

Estimación de la precisión de dispositivos ZigBee para el posicionamiento en una cancha
de microfútbol

Yinner Jaider López Puerres

Edwin Andrés Urbina De La Cruz

Universidad CESMAG

Facultad de Ingeniería

Programa de Ingeniería Electrónica

San Juan de Pasto

2024

Estimación de la Precisión de Dispositivos ZigBee Para el Posicionamiento en una
Cancha de Microfútbol

Yinner Jaider López Puerres

Edwin Andrés Urbina De La Cruz

Asesor

Mag. Mario Henao

Universidad CESMAG

Facultad de Ingeniería

Programa de Ingeniería Electrónica

San Juan de Pasto

2024

Nota de aceptación

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad CESMAG para optar por el título de Ingeniero Electrónico.

Mg. Mario Fernando Henao Rosero

Asesor

Jurado

Jurado

San Juan de Pasto, mayo 2024

Página de nota de exclusión de responsabilidad intelectual.

“El pensamiento que se expresa en esta obra es exclusiva responsabilidad del autor y no compromete la ideología de la Universidad CESMAG”.

Dedicatoria

A mis padres, cuya incondicionalidad y apoyo constante fueron la fuerza motriz que me impulsó a lo largo de mi carrera universitaria. A mi madre y a mi padre, les debo cada logro y cada paso dado en este camino. Su amor y sacrificio son el fundamento de este logro. A mi hermano, cuyo carisma y alegría siempre lograron arrancarme una sonrisa incluso en los momentos más difíciles. A mi abuela, que, aunque ya no está con nosotros, su amor y sus consejos siempre estarán presentes en mi corazón. Cumplí con lo que un día me aconsejaste y este trabajo de grado es en gran parte gracias a ti. A mi novia, mi compañera y confidente, que, con su paciencia y sabiduría, siempre estuvo a mi lado, escuchándome y aconsejándome. Nunca dejaste de recordarme lo capaz que soy y eso me dio la fuerza para seguir adelante.

Edwin Andrés Urbina De la Cruz.

Dedico este trabajo principalmente a Dios, por concederme la vida, la salud y la sabiduría necesarias para alcanzar este momento tan significativo en mi formación profesional. A mis padres, cuya guía y amor incondicional me han permitido llegar hasta aquí. Su apoyo constante y sacrificio han sido la base de todos mis logros y a mis hermanas, quienes siempre me han inspirado a dar lo mejor de mí. Su compañía y ánimo han sido esenciales para culminar esta etapa.

Yinner Jaider López Puerres

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a los directivos de nuestra universidad por ser siempre amables y atentos. Su disposición y apoyo han sido fundamentales en nuestro camino académico. Extendemos este agradecimiento a todas las personas que forman parte de la universidad, ya que cada uno, de alguna manera, ha contribuido a nuestra formación.

Deseamos agradecer de manera especial a los docentes del área de Electrónica, quienes siempre estuvieron dispuestos a responder nuestras preguntas y se esforzaron al máximo para que comprendiéramos y aprendiéramos cada tema que se planteaban compartírnos. Su dedicación y paciencia fueron cruciales para nuestro desarrollo académico.

Un agradecimiento muy especial al Mg. Mario Fernando Henao Rosero, nuestro asesor. Su guía fue invaluable durante la realización de nuestro trabajo de grado. Agradecemos su constante atención, paciencia, la resolución de cada duda y conocimientos compartidos. Gracias por el tiempo dedicado y por trabajar en equipo para lograr la culminación de este trabajo.

Finalmente, queremos agradecer a nuestros compañeros que nos acompañaron durante la carrera y que hicieron de este un ambiente más alegre y propicio para el aprendizaje. Su compañía y apoyo hicieron de esta experiencia algo memorable y enriquecedor.

Tabla de contenido

Tabla de contenido	7
1. El Problema de Investigación	20
<i>1.1</i> Objeto de investigación.....	20
<i>1.2</i> Línea de Investigación	20
<i>1.3</i> Sub Línea de Investigación	20
<i>1.4</i> Planteamiento o Descripción del Problema	20
<i>1.5</i> Formulación del Problema	21
<i>1.6</i> Objetivos	21
1.6.1 Objetivo General	21
1.6.2 Objetivos Específicos	22
<i>1.7</i> Justificación.....	22
<i>1.8</i> Viabilidad	23
<i>1.9</i> Delimitación	23
2. Tópicos del Marco Teórico.....	24
<i>2.1</i> Antecedentes	24
2.1.1 Diseño e Implementación de un Sistema de Monitorización de Práctica Deportiva de Baloncesto en Entornos Contextuales WPAN/WBAN.....	24

2.1.2 Sistema Prototipo de Posicionamiento Local Destinado a la Búsqueda de Libros Utilizando Tecnología ZigBee	27
2.1.3 Diseño e Implementación de un Prototipo Para la Localización de un Objeto en Movimiento Usando ZigBee	30
2.1.4 Protocolo ZigBee (IEEE 802.15.4)	33
2.1.5 Cálculo de distancia entre los nodos de una red inalámbrica ZigBee en función del parámetro RSSI.....	35
2.2 Enunciados de los Supuestos Teóricos	38
2.2.1 Protocolo ZigBee.....	38
2.2.2 Pérdidas de propagación.....	39
2.2.3 Geometría basada en la localización	40
2.2.4 Método de posición basados en medición de RSSI.....	42
2.2.5 Machine Learning.....	42
2.3 Variables de estudio	44
2.3.1 Definición nominal de variables.....	44
2.3.2 Definición operativa de variables	45
2.4 Hipótesis.....	47
2.4.1 Hipótesis de Investigación.....	47
2.4.2 Hipótesis nula	47
2.4.3 Hipótesis alternativa	47

3.	Metodología	47
3.1	Enfoque	47
3.2	Paradigma.....	48
3.3	Método	48
3.4	Tipo de diseño	48
3.5	Diseño de investigación	48
3.6	Universo	49
3.7	Muestra.....	49
3.8	Técnicas de recolección de información	49
3.8.1	Validez de la técnica.....	51
3.8.2	Confiabilidad de la técnica	51
3.9	Instrumentos de recolección de información	53
4.	Resultados de la investigación	55
4.1	Diseño del sistema de posicionamiento en una cancha de microfútbol basado en ZigBee	55
4.2	Montaje electrónico y algoritmo para la toma de datos dentro de la Cancha #1 de la Universidad CESMAG	58
4.2.1	Código Arduino para consulta y almacenamiento de parámetro RSSI dentro de la cancha.	59
4.2.2	Adquisición de datos en la Cancha #1 de la Universidad CESMAG.....	62

	10
4.3 Ajuste del modelo de propagación Log Distancia.	63
4.3.1 Parámetros generales para el ajuste de Modelos de Propagación en la Cancha #1 de la Universidad CESMAG.....	65
4.3.2 Ajuste Modelo de Propagación en 1/4 de la Cancha #1 de la Universidad CESMAG.	79
4.3.3 Ajuste Modelo de Propagación en un espacio 5 x 5 mts en la Cancha #1 de la Universidad CESMAG.	84
4.3.4 Trilateración y obtención de las coordenadas con el ajuste del modelo Log Distancia en un espacio de 5 x 5 mts en el espacio deportivo.	91
4.4 Ajuste del modelo de Machine Learning en un espacio de 5 x 5 mts de la Cancha #1 de la Universidad CESMAG.....	96
4.5 Comparación entre el Modelo de Propagación ajustado y el Modelo de Machine Learning	103
4.6 Análisis de los resultados	105
4.7 Implementación de los sistemas de posicionamiento dentro de la Cancha #1 de la Universidad CESMAG.	111
5. Conclusiones y Trabajos futuros.....	120
6. Referencias.....	125
7. Anexos	128
7.1 Especificaciones y configuración de la red.	128
7.1.1 Creación y prueba de las tramas con el Software XCTU	133

7.2	Diseño y alimentación energética para los Dispositivos finales.	137
7.2.1	Implementación de dispositivos y componentes para el Xbee coordinador. 138	
7.3	Código Arduino para la recolección de datos dentro de la Cancha #1 de la Universidad CESMAG	140
7.4	Códigos Matlab para el procesamiento de datos, entrenamiento de modelo ajustado y la trilateración.	146
7.5	Código de Python para la obtención de datos por puerto serial y predicción de las coordenadas	148
7.6	Entrenamiento y datos para el modelo de Machine Learning	149

Lista de Tablas

Tabla 1. Parámetros de Entrada	28
Tabla 2. Eficiencia de las pruebas.....	29
Tabla 3.Coordenadas reales vs. Coordenadas calculadas	32
Tabla 4. RSSI medido promedio, distancia teórica y errores calculados a partir de las mediciones realizadas.	36
Tabla 5. Exponentes de pérdida de trayecto para diferentes entornos	40
Tabla 6. Tabla para la recolección de datos	54
Tabla 7. Trama enviada, utilizada para obtener el RSSI.....	56
Tabla 8. Trama de respuesta esperada.....	57
Tabla 9. Representación de bits y colores de posibles errores.....	58
Tabla 10. Ajuste en el cálculo de las coordenadas, considerando los XBee referencia....	93
Tabla 11. Entrenamientos realizados para obtener las mejores predicciones en el algoritmo de Machine Learning.....	99
Tabla 12. Predicciones de los Modelos Implementados en las coordenadas 5x5 metros	108
Tabla 13. Porcentajes de predicción de los modelos según las coordenadas.....	110
Tabla 14. Especificaciones del equipo usado para la implementación del modelo de Machine Learning	115
Tabla 15. Descripción de topologías de red soportadas por XBee-PRO S1	128
Tabla 16. Pines de Conexión XBee-PRO S1	129
Tabla 17. Configuraciones Comunicación serial XCTU con XBee-PRO S1	131
Tabla 18. Configuración de red Coordinador	131

Tabla 19. Configuración de red Dispositivos finales.....	132
Tabla 20. Nivel de transmisión de dispositivos XBee Pro S1	133

Lista de Figuras

Figura 1. Esquema gráfico del escenario	25
Figura 2. Mapa de cobertura y potencia en dBm	25
Figura 3. Power Delay profile, Receptor Canasta Derecha	26
Figura 4. Ejemplo de red ZigBee usando topología de red Mesh.....	27
Figura 5. Ejemplo de Estimación de la Posición.....	28
Figura 6. Comparación potencia vs distancia y potencia recibida como RSSI.....	30
Figura 7. Ejemplo de potencia recibida en un nodo.....	30
Figura 8. Comparación valor RSSI teórico y práctico	31
Figura 9. Tecnologías en 2.4 GHz	33
Figura 10. Topologías de red	33
Figura 11. Ejemplo de red ZigBee	34
Figura 12. Triangulación basada en AOA.....	41
Figura 13. Red Neuronal.....	43
Figura 14. Ejemplo montaje dentro del campo de microfútbol	46
Figura 15. Ejemplo medición en la Cancha #1 de la Universidad CESMAG	50
Figura 16. Cancha de microfútbol con ejes X, Y; Z	52
Figura 17. Ejemplo coordenadas en la Cancha #1 de la Universidad CESMAG	54
Figura 18. Esquema de conexión microcontrolador Arduino y XBee-PRO S1.....	59
Figura 19. Diagrama de flujo código Arduino y Coordinador XBee.....	60
Figura 20. Representación de la secuencia de mediciones entre los dispositivos finales y el coordinador.	61
Figura 21. Divisiones de la cancha para la toma de datos en los ejes (X, Y)	63
Figura 22. Línea de tiempo de los estudios realizados a las diferentes potencias	64

Figura 23. Ubicación de todos los dispositivos en la Cancha #1 de la Universidad CESMAG.....	65
Figura 24. Escenario de la Cancha #1 de la Universidad CESMAG.....	69
Figura 25. Potencia recibida en la Cancha #1 Unicesmag con potencia transmitida igual a 14 dBm.....	70
Figura 26. Ajuste de los modelos de una sola pendiente para cada módulo XBee.....	71
Figura 27. Comparación de error del ajuste del modelo en la aproximación de la distancia en cada dispositivo a 14dBm	72
Figura 28. Histograma de potencias para un punto de la Cancha #1 de la Universidad CESMAG.....	74
Figura 29. Histograma de potencias no adecuadas para un punto en la Cancha #1 Unicesmag.....	75
Figura 30. Mapas de calor RSSI filtrados para los XBee a 18dBm.....	76
Figura 31. Intersección de los modelos para el XBee1 con datos filtrados a 18dBm.....	77
Figura 32. Comparación de error de los dispositivos en toda la Cancha #1 a 18dBm.	78
Figura 33. Representación del error del XBee1 en toda la Cancha #1 de la universidad CESMAG.....	79
Figura 34. Distribución de los dispositivos en ¼ de la Cancha #1 de la Universidad CESMAG.....	80
Figura 35. Mapa de calor de los módulos XBee en ¼ de Cancha #1 Unicesmag, con datos filtrados (18dBm).....	81
Figura 36. Representación de los ajustes del modelo para los dispositivos XBee.....	82

Figura 37. Comparación de error de los dispositivos en ¼ de Cancha #1 Unicesmag, a 18dBm.....	83
Figura 38. Mapa de calor del error del XBee1 en ¼ de cancha (18dBm).....	84
Figura 39. Ubicación de los Dispositivos XBee en un espacio de 5 x 5 mts en la Cancha #1 de la Universidad CESMAG.....	85
Figura 40. Escenario de 5x5 metros de la Cancha #1 de la Universidad CESMAG	86
Figura 41. Distribución de los datos obtenidos en coordenadas aleatorias.....	87
Figura 42. Representación de los valores RSSI en dispositivos XBee usando el promedio \pm desviación estándar.....	88
Figura 43. Ajuste de los modelos de doble pendiente para los dispositivos XBee.....	89
Figura 44. Comparación del error de los dispositivos con el ajuste del modelo, en un espacio de 5x5 mts en la Cancha #1 de la Universidad CESMAG.	90
Figura 45. Combinaciones para la trilateración utilizadas en los dispositivos XBee.	92
Figura 46. Comparación de error de combinación de Trilateración de los dispositivos... ..	95
Figura 47. Análisis del error en las predicciones con XBee 4 como referencia y el modelo de propagación ajustado.....	96
Figura 48. Modelo de red neuronal para la predicción de coordenadas X e Y.	
Figura 49. Comportamiento del error en las predicciones del modelo de Machine Learning.....	101
Figura 50. Análisis del error cuadrático para las coordenadas (X, Y), con el modelo de Machine Learning.	102
Figura 51. Comparación del error en la predicción de la posición para el Modelo Ajustado y el Modelo de Machine Learning.....	103

Figura 52. Comparación del error en los tres estudios para la predicción de las coordenadas (X, Y).	106
Figura 53. Funcionamiento de los sistemas de posicionamiento implementados en el espacio deportivo	112
Figura 54. Diagrama de Flujo para el programa Arduino para la predicción de coordenadas.....	114
Figura 55. Funcionamiento de la implementación del modelo de Machine Learning para la obtención de coordenadas usando comunicación serial.....	116
Figura 56. Diagrama de flujo programa en Python para la predicción de coordenadas usando modelo entrenado de Machine Learning.	117
Figura 57. Tiempos de ejecución en la predicción de coordenadas para el modelo de propagación ajustado.	118
Figura 58. Tiempos de ejecución en la predicción de coordenadas para el modelo de Machine Learning.	119
Figura 59. Vista superior XBee-PRO S1	128
Figura 60. Especificaciones de alimentación XBee-PRO S1	129
Figura 61. Interfaz de creación de trama consola XCTU.....	133
Figura 62. End Device y fuente de alimentación aislada	137
Figura 63. Diseño de PCB.....	138
Figura 64. Implementación de los dispositivos y componentes en la placa PCB diseñada (lado superior).....	139
Figura 65. Lado inferior placa PCB.	139

Introducción

Las tecnologías inalámbricas forman parte integral del día a día y, con su avance, se evidencian las ventajas de realizar interconexiones entre dispositivos para facilitar tareas y satisfacer necesidades humanas. Gracias a los microcontroladores, se han desarrollado dispositivos más pequeños y portátiles, que se han acoplado a los protocolos de comunicación actuales, manejando frecuencias similares, con costos bajos y pérdidas aceptables. El uso de la tecnología inalámbrica se ha vuelto indispensable en la vida cotidiana, ya sea para hacer una llamada, geolocalizar mediante GPS o navegar por internet.

En la actualidad, las aplicaciones de la tecnología inalámbrica están destinadas a hacer la vida más fácil y cómoda, ya sea brindando seguridad o permitiendo la interconexión a kilómetros de distancia. En ámbitos del diario vivir, como el deporte, esta tecnología se ha venido adaptando para proporcionar datos precisos que sirven como método de ayuda para jugadores y entrenadores (Oliveira Rodríguez & García Fernández, 2021). En cuanto a la monitorización de sus movimientos se puede ver reflejado en el fútbol con tecnologías avanzadas capaces de evaluar movimientos en tiempo real.

En esta investigación, se propuso la construcción y evaluación de un sistema de posicionamiento para un jugador utilizando la tecnología ZigBee, un protocolo de comunicación que permite el acceso a parámetros involucrados en la comunicación, tales como la potencia recibida. Con estas características, se evaluó la precisión de los dispositivos para determinar la posición en la Cancha #1 de la Universidad CESMAG. Este sistema utilizó una interconexión por nodos mediante una topología de red en estrella para realizar mediciones de RSSI entre los nodos y almacenarlas con la ayuda de un microcontrolador para su posterior procesamiento.

El ajuste de los modelos de propagación y de aprendizaje automático (Machine Learning) se llevó a cabo a partir de las mediciones de RSSI entre los dispositivos. Cada ajuste proporcionó información valiosa para tomar decisiones respecto a ajustes posteriores, permitiendo así alcanzar la mejor aproximación de cada modelo. Dado que el trabajo de grado se enfoca exclusivamente en determinar la precisión de los dispositivos XBee S1 Pro para medir la posición de un jugador en la Cancha #1 de la Universidad CESMAG, al final, se compararon los errores de cada modelo para determinar su precisión en la localización de coordenadas.

En el documento, se puede encontrar la descripción del problema de la investigación en la sección 1.4. Además, los objetivos e hipótesis de la investigación se encuentran en la sección 1.6 y 2.4 respectivamente, los procedimientos y hallazgos obtenidos se presentan en la sección 4, donde se describe paso a paso el análisis realizado. Los detalles específicos de dicho análisis se encuentran en la sección 4.6.

Para configuraciones adicionales realizadas durante la implementación de los sistemas de posicionamiento y los códigos implementados en el microcontrolador para la obtención de los datos, se puede consultar la sección 6.

1. El Problema de Investigación

1.1 Objeto de investigación

Precisión de dispositivos ZigBee para posicionamiento en una cancha de microfútbol.

1.2 Línea de Investigación

Comunicaciones Electrónicas: El área de Comunicaciones Electrónicas de la Institución Universitaria CESMAG centra su trabajo investigativo en el estudio de las tecnologías, métodos, técnicas, algoritmos y desarrollos utilizados en los procesos de transmisión, recepción y procesamiento de la información de manera local y remota mediante el uso de circuitos electrónicos. Los trabajos desarrollados en esta área incluyen el diseño, modelado, simulación, análisis de sistemas análogos y digitales para telecomunicaciones, telemetría y redes de datos (Programa de Ingeniería Electrónica, 2015).

1.3 Sub Línea de Investigación

Telemetría: La telemetría es una rama de la ingeniería dedicada a medir cantidades físicas mediante interfaces electrónicas y transmitir esta información a un centro de gestión a través de diversos medios. Esto permite monitorear y controlar sistemas electrónicos, donde los datos transmitidos se analizan para mejorar continuamente. Para ello se integran la electrónica, la informática y las telecomunicaciones (Valencia & Giraldo, 2010).

1.4 Planteamiento o Descripción del Problema

Hoy en día el uso de la tecnología se ha convertido en una práctica necesaria dentro del ámbito deportivo ya que se puede analizar y monitorear la actividad física, el rendimiento, la

distancia recorrida, velocidad, aceleración y ubicación dentro del campo o el estado de jugadores entre otros factores, gracias a todo tipo de sensores y cámaras. Estas dan información en tiempo real de la posición, velocidad o frecuencia cardíaca, información que puede ser tratada para analizar tendencias o posibles mejoras en el juego. Sin embargo, para implantar este tipo de sistemas se requiere de tecnología, equipos e infraestructura de alto valor económico (Csanalosi, Dobreff, Pašić, Molnar, & Toka, 2020).

Por lo anterior ZigBee es un tipo de comunicación inalámbrica de bajo costo y puede ser utilizado para la monitorización de la práctica del deporte y sus características mediante sensores como se aprecia en la investigación (Picallo Guembe, 2016). De este modo por su tamaño y su eficiencia energética puede ser implementada para monitorear la posición, aun así, se requiere encontrar un modelo de propagación ajustado a las condiciones de canchas de microfútbol en un espacio semi abierto que determine la posición de un jugador.

1.5 Formulación del Problema

¿Cuál es la precisión de un sistema de posicionamiento basado en ZigBee en una cancha de microfútbol?

1.6 Objetivos

1.6.1 Objetivo General

Determinar la precisión de un sistema de posicionamiento de un jugador basado en ZigBee, dentro de una cancha de microfútbol, haciendo uso de la técnica RSSI.

1.6.2 *Objetivos Específicos*

- Diseñar un sistema de posicionamiento en una cancha de microfútbol basado en ZigBee utilizando mediciones de RSSI y ajustando un modelo de propagación.
- Diseñar un sistema de posicionamiento en una cancha de microfútbol basado en ZigBee, utilizando mediciones RSSI y ajustando un algoritmo de Machine Learning.
- Implementar los sistemas de posicionamiento dentro de la Cancha #1 de la universidad CESMAG
- Analizar la posición estimada que arrojan los sistemas, con respecto a la ubicación real del jugador.

1.7 *Justificación*

Actualmente el uso de sistemas de posicionamiento permite reproducir las trayectorias de jugadores en el campo de juego, lo que da acceso a una información crucial para el análisis de los movimientos de jugadores. Sin embargo, la tecnología con la precisión suficiente para este tipo de aplicaciones es muy costosa y de difícil acceso.

Por esta razón, se propone determinar si el uso de la tecnología ZigBee es lo suficientemente preciso para establecer la posición que un jugador, dentro de la cancha #1 de la Universidad CESMAG, considerando su facilidad de análisis y bajo costo. De ser así, podría ser aplicable a otros deportes que requieran obtener la posición para un monitoreo del jugador, esto mediante mediciones RSSI.

Se optó por la tecnología ZigBee debido a que su tamaño permite una fácil instalación en el espacio deportivo. Determinar la posición del jugador dentro del campo deportivo es necesario para que los directores técnicos puedan realizar un monitoreo del juego, saber cuáles errores en sus movimientos son recurrentes, y cuales se deben corregir, de este modo los dispositivos XBee

se plantean como una alternativa para el análisis deportivo, que puede ser explotada más a profundidad en un futuro.

1.8 Viabilidad

Para la realización de este proyecto se contó con el permiso de la Universidad CESMAG para el uso de los dispositivos ZigBee con los que se cuenta en el laboratorio. De la misma manera se dispuso el espacio de las canchas de microfútbol o aulas para pruebas preliminares y los recursos humanos necesarios para realizar el estudio del sistema y el montaje de los dispositivos.

También se tuvo presente dos aspectos más para la viabilidad del proyecto, el primero es el aspecto técnico ya que los dispositivos ZigBee registran la potencia que reciben de otros nodos y permiten extraerla a través de una API para ser procesados por otro tipo de microcontrolador y el segundo aspecto es financiero dado que los investigadores cuentan con la disponibilidad económica para hacer el montaje del prototipo del proyecto en las canchas que ofrece la Universidad CESMAG.

1.9 Delimitación

El proyecto tiene como fin la aplicación de la tecnología inalámbrica ZigBee para determinar la posición de jugadores en el espacio deportivo, utilizando mediciones de RSSI, implementado en las canchas de microfútbol de la Universidad CESMAG, utilizando 5 dispositivos ZigBee con microcontroladores para la implementación del prototipo.

2. Tópicos del Marco Teórico

2.1 Antecedentes

2.1.1 *Diseño e Implementación de un Sistema de Monitorización de Práctica Deportiva de Baloncesto en Entornos Contextuales WPAN/WBAN*

El primer antecedente elaborado por Picallo Guebe (2016), está enfocado en la monitorización de práctica deportiva de baloncesto. Esta presenta tres topologías de red que son soportadas por ZigBee, en la cual destaca una, permitiendo fiabilidad en las comunicaciones. Tomando conceptos básicos de métodos de transferencia de datos y direccionamiento en IEEE 802.15.4 y ZigBee.

El direccionamiento aplicado en este antecedente permitió que varios dispositivos interactúen entre sí para su comunicación de manera inalámbrica, además se hace el análisis de cómo influyen los fenómenos de propagación radioeléctrica y sus diferentes tipos, pérdidas de potencia en el trayecto, en el espacio libre o diferentes escenarios.

Para la recolección e interpretación de los datos obtenida se utiliza el programa Matlab, y dispositivos ZigBee, transmisor y receptor, la técnica usada es la de trazado de rayos (Ray-Launching), basada en óptica geométrica que consiste en lanzar rayos desde el transmisor y de este modo cada rayo se calcula de forma individual el nivel de potencia.

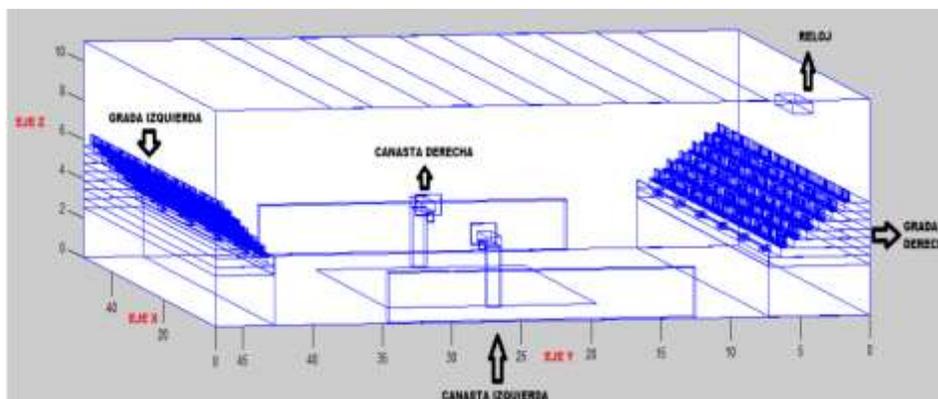
Los resultados se obtienen a partir de la recolección de datos de potencia en los nodos receptores, para la representación gráfica de estos datos hacen falta los mapas de cobertura y los perfiles de retardo de potencia (PDP, Power Delay Profile) para cada punto. Los primeros dan una información de una potencia obtenida en dBm, en un espacio con altura fija y los PDP-s presentan la potencia en dBm de los rayos que llegan al receptor en un instante de tiempo, dado

en nanosegundos, con este tipo de información se evaluó el grado de viabilidad de la comunicación establecida.

El campo en el que se hicieron las mediciones se puede observar en la Figura 1.

Figura 1. Esquema gráfico del escenario

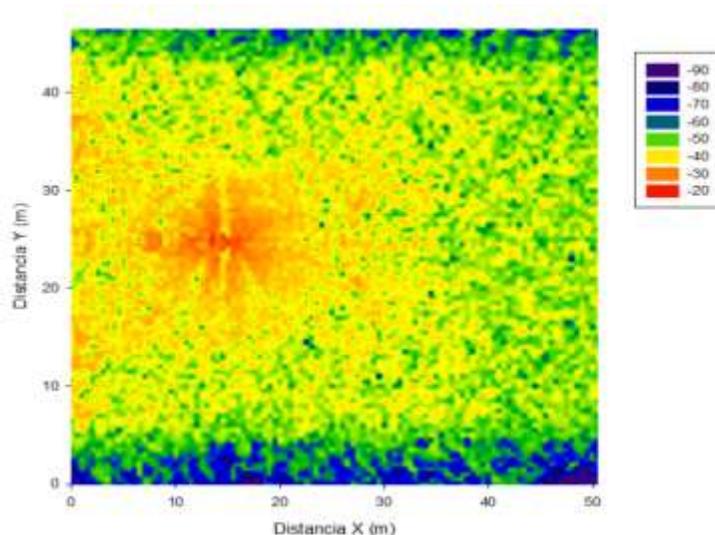
Fuente: (Picallo Gumbre, 2016)



El mapa de cobertura de una de las mediciones de potencia, colocada una antena transmisora encima del tablero de la canasta izquierda, en uno de los escenarios dentro del campo, se puede observar en la Figura 2.

Figura 2. Mapa de cobertura y potencia en dBm

Fuente: (Picallo Gumbre, 2016)

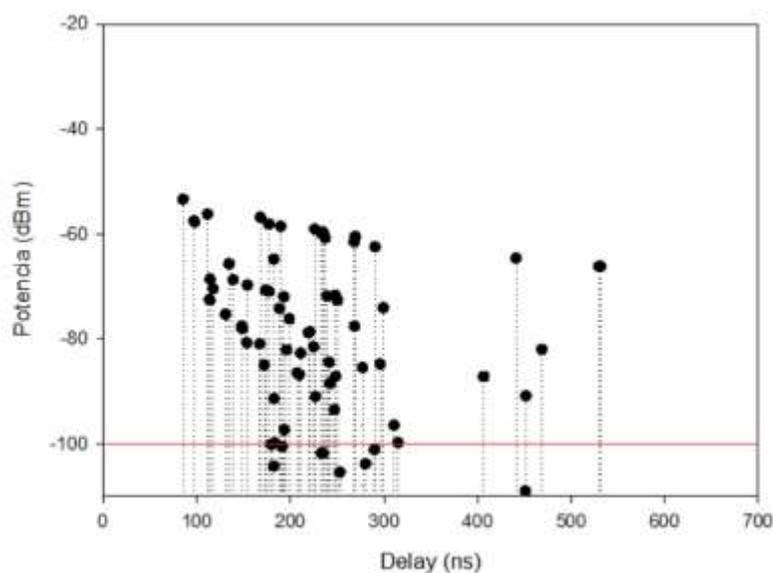


Los resultados corresponden a potencia (dBm), se puede observar que en los laterales de la Figura 2 muestran franjas de menor potencia ya que ahí se encuentran las gradas. En el punto donde está colocada la antena transmisora, se puede ver el tablero delimitado de forma muy precisa por la pérdida de potencia que produce este.

los resultados del PDP (Power Delay Profile), se aprecian en la Figura 3. En donde se evidencia que los primeros rayos que llegan son los de visión directa y aparecen cerca de los 80 nanosegundos.

Figura 3. Power Delay profile, Receptor Canasta Derecha

Fuente:(Picallo Guembe, 2016)



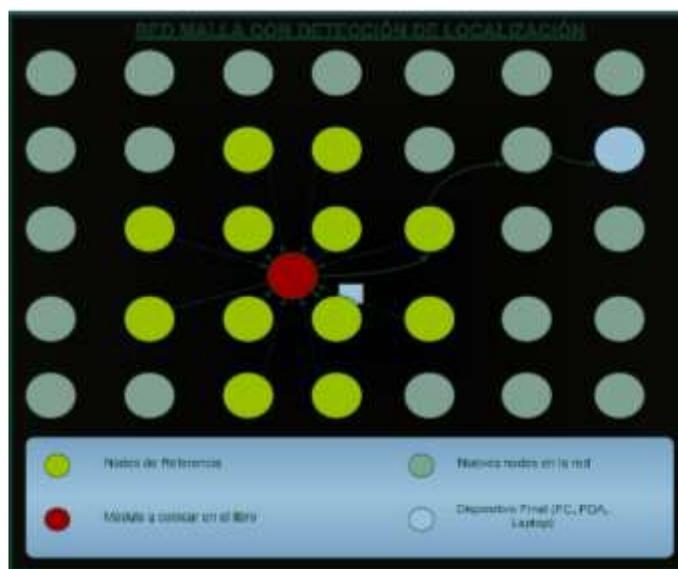
Este antecedente presenta una forma de obtención de potencia en diferentes puntos como también la información que brinda PDP, gracias a estos datos se puede obtener la distancia entre transmisor y receptor.

2.1.2 Sistema Prototipo de Posicionamiento Local Destinado a la Búsqueda de Libros Utilizando Tecnología ZigBee

En este antecedente elaborado por Bravo & Zambrano (2004), se propone el diseño de una red inalámbrica para el posicionamiento de libros, consta de nodos de comunicación interconectados y una base que se encarga de recibir toda la información que finalmente se conecta por medio de una interfaz RS-232 a una computadora. Debido a que el libro se encuentra en constante movimiento por las instalaciones de la biblioteca se hizo uso de una topología punto a punto sin restricción, conocida también como “ Mesh Networking”, que consiste en utilizar los mismos dispositivos de la red para transportar la información entre puntos que no tienen comunicación directa, como se puede observar en la Figura 4 .

Figura 4. Ejemplo de red ZigBee usando topología de red Mesh

Fuente: (Bravo & Zambrano, 2004)

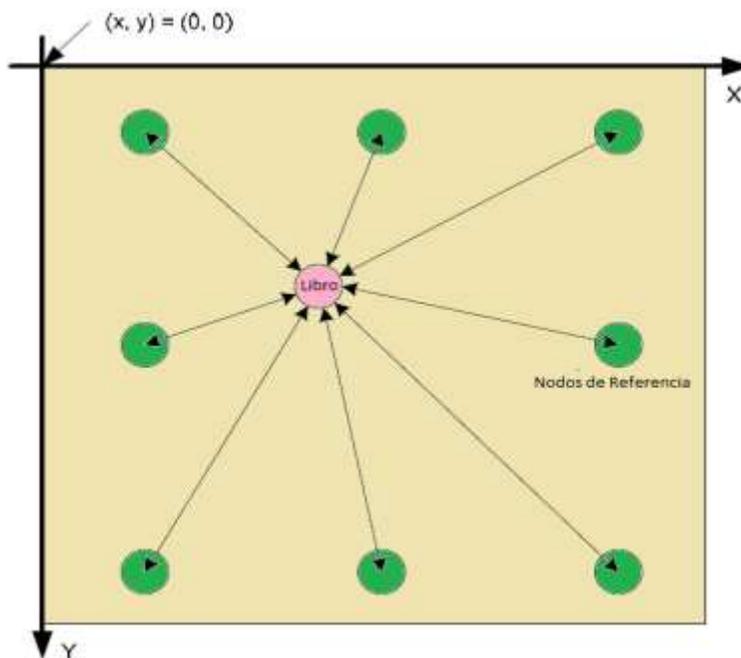


Para el cálculo de la posición se hace uso de una técnica denominada RSSI, en esta se miden valores que indican la intensidad de la señal recibida, esta medida de potencia está

expresada en dB y disminuye mientras la distancia aumenta, de esta forma se puede implementar el sistema de posicionamiento, como se aprecia en la Figura 5.

Figura 5. Ejemplo de Estimación de la Posición.

(Bravo & Zambrano, 2004)



En este caso se puede observar que existen nodos de referencia los cuales se encargan de recaudar los valores de RSSI enviados por el módulo instalado en el libro, y de esa forma realizar el cálculo de la posición.

Para realizar el cálculo se debe tener en cuenta dos parámetros de radio denominados A y n. El parámetro A es el valor absoluto de la potencia recibida por el nodo de referencia más cercano, esta potencia se expresa en dBm. El parámetro n se conoce como exponente de pérdidas y varía según el entorno donde está operando la red, y describe como la señal decae a medida que la distancia aumenta, para entender mejor los valores de entrada A, n, RSSI y de salida X, Y necesarios para realizar el cálculo de la posición se presenta la Tabla 1.

Tabla 1. Parámetros de Entrada

(Bravo & Zambrano, 2004)

NOMBRE	VAL. MIN	VAL. MAX	DESCRIPCIÓN
A	30	50	Valor Absoluto de la Potencia Recibida, en dBm a un metro de distancia del transmisor.
n – índice	0	31	Exponente de propagación de la señal. Depende el entorno.
RSSI	40	95	Indicador de Fuerza de la Señal Recibida medido en dBm. El valor ingresado es su valor absoluto.
X, Y	0	63.75	Valor en metros de las coordenadas relativas de X e Y

Se realizaron dos pruebas y se analizaron los datos mediante el software Z- Location Engine, mediante el cual se observó si la ubicación del objeto era la correcta, para la primera prueba se puso tres nodos más alejados y el restante en una posición más cercana. Para la segunda prueba se puso los 4 nodos más cerca del objeto (Bravo & Zambrano, 2004b). Los resultados obtenidos se muestran en la Tabla 2.

Tabla 2. Eficiencia de las pruebas

(Bravo & Zambrano, 2004)

	Nº Intentos	Nº Aciertos	Nº Desaciertos	% Error
Prueba #1	21	18	3	14.29%
Prueba #2	21	20	1	4.76%

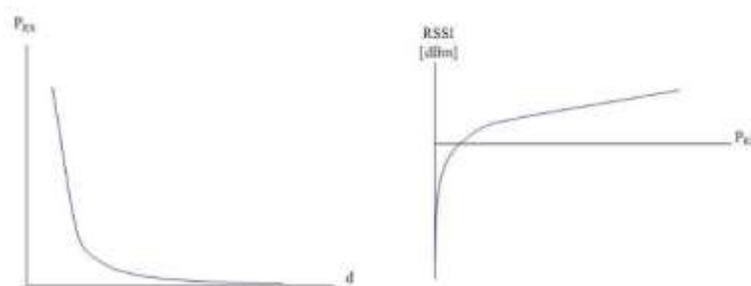
Este antecedente aporta a la investigación, el método de conexión, el modelo en la medida de la posición, el diseño de la red inalámbrica y los valores correspondientes a las entradas en un modelo RSSI.

2.1.3 *Diseño e Implementación de un Prototipo Para la Localización de un Objeto en Movimiento Usando ZigBee*

En este antecedente elaborado por Guerra (2010) se propone una interconexión de nodos y un receptor, y mediante el método RSSI se busca encontrar la posición de un objeto en movimiento. En este proyecto se tiene en cuenta que en sistemas embebidos la potencia recibida es convertida en un índice de potencia que incrementa cuando la distancia es menor, exponiendo de esta manera que la distancia es inversamente proporcional al RSSI. En la Figura 6 se puede apreciar lo anteriormente expuesto.

Figura 6. Comparación potencia vs distancia y potencia recibida como RSSI

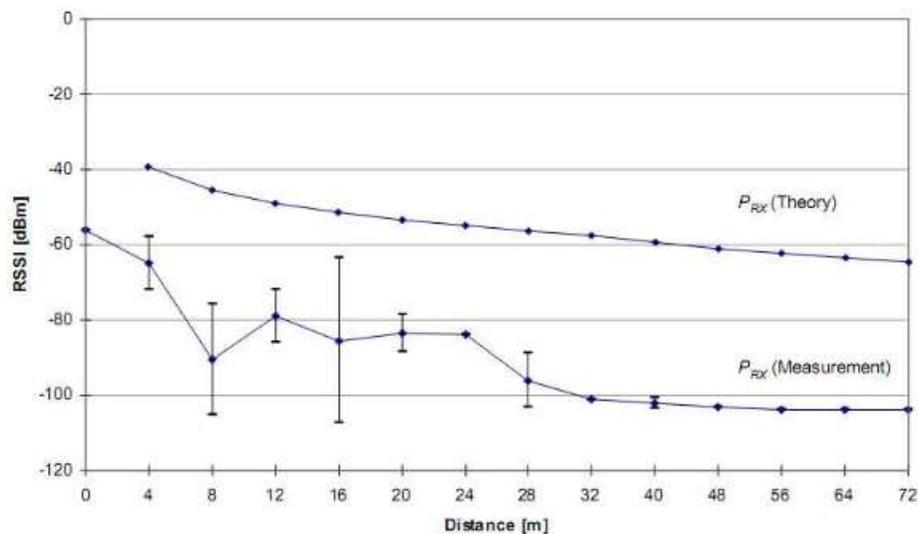
(Guerra, 2010)



Existen varios tipos de interferencia haciendo que degraden la calidad del índice RSSI, por lo tanto, este índice tendrá una varianza elevada y una entropía baja como se puede observar en la Figura 7.

Figura 7. Ejemplo de potencia recibida en un nodo

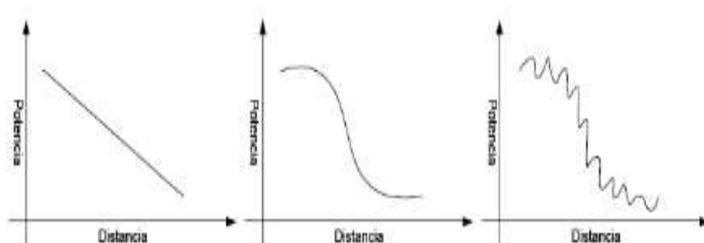
(Guerra, 2010)



En este caso se hizo uso de varios nodos de referencia y un nodo ciego encargado de recolectar las señales de todos los nodos de referencia. Luego se comparan las señales RSSI teóricas con las prácticas, y se puede apreciar que hay un cambio significativo cuando los componentes de sistema varían suavemente, y al final cuando se añaden componentes con múltiple trayectoria. La relación entre potencia y distancia de las mediciones fluctúa a medida que la distancia es más grande, esto se puede apreciar en la Figura 8.

Figura 8. Comparación valor RSSI teórico y práctico

(Guerra, 2010)



Aun teniendo en cuenta lo anterior se recomienda tener el mayor número de nodos posibles ya que esto hace más fiable la medida de la posición, porque en dado caso que uno de

los nodos sufra una interferencia quedan los restantes y se puede descartar el mencionado, en ese caso particular se deberá usar los nodos con el valor RSSI más alto.

En cuanto a problemas prácticos con el método RSSI se plantean los siguientes:

- Para una transmisión en 2.4GHz la atenuación a través de materiales de (20cm*10cm*40cm) es de 3dB, se recomienda evitar paredes de vidrio ya que en este caso la pérdida es de 19dB.
- Las características de propagación de la señal cambian frecuentemente.
- El RSSI depende del nivel de batería del nodo.

Con el actuador en movimiento se obtuvo una tabla en la cual se comparan las medidas reales en coordenadas (X, Y) en metros, con respecto a las señales calculadas mediante el método RSSI y se expone que, aunque existen diferencias entre los valores reales y los calculados la trayectoria del objeto es una. Todo lo anterior se puede apreciar en la Tabla 3.

Tabla 3. Coordenadas reales vs. Coordenadas calculadas

(Guerra, 2010)

Núm. veces	X[m]		Y[m]	
	Real	Calculada	Real	Calculada
1	1,5	1,5	3	3
2	2,75	3	3,5	3,5
3	2,5	2,5	2	2,25
4	2,25	2	0	0,5
5	1	0,75	0,75	0,75
6	0,75	1	4	4,25
7	2	2,25	5,25	5,75
8	3	3	7	6,75
9	2,75	3	6	6,25
10	2	1,75	3,75	4

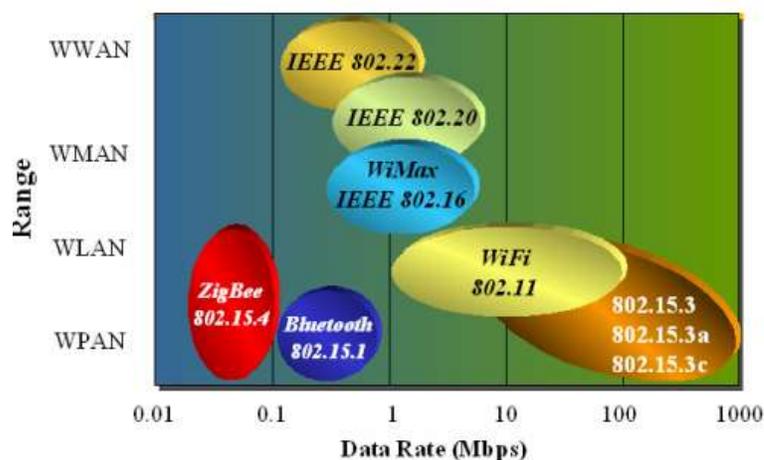
Este antecedente aporta a la investigación, el método de conexión, el modelo en la medida de la posición, el diseño de la red inalámbrica y los valores correspondientes a las entradas en un modelo RSSI.

2.1.4 Protocolo ZigBee (IEEE 802.15.4)

En el informe Moreno & Fernández (2007a) se presenta todo lo relacionado con el protocolo ZigBee. En principio ZigBee opera en las bandas libres de 2.4 GHz y 858 MHz para Europa y 915 MHz para estados unidos. La siguiente grafica presenta el espectro de ocupación en las bandas del protocolo 802 en la que se encuentra ZigBee.

Figura 9. Tecnologías en 2.4 GHz

(Moreno & Ruiz Fernandez, 2007a)



En cuanto a los nodos y topologías de red, en ZigBee puede haber 254 nodos y según la agrupación se pueden crear hasta 255 conjuntos/clusters de nodos permitiendo llegar a tener 64770 nodos en las topologías de red, estrella, malla o en grupos de árboles, como se puede ver en la siguiente figura:

Figura 10. Topologías de red

(Moreno & Ruiz Fernandez, 2007a)

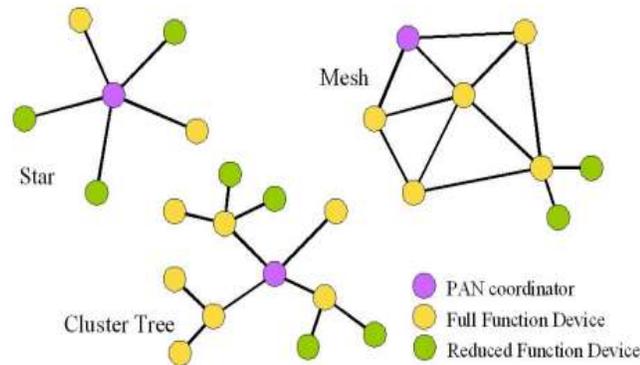


Figura 3. Topologías de Red

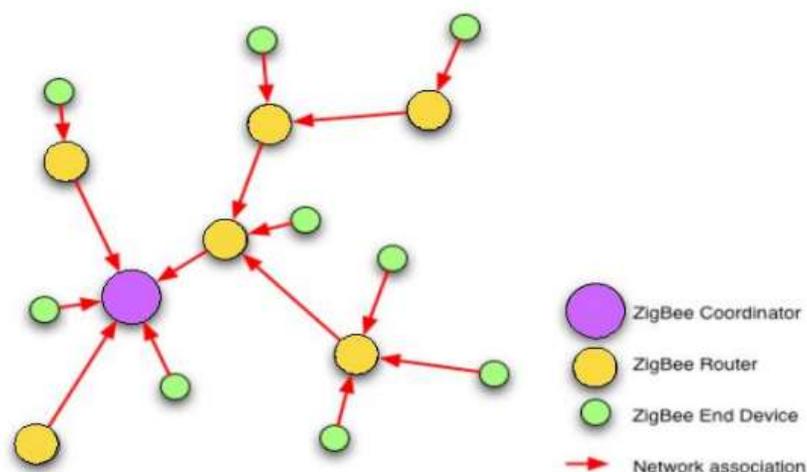
Para una buena comunicación ZigBee tiene 3 tipos de dispositivos:

- **Coordinador ZigBee (ZC):** solo debe existir 1 por red y sus funciones son las de encargarse de controlar la red y los caminos que tiene que seguir los dispositivos para comunicarse entre ellos
- **Router (ZR):** su función es encargarse del enrutamiento de saltos múltiples de los mensajes, este puede actuar como coordinador y también se asocia con el coordinador de la red o con otro router ZigBee.
- **Dispositivo final (ZED):** este se comunica con su nodo padre (el coordinador o un enrutador), no puede transmitir información a otros dispositivos y la mayor parte de tiempo puede estar dormido, aumentando la vida media de sus baterías.

La siguiente grafica presenta una posible configuración de la red:

Figura 11. Ejemplo de red ZigBee

(Moreno & Ruiz Fernandez, 2007a)



ZigBee permite que se puedan establecer rutas o caminos alternos para el intercambio de mensajes ante la caída de uno o más nodos que pertenecen a la ruta de comunicación.

El antecedente evidencia la información necesaria del protocolo para realizar cualquier tipo de configuración o red de ZigBee, además presenta las características de funcionamiento y operatividad del protocolo ZigBee que se tomó en cuenta para el proyecto.

2.1.5 Cálculo de distancia entre los nodos de una red inalámbrica ZigBee en función del parámetro RSSI

Según Vara & Poletto (2015), el indicador de potencia de la señal o RSSI es utilizado para medir la potencia en la antena del módulo receptor, aplicando la ecuación de Friis para transmisión en el espacio libre. Para hacer el respectivo cálculo se necesita de una distancia entre transmisor y receptor. Esta distancia se despeja, como se puede ver en la siguiente ecuación:

$$d = 10^{\frac{RSSI-A}{10n}} \quad (1)$$

Donde:

d = es la distancia entre el nodo transmisor y el receptor en metros.

A = es el valor de RSSI en la antena receptora a una distancia de un metro.

n = es una constante de perdida que depende del medio y de los obstáculos que existan entre nodos.

La Tabla 4 presenta los resultados obtenidos.

Tabla 4. RSSI medido promedio, distancia teórica y errores calculados a partir de las mediciones realizadas.

(Vara & Poletto, 2015)

Distancia Real	RSSI Medido Promedio	Distancia Teórica	Error absoluto (metros)	Error relativo
5	-52,333	4,591	0,409	0,082
10	-59,167	7,981	2,019	0,202
15	-61,667	9,986	5,014	0,334
20	-66,333	15,174	4,826	0,241
25	-69,167	19,563	5,427	0,217
30	-73,167	28,002	1,998	0,067
35	-75,167	33,501	1,499	0,043
40	-78,167	43,841	3,841	0,096
45	-79,667	50,151	5,151	0,114
50	-81,333	58,234	8,234	0,165

Como se puede ver, los valores obtenidos a partir del muestreo del parámetro RSSI con dos dispositivos XBee Serie 2 PRO presentan errores de hasta 8 metros, sin embargo, el error es relativo y se ve disminuido con distancias menores a 30 metros.

En los antecedentes propuestos se puede apreciar que todos usan técnicas RSSI para detectar la posición dentro de un espacio. En el caso de Picallo Guembe (2016) se usa esta técnica en un espacio cerrado para hacer la detección de un jugador de baloncesto en estado de reposo, esto se hace con el fin de monitorear prácticas deportivas. Se tienen en cuenta las tablas de pérdidas de propagación con un coeficiente teórico para ambientes cerrados, ya que se trabaja

en un espacio deportivo cubierto. Para el antecedente de Bravo & Zambrano (2004) se hace el reconocimiento de libros para una biblioteca implementando la tecnología ZigBee como una respuesta a identificar y posicionar los libros si se es necesario, en este caso no se tiene en cuenta el modelo de propagación y se usa una topología en malla que permite la interconexión de nodos múltiples en el medio. En el antecedente de Guerra (2010) se plantea la tecnología ZigBee como respuesta a la detección de un objeto en movimiento, en este caso en particular se hace un análisis de como el método RSSI es afectado por la distancia y el medio en que se propaga.

El aspecto diferenciador respecto a estudios previos radica en la elaboración de un modelo de propagación adecuado para los espacios deportivos semiabiertos de la Universidad CESMAG. En la presente investigación se consideró también el eje Z, ya que se tuvo en cuenta que las alturas de los jugadores no serán siempre las mismas, al igual que la altura a la que se encuentran ubicados los dispositivos finales. Con respecto al tratamiento de los datos, se optó por realizar una revisión manual de los datos obtenidos en cada punto y, posteriormente, filtrarlos para ajustar el modelo de propagación. Además, debido a las nuevas tecnologías disponibles, se ajustó y entrenó un modelo de aprendizaje automático (machine Learning) para predecir las coordenadas dentro del campo. En cuanto a la precisión, se realizó una comparación entre los ajustes de los modelos en los cuales se determinó cual tenía la mejor predicción de las coordenadas, los resultados y graficas representativas de estos errores se pueden apreciar en la sección 4.6.

2.2 Enunciados de los Supuestos Teóricos

2.2.1 Protocolo ZigBee

En el documento de Moreno & Fernández (2007b) se da a conocer que ZigBee es un estándar que define un conjunto de protocolos para el armado de redes inalámbricas de corta y baja velocidad de datos, que opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz.

El estándar ZigBee fue diseñado con las siguientes especificaciones:

- Bajo consumo de potencia y uso de batería.
- Bajo costo de dispositivos.
- Optimizado para ciclo efectivo de transmisión menor a 0.1%.
- Velocidad de transmisión menor a 250 Kbps.
- Comunicación a corto alcance (típico menor a 50 metros).

ZigBee es un estándar para usar en redes de sensores y actuadores que operan a batería.

Entre las aplicaciones que puede tener están:

- Domótica.
- Automatización.
- Reconocimiento remoto.
- Juguetes interactivos.
- Medicina.
- Etc.

Dentro de la tecnología ZigBee, los dispositivos XBee son utilizados en sus diferentes versiones para transmitir datos a cortas y largas distancias, además debido a su gran configurabilidad, estos dispositivos pueden ser usados para múltiples aplicaciones dentro del área de la ingeniería, dado que en su mayoría solamente intervienen para la transmisión de datos, se

plantea también la posibilidad de aplicaciones adicionales en las cuales se haga uso de todas sus características.

Para este caso los dispositivos XBee S1 pro cuentan con las características necesarias de comunicación, para recaudar las mediciones RSSI necesarias para el ajuste de los modelos que se implementaron.

2.2.2 Pérdidas de propagación

El modelo de pérdidas de propagación por distancia logarítmica describe cómo la potencia de la señal recibida disminuye de manera logarítmica a medida que aumenta la distancia entre el transmisor y el receptor. Este modelo se aplica tanto en canales de radio internos (dentro de edificios) como externos (al aire libre). En términos sencillos, esto significa que a medida que se aleja de la fuente de transmisión, la señal se debilita siguiendo un patrón predecible, que puede ser representado matemáticamente utilizando una escala logarítmica. Este modelo es útil porque ayuda a predecir la calidad de la señal y la cobertura en diferentes entornos (Rappaport, 2002). La pérdida media del trayecto entre el Trasmisor y Receptor se expresa en función de la distancia utilizando un exponente de pérdida de trayecto igual a n .

El modelo de pérdidas de propagación por Log-distancia viene dado por la ecuación:

$$PL(dB) = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) \quad (2)$$

Donde:

n = es el exponente de pérdida de trayectoria.

d_0 = es la distancia de referencia cercana que se determina a partir de mediciones cercanas al trasmisor.

d es la separación de trasmisor y receptor.

$PL(d_0)$ = es la pérdida de propagación en decibelios (dB) a la distancia de referencia d_0 , calculada utilizando el modelo de pérdidas por espacio libre de Friis.

El exponente de pérdida de trayectoria, denotado como n , es un parámetro fundamental en los modelos de propagación de señales, que indica la rapidez con la que la potencia de la señal se reduce a medida que la distancia entre el transmisor y el receptor aumenta. Este exponente varía según el entorno de propagación. En espacios abiertos o en línea de vista, n tiende a ser más bajo (cercano a 2), mientras que en entornos urbanos densos o interiores, donde hay muchos obstáculos, n puede ser significativamente mayor. Seleccionar una distancia de referencia apropiada para el entorno específico es crucial para obtener un valor preciso de n , la Tabla 5 indica los exponentes de pérdidas de trayecto en diferentes entornos.

Tabla 5. Exponentes de pérdida de trayecto para diferentes entornos

(Rappaport, 2002)

Entorno	Exponente de pérdida de trayectoria, n
Espacio libre	2
Área Urbana radio celular	2,37 a 3,5
Radio móvil urbana en sombra	3 a 5
En la línea de visión de un edificio	1,6 a 1,8
Obstrucciones en edificios	4 a 6
Obstrucciones en fábricas	2 a 3

La anterior tabla describe los valores del exponente n según el entorno, como son en espacio libre, radio celular urbana, con obstrucciones dentro de edificios, entre otros.

2.2.3 Geometría basada en la localización

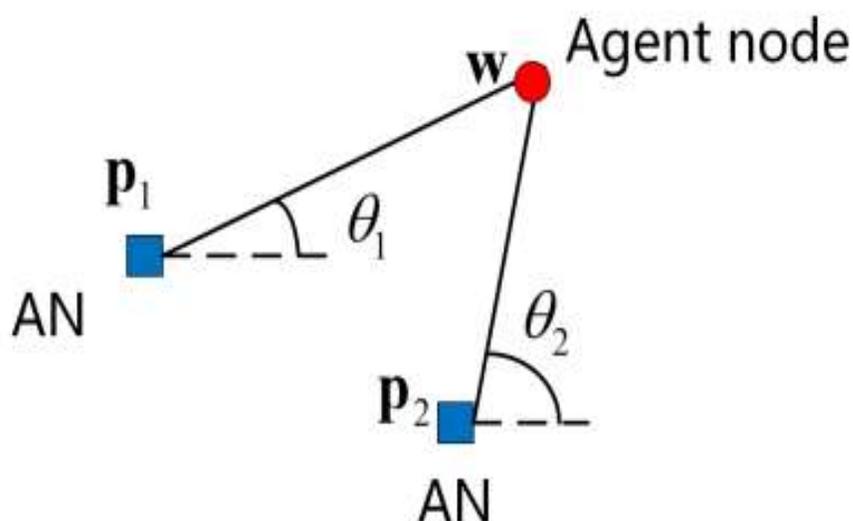
Según Xiao & Zeng (2022a), Existen tecnologías basadas en la geometría que explotan las propiedades geométricas de los triángulos para localizar un nodo agente. Los métodos geométricos tienen dos variantes principales: la trilateración y la triangulación.

La triangulación consiste en el cálculo de la posición de un objeto utilizando los ángulos y distancias medidas desde puntos de referencia conocidos. Este método utiliza la técnica AOA (Angle of Arrival - Ángulo de Llegada), donde se mide el ángulo en el que una señal llega respecto a una dirección de referencia. El AOA, también conocido como dirección de llegada, se refiere al ángulo en el que la señal se propaga entre los nodos de anclaje (ANs) y el objeto a localizar. Para llevar a cabo la triangulación y localizar el nodo agente, se necesitan al menos dos puntos de referencia o nodos de anclaje.

Por otro lado, la trilateración se basa en la medición de las distancias desde tres o más puntos de referencia conocidos hasta el nodo agente. Utilizando las intersecciones de los círculos formados por estas distancias, se puede determinar la ubicación precisa del nodo. Ambos métodos son fundamentales en diversas aplicaciones de localización, proporcionando la capacidad de determinar posiciones con alta precisión en entornos tanto interiores como exteriores. La siguiente grafica indica una posible configuración de triangulación:

Figura 12. Triangulación basada en AOA

(Xiao & Zeng, 2022b)



En donde:

- AN = son los dispositivos transmisores o puntos de referencia.
- Agente node = es el nodo o dispositivo al cual se necesita conocer su posición.

2.2.4 Método de posición basados en medición de RSSI

Una señal de radiofrecuencia sufre atenuación a medida que se aleja del punto de transmisión. Uno de los métodos más utilizados para estimar la posición en sistemas inalámbricos es la medición del nivel de potencia de las señales recibidas (RSSI). Esta técnica permite inferir la distancia desde el transmisor, basándose en el conocimiento de la pérdida de trayectoria y el nivel de potencia de la señal transmitida. Al medir el RSSI, es posible calcular la distancia entre el transmisor y el receptor mediante la siguiente ecuación (Rappaport, 2002).

$$RSSI = P_{Tx} - (L_{FS} + L_T + L_{T-x}) \quad (3)$$

Donde:

P_{Tx} = Potencia emitida por el transmisor [dBm]

L_{FS} = Pérdida por espacio libre

L_T = Pérdida por transmisión

L_{T-x} = Pérdida por transmisión hasta el punto de cálculo. Los valores de la potencia permiten calcular la distancia entre transmisor y receptor teniendo en cuenta los parámetros y escenarios establecidos.

2.2.5 Machine Learning

Según (Batta, 2020), Machine Learning o aprendizaje automático es una rama de la inteligencia artificial que se ocupa del desarrollo de algoritmos y métodos que permiten a los ordenadores aprender y hacer predicciones o tomar decisiones basadas en datos. En lugar de

programarse explícitamente para realizar una tarea específica, el aprendizaje automático utiliza patrones y relaciones en los datos para mejorar la eficiencia de determinadas tareas.

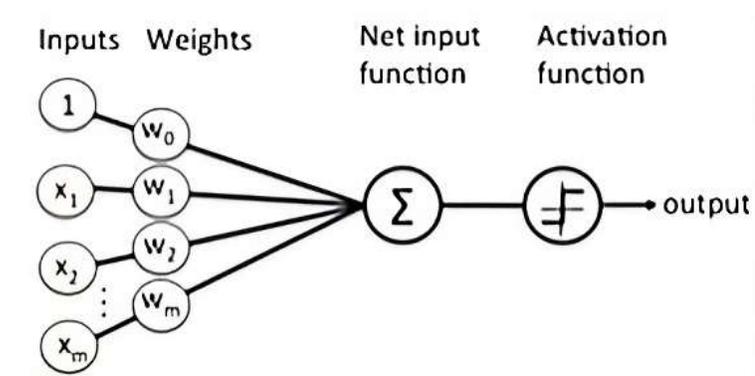
En el ámbito de Machine Learning, existen varios tipos de aprendizaje, entre ellos el aprendizaje supervisado y el aprendizaje no supervisado. El aprendizaje supervisado es una tarea de aprendizaje automático en la que se aprende una función que asigna entradas a salidas basándose en ejemplos de pares de entrada-salida. En el aprendizaje supervisado, una función se deriva de datos de entrenamiento etiquetados que consisten en un conjunto de ejemplos de entrenamiento. El conjunto de datos de entrada se divide en un conjunto de datos de entrenamiento y un conjunto de datos de prueba. El conjunto de datos de entrenamiento contiene las variables de salida que deben predecirse o clasificarse.

Aprendizaje no supervisado se denomina aprendizaje no supervisado porque, a diferencia del aprendizaje supervisado, no hay una respuesta correcta ni un profesor. Los algoritmos de aprendizaje no supervisado sólo aprenden algunas características de los datos. Cuando se añaden nuevos datos, se utilizan las características aprendidas previamente para reconocer las clases de datos. Se utiliza principalmente para la agrupación y la reducción de características.

Entre otras características del aprendizaje automático, las redes neuronales son una herramienta poderosa dentro del Machine Learning, son un conjunto de algoritmos que intentan identificar relaciones en una serie de datos mediante un proceso que imita el funcionamiento del cerebro humano. En este sentido, las redes neuronales se refieren a sistemas de neuronas, tanto orgánicas como artificiales. Las redes neuronales pueden adaptarse a los cambios en los datos de entrada para que la red obtenga el mejor resultado posible sin tener que rediseñar los criterios de salida.

Figura 13. Red Neuronal

(Batta, 2020)



Una red neuronal artificial funciona en tres capas. La capa de entrada recibe la información. La capa oculta procesa la entrada. Por último, la capa de salida envía la salida calculada.

Keras es una biblioteca de código abierto para crear y entrenar modelos de redes neuronales en Python. Ofrece una interfaz de alto nivel que facilita el diseño y la experimentación de redes neuronales profundas, y es compatible con varios marcos back-end como Tensor Flow, Theano y Microsoft Cognitive Toolkit (CNTK). Fácil de usar y modular, Keras permite a los desarrolladores crear rápidamente prototipos de modelos y ejecutarlos eficientemente en CPU y GPU, lo que permite que los modelos se ejecuten en diversas plataformas y dispositivos, Keras. (s.f.).

2.3 Variables de estudio

2.3.1 Definición nominal de variables

2.3.1.1 Distancia

Para asegurar una comunicación óptima entre los dispositivos XBee, es fundamental considerar la distancia entre el transmisor y el receptor, así como el alcance mínimo y máximo de los dispositivos. La técnica de trilateración, que se utiliza para determinar la posición de un objeto, requiere conocer con precisión la distancia entre los dispositivos. Además, es necesario

establecer las coordenadas del jugador, midiendo las distancias (X, Y) hasta puntos de referencia predefinidos. Estas medidas permiten calcular la posición exacta del jugador dentro del área de cobertura, garantizando una comunicación eficiente y precisa entre los dispositivos XBee.

2.3.1.2 Precisión en la detección de la posición

Debido a los ajustes realizados en el modelo de propagación y el modelo de Machine Learning, fue necesario comparar las coordenadas reales de la cancha con los resultados obtenidos por ambos modelos. Para determinar la precisión, se calculó el valor absoluto de la diferencia entre las coordenadas predichas por cada modelo y las coordenadas reales del campo, con base en esos errores se realizaron graficas que permitieron identificar que precisión presentaba cada modelo y determinar finalmente cual tenía las mejores predicciones.

2.3.2 Definición operativa de variables

2.3.2.1 Cinta métrica

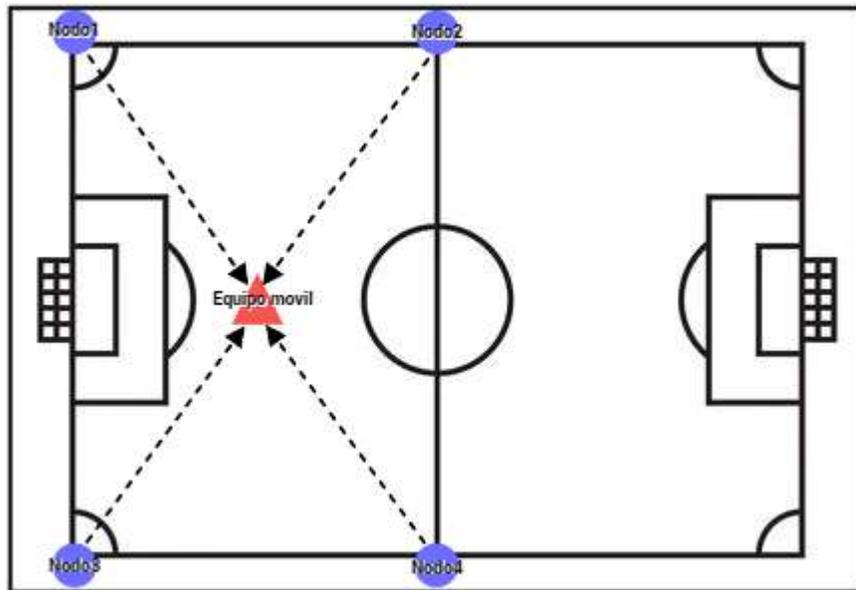
La cinta métrica es un instrumento de medición graduado que marca la longitud del metro y sus subdivisiones. Se utilizó este instrumento para medir las distancias entre el transmisor y el receptor a medida que se acercaban o alejaban. Para tomar las mediciones de RSSI dentro del campo, primero se establecieron los ejes de la cancha. Posteriormente, cada eje se dividió en espacios iguales, según el área específica del estudio. Las intersecciones de estas divisiones se utilizaron como puntos de referencia para adquirir los datos.

2.3.2.2 Posición

En el campo, los dispositivos finales fueron colocados en las esquinas de cada espacio designado para investigación. El dispositivo coordinador, por su parte, se desplazaba a través de las intersecciones que surgían de las divisiones de los ejes. Su función principal radicaba en la medición y almacenamiento de la intensidad de la señal recibida (RSSI).

La posición se midió de acuerdo a las coordenadas de origen, largo y ancho (X, Y) de la cancha y de los dispositivos dentro de la misma. La siguiente figura da una idea de cómo se ubicaron los dispositivos en la cancha de microfútbol.

Figura 14. Ejemplo montaje dentro del campo de microfútbol



2.3.2.3 Precisión

La precisión se estimó de acuerdo a la diferencia entre la posición real y estimada, con la ecuación de error cuadrático medio (Rappaport, 2002).

$$RMSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (P_r - P_{est})^2} \quad (4)$$

Dónde:

- n = Numero de posiciones tomadas en la cancha.
- P_r = Posición real
- P_{est} = Posición estimada

2.4 Hipótesis

2.4.1 Hipótesis de Investigación

Los sistemas de estimación de posición basados en ZigBee, usando la topología de red en estrella, tienen un porcentaje de error menor al 5%, haciendo uso de un modelo de propagación ajustado al entorno y mediante un algoritmo de Machine Learning. Como referencia se tiene el trabajo de Guerra (2010), donde se obtuvo un error promedio de 17.5% con respecto al aumento de la distancia y el parámetro RSSI.

2.4.2 Hipótesis nula

Los sistemas de estimación de posición basados en ZigBee, usando la topología de red en estrella, no obtienen un porcentaje de error menor al 5%, haciendo uso de un modelo de propagación ajustado al entorno y un algoritmo de Machine Learning.

2.4.3 Hipótesis alternativa

Los sistemas de estimación de posición basados en ZigBee, usando la topología de red en estrella, tienen un porcentaje de error entre el 5% y el 10 %, haciendo uso de un modelo de propagación ajustado al entorno y un algoritmo de Machine Learning.

3. Metodología

3.1 Enfoque

La investigación adopta un enfoque cuantitativo, ya que las variables relacionadas con el objeto de estudio se miden de manera numérica. Los resultados obtenidos facilitaron la evaluación de la precisión de los dispositivos XBee al establecer una conexión en estrella, para determinar la posición en una cancha de microfútbol.

3.2 Paradigma

Esta investigación se encuentra dentro del paradigma neopositivista, ya que se aplica conocimiento científico con el fin de que los resultados sean útiles y precisos.

3.3 Método

Ya que esta investigación es cuantitativa, el método que se utilizará es científico o también conocido como método empírico analítico (Quijano, 2009), ya que busca encontrar la precisión en la medición de la posición mediante un prototipo basado en ZigBee, con lo se quiere comprobar la hipótesis y resolver el problema de investigación.

3.4 Tipo de diseño

La investigación es de tipo descriptivo, dado que la finalidad es determinar la precisión en la medida de la posición y para lograrlo es necesario observar y recolectar datos, usando una topología de malla para la conexión y un método de medición de potencia o triangulación, para posteriormente interpretar y argumentar cual es la precisión porcentual de cada una de las técnicas para ubicar el jugador en una cancha de microfútbol.

3.5 Diseño de investigación

El diseño de esta investigación se define como experimento puro, dado que se pretende implementar un sistema de comunicación inalámbrica que usa el protocolo ZigBee, aplicando dos técnicas basadas en RSSI para estimar la posición de un dispositivo. El diseño experimental del trabajo es el siguiente:

RG1 X1 O1

RG1 X2 O2

RG1 – O3

Donde:

RG1: Conjunto de posiciones uniformemente distribuidas en la cancha de microfútbol.

X1: Sistema de estimación de la posición con modelo de propagación, basado en mediciones de RSSI.

X2: Sistema de estimación de la posición con modelo de propagación, basado en Machine Learning.

O1: Posición estimada.

O2: Posición estimada.

O3: Posición medida manualmente.

3.6 Universo

El universo corresponde al sistema de posicionamiento ZigBee implementado mediante una topología de red en estrella y se comprobó ajustando un modelo de propagación y un algoritmo de Machine Learning, usando las mediciones RSSI.

3.7 Muestra

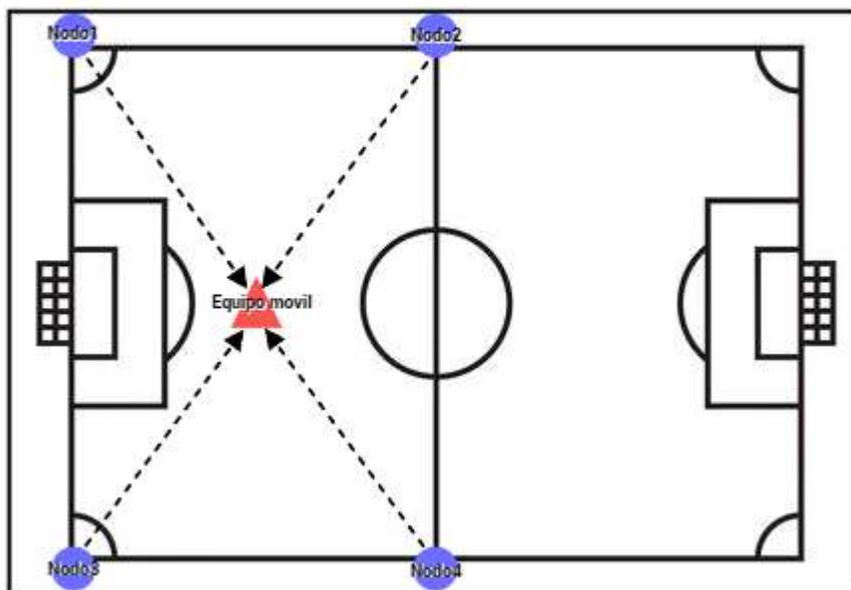
La muestra corresponde al sistema de posicionamiento ZigBee implementado en un jugador en una cancha de microfútbol, usando las técnicas de medición de potencia y un algoritmo de Machine Learning para determinar la posición.

3.8 Técnicas de recolección de información

Las mediciones se realizaron en la Cancha #1 de la Universidad CESMAG y se tomaron distintos puntos de referencia para realizar la recolección de datos en múltiples ocasiones, de este modo se minimiza la posibilidad de error en la Figura 15, se muestra un ejemplo de medición dentro de la cancha.

Figura 15. Ejemplo medición en la Cancha #1 de la Universidad CESMAG

Fuente: (30,235 Cancha Futbol Vectores, Ilustraciones y Gráficos - 123RF, n.d.)



La información necesaria para determinar la precisión en la medida de la posición se hizo con el método de observación directa, ya que se planteó contrastar la posición real medida con la cinta métrica, respecto a la medición que se hizo con el dispositivo XBee.

En este caso en particular es necesario considerar:

- Cuatro nodos o dispositivos finales
- Un dispositivo móvil principal encargado de medir la potencia recibida de los otros cuatro nodos en el desplazamiento dentro del espacio deportivo y mediante un microcontrolador se hizo la recolección de la información percibida por el equipo de medición.

3.8.1 Validez de la técnica

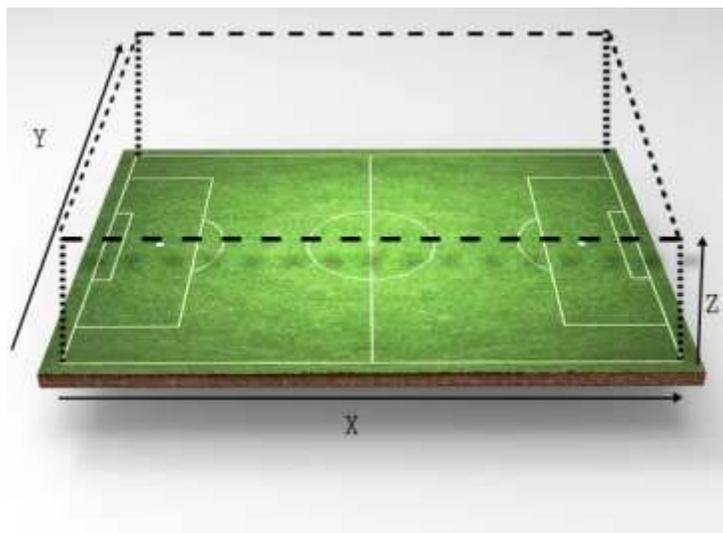
Las técnicas son válidas ya que han sido usadas en trabajos de investigación mencionados en los antecedentes, tales como el posicionamiento de libros, monitorización de prácticas deportivas y el uso de dispositivos XBee para monitorear un objeto móvil (Sección 2.1). Además, se ajustó un modelo de propagación y un modelo de Machine Learning usando mediciones de RSSI, una vez obtenidas las aproximaciones de cada modelo se procedió a comparar el error que arrojó cada técnica.

3.8.2 Confiabilidad de la técnica

La técnica es confiable ya que los datos de potencia obtenidos por los dispositivos XBee se recolectaron en múltiples ocasiones en distintos puntos de la Cancha #1 de la Universidad CESMAG, de este modo se pudo indagar en el comportamiento de los mismos y además filtrarlos de ser necesario.

Para realizar las mediciones de manera precisa, se llevó a cabo un cálculo de muestra probabilística con el objetivo de alcanzar una confianza del 95% y un margen de error del 10%. Como estrategia, se propuso segmentar el campo de fútbol en divisiones iguales, de 1 metro para el eje Y e 1.1 metros en el eje X, el total de puntos resultado de esta división fue el tamaño de la población. La metodología para la división del campo en los ejes (X, Y, Z) se detalla en la Figura 166.

Figura 16. Cancha de microfútbol con ejes X, Y; Z



De este modo el número de muestras necesario para que se cumplan los parámetros expuestos anteriormente se calcula de la siguiente forma:

$$n = \frac{Z^2 * p * q * N}{(N-1) * d^2 + Z^2 * p * q} \quad (5)$$

Donde:

N= tamaño de la población.

Z = es el coeficiente de nivel de confianza.

P = proporción aproximada del fenómeno en estudio en la población de referencia.

Q = proporción de la población de referencia que no presenta el fenómeno en estudio (1 - p). La suma de la p y la q siempre debe dar 1. Por ejemplo, si p= 0.8 q= 0.2.

Para el caso de estudio:

$$n = \frac{1.96^2 * 0.5 * 0.5 * 400}{(400-1) * 0.1^2 + 1.96^2 * 0.5 * 0.5} = 384.16 \approx 385 \quad (6)$$

Donde:

N = El total de puntos donde se toman datos en la cancha de microfútbol (400).

Z = es el coeficiente de nivel de confianza (1.96).

P = proporción aproximada del fenómeno en estudio en la población de referencia (0.5).

Q = proporción de la población de referencia que no presenta el fenómeno en estudio (0.5).

Para la cancha #1 de la Universidad CESMAG, se debieron tomar al menos 385 muestras para los estudios en los que se utilizó la totalidad del espacio para la investigación. Esto permitió obtener un margen de error aproximado del 10% y una confiabilidad del 95% en la ejecución. La posición se verificó comparando los resultados obtenidos de los modelos con las mediciones físicas realizadas en la cancha de microfútbol.

En los estudios donde se redujo el espacio para realizar las mediciones, se consideró la variabilidad del parámetro RSSI para ajustar adecuadamente los modelos.

3.9 Instrumentos de recolección de información

Los instrumentos que permitieron registrar los datos obtenidos fue por un dispositivo XBee coordinador que se encargó de medir la potencia de los cuatro nodos.

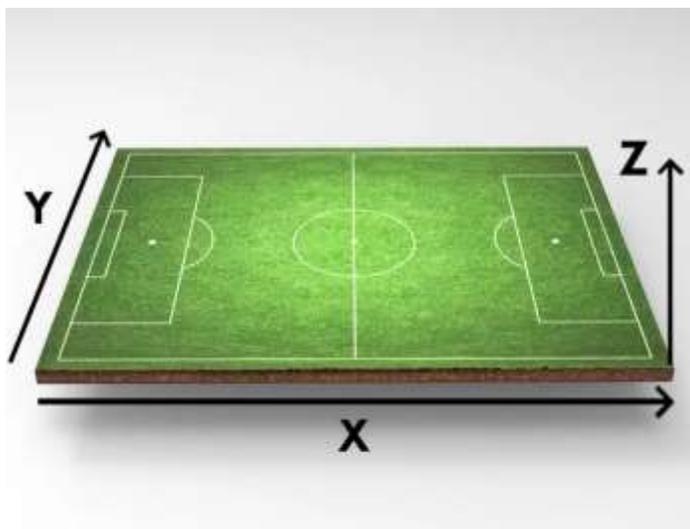
Para contrastar la información visual con la técnica de medición hizo falta un esquema plasmado en papel para poder anotar las mediciones del campo en las coordenadas (X, Y, Z) obtenidas con la cinta métrica. Además, los datos recolectados por los dispositivos de medición fueron recolectados individualmente para las dos técnicas, la primera se hizo con la técnica RSSI y el modelo de propagación adecuado a la Cancha #1 de la Universidad CESMAG; y en la segunda técnica los datos recolectados se procesaron mediante un algoritmo de Machine Learning. La forma en que se recolectaron los datos en las coordenadas (X,Y,Z) de la cancha y se adjuntó el RSSI de los XBee se puede apreciar en la Tabla 6.

Tabla 6. Tabla para la recolección de datos

Coordenada en X	Coordenada en Y	Coordenada en Z	Potencia XBee1	Potencia XBee2	Potencia XBee3	Potencia XBee4
--------------------	--------------------	--------------------	-------------------	-------------------	-------------------	-------------------

Figura 17. Ejemplo coordenadas en la Cancha #1 de la Universidad CESMAG

(30,235 Cancha Futbol Vectores, Ilustraciones y Gráficos - 123RF, n.d.)



Los datos recopilados por los dispositivos de medición se procesaron en Matlab, una plataforma de software donde se llevó a cabo los cálculos necesarios para los modelos. Esto permitió obtener las distancias y errores correspondientes para ambos métodos.

4. Resultados de la investigación

4.1 Diseño del sistema de posicionamiento en una cancha de microfútbol basado en ZigBee

Para el diseño de la red ZigBee dentro de la Cancha #1 de la Universidad CESMAG hizo falta utilizar cinco dispositivos XBee S1 pro. En este caso se optó por configurar uno de los dispositivos como coordinador y los cuatro dispositivos restantes como dispositivos finales donde se realizaron las consultas del parámetro RSSI. Teniendo en consideración que se creó una red, debe haber un coordinador (Moreno & Fernández, 2007) y el número de dispositivos finales se optó por 4, ya que estos abarcan estratégicamente el total de cancha #1 de la Universidad CESMAG, en donde se realizaron las mediciones de RSSI.

La configuración de los dispositivos XBee se realizó mediante el software XCTU de Digi International. Este provee una interfaz que permite configurar múltiples dispositivos además de una consola que permite probar las configuraciones de red establecidas. La configuración de los cuatro dispositivos y sus especificaciones, además de los requerimientos de red que se tuvieron en cuenta se pueden visualizar en el anexo 7.1.

Una vez configurada la red, era necesario que el dispositivo coordinador realice las consultas a los dispositivos finales. Para esto se hizo uso de la herramienta para la creación de tramas del software XCTU, el proceso detallado de la creación de la trama se puede ver en el anexo 7.1.1. A continuación, se presentan un ejemplo de la trama de solicitud y la trama de respuesta de los dispositivos XBee.

Trama de solicitud XBee Coordinador

7E 000F 17 01 0000000000000000 0002 02 6462 1D

Tabla 7. Trama enviada, utilizada para obtener el RSSI.

Descripción	Instrucción
<i>Start delimiter (Delimitador inicial)</i>	7E
<i>Length (Longitud de Trama)</i>	000F(15)
<i>Frametype (Tipo de Trama)</i>	17(RemoteATCommandRequest)
<i>FrameID (Identificador de Trama)</i>	01(1)
64 – <i>bitdest. address(Dirección Destino)</i>	0000000000000000
16 – <i>bitdest. address(Dirección Destino)</i>	0002
<i>Commandoptions (Opcion de Comando)</i>	02
<i>ATCommand (Comando AT)</i>	6462(db)
<i>Checksum (Suma de Verificación)</i>	1D

La trama de solicitud incluye la dirección de 16 bits de cada uno de los dispositivos finales, junto con el comando AT que solicita conocer el valor de RSSI percibido por cada dispositivo. Este valor de RSSI es crucial para asegurar la correcta ejecución de la solicitud en cada dispositivo. A continuación, se proporciona un ejemplo de respuesta de los dispositivos finales.

Trama de respuesta XBee Dispositivo final.

7E 0010 97 01 0013A2004090017F 0002 6462 00 24 76

Tabla 8. Trama de respuesta esperada.

Descripción	Respuesta
<i>Startdelimiter (Delimitador inicial)</i>	<i>7E</i>
<i>Length (Longitud de Trama)</i>	0010(16)
<i>Frametype (Tipo de Trama)</i>	97(<i>RemoteCommandResponse</i>)
<i>FrameID (Identificador de Trama)</i>	01(1)
64 – <i>bitsourceaddress(DirecciónRespuesta)</i>	0013A2004090017F
16 – <i>bitsourceaddress(Dirección Destino)</i>	0002
<i>ATCommand (Comando AT)</i>	6462(<i>db</i>)
<i>Status (Estado de la respuesta)</i>	00(<i>StatusOK</i>)
<i>Response (Respuesta solicitud (RSSI</i> <i>– Hexadecimal))</i>	24
<i>Checksum (Suma de Verificación)</i>	76

La trama de respuesta contiene la dirección de 64 bits de cada dispositivo, junto con el valor RSSI percibido, el cual se encuentra ubicado en el penúltimo byte de la trama. Después de verificar el correcto funcionamiento de la red utilizando el software XCTU, también fue necesario identificar posibles errores que podrían ocurrir durante las mediciones. Estos errores fueron identificados para su posterior inclusión en el código del microcontrolador encargado de realizar las solicitudes a los dispositivos finales. Los posibles errores se detallan en el anexo 7.1.1.

4.2 Montaje electrónico y algoritmo para la toma de datos dentro de la Cancha #1 de la Universidad CESMAG

Para recopilar datos dentro de la cancha, fue necesario implementar un montaje con un microcontrolador Arduino. Este se comunica mediante una interfaz serial con el dispositivo Coordinador y envía las tramas hacia los dispositivos finales. Además, se utilizó un adaptador de micro SD para almacenar los datos RSSI recopilados dentro de la cancha. Para la detección de errores, se emplearon cuatro LEDs de estado: dos rojos y dos verdes. Estos LEDs utilizan códigos de colores para facilitar la identificación de errores en la inicialización de los dispositivos.

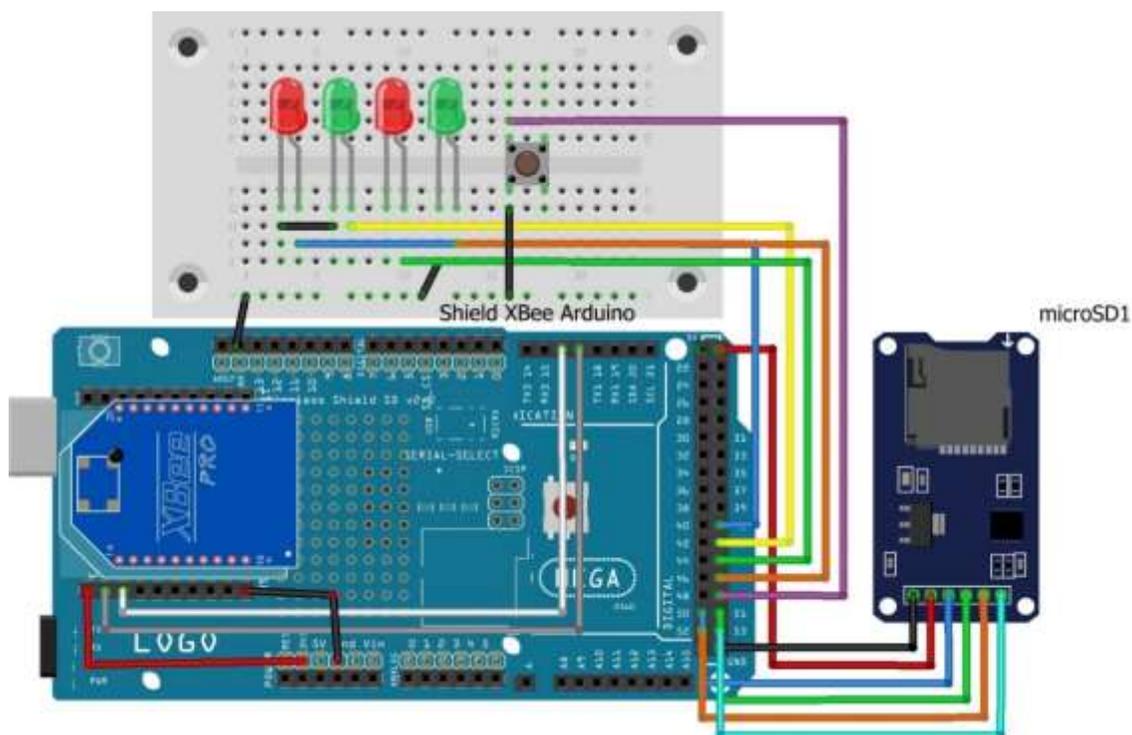
Tabla 9. Representación de bits y colores de posibles errores.

Estado de los colores	Representación en Bits (Rojo, Verde, Rojo, Verde)	Descripción
Esperando botón	0 1 0 0	Led verde encendido, estado normal del dispositivo.
Error de inicialización	1 1 1 1	Todos los leds Encendidos
Error en SD	1 1 1 0	Leds Rojos y Verde encendidos
Midiendo correctamente	0 1 0 1	Leds Verdes encendidos
Error en la medición	1 0 1 0	Leds Rojos Encendidos

Dentro de los códigos de colores, el valor uno representa un LED encendido y el valor cero representa un LED apagado. Además, se incorporó un pulsador en el montaje, lo que

permite, una vez que todos los dispositivos están inicializados, comenzar con la medición del parámetro RSSI de los 4 dispositivos finales. Se adjunta un esquema gráfico del montaje de prueba en la Figura 188.

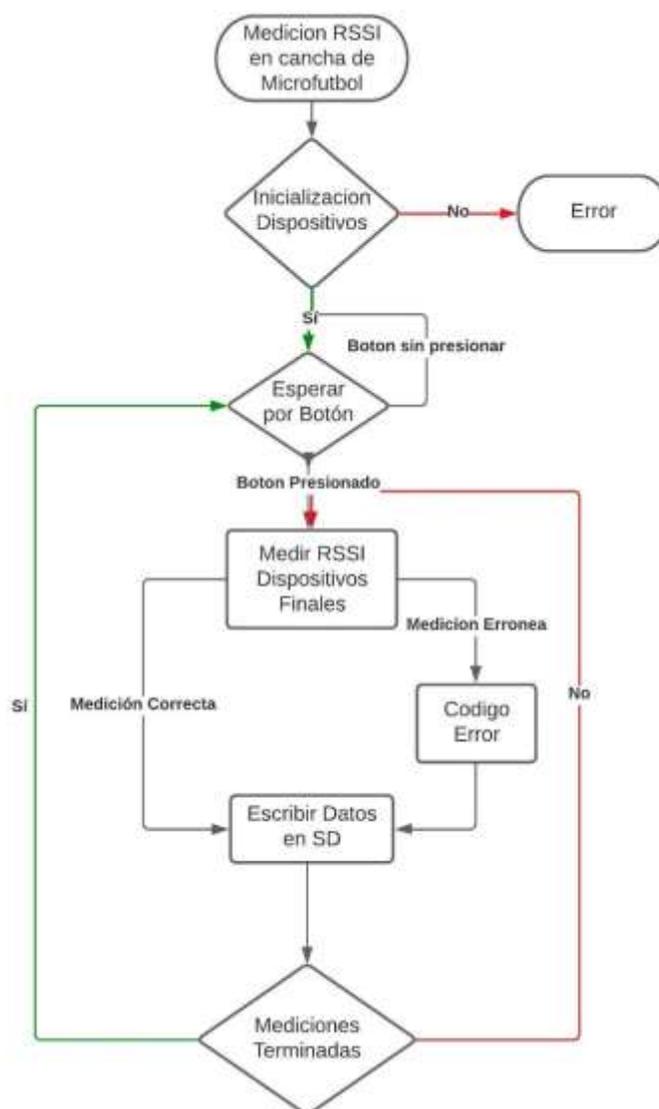
Figura 18. Esquema de conexión microcontrolador Arduino y XBee-PRO S1



4.2.1 Código Arduino para consulta y almacenamiento de parámetro RSSI dentro de la cancha.

Se seleccionó el microcontrolador Arduino en su versión Mega 2560. Se le acopló un Shield para ubicar el módulo XBee en la parte superior. Para desarrollar el código C para el microcontrolador, fue necesario utilizar librerías que permitieran el manejo del módulo SD, así como también se consideró el uso de las tramas generadas en el software XCTU para consultar el parámetro RSSI. El diagrama que explica el funcionamiento del código se presenta en la Figura 199.

Figura 19. Diagrama de flujo código Arduino y Coordinador XBee



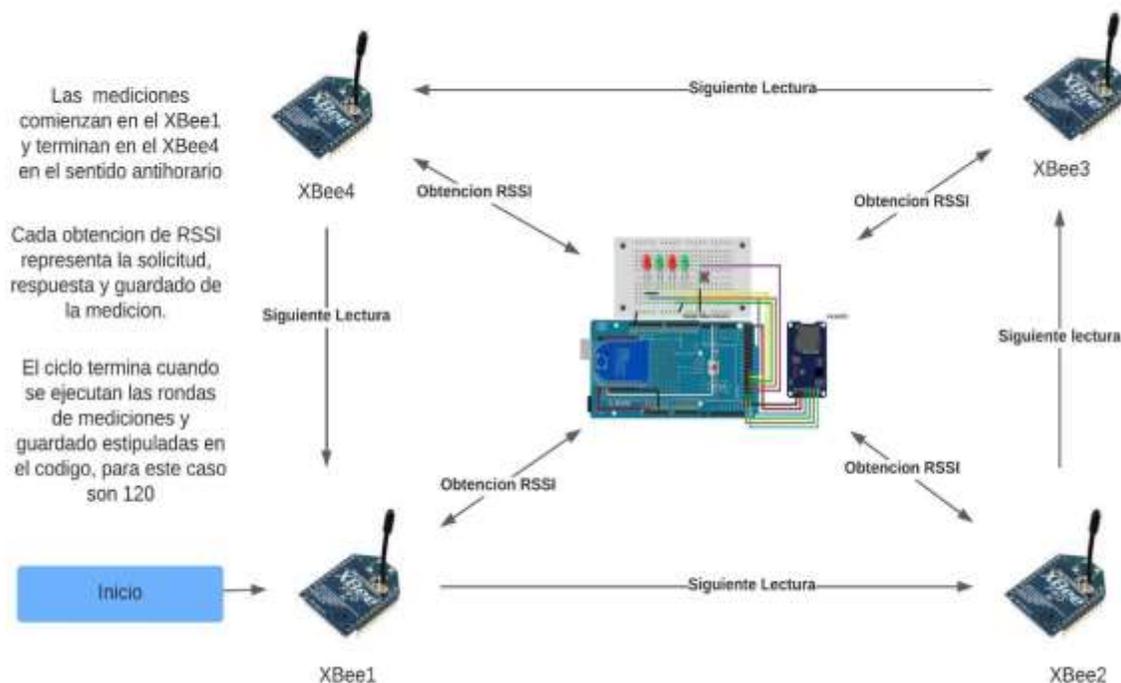
Cuando el dispositivo se pone en funcionamiento, el primer paso es la inicialización, en la cual se realizó la creación del archivo .csv para guardar los valores RSSI adquiridos por el XBee coordinador. Si la inicialización de la microSD falla o la comunicación serial entre el Arduino y el módulo XBee es errónea se presentará un patrón de colores de error en los leds.

Una vez la inicialización está completa, el microcontrolador pasa a un estado normal en el cual espera una instrucción para empezar a recolectar mediciones, este estado se denomina

espera por botón, y la instrucción necesaria para pasar a medir y guardar los RSSI es presionar el pulsador incorporado en el montaje.

Al presionar el pulsador, el microcontrolador y el coordinador XBee pasan al estado de *medición de RSSI de los dispositivos finales*. En este estado, se envía la trama de comando remoto AT (Remote AT Command) a cada uno de los dispositivos finales. Esta trama de solicitud incluye la dirección única y el checksum de cada dispositivo para su identificación. La trama de respuesta contiene el valor de RSSI de cada dispositivo en el penúltimo byte. Si el estado de la trama no es correcto o presenta algún error, el microcontrolador lo detecta y registra un código de error en un archivo .csv para su posterior análisis.

Figura 20. Representación de la secuencia de mediciones entre los dispositivos finales y el coordinador.

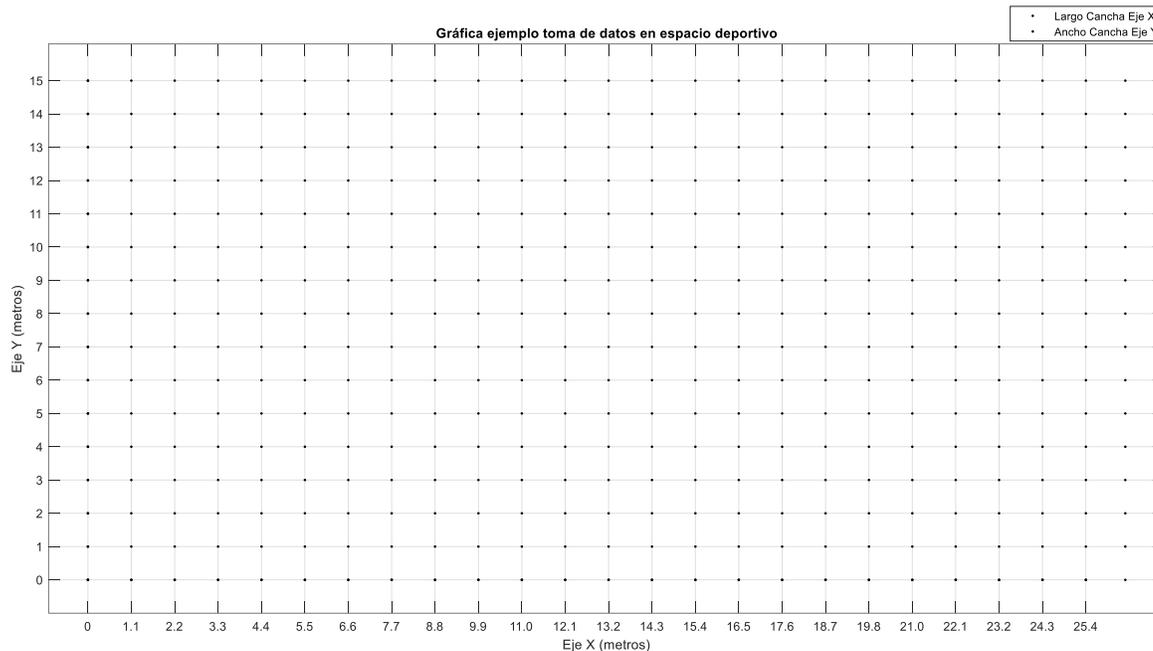


El tiempo que toma realizar una medición del valor RSSI en cada uno de los dispositivos se puede configurar en el código del microcontrolador, en los estudios de las secciones 4.3.1.1, 4.3.1.2, y 4.3.2 se configuro un tiempo de 30 ms para realizar la petición y respuesta de cada dispositivo final, y para el caso del último estudio de la sección 4.3.3, se aumentó el tiempo a 50 ms debido a que se determinó que tomar las muestras en más tiempo reducía los errores que se presentaban en los datos recolectados y disminuía su variación. Cada vez que se toma una medida en un dispositivo final, esta se guarda en el archivo csv en la tarjeta SD. Una vez que se completan las mediciones en cada uno de los dispositivos finales, el sistema vuelve al estado de espera por botón para iniciar la medición de los dispositivos en otra área de la cancha.

4.2.2 Adquisición de datos en la Cancha #1 de la Universidad CESMAG.

Para recopilar datos dentro de la cancha, fue necesario medir las dimensiones de la Cancha #1 de la Universidad CESMAG, seleccionada por su disponibilidad de alimentación eléctrica, y desarrollar un método para iniciar la recolección de datos. Se decidió dividir el campo en ejes X, Y Z. En el eje X, se obtuvo una medida de 26.3 metros; en el eje Y, una medida de 15 metros; y el eje Z se encuentra a una altura de 1.10 metros sobre el suelo. Una vez confirmadas estas medidas, se propuso iniciar la recolección de datos siguiendo la distribución planificada, como se muestra en la Figura 21.

Figura 21. Divisiones de la cancha para la toma de datos en los ejes (X, Y)



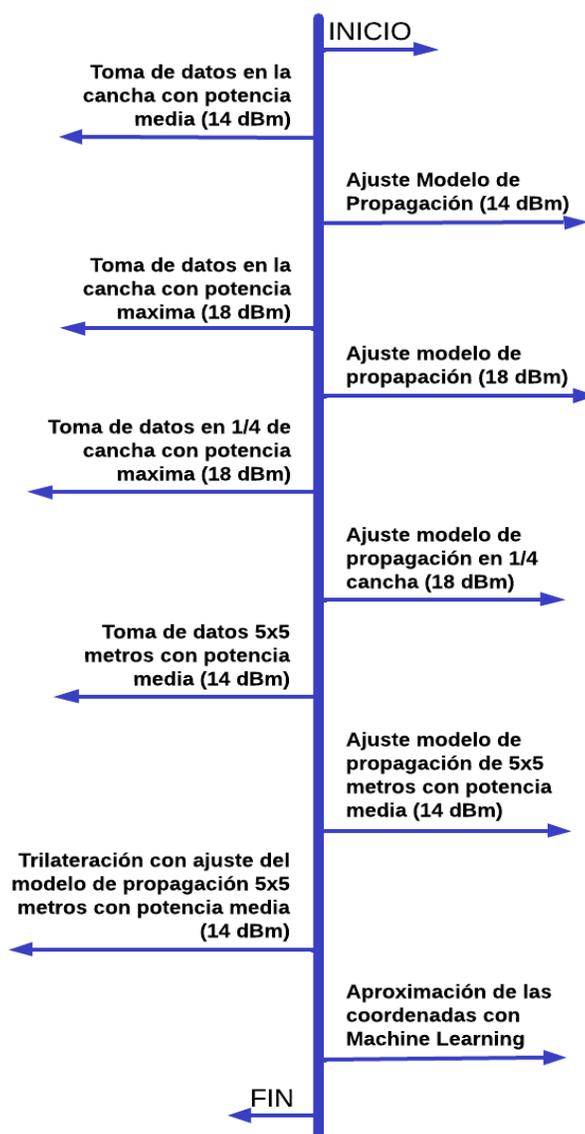
Al realizar las divisiones de la Cancha en los ejes (X, Y) se genera una serie de coordenadas en toda la cancha, cada coordenada representa una medida realizada. Debido a que se realizaron varios estudios a diferentes potencias y distancias dentro del espacio deportivo, la figura anterior brinda una visión general de cómo se realizó la recolección de datos dentro de la cancha de microfútbol. Aun así, siempre serán considerados los mismos ejes y altura respectiva en el eje Z para el dispositivo coordinador.

4.3 Ajuste del modelo de propagación Log Distancia.

Para determinar la precisión de los dispositivos XBee en la Cancha #1 de la Universidad CESMAG, se realizaron estudios con diversos enfoques que, con el tiempo, demostraron un mejor comportamiento en cuanto a la distancia entre el dispositivo final y el coordinador. Cada estudio preliminar contribuyó a tomar decisiones sobre cómo mejorar el ajuste del modelo para cada dispositivo y reducir el error en la distancia entre el dispositivo final y el coordinador.

Inicialmente, se variaron las potencias de trabajo de la red para encontrar la mejor aproximación de la distancia entre los dispositivos. Posteriormente, se tomó en cuenta el espacio en el que se realizaba el estudio, por lo que se utilizaron porciones más pequeñas del campo para realizar las mediciones y ajustar los modelos de cada dispositivo final. Esto se hizo con la intención de minimizar el error al realizar la triangulación para obtener las coordenadas. A continuación, en la Figura 22, se presenta una línea de tiempo que representa los estudios realizados.

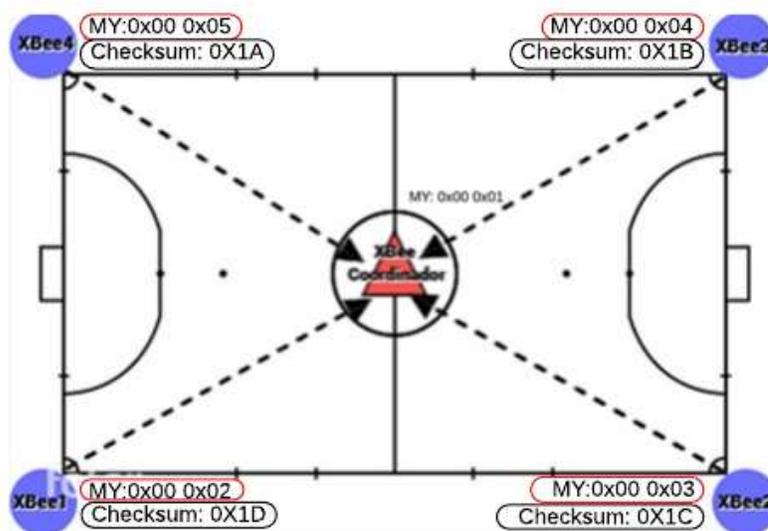
Figura 22. Línea de tiempo de los estudios realizados a las diferentes potencias



4.3.1 Parámetros generales para el ajuste de Modelos de Propagación en la Cancha #1 de la Universidad CESMAG.

Para los estudios realizados en todo el espacio deportivo se ubicó un dispositivo en cada esquina de la cancha teniendo en cuenta la distribución planteada en el apartado 4.2.2. Una representación de la ubicación de los dispositivos se presenta en la Figura 233.

Figura 23. Ubicación de todos los dispositivos en la Cancha #1 de la Universidad CESMAG.



Una vez recopiladas todas las mediciones de potencia RSSI necesarias en el espacio deportivo, fue imperativo realizar una limpieza de los datos para abordar posibles errores que puedan haberse producido durante la recolección de los datos y que se registraron en el archivo CSV.

Para calcular la distancia desde los cuatro dispositivos finales hasta el coordinador, se empleó una distribución de puntos en la cancha de manera uniforme. Con base en las coordenadas resultantes de la división de la cancha, se realizaron los cálculos pertinentes para

determinar las distancias requeridas. Este cálculo es esencial para evaluar el error en el modelo que se implementó utilizando las potencias recopiladas en el campo. Las fórmulas para realizar los cálculos de distancias desde el dispositivo final al coordinador se presentan a continuación.

$$d_{XBee1} = \sqrt{x^2 + y^2 + (AED - z)^2} \quad (7)$$

$$d_{XBee2} = \sqrt{(DX - x)^2 + y^2} \quad (8)$$

$$d_{XBee3} = \sqrt{(DX - x)^2 + (DY - y)^2} \quad (9)$$

$$d_{XBee4} = \sqrt{(x)^2 + (DY - y)^2} \quad (10)$$

Donde

x = Distancia desde el coordinador hasta el dispositivo final en el ejeX [mts]

y = Distancia desde el coordinador hasta el dispositivo final en el ejeY [mts]

AED = Altura a la que se encuentran los dispositivos End Device [mts]

DX = Distancia del espacio de medida en el ejeX [mts]

DY = Distancia del espacio de medida en el eje Y [mts]

Una vez determinadas las distancias reales en metros mediante las ecuaciones previamente descritas, se procedió a calcular las distancias logarítmicas. Estos valores son fundamentales para el desarrollo del modelo de propagación. El cálculo de la distancia logarítmica se inicia con la distancia más pequeña obtenida durante el proceso de cálculo de las distancias reales. Este valor se utiliza junto con todas las distancias reales calculadas para los dispositivos XBee, mediante la siguiente fórmula (Rappaport, 2002).

$$d_{log} = 10 \log_{10} \left(\frac{d_o}{d_{real}} \right) \quad (11)$$

Donde

d_o = Es la distancia de referencia o la distancia mas pequeña obtenida en el estudio [mts].

d_{real} = Distancia real calculada desde cada punto de la cancha al End device [mts].

Una vez calculada la distancia logarítmica, se ajusta el modelo de propagación log-distancia con los datos de potencia de señal recibida (RSSI) obtenidas, luego se realiza un ajuste estadístico mediante técnicas de regresión lineal para establecer una relación entre la distancia logarítmica y los valores de RSSI medidos; este ajuste permite determinar un parámetro n clave, que representa la pendiente de regresión realizada para ajustar los valores y que describe la atenuación de la señal en el entorno específico, validando el modelo con datos adicionales para asegurar que las predicciones sean precisas y fiables para estimar la distancia entre dispositivos XBee, se puede calcular el valor de la distancia entre el Coordinador y los dispositivos finales que arroja el ajuste del modelo con la siguiente formula (Rappaport, 2002).

$$d_{modelo} = \frac{d_o}{10^{\frac{0.1 * PR - P_o}{n}}} \quad (12)$$

Donde:

d_o = Distancia menor de referencia calculada [mts]

PR = Potencia RSSI promedio en cada coordenada [dBm]

P_o = Intercepto del modelo [dBm]

n = Pendiente del modelo (Variable de pérdida de trayecto).

El intercepto y las pendientes para los modelos de propagación fueron fruto de pasar los datos de potencia obtenidos en relación con su distancia logarítmica a un modelo de regresión lineal en Matlab, proporcionando las pendientes de los modelos como también los intercepto con los cuales se pudo realizar la predicción de la distancia entre los dispositivos finales

y el coordinador. Utilizando la distancia calculada por el modelo y la distancia real, se logró determinar el error de cada punto con respecto a los Dispositivos finales (Rappaport, 2002).

$$Error_{modelo} = |(distancia_{modelo} - distancia_{real})| \quad (13)$$

Esta evaluación fue crucial para tomar decisiones respecto al comportamiento de la precisión de los dispositivos, permitiendo variar las características de potencia y distancia en sus respectivos estudios. Más detalles de los ajustes realizados pueden apreciarse en los índices 4.3.1.1 y 4.3.1.2 en los cuales se varían las potencias en la totalidad del espacio deportivo. Además, se realizaron estudios en espacios específicos de la cancha debido a los resultados obtenidos en los índices previamente mencionados, los resultados para los ajustes en porciones más pequeñas de la cancha se pueden encontrar en las secciones 4.3.2 y 4.3.3, en las cuales se considera la mitad de la cancha y un espacio de 5x5 metros en el campo respectivamente.

4.3.1.1 Primer ajuste Modelo de Propagación en la Cancha #1 de la Universidad CESMAG.

Para este estudio específico, se configuro una potencia de transmisión de 14 dBm en todos los dispositivos, una vez recopiladas las distancias y mediciones RSSI dentro de la Cancha #1 de la Universidad CESMAG, se procedió a eliminar los datos con errores de lectura detectados por el coordinador durante las mediciones en cada coordenada, posteriormente se calculó un promedio de las potencias obtenidas en cada punto de la cancha para cada dispositivo XBee. No se llevó a cabo ningún otro tipo de filtrado adicional de los datos.

El escenario de la Cancha #1 de la Universidad CESMAG donde se realizó las mediciones de RSSI se la puede apreciar en Figura 244.

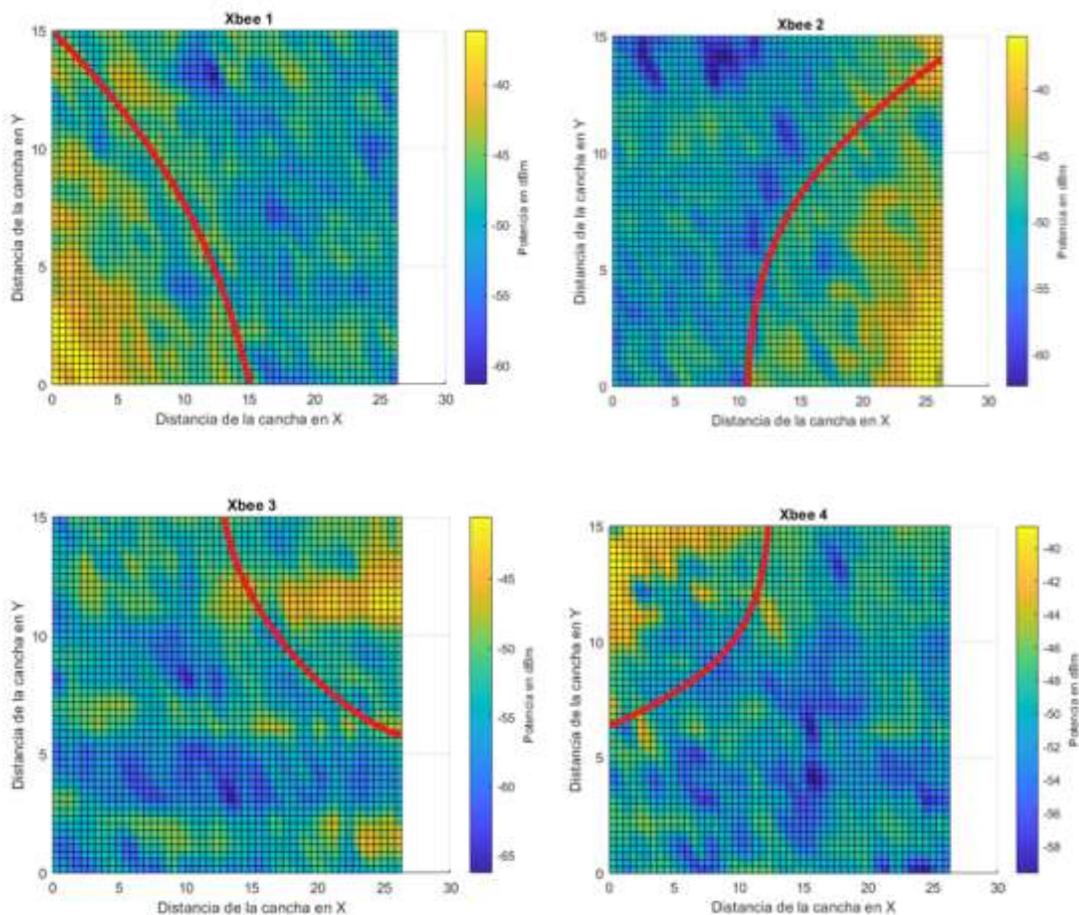
Figura 24. Escenario de la Cancha #1 de la Universidad CESMAG



El tiempo estimado para las mediciones de RSSI en el escenario, presentado anteriormente, requirió de dos días, para la totalidad de la toma de datos.

Con el objetivo de visualizar de manera más efectiva las potencias capturadas en todos los puntos, se recurrió a un mapa de calor. Este tipo de representación gráfica proporciona información sobre la variabilidad de las mediciones en el espacio deportivo. Los mapas de calor se pueden apreciar en la Figura 255.

Figura 25. Potencia recibida en la Cancha #1 Unicesmag con potencia transmitida igual a 14 dBm

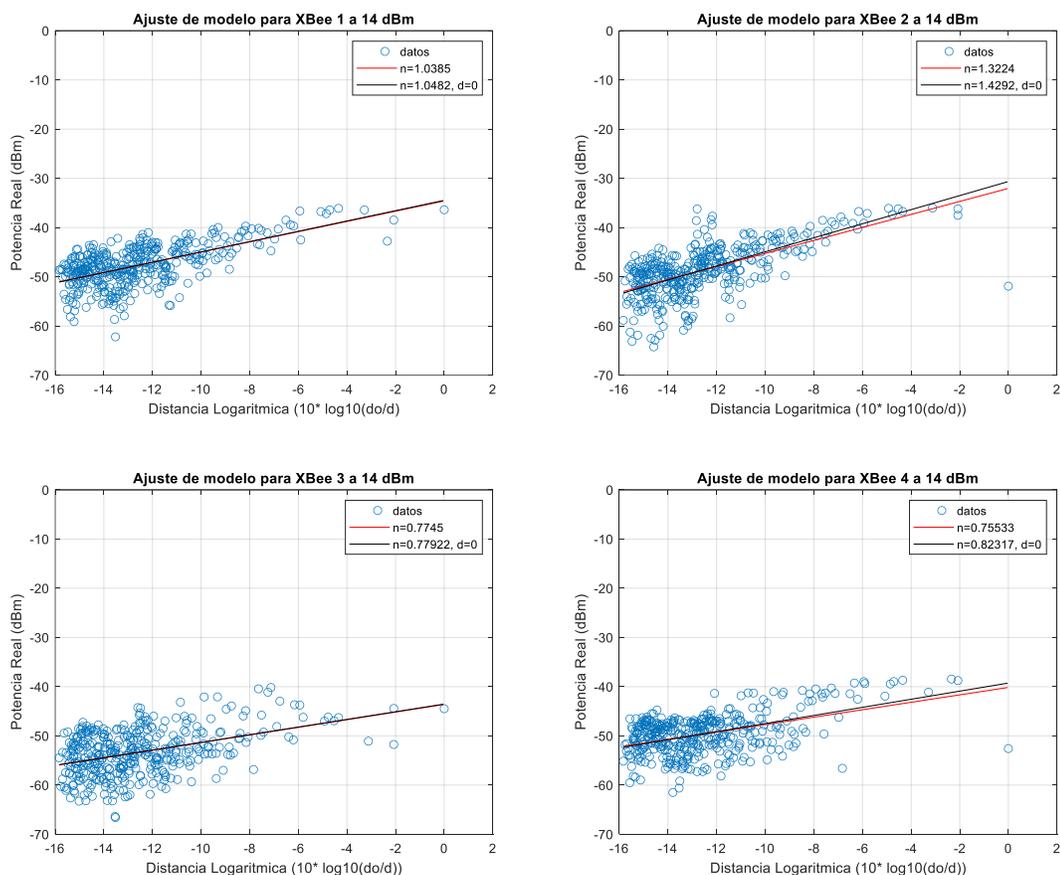


En el mapa de calor de los dispositivos, se observa una significativa variabilidad de RSSI en varios puntos del área de localización, especialmente a medida que el dispositivo coordinador se aleja de cada uno de los Dispositivos finales. Las líneas rojas en el mapa de calor indican dentro de esta la mayor intensidad de las potencias arrojadas por los dispositivos finales, así también la intensidad que se encuentra fuera de la línea roja indica una deficiencia de potencia por parte de estos mismos. Por lo tanto, también se debe tener en cuenta este mismo criterio para los mapas de calor siguientes.

Para este estudio en particular se ajustó un modelo de una sola pendiente para cada dispositivo final, de este modo se pudo obtener la aproximación del modelo con respecto a la

distancia real en cada punto de la cancha, el ajuste de los modelos para cada Dispositivo final se aprecia en la Figura 266 .

Figura 26. Ajuste de los modelos de una sola pendiente para cada módulo XBee.

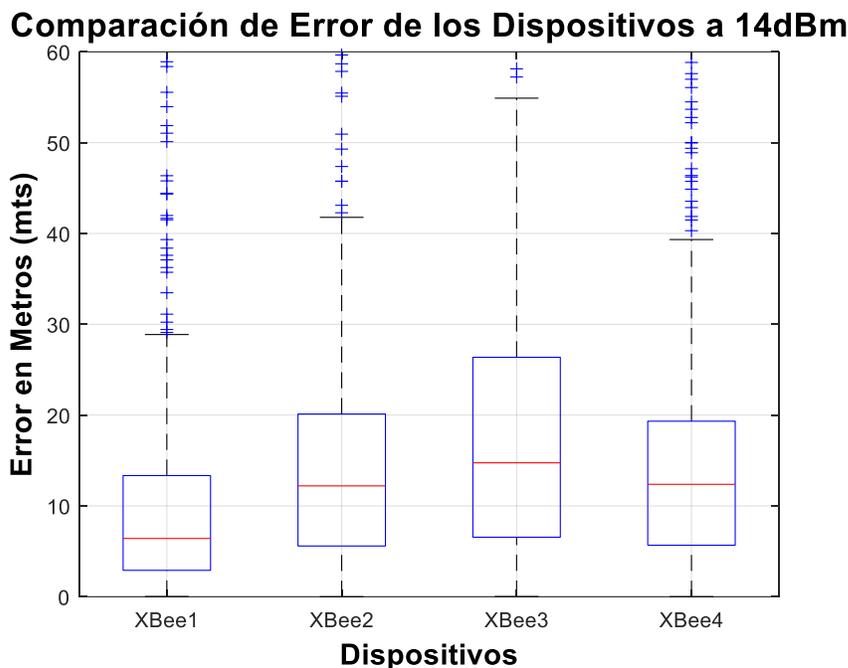


Una vez ajustados los modelos, se tienen los valores del parámetro n e intercepto para cada uno de los XBee, este ajuste corresponde a una regresión lineal y brinda las fórmulas necesarias para realizar el cálculo de la distancia entre dispositivos finales y el coordinador, una descripción más detallada del procedimiento se puede encontrar en la sección 4.3.1.

Posteriormente, se puede verificar el error de las distancias ajustadas por el modelo con respecto a las distancias reales. Dado que cada dispositivo final tiene un ajuste distinto, es necesario considerar el error en la distancia entre cada uno y el dispositivo coordinador. La

representación del error ayuda a determinar la precisión de la medida entre cada dispositivo final y el coordinador. Esto se hace con el fin de minimizar el error y obtener la mayor precisión posible para realizar la trilateración. En la Figura 277, se presenta el error de cada dispositivo.

Figura 27. Comparación de error del ajuste del modelo en la aproximación de la distancia en cada dispositivo a 14dBm



En la imagen anterior, se aprecian los errores que presenta el ajuste del modelo al calcular la distancia entre cada dispositivo final y el dispositivo coordinador, además, se observa que los errores de todos los XBee no son aceptables para realizar la trilateración en el campo. El XBee 3 con el mayor error tiene una mediana de 15 metros, mientras que el XBee 1 con el menor error tiene una mediana de aproximadamente 7 metros. Es notable que los cuartiles superiores y los valores atípicos destacan considerablemente, indicando una dispersión amplia y errores importantes. Estos no son favorables para el estudio en curso, ya que se evidencia una precisión insuficiente en la medición de las distancias por parte de los dispositivos XBee.

Debido a estos resultados, se decidió realizar un estudio incrementando la potencia de transmisión de los dispositivos al máximo. Esto se hizo con el fin de obtener mejores predicciones al ajustar los modelos de propagación, para que sean acordes con la aplicación que se desea llevar a cabo. Además, resultó necesario hacer una limpieza más exhaustiva de los datos, lo cual se explicará en la sección 4.3.1.2.

4.3.1.2 Segundo ajuste Modelo de Propagación en toda la Cancha #1 de la Universidad CESMAG.

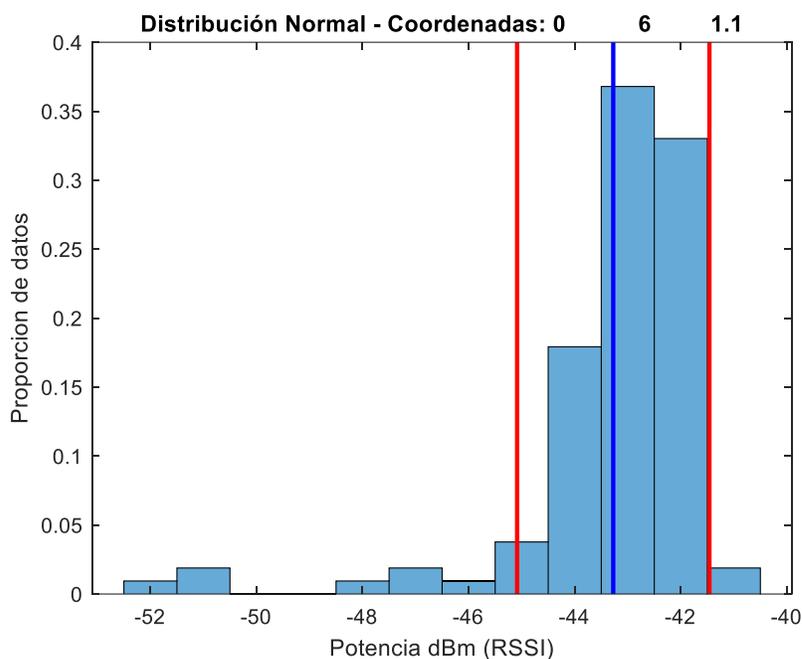
Para abordar el problema del estudio anterior a 14 dBm, los dispositivos se configuraron a la potencia más alta permitida por los dispositivos 18dBm, se utilizaron datos nuevos recopilados en la totalidad del espacio deportivo siguiendo los lineamientos del índice 4.3.1 . Posteriormente, se elaboró un histograma para cada coordenada, junto con sus datos de potencia. El objetivo de este análisis es determinar si el cálculo del promedio es la medida adecuada para la construcción del modelo de propagación y para abordar el error elevado que se observó usando potencias más bajas.

Si los datos mostraban una distribución no deseada y el cálculo del promedio no proporcionaba una respuesta lógica a la medida de potencia en cada coordenada, fue necesario explorar alternativas para procesar los datos de manera más efectiva. En caso de que se identificaran patrones irregulares o sesgos en ciertas coordenadas, fue necesario descartar esos datos específicos

Este enfoque de análisis más detallado permitió tomar decisiones informadas sobre el tratamiento de los datos, garantizando así la robustez y confiabilidad del modelo de propagación desarrollado.

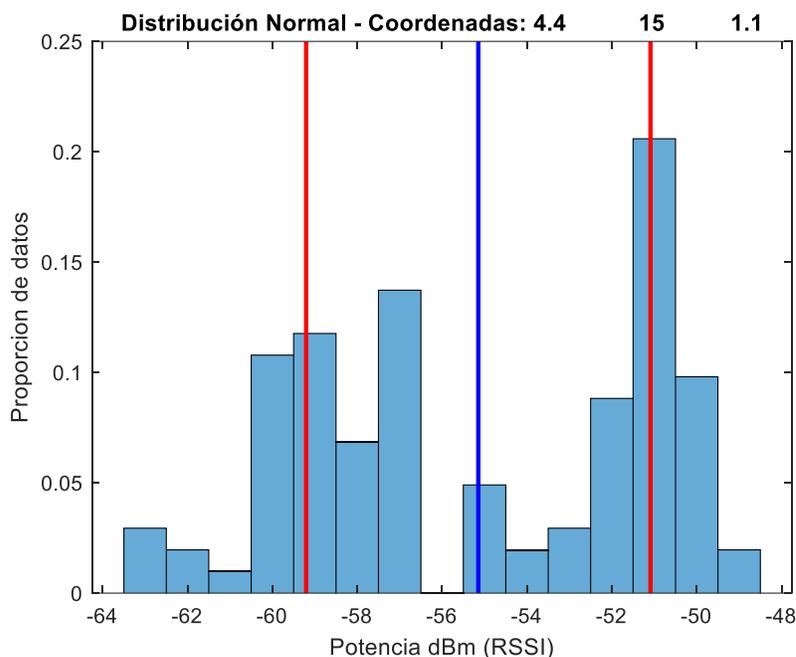
A continuación, en la Figura 28, se exhiben histogramas calculados en diversas coordenadas alrededor de la Cancha #1 de la Universidad CESMAG, Esta representación gráfica ofrece una visión detallada del comportamiento de los datos en distintos puntos, permitiendo la identificación de posibles patrones o tendencias.

Figura 28. Histograma de potencias para un punto de la Cancha #1 de la Universidad CESMAG.



Las líneas rojas en los histogramas representan la media más y menos la desviación estándar, mientras que la línea azul refleja el promedio de los datos. En la Figura 288 se aprecia, que la proporción de datos tiene una tendencia clara y la variación no es significativa, este tipo de lecturas indican que es posible tomar el promedio para realizar el ajuste del modelo.

Figura 29. Histograma de potencias no adecuadas para un punto en la Cancha #1 Unicesmag.



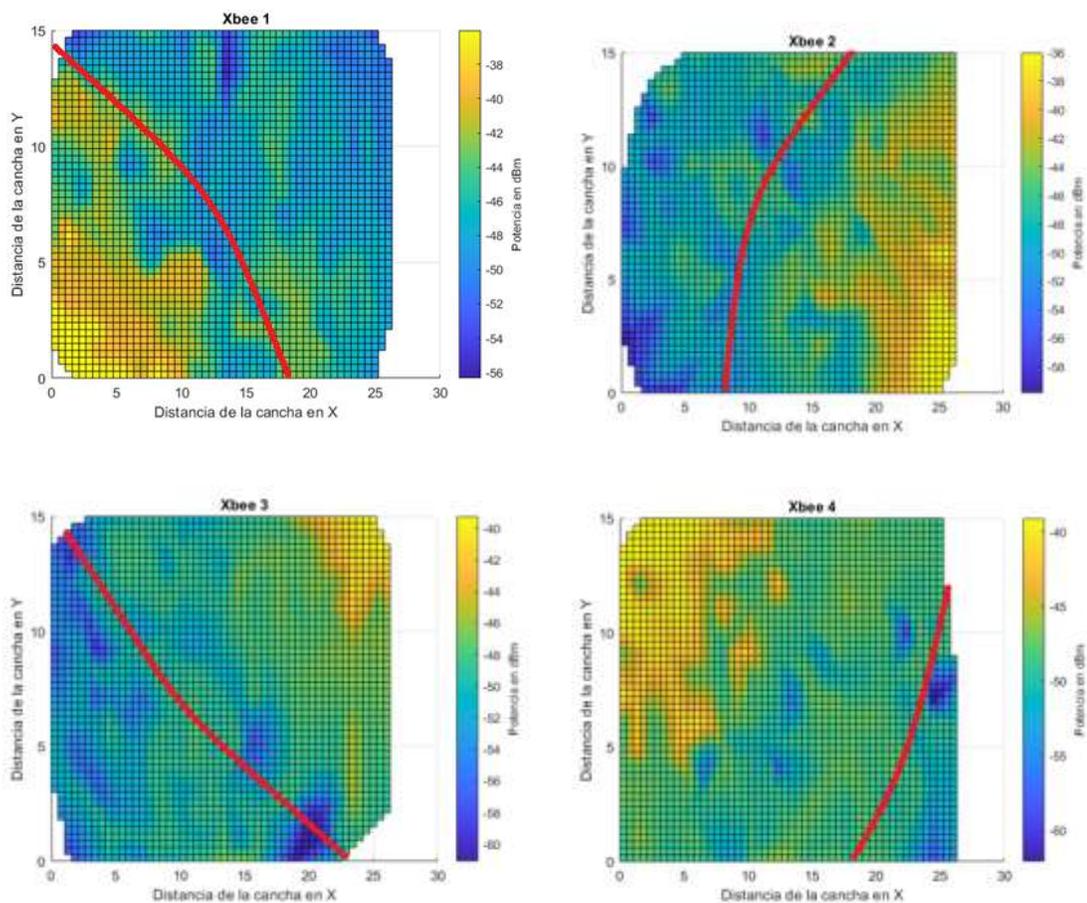
En la Figura 299, se observa una dispersión equitativa de los datos en diversas potencias, sin mostrar una tendencia clara. Esta falta de patrón definido plantea un desafío, ya que la ausencia de una dirección clara dificulta la construcción de un modelo predictivo coherente.

Una vez identificados los patrones encontrados en las lecturas de cada dispositivo final, es necesario conservar los datos que presentan comportamientos como los expuestos en la Figura 288, en la cual se aprecia que se puede tomar el promedio como una opción para ajustar el modelo de propagación, en cuanto a los puntos donde las lecturas de RSSI fueron iguales o similares a las de la Figura 299, se debieron eliminar las mediciones y la coordenada del conjunto de datos, esto se hizo con el fin de tener un ajuste del modelo más acorde y con unas mediciones que representen mejor el comportamiento en la cancha.

Después de filtrar el conjunto de datos en su totalidad, se utilizaron los datos para realizar un mapa de calor para cada dispositivo final, y así entender mejor las variaciones dentro de la Cancha #1 de la Universidad CESMAG. Este análisis preliminar facilitó la posterior aplicación

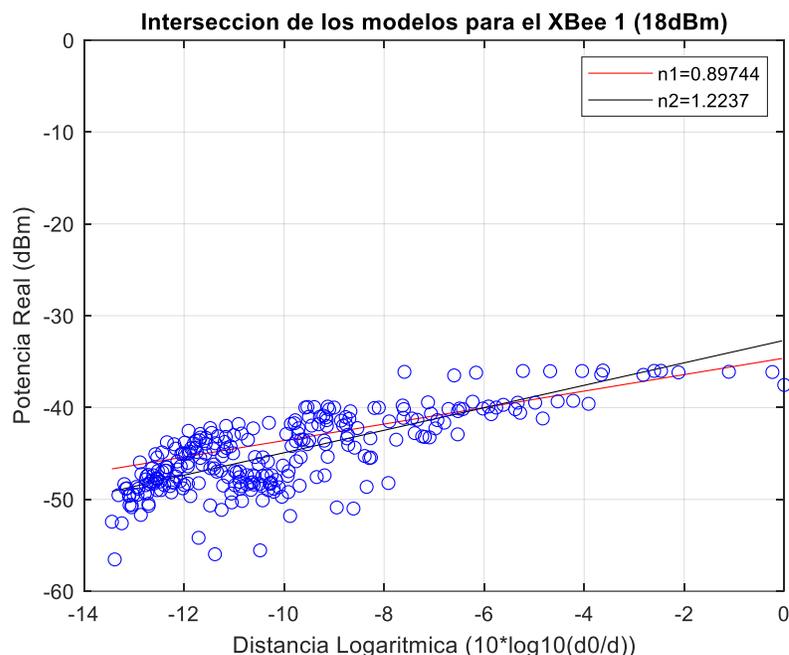
del modelo. A continuación, en la Figura 30, se presentan los mapas de calor para cada uno de los XBee.

Figura 30. Mapas de calor RSSI filtrados para los XBee a 18dBm



Una vez examinados los mapas de calor, se procedió a examinar la distribución de los datos para determinar qué modelo se ajustaría mejor a cada uno de los dispositivos. Debido a la dispersión similar de los datos se optó por aplicar un modelo de doble pendiente, llevando a cabo el procedimiento expuesto en el índice 4.3.1. A continuación, se puede apreciar el ajuste del modelo realizado para el XBee 1, debido a la similitud en la dispersión de los datos observada en los mapas de calor, se expondrá solamente el ajuste para el XBee 1.

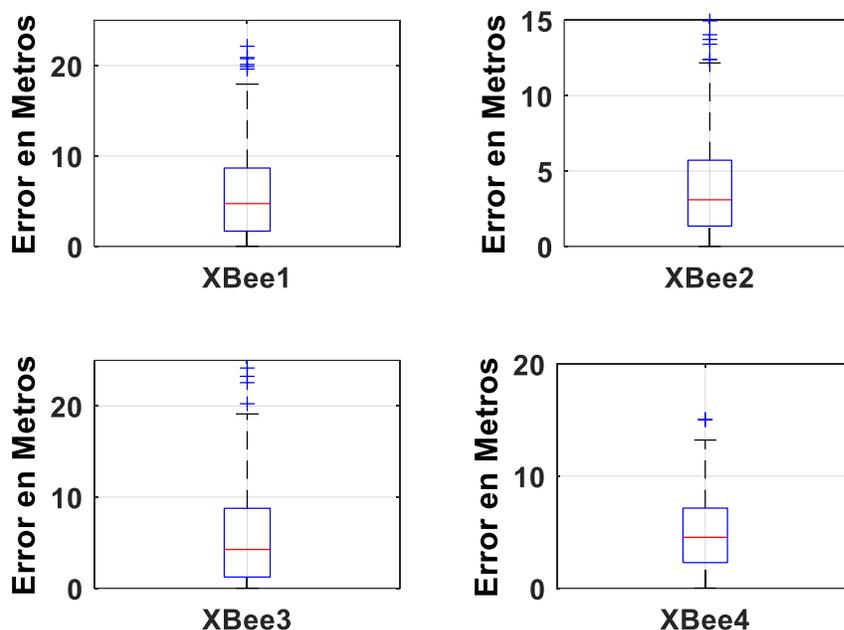
Figura 31. Intersección de los modelos para el XBee1 con datos filtrados a 18dBm



En la figura se observa que los datos se dividieron y se trazaron las pendientes para adaptarse mejor al conjunto completo de datos. Como resultado, se obtuvieron dos modelos con pendientes distintas que se interceptan. Esta intersección se determinó mediante un proceso de inspección visual, identificando el punto óptimo donde las pendientes se ajustan mejor al conjunto de datos. Dado que son dos modelos, esta intersección también ayudó a verificar a partir de qué punto se aplica uno u otro modelo. Para calcular la distancia más precisa, se probaron los resultados del modelo mediante un proceso de ensayo y error, evaluando si la aproximación de cada uno era adecuada. Los resultados de ajustar los modelos en cada dispositivo y determinar el error en las predicciones se puede observar en la Figura 322.

Figura 32. Comparación de error de los dispositivos en toda la Cancha #1 a 18dBm.

Comparación de Error de los Dispositivos a 18dBm

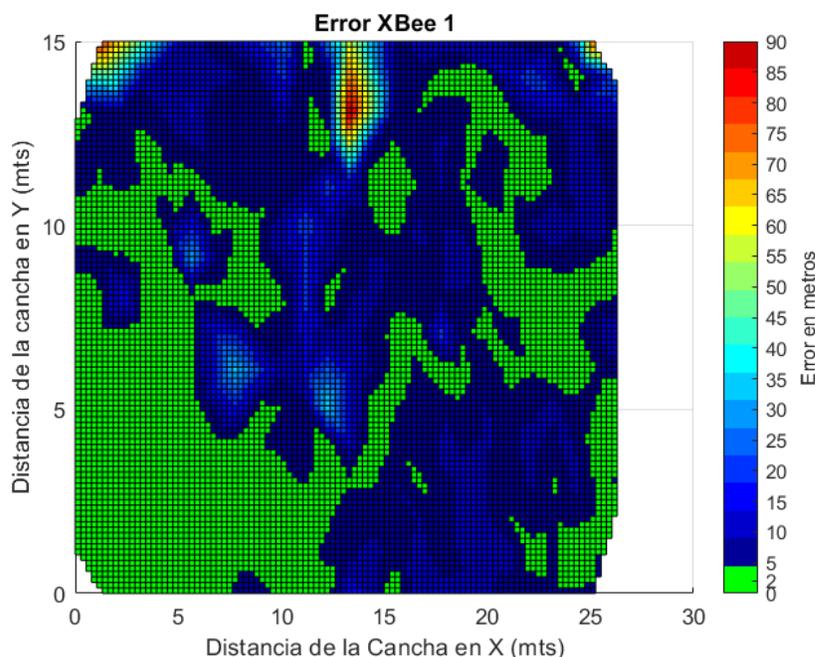


La comparación del error entre el dispositivo XBee coordinador y los Dispositivos finales a una potencia de 18 dBm muestra una mejora significativa. En este caso, la mediana de todos los XBee es inferior a 5 metros, con los cuartiles inferiores extremos cercanos a 0 y cuartiles superiores extremos que no superan los 20 metros. Además, se observa la presencia de valores atípicos que sobresalen en todos los dispositivos XBee.

Aunque se aprecia una mejora en la medida de la distancia para cada uno de los dispositivos finales, sigue siendo tendencia que el error más bajo se encuentra en las potencias más altas y distancias menores, de este modo es necesario tener una confirmación del comportamiento del error de los dispositivos dentro del espacio deportivo. Para esto se realizó un mapa de calor en el cual se puede visualizar donde se concentra el menor error y en qué áreas de operación el dispositivo es más preciso.

Figura 33. Representación del error del XBee1 en toda la Cancha #1 de la universidad

CESMAG

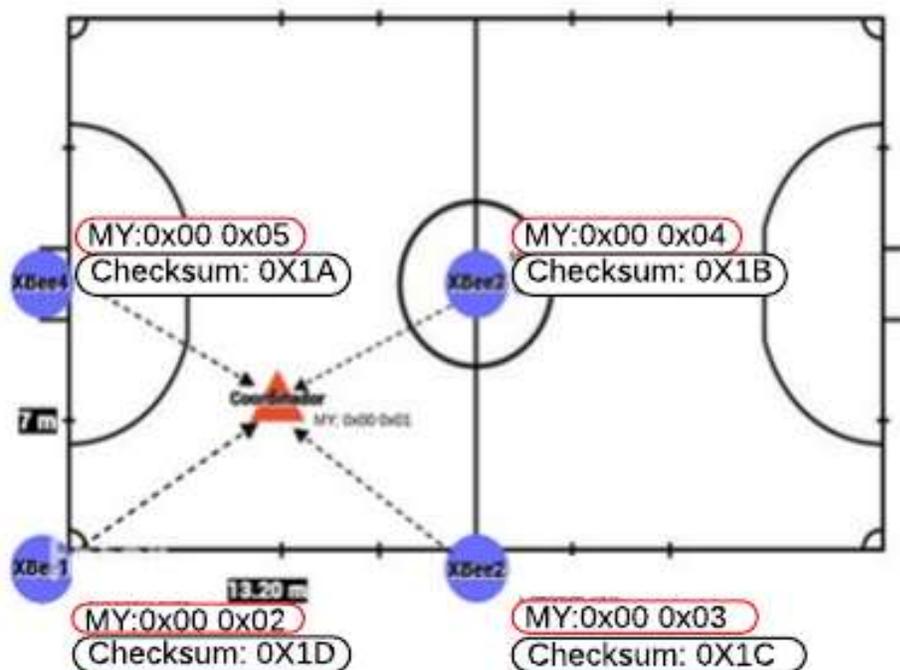


El mapa de calor indica que el error es más bajo en el espacio correspondiente a $\frac{1}{4}$ de la Cancha #1 Unicesmag. Con esto en mente es necesario realizar el ajuste de un modelo donde se utilice solo esa porción del campo.

4.3.2 Ajuste Modelo de Propagación en 1/4 de la Cancha #1 de la Universidad CESMAG.

Para desarrollar este estudio se ubicaron los dispositivos en $\frac{1}{4}$ de la cancha, se configuraron todos los dispositivos a la potencia máxima 18dBm, además, se realizó la toma de las medidas teniendo en cuenta la metodología utilizada en el inciso 4.3.1. También se realizó el tratamiento de datos utilizado para el estudio en la totalidad de la cancha a 18 dBm expuesto en el inciso 4.3.1.2. A continuación, en la Figura 344 se puede observar la ubicación de los dispositivos en la Cancha #1 de la Universidad CESMAG.

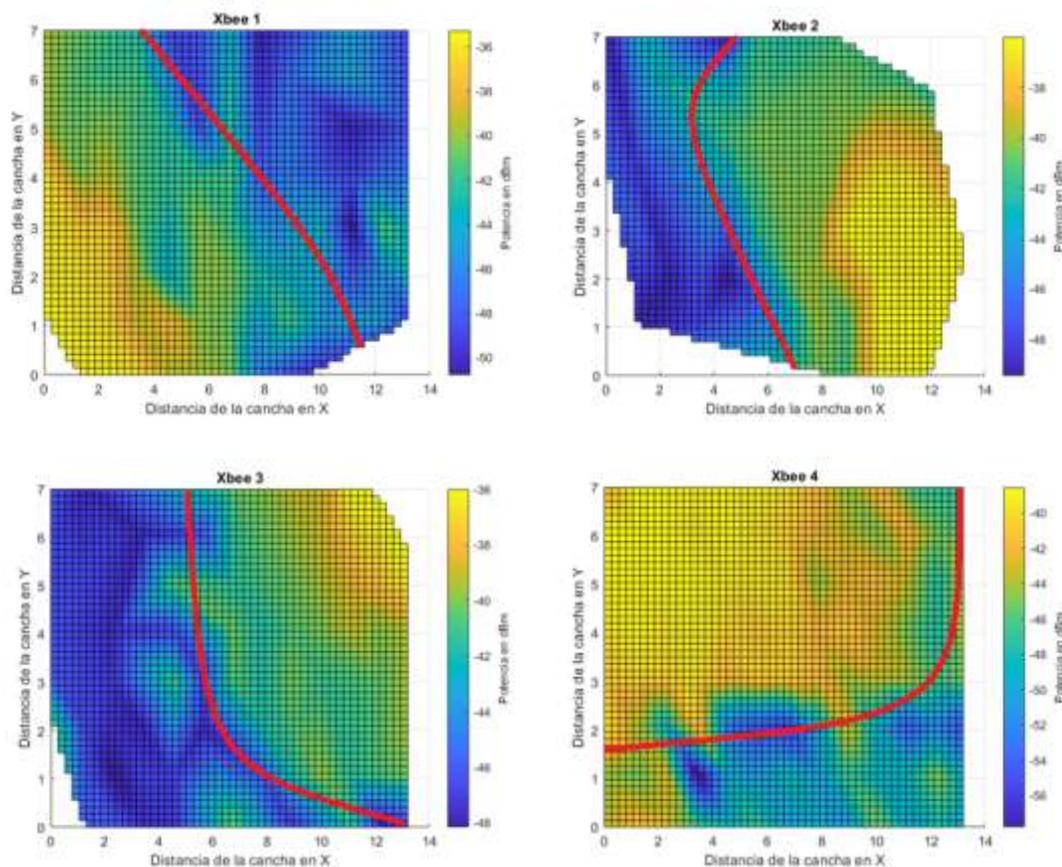
Figura 34. Distribución de los dispositivos en 1/4 de la Cancha #1 de la Universidad CESMAG.



Para este estudio se tomaron 104 puntos en total en el espacio delimitado en la cancha, los datos recopilados en esta porción de la cancha fueron filtrados teniendo en cuenta la distribución de los datos, al igual que se hizo en el estudio de la sección 4.3.1.2, se tomaron los datos en las coordenadas en las cuales el promedio representaba la totalidad de los datos recolectados y los datos que tenían demasiada variabilidad se descartaron y se realizó el ajuste del modelo con las coordenadas restantes.

Para entender mejor el comportamiento de los datos en la mitad de la Cancha #1 Unicesmag, en la Figura 355, se presentan los mapas de calor correspondiente a los datos utilizados para realizar el ajuste de modelo en cada dispositivo.

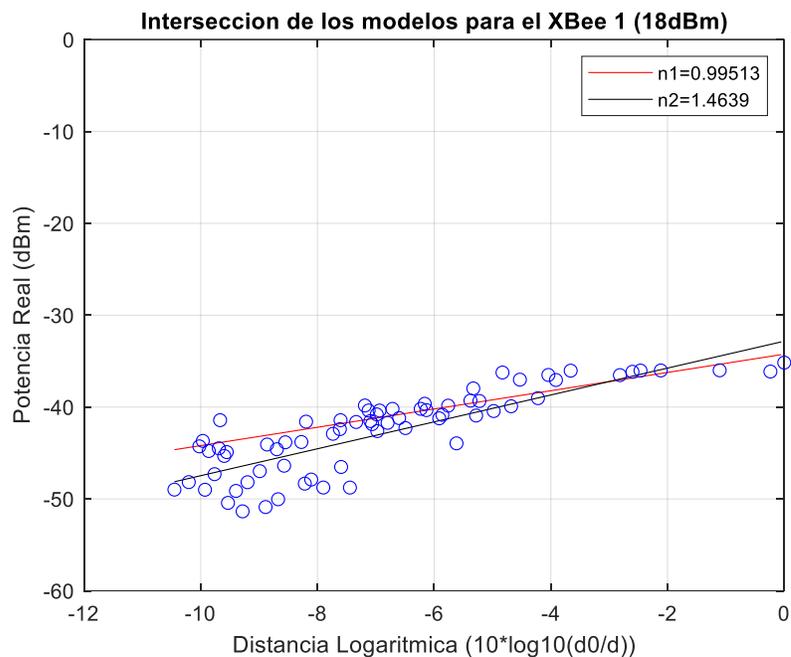
Figura 35. Mapa de calor de los módulos XBee en $\frac{1}{4}$ de Cancha #1 Unicesmag, con datos filtrados (18dBm).



A partir de los mapas de calor obtenidos, se decidió utilizar exclusivamente los dispositivos XBee 1, 2 y 3. Esta elección se fundamentó en que el XBee 4, debido al entorno en el que estaba ubicado dentro de la Cancha #1 de la Universidad CESMAG, mostraba una variación anómala en los datos de RSSI. Aunque el XBee 4 se movía en los ejes X e Y igual que los otros dispositivos, los valores de potencia no variaban como se esperaba e incluso se repetían a pesar de la distancia respecto al coordinador. Esto dificultaba la obtención de un modelo de estimación de distancia con la definición necesaria para el estudio. Considerando esta situación, se procedió a ajustar el modelo de propagación para cada uno de los dispositivos XBee

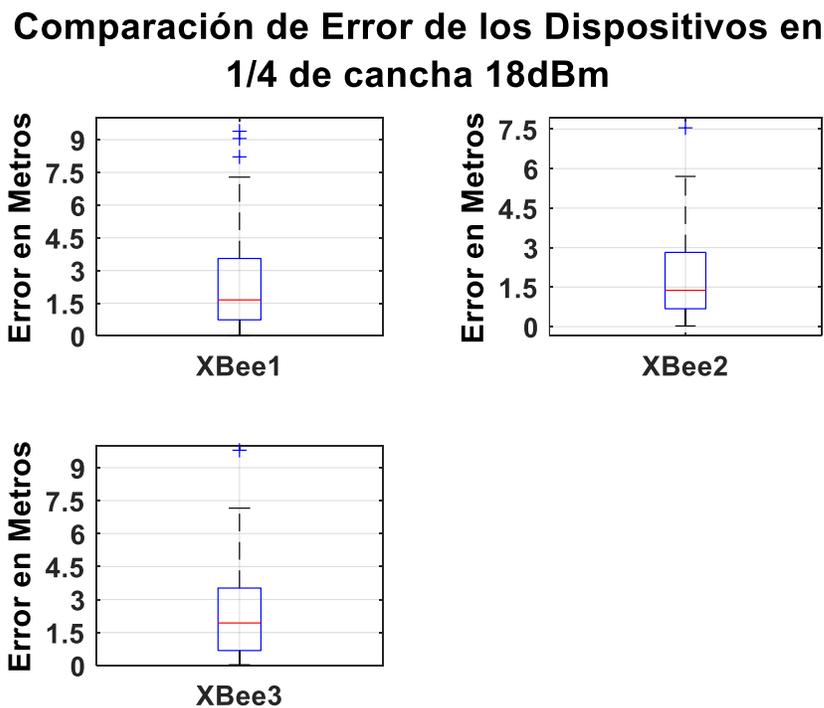
mencionados anteriormente. La Figura 366 ilustra el ajuste del modelo realizado en dichos dispositivos.

Figura 36. Representación de los ajustes del modelo para los dispositivos XBee



En la figura se observa que nuevamente se ajustaron dos modelos para aproximarse de la mejor manera posible a la totalidad de los datos. Del mismo modo, una vez obtenidos los valores de distancia necesarios, calculados a partir del ajuste del modelo, se generó la gráfica de error correspondiente. Esta herramienta permitió tomar decisiones respecto a cada uno de los dispositivos.

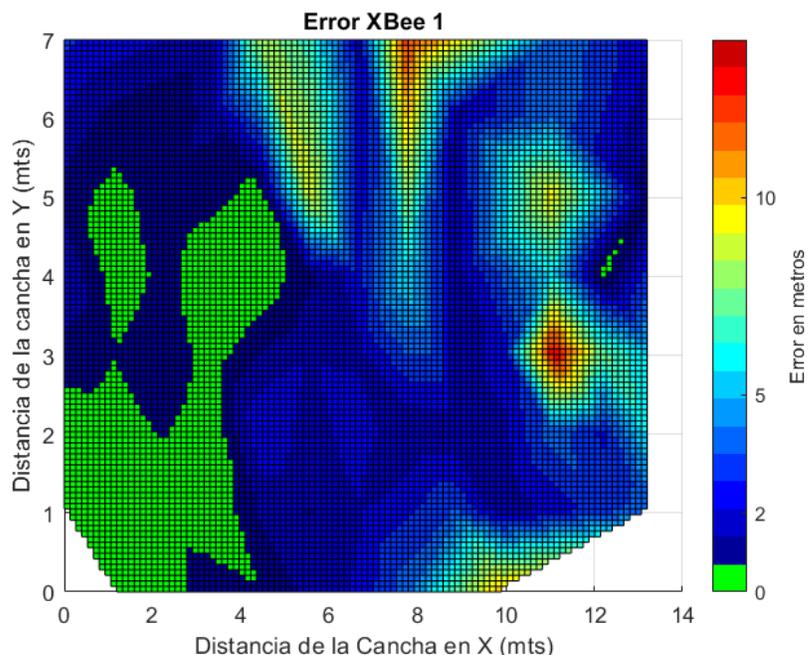
Figura 37. Comparación de error de los dispositivos en 1/4 de Cancha #1 Unicesmag, a 18dBm



A medida que la distancia entre los dispositivos XBee se reduce, el error en metros de cada dispositivo disminuye. En este escenario, la mediana de los errores en los diagramas se aproxima más a cero, lo que sugiere un procedimiento eficaz del modelo de propagación. también se observa una disminución en el número de valores atípicos en este caso.

Basándose en este resultado y utilizando el XBee1 como punto de referencia, se elaboró un mapa de calor del error en la distancia arrojado por el modelo de propagación en este dispositivo. Esto proporcionará una visión del comportamiento del error y, además, información para avanzar hacia la construcción de un modelo que reduzca el error en la estimación de la posición.

Figura 38. Mapa de calor del error del XBee1 en 1/4 de cancha (18dBm)



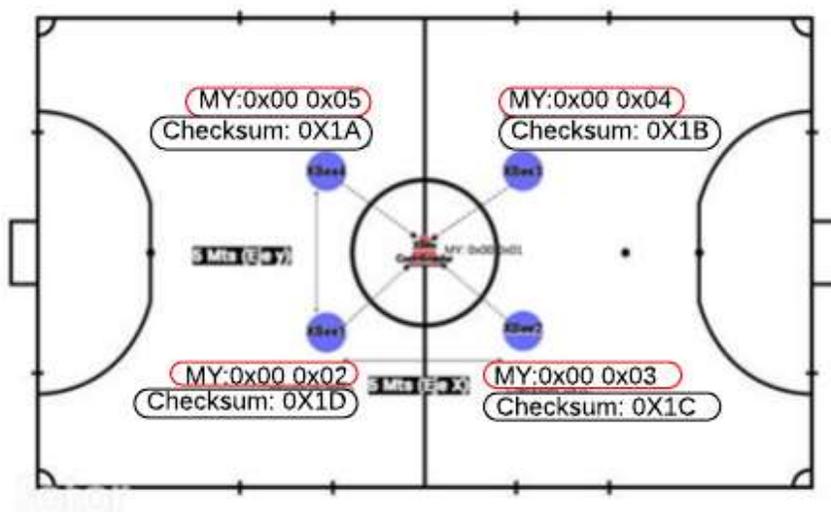
En la figura se observa que el error en 1/4 del espacio deportivo, en la totalidad de las coordenadas, se reduce. Sin embargo, a pesar de este resultado, no es factible realizar una trilateración para obtener las coordenadas X e Y estimadas por el modelo. Esto se debe a que en las distancias más alejadas del XBee1, existen errores de entre 5 y 10 metros, lo que resultaría en una alta imprecisión al obtener las coordenadas predichas. Por este motivo, y tras analizar nuevamente el mapa de calor, se tomó la decisión de ajustar un modelo para un espacio de 5 metros en el eje X y 5 metros en el eje Y.

4.3.3 Ajuste Modelo de Propagación en un espacio 5 x 5 mts en la Cancha #1 de la Universidad CESMAG.

Para este estudio los dispositivos se configuraron a una potencia media de 14dBm debido a que esta proporciona una mejor variabilidad en los datos. La decisión de llevar a cabo el estudio en un espacio de 25 metros cuadrados se basó en los resultados obtenidos en los estudios

previos, Además, la elección de ubicar el estudio en el centro del espacio deportivo se fundamentó en los hallazgos del estudio referente al índice 4.3.2, donde se observó un alto margen de error en los datos del XBee 4. Esto llevó a la conclusión de que el entorno ejerce una notable influencia en la percepción de la potencia RSSI por parte de los dispositivos. Por consiguiente, se determinó la disposición de los XBee como lo muestra la Figura 399.

Figura 39. Ubicación de los Dispositivos XBee en un espacio de 5 x 5 mts en la Cancha #1 de la Universidad CESMAG.



Se recopilaban datos de forma sistemática, tomando mediciones separadas por un metro en cada eje dentro del espacio designado para el estudio. En total, se recolectaron datos en 36 coordenadas del área de interés. Además, se consideró el tiempo requerido para adquirir cada muestra, estableciendo un valor de 50 milisegundos por cada medición con el objetivo de reducir la variabilidad en los datos. Sin embargo, debido al aumento en el tiempo de adquisición, se limitó el número de mediciones a 100, ya que tomar más muestras requería un tiempo considerable.

La ubicación de los Dispositivos XBee en un espacio de 5x5 metros dentro del escenario de la Cancha #1 de la Universidad CESMAG donde se realizó las mediciones de RSSI se puede apreciar en la Figura 4040, en donde como limitaciones se tiene que los dispositivos finales y el XBee coordinador se encuentran casi a la misma altura, con el fin de obtener datos fiables y así una mejor estimación de la precisión de los mismos.

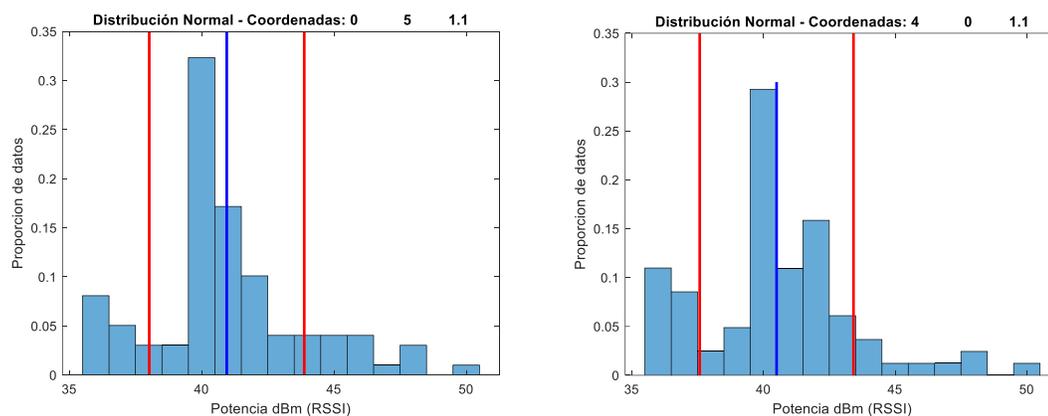
Figura 40. Escenario de 5x5 metros de la Cancha #1 de la Universidad CESMAG



El tiempo estimado para las mediciones de RSSI en el escenario, presentado anteriormente, requirió de 4 horas, para la totalidad de la toma de datos.

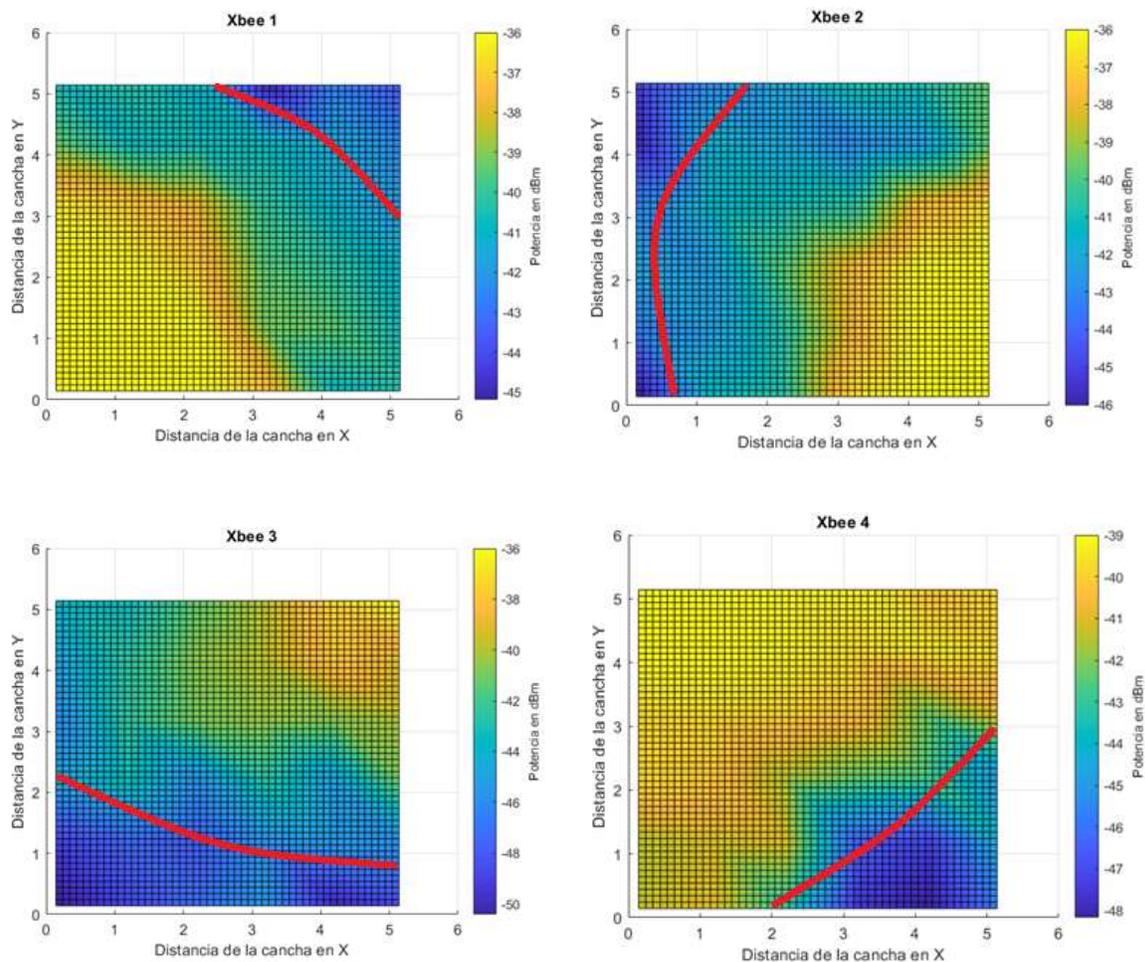
Una vez obtenidos los valores de RSSI, se llevó a cabo una inspección visual de los datos para su análisis. En este caso particular, los datos no mostraron una variación significativa, por lo que se optó por calcular el promedio de los datos del estudio. Posteriormente, se utilizó la desviación estándar para separar los datos que se encontraban dentro de este rango y se procedió con el ajuste del modelo. La disposición de los datos de estudio se presenta en la Figura 411.

Figura 41. Distribución de los datos obtenidos en coordenadas aleatorias.



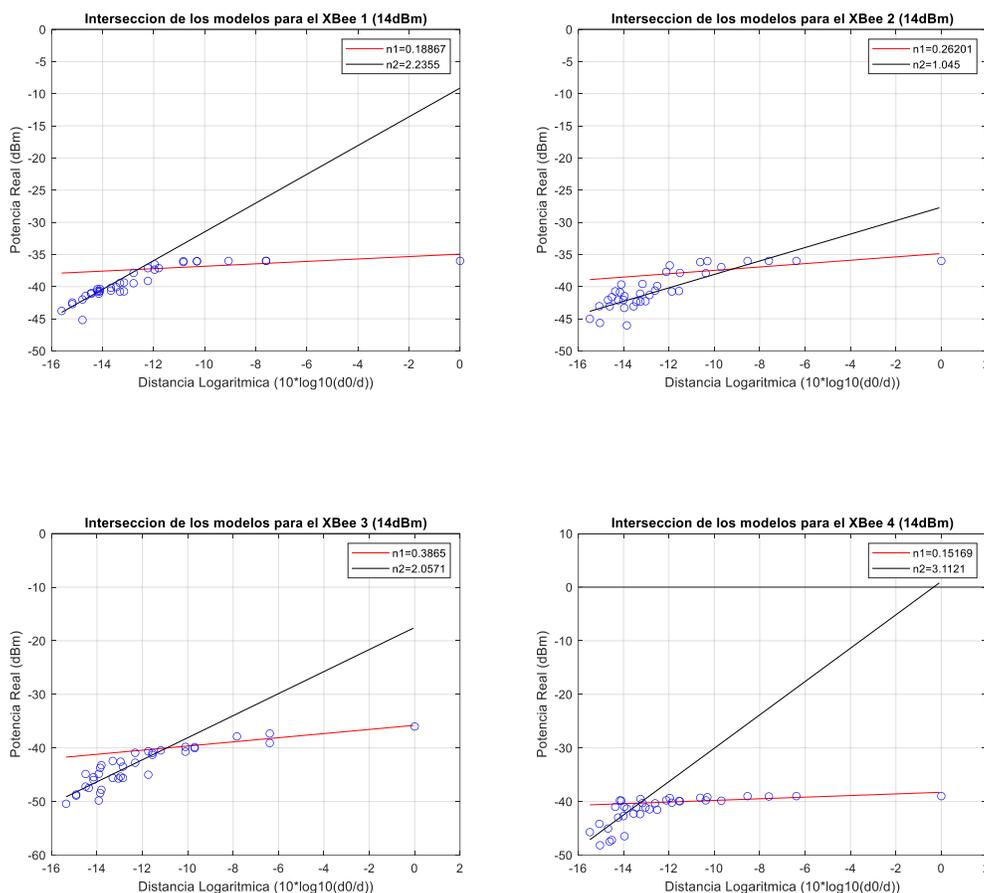
En la figura anterior, se muestra la desviación estándar representada mediante líneas rojas y el promedio mediante una línea central azul, solo se consideraron los datos que se encuentran entre las líneas rojas que representan la desviación estándar, a partir de esos datos se hizo el cálculo del promedio. Una vez calculados los promedios necesarios, se procedió a representarlos en mapas de calor para observar su comportamiento, esto se puede apreciar en la Figura 422.

Figura 42. Representación de los valores RSSI en dispositivos XBee usando el promedio \pm desviación estándar.



Dado que el comportamiento del parámetro RSSI en todos los dispositivos mostraba una variación significativa y la inspección visual de la Figura 42. Representación de los valores RSSI en dispositivos XBee usando el promedio \pm desviación estándar., sugirió que todos los datos de los dispositivos eran adecuados para realizar el ajuste del modelo, se procedió con el ajuste y se obtuvieron los resultados que se muestran en la Figura 433.

Figura 43. Ajuste de los modelos de doble pendiente para los dispositivos XBee.

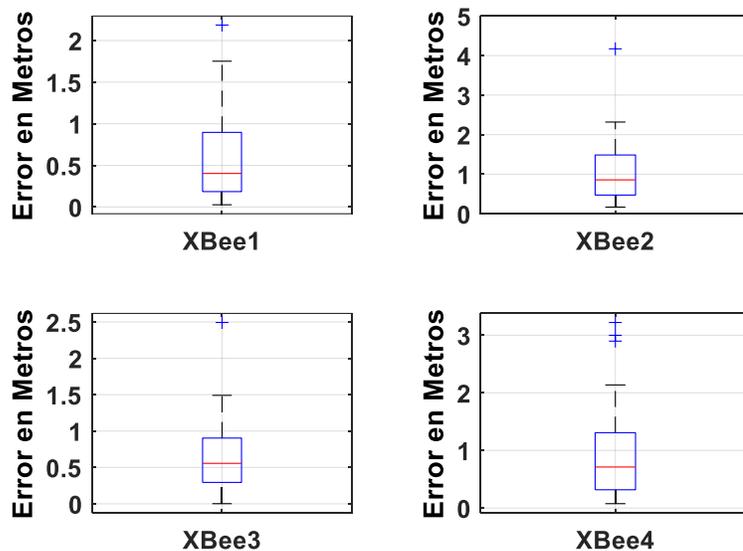


Para cubrir todos los datos y obtener una predicción aceptable sobre la distancia de los dispositivos finales al coordinador, se optó por ajustar modelos de doble pendiente. Este enfoque consiste en dividir el conjunto de datos en dos segmentos distintos, cada uno con su propia pendiente o tasa de variación de la potencia de la señal recibida (RSSI) con respecto a la distancia. La primera pendiente se ajusta a los datos más cercanos al coordinador, donde la señal suele ser más fuerte y menos afectada por obstáculos. La segunda pendiente se aplica a los datos más alejados, donde la señal tiende a debilitarse más rápidamente y puede ser más susceptible a interferencias y atenuación. Al usar dos pendientes, el modelo puede adaptarse mejor a las diferentes características del entorno y proporcionar estimaciones de distancia más precisas a lo largo de todo el rango de medición. Este ajuste permite capturar mejor la variabilidad en la

potencia de la señal y mejorar la precisión de los dispositivos finales. Los resultados obtenidos en el cálculo de la distancia se pueden apreciar en la Figura 444

Figura 44. Comparación del error de los dispositivos con el ajuste del modelo, en un espacio de 5x5 mts en la Cancha #1 de la Universidad CESMAG.

Comparación de Error de los Dispositivos a 14dBm 5x5 Metros



En este caso, se observa que los errores a una distancia de 5x5 metros en todos los dispositivos son menores a 2,5 metros, arrojado por el XBee2 siendo este el que presenta el mayor rango de error, teniendo en cuenta el cuartil extremo superior de este.

La distribución de la mediana en todos los dispositivos muestra que el error se encuentra a menos de 1 metro. Además, el cuartil extremo inferior de todos los dispositivos se aproxima a cero y el cuartil extremo superior está por debajo de 2,5 metros. Esto da a entender que existe un rango de error entre 2,5 y 0 metros para todos los dispositivos, Sin embargo, sigue habiendo presencia de valores atípicos en todos los dispositivos XBee.

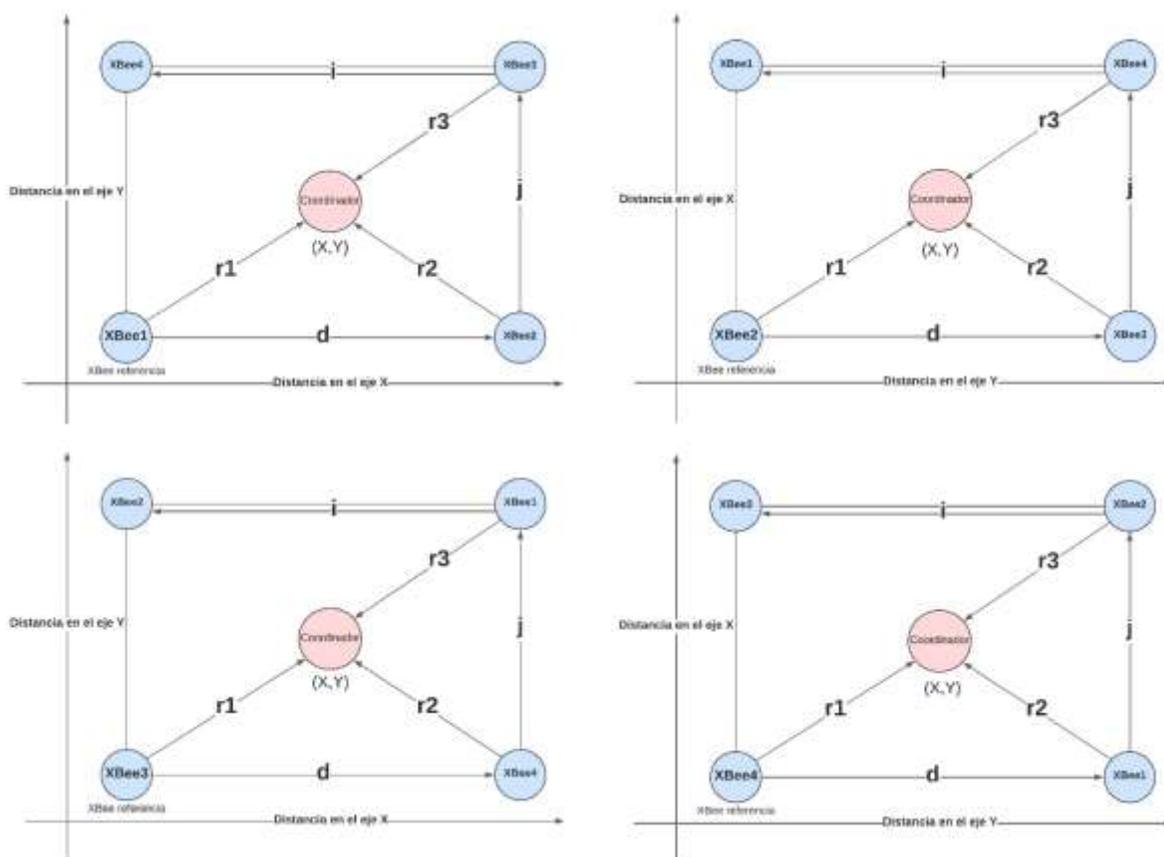
Con ayuda de los resultados obtenidos en el estudio, se decidió llevar a cabo la trilateración. Teniendo en cuenta que se habían considerado tanto las distancias entre

dispositivos como los ajustes de potencia, se llegó a la conclusión de que esta era la mejor aproximación que un modelo ajustado podría proporcionar para estos dispositivos.

4.3.4 Trilateración y obtención de las coordenadas con el ajuste del modelo Log Distancia en un espacio de 5 x 5 mts en el espacio deportivo.

Para realizar la predicción de las coordenadas finales (X, Y) a partir de las distancias obtenidas mediante el modelo de propagación, fue necesario adaptar el cálculo de la trilateración. Se realizaron combinaciones de tres dispositivos para determinar la combinación que presentara el menor error. Se descartó la opción de utilizar los cuatro dispositivos debido a que cada uno de ellos tiene valores de error que podrían afectar la precisión de la predicción de las coordenadas. Considerando esto, se optó por buscar la combinación que minimizara el error. Las combinaciones y disposiciones de los ejes para cada una se presentan en la Figura 45.

Figura 45. Combinaciones para la trilateración utilizadas en los dispositivos XBee.



En la figura anterior, se puede observar que, para cada combinación, tanto los ejes como la disposición de los dispositivos varían. Para realizar el cálculo de la trilateración, es crucial entender que siempre uno de los dispositivos XBee debe estar en el mismo eje que el XBee de referencia. Esto asegura que las fórmulas para el cálculo de las coordenadas puedan aplicarse correctamente. Para este caso de estudio, tanto el dispositivo en la misma línea del XBee de referencia como el tercer dispositivo involucrado en la trilateración se encuentran a la misma distancia en los ejes i y j . Es importante destacar que estas distancias pueden variar dependiendo de la disposición de los dispositivos que se desee adoptar. A continuación, se presentan las fórmulas implicadas en el cálculo de las coordenadas. Antes de realizar cualquier predicción, se

realizaron pruebas de las fórmulas con los valores reales calculados trigonométricamente para confirmar la veracidad de la técnica (Moreno, Benítez, López, Medina, & De la Calleja, 2012).

$$x = \frac{r_1^2 - r_2^2 + d^2}{2*d} \quad (14)$$

$$y = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2*j} - \frac{i}{j} * x \quad (15)$$

Donde

r_1 = Distancia del XBee referencia al coordinador.

r_2 = Distancia del XBee en el mismo eje del XBee referencia al coordinador.

r_3 = Distancia del tercer XBee al coordinador.

d = Distancia del XBee de referencia al XBee en su mismo eje.

i = Distancia paralela del tercer XBee al XBee Referencia.

j = Distancia paralela del XBee referencia al tercer XBee.

Dado que las combinaciones realizadas para el cálculo de las coordenadas giran los dispositivos en torno a sus ejes originales, las coordenadas calculadas se ven alteradas. Por lo tanto, es necesario realizar un ajuste al final para no alterar las fórmulas destinadas al cálculo. Los ajustes realizados se presentan en la Tabla 10.

Tabla 10. Ajuste en el cálculo de las coordenadas, considerando los XBee referencia.

En la tabla el número 5 corresponde a la distancia en <i>mts</i> en cada eje.	XBee 1 Referencia	XBee 2 Referencia	XBee 3 Referencia	XBee 4 Referencia
Modificación en X	No aplica	5- y	5- x	y
Modificación en Y	No aplica	x	5 - y	5 - x

Una vez realizado el ajuste de las coordenadas para todos los dispositivos se procedió a realizar el cálculo del error en la predicción de las coordenadas haciendo uso de la siguiente formula.

$$Error_{Trilateración} = \sqrt{(x'| - x)^2 + (y'| - y)^2} \quad (16)$$

Donde

x' = Valor real de la coordenada X.

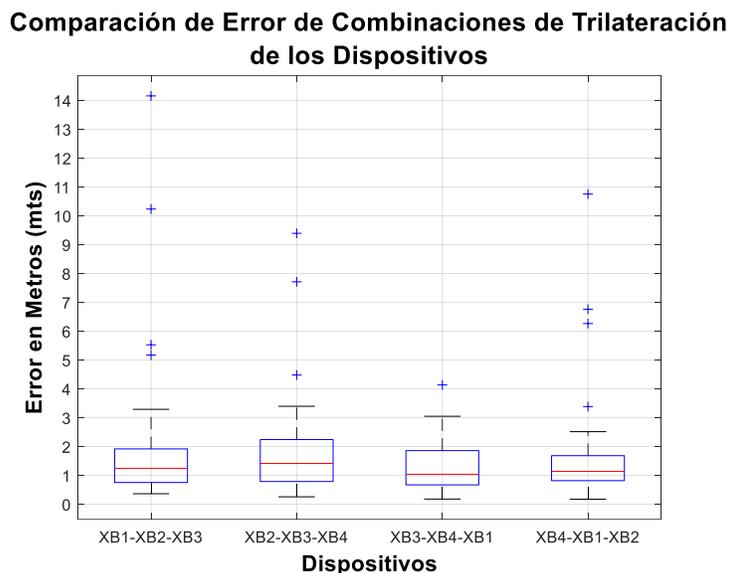
x = Valor calculado con el ajuste del modelo de la coordenada X.

y' = Valor real de la coordenada Y.

y = Valor calculado con el ajuste del modelo de la coordenada Y.

El cálculo anterior debió realizarse para cada una de las combinaciones efectuadas en la trilateración. Esto con el fin de determinar cuál de las combinaciones tenía el menor error en la predicción de las coordenadas. Los resultados se pueden apreciar en la Figura 466.

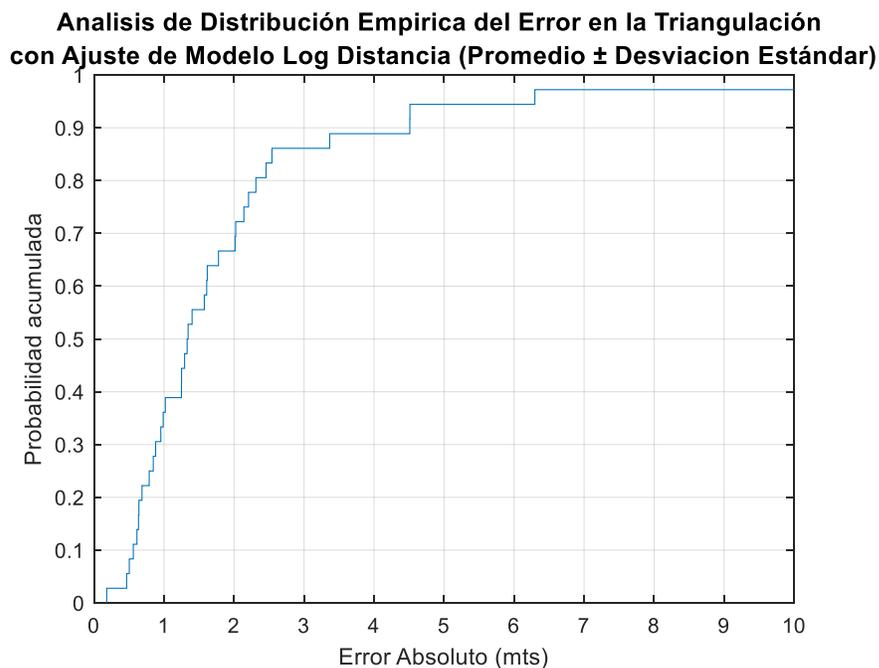
Figura 46. Comparación de error de combinación de Trilateración de los dispositivos



Para la implementación de la trilateración, se utilizaron distintos enfoques, empleando 3 dispositivos para cada combinación. Se observa que las combinaciones con menor error son las que tienen a los dispositivos XBee 3 y 4 de referencia. En estas combinaciones, la mediana del error se sitúa alrededor de 1 metro, mientras que los cuartiles superiores están por debajo de los 2 metros de error. Esto sugiere que, a medida que la distancia entre los dispositivos se reduce, también disminuye el error en los modelos de propagación empleados.

Con base en la Figura 46, se determina que la mejor combinación para el cálculo de las coordenadas en esa sección de la Cancha # 1 de la Universidad CESMAG. es aquella que tiene como dispositivo de referencia al XBee 4. Finalmente, se procede a determinar el porcentaje de las aproximaciones realizadas por el modelo que se encuentran en ciertos rangos de error. Para esto, se considera la Figura 47, donde se indica el comportamiento del error y su probabilidad con respecto a los resultados.

Figura 47. Análisis del error en las predicciones con XBee 4 como referencia y el modelo de propagación ajustado.



En la imagen anterior, se observa que el comportamiento del error, en relación con su frecuencia de ocurrencia dentro de las predicciones realizadas, tiene un error por debajo de los 2 metros en al menos el 65 por ciento de las ocurrencias. Esto indica que el modelo es eficaz en comparación con otros sistemas de localización, como el GPS que posee un margen de error de 4,9 metros (Khan SZ, Mohsin M, Iqbal W. (2021)).

4.4 Ajuste del modelo de Machine Learning en un espacio de 5 x 5 mts de la Cancha #1 de la Universidad CESMAG.

Para ajustar el modelo de Machine Learning, se utilizó Python junto con la librería Keras para el aprendizaje automático. Se implementó una red neuronal con cuatro entradas, correspondientes a las potencias RSSI adquiridas. Además, se incluyeron tres capas ocultas

encargadas de predecir las coordenadas X e Y. Para el entrenamiento del modelo, se utilizó el 80 por ciento de los datos de cada coordenada recopilados anteriormente.

La creación de la red neuronal estuvo sujeta a las recomendaciones de Basogain (1998), quien en su libro resalta que la neurona artificial se diseñó para imitar las características fundamentales del funcionamiento de una neurona biológica. En su esencia, esta neurona recibe un conjunto de entradas, donde cada una representa la salida de otra neurona. Cada entrada se multiplica por su respectivo "peso" o ponderación, que simula el grado de conexión de la red neuronal. Debido a esto se tiene como referencia que la capa oculta inmersa en la red neuronal debe tener al menos el doble de la capa de entrada.

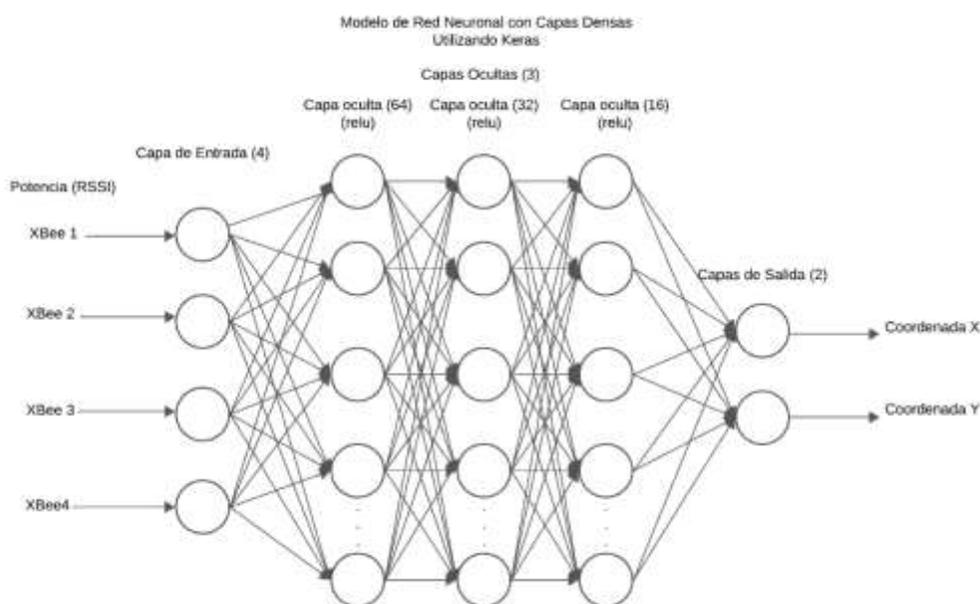
Según Tablada (2009), las entradas de una red neuronal artificial son el estímulo que reciben las neuronas en las capas ocultas para determinar el peso de la decisión a ser tomada y pasada finalmente como respuesta, también se hace énfasis en que una red neuronal puede o no tener capas ocultas, ya que estas se determinan según el estudio y pueden ser modificados y adaptados para realizar distintas tareas.

Teniendo en cuenta esta premisa, se optó por diseñar una red neuronal densamente conectada, donde todas las capas de entrada, ocultas y de salida están interconectadas. Para lograrlo, se empleó un enfoque iterativo: se comenzó con tres entradas en las capas ocultas como punto de partida, y luego se redujo a la mitad el número de entradas en cada capa oculta subsiguiente, hasta alcanzar el número adecuado de capas para la predicción deseada (Basogain, 1998).

Para activar las capas ocultas de la red neuronal, se consideraron dos tipos de funciones de activación ampliamente utilizadas en la actualidad: la función sigmoide y la función de tangente hiperbólica. En el caso de la función sigmoide, es necesario normalizar los datos, ya que

sus entradas deben estar en el rango de 0 a 1 para realizar predicciones. Por otro lado, la función de tangente hiperbólica tiene un rango más amplio, lo que permite entrenar con los valores RSSI sin necesidad de normalización (Basogain, 1998).

Figura 48. Modelo de red neuronal para la predicción de coordenadas X e Y.



Para alcanzar el ajuste adecuado del modelo que generó las mejores predicciones, fue necesario evaluarlo con diferentes parámetros. En la Tabla 11 se resumen los resultados de los modelos de Machine Learning desarrollados para predecir coordenadas. El propósito de esta tabla es proporcionar una visión clara y concisa de los parámetros clave utilizados en los modelos, así como de su desempeño respectivo. Permite una comparación entre diferentes configuraciones de modelos, resaltando el modelo 10 el cual mostró el mejor rendimiento. Además, facilita la interpretación de los resultados al mostrar variables importantes a considerar durante el entrenamiento de la red neuronal.

Tabla 11. Entrenamientos realizados para obtener las mejores predicciones en el algoritmo de Machine Learning.

Modelo	Numero de capas ocultas	Tipo de conexión entre capas	Numero de neuronas por capa	Funcion de Activación	Optimizador	Numero de ciclos de entrenamiento	Mejor Perdida del modelo (Loss)	Mejor Coeficiente de determinación (R2 Score)
1	1	Dense	8	ReLU	Adam	100, 150, 200	1.434	0.513
2	1	Dense	16	ReLU	Adam	100, 150, 200	1.405	0.523
3	1	Dense	32	ReLU	Adam	100, 150, 200	1.209	0.540
4	1	Dense	64	ReLU	Adam	100, 150, 200	1.181	0.568
5	2	Dense	64, 8	ReLU	Adam	100, 150, 200	1.959	0.335
6	2	Dense	64, 16	ReLU	Adam	100, 150, 200	1.254	0.564
7	2	Dense	64, 32	ReLU	Adam	100, 150, 200	1.146	0.583
8	2	Dense	64, 64	ReLU	Adam	100, 150, 200	1.226	0.562
9	3	Dense	64, 32, 8	ReLU	Adam	100, 150, 200	1.116	0.621
10	3	Dense	64, 32, 16	ReLU	Adam	100, 150, 200	0.978	0.692
11	5	Dense	128, 64, 32, 16, 8	ReLU	Adam	50, 100, 150, 200	1.037	0.647
12	4	Dense	64, 32, 16, 8	ReLU	Adam	50, 100, 150	1.066	0.638

En la Tabla 11 se resumen los entrenamientos realizados para predecir las coordenadas X e Y. El propósito de los entrenamientos fue encontrar la red que al ser evaluada con los datos test, entregue el valor más acercado a cero en la función de perdida (Loss) y el más acercado a 1 en el coeficiente de determinación. Los detalles sobre la elección de parámetros como el número de capas ocultas, el tipo de interconexión entre capas, la función de activación, y el número de ciclos de entrenamiento se presentan a continuación:

- Numero de Capas Ocultas y Numero de Neuronas por Capa: La red neuronal tiene un total de 4 entradas las cuales corresponden a los valores RSSI de los dispositivos finales y 2 salidas correspondientes a las coordenadas X e Y, se comenzó con el doble de las entradas para el numero de neuronas en el modelo 1 de la Tabla 11 y se fue duplicando el número de neuronas por capa hasta obtener el mejor resultado en el modelo 10, después se procedió a aumentar el número de

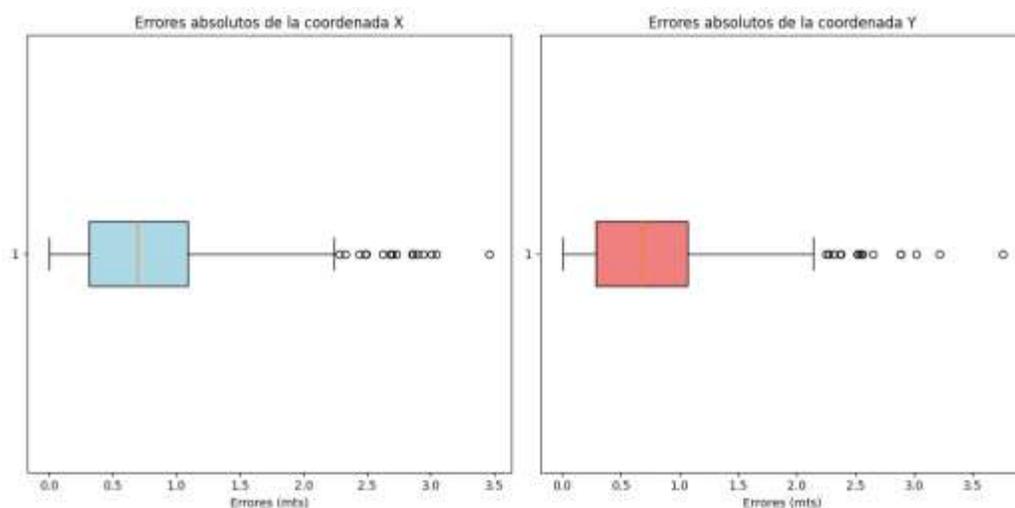
capas ocultas hasta obtener los mejores resultados de la función de pérdida y el coeficiente de determinación.

- **Tipo de Conexión Entre Capas:** se hizo uso del tipo de interconexión Dense lo cual indica que cada capa anterior está conectada para la siguiente, se hizo uso de este tipo de conexión debido a que por su estructura tiene la capacidad de aprender representaciones más profundas a partir de los datos de potencia.
- **Funcion de Activación:** se escogió la función ReLU debido a que tiene la capacidad de recibir datos de entrenamiento tanto negativos como positivos en el rango de potencias que se manejaron que fue desde -36 dBm hasta los -50 dBm, así que no hubo la necesidad de normalizar los datos, de esta forma se pueden pasar los datos al modelo tal y como el microcontrolador los recibe optimizando en pequeña medida los tiempos de predicción.
- **Numero de ciclos de entrenamiento:** en la Tabla 11 se puede apreciar que en la mayoría de modelos entrenados se entrenan 100, 150 y 200 veces el modelo, cabe resaltar que también se llevaron pruebas tanto por debajo de 100 y por encima de 200 ciclos de entrenamiento, obteniendo peores resultados, de igual forma también se probaron valores intermedios a los rangos planteados en la tabla, de esa forma solo se presentan los mejores valores de pérdida y coeficiente de determinación contemplados en este rango.
- **Pérdida del Modelo y Coeficiente de Determinación:** el modelo 10 de la Tabla 11 fue el utilizado para realizar las predicciones en adelante en este caso se obtuvo un valor de la función de pérdida de 0.978 lo cual indica que tan lejos están las predicciones del modelo con respecto a los valores reales, en este caso las salidas

se indican en metros, para tener una mejor visualización de los errores del modelo se puede consultar la Figura 49. En cuanto al coeficiente de determinación en la mejor predicción se encuentra en 0.692 lo cual indica que el 69.20 % de la variación en los datos puede ser explicada por el modelo.

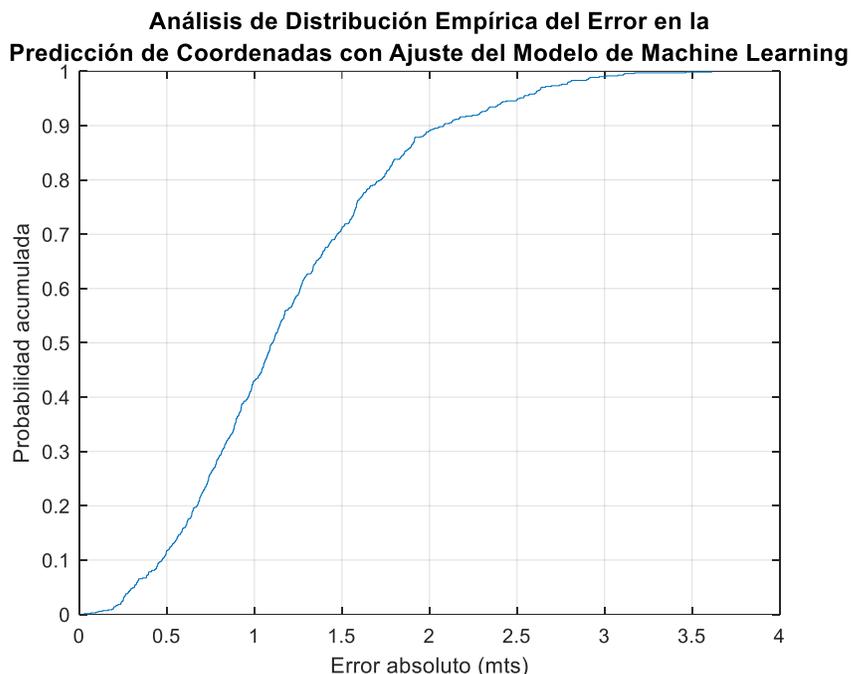
Una vez entrenado el modelo, se pasó a probar su efectividad en la predicción de las coordenadas X e Y, el comportamiento del error con respecto a las coordenadas se puede apreciar en la siguiente figura.

Figura 49. Comportamiento del error en las predicciones del modelo de Machine Learning.



En la figura anterior se puede apreciar que el error se encuentra en su mayoría concentrado entre 0 y 1.5 mts, para entender mejor el porcentaje de apariciones de error en las predicciones, se puede observar la Figura 5050.

Figura 50. Análisis del error cuadrático para las coordenadas (X, Y), con el modelo de Machine Learning.



En la figura se observa que las predicciones del modelo de Machine Learning entrenado presentan un error de 0.5 metros en al menos un 10% de las ocasiones. Además, el 90% de las predicciones se encuentran por debajo de los dos metros de error. Adicionalmente, se puede ver que el error máximo en las predicciones es de 3.5 metros. Aunque este modelo tiene un margen de error considerable en algunas mediciones, se ajusta adecuadamente a las variaciones de potencia de la señal. Esto indica que, a pesar de las fluctuaciones en la potencia de la señal, el modelo mantiene un rendimiento aceptable, proporcionando valores de error que, aunque no son insignificantes, permanecen dentro de límites razonables para la aplicación prevista. Una comparación de los modelos implementados a lo largo de esta investigación puede encontrarse en la sección 4.6.

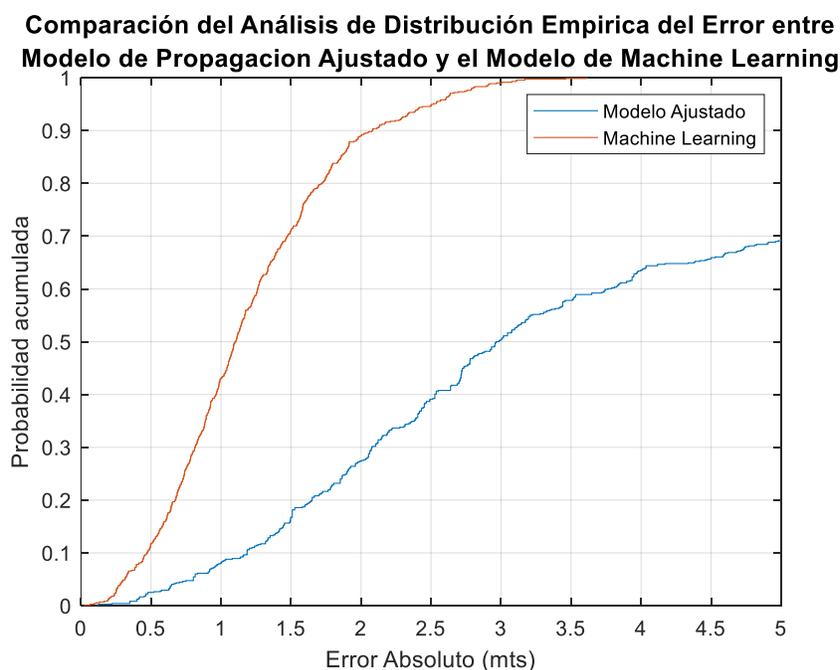
4.5 Comparación entre el Modelo de Propagación ajustado y el Modelo de Machine Learning

En este apartado, se realiza una comparación exhaustiva del error asociado a la predicción de coordenadas entre los dos modelos considerados: el modelo Log Distancia y el modelo de Machine Learning.

Para lograr una evaluación precisa, se analiza detalladamente el rendimiento de ambos modelos en términos del error absoluto presente en la triangulación y en la predicción de coordenadas. Se examina cómo estos errores se distribuyen a lo largo de los conjuntos de datos, utilizando gráficos de distribución empírica del error acumulado.

Este análisis comparativo proporciona una comprensión de cómo se comporta el Modelo ajustado y de Machine Learning en términos de precisión, lo que es fundamental para la toma de decisiones informadas sobre su aplicación.

Figura 51. Comparación del error en la predicción de la posición para el Modelo Ajustado y el Modelo de Machine Learning



Cabe resaltar que, para realizar la comparación de los dos modelos considerados, se utilizó el 20 por ciento de los datos medidos en cada coordenada, de acuerdo con lo expuesto en la sección

4.4.

4.6 Análisis de los resultados

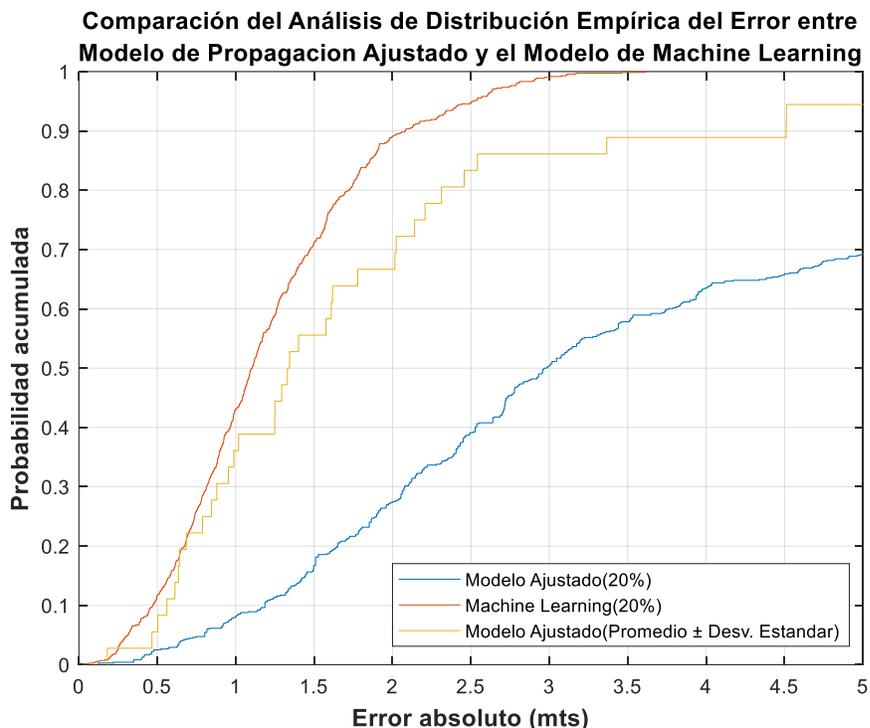
Con base en los resultados obtenidos se hizo una comparación de cada uno de los estudios con respecto a los comportamientos de los errores en la predicción de las coordenadas.

Los modelos utilizados para realizar esta comparación son:

- Modelo de propagación ajustado usando el promedio \pm la desviación estándar que se encuentra en la sección 4.3.3 y su cálculo de trilateración implementado en la sección 4.3.4.
- Modelo entrenado de Machine Learning con el 80% de los datos recolectados por los dispositivos finales, y probado con el 20 % de los datos. Se encuentra en la sección 4.4.
- Modelo de propagación ajustado, realizando las predicciones con el 20% de los datos de prueba del modelo de Machine Learning. Se encuentra en la sección 4.5, para el cálculo de las coordenadas, se utilizó el procedimiento de trilateración de la sección 4.3.4.

En esta comparación se aprecia que el modelo de Machine Learning presenta un menor error en la predicción de la estimación de la posición. La comparativa se puede apreciar mejor en la Figura 5252.

Figura 52. Comparación del error en los tres estudios para la predicción de las coordenadas (X, Y).



Para entender el comportamiento respecto al error en la obtención de las coordenadas X e Y para los modelos y las predicciones realizadas, se debe tener en cuenta que todas las predicciones, se llevaron a cabo en un espacio de 25 metros cuadrados en el centro del espacio deportivo como se puede apreciar en la Figura 399.

La Figura 52 del análisis de la distribución empírica muestra que el algoritmo de Machine Learning presenta el mejor rendimiento, con una curva más próxima a cero. Esto indica que, en comparación, sus predicciones son más precisas. Aunque el modelo de propagación utilizando el promedio \pm la desviación estándar también ofrece buenas aproximaciones, el algoritmo de Machine Learning destaca con un 90% de sus predicciones por debajo de los dos metros de error. Por otro lado, el modelo de propagación logra mantener alrededor del 65% de sus predicciones dentro de este rango de precisión. La peor aproximación ocurre cuando se

aplican al modelo de propagación el 20% de los datos de prueba del algoritmo de Machine Learning, resultando en errores altos y menos del 30% de las predicciones con errores menores a dos metros, una descripción más detallada de estos comportamientos puede observarse en la Tabla 13.

Tabla 12. Predicciones de los Modelos Implementados en las coordenadas 5x5 metros

Coordenada Real en X [metros]	Coordenada Real en Y [metros]	Aproximación Modelo de propagación en Eje X, promedio \pm desviación estándar [metros]	Aproximación Modelo de propagación en Eje Y, promedio \pm desviación estándar [metros]	Aproximación del modelo de propagación con los datos de entrenamiento de ML coordenada X [metros]	Aproximación del modelo de propagación con los datos de entrenamiento de ML coordenada Y [metros]	Aproximación Modelo Machine Learning en eje X [metros]	Aproximación Modelo Machine Learning en eje Y [metros]	Error Modelo de propagación con los datos \pm la desviación estándar [metros]	Error Modelo de propagación con los datos de entrenamiento de ML [metros]	Error en la predicción de coordenadas con el modelo Machine Learning [metros]
0	0	-0,8309	-0,0314	21,063	12,2581	0,6098	0,911	0,9876	25,8083	1,1314
0	1	-1,041	0,5333	16,9377	11,5069	0,6969	1,1094	1,3296	20,5973	0,8866
0	2	1,6893	1,6726	6,7044	7,2894	1,192	2,1928	1,6173	8,9637	1,5377
0	3	-0,0654	1,9086	8,319	8,5529	0,9865	1,9543	1,25	10,4823	1,838
0	4	1,2147	4,7798	44,1119	18,1362	0,905	3,6229	1,2487	47,4564	1,2849
0	5	1,4817	5,0487	28,6724	18,811	1,2343	4,0135	1,3435	32,4285	1,8306
1	0	-0,5449	0,7048	1,7249	3,8357	1,4515	1,1448	1,7779	4,0929	1,322
1	1	1,7874	0,4398	9,3036	6,6155	1,2736	1,5066	0,9537	11,4582	0,7931
1	2	3,0603	1,0506	4,8328	6,5	1,2639	2,0415	2,2073	7,0063	0,9062
1	3	1,7999	2,6101	6,9883	6,1372	1,576	2,5126	0,8461	8,1872	1,3048
1	4	0,261	5,2307	11,7036	6,3096	1,7347	3,8334	1,4006	12,0703	1,0015
1	5	1,747	5,0487	8,7185	9,4524	1,4863	4,0185	0,6127	10,555	1,302
2	0	2,9916	-0,4189	1,9988	3,2784	1,9543	1,216	1,0182	3,4142	1,4433
2	1	-0,3171	0,4978	2,3507	3,6412	1,7075	1,1639	2,5414	3,2246	0,7739
2	2	3,6227	0,7752	2,4222	3,5942	2,1596	1,8144	2,0151	2,4677	1,0561
2	3	2,5586	3,5163	5,7627	3,6983	2,3072	3,0104	0,5609	4,7866	1,3396
2	4	1,9947	4,7654	19,02	8,0955	2,3299	3,9185	0,641	20,6186	1,0164
2	5	1,7727	5,4843	11,1208	8,5548	2,1722	4,0983	0,5035	12,5449	1,2713
3	0	5,4024	-0,8209	17,608	3,474	3,5971	1,1578	2,4573	16,9705	1,5469
3	1	6,2087	-2,1682	3,5897	3,1696	2,8167	1,4888	4,5124	3,1225	1,0586
3	2	3,2647	2,2741	5,3414	3,0837	2,8192	2,1872	0,1811	3,6728	1,3106
3	3	3,7381	2,4951	11,7615	3,5528	2,7904	2,8779	0,8797	10,5198	1,212
3	4	2,4771	4,5654	37,4957	6,8824	2,8366	3,94	0,788	37,5769	1,023

3	5	2,5175	7,3674	13,4719	6,9302	3,1279	4,4116	2,3118	12,9491	1,0753
4	0	23,551	-21,725	5,2639	2,5278	4,3679	0,3413	29,2381	3,4533	0,6129
4	1	4,2658	-0,1461	4,3868	2,5416	3,9435	1,1449	1,2935	1,9044	0,8063
4	2	3,6789	2,1965	4,429	2,4147	3,6627	1,7135	0,4657	1,3885	1,0348
4	3	3,6804	3,6482	5,3141	3,1847	3,4382	3,0335	0,6851	3,3427	1,1846
4	4	2,4494	5,2517	16,6751	6,1951	2,938	4,0554	2,0238	16,7623	1,3842
4	5	4,4078	4,5651	7,5755	5,9761	3,7499	4,1184	0,6349	6,6403	1,2238
5	0	3,0097	0,3342	4,6684	2,518	4,0638	1,0149	2,1405	2,9068	1,4668
5	1	-0,9641	2,6821	6,4046	2,5158	4,2113	1,1957	6,2969	2,8203	1,0223
5	2	6,2794	1,0036	6,8694	2,682	4,0722	2,3263	1,6093	3,2083	1,4456
5	3	3,8008	3,9686	7,887	2,4071	4,1771	3,1081	1,5752	3,9206	1,177
5	4	2,6564	7,906	20,0833	3,6539	3,6911	4,0757	4,5108	17,2164	1,4832
5	5	2,8997	7,6524	9,0135	2,4138	4,2769	4,1336	3,3661	5,9877	1,2658

Teniendo en cuenta la tabla anterior, en la cual se evidencia las predicciones realizadas por los modelos aplicados y el error en la predicción de cada uno, se entiende que en algunos casos las predicciones hechas por los modelos tuvieron errores de más de un metro, bien sea en el eje X o eje Y, como es en el caso de la coordenada (2,2), sin embargo, existieron predicciones muy aproximadas hechas por los modelos, como es en el caso de la coordenada (3,2).

Teniendo en cuenta que las predicciones expuestas en la Tabla 122, se decidió analizar los errores absolutos en la aproximación de las coordenadas X e Y. Se establecieron rangos de error para evaluar la cantidad de predicciones dentro de estos rangos para cada modelo probado. En la Tabla 133, se presenta la proporción de apariciones por debajo de cada rango de error, permitiendo una comparación clara entre los modelos. El porcentaje indica cuántas predicciones se encuentran por debajo de un límite específico, lo que ayuda a determinar cuál modelo tuvo más aproximaciones precisas y, por lo tanto, el mejor rendimiento.

Tabla 13. Porcentajes de predicción de los modelos según las coordenadas

Estudios Realizados	Porcentaje de valores de error por debajo de 0.5mts en la predicción.	Porcentaje de valores de error por debajo de 0.7mts en la predicción.	Porcentaje de valores de error por debajo de 1 mt en la predicción.	Porcentaje de valores de error por debajo de 1.5mts en la predicción.
-Modelo de Propagación Ajustado (Promedio \pm Desviación Estándar).	5,56 %	22,22 %	36,11 %	55,56 %
-Modelo de Propagación Ajustado (probado con el 20 % Datos Test Machine Learning).	2,5%	4,33 %	8,10 %	16,75 %
-Modelo Machine Learning (probado con el 20% Datos Test)	11,73%	22,21 %	43,02 %	71,00 %

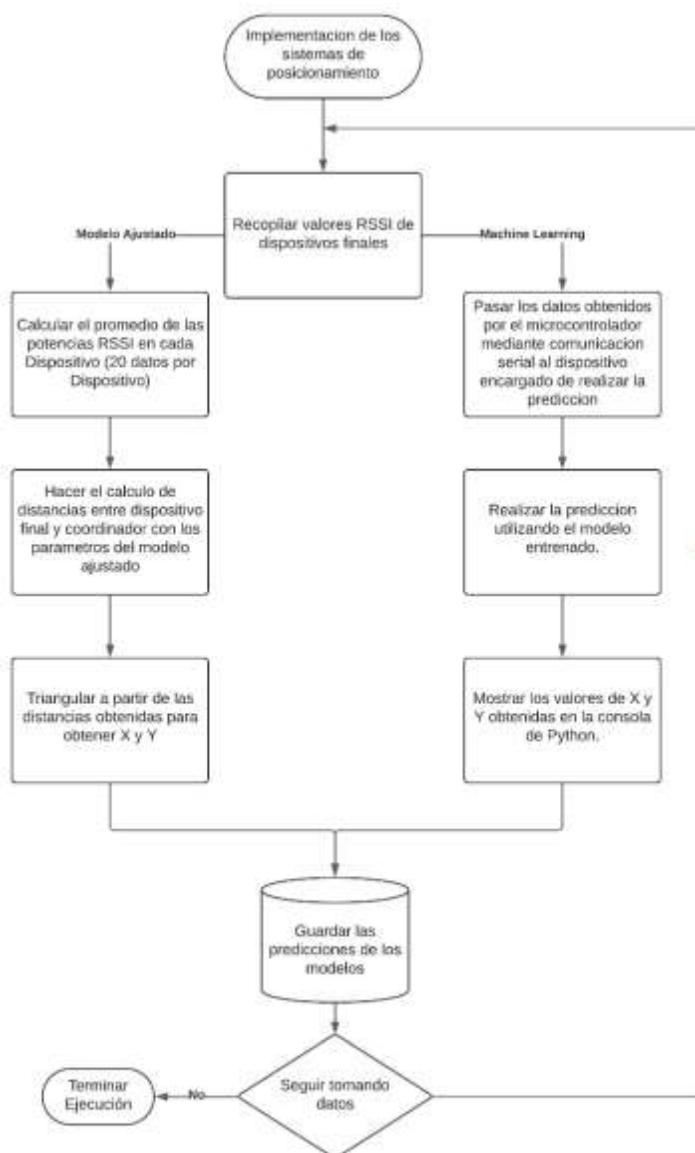
A partir de los resultados, se concluye que, aunque el comportamiento general no es el mejor para los tres modelos, el Modelo Machine Learning probado con el 20% de los datos, es el que mejor se comportó. En particular, muestra un rendimiento notablemente superior en los rangos de error por debajo de 1 metro y 1.5 metros, alcanzando un 43.02% y 71.00% respectivamente. En contraste, el Modelo de Propagación Ajustado y probado con el 20% Datos

Test de Machine Learning tuvo el peor rendimiento, con los porcentajes más bajos en todos los rangos de error, lo que indica que este ajuste no fue eficaz en comparación con los otros modelos. El Modelo de Propagación Ajustado utilizando la técnica de filtrado de datos del Promedio \pm Desviación Estándar, mostró un rendimiento intermedio, siendo mejor que el modelo ajustado con datos de Machine Learning, pero inferior al modelo puramente basado en Machine Learning.

4.7 Implementación de los sistemas de posicionamiento dentro de la Cancha #1 de la Universidad CESMAG.

Para la implementación de los sistemas en el espacio deportivo se propone considerar el siguiente diagrama de flujo con el fin de entender cómo es el funcionamiento de los sistemas de posicionamiento.

Figura 53. Funcionamiento de los sistemas de posicionamiento implementados en el espacio deportivo



Para probar los modelos, una vez recolectadas las mediciones para cada estudio, se utilizó la plataforma Matlab para analizar los resultados. En el caso de la creación de la red neuronal para el algoritmo de Machine Learning, se empleó el lenguaje de programación Python junto con la librería Keras. Esta combinación permitió desarrollar y entrenar el modelo de Machine Learning. Para la implementación práctica, es crucial considerar los modelos ajustados y

determinar la manera de integrarlos en un microcontrolador. Esto permitió que el microcontrolador realice las predicciones correspondientes en tiempo real, asegurando que el sistema funcione de manera autónoma y precisa en su entorno operativo.

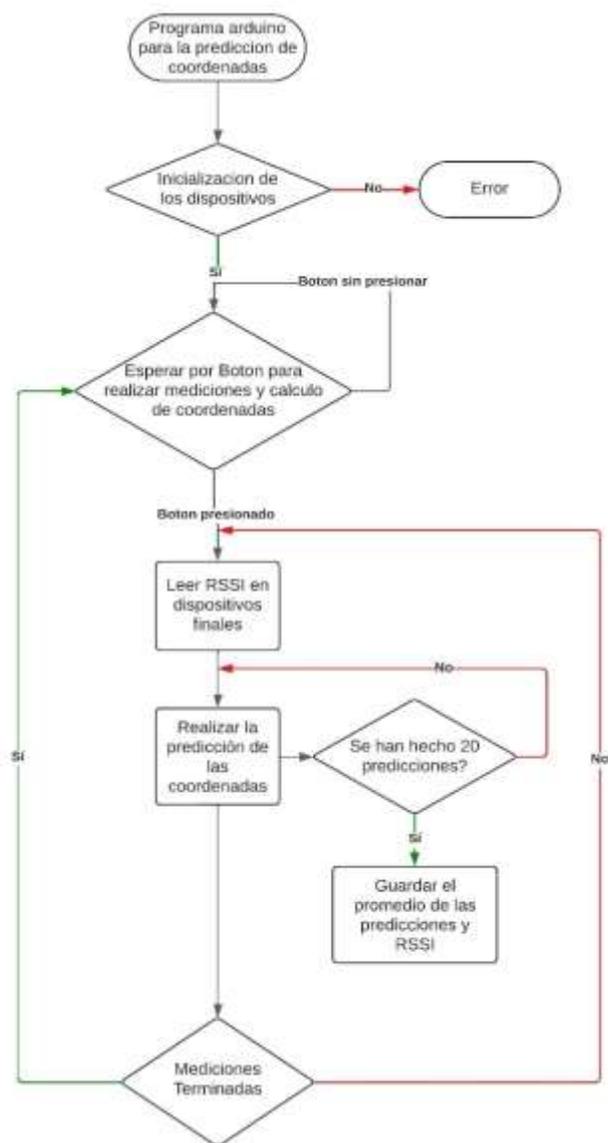
En principio se debe considerar que los sistemas de posicionamiento o el sistema debe ser implementado en un MCU (Unidad de Microcontrolador). En el caso del sistema de Machine Learning se necesita de una herramienta que permita implementar y ejecutar modelos de redes neuronales para MCU. Con el fin de optimizar el código y los modelos para trabajar con las limitaciones de recursos, en tal caso ya existen tecnologías que han sido implementadas para ejecutar modelos de Machine Learning en microcontrolador, tales como es el caso de NNOM (Red Neural en Microcontrolador) este es un marco de red neuronal fácil de usar para desarrolladores integrados que trabajan con MCU. Majianjia, J. (2019). Sin embargo, en el desarrollo de este trabajo, la transmisión de los valores de RSSI se hicieron al computador para que el modelo de Machine Learning haga las estimaciones de la posición.

El funcionamiento de los sistemas comienza con la inicialización de los dispositivos. Si no se detecta ningún error durante este proceso, se procede a la obtención del RSSI de los dispositivos XBee. Una vez recopiladas las potencias, los datos se envían a los modelos para su procesamiento. Estos modelos generan las coordenadas (X, Y), y las predicciones resultantes son guardadas y transmitidas a otra unidad de visualización de coordenadas.

Para una mejor comprensión de la implementación de los sistemas, la explicación del funcionamiento para el caso del modelo ajustado se puede remitir a la *Figura 20. Representación de la secuencia de mediciones entre los dispositivos finales y el coordinador*. En esta figura se representa la forma en que se realizan las mediciones y se guardan los valores en una tarjeta micro SD. Para la implementación, no solo se recolectaron los datos, sino que también se realizó

el cálculo del promedio de los valores RSSI y el cálculo de las coordenadas X e Y en el microcontrolador. Posteriormente, los resultados se guardaron en la tarjeta SD en un archivo CSV. El diagrama de flujo que explica el funcionamiento del código en el microcontrolador se puede observar en la Figura 54.

Figura 54. Diagrama de Flujo para el programa Arduino para la predicción de coordenadas



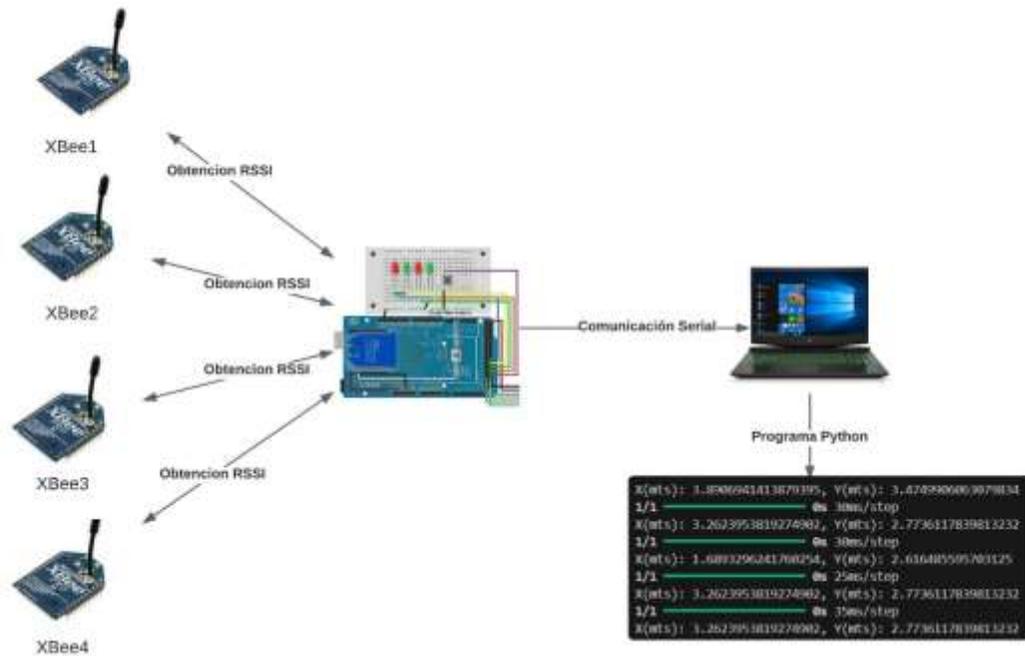
Para la implementación del sistema de predicción de coordenadas utilizando el algoritmo de Machine Learning entrenado previamente, y evidenciando que el microcontrolador no tenía las capacidades necesarias para ejecutar dicho algoritmo, se hizo uso de la comunicación serial para transferir los datos medidos por el microcontrolador a un computador con la capacidad de realizar las predicciones, cuyas especificaciones se exponen en la Tabla 14.

Tabla 14. Especificaciones del equipo usado para la implementación del modelo de Machine Learning

Componente	Especificación
Procesador	Intel(R) Core(TM) i5-8300H CPU
Núcleos y velocidad del procesador	4 núcleos a 2.30 GHz
Arquitectura	64 bits
RAM	16 GB
Tarjeta de Video	Nvidia Geforce GTX 1050 TI 4GB

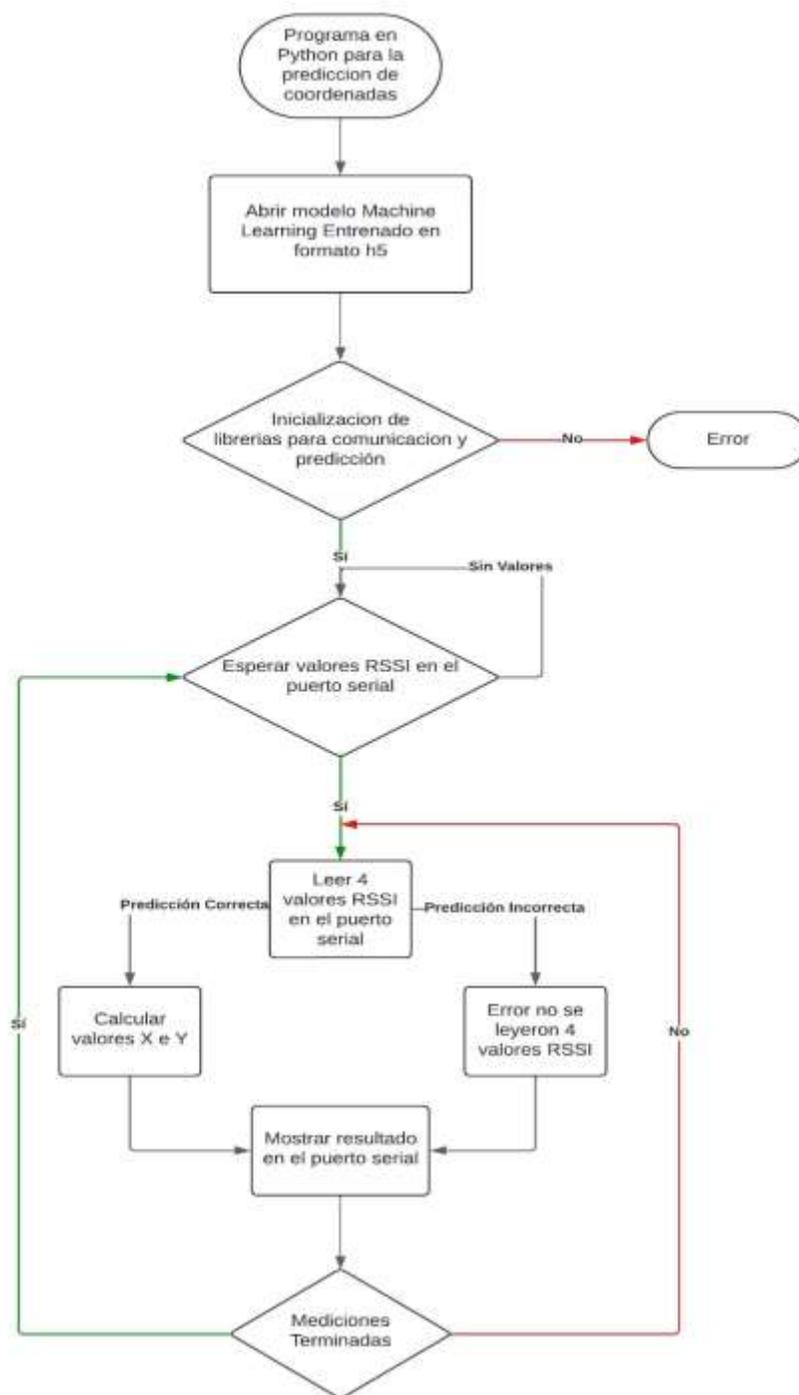
El código del microcontrolador encargado de realizar las mediciones fue levemente modificado para que enviara los valores mediante comunicación serial. Con un programa realizado en Python, se recibieron los valores de RSSI de los cuatro dispositivos finales y se realizó la predicción de las coordenadas X e Y. En la Figura 55 se presenta cómo se realizan las predicciones, utilizando un microcontrolador conectado a una computadora que emplea un programa en Python para realizar las predicciones de las coordenadas X e Y.

Figura 55. Funcionamiento de la implementación del modelo de Machine Learning para la obtención de coordenadas usando comunicación serial.



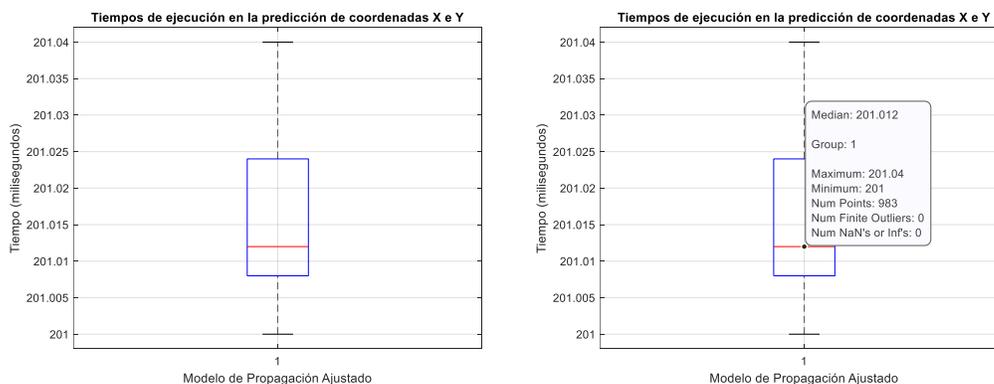
El programa de Python permitió recibir los datos RSSI de los cuatro dispositivos finales y realizar las predicciones a partir del modelo guardado que se entrenó previamente con la herramienta de Google Colab. Si se desea indagar el código se puede visitar la sección de Anexos. En la Figura 56 se presenta un diagrama de flujo correspondiente al programa en Python.

Figura 56. Diagrama de flujo programa en Python para la predicción de coordenadas usando modelo entrenado de Machine Learning.



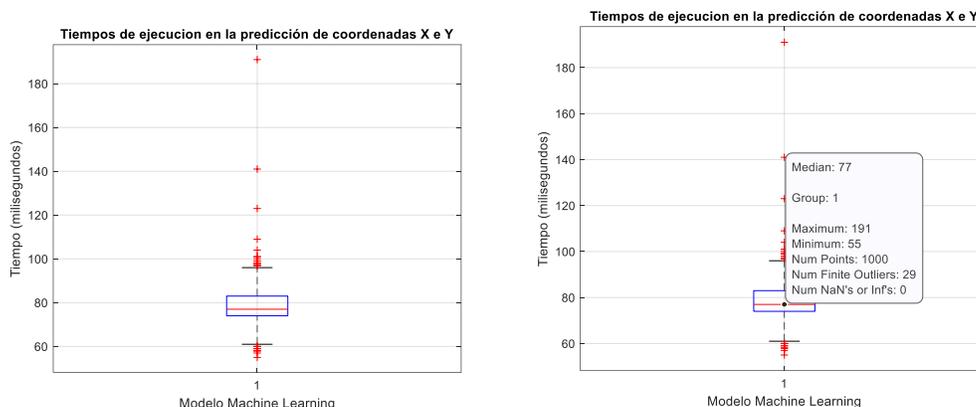
Una vez realizada la implementación y comparada la precisión de cada uno de los modelos en la predicción de coordenadas, también fue necesario evaluar cuánto tiempo le toma a cada uno realizar las predicciones tomando 1000 mediciones en los cuatro dispositivos finales y realizando los cálculos de las coordenadas como se indica en la Figura 53. En el caso del modelo de propagación, todo el cálculo se realiza en el microcontrolador. En cambio, en el modelo de Machine Learning, solo se recolectaron las mediciones con la ayuda del microcontrolador y luego se realizó la predicción en un dispositivo externo usando Python. Para el caso del modelo de Machine Learning, solo se considera el tiempo en el cual el computador hace la predicción. Por el contrario, en el caso del modelo de propagación, todo el cálculo se realiza en el microcontrolador, se considera el tiempo en el que se lleva a cabo el cálculo de las coordenadas.

Figura 57. Tiempos de ejecución en la predicción de coordenadas para el modelo de propagación ajustado.



Para el caso del modelo de propagación ajustado, en la Figura 57 se puede apreciar que los tiempos de ejecución por cada obtención de medidas y el posterior cálculo de una coordenada tienen una mediana de 201 ms. Además, los tiempos no presentan una gran variación, y sus límites superiores e inferiores no se alejan demasiado de la mediana, lo que indica una estabilidad en la predicción de las coordenadas.

Figura 58. Tiempos de ejecución en la predicción de coordenadas para el modelo de Machine Learning.



Debido a las limitaciones del microcontrolador destinado para el proyecto, el modelo de Machine Learning requiere una fuente externa de procesamiento capaz de realizar las predicciones de las coordenadas. Los tiempos de predicción son más bajos utilizando un terminal con las especificaciones planteadas en la Tabla 144, y solo se considera el tiempo necesario para que el computador obtenga las coordenadas con el modelo previamente entrenado.

En la Figura 58 se puede apreciar que la mediana de los tiempos de predicción es de aproximadamente 77 ms, En este caso particular, se observa una mayor variabilidad en el rango de tiempos de predicción. Sin embargo, también se debe tener en cuenta que la precisión del modelo de Machine Learning supera a la del modelo de propagación ajustado.

El modelo de Machine Learning es más preciso y el entorno en el cual se realizan las predicciones es más rápido, aunque no se tiene en cuenta el tiempo de transmisión de los datos mediante el puerto serial. En contraste, el Modelo Ajustado, del cual se obtuvieron las variables para poder realizar el cálculo en el microcontrolador, es un poco más lento, pero permite realizar el cálculo en el mismo entorno, sin necesidad de comunicación externa.

5. Conclusiones y Trabajos futuros

Para abordar el problema de investigación sobre la precisión de un sistema de posicionamiento basado en ZigBee en una cancha de microfútbol, este trabajo de grado se centró en la estimación de la precisión de los dispositivos ZigBee en el posicionamiento de un jugador en los ejes X e Y, utilizando dispositivos XBee y la potencia RSSI. Para esto a diferencia de los antecedentes presentados en relación al trabajo realizado en este documento se aplicó un modelo de propagación ajustado al entorno y técnicas de Machine Learning para realizar una trilateración de coordenadas, de este modo se logró estimar la precisión de los dispositivos XBee en la Cancha de microfútbol #1 de la Universidad CESMAG.

En el desarrollo del presente trabajo de grado se concluye que en base a las hipótesis planteadas en la sección 2.4, considerando que los sistemas de posición presentados en el desarrollo de este trabajo y en base a los resultados obtenidos teniendo en cuenta que el sistema de posicionamiento con el porcentaje de valores de error por debajo de 1.5mts en la predicción en la estimación de la posición es el Modelo Machine Learning (probado con el 20% Datos Test), con un porcentaje de 70%, se concluye que se cumple la hipótesis nula en la sección 2.4.2, dado que no se obtuvo un error menor a 5% haciendo uso de un modelo de propagación ajustado al entorno de la Cancha #1 de la Universidad CESMAG y a un algoritmo de Machine Learning basado en redes neuronales.

Los análisis respectivos de los resultados de los sistemas no fueron favorables, ya que fue necesario realizar cambios en los dispositivos XBee y el espacio en la toma de datos dentro de la Cancha #1 de la Universidad CESMAG. En general, el rendimiento de los dispositivos XBee indicó que la precisión de la estimación de la posición no era satisfactoria cuando se implementó el sistema en la Cancha #1 de la Universidad CESMAG. Esto se debió, en gran parte, a la

potencia insuficiente de los dispositivos XBee para transmitir datos de manera efectiva en un área tan amplia, lo que resultó en una alta variabilidad de los datos.

Al reducir el espacio de toma de datos, se observó una mejora significativa en la estimación de la precisión de la posición, como se evidencia en la sección 4.3.1.1, Figura 27.

Comparación de error del ajuste del modelo en la aproximación de la distancia en cada dispositivo a 14dBm, considerando el XBee 1 con el menor error tuvo una mediana de aproximadamente 7 metros y en comparación presentado en la sección 4.3.3, Figura 44.

Comparación del error de los dispositivos con el ajuste del modelo, en un espacio de 5x5 mts en la Cancha #1 de la Universidad CESMAG., la distribución de la mediana que tuvieron todos los dispositivos, muestra que se encuentra a menos de 1 metro.

Esto sugiere que los dispositivos XBee, en la aplicación dada en el presente trabajo, y con la técnica RSSI, funcionan de manera más eficiente en áreas más pequeñas, donde la distancia entre los nodos es limitada y la variabilidad de los datos es poca.

La aplicación de técnicas de machine Learning mejoró considerablemente la estimación de la precisión de la posición. Como se evidencia en la sección 4.6, Tabla 13. Porcentajes de predicción de los modelos según las coordenadas. Los modelos de machine Learning fueron capaces de adaptarse mejor a las variaciones en la señal RSSI, proporcionando estimaciones más precisas en comparación con los modelos de propagación ajustado (promedio \pm desviación estándar) y modelos de propagación ajustado (20% datos test machine Learning).

Una de las limitaciones más significativas fue la potencia limitada de los dispositivos XBee PRO S1, lo cual afectó la precisión del sistema en áreas más grandes. Esto destaca la necesidad de considerar las especificaciones del hardware al diseñar sistemas de posicionamiento basados en RSSI.

Por lo tanto, para mejorar y expandir el sistema de posicionamiento implementado, se puede considerar una serie de acciones interconectadas que abordan diversos aspectos técnicos y prácticos.

En primer lugar, es fundamental investigar y utilizar dispositivos con mayor potencia de transmisión y recepción que operen de manera efectiva en áreas más grandes sin pérdida significativa de precisión.

Además, el desarrollo y entrenamiento de modelos de Machine Learning más avanzados, incluyendo redes neuronales profundas y algoritmos de aprendizaje por refuerzo, puede mejorar notablemente la precisión de la estimación de la posición. Estos modelos pueden aprender de grandes cantidades de datos y adaptarse a variaciones en el entorno, proporcionando estimaciones más fiables. Así aplicar técnicas de Machine Learning en este estudio puede mejorar las estimaciones de la posición a diferencia de las técnicas aplicadas en los antecedentes.

En esta investigación se llevó a cabo el entrenamiento y la implementación de Machine Learning para la predicción de coordenadas. Además de poder seguir entrenando el algoritmo con nuevos datos, también se puede realizar la implementación dentro de un microcontrolador, haciendo uso de herramientas que permitan reducir al máximo la complejidad del algoritmo y, a su vez, optimizarlo para que obtenga buenas predicciones y no sea tan complejo, permitiendo su ejecución en un mismo entorno por un microcontrolador.

Es crucial también probar el sistema en diferentes entornos, como interiores, exteriores y áreas urbanas, para evaluar su robustez y adaptabilidad. Estas pruebas permitirán identificar y mitigar factores ambientales que puedan afectar la precisión del sistema, como la interferencia de señales o la presencia de obstáculos.

Realizar un análisis exhaustivo de diferentes modelos de propagación de señales es esencial para identificar aquellos que mejor se adapten a las características del entorno donde se implementará el sistema. Cada entorno tiene características únicas que afectan la propagación de señales, y seleccionar el modelo adecuado puede mejorar significativamente el rendimiento del sistema y así mejorar la estimación de la precisión de los dispositivos XBee.

La integración de sistemas con otros tipos de sensores en el MCU (Unidad de Control de Microprocesador), como sensores de movimiento, acelerómetros y giroscopios, es otro paso importante. La fusión de datos provenientes de múltiples fuentes puede proporcionar una estimación de posición más precisa y fiable, compensando las limitaciones individuales de cada sensor.

Para garantizar que las configuraciones del sistema sean óptimas antes de su implementación en el mundo real, se puede crear una plataforma de simulación. Esta plataforma permitirá probar y validar diferentes configuraciones del sistema, ahorrando tiempo y recursos al identificar posibles problemas y optimizar el diseño en un entorno controlado.

Finalmente, debido a las limitaciones del estudio realizado y los resultados obtenidos, se propone continuar con la implementación de sistemas de posicionamiento para el seguimiento de jugadores en una cancha de microfútbol.

6. Referencias

- Oliveira Rodríguez, L. A., & García Fernández, R. (2021). Low cost EPTS (Electronic Performance & Tracking System) development using IoT devices. En 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-4). Chaves, Portugal. <https://doi.org/10.23919/CISTI52073.2021.9476415>
- 30,235 cancha Futbol Vectores, Ilustraciones y Gráficos - 123RF. (n.d.). Retrieved October 21, 2022, from https://es.123rf.com/clipart-vectorizado/cancha_futbol.html
- Basogain, X. (1998). *REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES*.
- Bravo, E. M., & Zambrano, E. S. (2004). *Sistema Prototipo de Posicionamiento Local Destinado a la Búsqueda de Libros Utilizando Tecnología ZigBee*.
- Digi. (2018). XBee/XBee-PRO S1 802.15.4 (Legacy) RF Modules User Guide. *Digi International, 4*.
- Guerra, L. (2010). *DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA LOCALIZACIÓN DE UN OBJETO EN MOVIMIENTO MEDIANTE ZIGBEE*.
- Moreno, J. M., & Ruiz Fernandez, D. (2007a). Informe Técnico: Protocolo ZigBee (IEEE 802.15.4). In *Dtic* (Vol. 802).
- Moreno, J. M., & Ruiz Fernandez, D. (2007b). Informe Técnico: Protocolo ZigBee (IEEE 802.15.4). In *Dtic* (Vol. 802).
- Picallo Guembe, I. (2016). *Diseño e Implementación de un Sistema de Monitorización de Práctica Deportiva de Baloncesto en Entornos Contextuales WPAN/WBAN*.
- Quijano, A. (2009). *Guía de Investigación Cuantitativa - PDF Descargar libre.pdf*.

Rappaport, T. S. (2002). *Wireless communications: Principles and practice* (2nd ed.).

<http://revistas.unne.edu.ar/index.php/eitt/article/view/275/241>

Tablada, J., & Torres, G. (2009). *Redes Neuronales Artificiales Introducción Modelo neuronal de McCulloch-Pitts*. 22–30.

Vara, N., & Poletto, G. A. (2015). *Función Del Parámetro Rssi*. 54(0), 3.

Xiao, Z., & Zeng, Y. (2022a). An overview on integrated localization and communication towards 6G. *Science China Information Sciences*, 65(3), 1–35.

<https://doi.org/10.1007/s11432-020-3218-8>

Xiao, Z., & Zeng, Y. (2022b). An overview on integrated localization and communication towards 6G. *Science China Information Sciences*, 65(3), 1–35.

<https://doi.org/10.1007/s11432-020-3218-8>

Valencia, Á. C., & Giraldo, A. E. (2010). Prototipo de un sistema de telemetría y control para seguridad en vehículos, soportado en redes móviles. *Universidad Católica Popular de Risaralda, Pereira*.

Csanalosi, Gabor & Dobreff, Gergely & Pašić, Alija & Molnar, Marton & Toka, László. (2020). Low-Cost Optical Tracking of Soccer Players. 10.1007/978-3-030-64912-

8_3. Moreno, R. C. J., Benitez, R. A., López, D. E., Medina, N. M. A., & De la Calleja, M. J. (2012). Auto-localización para robots móviles utilizando triangulación y trilateración.

Universidad Politécnica de Puebla. *Revista Visión Politécnica*, (1), 25-30.

<http://repositorio.uppuebla.edu.mx:8080/xmlui/handle/123456789/206>

Mahesh, Batta. (2020). Machine Learning Algorithms -A Review. 10.21275/ART20203995.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Bueno, F. (2019). Redes neuronales: entrenamiento y comportamiento.

7. Anexos

7.1 Especificaciones y configuración de la red.

Como primer paso hizo falta indagar en la capacidad de los módulos XBee-PRO S1, se tuvo en cuenta la frecuencia de funcionamiento siendo de 2.4 GHz, además de las topologías de red soportadas que se presentan en la Tabla 15.

Tabla 15. Descripción de topologías de red soportadas por XBee-PRO S1

(Digi, 2018)

Especificación	XBee-PRO S1
Topologías de red soportadas	Point- to-point, point to multipoint and peer to peer
Número de canales	12 direct sequence channels
Opciones de direccionamiento	PAN ID, Channel and addressess

En el caso del sistema de posicionamiento hará falta una configuración de red en estrella abastecida por la configuración point to multipoint, además se brindan unas especificaciones necesarias para la configuración de la red de lo dispositivos.

Teniendo en cuenta que los dispositivos de red cumplieron con los requerimientos de topología para el diseño de la red, se procedió a indagar sobre su modelo de conexión y pines disponibles que se describen en la Tabla 16 y de una vista superior del dispositivo que se aprecia en la Figura 59.

Figura 59. Vista superior XBee-PRO S1

(Digi, 2018)



Tabla 16. Pines de Conexión XBee-PRO S1

(Digi, 2018)

Pin	Name	Direction	Descripción
1	VCC	-	Power supply
2	DOUT	Output	UART data out
10	GND		Ground
11	AD4/DI	Either	Analog input 4 or digital I/O 4
	O4		

En la conexión del módulo Coordinador se hizo uso de los pines de alimentación uno y diez teniendo un rango de operación que se describe en la Figura 60.

Figura 60. Especificaciones de alimentación XBee-PRO S1

(Digi, 2018)

Specification	XBee	XBee-PRO
Supply voltage	2.8 - 3.4 V	2.8 - 3.4 V
Transmit current (typical)	45 mA (@ 3.3 V)	<ul style="list-style-type: none"> ■ 250 mA (@3.3 V) (150 mA for international variant) RPSMA module only. ■ 340 mA (@3.3 V) (180 mA for international variant)
Idle/receive current (typical)	50 mA (@ 3.3 V)	55 mA (@ 3.3 V)
Power-down current	< 10 μ A	< 10 μ A

Para la comunicación con el microcontrolador, en el XBee coordinador se hizo uso de los pines dos y tres (DIN, DOUT), utilizando la interfaz UART para el intercambio de datos en serie, en este caso el microcontrolador envía los datos desde el XBee-PRO S1 a través del pin tres como una señal serie asíncrona, y los recibirá a través del pin dos, también se consideró la tasa de transferencia en baudios igual, tanto en el módulo de conexión como en el microcontrolador que será el encargado de procesar los datos recibidos de las consultas realizadas a los dispositivos finales.

Los dispositivos XBee tienen dos modos de operación, el modo transparente es el modo de funcionamiento por defecto donde se ponen en cola todos los datos UART que se reciben a través del pin DIN para su transmisión mediante radio frecuencia y el modo API (Aplicación Programming Interface), en este modo todos los datos que entran y salen del dispositivo están contenidos en tramas que definen eventos dentro del módulo, para el diseño de la red se hizo uso del modo API ya que mediante comandos AT se realizaron las peticiones mediante código desde el microcontrolador.

Para la conexión del XBee mediante comunicación serial con el software XCTU hizo falta utilizar un adaptador UART USB, las configuraciones para la comunicación del dispositivo y el software se presentan en la Tabla 17.

Tabla 17. Configuraciones Comunicación serial XCTU con XBee-PRO SI

(Digi, 2018)

Configuración	Valor Elegido
Baud rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

Los parámetros que se tuvieron en cuenta para la configuración cambiaron tanto en el Coordinador como en los dispositivos finales, con excepciones como el canal y la identificación PANID de la red que se mantuvieron iguales en todos los casos para que la comunicación sea exitosa, además el modo API de los módulos debió ser activado y el dispositivo coordinador debió ser configurado en modo broadcast para que se pudiera comunicar con todos los dispositivos, la identificación MY fue configurada distinta en cada módulo para que puedan ser identificados al momento de usar comandos AT para las consultas del parámetro RSSI, a continuación se presentan las configuraciones tenidas en cuenta para el Coordinador y los dispositivos finales.

Tabla 18. Configuración de red Coordinador

Parámetro	Configuración
CH – Channel	C
ID – PAN ID	AAAA
DH	0
DL	FFFF
MY	0001

CE- Coordinador Enable	Coordinator[1]
BD - Interface Data Rate	9600
AP – API Enable	API enabled [1]
PL – Power Level	Medium[4]

Tabla 19. Configuración de red Dispositivos finales.

Parámetro	Configuración
CH – Channel	C
ID – PAN ID	AAAA
DH	0
DL	0
MY	0002, 0003, 0004, 0005
CE- Coordinador Enable	Dispositivo Final[0]
BD - Interface Data Rate	9600
AP – API Enable	API enabled [1]
PL – Power Level	Medium[4]

En las dos tablas se puede apreciar que el parámetro Power Level que determina la potencia a la que transmitirán los módulos, está configurado en un nivel medio, posteriormente este parámetro puede modificarse, ya que la potencia de transmisión influirá en la toma de datos en el espacio deportivo, si los valores de RSSI captados por el transmisor en los dispositivos Finales no tienen variaciones suficientes será necesario decrementar el nivel de potencia de transmisión para obtener valores más diversos en el espacio de juego, esto se hará con el fin de

ajustar el modelo de propagación y la implementación del algoritmo de Machine Learning, el nivel de transmisión y sus escalas en dBm pueden apreciarse en la tabla 12.

Tabla 20. Nivel de transmisión de dispositivos XBee Pro S1

(Digi, 2018)

Power Level	XBee – PRO
0	10 dBm
1	12 dBm
2	14 dBm
3	16 dBm
4	18 dBm

7.1.1 Creación y prueba de las tramas con el Software XCTU

Para la prueba de comandos AT desde la consola de XCTU hizo falta tener conectados mediante comunicación serial a un terminal portátil los dos dispositivos con la ayuda de un adaptador USB como se apreciar en la Figura 62 , posteriormente la creación de la trama para él envió se hizo desde la herramienta *XBee API frames Generator* incluido en la consola de XCTU y se configuro una trama del tipo Remote AT Command, con el MY de identificación del End Device 1 el cual se puede revisar en la Tabla 19, en la Figura 61, se puede apreciar la interfaz de configuración de creación de la trama.

Figura 61. Interfaz de creación de trama consola XCTU

XBee API Frames Generator

This tool allows you to generate any kind of API frame and copy its value. Just fill in the required

Protocol: 802.15.4 Mode: API 1 - API Mode Without Esc.

Frame type: 0x17 - Remote AT Command

Frame parameters:

Start delimiter	7E
Length	00 0F
Frame type	17
Frame ID	01
64-bit d... address	00 00 00 00 00 00 00 00
16-bit d... address	00 02
Remote ...options	02
AT command	db

Generated frame:

```
7E 00 0F 17 01 00 00 00 00 00 00 00 00 00 02
02 64 62 1D
```

Byte count: 19

Copy frame Close OK

El comando Remote AT command crea una trama de 19 bytes, esta realiza una consulta de la dirección MAC de 64 bits y el parámetro RSSI mediante la dirección de identificación de 16 bits de cada End Device, la estructura de la trama de envío se encuentra a continuación.

Remote AT Command Request (API 1)

7E000F1701000000000000000000000020264621D

Startdelimiter: 7E

Length: 000F(15)

Frametype: 17(RemoteATCommandRequest)

FrameID: 01(1)

64 – bitdest. address: 0000000000000000

16 – bitdest. address: 0002

Commandoptions: 02

ATCommand: 6462(db)

Checksum: 1D

Una vez realizado el envío de esta trama, el dispositivo End Device configurado con la dirección de 16 bits correspondiente, responde con una trama de 20 bytes y describe el estado de la comunicación en el byte número 18, a continuación, se presentan una lista de las posibles respuestas e interpretaciones a tener en cuenta.

Posibles respuestas en	Interpretación
Status	
0x00	Comando ejecutado correctamente.
0x01	Comando AT demasiado largo
0x02	Comando AT no compatible
0x03	Tiempo de espera demasiado largo
0x04	Error de modo de comando

0x05	Error en dirección MY
0x06	Comando AT desconocido
0x07	Parámetro no compatible
0x08	Error en la transmisión
0x09	Error en comando interno

Para que la respuesta del End Device sea exitosa el valor correspondiente en el byte 18 debe ser 0x00, a continuación, se presenta la trama recibida con la consulta del parámetro RSSI y la dirección MAC de 64 bits contenida en la trama.

Remote Command Response (API 1)

7E001097010013A2004090017F00026462002476

Startdelimiter: 7E

Length: 0010(16)

Frametype: 97(RemoteCommandResponse)

FrameID: 01(1)

64 – bitsourceaddress: 0013A2004090017F

16 – bitsourceaddress: 0002

ATCommand: 6462(db)

Status: 00(StatusOK)

Response: 24

Checksum: 76

Dentro de la trama de respuesta se aprecia que el byte número 19 entrega el valor del parámetro RSSI en hexadecimal, este valor representa la fuerza de la señal percibida por el receptor, en este caso particular solo se realizó una prueba a diez metros de distancia para corroborar que la comunicación sea exitosa, además se debe tener en cuenta que este valor se entrega en positivo, pero en realidad el parámetro RSSI es una medida en dBm y una vez convertida a decimal se debe tomar con símbolo negativo.

7.2 Diseño y alimentación energética para los Dispositivos finales.

Los dispositivos finales estarán ubicados en las esquinas del espacio deportivo y por ende no tendrán a la mano una fuente de energía constante, dado que es un espacio semiabierto hubo la necesidad de diseñar contenedores para los XBee en el programa de diseño 3D gratuito Thinkercad en el cual se hizo el contenedor aparte para la fuente de energía y el módulo XBee para que puedan ser manipulados de manera más fácil, en la Figura 62 se presenta el montaje del dispositivo dispositivos finales en su contenedor.

Figura 62. End Device y fuente de alimentación aislada



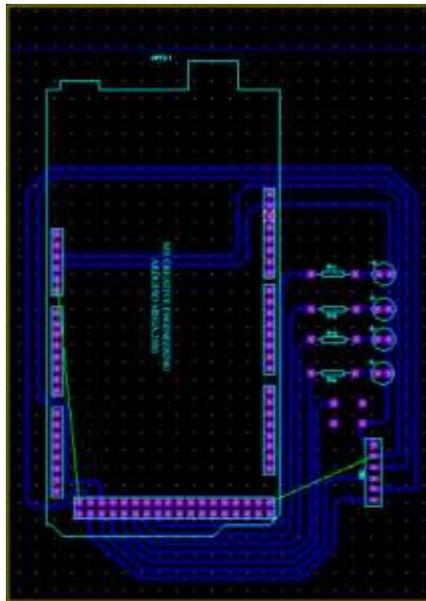
Debido a que los módulos XBee son de bajo consumo, pueden ser alimentados con una pila de 9 V, el voltaje de operación es regulado mediante un circuito que tiene una salida USB de 5 V y que después se conecta al adaptador USB de XBee y cuenta con un regulador de voltaje

incorporado que entrega los 3.3 V de operación y 250mAh entregados por la batería son suficientes para que el módulo trabaje y se realicen las consultas desde el módulo Coordinador.

7.2.1 *Implementación de dispositivos y componentes para el Xbee coordinador.*

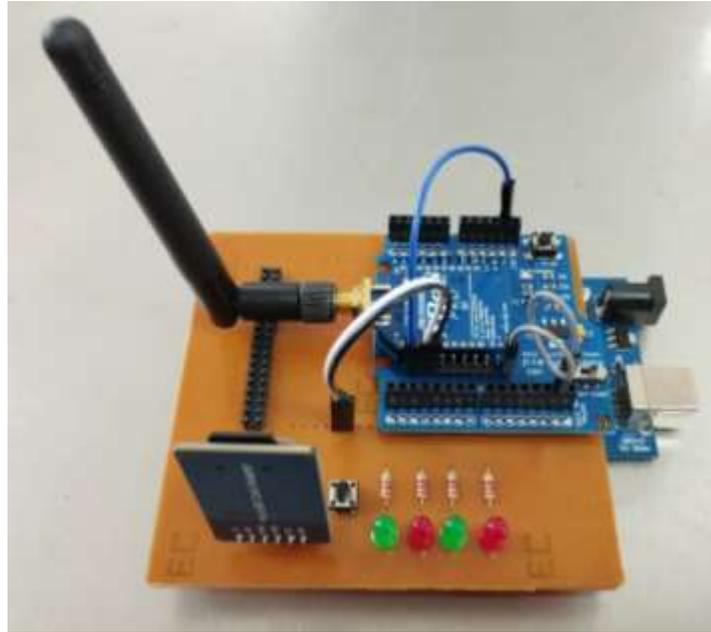
Una de las funciones del Xbee coordinador junto con Arduino es informar el buen funcionamiento de todos los dispositivos de la red, por ende, en este caso se optó por realizar un montaje en PCB teniendo como función acoplar de manera correcta el Xbee coordinador, Arduino, pulsador de inicio, micro SD y los leds de estado que indican el buen funcionamiento de todos los dispositivos o indica alguna anomalía que se puede presentar.

Figura 63. Diseño de PCB



Luego se implementó los dispositivos y componentes en la placa PCB diseñada como lo ilustra

Figura 64. Implementación de los dispositivos y componentes en la placa PCB diseñada (lado superior)



También se tiene en cuenta que el diseño de la PCB tiene componentes en el lado superior como se ilustra en la figura 26 en la que se visualiza el adaptador USB, leds de estado, pulsador de inicio y el ZigBee coordinador acoplado con el Shield Xbee PRO que a su vez este se acopla con el Arduino desde la parte inferior de la PCB como se ilustra en la

Figura 65. Lado inferior placa PCB.



7.3 Código Arduino para la recolección de datos dentro de la Cancha #1 de la

Universidad CESMAG

```

1  #include <SPI.h> // incluye libreria interfaz SPI
2  #include <SD.h> // Incluye libreria SD
3
4  // Librería tarjeta SD
5  #define SSpin 53 // pin CS Arduino mega
6  #define NOM_ARCHIVO "tesis.csv"
7  File archivo; //Crea un archivo
8
9  // Estados del sistema
10 #define EST_INICIAL 1
11 #define EST_MIDIENDO 2
12 #define EST_ESCRIBIENDO 3
13 #define EST_ESPERA_BOTON 4
14 #define EST_ERROR 5
15 #define EST_ERROR_SD 6
16
17 // Pines para LEDs
18 #define PIN_LED_ROJO_1 41
19 #define PIN_LED_VERDE_1 43
20 #define PIN_LED_ROJO_2 45
21 #define PIN_LED_VERDE_2 47
22
23 // Pines para botones
24 #define PIN_BOTON 49
25
26 // Número de ciclos de lectura
27 #define NUM_CICLOS 120
28
29 // Configuraciones XBee
30 #define XBEE1 0x00,0x02,0x1D
31 #define XBEE2 0x00,0x03,0x1C
32 #define XBEE3 0x00,0x04,0x1B
33 #define XBEE4 0x00,0x05,0x1A
34
35 // Variables globales
36 int estado = EST_INICIAL;
37 int i_ciclo = 0;
38 int medidas[4];
39
40
41 void setup()
42 {
43   // Inicializar leds y botón
44   InicializarLeds();
45
46   // Actualizamos estado

```

```

47 ActualizarEstado(EST_INICIAL);
48
49 // Inicializar serial xbee
50 // Inicializar serial debug
51 InicializarSeriales();
52
53 // Inicializar librería SD
54 int sd_return = InicializarLibreriaSd();
55 //Serial.print("Inicializacion SD: "); Serial.println(sd_return);
56
57 // Actualizamos estado
58 if( sd_return != 0 )
59     ActualizarEstado(EST_ERROR_SD);
60 else
61     ActualizarEstado(EST_ESPERA_BOTON);
62 }
63
64
65 void loop()
66 {
67     switch(estado)
68     {
69         // Si hay un error, lo atendemos
70         case EST_ERROR:
71         case EST_ERROR_SD:
72             break;
73
74         case EST_ESPERA_BOTON:
75             //Serial.println("Esperando botón");
76             if( digitalRead(PIN_BOTON) == LOW )
77             {
78                 InicializarLibreriaSd();
79                 ActualizarEstado(EST_MIDIENDO);
80                 i_ciclo = 0;
81             }
82             break;
83
84         case EST_MIDIENDO:
85             // La primera vez se abre el archivo
86             if(i_ciclo == 0)
87             {
88                 archivo = SD.open(NOM_ARCHIVO, FILE_WRITE);
89                 archivo.println("XBee1, XBee2, Xbee3, XBee4");
90             }
91
92             // tomar medidas
93             TomarMedidas();
94
95             // guardar medidas
96             GuardarMedidas();
97
98             // actualizar el contador de ciclos

```

```

99     i_ciclo++;
100
101     // Cambiar estado cuando lleguemos a num_ciclos
102     if(i_ciclo == NUM_CICLOS)
103     {
104         archivo.close();
105         ActualizarEstado(EST_ESPERA_BOTON);
106     }
107     break;
108
109 }
110
111 delay(100);
112 }
113
114
115 void InicializarLeds()
116 {
117     pinMode(PIN_LED_ROJO_1, OUTPUT);
118     pinMode(PIN_LED_VERDE_1, OUTPUT);
119     pinMode(PIN_LED_ROJO_2, OUTPUT);
120     pinMode(PIN_LED_VERDE_2, OUTPUT);
121     pinMode(PIN_BOTON, INPUT_PULLUP);
122 }
123
124 void InicializarSeriales()
125 {
126     Serial.begin(9600);
127     Serial2.begin(9600);
128     while(!Serial){}
129     while(!Serial2){}
130 }
131
132 int InicializarLibreriaSd()
133 {
134     // Inicializamos la librería
135     if( !SD.begin(SSpin) )
136         return -1;
137
138     // Verificamos si existe una tarjeta SD y si no está bloqueada
139     if( !SD.exists(NOM_ARCHIVO) )
140     {
141         archivo = SD.open(NOM_ARCHIVO, FILE_WRITE);
142         if (archivo)
143         {
144             archivo.println("XBee1, XBee2, Xbee3, XBee4");
145             archivo.close();
146         }
147         else return -2;
148     }
149
150     // Remover el archivo de la memoria

```

```

151  if( !SD.remove(NOM_ARCHIVO) )
152      return -3;
153
154  return 0;
155 }
156
157 void ActualizarEstado(int estadoNuevo)
158 {
159     estado = estadoNuevo;
160     switch(estadoNuevo)
161     {
162         case EST_INICIAL:
163             digitalWrite(PIN_LED_ROJO_1, LOW);
164             digitalWrite(PIN_LED_VERDE_1, LOW);
165             digitalWrite(PIN_LED_ROJO_2, LOW);
166             digitalWrite(PIN_LED_VERDE_2, HIGH);
167             break;
168         case EST_MIDIENDO:
169             digitalWrite(PIN_LED_ROJO_1, LOW);
170             digitalWrite(PIN_LED_VERDE_1, LOW);
171             digitalWrite(PIN_LED_ROJO_2, HIGH);
172             digitalWrite(PIN_LED_VERDE_2, LOW);
173             break;
174         case EST_ESCRIBIENDO:
175             digitalWrite(PIN_LED_ROJO_1, LOW);
176             digitalWrite(PIN_LED_VERDE_1, LOW);
177             digitalWrite(PIN_LED_ROJO_2, HIGH);
178             digitalWrite(PIN_LED_VERDE_2, HIGH);
179             break;
180         case EST_ESPERA_BOTON:
181             digitalWrite(PIN_LED_ROJO_1, LOW);
182             digitalWrite(PIN_LED_VERDE_1, HIGH);
183             digitalWrite(PIN_LED_ROJO_2, LOW);
184             digitalWrite(PIN_LED_VERDE_2, LOW);
185             break;
186         case EST_ERROR_SD:
187             digitalWrite(PIN_LED_ROJO_1, HIGH);
188             digitalWrite(PIN_LED_VERDE_1, HIGH);
189             digitalWrite(PIN_LED_ROJO_2, HIGH);
190             digitalWrite(PIN_LED_VERDE_2, LOW);
191             break;
192         default: // estado de error
193             digitalWrite(PIN_LED_ROJO_1, HIGH);
194             digitalWrite(PIN_LED_VERDE_1, HIGH);
195             digitalWrite(PIN_LED_ROJO_2, HIGH);
196             digitalWrite(PIN_LED_VERDE_2, HIGH);
197             break;
198     }
199 }
200
201 // Función para requerir la potencia de 4 xbees
202 void TomarMedidas()

```

```

203 {
204   medidas[0] = RequerirPotencia(XBEE1);
205   medidas[1] = RequerirPotencia(XBEE2);
206   medidas[2] = RequerirPotencia(XBEE3);
207   medidas[3] = RequerirPotencia(XBEE4);
208
209   bool todasOk = true;
210   for(int i=0; i<4; i++)
211     if(medidas[i] < 0)
212       todasOk = false;
213
214   if( !todasOk )
215   {
216     digitalWrite(PIN_LED_ROJO_1, HIGH);
217     digitalWrite(PIN_LED_ROJO_2, HIGH);
218     digitalWrite(PIN_LED_VERDE_1, LOW);
219     digitalWrite(PIN_LED_VERDE_2, LOW);
220   }
221   else
222   {
223     digitalWrite(PIN_LED_ROJO_1, LOW);
224     digitalWrite(PIN_LED_ROJO_2, LOW);
225     digitalWrite(PIN_LED_VERDE_1, HIGH);
226     digitalWrite(PIN_LED_VERDE_2, HIGH);
227   }
228 }
229
230 //Funcion para requerir la potencia de cada uno de los XBee
231 int RequerirPotencia(byte idRec1, byte idRec2, byte checksum)
232 {
233   // Variable recepción de mensajes
234   byte inbuf[20];
235
236   // Ajuste del mensaje
237   byte mensaje[] = {0x7E, 0x00, 0x0F, 0x17, 0x01, 0x00, 0x00, 0x00, 0x00,
238 0x00, 0x00, 0x00, 0x00, idRec1, idRec2, 0x02, 0x64, 0x62, checksum};
239
240   // Envío del mensaje al módulo correspondiente
241   int sent = Serial2.write(mensaje, 19);
242   delay(200);
243
244   // Recepción de respuesta
245   int read = 0;
246   while(Serial2.available() > 0)
247   {
248     if(read < 20) inbuf[read] = Serial2.read();
249     read++;
250   }
251
252   // Mensaje de depuración
253   //Serial.print("Enviados: "); Serial.print(sent); Serial.print(" bytes.
254 ");

```

```

255 //Serial.print("Recibidos: "); Serial.print(read); Serial.print(" bytes.
256 ");
257
258 // Análisis de respuesta
259 if(read < 19)
260 {
261     // Se descarta el resto del mensaje
262     delay(100);
263     while(Serial2.available() > 0) Serial2.read();
264     return -1;
265 }
266 else if(read == 19)
267 {
268     // Verificar si hubo un error
269     if( inbuf[17] == 0x00 )
270         return -2;
271     else
272         return -3;
273 }
274 else if(read == 20)
275     return int(inbuf[18]);
276 else
277     return -4;
278 }
279
280 // Funcion para guardar medidas
281 int GuardarMedidas()
282 {
283     for(int i=0; i<4; i++)
284     {
285         archivo.print(medidas[i]);
286         archivo.print(",");
287     }
288     archivo.println("");
289 }

```

Para acceder al archivo de código de Arduino, consulte el archivo adjunto a este documento que contiene los códigos del proyecto. Navegue a la carpeta “Códigos Proyecto de Grado” y luego a la carpeta número 2, donde se encuentran los códigos de Arduino utilizados en toda la elaboración del proyecto.

7.4 Códigos Matlab para el procesamiento de datos, entrenamiento de modelo ajustado y la trilateración.

```

1  %Modelo de Propagacion Para XBee 1
2  clear; clc; close all;
3  %Abrir datos para trabajar
4  datos = load('XBeel_Filtrado.mat').XBeel;
5
6  whos datos
7
8  %Plotear potencia real vs distancia logaritmica
9  figure
10 stem(datos(:,6), datos(:,4), 'LineStyle', 'None');
11 title("Dispersión de datos XBee 1 (14 dBm)");
12 xlabel("Distancia Logaritmica (10*log10(do/d))");
13 ylabel("Potencia Real (dBm)");
14 hold off;
15
16 % Filtrar las filas según la condición
17 condicion1 = (datos(:, 6) > -13);
18 condicion2 = (datos(:, 6) < -14);
19
20 %Crear las nuevas matrices
21 matriz_modelo_mayores = datos(condicion1, :);
22 matriz_modelo_menores = datos(condicion2, :);
23
24 %Graficar datos de las matrices resultantes
25 figure
26 stem(matriz_modelo_mayores(:,6), matriz_modelo_mayores(:,4), 'LineStyle',
27 'None');
28 title("Dis Log Vs Pot Medida (Valores mayores)");
29 xlabel("Distancia Logaritmica");
30 ylabel("Potencia Real");
31 hold off;
32 figure
33 stem(matriz_modelo_menores(:,6), matriz_modelo_menores(:,4), 'LineStyle',
34 'None');
35 title("Dis Log Vs Pot Medida (Valores menores)");
36 xlabel("Distancia Logaritmica");
37 ylabel("Potencia Real");
38 hold off;
39
40 %Modelo para valores mayores y Grafico del modelo 1 (mayores)
41 x1 = min(matriz_modelo_mayores(:, 6)):0.1:max(matriz_modelo_mayores(:,
42 6));
43 mod1_valores_mayores = fitlm(matriz_modelo_mayores(:, 6),
44 matriz_modelo_mayores(:, 4));
45 b1 = mod1_valores_mayores.Coefficients.Estimate(1);
46 m1 = mod1_valores_mayores.Coefficients.Estimate(2);
47 y1 = m1 * x1 + b1;
48

```

```

49 figure
50 plot(x1, y1, 'r');
51 hold on;
52 stem(matriz_modelo_mayores(:, 6),matriz_modelo_mayores(:, 4),
53 'LineStyle', 'None');
54 grid on;
55 legend("datos", "n=" + m1);
56 title("Modelo para distancias logaritmicas mayores a -7");
57 xlabel("Distancia Logaritmica");
58 ylabel("Potencia Real (dBm)");
59 hold off;
60
61 %Modelo para valores menores y Grafico del modelo 2 (menores)
62 x2 = min(matriz_modelo_menores(:, 6)):0.1:max(matriz_modelo_menores(:,
63 6));
64 model2_valores_menores = fitlm(matriz_modelo_menores(:, 6),
65 matriz_modelo_menores(:, 4));
66 b2 = model2_valores_menores.Coefficients.Estimate(1);
67 m2 = model2_valores_menores.Coefficients.Estimate(2);
68 y2 = m2 * x2 + b2;
69
70 figure
71 plot(x2, y2, 'r');
72 hold on;
73 stem(matriz_modelo_menores(:, 6), matriz_modelo_menores(:, 4),
74 'LineStyle', 'None');
75 grid on;
76 legend("datos", "n=" + m2);
77 title("Modelo para distancias logaritmicas menores a -8");
78 xlabel("Distancia Logaritmica");
79 ylabel("Potencia Real (dBm)");
80 hold off;
81
82 %Modelo para todos los datos
83
84 x3 = min(datos(:, 6)):0.1:max(datos(:, 6));
85 model3_total = fitlm(datos(:, 6), datos(:, 4));
86 b3 = model3_total.Coefficients.Estimate(1);
87 m3 = model3_total.Coefficients.Estimate(2);
88 y3 = m3 * x3 + b3;
89 y4_mayores = m1 * x3 + b1;
90 y5_menores = m2 * x3 + b2;
91 figure
92 % plot(x3, y3, 'r');
93 plot(x3, y4_mayores, 'r');
94 hold on;
95 plot(x3, y5_menores, 'black');
96 hold on;
97 stem(datos(:, 6), datos(:, 4), 'LineStyle', 'None', 'MarkerEdgeColor',
98 'blue' );
99 grid on;
100 legend("n1=" + m1, "n2=" + m2);

```

```

101 title("Interseccion de los modelos para el XBee 1 (14dBm)");
102 xlabel("Distancia Logaritmica (10*log10(d0/d))");
103 ylabel("Potencia Real (dBm)");
104 hold off;
105 %Buscar la interseccion entre los puntos
106 Xa = ((b2 - b1) / (m1-m2));
107 Ya = (m1 * Xa + b1);

```

Debido a que en cada estudio se requirió ajustar el modelo para cada dispositivo final y procesar datos con códigos distintos, no es posible incluir toda esta información en el documento. Los códigos completos, organizados por carpetas, se encuentran en el archivo adjunto al proyecto. En la carpeta "Códigos Proyecto de Grado", dentro de la carpeta 1, se encuentran todos los códigos utilizados para la realización del proyecto, organizados en subcarpetas.

7.5 Código de Python para la obtención de datos por puerto serial y predicción de las coordenadas

```

1 import serial
2 import tensorflow as tf
3 import numpy as np
4
5 # Registrar la función de perdida debido a un error en la carga del modelo
6 h5
7 @tf.keras.utils.register_keras_serializable()
8 def mse(y_true, y_pred):
9     return tf.reduce_mean(tf.square(y_true - y_pred))
10
11 # Ruta del archivo del modelo
12 model_path = 'C:/Users/urbin/OneDrive/Escritorio/Arduino y python
13 TESIS/Python_Tesis/modelo_keras_XBee.h5'
14
15 # Cargar el modelo entrenado
16 model = tf.keras.models.load_model(model_path, custom_objects={'mse':
17 mse})
18
19 # Mostrar las características del modelo
20 model.summary()
21
22 # Configurar la conexión serial (ajustar el puerto y la velocidad según
23 sea necesario)
24 ser = serial.Serial('COM6', 9600)
25
26 while True:
27     try:

```

```

28     # Leer una línea de datos del puerto serial
29     line = ser.readline().decode('utf-8').strip()
30     # Dividir la línea en una lista de valores
31     values = list(map(int, line.split(',')))
32
33     if len(values) == 4:
34         # Convertir los valores en un array de NumPy
35         input_data = np.array([values])
36
37         # Realizar predicciones usando el modelo
38         predictions = model.predict(input_data)
39
40         # Etiquetar las predicciones
41         pred_x = predictions[0][0]
42         pred_y = predictions[0][1]
43
44         # Mostrar las predicciones con etiquetas
45         print(f'X(mts): {pred_x}, Y(mts): {pred_y}')
46     else:
47         print(f'Error: Se esperaban 4 valores, pero se recibieron
48 {len(values)} valores')
49     except Exception as e:
50         print(f'Error al leer el puerto serial: {e}')
51         break
52
53 ser.close()

```

Para acceder al archivo del modelo entrenado y ejecutar el código, consulte la carpeta adjunta al documento en "Códigos Proyecto de Grado" y luego en la carpeta 4. Dentro, se encuentran los códigos de las pruebas realizadas, el proyecto se ejecutó en Python 3.12.3 con la ayuda de Visual Studio Code.

7.6 Entrenamiento y datos para el modelo de Machine Learning

El entrenamiento del algoritmo de Machine Learning se realizó en Google Colab, donde también se llevaron a cabo las pruebas y se obtuvo el archivo del modelo entrenado para la aplicación en Python, desarrollada con la ayuda de Visual Studio Code. Los datos y el enlace al proyecto de Google Colab se pueden consultar en el archivo adjunto al documento, en la carpeta "Códigos Proyecto de Grado", específicamente en la carpeta 3.

 UNIVERSIDAD CESMAG <small>MIT: 000.109.207-7 VIGILANCIA MINISTERIO</small>	CARTA DE ENTREGA TRABAJO DE GRADO O TRABAJO DE APLICACIÓN – ASESOR(A)	CÓDIGO: AAC-BL-FR-032
		VERSIÓN: 1
		FECHA: 09/JUN/2022

San Juan de Pasto, 10 de septiembre de 2024

Biblioteca
REMIGIO FIORE FORTEZZA OFM. CAP.
Universidad CESMAG
Pasto

Saludo de paz y bien.

Por medio de la presente se hace entrega del Trabajo de Grado / Trabajo de Aplicación denominado, Estimación de la precisión de dispositivos ZigBee para el posicionamiento en una cancha de microfútbol presentado por el (los) autor(es) Yinner Jaider López Puerres y Edwin Andrés Urbina De la Cruz del Programa Académico Ingeniería Electrónica al correo electrónico biblioteca.trabajosdegrado@unicesmag.edu.co. Manifiesto como asesor(a), que su contenido, resumen, anexos y formato PDF cumple con las especificaciones de calidad, guía de presentación de Trabajos de Grado o de Aplicación, establecidos por la Universidad CESMAG, por lo tanto, se solicita la paz y salvo respectivo.

Atentamente,



Mario Fernando Henao Rosero
Numero de documento: 4.612.719
Programa académico: Ingeniería Electrónica
Teléfono de contacto: 316 345 5039
Correo electrónico: mfhenao@unicesmag.edu.co

 UNIVERSIDAD CESMAG <small>NI: 600.109.387-7 NÚMERO DE REGISTRO</small>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

INFORMACIÓN DEL (LOS) AUTOR(ES)	
Nombres y apellidos del autor: Yinner Jaidier López Puerres	Documento de identidad: 1004410331
Correo electrónico: yinner2288@gmail.com	Número de contacto: 3176516920
Nombres y apellidos del autor: Edwin Andrés Urbina De la Cruz	Documento de identidad: 1007301425
Correo electrónico: urbinaedwin71@gmail.com	Número de contacto: 3234360123
Nombres y apellidos del autor:	Documento de identidad:
Correo electrónico:	Número de contacto:
Nombres y apellidos del autor:	Documento de identidad:
Correo electrónico:	Número de contacto:
Nombres y apellidos del asesor: Mario Fernando Henao Rosero	Documento de identidad: 4612719
Correo electrónico: mfhenao@unicesmag.edu.co	Número de contacto: 3163455039
Título del trabajo de grado: Estimación de la Precisión de Dispositivos ZigBee Para el Posicionamiento en una Cancha de Microfútbol	
Facultad y Programa Académico: Facultad de Ingeniería – Ingeniería Electrónica	

En mi (nuestra) calidad de autor(es) y/o titular (es) del derecho de autor del Trabajo de Grado o de Aplicación señalado en el encabezado, confiero (conferimos) a la Universidad CESMAG una licencia no exclusiva, limitada y gratuita, para la inclusión del trabajo de grado en el repositorio institucional. Por consiguiente, el alcance de la licencia que se otorga a través del presente documento, abarca las siguientes características:

- a) La autorización se otorga desde la fecha de suscripción del presente documento y durante todo el término en el que el (los) firmante(s) del presente documento conserve(mos) la titularidad de los derechos patrimoniales de autor. En el evento en el que deje(mos) de tener la titularidad de los derechos patrimoniales sobre el Trabajo de Grado o de Aplicación, me (nos) comprometo (comprometemos) a informar de manera inmediata sobre dicha situación a la Universidad CESMAG. Por consiguiente, hasta que no exista comunicación escrita de mi(nuestra) parte informando sobre dicha situación, la Universidad CESMAG se encontrará debidamente habilitada para continuar con la publicación del Trabajo de Grado o de Aplicación dentro del repositorio institucional. Conozco(conocemos) que esta autorización podrá revocarse en cualquier momento, siempre y cuando se eleve la solicitud por escrito para dicho fin ante la Universidad CESMAG. En estos eventos, la Universidad CESMAG cuenta con el plazo de un mes después de recibida la petición, para desmarcar la visualización del Trabajo de Grado o de Aplicación del repositorio institucional.

 UNIVERSIDAD CESMAG <small>NI: 600.109.387-7</small> <small>NOVA UNIVERZITAT</small>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

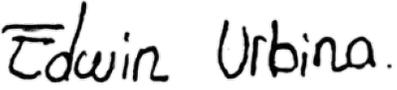
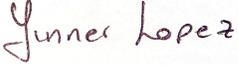
- b) Se autoriza a la Universidad CESMAG para publicar el Trabajo de Grado o de Aplicación en formato digital y teniendo en cuenta que uno de los medios de publicación del repositorio institucional es el internet, acepto(amos) que el Trabajo de Grado o de Aplicación circulará con un alcance mundial.
- c) Acepto (aceptamos) que la autorización que se otorga a través del presente documento se realiza a título gratuito, por lo tanto, renuncio(amos) a recibir emolumento alguno por la publicación, distribución, comunicación pública y/o cualquier otro uso que se haga en los términos de la presente autorización y de la licencia o programa a través del cual sea publicado el Trabajo de grado o de Aplicación.
- d) Manifiesto (manifestamos) que el Trabajo de Grado o de Aplicación es original realizado sin violar o usurpar derechos de autor de terceros y que ostento(amos) los derechos patrimoniales de autor sobre la misma. Por consiguiente, asumo(asumimos) toda la responsabilidad sobre su contenido ante la Universidad CESMAG y frente a terceros, manteniéndola indemne de cualquier reclamación que surja en virtud de la misma. En todo caso, la Universidad CESMAG se compromete a indicar siempre la autoría del escrito incluyendo nombre de(los) autor(es) y la fecha de publicación.
- e) Autorizo(autorizamos) a la Universidad CESMAG para incluir el Trabajo de Grado o de Aplicación en los índices y buscadores que se estimen necesarios para promover su difusión. Así mismo autorizo (autorizamos) a la Universidad CESMAG para que pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

NOTA: En los eventos en los que el trabajo de grado o de aplicación haya sido trabajado con el apoyo o patrocinio de una agencia, organización o cualquier otra entidad diferente a la Universidad CESMAG. Como autor(es) garantizo(amos) que he(hemos) cumplido con los derechos y obligaciones asumidos con dicha entidad y como consecuencia de ello dejo(dejamos) constancia que la autorización que se concede a través del presente escrito no interfiere ni transgrede derechos de terceros.

Como consecuencia de lo anterior, autorizo(autorizamos) la publicación, difusión, consulta y uso del Trabajo de Grado o de Aplicación por parte de la Universidad CESMAG y sus usuarios así:

- Permiso(permitimos) que mi(nuestro) Trabajo de Grado o de Aplicación haga parte del catálogo de colección del repositorio digital de la Universidad CESMAG por lo tanto, su contenido será de acceso abierto donde podrá ser consultado, descargado y compartido con otras personas, siempre que se reconozca su autoría o reconocimiento con fines no comerciales.

En señal de conformidad, se suscribe este documento en San Juan de Pasto a los 11 días del mes de septiembre del año 2024

	
<small>Firma del autor</small>	<small>Firma del autor</small>
Nombre del autor: Edwin Andrés Urbina De la Cruz	Nombre del autor: Yinner Jaider López Puerres
 <small>Firma del autor</small>	 <small>Firma del autor</small>
Nombre del autor:	Nombre del autor:
 <hr/> Nombre del asesor: Mario Fernando Henao Rosero	