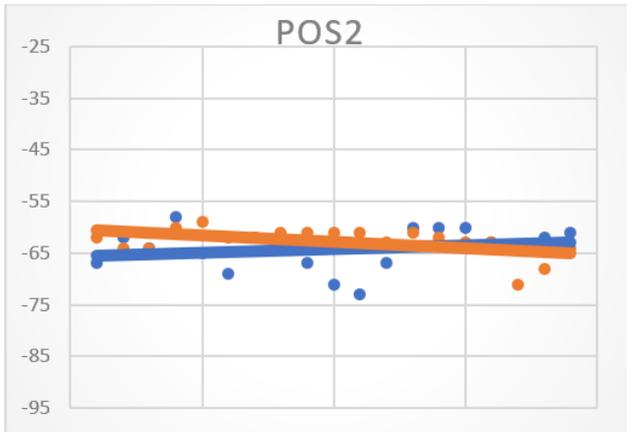
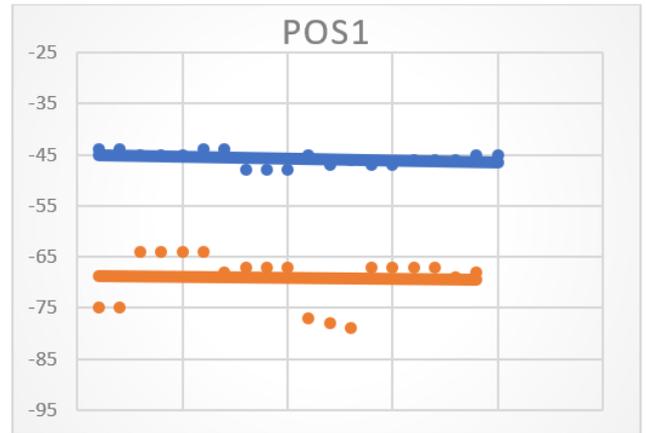
PRUEBA 2	
LOCALIZACIÓN	PASILLO
POSICIÓN RPi	VERTICAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	RPi1
POS1 (AZUL)	RPi2
POS3 (NARANJA)	RPi3
CONDICIÓN ESPECIAL	WiFi de las balizas activado



PRUEBA 3	
LOCALIZACIÓN	PASILLO
POSICIÓN RPi	VERTICAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	MÓVIL
POS1 (AZUL)	RPi2
POS3 (NARANJA)	RPi3
WiFi y datos móvil	Desactivado
WiFi RPi's	Activado



PRUEBA 4	
LOCALIZACIÓN	PASILLO
POSICIÓN RPi	VERTICAL
ALTURA	MENOR
PERIODO	500 ms
ESCÁNER	MÓVIL
POS1 (AZUL)	RPi2
POS3 (NARANJA)	RPi3
WiFi y datos móvil	Desactivado
WiFi RPi's	Desactivado



ANEXO VIII: RESULTADOS DE LA FASE DE EXPERIMENTACIÓN. *EXPERIMENTO 4*

	J4	J5	PASJ	HALL	G1	G2	G3	G4
RPi2	-87,773959	-84,488664	-82,592504	-68,705863	-77,381379	-70,545776	-61,077947	-51,239374
RPi3	-81,836249	-76,543986	-78,40174	-61,614163	-81,382634	-76,498332	-73,609393	-78,341703
RPi4	-90	-90	-87,372756	-76,240704	-90	-90	-83,870462	-90
RPi5	-80,685184	-77,585391	-83,196694	-66,804938	-42,645389	-45,05287	-54,780944	-61,948121
	J4	J5	PASJ	HALL	G1	G2	G3	G4
RPi2	2,22604144	5,5113363	7,40749562	21,2941368	12,6186206	19,4542244	28,922053	38,7606256
RPi3	8,16375057	13,4560135	11,5982595	28,3858369	8,61736638	13,5016685	16,3906073	11,6582968
RPi4	0	0	2,62724438	13,759296	0	0	6,12953842	0
RPi5	9,31481599	12,4146093	6,8033062	23,1950616	47,3546112	44,94713	35,2190563	28,0518788
	PASG	L1	L2	PASL	K3	K2	K1	PASK
RPi2	-57,697161	-76,880281	-79,959466	-82,221933	-90	-85,863971	-80,453579	-84,97681
RPi3	-66,289013	-74,932993	-80,690035	-79,084662	-64,069197	-62,64564	-56,225962	-60,419075
RPi4	-78,604959	-90	-90	-80,510033	-40,924289	-51,520906	-63,487063	-57,817374
RPi5	-57,62336	-82,532494	-89,198557	-82,849321	-87,491903	-84,901266	-82,667313	-84,108171
	PASG	L1	L2		K3	K2	K1	
RPi2	32,3028393	13,1197188	10,0405342	7,77806673	0	4,13602938	9,54642118	5,02318977
RPi3	23,7109866	15,0670068	9,30996471	10,9153384	25,9308028	27,3543598	33,7740377	29,5809248
RPi4	11,3950414	0	0	9,48996654	49,075711	38,4790938	26,5129372	32,1826255
RPi5	32,3766405	7,46750613	0,80144259	7,1506791	2,50809734	5,09873393	7,33268719	5,89182909