



**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
DETECCIÓN DE PROXIMIDAD ORIENTADO A
ESTABLECIMIENTOS COMERCIALES, BASADO EN UN
SISTEMA DE LOCALIZACIÓN (LS) UTILIZANDO TECNOLOGÍA
BLUETOOTH DE BAJA ENERGÍA (BLE) E INTELIGENCIA**

ARTIFICIAL

PRESENTADA POR

GUSTAVO ENRIQUE VIZCARDO GENSOLLEN

MIGUEL ANGEL YANCE ECHEGARAY

ASESORES

FERNANDO MAURICIO JIMENEZ MOTTE

ELVIS RIVERA RIVERA

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO

LIMA – PERÚ

2017



**Reconocimiento - No comercial - Sin obra derivada
CC BY-NC-ND**

Los autores permiten que se pueda descargar esta obra y compartirla con otras personas, siempre que se reconozca su autoría, pero no se puede cambiar de ninguna manera ni se puede utilizar comercialmente.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



USMP
UNIVERSIDAD DE
SAN MARTÍN DE PORRES

FACULTAD DE
INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
DETECCIÓN DE PROXIMIDAD ORIENTADO A
ESTABLECIMIENTOS COMERCIALES, BASADO EN UN
SISTEMA DE LOCALIZACIÓN (LS) UTILIZANDO
TECNOLOGÍA BLUETOOTH DE BAJA ENERGÍA (BLE) E
INTELIGENCIA ARTIFICIAL**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO**

PRESENTADA POR

VIZCARDO GENSOLLEN GUSTAVO ENRIQUE

YANCE ECHEGARAY MIGUEL ANGEL

Lima - Perú

2017

ÍNDICE

	Página
RESUMEN	xi
ABSTRACT	xii
INTRODUCCIÓN	xiii
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Antecedentes	1
1.2 Bases teóricas	4
1.3 Modelamiento	12
CAPÍTULO II. METODOLOGÍA	55
1.4 Materiales	55
CAPÍTULO III. DESARROLLO DEL PROYECTO	60

2.1 Análisis	60
2.2 Diseño	78
CAPÍTULO IV. PRUEBAS Y RESULTADOS	94
CAPÍTULO V. DISCUSIÓN Y APLICACIONES	113
CONCLUSIONES	115
RECOMENDACIONES	117
FUENTES DE INFORMACIÓN	
Error! Bookmark not defined.	

ÍNDICE DE FIGURAS

	Página
Figura 1. Gráfico estadístico de como las empresas centran su publicidad	2
Figura 2. Gráfico estadístico de la preocupación de los consumidores	3
Figura 3. Método de filtrado de Kalman ordinario	11
Figura 4. Dos correcciones del resultado del filtrado de Kalman	11
Figura 5. Intensidad de RSSI según sus límites	14
Figura 6. Modelo de radio propagación en espacio libre	15
Figura 7. Símbolo de Bluetooth	15
Figura 8. Los 79 canales en bluetooth	16
Figura 9. Los 40 canales de Bluetooth Low Energy	20
Figura 10. Beacon marca Estimote	20
Figura 11. Estructura del paquete de publicidad conformado por 47 bytes	22

Figura 12. Captura del segmento datos del paquete de Publicidad iBeacon, programa Packet Sniffer de Texas Instruments	23
Figura 13. Captura del segmento datos del paquete de Publicidad iBeacon, programa Packet Sniffer de Texas Instruments	24
Figura 14. Segmento datos del paquete de Publicidad de iBeacon	24
Figura 15. Neurona Biológica	30
Figura 16. Arquitectura de Neurona Artificial	30
Figura 17. Red neuronal de una capa	31
Figura 18. Red neuronal de múltiples capas	32
Figura 19. Red de capa competitiva	32
Figura 20. Función identidad	33
Figura 21. Función escalón unitario binario	34
Figura 22. Función escalón unitario bipolar	34
Figura 23. Función sigmoide binaria	35
Figura 24, Función tangente hiperbólica	37
Figura 25. ReLU	39
Figura 26. Arquitectura genérica de un perceptrón multicapa	40
Figura 27. Esquema del modelo de rutas de Aruba Networks	49
Figura 28. Esquema del modelo de áreas de Aruba Networks	49
Figura 29. Diagrama pictórico del proyecto	52
Figura 30. Diagrama de bloques del proyecto	53
Figura 31. RSSI aleatorio	61

Figura 32. RSSI con filtro de media aritmética de orden 2	61
Figura 33. RSSI con filtro de media aritmética de orden 3	62
Figura 34. RSSI con filtro de media aritmética de orden 4	62
Figura 35. RSSI con filtro de media aritmética de orden 5	63
Figura 36. RSSI con filtro de media aritmética de orden 10	63
Figura 37. RSSI con filtro de media aritmética de orden 15	64
Figura 38. Desviación estándar vs. Orden de filtro media aritmética	64
Figura 39. RSSI con filtro de mediana de orden 2	65
Figura 40. RSSI con filtro de mediana de orden 3	66
Figura 41. RSSI con filtro de mediana de orden 4	66
Figura 42. RSSI con filtro de mediana de orden 5	67
Figura 43. RSSI con filtro de mediana de orden 10	67
Figura 44. RSSI con filtro de mediana de orden 15	68
Figura 45. Desviación estándar vs. Orden de filtro de mediana	68
Figura 46. RSSI con filtro de media móvil de orden 2	69
Figura 47. RSSI con filtro de media móvil de orden 3	69
Figura 48. RSSI con filtro de media móvil de orden 4	70
Figura 49. RSSI con filtro de media móvil de orden 5	70
Figura 50. RSSI con filtro de media móvil de orden 10	71
Figura 51. RSSI con filtro de media móvil de orden 15	71
Figura 52. Desviación estándar vs orden de filtro de media móvil	72

Figura 53. Perceptrón múltiple capa	73
Figura 54. Plano del laboratorio de ingeniería eléctrica 06-A-3	78
Figura 55. Plano del laboratorio de ingeniería eléctrica 06-A-3, incluye las zonas	79
Figura 56. Área de trabajo con 2 beacons	81
Figura 57. Área de trabajo con 3 beacons, primera ubicación en el techo	82
Figura 58. Área de trabajo con 3 beacons, segunda ubicación en el techo	82
Figura 59. Área de trabajo con 3 beacons ubicados sobre los módulos	83
Figura 60. Área de trabajo con 4 beacons, ubicación en el techo	83
Figura 61. Número de neuronas en capa oculta vs error (2 neuronas de entrada)	86
Figura 62. Número de neuronas en capa oculta vs error (3 neuronas de entrada)	87
Figura 63. Número de neuronas en capa oculta vs error (4 neuronas de entrada)	87
Figura 64. Red neuronal final	89
Figura 65. Diagrama de bloques del proceso final de la detección de zona	90
Figura 66. Diagrama de flujo de la aplicación en Android (Cliente).	91
Figura 67. Diagrama de flujo de aplicación en C# (Servidor)	92
Figura 68. Sub diagrama de flujo de aplicación en C# (Servidor)	93
Figura 69. RSSI obtenido en las diferentes "Zonas"	94
Figura 70. RSSI obtenido luego de modificar la frecuencia de los beacons	95
Figura 71. RSSI con filtro de media móvil de orden 2	96

Figura 72. RSSI con filtro de media móvil de orden 3	96
Figura 73. RSSI con filtro de media móvil de orden 4	97
Figura 74. RSSI con filtro de media móvil de orden 5	97
Figura 75. RSSI con filtro de media móvil de orden 10	98
Figura 76. RSSI con filtro de media móvil de orden 15	98
Figura 77. Orden del filtro de media móvil vs Desviación estándar en Zona 1	99
Figura 78. Orden del filtro de media móvil vs Desviación estándar en Zona 2	99
Figura 79. Orden del filtro de media móvil vs Desviación estándar en Zona 3	100
Figura 80. Orden del filtro de media móvil vs Desviación estándar en Zona 4	100
Figura 81. Orden del filtro de media móvil vs Desviación estándar en Zona 5	101
Figura 82. Orden del filtro de media	101
Figura 83. RSSI con filtro de media móvil de orden 4	102
Figura 84. RSSI filtrado y normalizado	103
Figura 85. Disminución de error cuadrático medio por época de entrenamiento	104
Figura 86. Regresión de la red neuronal	104
Figura 87. RSSI obtenido por zona con datos sin filtrar	105
Figura 88. RSSI obtenido por zona con datos filtrados	106
Figura 89. Área de trabajo con 3 beacons, segunda ubicación en el techo	107
Figura 90. RSSI obtenido por zona con datos sin filtrar	107
Figura 91. RSSI obtenido por zona con datos filtrados	108
Figura 92. Área de trabajo con 3 beacons ubicados sobre los módulos	108

Figura 93. RSSI por zona con datos sin filtrar	109
Figura 94. RSSI obtenido por zona con datos filtrados	109
Figura 95. Error de ubicación obtenido en la Zona 1	110
Figura 96. Error de ubicación obtenido en la Zona 2	111
Figura 97. Error de ubicación obtenido en la Zona 3	111
Figura 98. Error de ubicación obtenido en la Zona 4	111
Figura 99. Error de ubicación obtenido en la Zona 5	112
Figura 100. Error de ubicación obtenido en la Zona 6	112

ÍNDICE DE TABLAS

	Página
Tabla 1. Clases de Bluetooth	17
Tabla 2. Comparación de la tasa de envío de datos por cada versión de Bluetooth	19
Tabla 3. Analogía de red neuronal biológica y red neuronal artificial	30
Tabla 4. Tabla de parámetros de la tesis	54
Tabla 5: Compuerta NAND	73
Tabla 6: Salidas de las neuronas en la capa oculta	74
Tabla 7: Actualización de pesos	76
Tabla 8. Neuronas de salida	77
Tabla 9: Error de red neuronal	88

RESUMEN

En este trabajo de investigación, se llevó a cabo un exhaustivo análisis sobre el problema de la localización en interiores (indoor). Existen diversas tecnologías que nos pueden ayudar a determinar la localización en interiores, en nuestro caso decidimos usar beacons marca Estimote que son transmisores Bluetooth de Baja Energía (BLE) que emiten señales broadcast (Es la transmisión de datos que serán recibidos por toda la red), a su vez decidimos diseñar un perceptrón múltiple capa a ser entrenado por la regla delta generalizada backpropagation (Propagación hacia atrás de errores o retropropagación) encargado de procesar la data RSSI (Received Signal Strength Indication) de los Beacons captada por el dispositivo móvil y de esta forma determinar la zona en la que nos posicionamos. Nuestras pruebas demostrarán que nuestra localización en interiores (indoor) alcanzará una precisión que se encuentre entre los 1 y 4 metros tal y como lo estipula el fabricante.

ABSTRACT

In this research, a thorough analysis was carried out on the problem of indoor location. There are several technologies that can help us to determine the location indoors, in our case we decided to use Beacons of the brand Estimote which are transmitters of Bluetooth Low Energy (BLE) that emit broadcast signals, in turn we decided to design a multiple perceptron layer to be trained by the generalized delta rule (backpropagation) in charge of processing the RSSI information of the Beacons captured by the mobile device and in this way determine the area in which we are positioned. Our tests will show that our indoor location will reach a precision in the range of 1 to 4 meters as indicated by Estimote company.

INTRODUCCIÓN

El tema de la presente tesis consiste en diseñar e implementar un prototipo de detección de proximidad entre consumidores y productos mediante tecnología Bluetooth de Baja Energía (BLE) e inteligencia artificial aplicado a supermercados para el envío de promociones y recopilación de información estadística del consumidor, de manera que, las empresas puedan realizar campañas de marketing más efectivas sin invadir la privacidad de sus consumidores.

En la tesis, se aplican los últimos avances tecnológicos de tendencia global, de manera que el costo para los consumidores y empresas se reduzca. Asimismo, no es un riesgo potencial para las personas, debido a que la tecnología utilizada cuenta con estándares internacionales del cuidado de la salud y pueda ser aplicada a cualquier tipo de establecimiento comercial.

La tesis comprende cuatro (4) capítulos. El primero aborda la investigación, descripción y análisis del estado del arte alrededor del mundo.

En el segundo, se explica el marco teórico, los materiales y metodología empleados para lograr el objetivo de la tesis, también se presentará el diseño y desarrollo empleados en la tesis. En el tercero, se muestran las pruebas realizadas, los resultados obtenidos y el diseño de cada parte del perceptrón múltiple capa a ser entrenada por la regla delta generalizada (backpropagation). Y en el último capítulo, se analizaron e interpretaron los resultados obtenidos en el perceptrón múltiple capa a ser entrenada por la regla delta generalizada (backpropagation) de tal manera que se pueda validar la propuesta de solución.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes

En la actualidad las empresas centran su publicidad basándose en los siguientes puntos Experian Data Quality (2013):

- Datos de contacto del cliente (54%).
- Cifras de ventas (44%).
- Demográficas (38%).
- Prioridades de los consumidores (31%).
- Comportamiento del consumidor (26%).
- Geolocalización (20%).

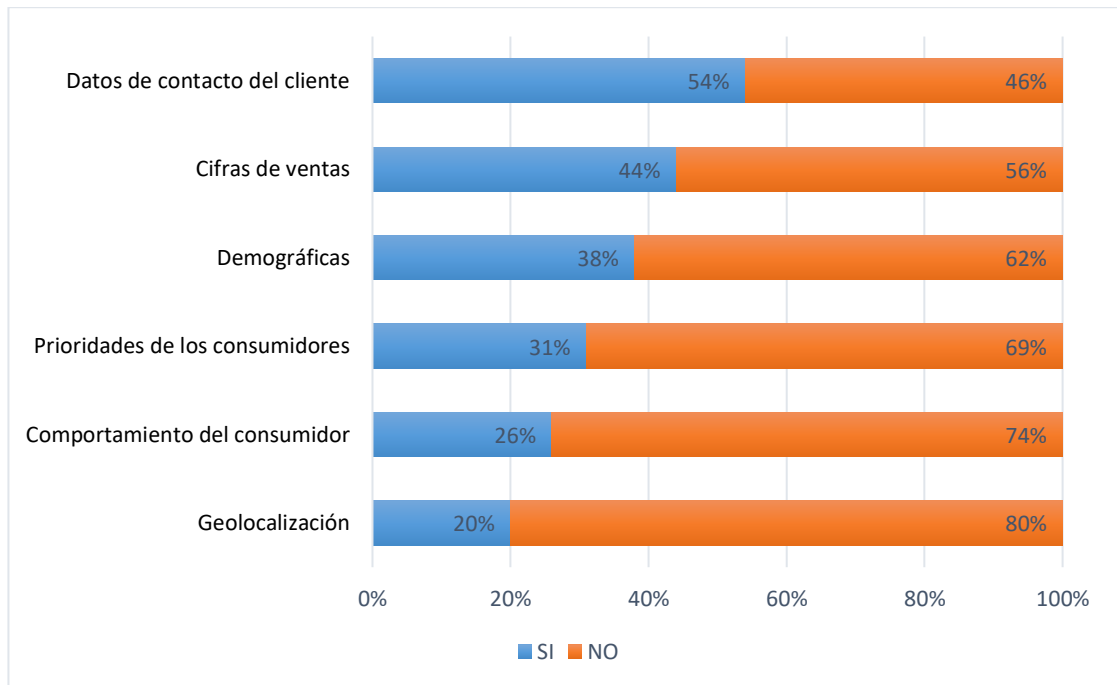


Figura 1. Gráfico estadístico de como las empresas centran su publicidad.

Elaboración: los autores

En el gráfico de la figura 1, se muestra que muy pocas empresas centran su publicidad en geolocalización, siendo totalmente opuesto con respecto a los datos de contacto del cliente, siendo así una táctica de marketing invasiva a la privacidad de los usuarios.

Existe una preocupación por parte de los consumidores con respecto a sus datos personales A. Khan & Davis (2016).

- No confío brindar mis datos a las tiendas (69%).
- Me siento espiado (67%).
- Las tiendas usarán mi información para su beneficio (61%).
- Los menores de edad con teléfono inteligente pueden ser rastrear sin el consentimiento de los padres (50%).

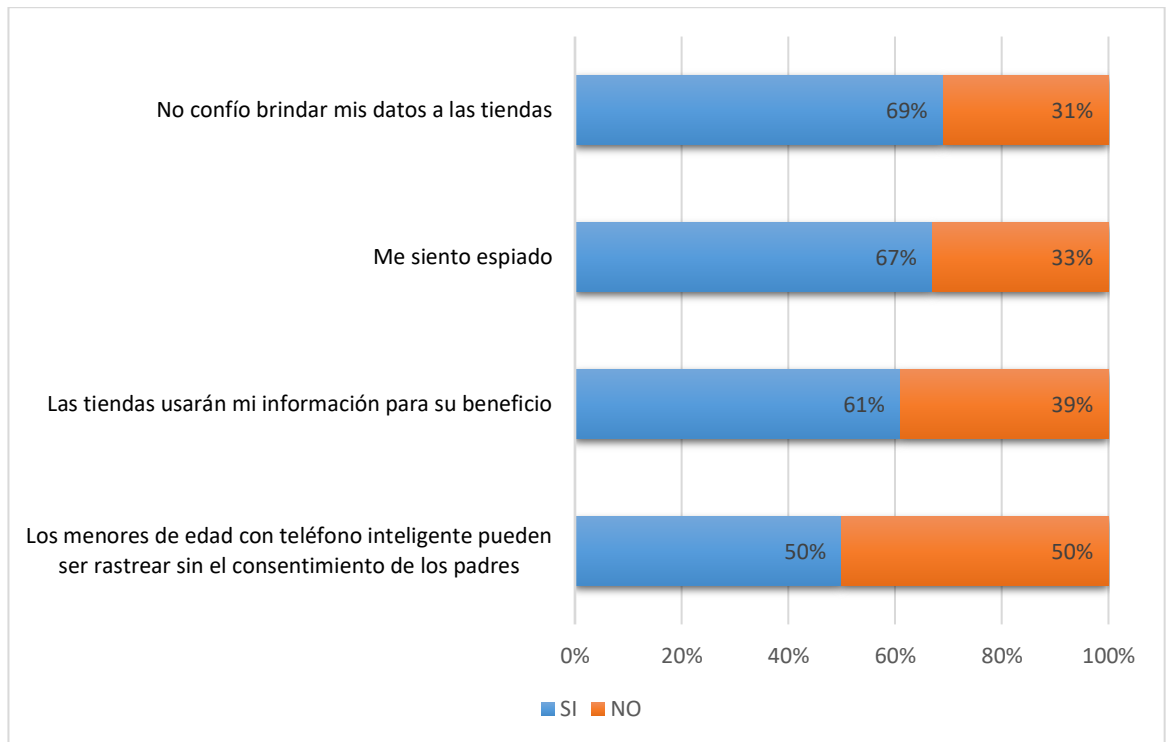


Figura 2. Gráfico estadístico de la preocupación de los consumidores.

Fuente: A. Khan & Davis (2016).

1.1.1 Definición del problema

La presente investigación brinda una alternativa para que las empresas obtengan información de los clientes, y así poder realizar campañas de marketing de manera más efectiva, obtener datos demográficos, de comportamiento y localización de los consumidores dentro de los establecimientos comerciales, cabe mencionar no supondrá una vulneración a los datos de los clientes.

Para reforzar la confianza de los consumidores hacia el uso de la solución propuesta, fueron almacenados los siguientes datos: nombre, edad y sexo; los demás datos propuestos, como lo son el comportamiento y localización de los consumidores serán obtenidos durante el funcionamiento de la solución propuesta.

1.2 Bases teóricas

1.2.1 Objetivo general:

Diseñar e implementar un prototipo de detección de zonas entre consumidores y productos mediante tecnología Bluetooth de Baja Energía (BLE) e inteligencia artificial aplicado a establecimientos comerciales para el envío de promociones y recopilación de información estadística del cliente.

1.2.2 Objetivos específicos:

- Diseñar el prototipo de detección de zonas mediante tecnología Bluetooth de Baja Energía (BLE) con una precisión no mayor a 3 metros.
- Diseñar un perceptrón múltiple capa a ser entrenado por la regla delta generalizada (backpropagation) que permita clasificar los RSSI recibidos dentro de zonas delimitadas.
- Implementar el prototipo en laboratorio simulando el área de trabajo final (establecimiento comercial).
- Diseñar una aplicación (App) para el cliente, permitiéndole recibir la promoción del producto de acuerdo a la ubicación dentro del establecimiento comercial, a través de la tecnología BLE.
- Diseñar un entorno gráfico en C# que permita recopilar información estadística del cliente, de las zonas con mayor demanda de visita y del tiempo de permanencia del cliente por zona.
- Lograr reducir la cantidad de beacons en la solución final, en comparación con una solución convencional de un beacon por zona.

1.2.3 Justificación:

Viabilidad técnica

A nivel tecnológico la tendencia global de los últimos cinco (5) años apunta a la fusión de tecnología Bluetooth de Bajo Energía (BLE), técnica de sensor data fusión e inteligencia artificial, dicha tecnología está desplazando a la tecnología RFID (*Radio Frequency Identification*) y Wi-Fi (tecnología de comunicación inalámbrica) en soluciones de localización en interiores.

En el estado del arte, podemos observar que empresas como Google Developers (2017) y Apple Developer (2017), están implementando soluciones con tecnología Bluetooth de Baja Energía (BLE).

Viabilidad económica

Una solución de este tipo puede implicar una inversión 6 dólares por metro cuadrado para el dueño del establecimiento comercial, sin embargo, soluciones de este tipo podrían producir un impacto de crecimiento sobre los ingresos del establecimiento comercial.

Viabilidad social

El proyecto no supone un riesgo potencial para la salud de las personas pues opera con dispositivos Bluetooth de Baja Energía (BLE) que cuenta con estándares internacionales del cuidado de la salud con la norma ISO/IEEE 11073-20601 de Dispositivo de Salud Personal, autores Avila & Reyes (2017) y Bluetooth SIG (2017).

Viabilidad operativa

Tanto los beacons, la aplicación (App) y el sistema en C# se puede adecuar a la solución y a cualquier tipo de establecimiento comercial, tanto el dispositivo electrónico y software no invadirán la privacidad de los clientes.

1.2.4 Estado del arte

El autor Muhammad (2007) indicó que la predicción de localización móvil usando la propagación de pérdidas de trayectoria (intensidad de señal) es una tarea compleja y difícil, la exactitud depende del ambiente (multi trayectorias, línea de vista, ensombrecimiento), el modelo de pérdidas en el espacio, número de estaciones bases, técnicas, tecnologías utilizadas y las condiciones ambientales que afectan la propagación de las señales de radio. Muchas técnicas han sido propuestas recientemente, mayormente en métodos linealizados, geométricos o similares.

Modelar la propagación de señales de radio con modelos matemáticos es altamente complejo, envolviendo numerosas variables que interactúan. En adición, las multi trayectorias, difracción y la no línea de vista (NLOS) causan problemas; asimismo, las condiciones ambientales afectan la propagación de señales de radio.

Un enfoque alternativo basado en redes neuronales artificiales es utilizado en la investigación de Muhammad (2007), debido a las ventajas que posee como flexibilidad incrementada para adaptarse a diferentes ambientes y capacidad de procesar los datos en paralelo a alta velocidad.

El autor Hassan (2016) indicó que existen muchos factores que influyen en las mediciones de intensidad de señal, por lo que es lógico que los cálculos no se realizan en base a una sola medición, sino al resultado de filtrar un grupo de mediciones.

Las redes neuronales proporcionan un esquema de ambiente adaptativo, aumentando la robustez a cambios en el ambiente, el esquema de ambiente adaptativo mostro un buen desempeño y menor sensibilidad a variaciones en el espacio circundante.

Se utilizó la red neuronal como un clasificador para obtener la ubicación (habitaciones) de los móviles dentro de una casa. También se puede observar que el número de neuronas en la capa oculta afectaba en la

precisión del sistema mientras que la cantidad de capas ocultas no suponían una variación en la precisión del sistema más si en la carga computacional requerida para llevar a cabo el proceso de ubicación.

Los autores Kalbandhe & Patil (2016) realizaron una comparación entre las diferentes tecnologías utilizadas para realizar la localización en interiores.

La Medalla Activa (Active Badge) fue el primer intento usado para un sistema de localización. Esta medalla enviaba señales infrarrojas cada 10 segundos, notificando al usuario. Necesitaba línea de vista directa entre la medalla y el receptor y las señales infrarrojas tenían muy corto rango de alcance.

En el método RFID era necesario colocar múltiples receptores RFID en las distintas ubicaciones en las que se quería localizar al usuario. Su principal debilidad era el alto costo de implementación.

La tecnología WLAN (Wireless Local Area Network) es ampliamente utilizada en donde la intensidad de señal es censada constantemente. Ha sido implementada en aeropuertos, centros comerciales, librerías, estaciones de metro, etc., en donde se utilizó la infraestructura de Wi-Fi previamente desplegada. Un punto débil de dicha tecnología es la dependencia de dicha infraestructura y se vuelve incompatible en lugares en los que dicha red no ha sido implementada. Dicha tecnología se ha utilizado ampliamente con dos métodos diferentes para obtener la ubicación, las metodologías basadas en la propagación y fingerprinting.

- Metodologías basadas en propagación, utilizan la triangulación o trilateración identificándose con transmisores en ubicaciones conocidas, estos métodos son altamente predispuestos al error en ubicaciones de interiores al tomar en cuenta la naturaleza dinámica de las señales de radio.

- Metodologías basadas en fingerprinting: Evalúan la posición creando vastos arreglos de información que son acumulados y una investigación y exploración a profundidad son realizadas para lograr obtener una guía para la ubicación.

Los autores también proponen un sistema de posicionamiento en interiores basado en RSSI utilizando bluetooth de bajo consumo. El sistema es eficiente en cuanto a la energía utilizada, con otras tecnologías como Wi-Fi, GSM (Global System for Mobile communications) o GPS (Global Positioning System).

Los autores Wang, Ye, Cheng, & Wang (2015) brindaron un ejemplo que la localización en interiores es usada para estudiar el interés y comportamiento de los consumidores dentro de una tienda por departamentos. Con un entendimiento profundo del comportamiento del usuario, la tienda puede cambiar su estructura y la ubicación de los productos para satisfacer a sus consumidores.

Los autores utilizan la tecnología BLE debido a su bajo consumo energético y su amplia adopción. Esto se traduce en que dicha tecnología pueda tener un amplio tiempo de vida, así como también puede ser utilizado en muchas aplicaciones y plataformas diferentes.

Los autores Razavi, Valkama, & Lohan (2015) observaron que una de las técnicas más utilizadas para realizar el cálculo de ubicación es la basada en fingerprinting usando la intensidad de señal recibida (RSSI). Una de las ventajas más claras de utilizar la localización basada en RSSI es la facilidad de su implementación en una multitud de dispositivos móviles a nivel de la Interfaz de Programación de Aplicación (API) debido a que el nivel de RSSI es directamente accesible a través de dicha interfaz. El punto débil del fingerprinting es su necesidad de almacenar y transmitir una gran cantidad de datos de ubicación, esto se vuelve más grave cuando se tiene

en cuenta los recursos limitados de los dispositivos móviles (memoria, energía y capacidades computacionales).

Un método alternativo para calcular la localización de un dispositivo móvil es la localización por pesos de centroides (WCL) que disminuye considerablemente la dificultad y tiempo necesarios para realizar la localización. Este método conlleva la desventaja de disminuir considerablemente la precisión del sistema.

Los autores introducen en la investigación un nuevo método basado en clústeres k-medios en tecnología Wi-Fi, su modelo permite tener estimaciones de niveles en un edificio con precisiones similares a las del método de fingerprinting pero reduciendo considerablemente la complejidad y velocidad del algoritmo. Así mismo, lograron disminuir el tamaño de la base de datos almacenada transmitiendo solo las cabeceras de los clústeres y sus etiquetas de pisos correspondientes.

Los autores Zafari & Papapanagiotou (2015) observaron que usar la tecnología Bluetooth de Baja Energía (BLE) también conocido como bluetooth inteligente (Bluetooth Smart), particularmente diseñado para mantener un bajo consumo de energía y baja latencia manteniendo un alto rendimiento, pero obteniendo como resultado un menor rango de trabajo.

Los autores utilizaron un filtro de partículas (Particle filtering) pudiendo disminuir el error hasta un mínimo de 0.27 metros, indicaron que al utilizar solo el filtro por partículas se pudo obtener un mejor desempeño que utilizando filtros de Kalman, al mismo tiempo el algoritmo se vuelve computacionalmente más eficiente.

Para reducir las fluctuaciones del RSSI, la aplicación continuamente promedia los RSSI recibidos y usa estos promedios de RSSI para calcular las distancias entre los Beacon y el dispositivo móvil a través del modelo de ensombrecimiento normal-logarítmico. Las pruebas se realizaron en un área de 11x6 metros, se utilizaron diferentes cantidades de

Beacons (entre 3 y 6 Beacons, aumentando de uno en uno) y partículas (entre 400 y 2000, con un incremento de 200 por prueba), en los resultados obtuvieron que el comportamiento ideal del sistema era obtenido con 5 Beacons y 1000 partículas, mientras que el peor desempeño era obtenido con 3 Beacons y 400 partículas. Los autores pudieron observar que un mayor número de Beacons no aumentaba significativamente el desempeño del sistema y que la interferencia entre estos aumentaba en gran medida debido al tamaño del área estudiada.

Los autores Guo et al. (2016) propusieron un algoritmo de localización en interiores basado en filtros de Kalman, específicamente en el método de medidas descartadas y el método de nueva ganancia, con la meta de reducir los problemas de interferencia y pérdida de línea de vista (NLOS).

Los autores utilizaron el método de tiempo de llegada (TOA) con 3 estaciones base para poder obtener la ubicación del dispositivo móvil, con este método, la precisión está altamente influenciada por la precisión de los relojes de los dispositivos, en el caso particular de los autores, la precisión mínima de sus relojes era de 10 ns, obteniendo un error mínimo de localización de 3m.

Los autores realizaron pruebas en un espacio de 20x30 metros y compararon resultados antes y después de aplicar el filtro de Kalman con lo que obtuvieron los siguientes resultados:

- Antes de aplicar el filtro de Kalman obtuvieron un error de ubicación mínimo de 3 m.

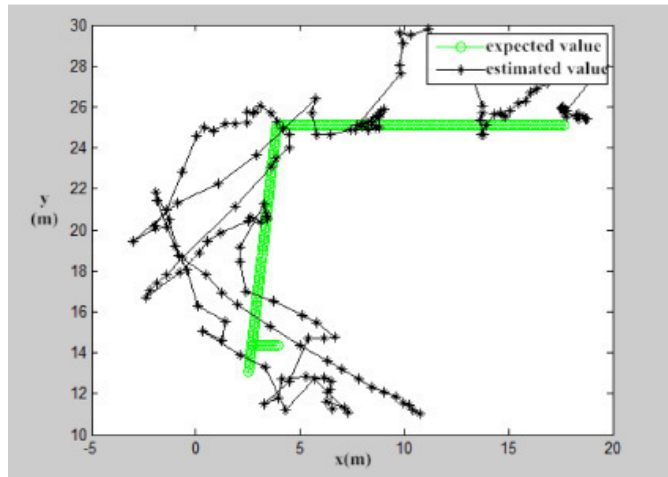


Figura 3. Método de filtrado de Kalman ordinario.

Fuente: Guo et al. (2016).

- Luego de aplicar el filtro de Kalman obtuvieron un error de ubicación mínimo de 1.35 m.

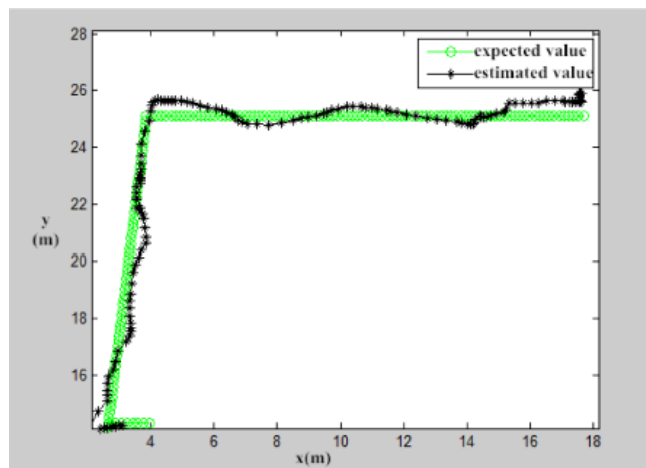


Figura 4. Dos correcciones del resultado del filtrado de Kalman.

Fuente: Guo et al. (2016).

Los autores M. Khan, Kai, & Gul (2017) utilizaron la tecnología Wi-Fi debido a que permite ahorrar costos y esfuerzos de implementación. Compararon dos métodos diferentes para obtener la ubicación del dispositivo móvil:

- Trilateración: Se pudo observar que era muy sensible al ruido externo.

- Fingerprinting: Se pudo observar que la precisión de la ubicación era altamente susceptible a cambios en el ambiente implementado.

Los autores propusieron un sistema híbrido, combinando ambos métodos y adicionando un filtro de Kalman llamado Unscented Kalman Filter obteniendo como resultado un sistema con un mayor grado de precisión y robustez considerando necesario el mayor grado de carga computacional para alcanzar dicho objetivo. El filtro Unscented Kalman Filter fue usado debido a su mejor desempeño en sistemas altamente no lineales, su fácil implementación y modificación fue ideal para realizar prototipos.

1.3 Modelamiento

1.3.1 RSSI

Según los autores Dong & Dargie (2012) el RSSI representa la relación que existe entre el poder transmitido y recibido, esta relación es empleada para medir la distancia de separación entre un transmisor y un receptor cuando la onda electromagnética se propaga en un enlace de línea de vista.

Si hay un enlace entre dos nodos y en el entorno no se produce ninguna interferencia de la señal, la potencia de la señal recibida (P_r) está relacionada con la distancia (d), entre los nodos de transmisión y recepción en la ley del cuadro inverso.

$$P_r \propto d^{-2} \quad (1)$$

La ecuación 1 expresa una relación ideal entre el RSSI y la distancia relativa. En el mundo real existen muchos factores que alteran el valor de intensidad de la señal recibida, como por ejemplo la reflexión, la refracción, la difracción, la dispersión a causa de los objetos en el ambiente

cercano, etc. Debido a estas múltiples alteraciones de la señal de radio, la potencia recibida puede disminuir alterando la relación P_r y d .

$$P_r \propto d^{-\gamma} \quad (2)$$

En la ecuación 2 γ , representa el exponente de pérdida, otro factor que afecta la potencia recibida, por lo tanto, se ve afectada la predicción de ubicación de la polarización de la antena. Para alcanzar la potencia máxima recibida, la antena del nodo receptor deberá contar con la misma orientación del nodo de transmisión. Las pérdidas causadas por una polarización de antena desalineada (L) se expresa:

$$L = 20 \log(\cos \theta) \quad (3)$$

El RSSI es el indicador de intensidad de la señal de radio en dBm, n es la constante de propagación de la señal, d es la distancia relativa entre los nodos de comunicación y A es una intensidad de señal recibida de referencia en dBm. El valor del RSSI medido cuando la distancia de separación entre el receptor y el transmisor son de un metro.

$$RSSI = -(10 \times n) \log_{10}(d) - A \quad (4)$$

En el sistema IEEE 802.11 se indica que, a mayor número representado por el RSSI, mayor será la intensidad de señal. Asimismo, se indica, que el RSSI está representado por valores negativos, por lo que mientras el número se acerque más a 0, mayor será la intensidad de señal. IEEE org. (2017).

En una escala de 0 a -80 RSSI:

0: señal ideal, difícil de lograr en la práctica.

-40 a -60: señal idónea con tasas de transferencia estables.

-60: enlace bueno; se puede lograr una conexión estable al 80%.

-70: enlace normal -bajo; es una señal medianamente buena, aunque se pueden sufrir problemas con lluvia y viento.

-80: es la señal mínima aceptable para establecer la conexión; puede ocurrir caídas, que se traducen en corte de comunicación (pérdida de llamada, pérdida de datos, mensajes (sms) corruptos (ilegibles)).

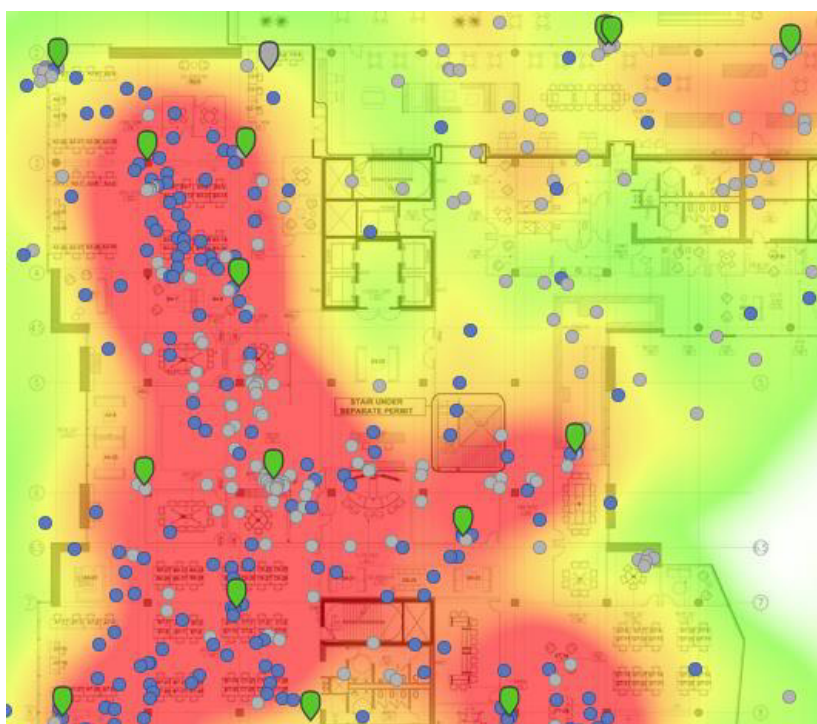


Figura 5. Intensidad de RSSI según sus límites.

Fuente: IEEE org. (2017).

En la figura 5, podemos observar un ejemplo de un plano en el cual se señala la ubicación de los puntos de acceso inalámbricos, los colores corresponden a la intensidad de RSSI propagado en todo el plano.

La idea de la estimación de la distancia por medio de RSSI se basa en el caso ideal del supuesto de que la intensidad de la señal recibida

es una función de la transmisión de energía y la distancia en el camino entre dos dispositivos de radio, como se muestra en la figura 6.

En la figura 6, también se puede apreciar, la relación ideal entre RSSI y distancia en espacio libre, la cual presenta su pico máximo en la distancia 0 metros con una intensidad de -10 dBm en cuanto al RSSI.

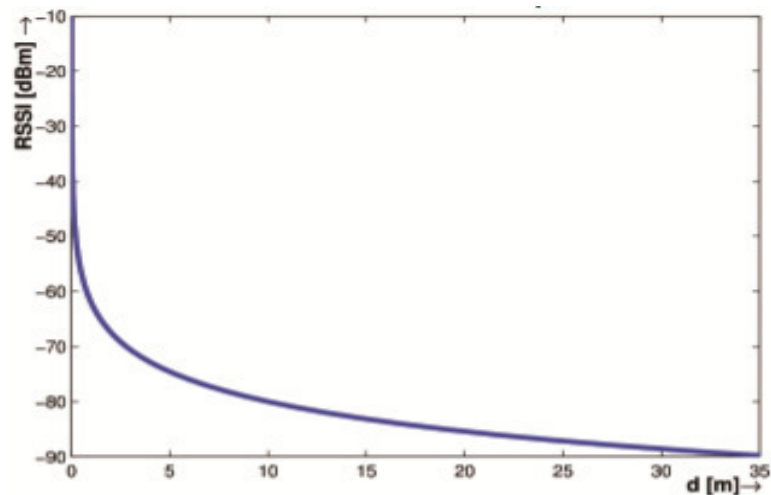


Figura 6. Modelo de radio propagación en espacio libre.

Fuente: IEEE org. (2017).

1.3.2 Tecnología Bluetooth



Figura 7. Símbolo de Bluetooth.

Fuente: www.bluetooth.com.

“Bluetooth es una tecnología de conectividad inalámbrica de baja potencia utilizada para transmitir audio, datos e información entre dispositivos” Bluetooth Technology Website (2017). El autor Muller (2001)

nos menciona que “La tecnología inalámbrica Bluetooth utiliza la banda de radio ISM (industrial, científico y medico) entre 2.4 - 2.484 GHz, la cual no requiere una licencia de operación”.

Con el Bluetooth ya no es necesario el cableado convencional lo que implica fallos, interferencia y gasto innecesarios, tampoco es necesario la compatibilidad entre hardware y software ya que se pueden comunicar diferentes dispositivos electrónicos.

La administración del Bluetooth es bastante simple, solo se requiere hacer un enlace entre los dispositivos electrónicos que se desean conectar entre sí y no es necesario ningún tipo de configuración de los equipos, solo se requiere activar el Bluetooth, en el dispositivo electrónico, y automáticamente, empieza la búsqueda de otros dispositivos en la zona de comunicación para que se pueda llevar a cabo la conexión.

Bluetooth utiliza tecnología llamada FHSS (Frecuency-Hopping Speed Spectrum) donde los datos transmitidos se dividen en paquetes y luego cada paquete es transmitido en uno de los 79 canales designados, el ancho de banda de cada canal es de 1 MHz empezando en la frecuencia 2402 MHz, por lo general se realizan 1600 saltos por segundo.

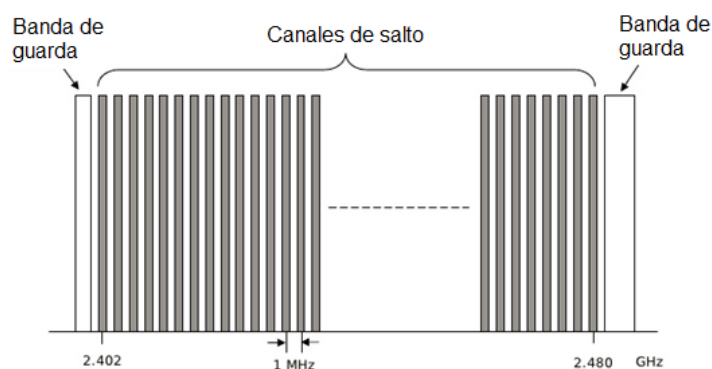


Figura 8. Los 79 canales en bluetooth.

Fuente: www.quora.com.

En el estándar Bluetooth 4.0 + LE, se utiliza un ancho de banda de 2 MHz; por lo tanto, se tienen 40 canales designados, se utiliza una variación de FHSS llamada AFH (Adaptive Frequency-Hopping Spread Spectrum) el cual omite canales con interferencia y brinda en una mejor comunicación.

Clases de Bluetooth

Tabla 1. *Clases de Bluetooth.*

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance (m)
Clase 1	100	20	100
Clase 2	2.5	4	10
Clase 3	1	0	1

Fuente: www.bluetooth.com.

Versiones de Bluetooth

Versión 1.1:

Fue lanzado en el año 1998 permitiendo la transmisión de datos con una distancia de un par de metros, las versiones 1.0 y 1.0b presentaron diversos problemas de comunicación entre las diferentes marcas y clases de equipos.

En el año 2002 fue lanzada la versión 1.1 donde se resolvieron los problemas en las versiones previas, esto permitió la transmisión simultánea de voz y datos con aproximadamente 1 Mbps de velocidad, también fue reconocido como un estándar de comunicación IEEE 802.15.

Versión 2.0 + EDR:

Fue lanzado en el año 2004 siendo compatible con la versión anterior, permite a los usuarios una conexión más fácil, detectando y conectando automáticamente. La principal diferencia es la tasa de datos mejorada EDR (Enhanced Data Rate) que brinda una mayor velocidad de transmisión de datos de 2 Mbps.

Versión 3.0 + HS:

Aprobada en el año 2009 incorpora la característica HS (High Speed) que permite una transmisión de datos en alta velocidad, de esta forma puede soportar velocidades de transmisión de datos de 24 Mbps, esto permite la transferencia de archivos de video y música. Presenta la desventaja de un mayor consumo de energía.

Versión 4.0:

La versión BLE fue lanzada en el año 2010, presenta diversas mejoras como el menor consumo de energía, mejora en la seguridad y aumenta la distancia de transmisión. Permite la transmisión de datos de 25 Mbps hasta 32 Mbps.

Versión 5.0

La última versión fue lanzada en el año 2017, tiene como principal ventaja que duplica la velocidad de las versiones anterior llegando a transmitir 100 Mbps, duplica la distancia de alcance llegando a los 200 metros y tener 8 veces más conexiones. Esta versión también soporta la baja energía introducida en la versión 4.0.

Tabla 2. Comparación de la tasa de envío de datos por cada versión de Bluetooth.

Versión	Tasa de envío de datos (Mbps)
Versión 1.2	1
Versión 2.0 + EDR	2
Versión 3.0 + HS	24
Versión 4.0	32
Versión 5.0	100

Fuente: www.bluetooth.com.

1.3.3 Bluetooth Low Energy (BLE)

“Bluetooth LE habilita conexiones inalámbricas de ráfaga corta y usa múltiples topologías de red, Broadcast es una topología de red que establece comunicaciones de dispositivo de uno a muchos (1:m). La topología de transmisión de Bluetooth LE optimiza el intercambio de información localizada, lo que la hace ideal para soluciones de baliza, como información de punto de interés (POI) y servicios de búsqueda de artículos y rutas”, fuente Bluetooth Technology Website (2017).

BLE utiliza la banda 2.4 GHz para transmitir, dicha banda se divide en 40 canales separados en 2 MHz. Estos se dividen en tres (3) canales de Publicidad (Advertising) y treinta y siete (37) canales de datos, los canales de publicidad se encargan de transmitir el mensaje a los demás dispositivos dentro del radio de propagación, los canales de datos se encargan de transmitir datos una vez exista conexión entre dispositivos. Los canales de publicidad se encuentran en las frecuencias 2402 MHz, 2426

MHz y 2480 MHz, los canales de datos se encuentran en intervalos de 2 MHz, según podemos observar en la figura 9.

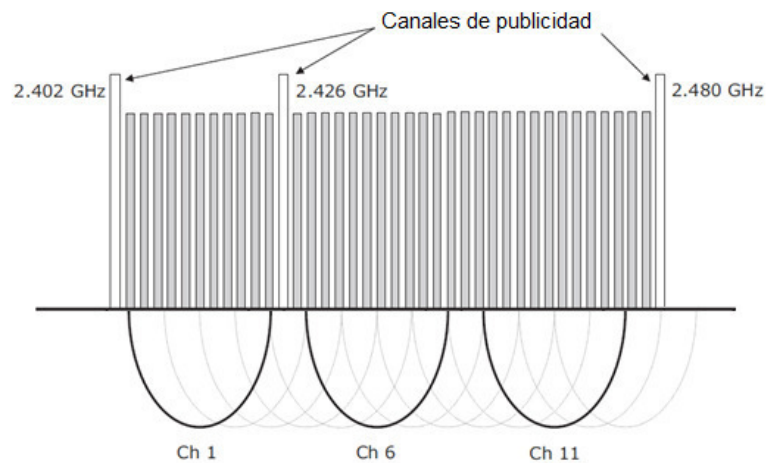


Figura 9. Los 40 canales de Bluetooth Low Energy.

Fuente: www.quora.com.

1.3.4 Beacon



Figura 10. Beacon marca Estimote.

Fuente: www.estimote.com.

El Beacon Kontakt.io (2013), Estimote (2012) y Developers (2012) son dispositivos electrónicos basados en tecnología Bluetooth Low Energy (BLE) que emite señales de radio llamadas advertising, dichas señales pueden ser recibidas por cualquier dispositivo equipado con tecnología Bluetooth 4.0, el cual no necesita estar emparejado con el Beacon para poder recibir la señal de advertising.

Cuando un dispositivo recibe el paquete de publicidad, puede calcularse la distancia y la posición del Beacon. Estas habilidades son bastante útiles para aplicaciones en lugares como museos, tiendas, hospitales, almacenes, etc.

El Beacon trabaja con una pila tipo botón CR2477, al poseer un bajo consumo de energía, puede funcionar durante 24 meses sin ningún inconveniente.

El alcance máximo de la señal de radio varía según la potencia de transmisión, también influye los dispositivos en el entorno y el lugar a instalar. La potencia y el intervalo de señal pueden ser configurados por el usuario.

Características principales:

- Difusión de paquetes publicitarios basado en Bluetooth LE 4.0.
- Compatibilidad con todos los dispositivos Bluetooth 4.0.
- Compatibilidad con los estándares Apple iBeacon y Google Eddystone.
- Parámetros configurables protegidos por contraseña:
 - UUID de proximidad, valores *Major* y *Minor*.
 - Nombre del dispositivo.
 - Nivel de potencia de transmisión.
 - Intervalos de publicidad.
- Actualización de firmware por aire.
- Bajo consumo de energía.
- Vida útil de la batería de hasta 24 meses.

Se pueden encontrar tres tipos de dispositivos: el Bluetooth que sólo soporta el modo "clásico", el Bluetooth Smart Ready que soporta el modo "clásico" y LE, y el Bluetooth Smart que sólo soporta el modo LE. Los nuevos dispositivos electrónicos están equipados con Bluetooth 4.0 por lo

tanto son "Bluetooth Smart Ready", los Beacons sólo soportan el protocolo de bajo consumo por lo tanto son "Bluetooth Smart".

Funcionamiento de la comunicación

La comunicación BLE consta de dos partes: Publicidad (Advertising) y Conectividad (Connecting).

El paquete de publicidad (advertising) es un mecanismo unidireccional, los dispositivos que buscan ser descubiertos se encargan de transmitir paquetes de información en intervalos de 20 ms a 10 segundos, los paquetes de Publicidad pueden tener hasta 47 bytes de longitud y constan de 1 byte de preámbulo, 4 bytes de dirección de acceso, entre 2 y 39 bytes de canal publicitario PDU y 3 bytes de CRC.

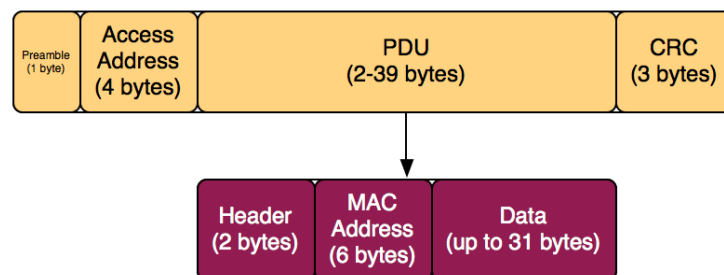


Figura 11. Estructura del paquete de publicidad conformado por 47 bytes.

Fuente: Adam Warski (2014).

Para el paquete de publicidad siempre la dirección de acceso (access address) es 0x8E89BED6 y el segmento datos siempre es diferente por cada conexión.

La PDU está conformada por 2 bytes para la cabecera, 6 bytes para la dirección MAC del dispositivo y 31 bytes de datos de información.

Los dispositivos BLE operan en modo de publicidad sin conexión, pero también pueden permitir conexiones.

BLE en los Beacons

Los Beacons solo utilizan el paquete de Publicidad (Advertising) para transmitir paquete de datos en intervalos regulares, dichos datos son recogidos por dispositivos móviles.

Si intentamos interceptar un paquete de Publicidad de iBeacon, en nuestro caso procedente de un Beacon marca Estimote, veremos los siguientes datos:

```
1 02 01 06 1A FF 4C 00 02 15 B9 40 7F 30 F5 F8 46 6E AF F9 25 55 6B 57 FE 6D 00 49 00 0A C5
```

Figura 12. Captura del segmento datos del paquete de Publicidad iBeacon, programa Packet Sniffer de Texas Instruments.

Fuente: Adam Warski (2014).

Según la figura 12, el segmento de datos del paquete de Publicidad nos muestra solo 30 bytes, lo que se ajusta al límite de 31 bytes.

El segmento de datos del paquete de publicidad iBeacon de Apple está conformado por 9 bytes del prefijo iBeacon, 16 bytes de UUID de proximidad, 2 bytes para el Major, 2 bytes para el Minor y 1 byte para el poder de transferencia.

1	02 01 06 1A FF 4C 00 02 15:	iBeacon prefix (fixed except for 3rd byte - flags)
2	B9 40 7F 30 F5 F8 46 6E AF F9 25 55 6B 57 FE 6D:	proximity UUID (here: Estimote's fixed UUID)
3	00 49:	major
4	00 0A:	minor
5	C5:	2's complement of measured TX power

Figura 13. Captura del segmento datos del paquete de Publicidad iBeacon, programa Packet Sniffer de Texas Instruments.

Fuente: Adam Warski (2014).

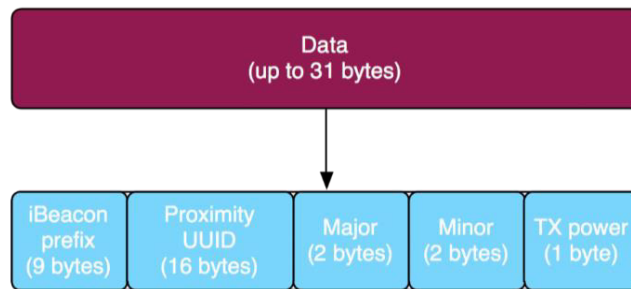


Figura 14. Segmento datos del paquete de Publicidad de iBeacon.

Fuente: Adam Warski (2014).

El UUID de proximidad sirve para identificar los Beacons entre ellos, el Major se utiliza para agrupar un conjunto de Beacons y el Minor se utiliza para identificar de forma individual respecto al conjunto de Beacons.

1.3.5 Tipos de filtros

Media aritmética

La media aritmética es el valor obtenido al sumar todos los datos y dividir el resultado entre el número total de datos.

La media aritmética se representa con el símbolo \bar{x} .

$$\bar{x} = \frac{a_1 + a_2 + \dots + a_n}{n} \quad (5)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i$$

Mediana (estadística)

La mediana representa el valor de posición central en un conjunto de datos ordenados de menor a mayor. La mediana se representa con el símbolo M_e .

Para calcular la mediana debemos seguir el siguiente procedimiento:

- Ordenamos los datos de menor a mayor.
- Si la serie tiene un número impar de medidas la mediana es la puntuación central.
- Si la serie tiene un número par de puntuaciones la mediana es la media entre las dos puntuaciones centrales.

Datos sin agrupar: Sean a_1, a_2, \dots, a_n los datos de una muestra ordenada de menor a mayor y designando la mediana como M_e , distinguimos dos casos:

- a) Si n es impar, la mediana es el valor que ocupa la posición $(n + 1)/2$ una vez que los datos han sido ordenados (en orden creciente o decreciente), porque este es el valor central. Es decir:

$$M_e = x_{(n+1)/2} \quad (6)$$

- b) Si n es par, la mediana es la media aritmética de los dos valores centrales. Cuando n es par, los dos datos que están en el centro de la muestra ocupan las posiciones $n/2$ y $(n/2) + 1$. Es decir:

$$M_e = \frac{\left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1} \right)}{2} \quad (7)$$

Moda (estadística)

La moda es el valor que tiene mayor frecuencia en un conjunto de datos. La moda se representa con el símbolo M_o .

Se puede hallar la moda para variables cualitativas y cuantitativas.

Hallar la moda de la distribución:

2, 3, 3, 4, 4, 4, 5, 5

$$M_o = 4$$

Si en un grupo hay dos o varias puntuaciones con la misma frecuencia y esa frecuencia es la máxima, la distribución es bimodal o multimodal, es decir, tiene varias modas.

1, 1, 1, 4, 4, 5, 5, 5, 7, 8, 9, 9, 9

$$M_o = 1, 5, 9$$

Cuando todas las puntuaciones de un grupo tienen la misma frecuencia, no hay moda.

2, 2, 3, 3, 6, 6, 9, 9

Si dos puntuaciones adyacentes tienen la frecuencia máxima, la moda es el promedio de las dos puntuaciones adyacentes.

0, 1, 3, 3, 5, 5, 7, 8

$$M_o = 4$$

Filtro media móvil (Moving Average Filters)

Según el autor Smith (1999) la media móvil es el filtro más común en DSP (Digital signal processor), principalmente porque es el más fácil de entender y usar. A pesar de su simplicidad, el filtro promedio móvil es óptimo para una tarea común: reducir el ruido aleatorio a la vez que se mantiene una respuesta de paso nítida. Esto lo hace el primer filtro para señales codificadas en el dominio del tiempo.

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} x(i+j) \quad (8)$$

Donde:

x: señal de entrada.

y: señal de salida.

M: número de muestras.

Filtro de Partículas

El filtro de partículas según el autor Martín Talavera (2012) es una técnica que implementa un filtro Bayesiano recursivo y tiene como objetivo realizar el seguimiento de una variable de interés y su evolución en el tiempo. El estado del sistema se compone por un conjunto finito de muestras (llamadas partículas) que, a su vez, cada una representa una hipotética posición ponderada. Cada cierto tiempo se obtienen muestras que comprueban la posición del móvil. El algoritmo del filtro de partículas es recursivo y opera en dos fases: predicción y actualización. En la fase de predicción, cada partícula se modifica de acuerdo con el modelo existente, incluyendo una aportación de ruido aleatorio para simular el efecto del ruido en la variable de interés. Luego, se pasa a la etapa de actualización donde el peso de cada partícula es evaluada basándose en la última información de observación

disponible. En cada iteración, las partículas con un peso muy bajo se eliminan y se realiza un nuevo muestreo.

Filtro de Kalman

En el año 1958, Kalman publicó el artículo "A new Approach to Linear Filtering and Prediction Problems", en dicho documento presenta un método de filtrado de información capaz de resolver problemas relacionados con valores aleatorios.

El filtro Kalman es un algoritmo de procesamiento de datos óptimo recursivo. Óptimo ya que minimiza un criterio determinado y porque incorpora toda la información que se le suministra para determinar el filtrado. Recursivo porque no precisa mantener los datos previos, lo que facilita su implementación en sistemas de procesamiento en tiempo real. Por último, algoritmo de procesamiento de datos, ya que es un filtro, pensado para sistemas discretos.

Este filtro se basa en estimar la cantidad de veracidad del fenómeno observado tanto por las medidas tomadas como por el conocimiento a priori del fenómeno citado. El procesamiento de los datos se lleva a cabo en dos etapas.

- Etapa de actualización, donde una función predice el estado del filtro de Kalman, proyectando que cantidad de información es verdad para futuras estimaciones. Durante esta etapa, nuevos datos son almacenados por medio de mediciones.
- Actualización de las medidas, donde los datos almacenados en el paso anterior son ponderados por un coeficiente llamado *Kalman Gain*, determinando la cantidad de información veraz observada.

Las ventajas del filtro de Kalman son: a) no necesita almacenar información pasada (ahorro de espacio de memoria) y b) es adaptativo, es decir, después de un ajuste inicial obtiene una salida óptima en la mayoría de las situaciones. El inconveniente que presenta es el coste computacional.

1.3.6 Red Neuronal Artificial (RNA)

El autor Fausett (1994) indica que una red neuronal es un sistema de procesamiento de información que posee ciertas características de desempeño similares a las redes neuronales biológicas. Se basan en las siguientes suposiciones:

- El procesamiento de la información sucede en múltiples elementos simples llamados neuronas.
- Las señales se transmiten entre neuronas por medio de enlaces de conexión.
- Cada enlace de conexión tiene un peso asociado, que, en una red neuronal típica, multiplica la señal transmitida.
- Cada neurona aplica una función de activación (usualmente no lineal) a su información de entrada para determinar su información de salida.

Una red neuronal está conformada por una cantidad de elementos de procesamiento simple (neuronas). Dicha neurona se encuentra unida a otra mediante un enlace de comunicación, asociado a un peso. El peso representa la información usada por la red para resolver un problema.

Tabla 3. Analogía de red neuronal biológica y red neuronal artificial.

Red neuronal biológica	Red neuronal artificial
Soma	Neurona
Dendritas	Entrada
Axón	Salidas
Sinapsis	Pesos

Fuente: Kevin (2016).

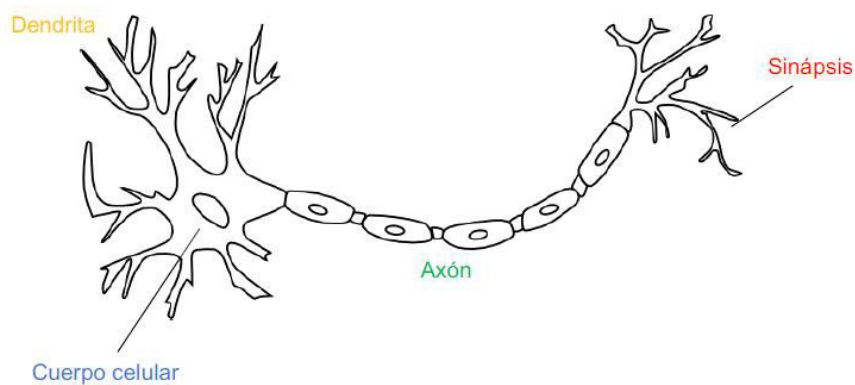


Figura 15. Neurona Biológica.

Fuente: Kevin (2016).

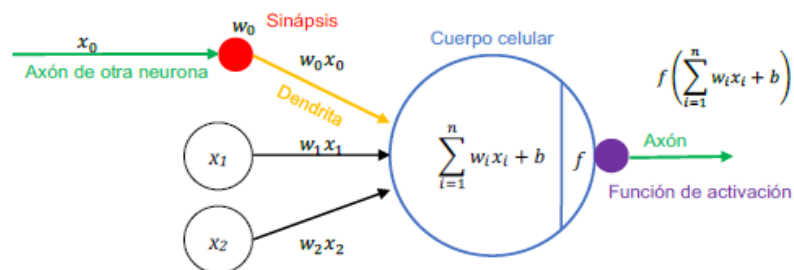


Figura 16. Arquitectura de Neurona Artificial.

Fuente (Kevin, 2016).

1.3.7 Arquitecturas típicas de la red neuronal

Red neuronal de una capa

Son redes que presentan una única capa de pesos de conexión. Las unidades pueden ser distinguidas como unidades de entrada y unidades de salida. La misma red neuronal puede ser utilizada para múltiples aplicaciones dependiendo de la interpretación de la red. Algunos ejemplos de uso son, en la clasificación de patrones y asociación de patrones.

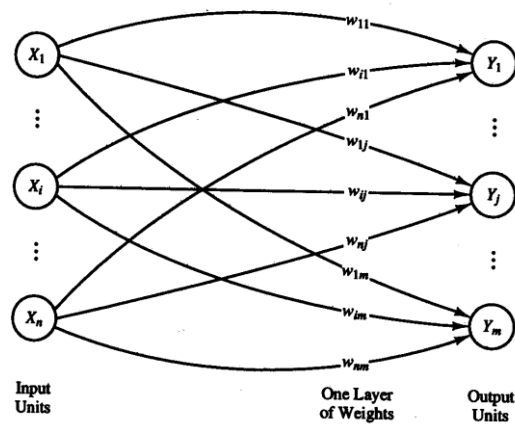


Figura 17. Red neuronal de una capa.

Fuente: Kevin (2016).

Red neuronal de múltiples capas

Es una red que tiene una o más capas (o niveles) de nodos (o capas ocultas). Las redes multicapa pueden resolver problemas más complicados, pero su entrenamiento puede suponer una mayor dificultad. En algunos casos, el entrenamiento puede ser más preciso debido a que es posible que este tipo de red neuronal solucione problemas que una red neuronal de capa simple entrenada no pueda procesar correctamente.

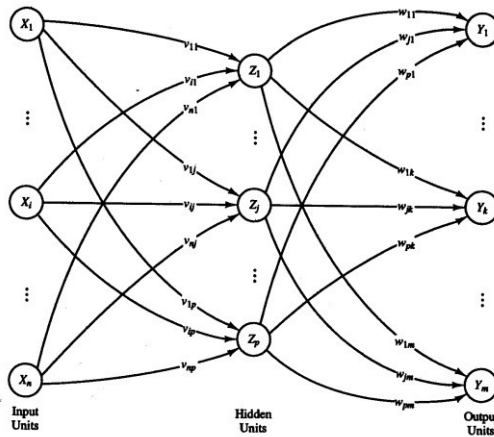


Figura 18. Red neuronal de múltiples capas.

Fuente: Kevin (2016).

Red de capa competitiva

Las neuronas que forman la red neuronal compiten entre ellas para determinar quién representara el patrón, la ganadora obtiene todo el aprendizaje del patrón. Este tipo de red presenta como objetivo formar grupo de patrones, categorías, representados por cada neurona.

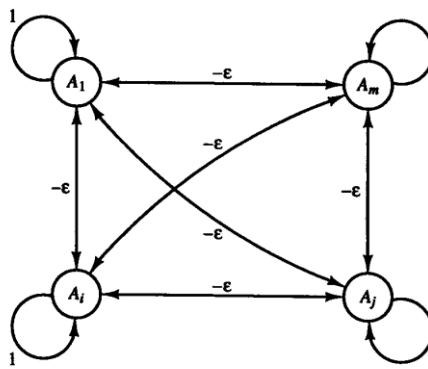


Figura 19. Red de capa competitiva.

Fuente: Kevin (2016).

1.3.8 Función de activación

La función de activación está encargada de asociar la información de entrada de la red neuronal con el siguiente estado de activación que posea la misma red neuronal; por lo tanto, puede ser considerada como una función de decisión.

Existen diferentes tipos de funciones de activaciones, entre las principales se encuentran:

Función identidad

Es una función que tiene como salida el mismo resultado que la entrada, por lo general su uso se limita a la última capa de la red neuronal.

$$f(x) = x, \quad \text{para todo } x \quad (9)$$

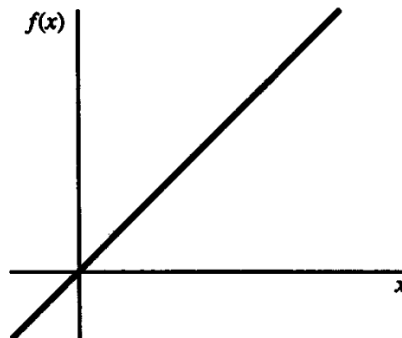


Figura 20. Función identidad.

Fuente Fausett (1994).

Función escalón unitario binario (Binary step function)

Esta función brinda un resultado binario "0" o "1" en base a un umbral o threshold " θ ". Esta función es usada en el algoritmo original de perceptrón.

$$f(x) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases} \quad (10)$$

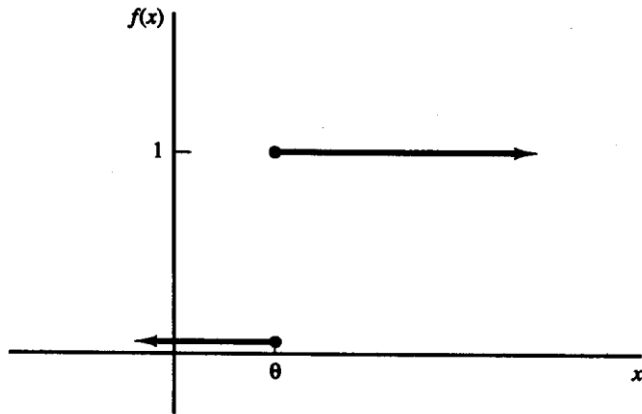


Figura 21. Función escalón unitario binario.

Fuente Fausett (1994).

Dónde: θ es el umbral de decisión (threshold).

Función escalón unitario bipolar (Bipolar step function)

Esta función brinda un resultado bipolar “1” o “-1” en base a un umbral o threshold “ θ ”.

$$f(x) = \begin{cases} 1, & x \geq \theta \\ -1, & x < \theta \end{cases} \quad (11)$$

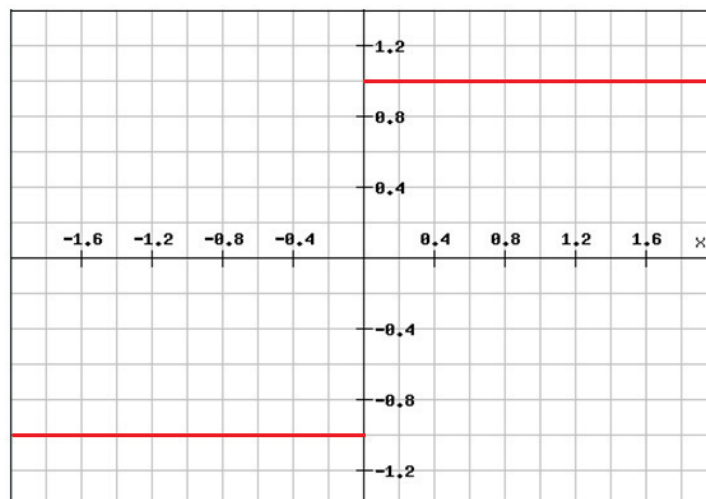


Figura 22. Función escalón unitario bipolar.

Fuente: los autores.

Función sigmoide binaria

Esta es una función suavizante que devuelve un resultado en el rango $[0,1]$, una de las ventajas de esta función es que es diferenciable; por lo tanto, se puede derivar. Esta característica es crucial para el desarrollo del algoritmo de retropropagación.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

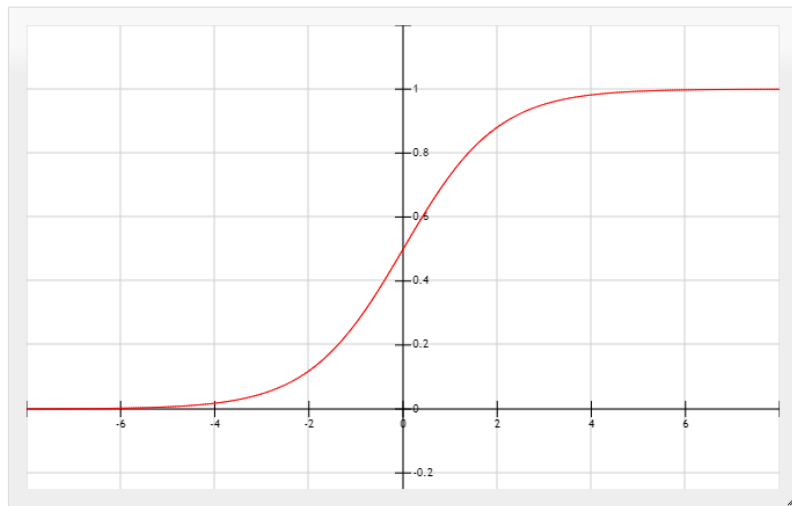


Figura 23. Función sigmoide binaria.

Fuente: los autores.

Su derivada es calculada de la siguiente manera:

$$\begin{aligned} f'(x) &= \frac{\partial}{\partial x} \left(\frac{1}{1 + e^{-x}} \right) \\ f'(x) &= \frac{e^{-x}}{1 + e^{-x}} \left(\frac{1}{1 + e^{-x}} \right) \end{aligned} \quad (13)$$

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

$$f'(x) = \frac{1 + e^{-x} - 1}{(1 + e^{-x})^2}$$

$$f'(x) = \frac{1 + e^{-x}}{(1 + e^{-x})^2} - \frac{1}{(1 + e^{-x})^2}$$

$$f'(x) = \frac{1}{1 + e^{-x}} - \left(\frac{1}{1 + e^{-x}}\right)^2$$

$$f'(x) = f(x) - f(x)^2$$

$$f'(x) = f(x)(1 - f(x))$$

Alguna vez, fue la función de activación más usada para entrenar redes neuronales en retropropagación, sin embargo, ya no se recomienda usarla por varias desventajas que presenta como el uso de mucho recurso computacional en comparación a otras funciones de activación.

Función tangente hiperbólica

También es conocido como función sigmoide bipolar dado que es una modificación de la función sigmoide binaria que devuelve un resultado en el rango $[-1,1]$. Es preferido a la función sigmoide binaria por razones de optimización. Por ejemplo, al centrar la data en 0, es más probable que la unidad de salida tenga un valor cercano a 0, lo cual significa que, al actualizar el valor del peso de conexión, el cambio es mínimo, que es preferible a que se actualice en manera de zigzag.

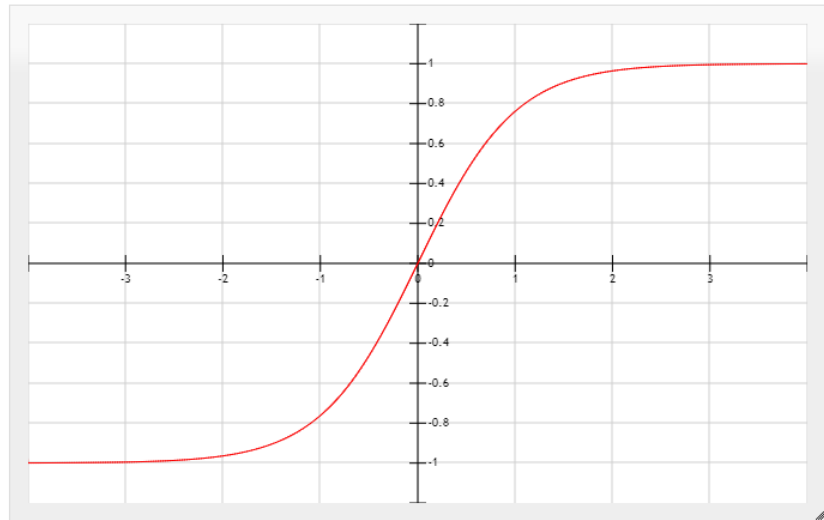


Figura 24, Función tangente hiperbólica.

Fuente: los autores.

La fórmula de la tangente es calculada de la siguiente manera:

$$f(x) = \tanh x = \frac{\sinh x}{\cosh x}$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f(x) = \frac{e^x - \frac{1}{e^x}}{e^x + \frac{1}{e^x}}$$

$$f(x) = \frac{\frac{e^{2x} - 1}{e^x}}{\frac{e^{2x} + 1}{e^x}}$$

$$f(x)f(x) = \frac{e^{-2x}}{e^{-2x}} \left(\frac{e^{2x} - 1}{e^{2x} + 1} \right)$$

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

(14)

$$f(x) = \frac{1}{1 + e^{-2x}} - \frac{e^{-2x}}{1 + e^{-2x}}$$

$$f(x) = \frac{1}{1 + e^{-2x}} - \frac{1 + e^{-2x} - 1}{1 + e^{-2x}}$$

$$f(x) = \frac{1}{1 + e^{-2x}} - \left(\frac{1 + e^{-2x}}{1 + e^{-2x}} - \frac{1}{1 + e^{-2x}} \right)$$

$$f(x) = \frac{1}{1 + e^{-2x}} - 1 + \frac{1}{1 + e^{-2x}}$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

Su derivada es calculada de la siguiente manera:

$$f'(x) = \frac{\partial}{\partial x} \left(\frac{\sinh x}{\cosh x} \right)$$

$$f'(x) = \frac{(\sinh' x)(\cosh x) - (\cosh' x)(\sinh x)}{\cosh^2 x}$$

$$f'(x) = \frac{(\cosh x)(\cosh x) - (\sinh x)(\sinh x)}{\cosh^2 x}$$

$$f'(x) = \frac{\cosh^2 x - \sinh^2 x}{\cosh^2 x}$$

$$f'(x) = 1 - \frac{\sinh^2 x}{\cosh^2 x}$$

$$f'(x) = 1 - \tanh^2 x$$

$$f'(h) = 1 - f(x)^2 \tag{15}$$

ReLU (Rectified Linear Unit)

Devuelve un resultado en el rango $[0, \infty]$. Es posible que sea el que más se asemeja al biológico.

$$f(x) = \max(0, x) \quad (16)$$

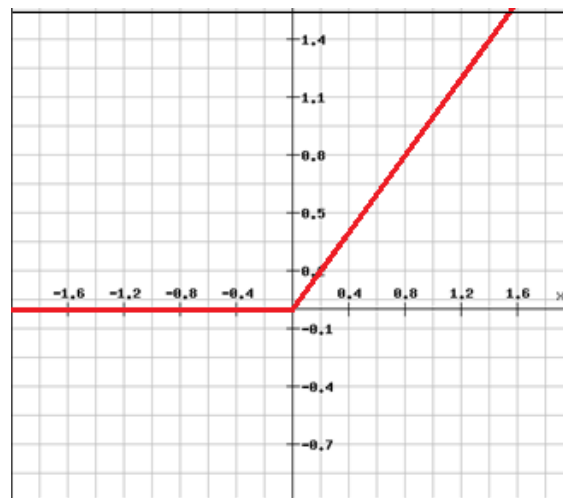


Figura 25. ReLU.

Fuente: los autores.

Para encontrar la derivada, que es la pendiente de una recta, se puede ver en el gráfico que cuando el valor es mayor a 0, la pendiente es 1, y cuando el valor es menor o igual a 0, la pendiente es 0. Por lo que la derivada puede ser definida de la siguiente manera:

$$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (17)$$

Cabe resaltar de que cuando el valor es igual a 0, la pendiente es indefinida, puesto a que puede tomar cualquier valor; sin embargo, es una decisión común darle el valor de 0 en la implementación.

1.3.9 Perceptrón multicapa con retropropagación

Como relata Kevin (2016) el perceptrón no podía reconocer patrones complicados como una compuerta lógica XOR, esto ocasionó que por muchos años se pierda interés en la investigación del perceptrón. La limitación más marcada del perceptrón original era que no contaba con una capa oculta y que la capa de salida sólo podía tener 2 valores, esto impedía implementar un método óptimo. Luego el uso de una función de activación sigmoide daría pasó a que se descubra una regla de aprendizaje conocido como la regla delta generalizada que permitía utilizar capas ocultas.

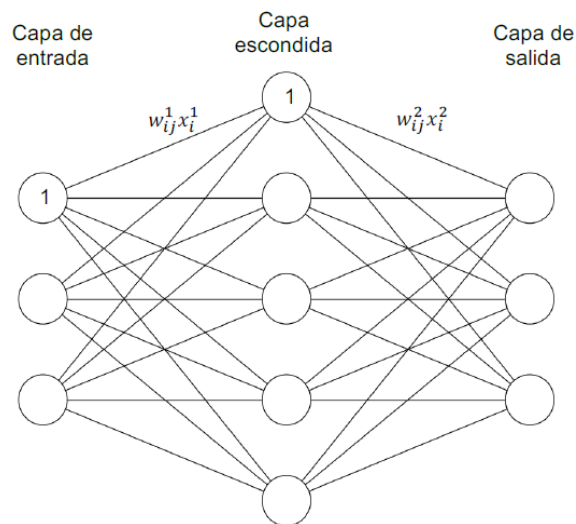


Figura 26. Arquitectura genérica de un perceptrón multicapa.

Fuente: Kevin (2016).

En la figura 26, tenemos una red neuronal con 1 capa oculta, esta será entrenada usando el método de retropropagación. En este algoritmo, los pesos se van a ajustar a medida que la red neuronal va avanzando de la capa de entrada a la capa de salida, y desde la capa de salida a la capa de entrada.

Inicialización

Por lo general, los pesos son inicializados de manera aleatoria, esto se puede lograr de varias maneras; sin embargo, sigue siendo un área activa

de investigación. Para este ejemplo se inicializará los pesos con un valor aleatorio en el rango [0,1].

Propagación

Para cada neurona de la capa escondida, se calcula el peso total neto y la unidad de salida usando una función de activación no lineal. En este ejemplo, se usará la función de sigmoide binaria. Este procedimiento debe ser repetido en la capa de salida cuando todas las neuronas de la capa escondida hayan sido procesadas. Si hay más de una capa escondida se debe esperar a que se terminen de procesar las neuronas de una capa para poder pasar a la siguiente, hasta llegar a la capa de salida.

$$\begin{aligned} net &= \sum_{i=0}^n (w_i In_i) \\ f(net) &= \frac{1}{1 + e^{-net}} \end{aligned} \quad (18)$$

Donde:

n: cantidad de neuronas de entrada

w_i: peso de la sinapsis

In_i: neurona de entrada

Luego de calcular el valor de cada neurona de salida, se calcula el costo de pérdida. Para este ejemplo, se usa el error cuadrático medio (ECM) de la capa de salida. El objetivo va a ser minimizar este error lo más posible.

$$E = \sum_{j=1}^N \frac{1}{2} (t_j - z_j)^2 \quad (19)$$

Dónde:

E: error cuadrático medio

N: cantidad de datos de salida

t: salida esperada de la red neuronal

z: salida obtenida por la red neuronal

Propagación hacia atrás (backpropagation)

Para cada peso “w” de la red, se quiere saber cómo es que un cambio en el peso afecta el error total. Este problema es resuelto usando la regla delta generalizada, para lo cual se halla la derivada parcial usando la regla de la cadena:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial z_j} \frac{\partial z_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (20)$$

Como el nombre implica, se empieza con los pesos que se conectan a la última capa w_{ij}^2 y se retrocede hasta la primera capa. De la regla de la cadena se obtiene que:

$$\begin{aligned} \frac{\partial E}{\partial y_j} &= \frac{\partial}{\partial y_j} \sum_{k=1}^n \frac{1}{2} (t_k - z_k)^2 \\ \frac{\partial E}{\partial z_j} &= \frac{\partial}{\partial z_j} \frac{1}{2} (t_j - z_j)^2 \\ \frac{\partial E}{\partial z_j} &= z_j - t_j \end{aligned} \quad (21)$$

$$\begin{aligned} \frac{\partial z_j}{\partial net_j} &= \frac{\partial}{\partial net_j} f(net_j) \\ \frac{\partial z_j}{\partial net_j} &= f'(net_j) \end{aligned} \quad (22)$$

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{k=1}^n (w_{kj} In_k)$$

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} (w_{ij} In_i)$$

$$\frac{\partial net_j}{\partial w_{ij}} = In_i \quad (23)$$

Entonces de (20) se puede resumir en:

$$\frac{\partial E}{\partial w_{ij}} = (z_j - t_j) f'(net_j) In_i \quad (24)$$

En esta ecuación, se puede apreciar por qué la función de activación debe ser diferenciable en el método de retropropagación. En este caso, estamos usando la función sigmoide binaria, por lo que:

$$f'(net_j) = f(net_j) (1 - f(net_j))$$

$$f'(net_j) = y_j (1 - z_j) \quad (25)$$

Para este caso:

$$\frac{\partial E}{\partial w_{ij}} = (z_j - t_j) z_j (1 - z_j) In_i \quad (26)$$

El nuevo peso de conexión puede ser calculado usando descenso de la gradiente estocástica de la siguiente manera:

$$w_{ij}(new) = w_{ij}(old) - \eta \frac{\partial E}{\partial w_{ij}} \quad (27)$$

Este cambio debe ser aplicado cuando se haya calculado el nuevo peso de todas las conexiones de la red neuronal. Para hallar el nuevo peso de las conexiones de la capa escondida, se requiere el valor original de las conexiones de la capa de salida.

En el caso de los pesos que se conectan a la capa escondida w_{ij}^1 , cada neurona contribuye al error de varias neuronas de la capa de salida, por la que se debe modificar la ecuación:

$$\begin{aligned}\frac{\partial E}{\partial z_j^l} &= \sum_{k=1}^n \frac{\partial E}{\partial z_k^{l+1}} \\ \frac{\partial E}{\partial z_j^l} &= \sum_{k=1}^n \frac{\partial E}{\partial net_k^{l+1}} \frac{\partial net_k^{l+1}}{\partial z_k^{l+1}} \\ \frac{\partial E}{\partial z_j^l} &= \sum_{k=1}^n \frac{\partial E}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial net_k^{l+1}} \frac{\partial net_k^{l+1}}{\partial z_k^{l+1}} \\ \frac{\partial E}{\partial z_j^l} &= \sum_{k=1}^n \frac{\partial E}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial net_k^{l+1}} w_{jk}^{l+1}\end{aligned}\tag{28}$$

Donde:

l : es el número de la capa

n : es la cantidad de unidades en la capa $l + 1$

En resumen, la ecuación para encontrar la derivada parcial se puede resumir de la siguiente manera:

$$\frac{\partial E}{\partial w_{ij}^l} = \delta_j^l I n_i^l\tag{29}$$

En el que:

$$\delta_j^l = \frac{\partial E}{\partial z_k^l} \frac{\partial z_k^l}{\partial net_k^l}$$

$$= \begin{cases} (t_j^l - z_j^l) f'(net_j^l), & \text{Para la capa de salida} \\ \left(\sum_{k=1}^n \delta_j^{l+1} w_{jk}^{l+1} \right) f'(net_j^l), & \text{Para las capas escondidas} \end{cases} \quad (30)$$

Con los valores calculados se puede calcular el nuevo peso de la conexión. En la ecuación se puede apreciar por qué los pesos de las neuronas se deben actualizar cuando se haya terminado el proceso de retropropagación. Este proceso de propagación y propagación hacia atrás puede ser repetido una cantidad de épocas establecidas o hasta que se llegue a un error deseable.

1.3.10 Banda 2.4 GHz

En el Perú la banda de frecuencia 2.4 GHz se define como bandas de uso libre o no licenciadas y está estipulado en el TUO (Texto único ordenado) del Reglamento General de la Ley de Telecomunicaciones DECRETO SUPREMO N° 027-2004-MTC.

A continuación, se detalla el artículo 28:

“Artículo 28.- Están exceptuados de la clasificación de servicios de la Ley, del Reglamento y de los Reglamentos Específicos que se dicten, las telecomunicaciones instaladas dentro de un mismo inmueble que no utilizan el espectro radioeléctrico y no tienen conexión con redes exteriores. También están exceptuados de contar con concesión, salvo el caso del numeral 4, de la asignación del espectro radioeléctrico, autorización, permiso o licencia, para la prestación de servicios de telecomunicaciones, de la clasificación de servicios de la Ley, del Reglamento y de los Reglamentos específicos que se dicten:

1. Aquellos servicios cuyos equipos, utilizando el espectro radioeléctrico transmiten con una potencia no superior a diez mili vatios (10mW) en antena (potencia efectiva irradiada). Dichos servicios no podrán operar en las bandas de frecuencias atribuidas a los servicios públicos de telecomunicaciones; salvo en las bandas de frecuencias 2400-2483,5 MHz y 5725-5850 MHz.

2. Aquellos servicios cuyos equipos, utilizando una canalización establecida en la banda 462, 550-462, 725 MHz y 467, 550-467, 725 MHz, transmiten con una potencia no superior a quinientos milivatios (500mW) en antena (potencia efectiva irradiada). Dichos equipos no podrán ser empleados para la prestación de servicios públicos de telecomunicaciones.

3. Aquellos servicios cuyos equipos, utilizando las bandas de 902-928 MHz, 2400 - 2483,5 MHz y 5725-5850 MHz transmiten con una potencia no superior a cien milivatios (100mW) en antena (potencia efectiva irradiada), y no sean empleados para efectuar comunicaciones en espacios abiertos. Dichos servicios no deberán causar interferencias a concesionarios de servicios públicos de telecomunicaciones.

4. Aquellos servicios cuyos equipos, utilizando las bandas de 902-928 MHz, 2400 - 2483,5 MHz y 5725-5850 MHz transmiten con una potencia no superior a cuatro vatios (4W) o 36 dBm en antena (potencia efectiva irradiada), en espacio abierto. En el caso de utilizar equipos bajo las condiciones señaladas en el numeral 4, para la prestación de servicios públicos de telecomunicaciones, se debe contar previamente con la concesión respectiva. En este caso, los concesionarios de servicios públicos de telecomunicaciones que empleen dichos equipos no requerirán del permiso para su instalación y operación, ni de la asignación de espectro radioeléctrico para su uso.”

PNAF

“El Plan Nacional de Atribución de Frecuencias contiene los cuadros de atribución de frecuencias de los diferentes servicios de telecomunicaciones en la República del Perú”, fuente (MTC, 2017).

Según documento del PNAF en la Sección IV. Cuadro de atribución de bandas de frecuencias, las frecuencias 2400-2483.50 MHz presenta las notas y observaciones P23 y P47 que se detallan a continuación:

“P23 Las siguientes bandas están destinadas para aplicaciones industriales, científicas y médicas (ICM):

13 553 – 13 567 kHz (frecuencia central 13 560 kHz),

26 957 – 27 283 kHz (frecuencia central 27 120 kHz),

40,66 – 40,70 MHz (frecuencia central 40,68 MHz),

902 – 928 MHz (frecuencia central 915 MHz),

2 400 – 2 500 MHz (frecuencia central 2 450 MHz),

5 725 – 5 875 MHz (frecuencia central 5 800 MHz), y

24 - 24,25 GHz (frecuencia central 24,125 GHz).

Los servicios de radiocomunicaciones que funcionan en estas bandas deben aceptar la interferencia perjudicial resultante de estas aplicaciones y en ningún caso podrán causar interferencias a aplicaciones ICM. Las bandas 902-928 MHz, 2400-2483,5 MHz y 5725-5850 MHz, están atribuidas a título secundario para los servicios fijo y/o móvil, público y/o privado. Aquellos que hagan uso de las frecuencias antes indicadas deberán sujetarse a la normativa establecida o que establezca el Ministerio.”

“P47 El servicio de radioaficionados por satélite podrá explotarse en las bandas 435-438 MHz, 1260-1270 MHz, 2400-2450 MHz, 3400-3410 MHz, y 5650-5670 MHz, siempre que no cause interferencia perjudicial a otros servicios explotados de conformidad con el Cuadro. Toda interferencia perjudicial causada por emisiones de una estación del servicio de radioaficionados por satélite será inmediatamente eliminada. La utilización de las bandas 1260-1270 MHz y 5650-5670 MHz por el servicio de radioaficionados por satélite se limitará al sentido Tierra-espacio.”

1.3.11 Modelo de referencia de posicionamiento de beacons

Para la correcta ubicación de los beacon se debe tener en cuenta los siguientes factores Hewlett Packard Enterprise Development LP (2015):

- La distribución de la planta a ubicar los beacon.
- El tipo de localización que se desee realizar al móvil.

No existe un despliegue de beacons único que sirva para todas las aplicaciones, pero los autores indican que se pueden generalizar en dos (2) grupos:

- Modelo de rutas: Se debe aplicar en configuraciones en donde se desea obtener una precisión de ubicación en línea recta. Es óptimo para áreas reducidas como pasillos o corredores.

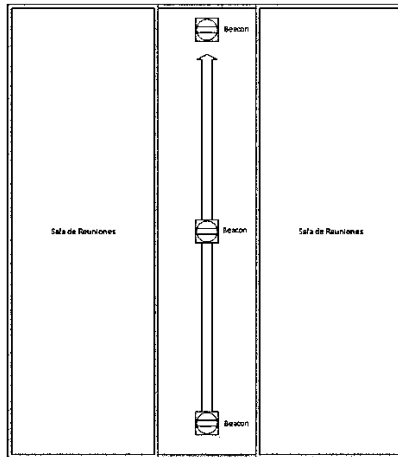


Figura 27. Esquema del modelo de rutas de Aruba Networks.

Fuente: Hewlett Packard Enterprise Development LP (2015).

El plano de la figura 27 es referencial, por lo tanto, no presenta medidas.

- Modelo de áreas: Su uso es óptimo en áreas de gran tamaño. Este modelo es necesario de implementar cuando se desea obtener una localización con precisión de dos dimensiones.

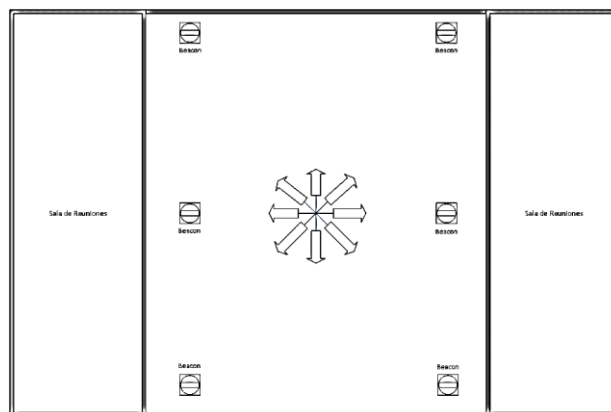


Figura 28. Esquema del modelo de áreas de Aruba Networks.

Fuente: Hewlett Packard Enterprise Development LP. (2015).

El plano de la figura 28 es referencial, por lo tanto, no presenta medidas.

Consideraciones de despliegue:

Espaciamiento: Para lograr una mejor ubicación, el autor recomienda desplegar los beacon con una separación de aproximadamente 12m. Desplegar los beacon a una menor distancia podría dar como resultado el obtener una mayor precisión en la ubicación; sin embargo, la distancia óptima entre beacons depende de factores como el modelo de despliegue, variación en la altura del techo y reflectancias en el ambiente.

Montaje: Puede realizarse en paredes o techo:

- En pared: Se utiliza en ambientes con techo abierto o en donde la altura del techo sea demasiado alta y dificulte su instalación.
- En techo: Es ideal para espacios en donde existe una preocupación por la estética, debido a que los beacon son instalados discretamente y fuera del campo visual. Adicionalmente, se mantiene a los beacon lejos del alcance de las personas y al estar ubicados por encima de los dispositivos a localizar, previene las atenuaciones causadas por una alta concentración de personas. Se debe tener en cuenta lo siguiente:
 - Los beacon deben ser instalados lejos de fuentes de calor como bombillas eléctricas o equipos de aire acondicionado.
 - Los beacon deben ser instalados por debajo de las baldosas de techo ya que estas causan grandes atenuaciones de la señal.
 - Se debe asegurar que la potencia de los beacon sea lo suficientemente fuerte para cubrir el área deseada.

- Si existen variaciones en la altura del techo, los beacons deberán de instalarse al mismo nivel que la menor altura del techo.

1.3.12 Diagrama pictórico del proyecto - Jerarquía H0



Figura 29. Diagrama pictórico del proyecto.

Fuente: los autores

1.3.13 Diagrama de bloques de la detección de la ubicación - Jerarquía H1

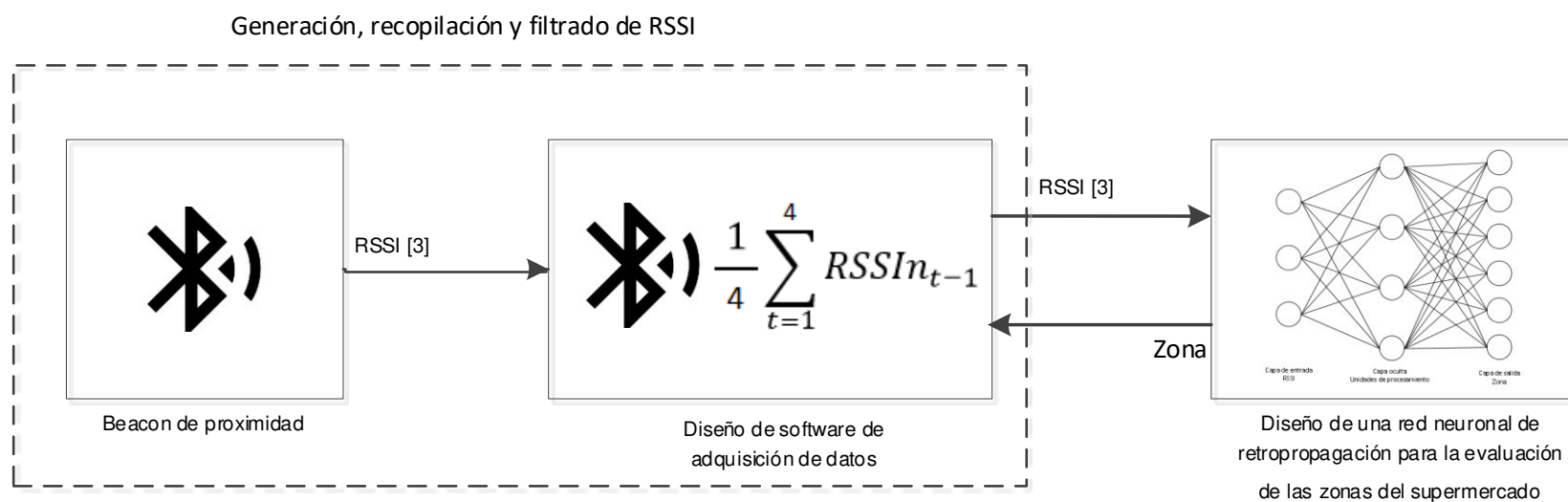


Figura 30. Diagrama de bloques del proyecto.

Fuente: los autores.

Tabla 4. *Tabla de parámetros de la tesis.*

Variable	Unidad	Tipo de dato	Descripción
RSSIn	dBm	Z={-100; ... ; -10}	Potencia de señal recibida, dispositivo móvil interpreta el dato como entero negativo.
n	-	Z=[1;20>	Orden del filtro.
σ	-	R=[0;20>	Desviación estándar del RSSIn.
RSSIn _f	dBm	R={-100; ... ; -10}	RSSIn procesado por filtro.
η	-	R={0 ; ... ; 1}	Tasa de aprendizaje de la red neuronal
I _n	-	R={0 ; ... ; 1}	Entrada de red neuronal, RSSIn filtrado y normalizado.
w _{ij}	-	R={0 ; ... ; 1}	Pesos de las sinapsis de la red neuronal.
Z _n	-	Z={0;1}	Salida de red neuronal.
e	-	R={0 ; ... ; 1}	Error cuadrático medio de la red neuronal.

Fuente: los autores.

CAPÍTULO II

METODOLOGÍA

2.1 Materiales

Para este proyecto de tesis, se emplearon los siguientes materiales.

- Beacon de proximidad
- Teléfono inteligente: Marca ONE Plus Modelo 3T, Marca Huawei Modelo P9 Lite
- PC: Marca Lenovo Modelo Z480
- Visual Studio Community 2017
- Matlab R2014a
- MIT App Inventor 2

A continuación, se detallan los materiales.

2.1.1 Beacon de proximidad

A continuación, se indican las especificaciones técnicas de los beacons de proximidad Estimote. Fuente (Estimote, 2017c).

- MCU: Bluetooth® SoC, procesador ARM® Cortex®-M4 de 32 bits con FPU, velocidad del núcleo de 64 MHz, Memoria Flash de 512 Kb, 64 kB de memoria RAM.
- Radio: Bluetooth® 4.2 LE estándar, alcance de hasta 70 metros, potencia de salida de -20 a +4 dBm en pasos de 4 dB, "modo susurro" -40 dBm, sensibilidad de -96 dBm, rango de frecuencia de 2400 MHz a 2483.5 MHz, 40 canales, separación de canales adyacentes de 2 MHz, modulación GFSK (FHSS), antena PCB meander, monopole, ganancia de antena de 0 dBi, tasa de datos inalámbrica de 1 Mbps (admite 2 Mbps).
- Fuente de alimentación: 1 x CR2477 - batería de celda primaria de litio 3.0V (reemplazable).
- Especificación ambiental: Temperatura de funcionamiento de 0 ° C a 60 ° C (32 ° F a 140 ° F), temperatura de almacenamiento (recomendada) de 15 a 30 °C (59 a 86 °F), humedad relativa (funcionamiento) de 20% a 80% de humedad relativa, humedad relativa (almacenamiento) de 10% a 90% de humedad relativa, sin condensación, a prueba de salpicaduras.
- Tamaño y peso: longitud 55 mm (2.16 pulgadas), ancho 38 mm (1.49 pulgadas), altura 18 mm (0.71 pulgadas), peso 30 g (1.06 onzas).

2.1.2 Teléfono inteligente OnePlus 3T

A continuación, se indican las especificaciones técnicas del teléfono inteligente marca OnePlus modelo 3T. Fuente: (OnePlus, 2017).

- CPU: Qualcomm® Snapdragon™ 821, quad core, Kryo™ 2x2.35 GHz, 2x1.6 GHz, GPU Adreno™ 530, RAM 6GB LPDDR4, almacenamiento 64GB / 128GB UFS 2.0.
- Conectividad: red GSM en las bandas de 850, 900, 1800, 1900 MHz, 4G LTE (Cat. 6).
- Wi – Fi: Wi-Fi 802.11 a/b/g/n/ac, bluetooth 4.2, NFC habilitado, GPS, GLONASS, BeiDou.

2.1.3 Teléfono inteligente Huawei P9 Lite

A continuación, se indican las especificaciones técnicas del teléfono inteligente marca Huawei modelo P9 Lite. Fuente (Huawei Technologies Co, 2017)

- Banda: red VNS-L31&L21 4G FDD en las bandas B1/B3/B7/B8/B20, red 3G UMTS en las bandas B1/B2/B8, red 2G GSM en las bandas 850/900/1800/1900MHz, red VNS-L22 4G FDD en las bandas B1/B3/B5/B7/B8/B19/B28, red DD B40 3G UMTS en las bandas B1/B5/B6/B8/B19, red 2G GSM en las bandas 850/900/1800/1900MHz, red VNS-L23 4G FDD en las bandas B2/B4/B5/B7/B28, red 3G UMTS en las bandas B1/2/4/5/8, red 2G GSM en las bandas 850/900/1800/1900MHz, red VNS-L53 4G FDD en las bandas B2/B4/B5/B7/B12/B17, red 3G UMTS en las bandas B1/2/4/5, red 2G GSM en las bandas 850/900/1800/1900MHz.
- Batería: 3000 mAh (típicamente).
- Memoria: 3GB RAM, 16GB Memoria Interna.
- Conectividad: Wi-Fi 2.4GHz 802.11b/g/n, Wi-Fi direct Wi-Fi hotspot, Micro-USB NFC(solo para VNS-L31&L21).
- Procesador: Kirin650 Quad-Core 2.0GHz + Quad-Core 1.7GHZ , 64-bit.

2.1.4 Laptop Lenovo IdeaPad Z480

A continuación, se indican las especificaciones técnicas de la laptop marca Lenovo modelo IdeaPad Z480. Fuente (Magitech, 2017).

- Procesador: Intel Core i5 3210M 2.5GHz (3.1GHz c/TB).
- Memoria RAM: 6GB DDR3.
- Conectividad Ethernet (RJ-45), Wi-Fi 802.11b/g/n, Bluetooth.

2.1.5 Software Visual Studio Community 2017

A continuación, se indican los requisitos del sistema para el uso del software Visual Studio Community 2017. Fuente (Microsoft, 2017).

- Sistemas operativos compatibles: Windows 10 versión 1507 o superior, Windows Server 2016, Windows 8.1 Core, Professional y Enterprise, Windows Server 2012 R2 Essentials, Standard, Datacenter, Windows 7 SP1 Home Premium, Professional, Enterprise, Ultimate.
- Hardware: Procesador de 1.8 GHz o más rápido. Dual-core o mejor recomendado.
- Memoria RAM: 2 GB de RAM, se recomiendan 4 GB de RAM (mínimo de 2,5 GB si se ejecuta en una máquina virtual).

2.1.6 Software Matlab R2014a

A continuación, se indican los requisitos del sistema para el uso del software Matlab R2014a. Fuente (The MathWorks, 2017).

- Sistema operativos: Windows 8.1, Windows 8, Windows 7 Service Pack 1, Windows Vista Service Pack 2, Windows XP Service Pack 3, Windows XP x64 Service Pack 2, Windows Server 2012, Windows Server 2008 R2 Service

Pack 1, Windows Server 2008 Service Pack 2, Windows Server 2003 R2 Service Pack 2.

- Procesador: Cualquiera Intel o AMD x86.
- Espacio en disco: 1 GB para el MATLAB, 3.4 GB para la instalación típica.
- RAM: 2048 MB.

2.1.7 Software MIT App Inventor 2

A continuación, se indican los requisitos del sistema para el uso del software MIT App Inventor 2. Fuente (Massachusetts Institute of Technology, 2017).

- Computadora y sistema operativo: Macintosh (con procesador Intel), Mac OS X 10.5, 10.6+, Windows XP, Windows Vista, Windows 7+, GNU / Linux: Ubuntu 8+, Debian 5+ (Nota: el desarrollo en vivo de GNU / Linux solo es compatible con conexiones WiFi entre la computadora y el dispositivo Android).
- Navegador: Mozilla Firefox 3.6 o superior, Apple Safari 5.0 o superior, Google Chrome 4.0 o superior.

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1 Análisis

3.1.1 Preprocesamiento de datos aplicando filtros

Se procedió a analizar diferentes tipos de filtros variando el orden de cada uno, de esta forma poder determinar qué filtro es el más adecuado para aplicarla en el proyecto de tesis, a continuación, se representarán los valores RSSI generados utilizando una función aleatoria entre el rango de -10 a -100, tal y como se presentarían en una aplicación real.

Todas las gráficas presentadas a continuación, fueron generadas usando el software matlab versión 2014a.

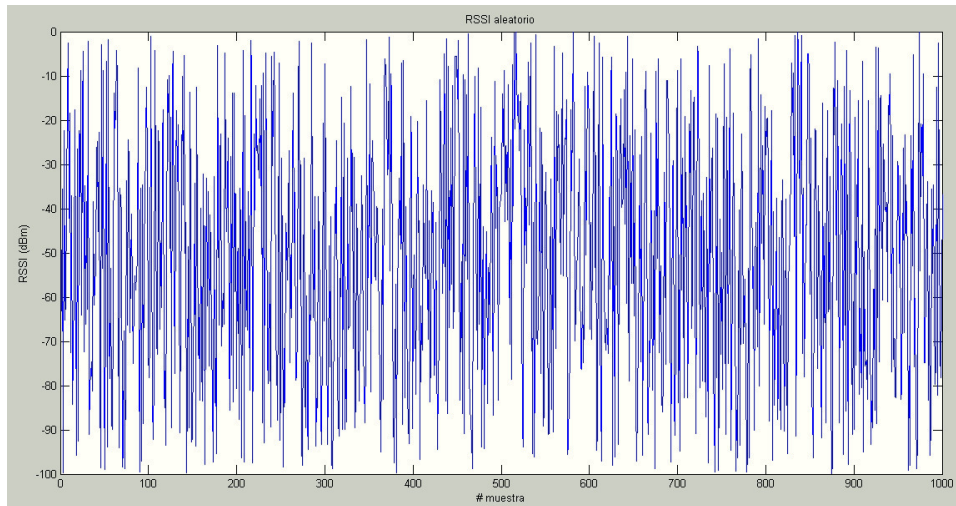


Figura 31. RSSI aleatorio.

Fuente: los autores.

La desviación estándar para esta muestra es de 28.5282.

Filtro de media aritmética

El primer filtro en ser analizado será la media aritmética, dicho filtro es el más simple en ser implementado, se realizaron pruebas aumentando el orden del filtro, desde orden 2 hasta orden 15.

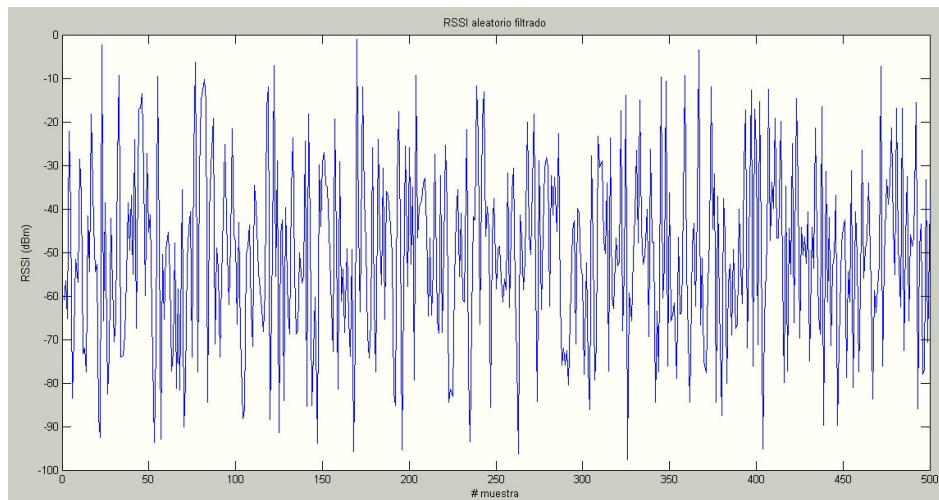


Figura 32. RSSI con filtro de media aritmética de orden 2.

Fuente: los autores.

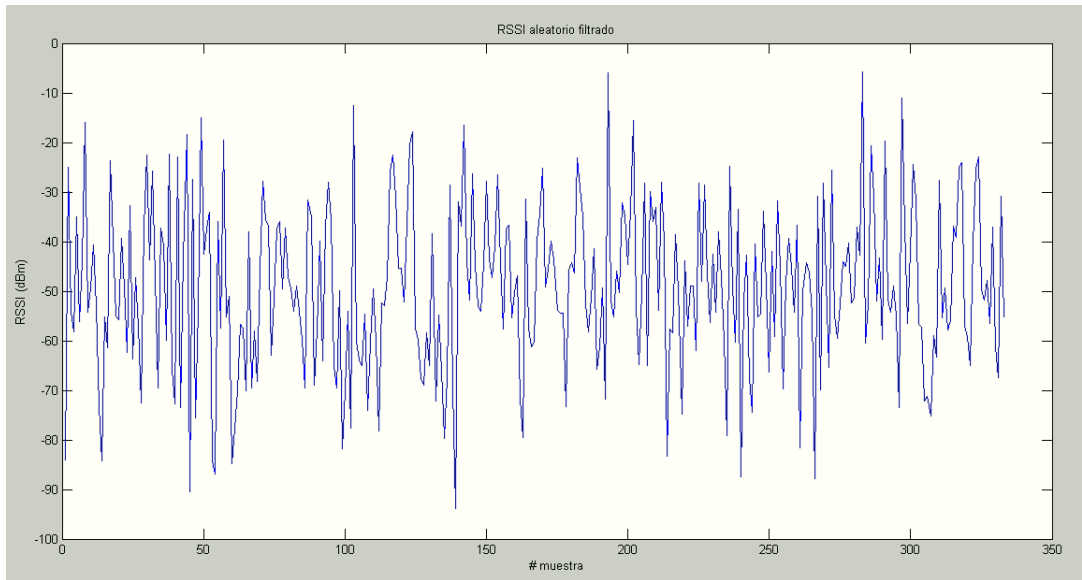


Figura 33. RSSI con filtro de media aritmética de orden 3.

Fuente: los autores.

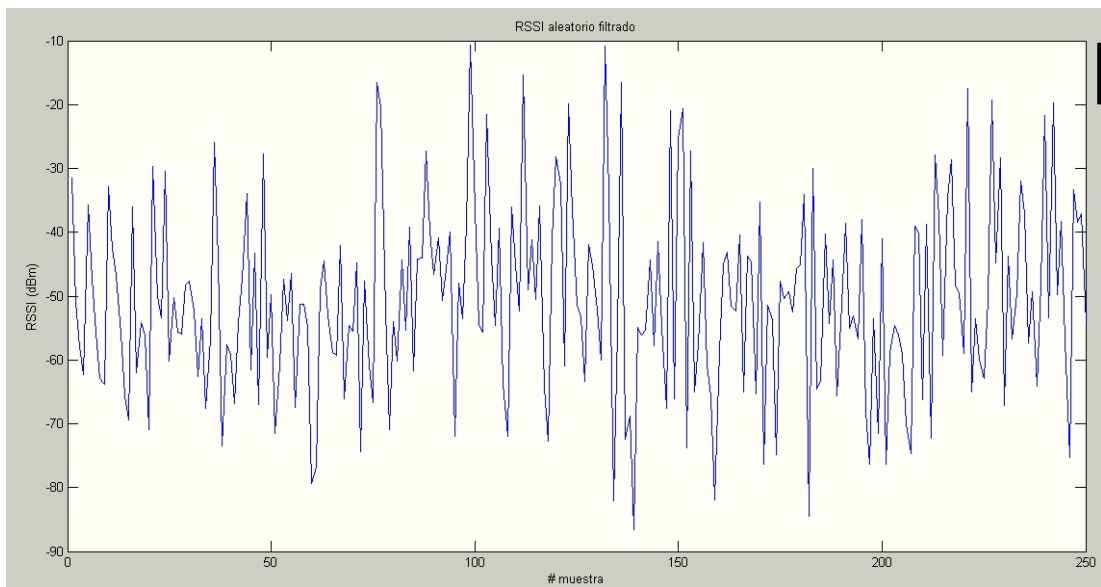


Figura 34. RSSI con filtro de media aritmética de orden 4.

Fuente: los autores.

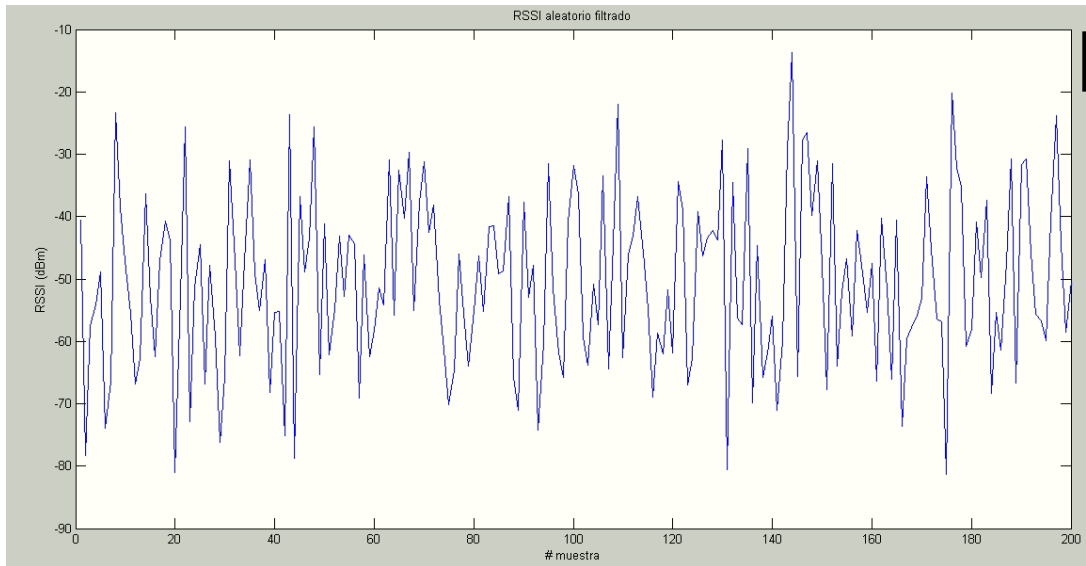


Figura 35. RSSI con filtro de media aritmética de orden 5.

Fuente: los autores.

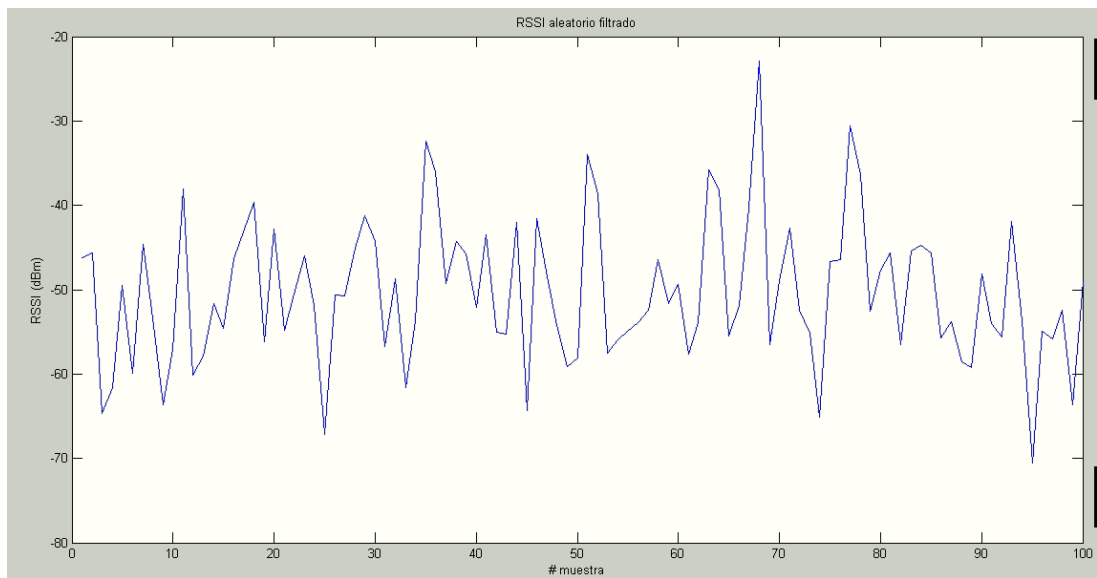


Figura 36. RSSI con filtro de media aritmética de orden 10.

Fuente: los autores.

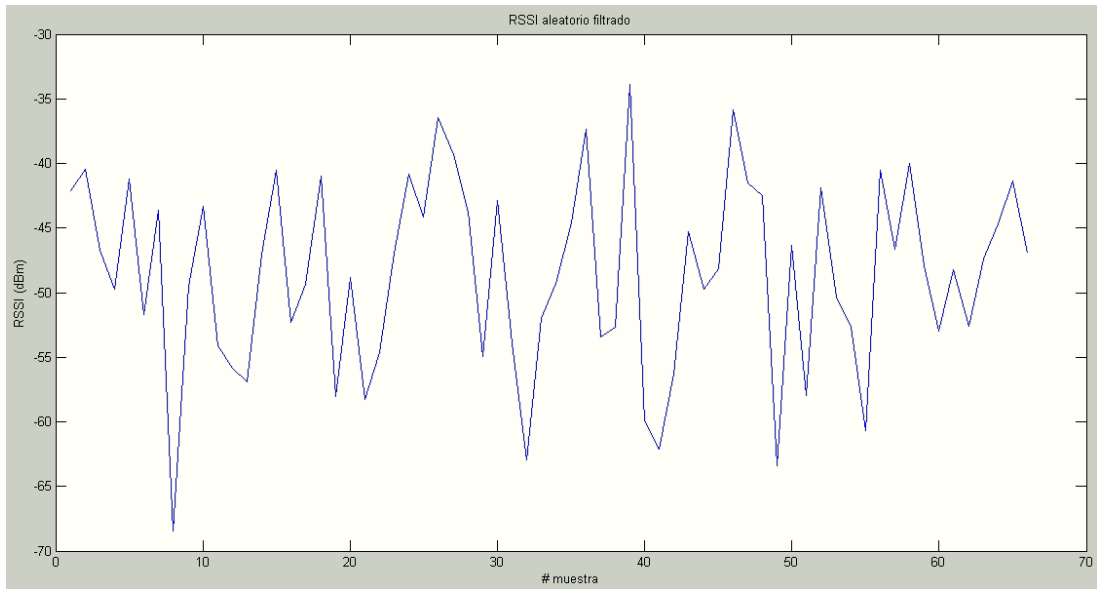


Figura 37. RSSI con filtro de media aritmética de orden 15.

Fuente: los autores.

A continuación, se muestra la disminución de la desviación estándar al variar el orden del filtro. Esta medida nos ayuda a observar la dispersión de los datos conforme se aumenta el orden del filtro:

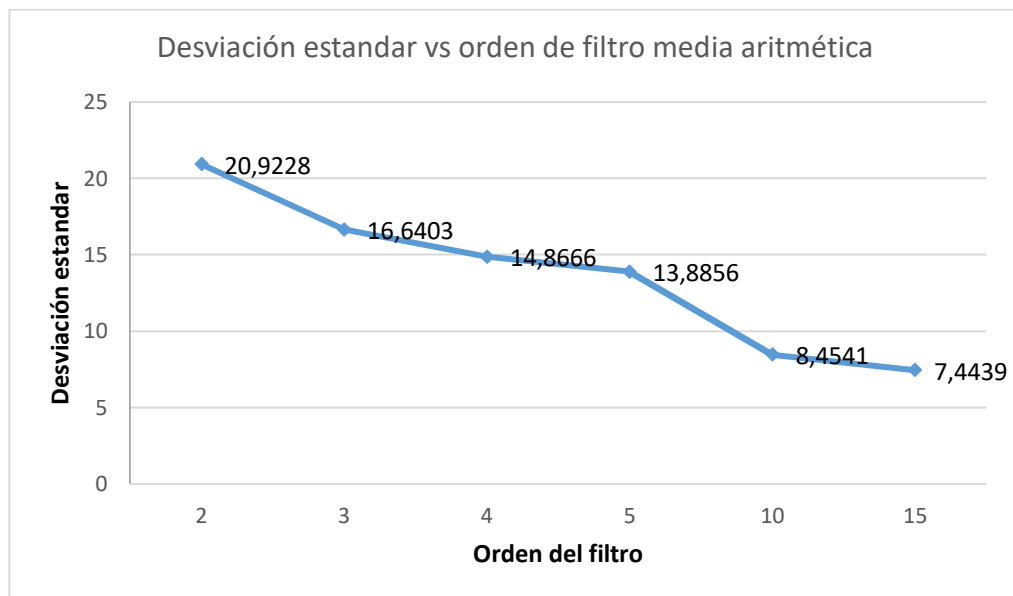


Figura 38. Desviación estándar vs. Orden de filtro media aritmética.

Fuente: los autores.

Como se puede observar, si bien la dispersión de los datos disminuye al aumentar el orden del filtro, esto trae como consecuencia la disminución de la cantidad de datos útiles, de acuerdo con los datos tomados, de 1000 datos mientras la data no había sido filtrada, hasta tan solo 66 muestras luego de que los datos fueron filtrados con el filtro de promedio aritmético de orden 15.

Filtro de mediana

De igual manera, se procedió al análisis del filtro de mediana, comenzando en el orden 2 hasta el orden 15, se trabajó con los mismos datos de entrada que el filtro anterior:

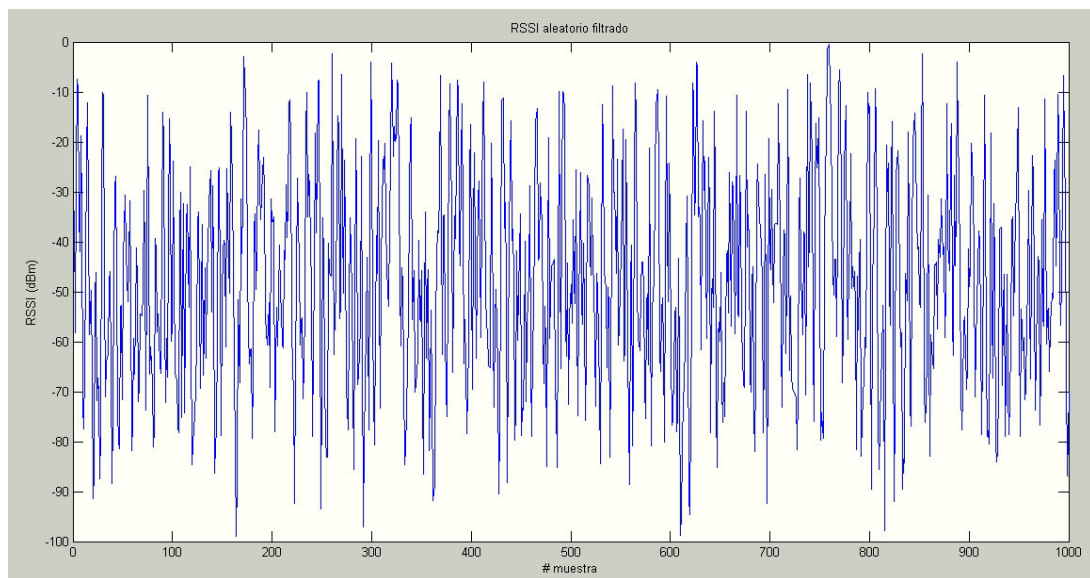


Figura 39. RSSI con filtro de mediana de orden 2.

Fuente: los autores.

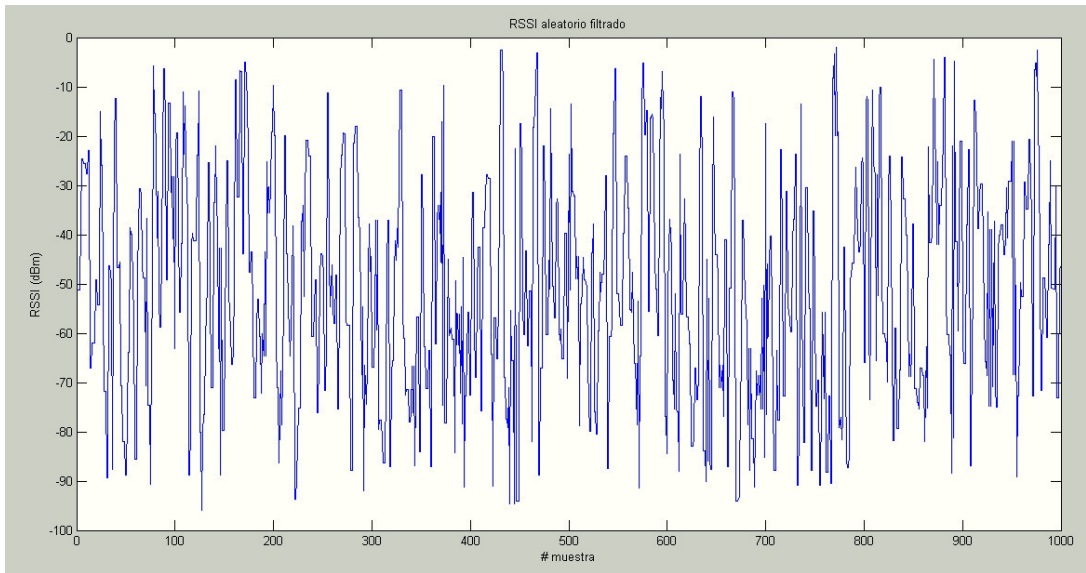


Figura 40. RSSI con filtro de mediana de orden 3.

Fuente: los autores.

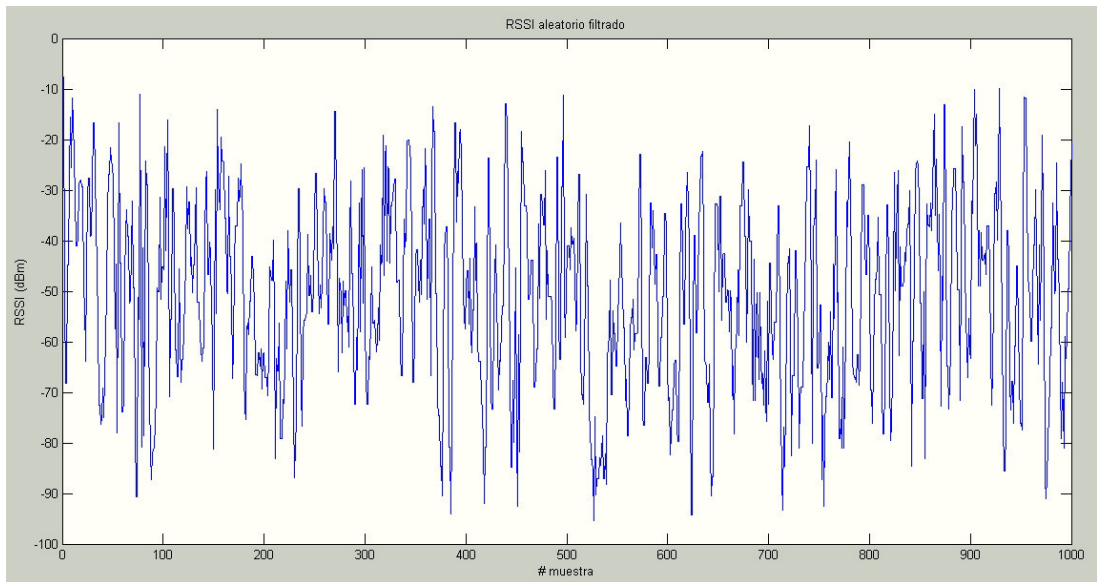


Figura 41. RSSI con filtro de mediana de orden 4.

Fuente: los autores.

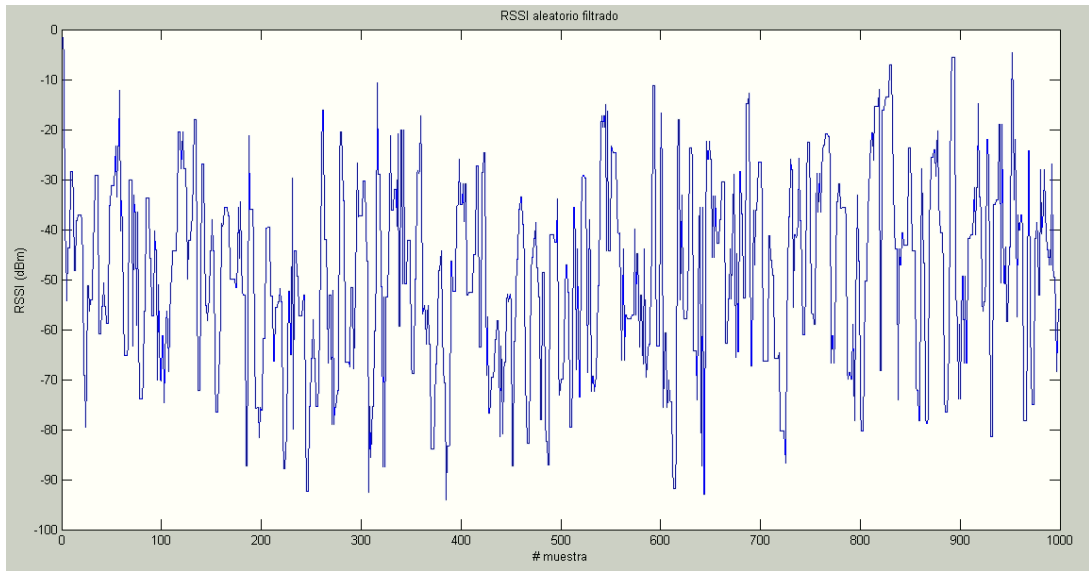


Figura 42. RSSI con filtro de mediana de orden 5.

Fuente: los autores.

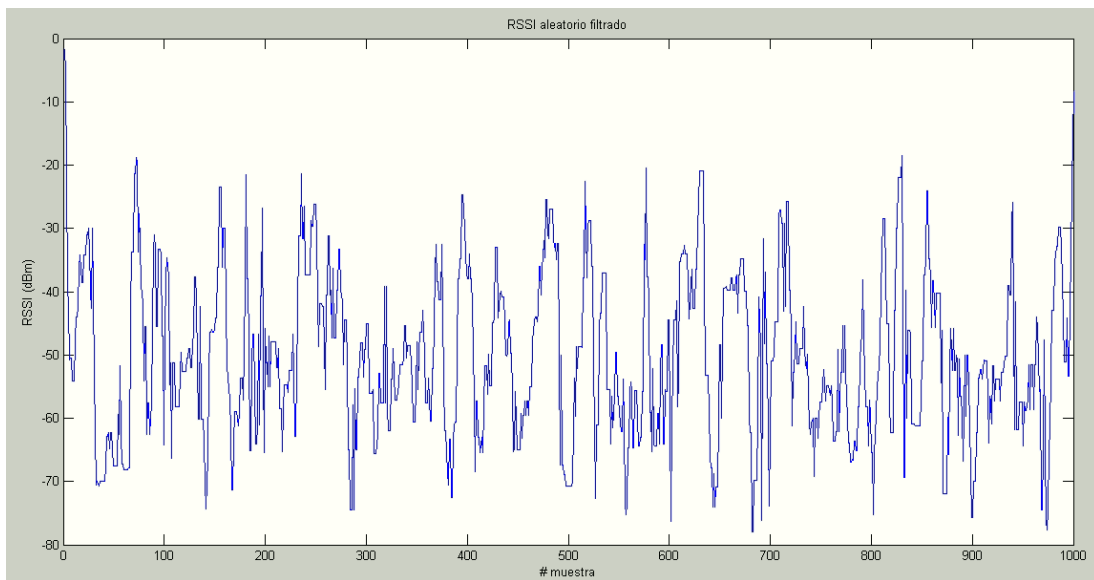


Figura 43. RSSI con filtro de mediana de orden 10.

Fuente: los autores.

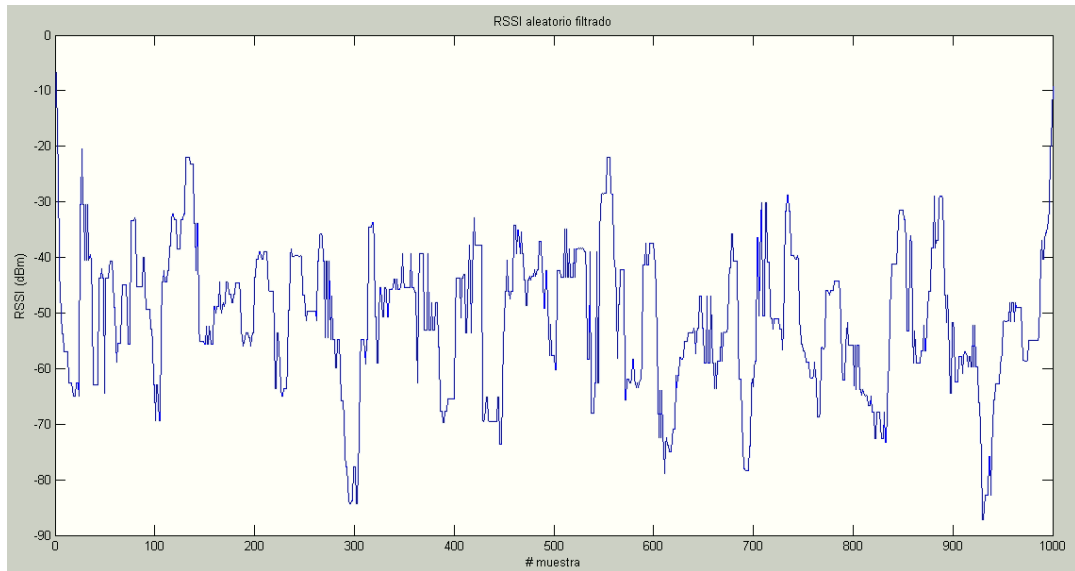


Figura 44. RSSI con filtro de mediana de orden 15.

Fuente: los autores.

A continuación, se muestra la gráfica de variación de gradiente respecto al orden del filtro antes mencionado:

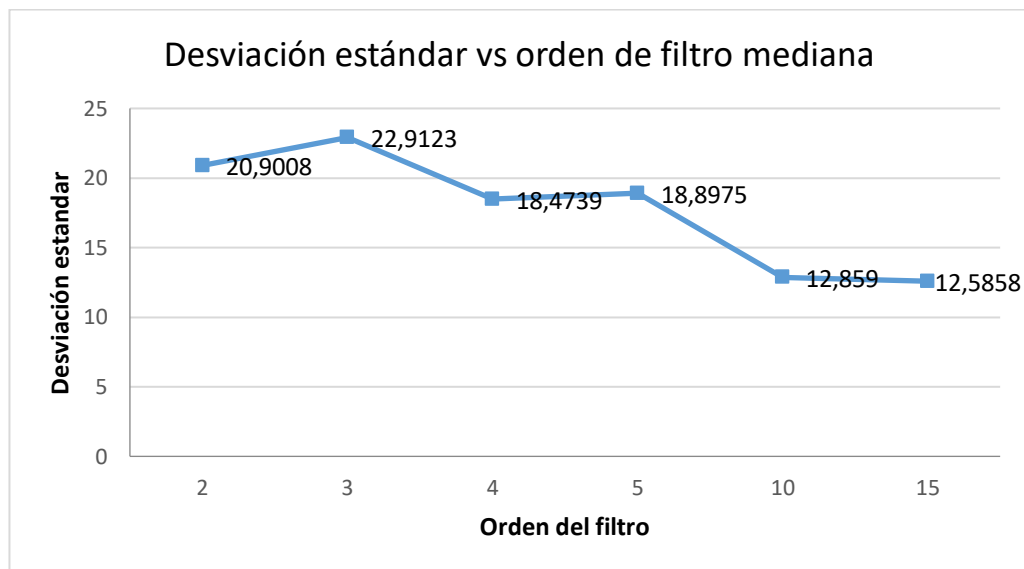


Figura 45. Desviación estándar vs. Orden de filtro de mediana.

Fuente: los autores.

Como se puede observar, este filtro mantiene el número de muestras original, sin embargo, presenta un desempeño inferior al filtro de promedio aritmético por lo que su uso fue descartado para la solución propuesta.

Filtro de media móvil

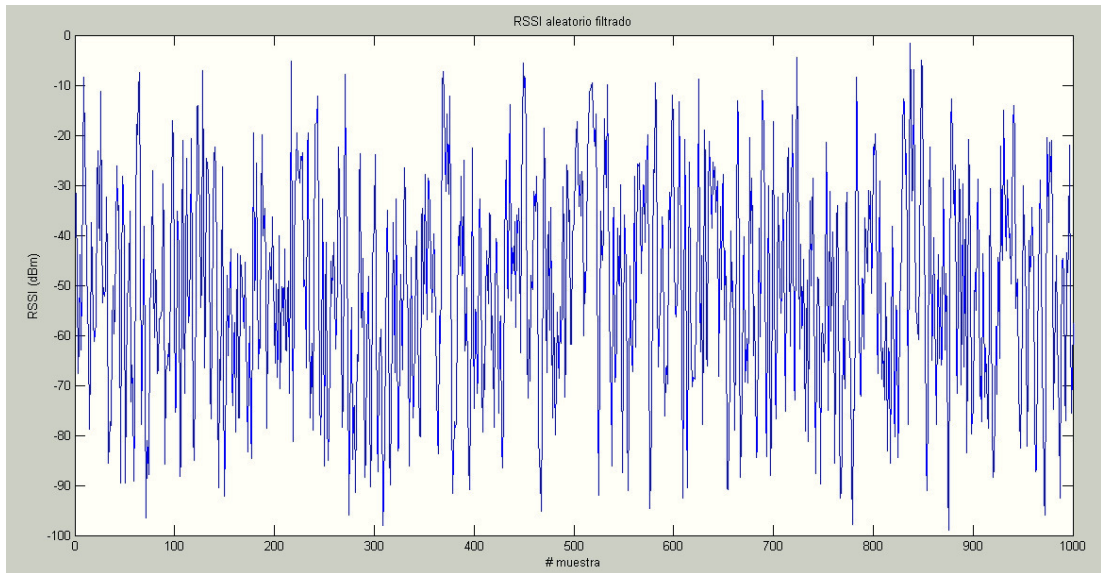


Figura 46. RSSI con filtro de media móvil de orden 2.

Fuente: los autores.

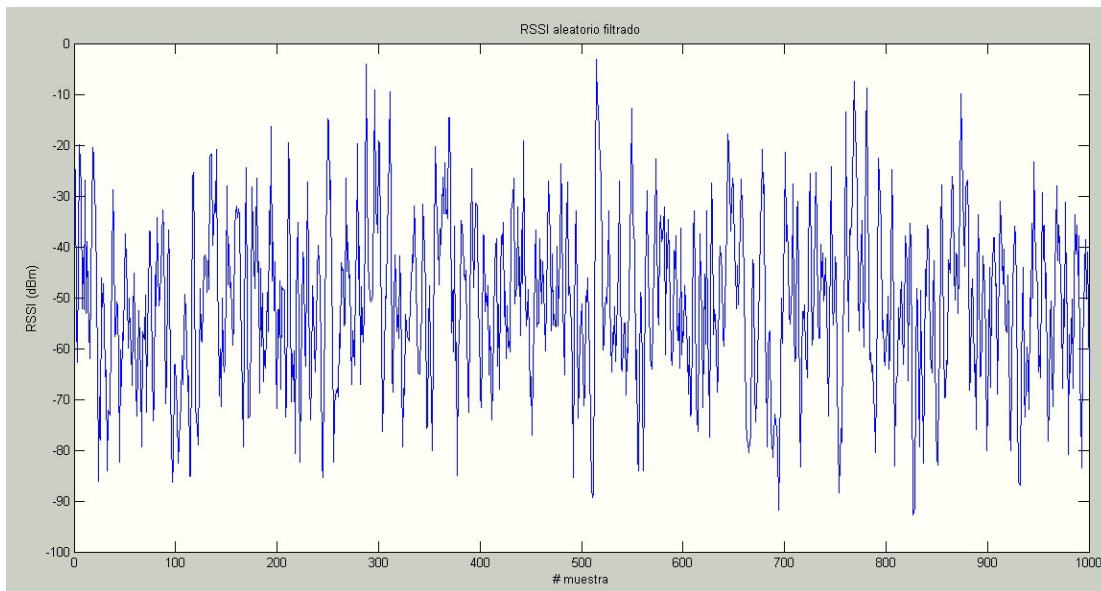


Figura 47. RSSI con filtro de media móvil de orden 3.

Fuente: los autores.

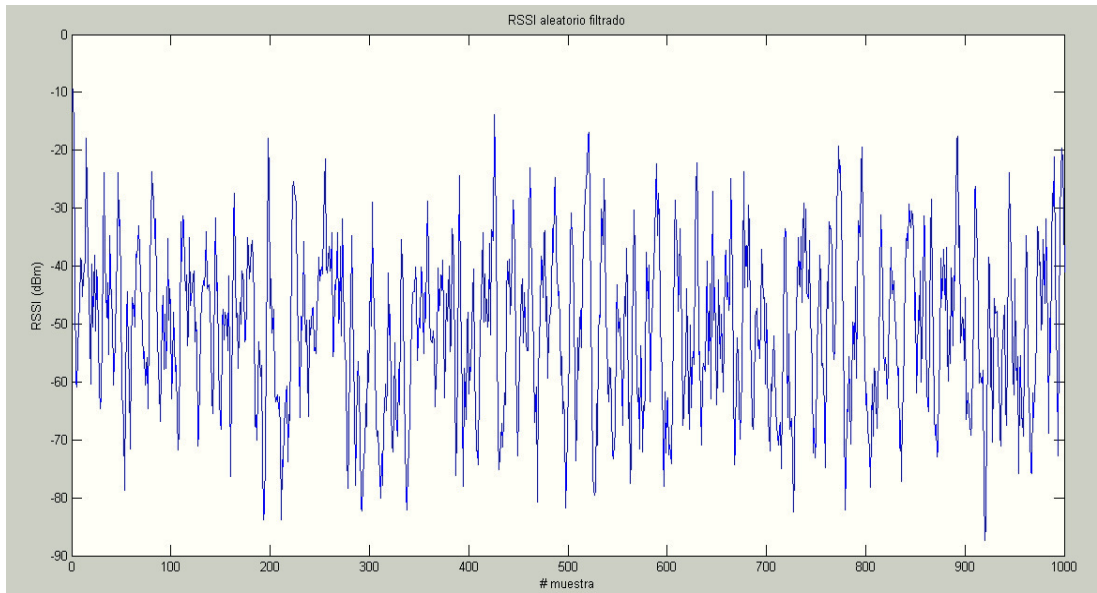


Figura 48. RSSI con filtro de media móvil de orden 4.

Fuente: los autores.

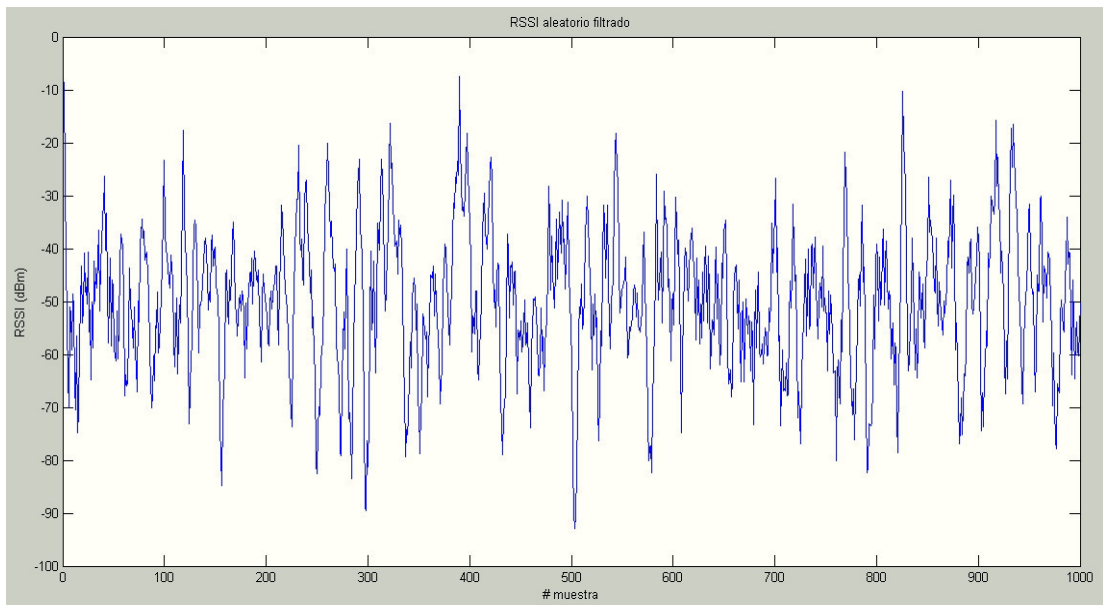


Figura 49. RSSI con filtro de media móvil de orden 5.

Fuente: los autores.

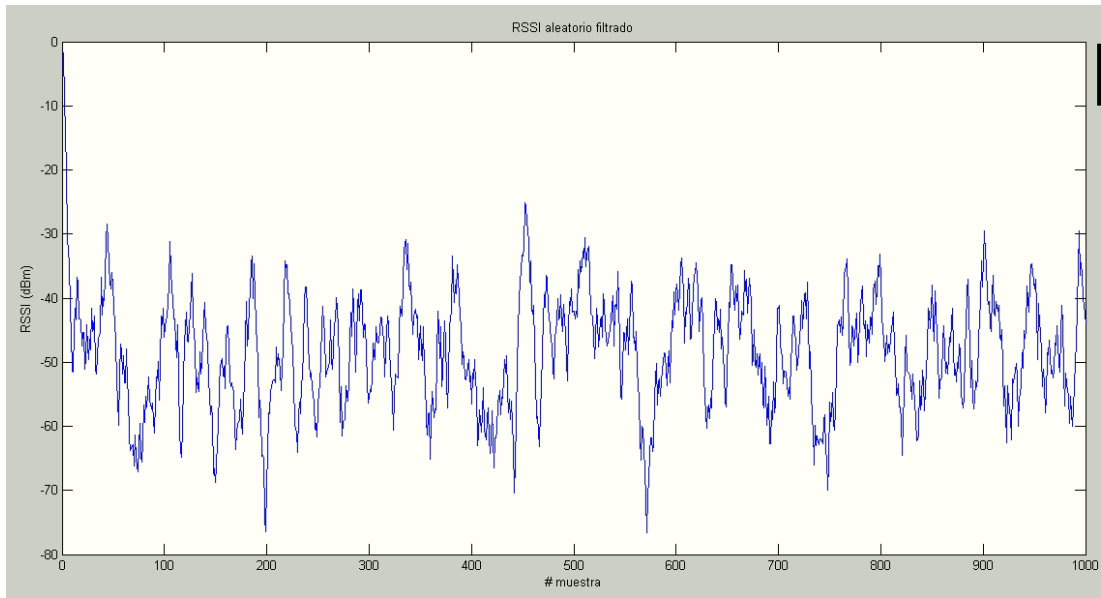


Figura 50. RSSI con filtro de media móvil de orden 10.

Fuente: los autores.

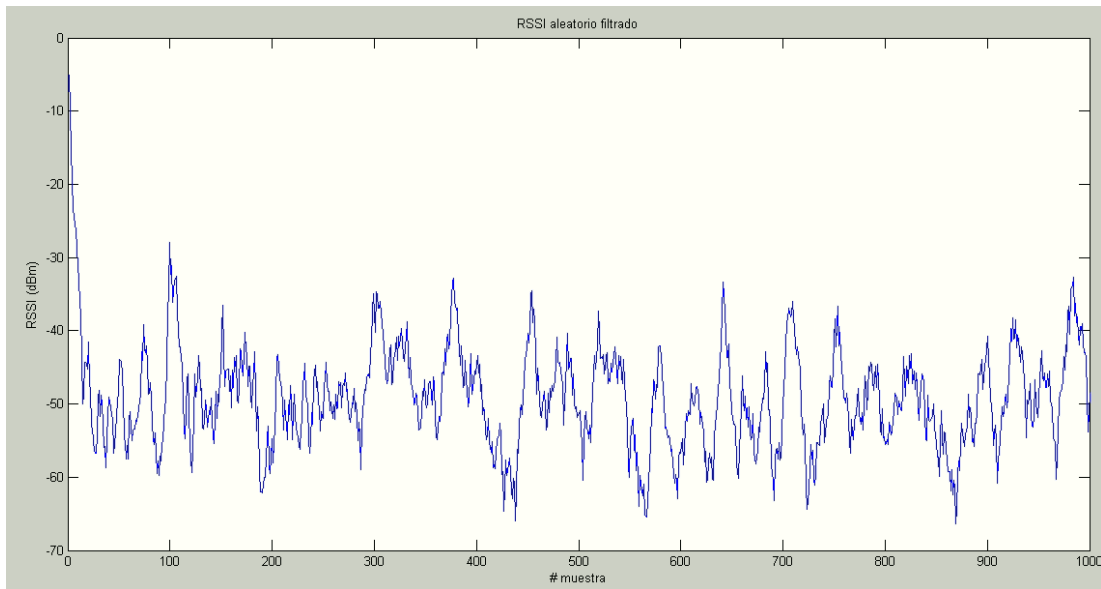


Figura 51. RSSI con filtro de media móvil de orden 15.

Fuente: los autores.

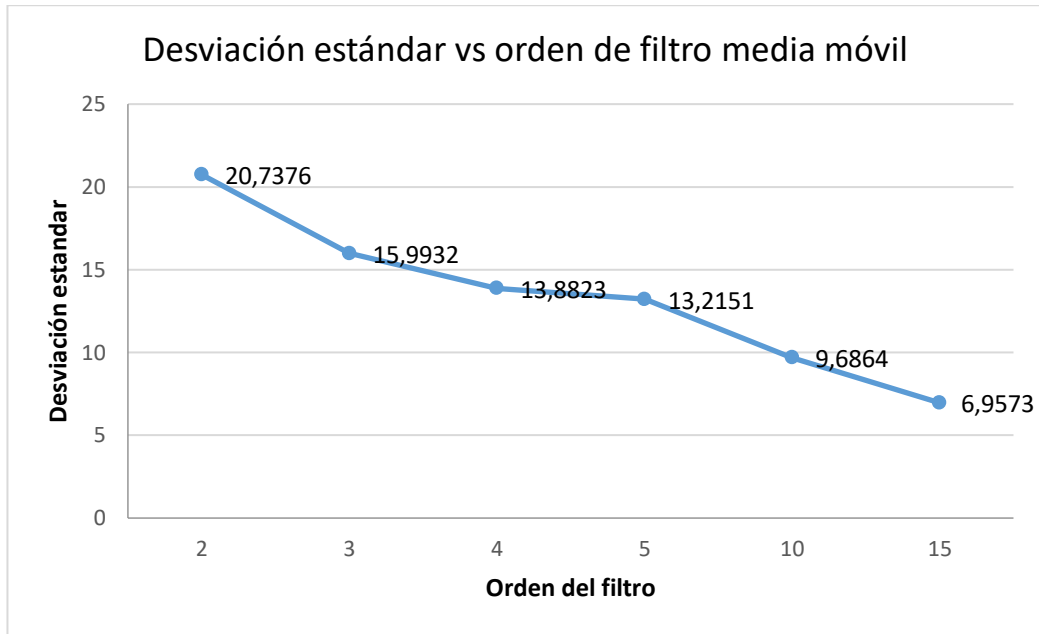


Figura 52. Desviación estándar vs orden de filtro de media móvil.

Fuente: los autores.

Como se puede observar, este filtro presenta la ventaja de no reducir la cantidad de muestras útiles, así mismo, presenta un mejor desempeño que el filtro de promedio aritmético por lo que finalmente será el filtro implementado para esta solución.

3.1.2 Perceptrón múltiple capa entrenado por la regla delta generalizada (backpropagation).

A continuación, se describió el cálculo matemático de un perceptrón múltiple capa entrenado por la regla delta generalizada, simulando la compuerta NAND.

Tabla 5: *Compuerta NAND.*

In_1	In_2	Z
0	0	1
0	1	1
1	0	1
1	1	0

Fuente: los autores.

Para la función de activación se usará una función sigmoideal binaria.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (31)$$

Se fijó la tasa de aprendizaje en $\eta = 0.1$ como se detalla a continuación:

La arquitectura del perceptrón múltiple capa utilizado para realizar el ejercicio es la siguiente:

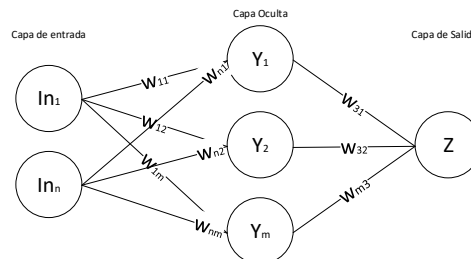


Figura 53. Perceptrón múltiple capa.

Fuente: los autores.

El primer paso por seguir es el cálculo de la salida entregada por la red neuronal por la neurona de salida “Z”, para realizar esto, los pesos de las sinapsis que interconectan las neuronas serán inicializadas utilizando valores aleatorios entre 0 y 1.

Luego se procede a realizar el cálculo de las entradas netas de las neuronas de la capa oculta:

$$net_{Y_1} = In_1 \times w_{11} + In_2 \times w_{21}$$

$$net_{Y_1} = 1 \times 0.32 + 0 \times 0.29 = 0.32 \quad (32)$$

Por consiguiente, la salida de la neurona Y1 fue la siguiente:

$$Y_1 = f(0.32) = \frac{1}{1 + e^{-0.32}}$$

$$Y_1 = 0.579 \quad (33)$$

Se procedió a realizar la misma operación para cada una de las neuronas de la capa oculta:

Tabla 6: Salidas de las neuronas en la capa oculta.

Neurona	Entrada neta (net_Y)	Salida (Y_n)
Y ₁	0.32	0.579
Y ₂	0.48	0.617
Y ₃	0.23	0.557

Fuente: los autores.

Finalmente, se calculó la entrada neta y salida de la neurona de la capa de salida:

$$\begin{aligned}
 net_Z &= Y_1 \times w_{31} + Y_2 \times w_{32} + Y_3 \times w_{33} \\
 net_Z &= 0.579 \times 0.23 + 0.617 \times 0.36 + 0.557 \times 0.44 \\
 net_Z &= 0.6 \\
 Z &= f(0.6) = \frac{1}{1 + e^{-0.6}} = 0.645 \tag{34}
 \end{aligned}$$

Luego se procede a realizar el cálculo del error de la red neuronal.

$$e = \frac{1}{2}(t_j - Z)^2 = \frac{1}{2}(1 - 0.645)^2 = 0.063 \tag{35}$$

Se procede a realizar el cálculo de la propagación hacia atrás, para esto se realiza el cálculo de los valores δ .

$$\begin{aligned}
 \delta_Z &= Z \times (1 - Z) \times e = 0.645 \times (1 - 0.645) \times 0.063 = 0.0144 \\
 \delta_{Y_1} &= Y_1 \times (1 - Y_1) \times w_{31} \times \delta_Z = 0.579 \times (1 - 0.579) \times 0.23 \times 0.0144 \\
 &= 8.07 \times 10^{-4} \\
 \delta_{Y_2} &= Y_2 \times (1 - Y_2) \times w_{32} \times \delta_Z = 0.617 \times (1 - 0.617) \times 0.36 \times 0.0144 \\
 &= 1.225 \times 10^{-3} \\
 \delta_{Y_3} &= Y_3 \times (1 - Y_3) \times w_{33} \times \delta_Z = 0.557 \times (1 - 0.557) \times 0.44 \times 0.0144 \\
 &= 1.563 \times 10^{-3} \tag{36}
 \end{aligned}$$

Luego se realiza la actualización de los pesos:

$$\Delta w_{11} = \eta \times \delta_{Y_1} \times In_1 = 0.1 \times 8.07 \times 10^{-4} \times 1 = 8.07 \times 10^{-5}$$

$$w_{11_{new}} = w_{11_{old}} + \Delta w_{11} = 0.32 + 8.07 \times 10^{-5} = 0.3200807 \quad (37)$$

Tabla 7: Actualización de pesos.

Peso	Δw_{ij}	$w_{ij_{new}}$
w_{11}	8.07×10^{-5}	0.3200807
w_{12}	1.225×10^{-4}	0.4801225
w_{13}	1.563×10^{-4}	0.2301563
w_{21}	0	0.29
w_{22}	0	0.5
w_{23}	0	0.41
w_{31}	8.337×10^{-4}	0.2308337
w_{32}	8.884×10^{-4}	0.3608884
w_{33}	8.021×10^{-4}	0.4408021

Fuente: los autores.

Luego se procede a realizar una verificación, ingresando nuevamente la misma entrada a la red neuronal y calculando el nuevo error:

Tabla 8. *Neuronas de salida.*

Neurona	Entrada neta (net_Y)	Salida (Y_n)
Y_1	0.3200807	0.579344
Y_2	0.4801225	0.617776
Y_3	0.2301563	0.557286
Z	0.6023331	0.646189

Fuente: los autores.

El nuevo error fue:

$$e = \frac{1}{2}(1 - 0.646189)^2 = 0.062591 \quad (38)$$

3.1.3 Especificaciones del diseño

Las especificaciones consideradas para el diseño final fueron las siguientes:

- El filtro a usarse es el filtro de media móvil, y este debió tener un orden igual a 4.
- La tasa de aprendizaje de la red neuronal debió de ser de 0.01.
- El error de la red neuronal debió de ser de 10^{-4} .
- El beacon estuvo ubicado a una altura de 2.41 metros.

3.2 Diseño

3.2.1 Área y zonas de trabajo

Para poder realizar la correcta implementación, se analizó el ambiente en donde se realizarán las pruebas (Laboratorio de Ingeniería Eléctrica 06-A-3 del pabellón de laboratorios) y se determinó que el área es de 6 x 10 m, a continuación, se presenta el plano de dicha área.

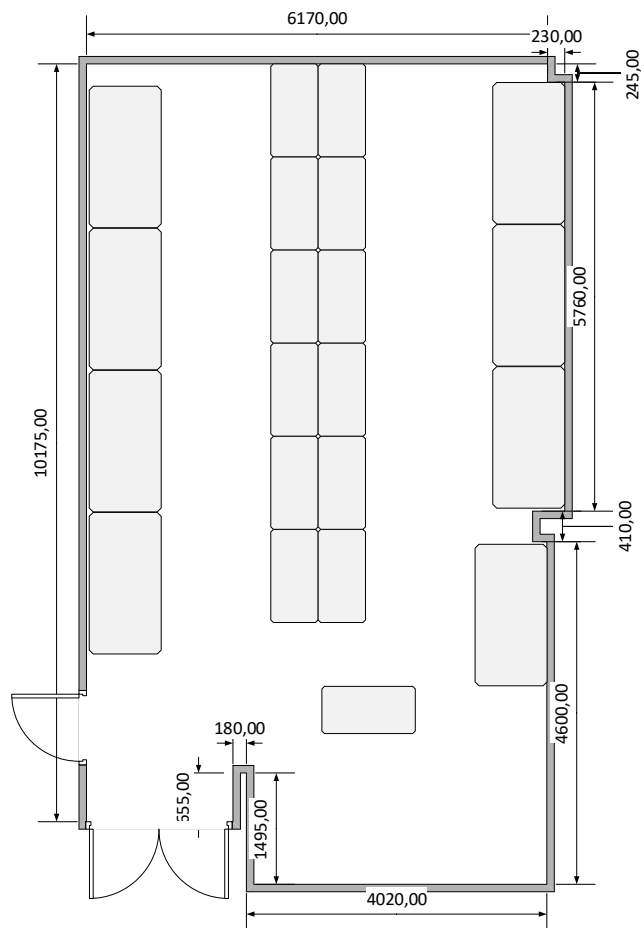


Figura 54. Plano del laboratorio de ingeniería eléctrica 06-A-3.

Fuente: los autores.

Para determinar el número de zonas en el área, se tomó en cuenta el error típico de la ubicación de los beacon en el modo de localización (1-4 m) que depende de las características específicas del ambiente, teniendo en cuenta los obstáculos presentes en el área de trabajo,

se separó el ambiente en seis (6) zonas independientes, detalladas a continuación (Estimote, 2017a).

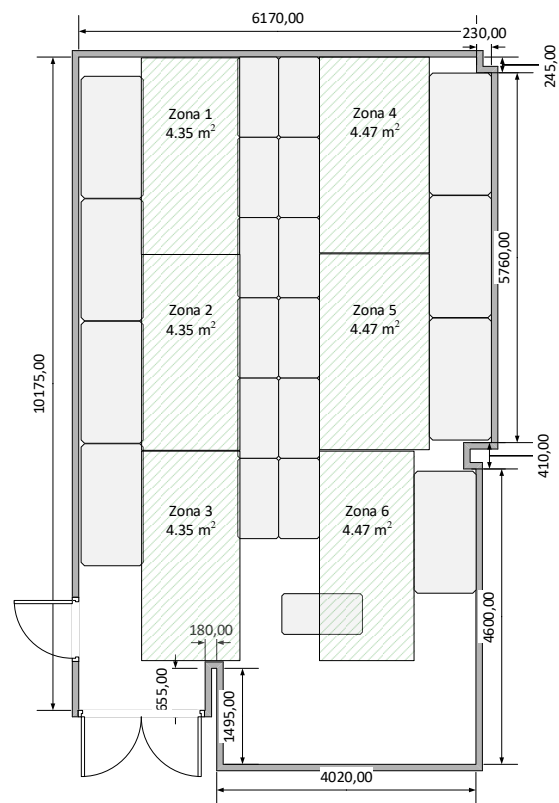


Figura 55. Plano del laboratorio de ingeniería eléctrica 06-A-3, incluye las zonas.

Fuente: los autores.

Como se detalla, en la investigación realizada por (Zafari & Papapanagiotou, 2015), indican que el despliegue de beacons en espacios de interiores a pequeña o gran escala no ha sido estandarizado, por lo que está basado puramente en la experiencia. Es por esto que para realizar la ubicación de los beacons dentro del área de prueba, se decidió basarse en el modelo de referencia de Aruba Networks (Hewlett Packard Enterprise Development LP, 2015) en el que indican que para espacios con techo cerrado, los beacons deben ser montados en el techo. En el área de trabajo, se cuenta con tres (3) alturas diferentes, el techo de concreto a 3.24 metros de altura, las rejillas para telecomunicaciones a 2.72 metros de altura y

finalmente, los ductos de aire acondicionado a 2.41 metros de altura, por lo tanto, los beacons serán ubicados a una altura de 2.41 metros de altura; sin embargo, como se menciona anteriormente, al no existir una regla establecida para la ubicación de los beacons, también fue considerado un escenario en donde los beacon se encuentren ubicados en las paredes del ambiente.

3.2.2 Filtro de media móvil

Para realizar el cálculo del filtro media móvil, se tomó en cuenta el número de muestras por segundo de RSSI enviadas por los beacon. La configuración de fábrica de los Beacon de proximidad de Estimote presenta un intervalo de broadcast de RSSI de 200 ms; sin embargo, fue modificado a 500 ms, lo que supone 2 muestras de RSSI por segundo, por lo que se decidió implementar el filtro para obtener un valor filtrado cada 2 segundos, esto significa implementar el filtro de media móvil de orden 4. Finalmente, el valor final de RSSI obtenido luego de filtrar las 4 muestras obtenidas en el lapso de tiempo establecido anteriormente, vendrá dado por la siguiente formula:

$$RSSIn_f = \frac{1}{4} \sum_{t=1}^4 RSSIn_{t-1} \quad (39)$$

3.2.3 Normalización de los RSSI

Asimismo, según lo detallado en las investigaciones de (Muhammad, 2007; Kriz, Maly, & Kozel, 2016; Kalbandhe & Patil, 2016), para obtener una mayor precisión en la ubicación y evitar errores en la lectura del RSSI, es conveniente normalizar los datos y reducir los picos obtenidos utilizando filtros.

Para normalizar los datos se analizó el valor mínimo de las muestras tomadas:

$$RSSI_{min} = -100 \quad (40)$$

Por consiguiente, el valor normalizado del RSSI obtenido estará dado por la siguiente formula:

$$In_n = \frac{RSSIn_f}{RSSI_{min}} = \frac{RSSIn_f}{-100} \quad (41)$$

3.2.4 Ubicación de beacons

Para determinar la cantidad total de beacons a ubicar dentro del área de trabajo, se realizaron pruebas con la red neuronal con diferente cantidad de beacons, comenzando con 2 beacon e incrementándolos de uno en uno hasta encontrar una caída en el performance de la red neuronal debido a la interferencia entre beacons, en los gráficos siguientes se detallan las distribuciones de beacon implementadas al realizar las pruebas.

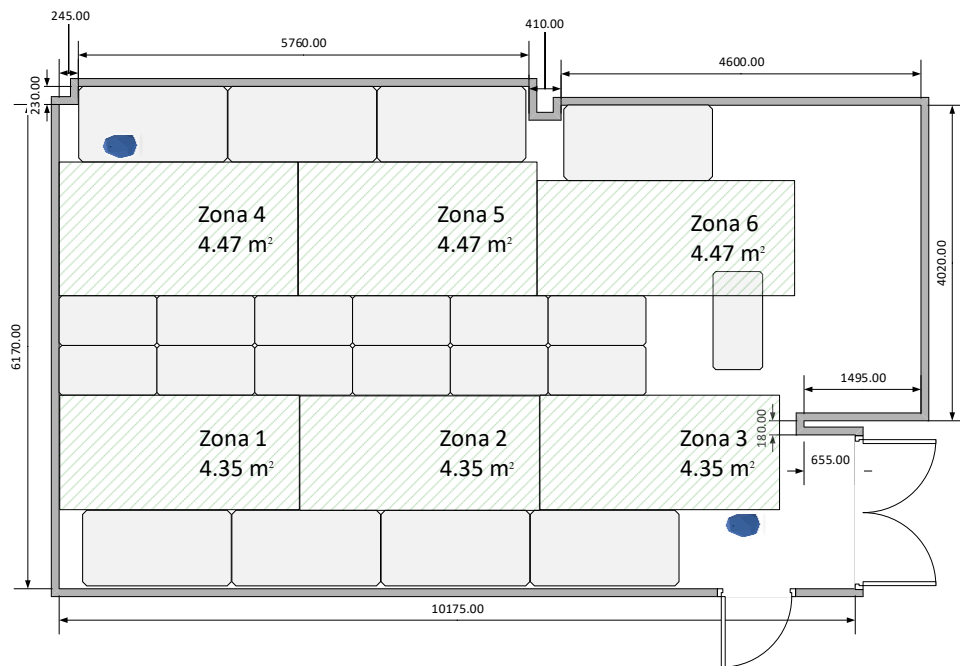


Figura 56. Área de trabajo con 2 beacons.

Fuente: los autores.

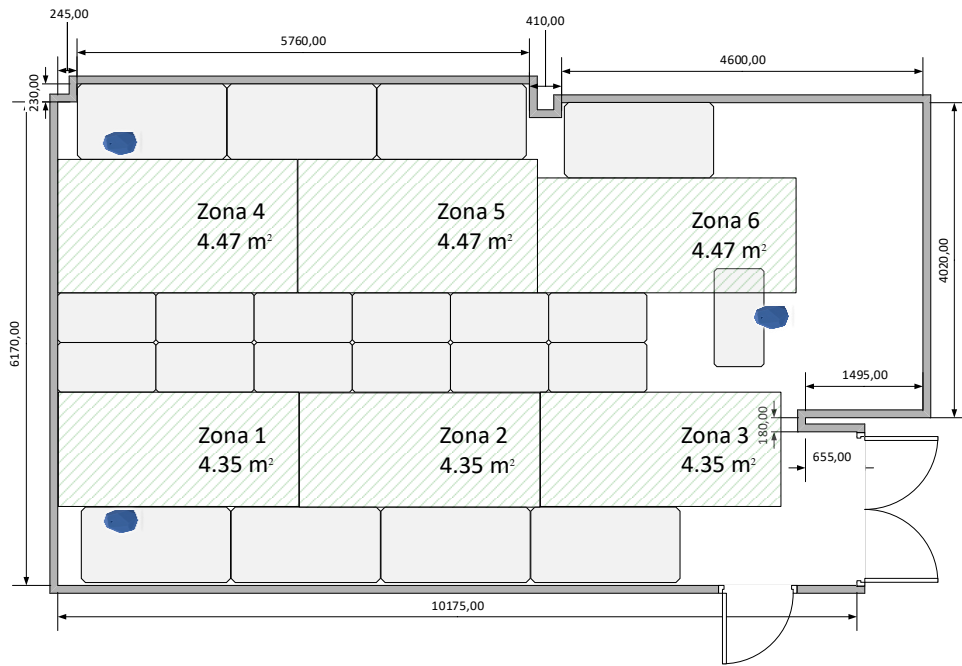


Figura 57. Área de trabajo con 3 beacons, primera ubicación en el techo.

Fuente: los autores.

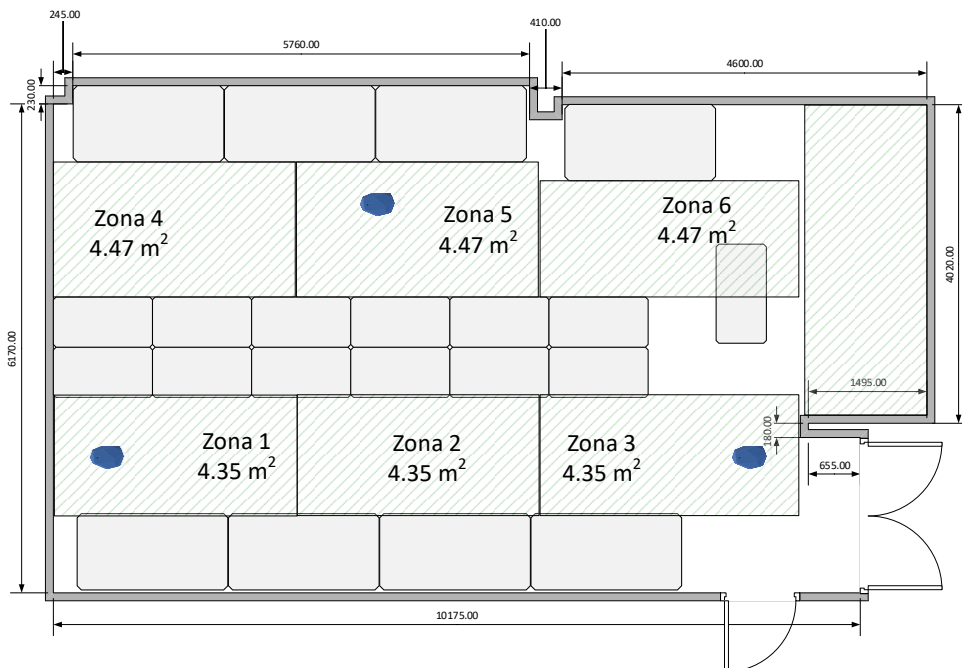


Figura 58. Área de trabajo con 3 beacons, segunda ubicación en el techo.

Fuente: los autores.

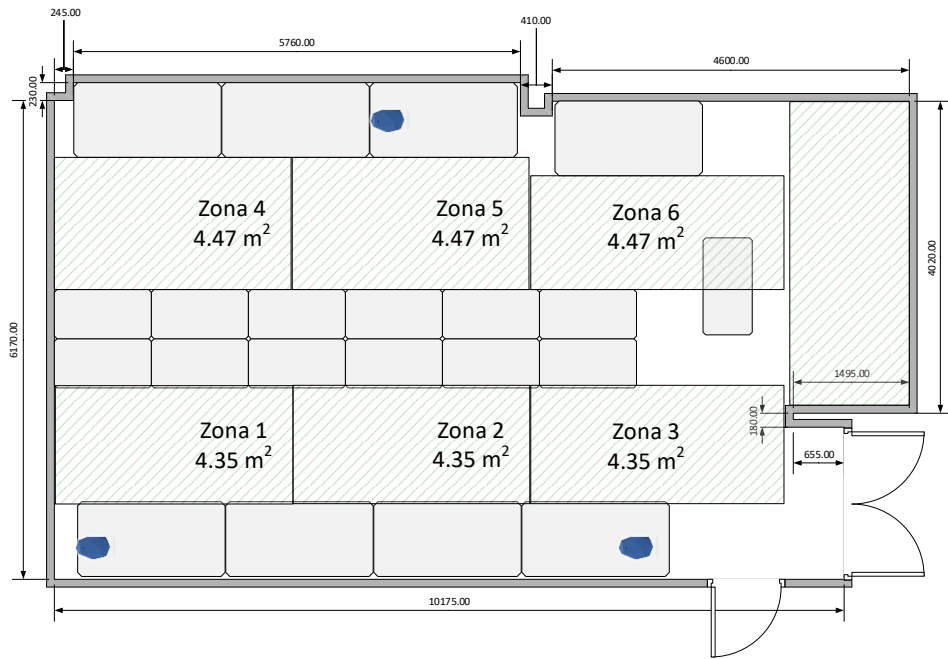


Figura 59. Área de trabajo con 3 beacons ubicados sobre los módulos.

Fuente: los autores.

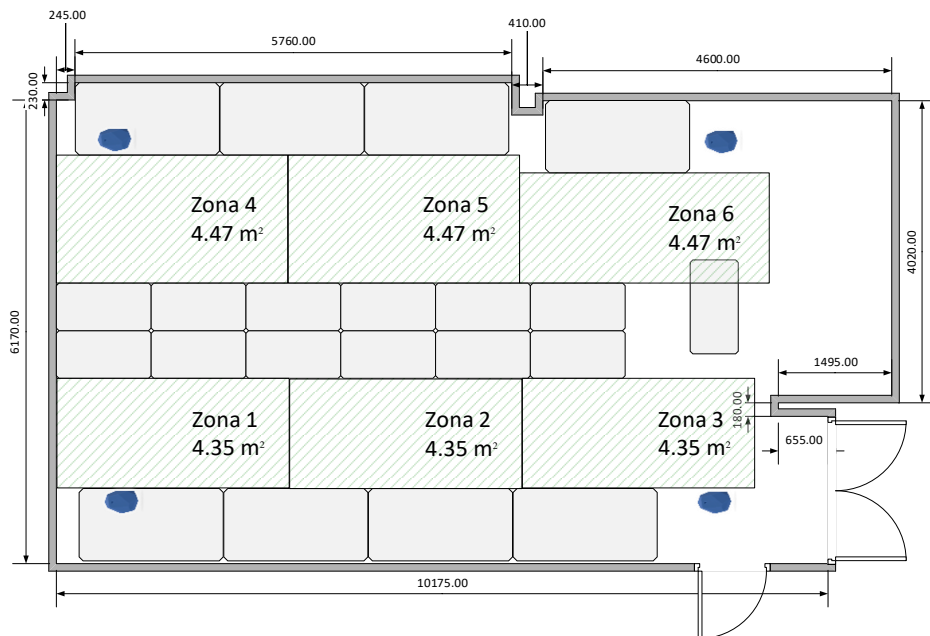


Figura 60. Área de trabajo con 4 beacons, ubicación en el techo.

Fuente: los autores.

3.2.5 Red Neuronal

A continuación, se procedió a realizar pruebas de la red neuronal tomando en cuenta todos los parámetros antes mencionados, para lo cual se utilizó la función `feedforwardnet` de Matlab, el código implementado para realizar las pruebas fue el siguiente:

Normalización de datos

```
x=x/(-100);
```

Filtro de media móvil de 4 muestras:

```
n=4;
```

```
p=(1/n)*ones(1,n);
```

```
q=1;
```

```
std(x1)
```

```
RSSIF=filter(p,q,x1);
```

Donde:

n: Número de muestras con las que se desea promediar, orden del filtro.

p: Numerador de la función de transferencia del filtro.

q: Denominador de la función de transferencia del filtro

x1: Vector de RSSI.

Vale recalcar, que se tendrán más variables "RSSIF" dependiendo del número de beacons que se esté utilizando, es decir, si se

tienen 4 beacons de entrada, se utilizaran las variables, $RSSI_{F1}$, $RSSI_{F2}$, $RSSI_{F3}$, $RSSI_{F4}$.

Red neuronal:

```
p=4;
```

```
net=feedforwardnet([p]);
```

```
net.trainFcn = 'traingd';
```

```
net.trainParam.epochs = 120000;
```

```
net.trainParam.goal=10(-4);
```

```
net.trainParam.lr=0.01;
```

```
net=train(net,b,result);
```

```
view(net)
```

Donde:

p: número de neuronas en la capa oculta

Traingd: Función de entrenamiento por descenso de gradiente.

net.trainParam.epochs: Número de épocas máximas a entrenar la red neuronal.

net.trainParam.goal: Error mínimo a alcanzar por la red neuronal.

net.trainParam.lr: Tasa de entrenamiento de la red neuronal.

b: Vector de entrada de la red neuronal, puede darse con los datos filtrados, sin filtrar, normalizados o sin normalizar.

result: Vector de salidas deseadas.

Para poder realizar las pruebas de la red neuronal, se realizaron mediciones durante 4 minutos aproximadamente, obteniendo un total de 250 muestras por cada una de las zonas utilizando las diferentes distribuciones de beacon, obteniendo matrices con tamaño desde 250x2 hasta los 250x4 por cada una de las zonas, dando un total de 1500x2 hasta 1500x4 analizando el total de las zonas.

Para poder realizar una mejor comparación del desempeño de la red neuronal, se realizaron múltiples pruebas con los datos de entrada normalizados y sin normalizar, así como también, con los datos filtrados y con los datos sin filtrar. También se procedió a determinar el número de neuronas en la capa oculta de la red neuronal. A continuación, se presentan los errores obtenidos por la red neuronal con cada una de las distribuciones de beacon y neuronas en la capa oculta:

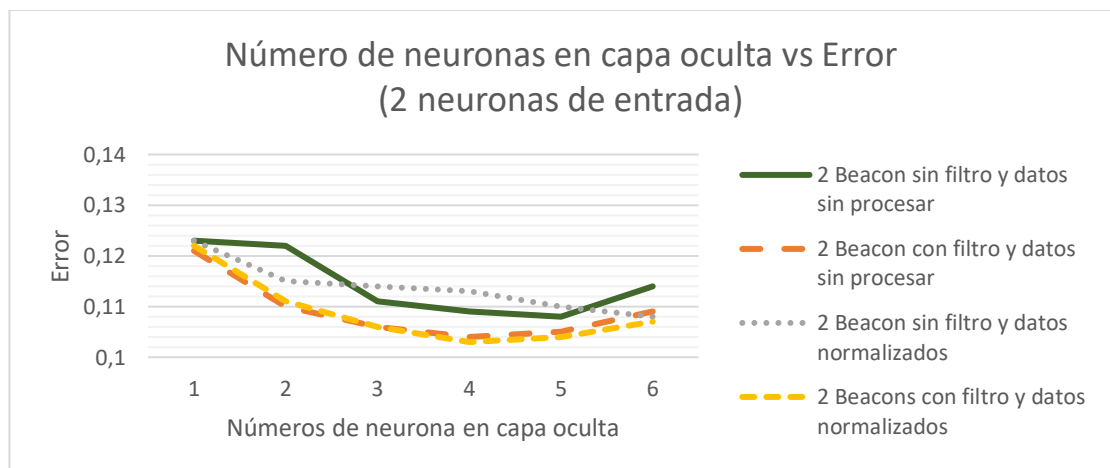


Figura 61. Número de neuronas en capa oculta vs error (2 neuronas de entrada).

Fuente: los autores.

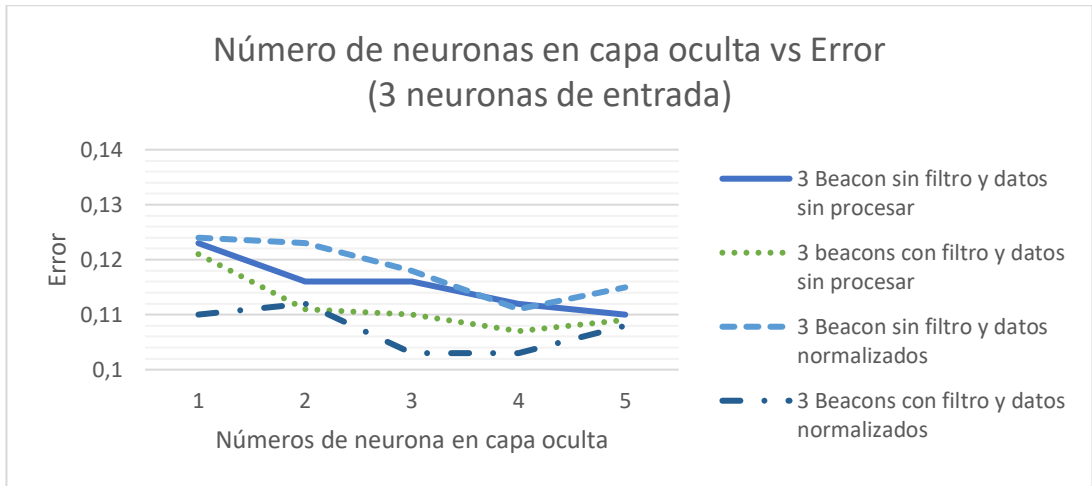


Figura 62. Número de neuronas en capa oculta vs error (3 neuronas de entrada).

Fuente: los autores.

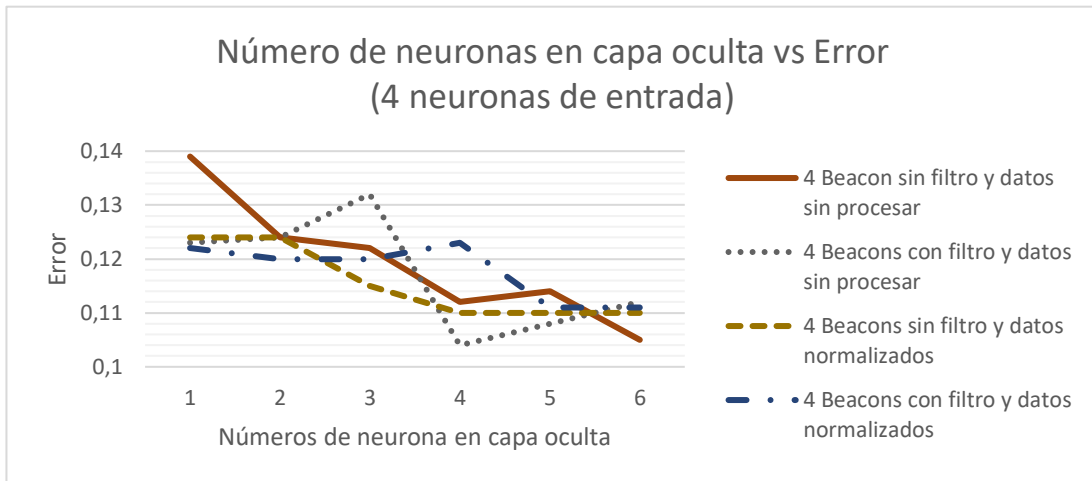


Figura 63. Número de neuronas en capa oculta vs error (4 neuronas de entrada).

Fuente: los autores.

A continuación, se detalla el mejor performance obtenido comparando el uso del filtro y la normalización para cada grupo de neuronas de entrada.

Tabla 9: *Error de red neuronal.*

Tipo de datos	Filtro	Número de neuronas de entrada	Número de neuronas en capa oculta	Error
Sin normalizar	si	2	4	0.104
Normalizados	si	2	4	0.103
Sin normalizar	si	3	4	0.107
Normalizados	si	3	4	0.103
Sin normalizar	si	4	4	0.104
Normalizados	no	4	4	0.11

Fuente: los autores.

Como se puede observar, para el espacio estudiado, se obtiene un performance similar tanto con 2 neuronas de entrada con los datos de entrada normalizados y filtrados, así como con 3 neuronas de entrada con sus datos normalizados y filtrados, en todos los casos se observa que el mejor performance se obtiene con 4 neuronas en la capa oculta. Por consiguiente, se optó por utilizar 3 beacons para la capa de entrada con 4 neuronas en la capa oculta.

En la figura 64, podemos observar la topología final de la red neuronal a implementar en el proyecto de tesis.

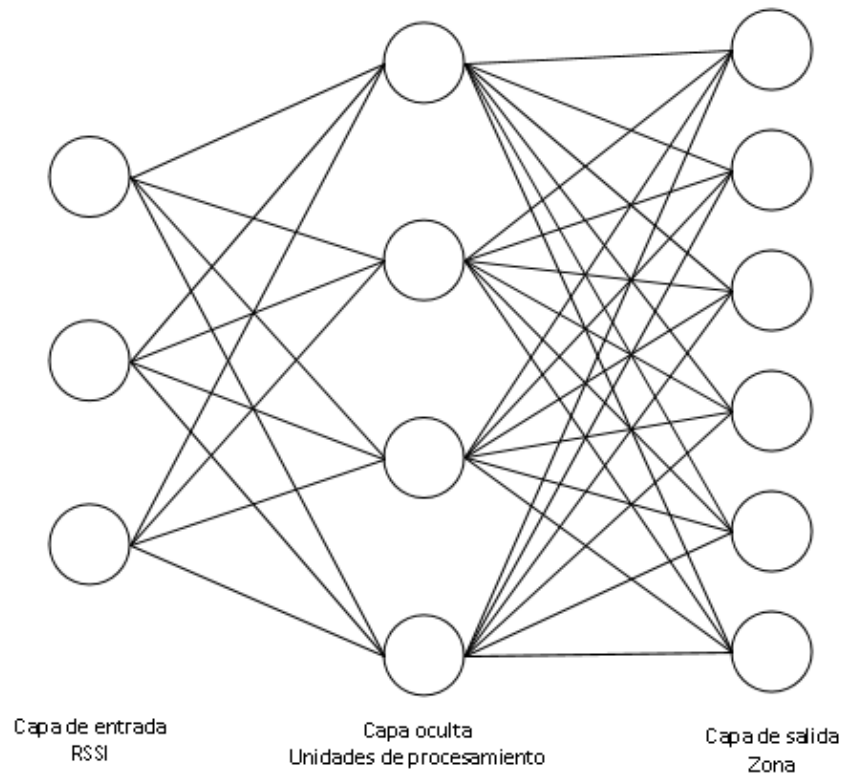


Figura 64. Red neuronal final.

Fuente: los autores.

Debido a que la red neuronal presento un mejor desempeño con los datos normalizados y utilizando el filtro aritmético, la forma final de la red neuronal más el preprocesamiento de datos será la siguiente:

3.2.6 Diseño final – Jerarquía H2

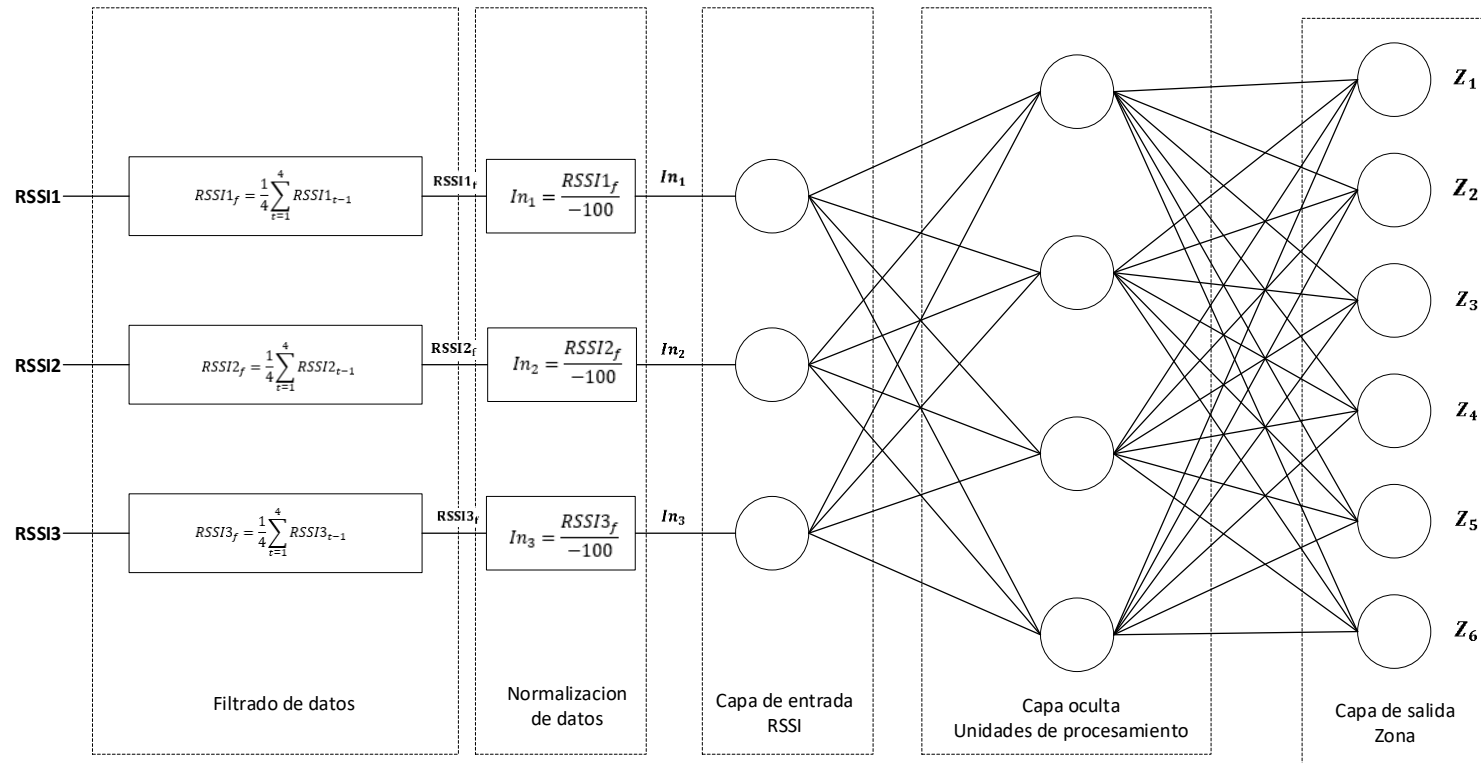


Figura 65. Diagrama de bloques del proceso final de la detección de zona.

Fuente: los autores.

3.2.7 Aplicación en Android (cliente)

Finalmente, se realizó la implementación de dicho diseño en los softwares MIT App Inventor 2 y Microsoft Visual Studio C#, a continuación, se detallan los diagramas de bloques de dichas secuencias de código:

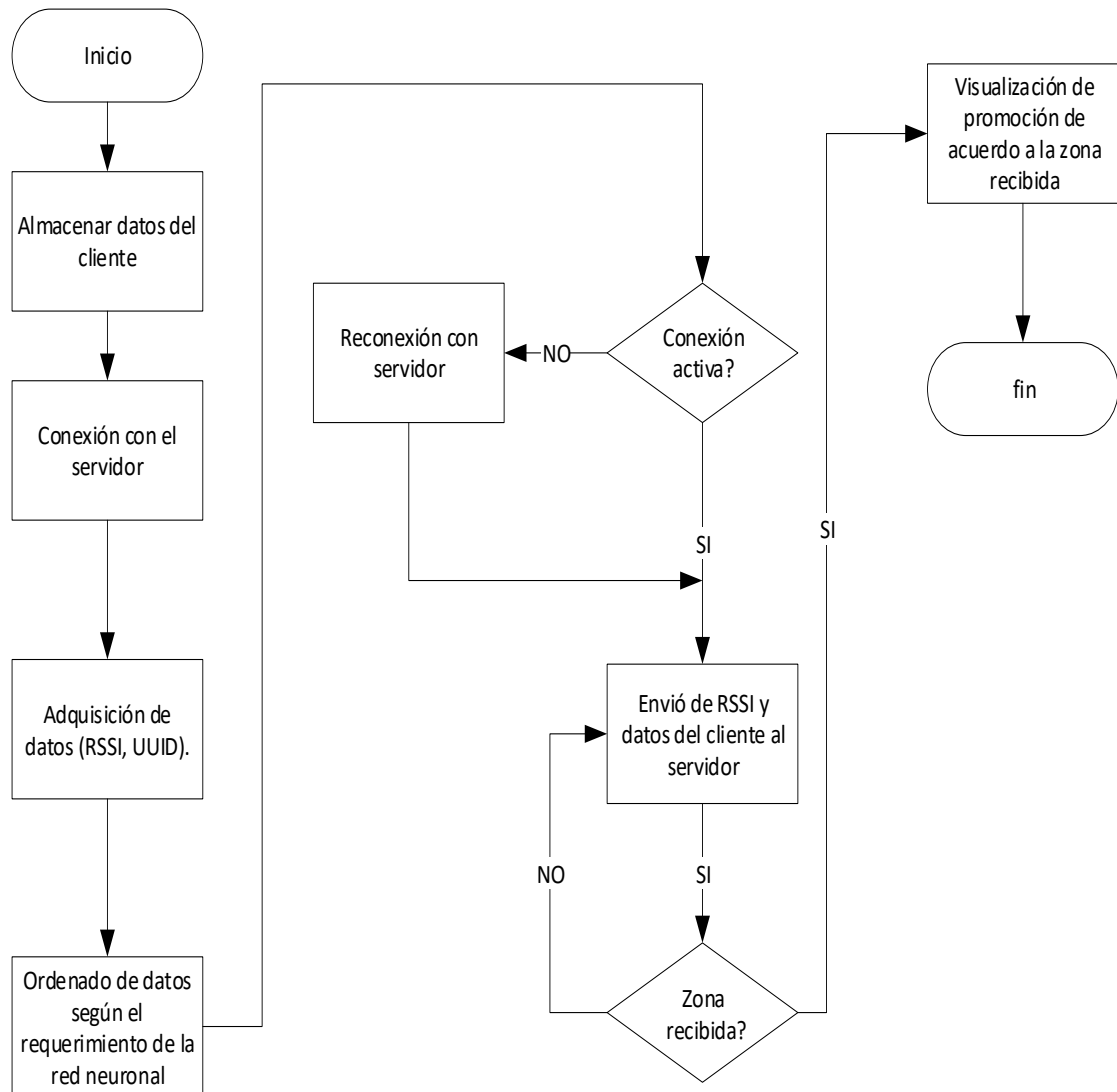


Figura 66. Diagrama de flujo de la aplicación en Android (Cliente).

Fuente: los autores.

3.2.8 Aplicación en C# (servidor)

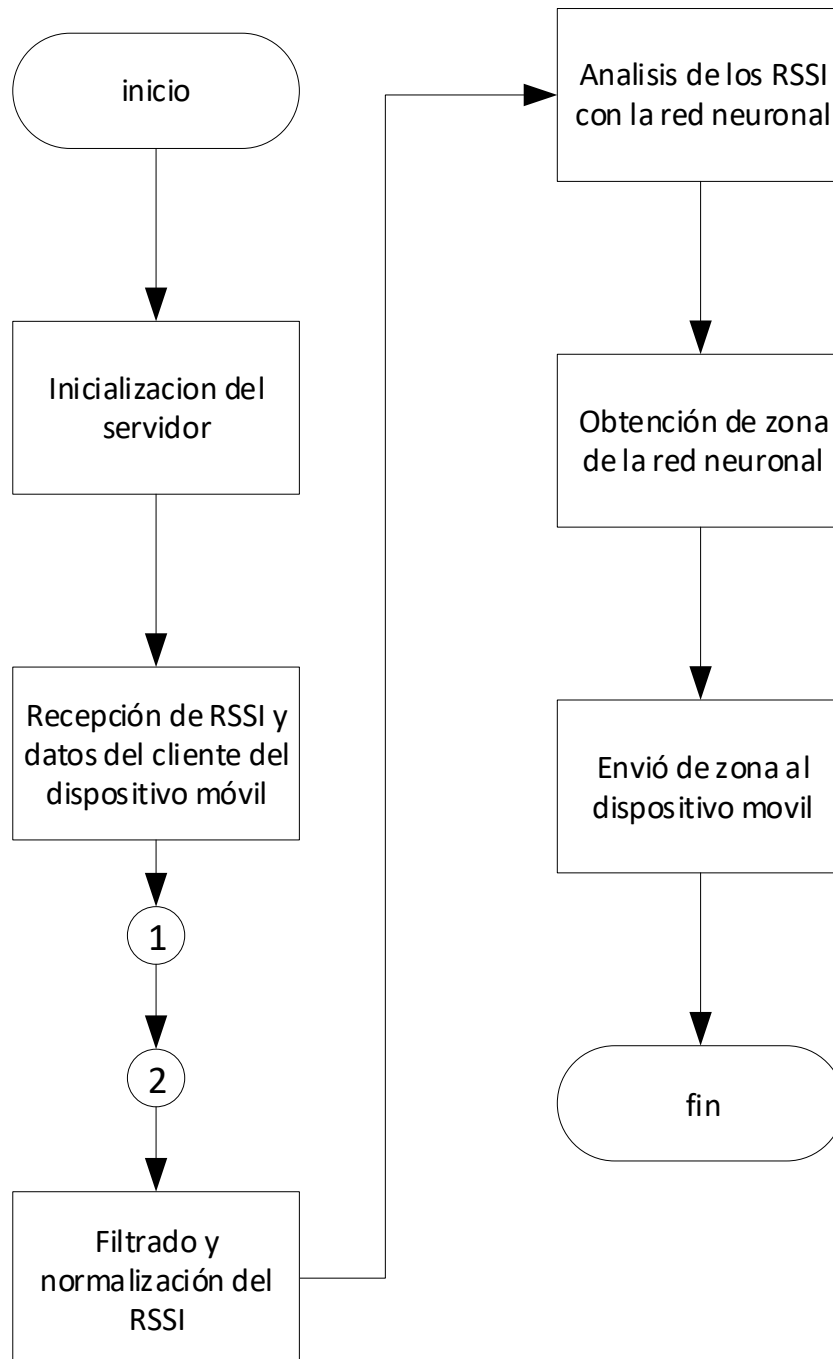


Figura 67. Diagrama de flujo de aplicación en C# (Servidor).

Fuente: los autores.

Finalmente, para recibir y almacenar la información de los usuarios en la aplicación del servidor, se agregaron líneas de código a la aplicación de C#. A continuación, se presenta el diagrama de flujo de dicha implementación:

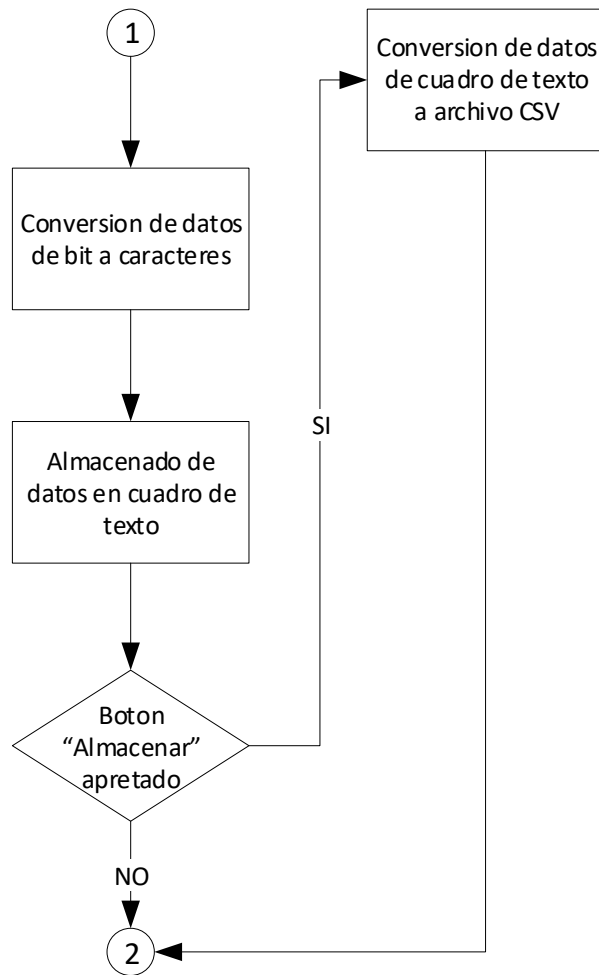


Figura 68. Sub diagrama de flujo de aplicación en C# (Servidor).

Fuente: los autores.

CAPÍTULO IV

PRUEBAS Y RESULTADOS

En primer lugar, se analizaron las muestras de RSSI recibidas por el dispositivo móvil las cuales se detallan a continuación:

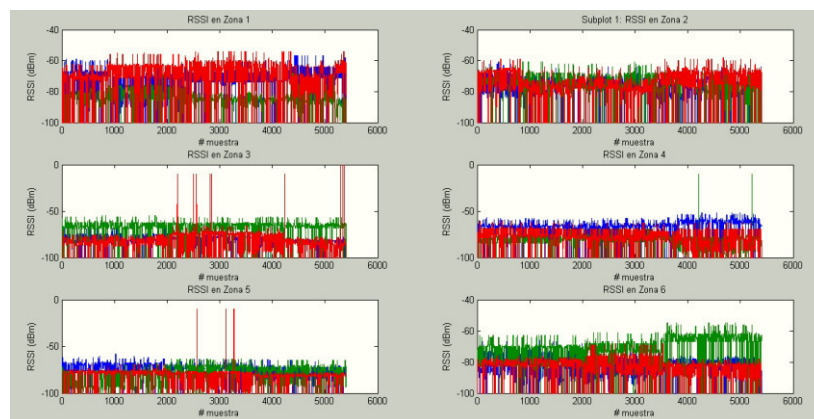


Figura 69. RSSI obtenido en las diferentes “Zonas”.

Fuente: los autores.

Como se puede observar en la figura 69, se reciben de manera constante valores de -100 (error), esto debido a que en la etapa de adquisición de datos tanto el teléfono móvil y los beacons trabajan con la misma frecuencia de reloj de 500 ms, pero, estos mismos no se encuentran sincronizados, por lo que el dispositivo móvil veía falsos errores.

Para poder resolver este constante error fue necesario, que los beacons encargados de generar el valor RSSI se les disminuyera la frecuencia de envío de RSSI a 100 ms, el teléfono móvil encargado de censar el valor RSSI mantuvo la frecuencia de 500 ms; de esta manera se redujeron las probabilidades de una medición errónea, en la figura 70 se puede observar las muestras de RSSI obtenidas de las diferentes zonas luego de cambiar las frecuencias de muestreo:

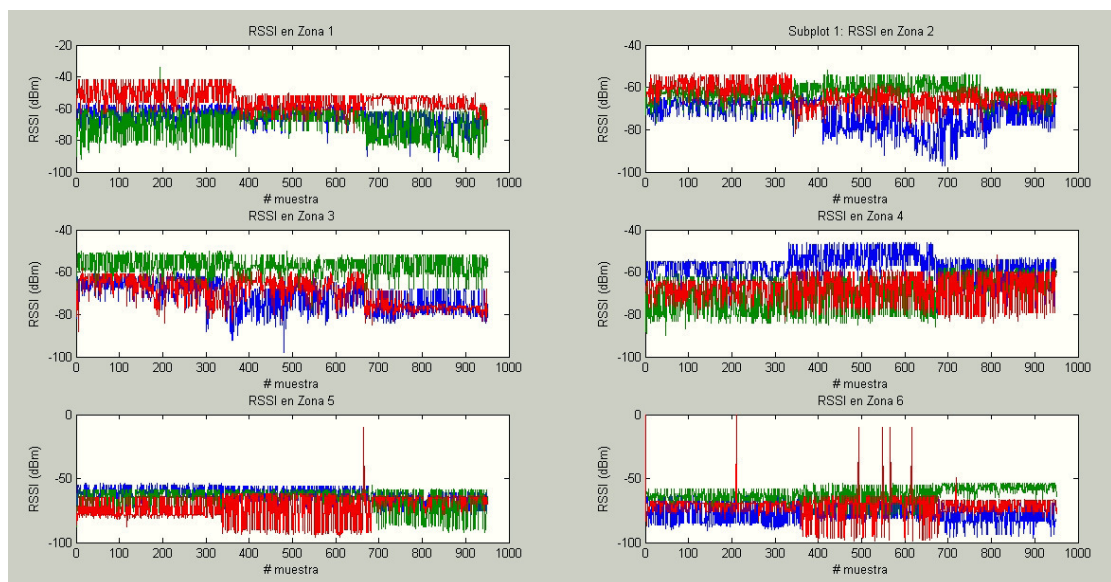


Figura 70. RSSI obtenido luego de modificar la frecuencia de los beacons.

Fuente: los autores.

Luego se procedió a aplicar el filtro de media móvil, luego se detallan los RSSI obtenidos luego de cambiar el orden del filtro, asimismo, se muestra una gráfica de disminución de desviación estándar respecto al orden del filtro.

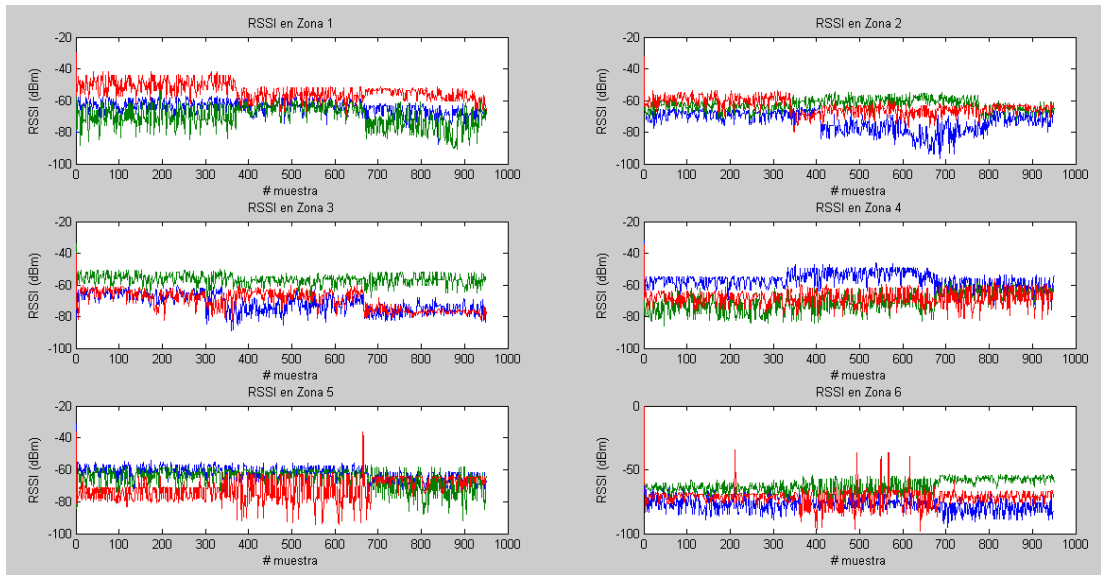


Figura 71. RSSI con filtro de media móvil de orden 2.

Fuente: los autores.

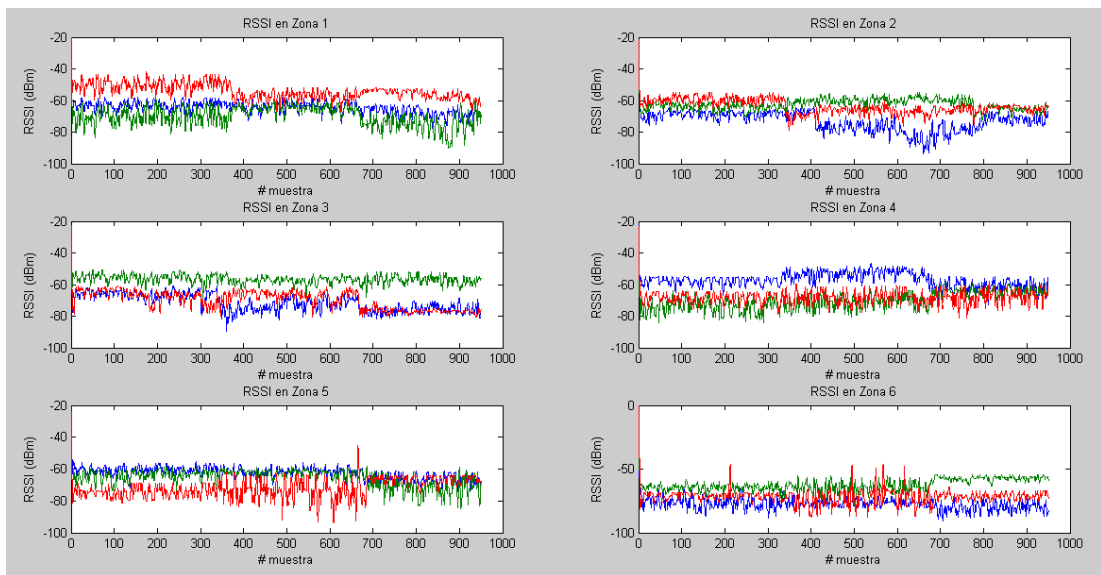


Figura 72. RSSI con filtro de media móvil de orden 3.

Fuente: los autores.

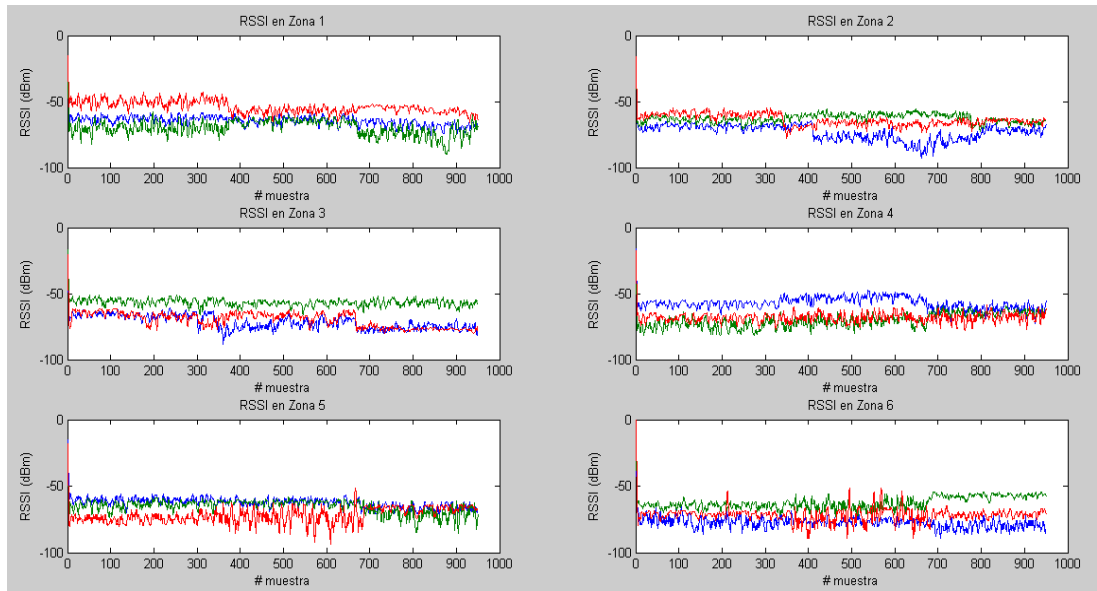


Figura 73. RSSI con filtro de media móvil de orden 4.

Fuente: los autores.

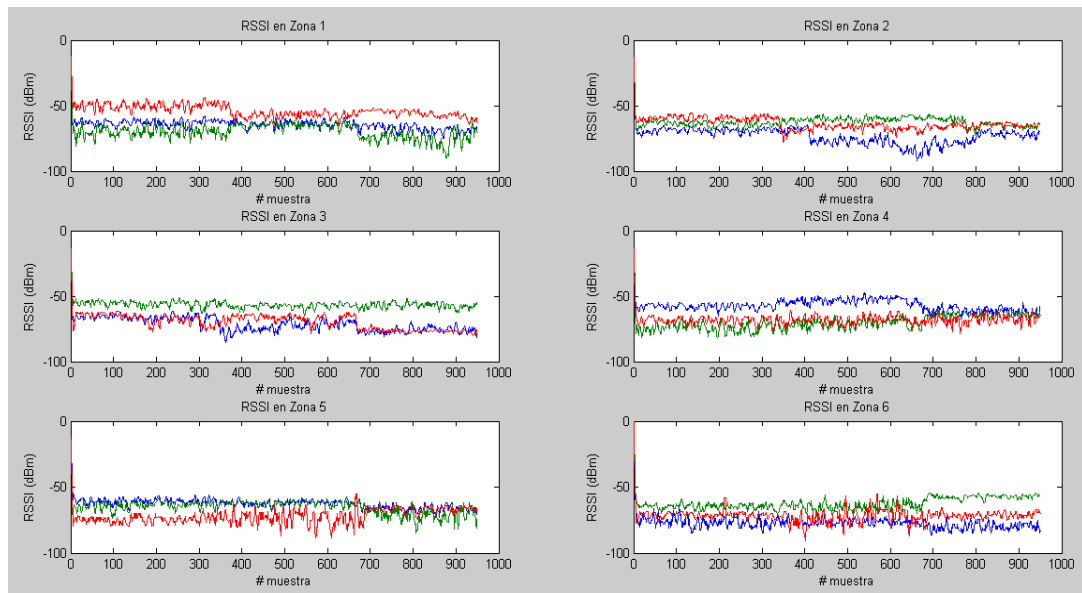


Figura 74. RSSI con filtro de media móvil de orden 5.

Fuente: los autores.

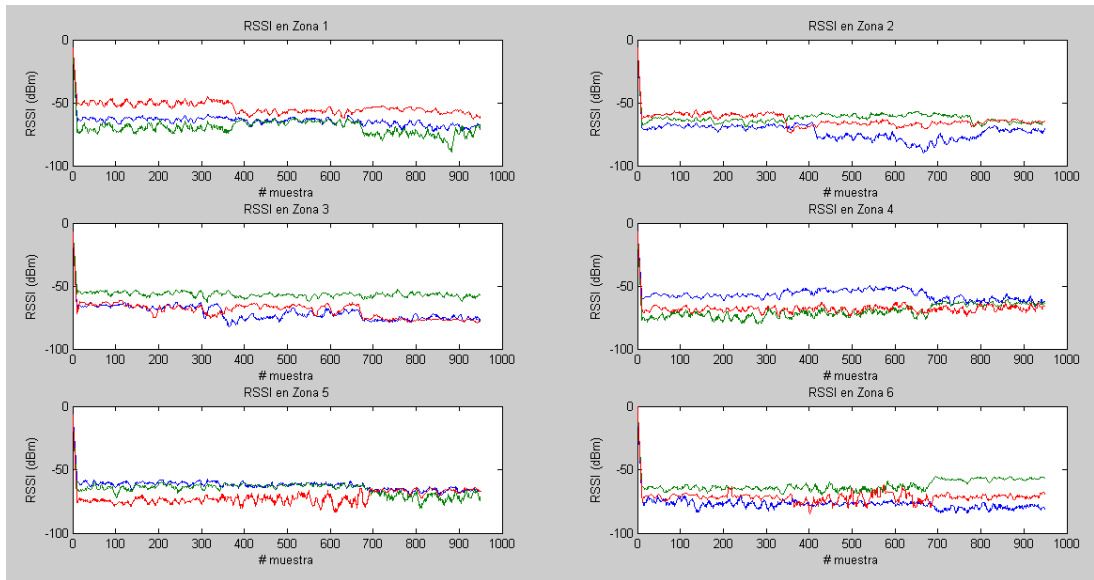


Figura 75. RSSI con filtro de media móvil de orden 10.

Fuente: los autores.

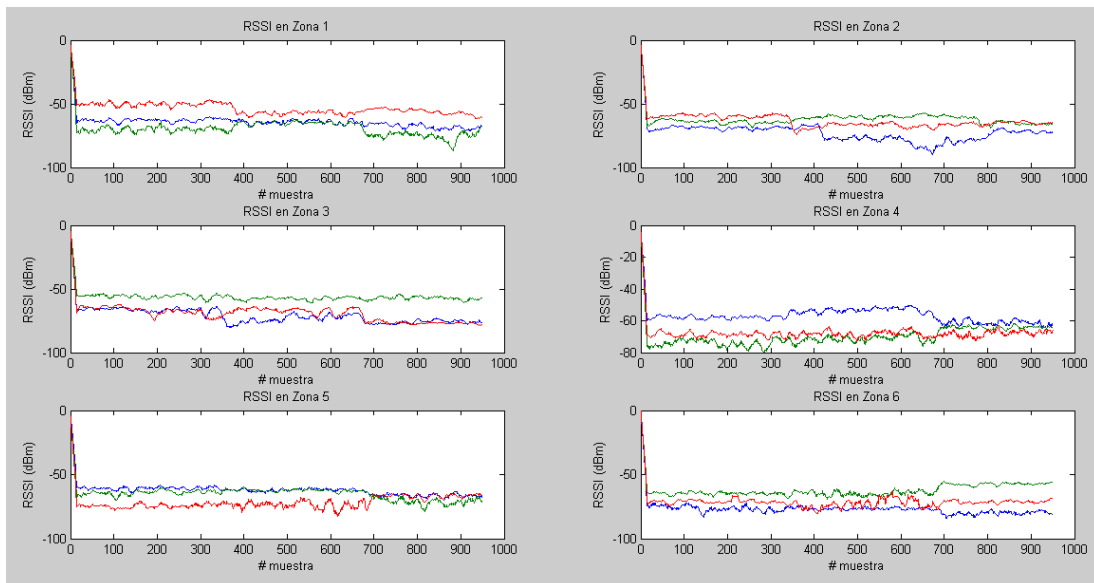


Figura 76. RSSI con filtro de media móvil de orden 15.

Fuente: los autores.

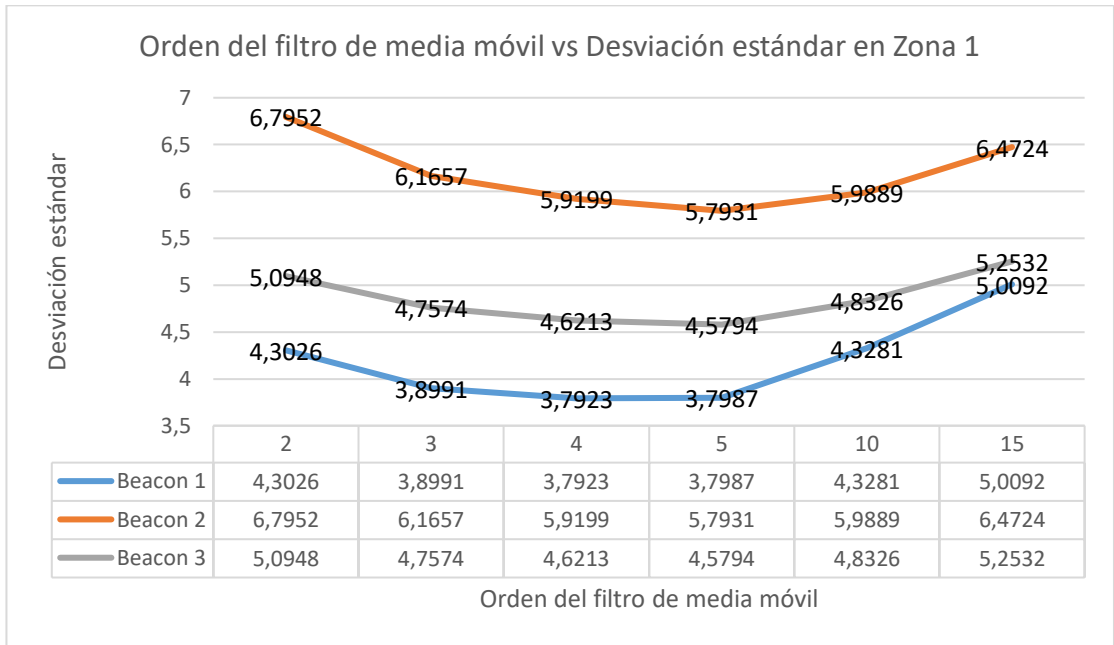


Figura 77. Orden del filtro de media móvil vs Desviación estándar en Zona 1.

Fuente: los autores.

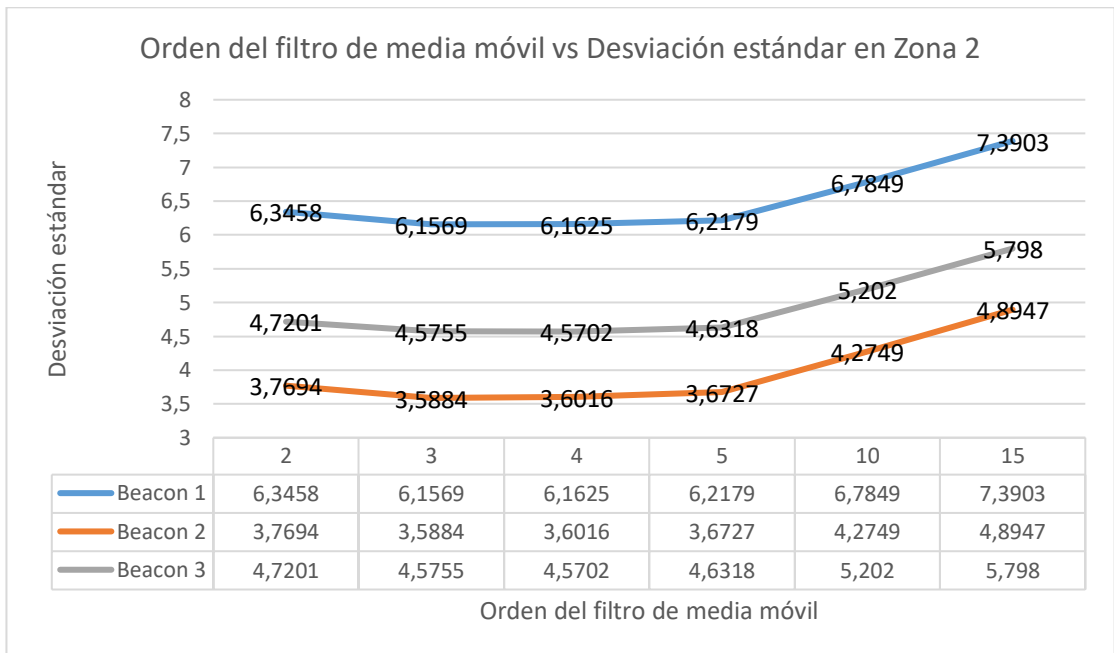


Figura 78. Orden del filtro de media móvil vs Desviación estándar en Zona 2.

Fuente: los autores.

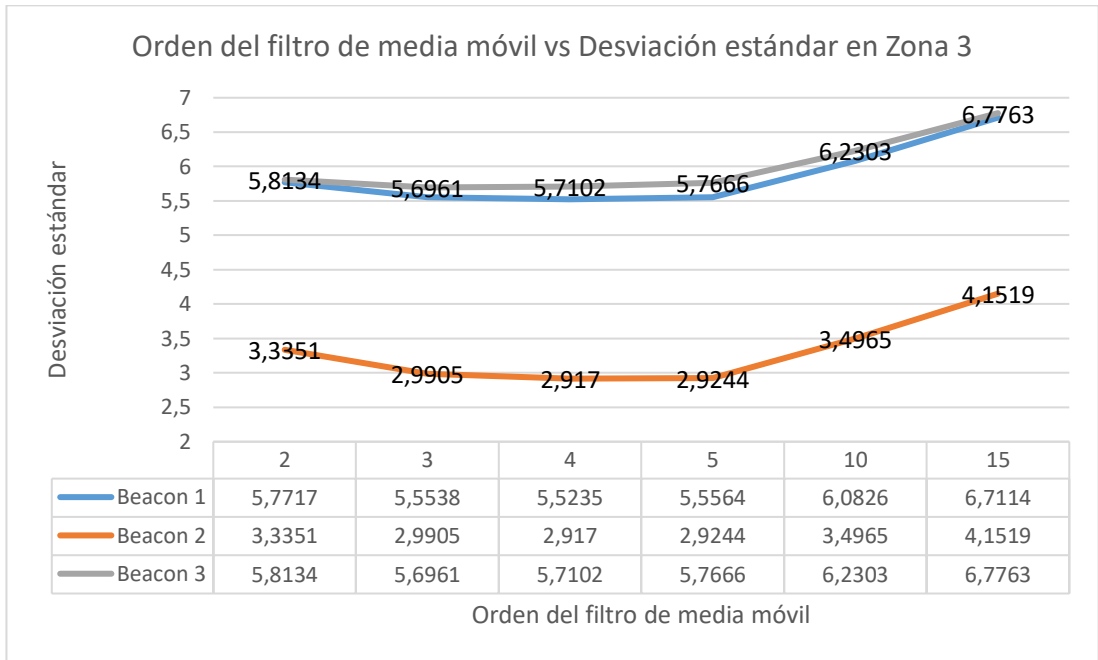


Figura 79. Orden del filtro de media móvil vs Desviación estándar en Zona 3.

Fuente: los autores.

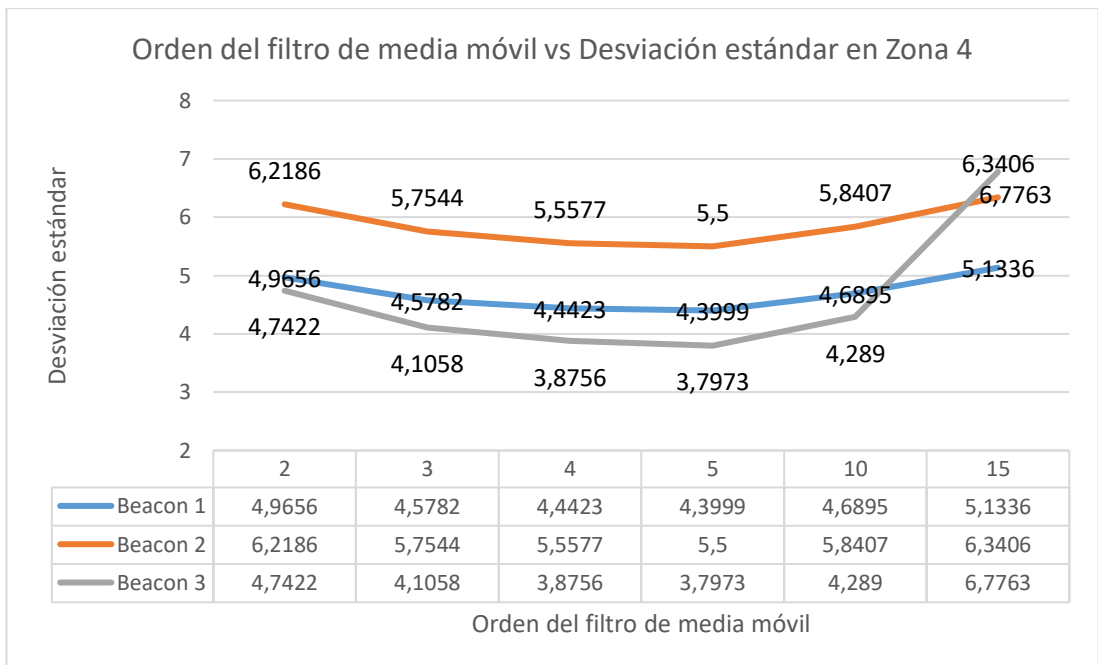


Figura 80. Orden del filtro de media móvil vs Desviación estándar en Zona 4.

Fuente: los autores.

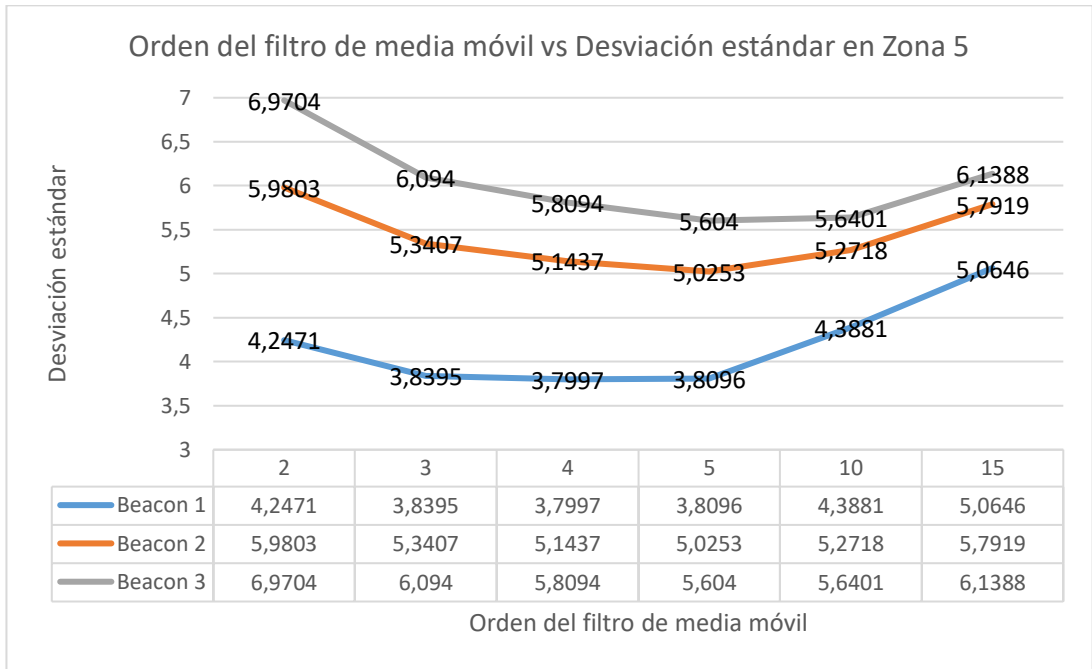


Figura 81. Orden del filtro de media móvil vs Desviación estándar en Zona 5.

Fuente: los autores.

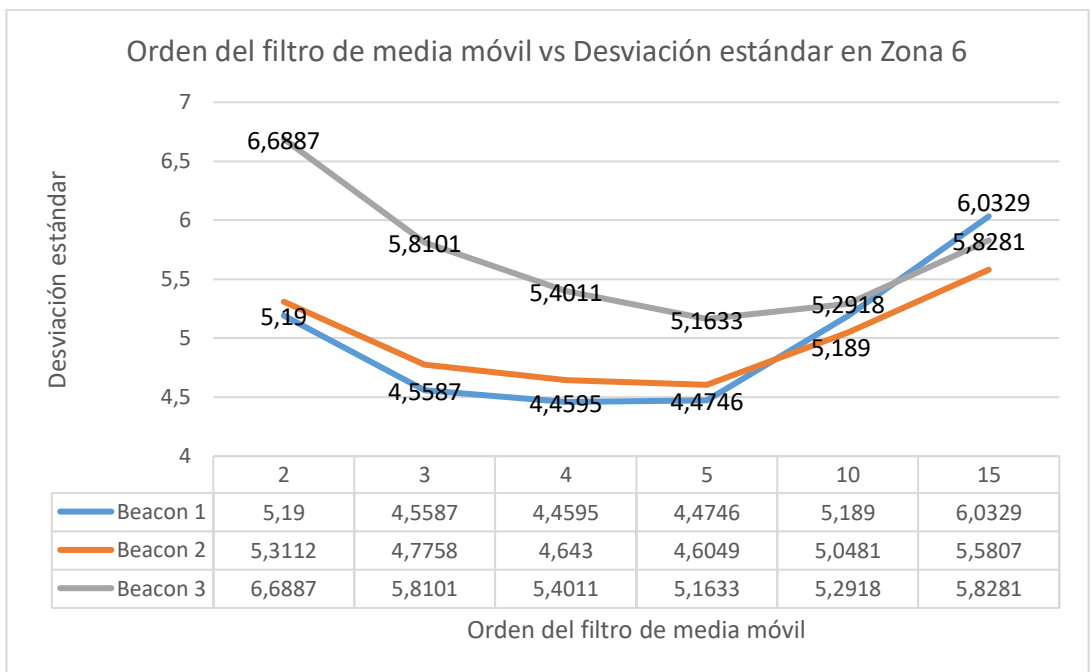


Figura 82. Orden del filtro de media.

Fuente: los autores.

Como se puede observar, el filtro de media móvil ofrece una disminución de la desviación estándar hasta el orden 4, pasado ese orden se puede observar que la desviación estándar nuevamente aumenta.

Posteriormente, se observó el comportamiento del RSSI luego de normalizar los datos. A continuación, se presentan los gráficos del RSSI con el filtro de media móvil de orden 4 y posteriormente, la misma información luego de ser normalizada.

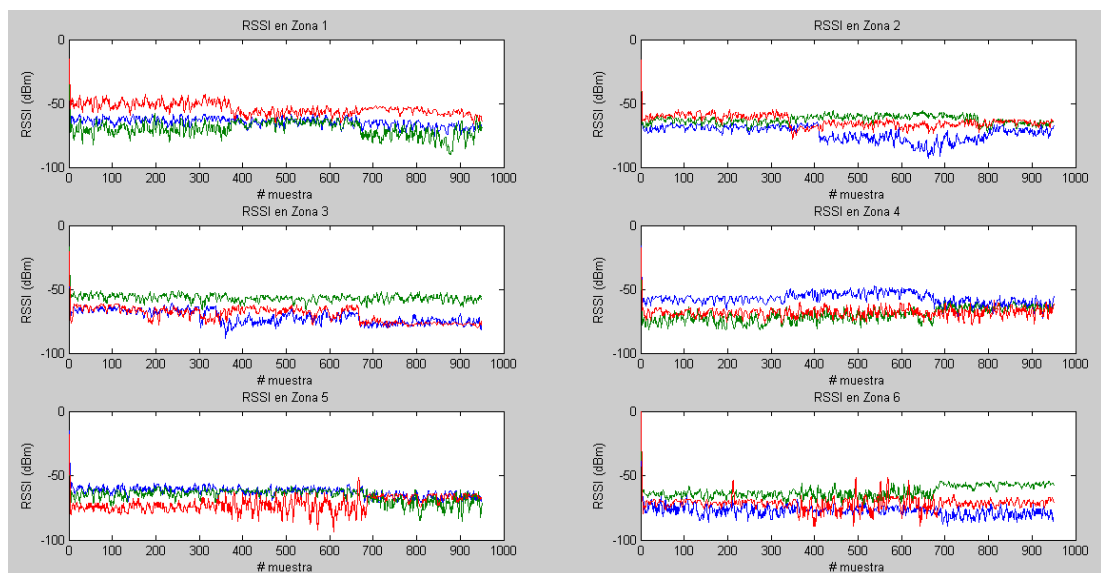


Figura 83. RSSI con filtro de media móvil de orden 4.

Fuente: los autores.

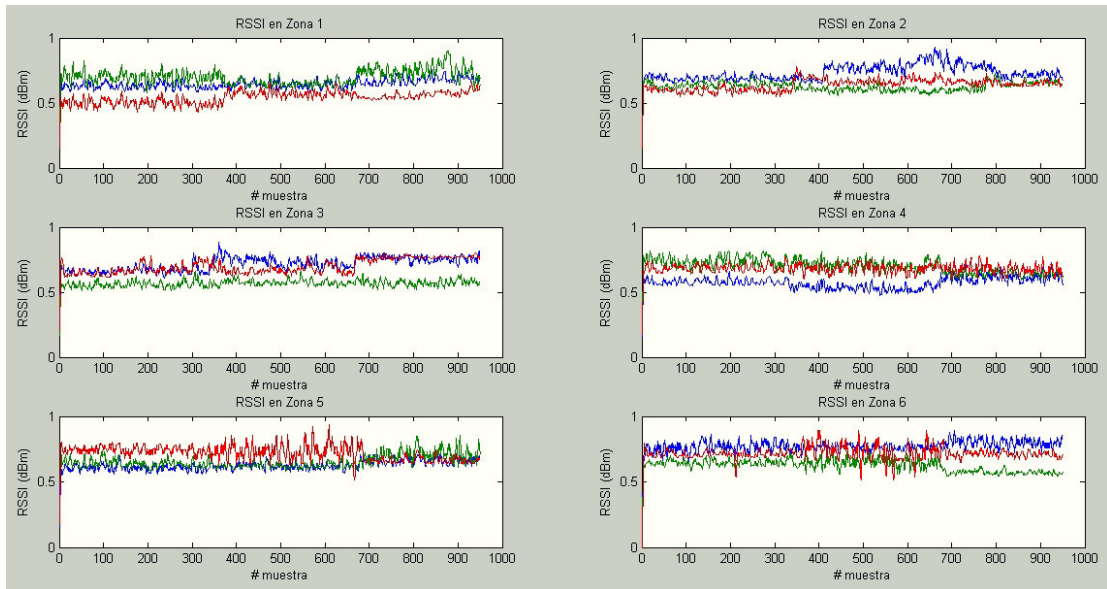


Figura 84. RSSI filtrado y normalizado.

Fuente: los autores.

Posteriormente, se procedió a realizar las pruebas del entrenamiento de la red neuronal, entrenándola con los datos obtenidos anteriormente luego de filtrar y normalizar el RSSI obtenido de los 3 Beacon, asimismo, se utilizaron los siguientes valores para entrenar la red neuronal:

$$\eta = 0.01$$

$$e_{\min} = 10^{-4}$$

A continuación, se detallan los gráficos de disminución de error cuadrático medio y regresión obtenida por la red neuronal.

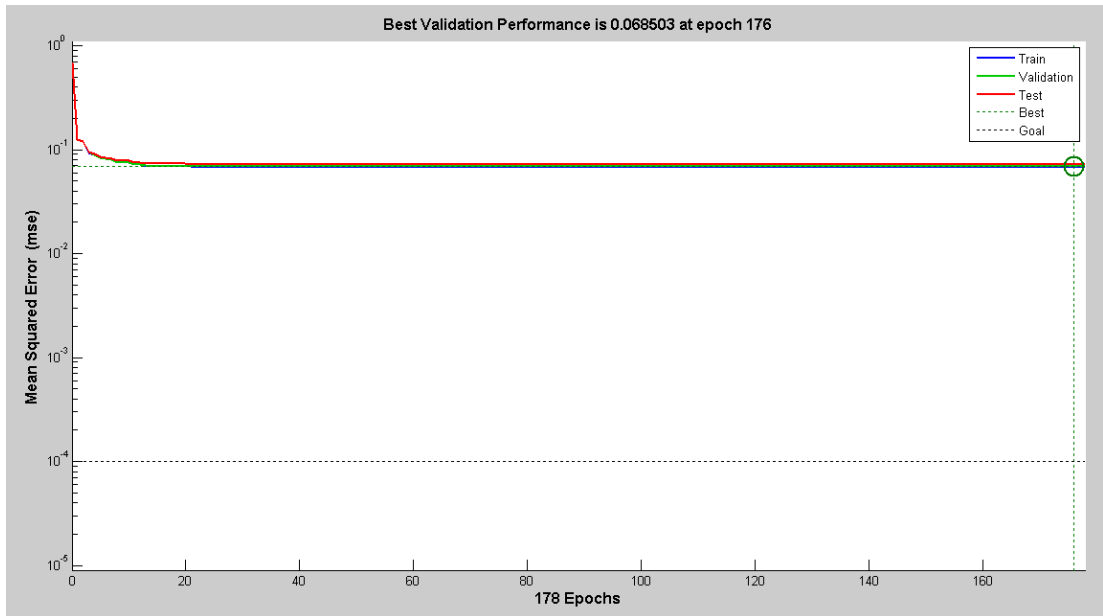


Figura 85. Disminución de error cuadrático medio por época de entrenamiento.

Fuente: los autores.

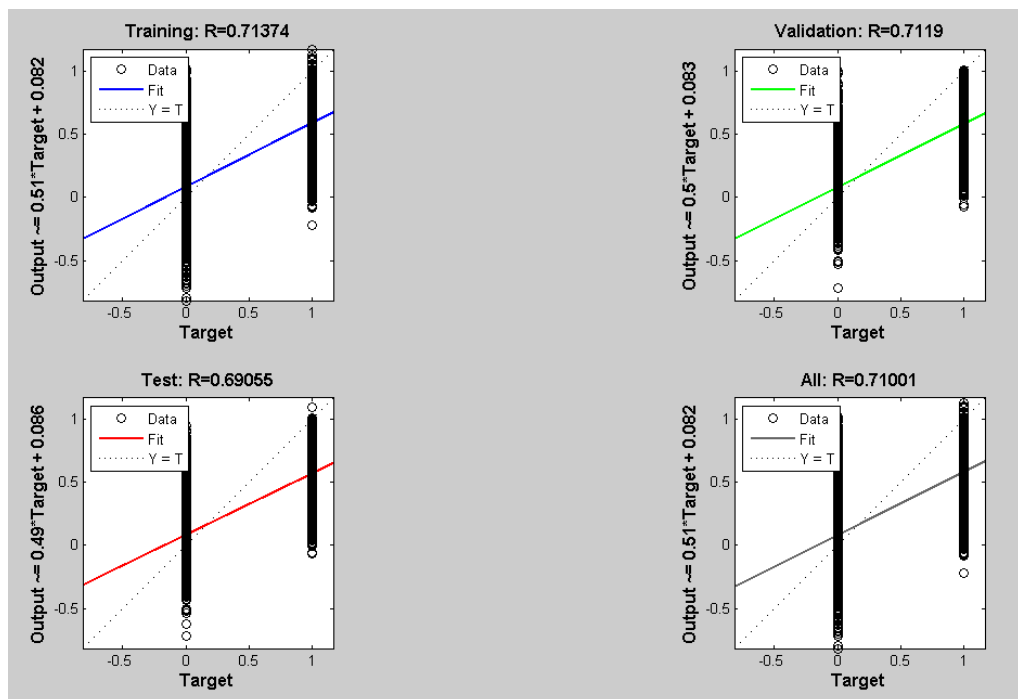


Figura 86. Regresión de la red neuronal.

Fuente: los autores.

Como se puede observar en la figura 85 y 86, la red neuronal no llegó a converger a pesar de que los datos fueron filtrados y normalizados, por lo tanto, se procedió con analizar la razón de la no convergencia de la red neuronal.

El primer paso para realizar esto fue observar la diferenciación de los RSSI por cada zona, en la figura 87 se muestran los RSSI sin filtro graficados en un plano tridimensional, cada eje representa la intensidad de señal recibida de cada uno de los beacon:

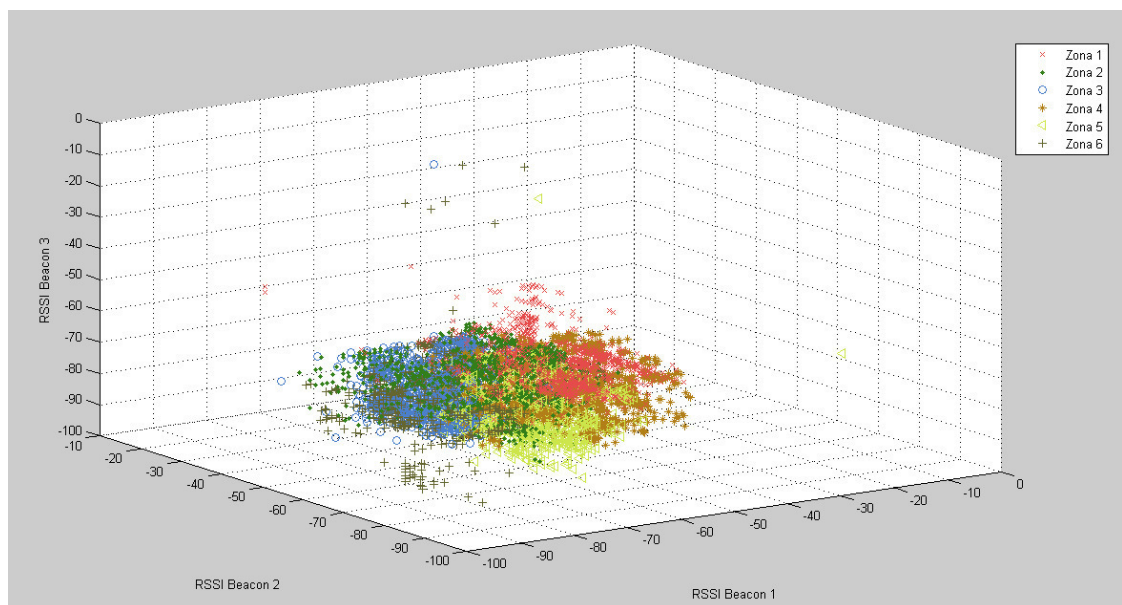


Figura 87. RSSI obtenido por zona con datos sin filtrar.

Fuente: los autores.

En la figura 87, se muestra los RSSI con filtro graficado en un plano tridimensional, cada eje representa la intensidad de señal recibida de cada uno de los beacon:

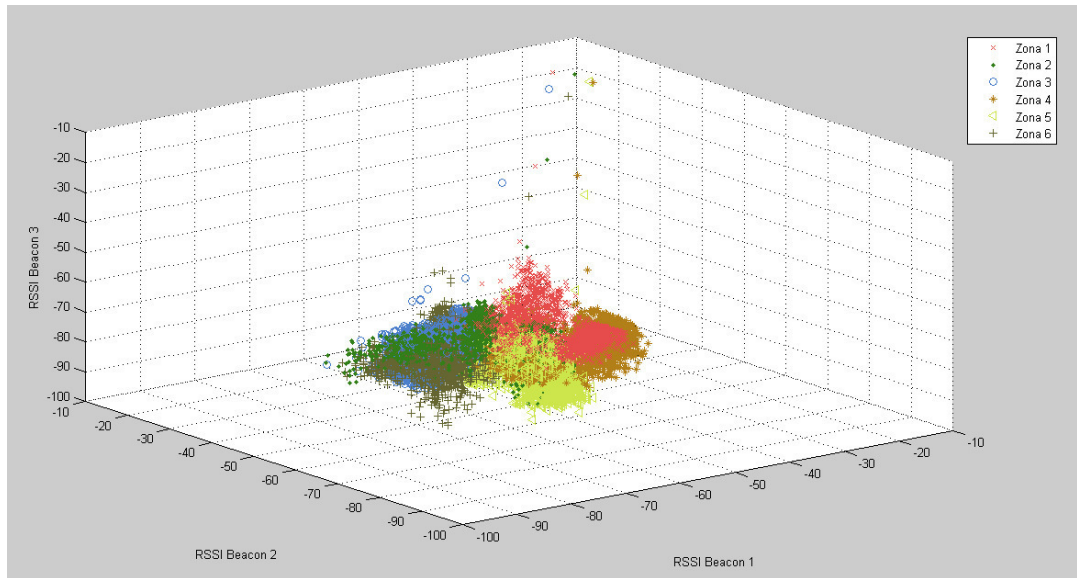


Figura 88. RSSI obtenido por zona con datos filtrados.

Fuente: los autores.

Como se puede observar, los clústeres no se encuentran debidamente limitados, por lo que la red neuronal no podrá clasificarlos al no ser estos diferenciables para cada zona. Para resolver este inconveniente, se analizaron otras distribuciones de beacons que permitan diferenciar con mayor facilidad los clústeres de RSSI obtenidos en cada zona, a continuación, se detallan las distribuciones analizadas, así como los resultados obtenidos en cada uno de ellas:

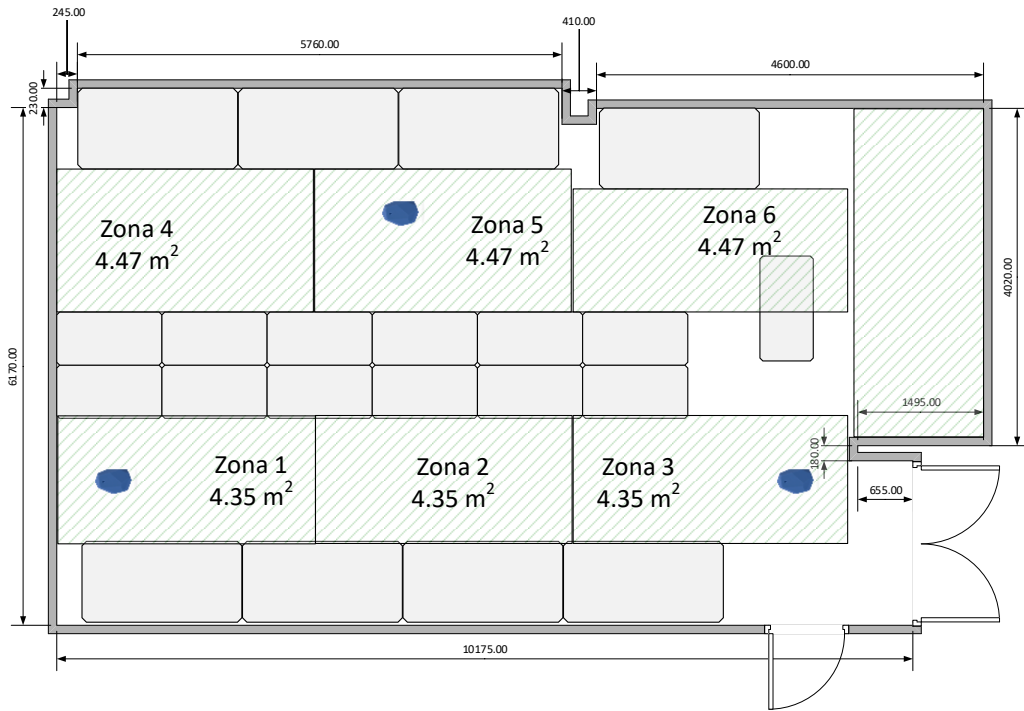


Figura 89. Área de trabajo con 3 beacons, segunda ubicación en el techo.

Fuente: los autores.

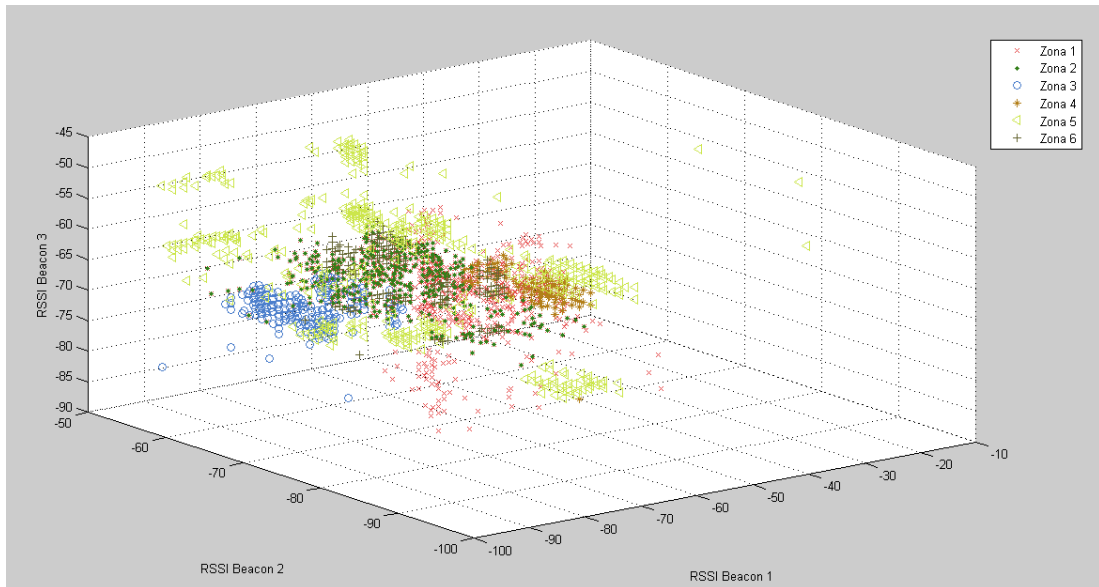


Figura 90. RSSI obtenido por zona con datos sin filtrar.

Fuente: los autores.

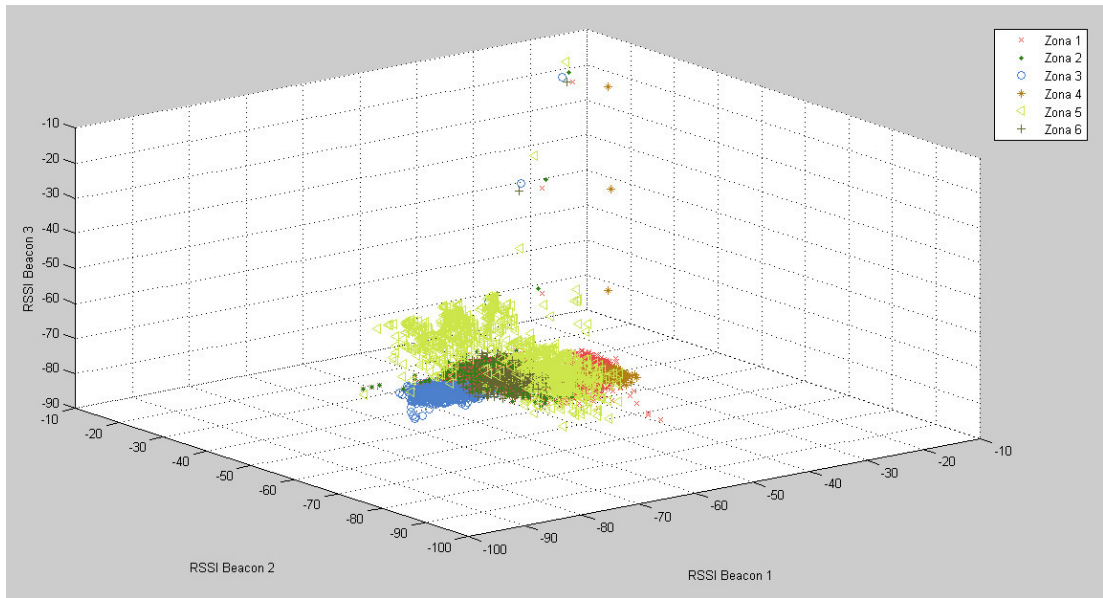


Figura 91. RSSI obtenido por zona con datos filtrados.

Fuente: los autores.

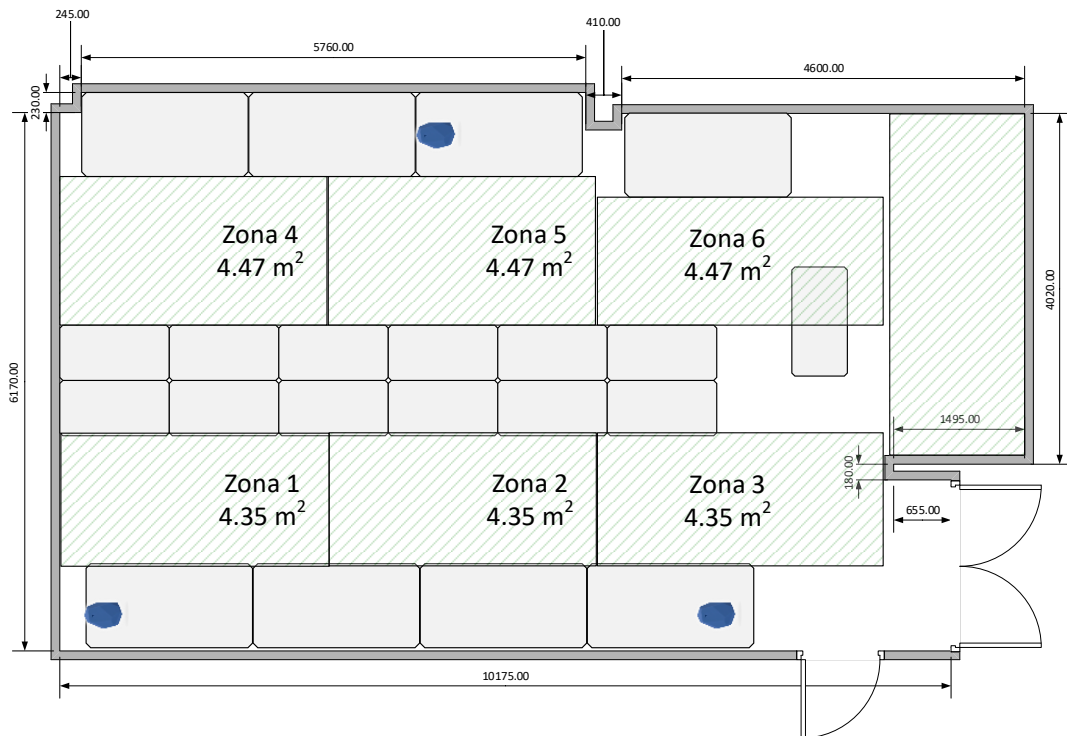


Figura 92. Área de trabajo con 3 beacons ubicados sobre los módulos.

Fuente: los autores.

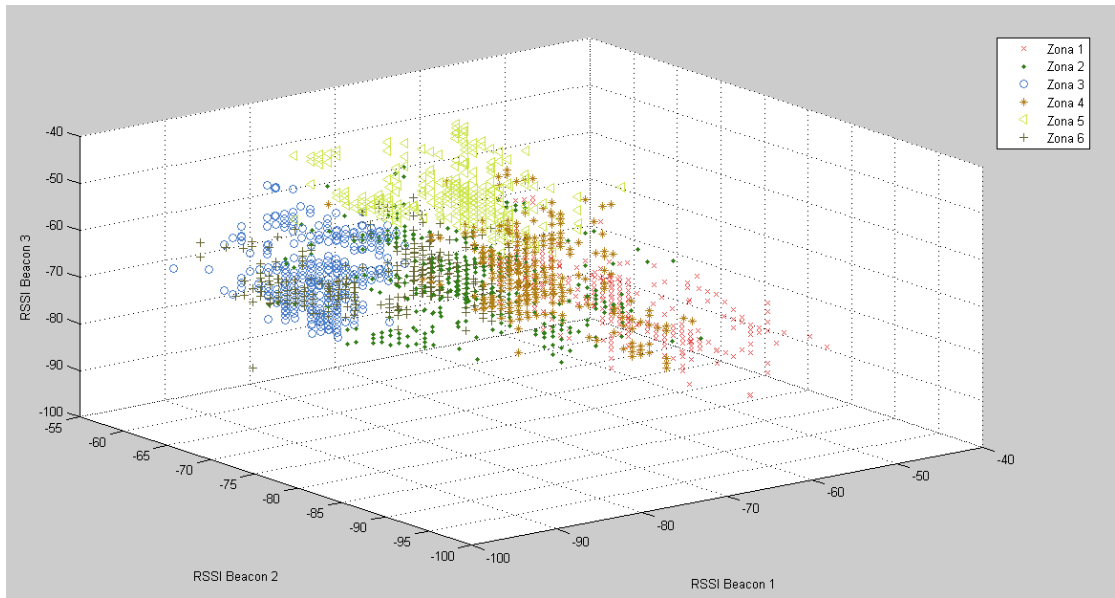


Figura 93. RSSI por zona con datos sin filtrar.

Fuente: los autores.

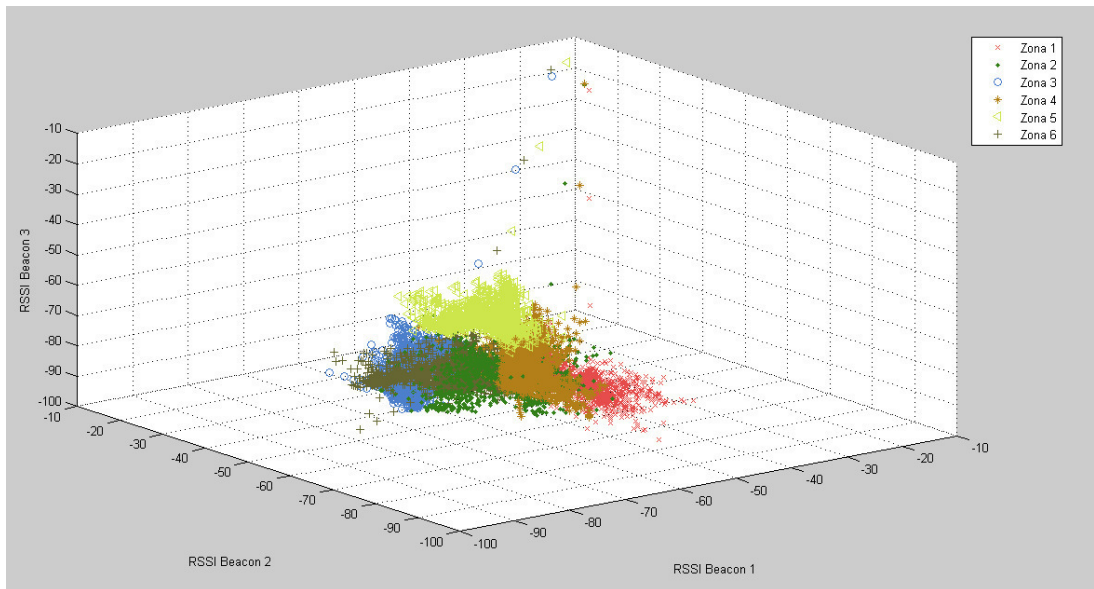


Figura 94. RSSI obtenido por zona con datos filtrados.

Fuente: los autores.

Como se puede observar, en ninguna de las posibles distribuciones de beacons es posible obtener una diferenciación de datos suficiente para la convergencia de la red neuronal, esto puede ser causado por múltiples

factores, como pueden ser la separación física entre cada zona, el tamaño de las zonas y las características específicas del ambiente.

Para realizar las diferentes pruebas de funcionamiento del sistema propuesto, fue necesario realizar recorridos dentro del área establecida, dentro de las zonas delimitadas así como también pasando de una zona a la siguiente, al haberse establecido el error de ubicación en tres (3) metros, se espera que en los extremos de las zonas extremas no se obtengan errores ya que cada zona fue delimitada en tres (3) metros lineales de largo, mientras que en los puntos intermedios de las zonas intermedias, se espera que exista un error de ubicación del 50%, cabe resaltar, que las pruebas fueron realizadas durante tres (3) horas de funcionamiento continuo de la solución propuesta, en donde fueron obtenidos más de 13 mil medidas independientes de RSSI, lo que resulta en 13 mil estimaciones de localización, a continuación se detalla el error obtenido en cada una de las zonas establecidas:

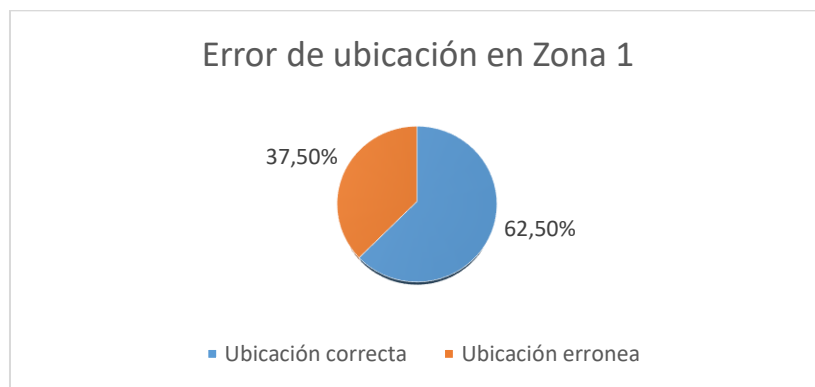


Figura 95. Error de ubicación obtenido en la Zona 1.

Fuente: los autores.

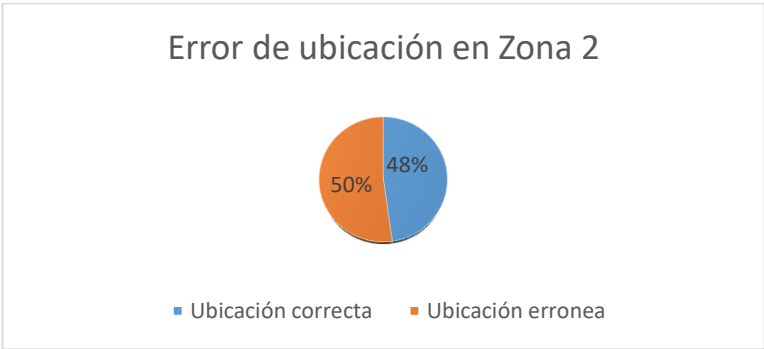


Figura 96. Error de ubicación obtenido en la Zona 2.

Fuente: los autores.

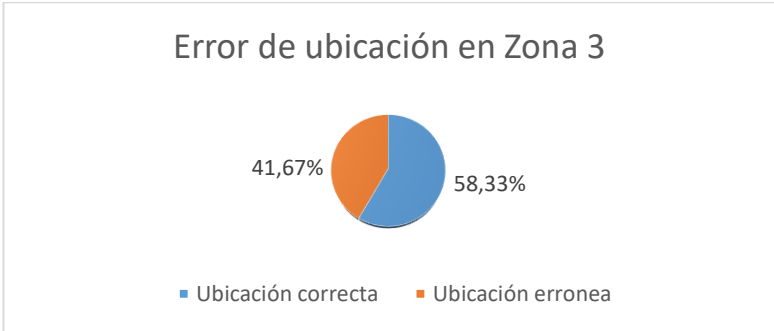


Figura 97. Error de ubicación obtenido en la Zona 3.

Fuente: los autores.

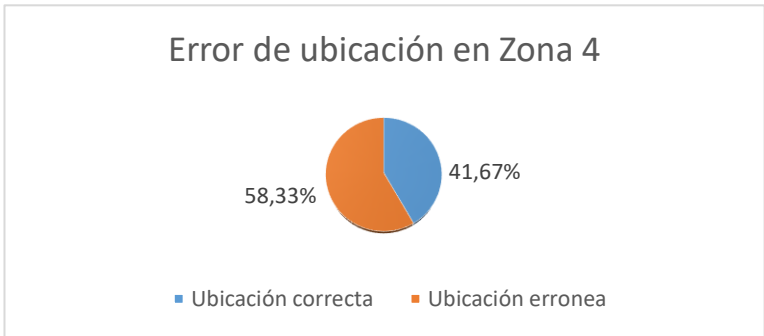


Figura 98. Error de ubicación obtenido en la Zona 4.

Fuente: los autores.

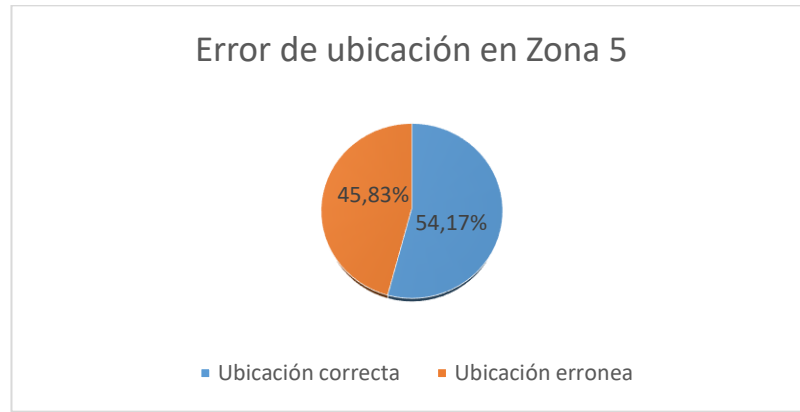


Figura 99. Error de ubicación obtenido en la Zona 5.

Fuente: los autores.

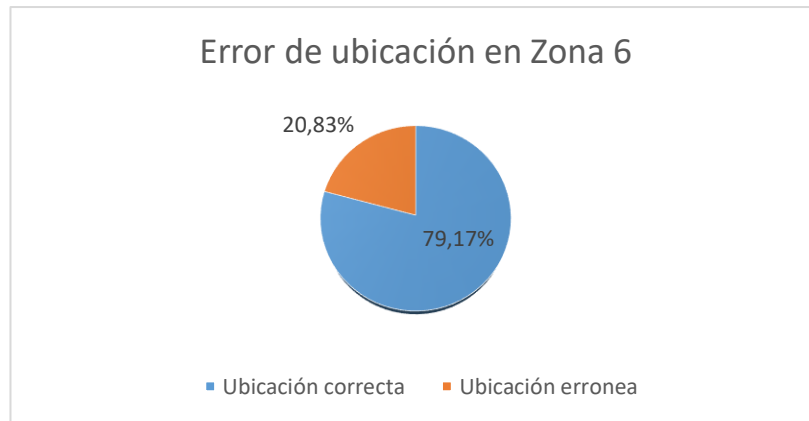


Figura 100. Error de ubicación obtenido en la Zona 6.

Fuente: los autores.

Como se puede observar, la tasa de error obtenida es en todos los casos, alta, debido a la no convergencia de la red neuronal causado por la no diferenciación de los clústeres de RSSI por zona según lo expuesto anteriormente.

CAPÍTULO V

DISCUSIÓN Y APLICACIONES

Como se puede observar, en los resultados obtenidos, en el área estudiada se presenta una gran cantidad de errores, esto debido a que, si bien el espacio presenta una distribución similar a la de un supermercado, el área continúa siendo de un tamaño bastante reducido, lo que no permite establecer un límite significativo entre cada zona.

En uno de los espacios en los que podría ser aplicada la solución sería dentro de un supermercado, en el que los anaqueles con productos tienen un mayor ancho, lo que supondría un mayor espaciamiento entre cada zona contigua, lo que podría resultar en una mayor precisión de ubicación.

Sin embargo, esta solución no solo podría ser implementada en supermercados, sino también en malls, tiendas por departamentos, universidades, etc., en las que sea necesario realizar la ubicación de un dispositivo móvil dentro de un área determinada.

Haciendo contraste con investigaciones pasadas, se puede observar que la solución propuesta no pretende realizar una localización específica del dispositivo móvil dentro del área, sino de clasificar la posible ubicación del móvil para mostrar servicios/promociones dentro del área en que se encuentren. Asimismo, la solución propuesta permite reducir la cantidad de beacon utilizados, ya que basándose en el RSSI de solo 3 beacons, se pudo realizar la clasificación de 6 zonas a más dependiendo del área de aplicación, mientras que el estándar del mercado permite tener solo 1 zona por cada beacon implementado.

CONCLUSIONES

1. Se observa que debido a la distancia de separación entre zonas establecidas no permite obtener una clasificación adecuada de cada una de las zonas, y, por consiguiente, no se logra observar el error esperado.
2. Se logró implementar exitosamente un perceptrón de múltiple capa entrenado por retropropagación, que permitió aumentar el número de zonas disponibles por beacon.
3. Se logró implementar la aplicación de servidor en C# que permite observar, procesar y almacenar las medidas de RSSI obtenidas de la aplicación de cliente.

4. Se logró implementar la aplicación de cliente en Android, que permite obtener las medidas de RSSI y datos del cliente, así como el envío/recepción de datos desde y hacia la aplicación del servidor.

5. Para lograr una medida correcta del RSSI de los beacon, es necesario modificar el tiempo de envío de RSSI de los beacon, ya que un alto tiempo de envío de datos resultara en mediciones de -100, lo que el dispositivo móvil interpretara como beacon fuera de rango.

RECOMENDACIONES

1. El prototipo puede ser mejorado utilizando un modelo diferente de beacon que posea una mayor velocidad de envío de RSSI, ya que el estudiado en esta solución tiene la limitación de poder realizar dicha tarea como mínimo cada 100 ms.
2. Se recomienda implementar protocolos de red empaquetados en la implementación del software tanto de cliente como de servidor, ya que de esta manera se puede ahorrar carga computacional que permitirá utilizar dichos recursos para implementar un mejor preprocesamiento al RSSI medido.
3. El software del cliente puede ser mejorado al implementarse en el software Android Studio, ya que dicho software permite implementar una auto adaptación del software a diferentes versiones del sistema operativo Android, diferentes dimensiones y resoluciones de pantalla y finalmente, poder realizar optimizaciones en la carga computacional requerida por el software.

4. Para eliminar el requisito de una red inalámbrica implementada en el ambiente a ubicar al cliente, se recomienda implementar un servidor web, al que el software implementado en el dispositivo móvil enviara los datos de igual manera que al servidor local.
5. Para mejorar el rango de error en los beacon, se recomienda utilizar técnicas de estimación lineal de datos como lo son los filtros de Kalman u filtro de partículas.

FUENTES DE INFORMACIÓN

Bibliográficas:

- Avila, L., & Reyes, C. (2017). Revisión estado del Arte de la tecnología Bluetooth. *Revista Investigación y Desarrollo en TIC*, 3(2).
- Dong, Q., & Dargie, W. (2012). Evaluation of the reliability of RSSI for indoor localization. En *Wireless Communications in Unusual and Confined Areas (ICWCUCA), 2012 International Conference on* (pp. 1–6). IEEE.
- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Prentice-Hall.
- Guo, T., Qiao, F., Liu, M., Xu, A., Liu, Q., & Zhang, N. (2016). An improved Kalman filtering method for indoor location. En *Computer and*

Communications (ICCC), 2016 2nd IEEE International Conference on
(pp. 1747–1751). IEEE.

Hassan, A. M. A. (2016). Indoor location tracking system using neural network based on bluetooth. En *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on* (pp. 73–78). IEEE.

Kalbandhe, A. A., & Patil, S. C. (2016). Indoor Positioning System using Bluetooth Low Energy. En *Computing, Analytics and Security Trends (CAST), International Conference on* (pp. 451–455). IEEE.

Khan, M., Kai, Y. D., & Gul, H. U. (2017). Indoor Wi-Fi positioning algorithm based on combination of Location Fingerprint and Unscented Kalman Filter. En *Applied Sciences and Technology (IBCAST), 2017 14th International Bhurban Conference on* (pp. 693–698). IEEE.

Martín Talavera, J. M. (2012). *Posicionamiento por radiofrecuencia basado en robots colaborativos* (B.S. thesis). Autonoma de Madrid.

Muller, N. J. (2001). *Bluetooth Demystified*. McGraw-Hill.

Razavi, A., Valkama, M., & Lohan, E.-S. (2015). K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization. En *Globecom Workshops (GC Wkshps), 2015 IEEE* (pp. 1–7). IEEE.

Smith, S. W. (1999). *The Scientist and Engineer's Guide to Digital Signal Processing* (2.^a ed.). San Diego, California: California Technical Publishing.

Electrónicas:

Apple Developer. (2017). iBeacon. Recuperado 18 de octubre de 2017, a partir de <https://developer.apple.com/ibeacon/>

Bluetooth SIG. (2017). Health Device Profile. Recuperado 19 de octubre de 2017, a partir de <https://www.bluetooth.com/specifications/assigned-numbers/health-device-profile>

Bluetooth Technology Website. (2017). How it works. Recuperado 20 de octubre de 2017, a partir de <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>

Estimote. (2017a). How precise are Estimote Beacons? Recuperado 28 de octubre de 2017, a partir de <http://community.estimote.com/hc/en-us/articles/201302836-How-precise-are-Estimote-Beacons->

Estimote, I. (2017b). Beacons. Recuperado 18 de octubre de 2017, a partir de <https://community.estimote.com/hc/en-us/sections/200199047-Beacons>

Estimote, I. (2017c). Technical specification of Estimote Beacons and Stickers. Recuperado 26 de octubre de 2017, a partir de <http://community.estimote.com/hc/en-us/articles/204092986-Technical-specification-of-Estimote-Beacons-and-Stickers>

- Experian Data Quality. (2013, diciembre). The state of data quality. Recuperado 18 de octubre de 2017, a partir de <http://www.experian.es/assets/consultoria/white-papers/the%20state%20of%20data%20quality.pdf>
- Google Developers. (2017, febrero). Beacons. Recuperado 18 de octubre de 2017, a partir de <https://developers.google.com/beacons/overview>
- Hewlett Packard Enterprise Development LP. (2015). Aruba Beacons Validated Reference Guide. Recuperado a partir de <https://community.arubanetworks.com/aruba/attachments/aruba/Aruba-VRDs/69/2/Aruba%20Beacons%20Validated%20Reference%20Guide.pdf>
- Huawei Technologies Co, L. (2017). Huawei P9 lite, la revolución en el diseño. Recuperado 26 de octubre de 2017, a partir de <http://www.huaweispain.com/smartphones/huawei-p9-lite/>
- Kevin, A. (2016). Diseño de un sistema de clasificación de señales de tránsito vehicular utilizando redes neuronales convolucionales. *Universidad San Ignacio de Loyola*. Recuperado a partir de <http://repositorio.usil.edu.pe/handle/USIL/2480>
- Khan, A., & Davis, M. (2016, junio). Hidden Secrets of Beacon Data. Recuperado 18 de octubre de 2017, a partir de <http://www.mmaglobal.com/research/hidden-secrets-beacon-data>

- Kriz, P., Maly, F., & Kozel, T. (2016). Improving Indoor Localization Using Bluetooth Low Energy Beacons. *Mobile Information Systems, 2016*, 1-11. <https://doi.org/10.1155/2016/2083094>
- Magitech. (2017). Laptop Lenovo IdeaPad Z480. Recuperado 26 de octubre de 2017, a partir de <https://www.magitech.pe/laptop-lenovo-ideapad-z480-59347661-intel-core-i5-3210m-2-5ghz.html>
- Massachusetts Institute of Technology. (2017). System Requirements | Explore MIT App Inventor. Recuperado 26 de octubre de 2017, a partir de <http://appinventor.mit.edu/explore/content/system-requirements.html>
- Microsoft. (2017). Visual Studio 2017 System Requirements. Recuperado 26 de octubre de 2017, a partir de <https://www.visualstudio.com/en-us/productinfo/vs2017-system-requirements-vs>
- MTC. (2017). Ministerio de Transportes y Comunicaciones. Recuperado 19 de noviembre de 2017, a partir de http://www.mtc.gob.pe/normas_legales/normas_legales.html
- Muhammad, J. (2007, febrero). *Artificial neural networks for location estimation and co-cannel interference suppression in cellular networks*. University of Stirling, Stirling. Recuperado a partir de <http://dspace.stir.ac.uk/handle/1893/221>
- OnePlus. (2017). OnePlus 3T. Recuperado 26 de octubre de 2017, a partir de <https://oneplus.net/3t/specs>

The MathWorks, I. (2017). System Requirements. Recuperado 26 de octubre de 2017, a partir de https://es.mathworks.com/support/sysreq/previous_releases.html

Wang, Y., Ye, Q., Cheng, J., & Wang, L. (2015). RSSI-Based Bluetooth Indoor Localization (pp. 165-171). IEEE. <https://doi.org/10.1109/MSN.2015.14>

Zafari, F., & Papapanagiotou, I. (2015). Enhancing iBeacon based micro-location with particle filtering. En *Global Communications Conference (GLOBECOM), 2015 IEEE* (pp. 1–7). IEEE. Recuperado a partir de <http://ieeexplore.ieee.org/abstract/document/7417504/>