



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**USO DE “WEARABLES” PARA FACILITAR EL  
DISTANCIAMIENTO SOCIAL EN AMBIENTES LABORALES:  
ESTUDIO DE FACTIBILIDAD**

**TRABAJO DE TITULACIÓN**

Previa a la obtención del Título de:

**MAGISTER EN TELECOMUNICACIONES**

Presentado por:

**MARICELA ELISET FREIRE BARBA**

**JAIME JAIR TORRES FRANCO**

**GUAYAQUIL – ECUADOR**

**AÑO: 2020**

## AGRADECIMIENTO

Gracias a la Santísima Trinidad por regalarme aquellos dones que me han permitido culminar este trabajo y sobretodo está etapa universitaria. Gracias a mi Virgen amada por acompañarme todos los días durante este camino, de su mano todo ha sido más fácil. Gracias a mi Ángel Custodio, a San José, a Santa Teresita del Niño Jesús y a San Juan Pablo II por su intercesión en el cielo durante todo el desarrollo de esta tesis. Gracias a mis padres por su apoyo constante, entrega, amor y guía durante mi vida. Gracias a mis compañeros por sus enseñanzas y anécdotas vividas durante nuestros estudios en las aulas, y a mi compañero Jair Torres por su ayuda en este trabajo. Gracias al Dr. Federico Domínguez por su guía y consejos para desarrollar esta tesis. Quiero terminar por agradecer a una personita que he conocido en estos últimos años, quien con tan solo un abrazo o un mensaje siempre me motivaba a continuar y a no desistir, gracias mi vida Aristóteles Amat.

“Dios no puede inspirar deseos irrealizables” Santa Teresita del Niño Jesús

**Maricela Freire Barba**

Agradezco a Dios por su cuidado durante todo el camino de pandemia, superando obstáculos y tomando la mejor decisión en estos tiempos. A mi esposa que me acompañó durante todo el recorrido de malas noches, compartiendo alegrías y frustraciones. A mi hija por darme la fuerza para no rendirme y ser ella la motivación que tengo para seguir adelante. A mis padres que siempre están ahí, dando un consejo, apoyándome en las buenas y en las malas, los amo demasiados, a todos. A todos mis amigos de la MET VI, a todos ellos les deseo muchos éxitos en las etapas que venideras, a los profesores por sus enseñanzas a lo largo de la maestría.

Gracias por todo

**Jaime Jair Torres Franco**

## DEDICATORIA

Dedico este trabajo a la Virgen María, aquella mujer perfecta que lo he hecho todo en mi vida y a la que le he entregado todo para que lo lleve a su hijo Jesús. A mis padres Pío Freire y Gloria Barba por su ayuda y paciencia durante todo este tiempo. A mis estudiantes y a aquellas personas que Dios ha puesto en mi camino para que de una u otra manera me muestren su ayuda o su apoyo a través de pequeños gestos de amor. A mi enamorado por su enorme paciencia en acompañarme durante este camino de estudios y motivarme a continuar con este trabajo final.

**Maricela Freire Barba**

Dedico este trabajo en primer lugar a Dios, por permitirme tener salud en estos tiempos difíciles de pandemia. A mi familia: mi hija por ser uno de los pilares de mi vida y llenarme de felicidad. A mi esposa por su paciencia, serenidad y por su gran apoyo. Y a mis padres por la ayuda que me han dado en todo este hermoso trayecto de aprendizaje. A mis amigos José C., Diego P. y José F. y Maricela F., gracias al grupo que se formó se llegó hasta la meta, y que hasta el momento seguimos siendo grandes amigos. Al Dr. Federico Domínguez por su tiempo y guía que nos ofreció en todo el camino del desarrollo de la tesis.

**Jaime Jair Torres Franco**

## TRIBUNAL DE EVALUACIÓN

Ph.D. María Antonieta Álvarez Villanueva

**SUBDECANA DE LA FIEC**



Ph.D. Federico Xavier Domínguez Bonini

**DIRECTOR DEL TRABAJO DE TITULACIÓN**



Ph.D. Juan Carlos Avilés Castillo

**MIEMBRO PRINCIPAL DEL TRIBUNAL**

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

MARICELA ELISET FREIRE BARBA  
Firmado digitalmente por  
MARICELA ELISET FREIRE BARBA  
Fecha: 2021.04.21 14:03:42  
-05'00'

-----  
Maricela Freire Barba

  
-----  
Jair Torres Franco

## RESUMEN

El mundo se enfrenta a una pandemia nunca antes experimentada en los tiempos modernos. El COVID-19 es una enfermedad viral que se ha extendido exponencialmente por todo el mundo, ocasionando así que muchos países adoptaran políticas de permanencia en casa o de bloqueo para controlar su propagación. Sin embargo, la estadía en casa prolongada puede causar efectos negativos como crisis económicas, desempleo, escasez de alimentos y problemas de salud mental de las personas.

Para detener la propagación del virus, se han recomendado medidas de etiqueta respiratoria, distanciamiento social e higiene de manos. En base en estas medidas, algunos países ya han logrado controlar la propagación del COVID-19 pero siguen en la búsqueda de métodos o protocolos tal como el distanciamiento social para garantizar un retorno seguro a las actividades laborales. Este trabajo presenta un análisis de factibilidad que permita al mercado nacional, el desarrollo de pequeños dispositivos electrónicos “wearables” para el control del distanciamiento social en ambientes laborales.

Para crear una solución óptima, en primera instancia se realizará un estudio mediante simulaciones aplicando técnicas de determinación de distancias (ranging) usando diferentes tecnologías inalámbricas tales como: UWB, Bluetooth LE, WiFi, RFID y NFC; posteriormente se evaluarán costos y componentes disponibles en el mercado para la construcción de los “wearables”, que permitan alertar a dos trabajadores que se encuentren a menos de 2 metros de distancia. Cabe mencionar que al monitorear la distancia entre 2 dispositivos en lugar de su ubicación se está logrando respetar la privacidad de los usuarios.

El documento está organizado de la siguiente manera: en el capítulo 1 se establecen los antecedentes y motivación de la propuesta planteada. En el capítulo 2 se describen las tecnologías inalámbricas aplicadas al distanciamiento social, así como las técnicas de ranging en ambientes interiores y un estudio de mercado de los diferentes dispositivos para distanciamiento social utilizados en el ámbito nacional e internacional. En el capítulo 3 se explica la arquitectura del sistema y el algoritmo de estimación de distancia para las diferentes tecnologías inalámbricas. En el capítulo 4 se muestran los resultados de estimación de distancia obtenidas mediante simulaciones junto con los cálculos teóricos. En el capítulo 5 se realiza el análisis de factibilidad en base a exactitud, escalabilidad, costos y componentes disponibles en el mercado para la implementación de los “wearables” en el mercado

nacional. Finalmente se muestran las conclusiones y recomendaciones de la investigación.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	i
DEDICATORIA .....	ii
TRIBUNAL DE EVALUACIÓN .....	iii
RESUMEN .....	v
ÍNDICE GENERAL.....	vi
ABREVIATURAS .....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS .....	xi
CAPÍTULO 1.....	1
1 ANTECEDENTES.....	1
<b>1.1 Descripción del problema</b> .....	1
<b>1.2 Justificación</b> .....	2
<b>1.3 Objetivos</b> .....	4
<b>1.3.1. Objetivo general</b> .....	4
<b>1.3.2. Objetivos específicos</b> .....	4
<b>1.4 Metodología</b> .....	4
<b>1.5 Resultados esperados</b> .....	5
CAPÍTULO 2.....	6
2 ESTADO DEL ARTE DE TECNOLOGÍAS DE POSICIONAMIENTO EN INTERIORES.....	6
<b>2.1 Introducción</b> .....	6

2.2	<b>Tecnologías inalámbricas para distanciamiento social</b> .....	6
2.2.1	WiFi .....	6
2.2.2	Bluetooth LE (BLE) .....	8
2.2.3	Identificación por radio frecuencia (RFID) .....	12
2.2.4	Banda ultra ancha (UWB) .....	14
2.2.5	Comunicación de campo cercano (NFC) .....	15
2.3	<b>Técnicas de ranging en ambientes interiores</b> .....	16
2.3.1	Indicador de fuerza de la señal recibida (RSSI) .....	16
2.3.2	Diferencia de tiempo de arribo (TDoA) .....	17
2.3.3	Ángulo de arribo (AoA) .....	18
2.3.4	Tiempo de arribo (ToA) .....	18
2.3.5	Tiempo de vuelo (ToF) .....	19
2.4	<b>Modelo para calcular el RSSI en ambientes interiores</b> .....	20
2.5	<b>Procesamiento de la señal RSSI</b> .....	22
2.5.1	Filtrado Gaussiano .....	22
2.5.2	Filtrado de Kalman .....	23
2.6	<b>Estudio de mercado de dispositivos de rastreo o distanciamiento social en el ámbito internacional y nacional</b> .....	24
CAPÍTULO 3 .....		30
3	<b>DISEÑO DEL WEARABLE PARA DISTANCIAMIENTO SOCIAL</b> .....	30
3.1	<b>Requerimientos de la implementación de wearables para distanciamiento social</b> .....	31
3.1.1	Software (Simulador OMNeT++) .....	31
3.1.2	Hardware .....	35



<b>3.2</b>	<b>Arquitectura del sistema usando las diferentes tecnologías inalámbricas.....</b>	<b>38</b>
<b>3.3</b>	<b>Algoritmos de estimación de distancia para las diferentes tecnologías inalámbricas.....</b>	<b>42</b>
CAPÍTULO 4.....		50
4	ANÁLISIS DE RESULTADOS.....	50
<b>4.1</b>	<b>Entorno de pruebas.....</b>	<b>50</b>
<b>4.2</b>	<b>Estimación de distancias obtenidas mediante simulaciones.....</b>	<b>51</b>
CAPÍTULO 5.....		60
5	ANÁLISIS DE FACTIBILIDAD DEL DISEÑO PROPUESTO.....	60
<b>5.1</b>	<b>Análisis comparativo de las tecnologías de posicionamiento en interiores en función de la exactitud.....</b>	<b>60</b>
<b>5.2</b>	<b>Análisis comparativo de las tecnologías de posicionamiento en interiores en función de la escalabilidad.....</b>	<b>61</b>
<b>5.3</b>	<b>Análisis comparativo de las tecnologías de posicionamiento en interiores en función de costos.....</b>	<b>62</b>
<b>5.4</b>	<b>Análisis comparativo de las tecnologías de posicionamiento en interiores en función de componentes disponibles en el mercado ....</b>	<b>65</b>
<b>5.5</b>	<b>Tabla comparativa de las tecnologías de posicionamiento en interiores para la implementación de wearables.....</b>	<b>66</b>
CONCLUSIONES.....		68
RECOMENDACIONES.....		69
BIBLIOGRAFÍA.....		70
ÁPENDICES.....		74

## ABREVIATURAS

<b>COVID-19</b>	Coronavirus 2019
<b>OMS</b>	Organización Mundial de la Salud
<b>COE</b>	Comité de Operaciones de Emergencias
<b>UWB</b>	Banda Ultra Ancha
<b>BLE</b>	Bluetooth de baja energía
<b>WiFi</b>	Wireless Fidelity
<b>RFID</b>	Identificación por Radiofrecuencia
<b>NFC</b>	Comunicación de campo cercano
<b>RSSI</b>	Indicador de fuerza de señal recibida
<b>GPS</b>	Sistema de posicionamiento global
<b>MIMO</b>	Múltiple entrada – múltiple salida
<b>RS-232</b>	Estándar recomendado 232
<b>UHF</b>	Frecuencia ultra alta
<b>ISM</b>	Banda de frecuencia industrial, científica y médica
<b>GHz</b>	Gigahercio
<b>MHz</b>	Megahercio
<b>Mbps</b>	Mega bit por segundo
<b>mW</b>	Mili vatios
<b>SSID</b>	Secuencia de identificador de red
<b>GAP</b>	Perfil de acceso genérico
<b>PDU</b>	Unidad de paquete de datos
<b>CRC</b>	Verificación de redundancia cíclica
<b>AGC</b>	Control automático de ganancia
<b>FHSS</b>	Espectro ensanchado por salto de frecuencia
<b>RF</b>	Radiofrecuencia
<b>kHz</b>	Kilohercios
<b>TDoA</b>	Diferencia de tiempo de arribo
<b>AoA</b>	Ángulo de arribo
<b>NLOS</b>	Sin línea de vista
<b>ToA</b>	Tiempo de arribo
<b>ToF</b>	Tiempo de vuelo
<b>TWR</b>	Rango de dos vías
<b>RTT</b>	Tiempo de ida y vuelta
<b>dBm</b>	Decibelios
<b>HTTP</b>	Protocolo de transferencia de hipertexto
<b>MAC</b>	Control de acceso al medio
<b>PHY</b>	Física
<b>IoT</b>	Internet de las cosas
<b>GPIO</b>	Pines de propósito general de entrada y salida

## ÍNDICE DE FIGURAS

Figura 1.1. Escenarios para determinar la distancia entre dos personas .....	5
Figura 2.1. Localización con WiFi basada en RSSI [4] .....	7
Figura 2.2. Formato del paquete publicitario [11].....	11
Figura 2.3. Distanciamiento entre dos personas que usan BLE [12] .....	12
Figura 2.4. Sistema de conteo usando UWB [12] .....	14
Figura 2.5. Modos de comunicación: (a) Pasivo (b) Activo [14] .....	15
Figura 2.6. Envío de mensajes entre dos dispositivos UWB.....	20
Figura 3.1 Módulos del framework INET 4.2.1 .....	32
Figura 3.2 Placa microcontroladora ESP-01 [39].....	35
Figura 3.3. Pines GPIO del ESP-01 [39].....	36
Figura 3.4 Arduino Nano 33 BLE [40] .....	37
Figura 3.5. Tags para determinar el distanciamiento social.....	38
Figura 3.6. Arquitectura del sistema con tecnología UWB y RFID .....	39
Figura 3.7. Arquitectura del sistema con tecnología WiFi .....	40
Figura 3.8. Arquitectura del sistema con tecnología BLE .....	40
Figura 3.9. Diagrama de conexiones con el ESP-01 .....	41
Figura 3.10. Diagrama de conexiones con el Arduino Nano 33 BLE .....	42
Figura 4.1. Escenario de oficina para la simulación.....	50
Figura 4.2. Entorno de pruebas utilizando hardware .....	51
Figura 4.3. Fluctuaciones de RSSI en un ambiente indoor.....	52
Figura 4.4. RSSI vs. Distancia usando UWB .....	53
Figura 4.5. RSSI vs. Distancia usando RFID .....	53
Figura 4.6. RSSI vs. Distancia usando WiFi .....	54
Figura 4.7. RSSI vs. Distancia usando BLE.....	54
Figura 4.8. ToF vs. Distancia usando UWB .....	55
Figura 4.9. RSSI vs. Distancia para BLE usando hardware.....	58
Figura 4.10. RSSI vs. Distancia para WiFi usando hardware .....	58
Figura 4.11. ToF vs. Distancia usando UWB .....	59

## ÍNDICE DE TABLAS

Tabla 2.1 Estados de operaciones de BLE .....	10
Tabla 2.2 Exponentes path-loss para diferentes ambientes [18] .....	21
Tabla 2.3 Wearables disponibles en el mercado internacional. ....	24
Tabla 2.4 Aplicaciones móviles disponibles en el mercado internacional. ....	28
Tabla 4.1 Coeficientes de correlación entre distancia y RSSI .....	55
Tabla 4.2 Distancias estimadas y error usando BLE .....	56
Tabla 4.3 Distancias estimadas y error usando WiFi .....	57
Tabla 4.4 Distancias estimadas y error usando UWB .....	59
Tabla 5.1 Costos de materiales para implementación de wearable UWB ....	63
Tabla 5.2 Costos de personal para desarrollo de wearables .....	63
Tabla 5.3 Costos del wearable para tecnología UWB .....	64
Tabla 5.4 Costos del wearable para tecnología RFID .....	64
Tabla 5.5 Costos del wearable para tecnología WiFi .....	64
Tabla 5.6 Costos del wearable para tecnología BLE .....	65
Tabla 5.7 Módulo UWB .....	65
Tabla 5.8 Módulo RFID .....	66
Tabla 5.9 Módulo WiFi .....	66
Tabla 5.10 Módulo BLE .....	66
Tabla 5.11 Comparación entre tecnologías usadas en ambientes indoor. ...	67

# CAPÍTULO 1

## 1 ANTECEDENTES.

### 1.1 Descripción del problema

El 11 de marzo del 2020 la Organización Mundial de la Salud (OMS) declaró la enfermedad respiratoria COVID-19 como una pandemia global; el mismo día el gobierno de Ecuador declaró emergencia sanitaria en todo el país para priorizar la preparación del sistema de salud y mitigar la propagación de la enfermedad en la población. Dentro de este contexto, al desarrollarse la emergencia durante los meses de abril, mayo y junio el Comité de Operaciones de Emergencia (COE) Nacional dispuso diversas medidas de distanciamiento social en áreas de acceso público y en lugares de trabajo con el objetivo de minimizar las oportunidades de contagio de COVID-19. En la ciudad de Guayaquil, al igual que en muchos otros municipios en el Ecuador y el mundo, las autoridades dispusieron [1]: “Los trabajadores deberán mantener una distancia de 2 metros durante la entrada, permanencia y salida del trabajo, y el uso de mascarillas quirúrgicas durante la jornada laboral.”

En ciertos países y ciudades desplegaron drones con altavoces para comunicar las reglas y las precauciones obligatorias del gobierno con residentes. Los dispositivos portátiles se han ordenado a la población para aplicar medidas de cuarentena, mientras que los datos de telecomunicaciones se han utilizado para monitorear la gran cantidad de personas. Las corporaciones como Apple, Huawei, Fitbit y Samsung están desarrollando aplicaciones para el rastreo de personas que pueden alertar si alguien se pone en contacto con pacientes con diagnóstico reciente de COVID-19 [2].

Sin embargo, implementar medidas de distanciamiento social en ambientes laborales y espacios públicos es una tarea complicada debido a la dificultad de mantener consistentemente una distancia segura entre personas y a la incertidumbre que existe en las autoridades sanitarias y la comunidad científica sobre la eficacia o particularidades de las medidas adoptadas [3]. La

incertidumbre en la eficacia y en la metodología de una correcta implementación de distanciamiento social ha generado desconfianza en el público de dichas medidas, impidiendo un ordenado y seguro retorno a las actividades económicas.

El distanciamiento social en ambientes laborales puede ser facilitado usando tecnologías telemáticas de posicionamiento en interiores. Desde hace ya varios años existen opciones tecnológicas de bajo costo para rastrear inventario y personas dentro de edificios. El uso UWB, Bluetooth LE, RFID, NFC, WiFi/Bluetooth fingerprinting, entre otros, permiten rastrear inventario y personas con niveles de precisión desde varios metros hasta centímetros [4]. Existen ya iniciativas comerciales para crear dispositivos “wearables” (ejemplo: pulseras, tags) que facilitan la conformidad de medidas de distanciamiento social mediante la detección de personas a distancias configurables [5]. Sin embargo, aún no está claro que tecnología podría brindar una solución práctica en términos de costos, precisión y factibilidad de implementación en el contexto del Ecuador.

## **1.2 Justificación**

El brote del COVID-19 no tiene precedentes y ha interrumpido la vida de millones de personas en todo el mundo. Esta pandemia ha abierto desafíos y oportunidades de investigación de las diferentes tecnologías aplicadas al distanciamiento social, medida adoptada con el fin de contener la propagación de este virus altamente contagioso. Actualmente muchos países del mundo están evaluando la idea de desplegar ciudades inteligentes para albergar una gama de servicios basados en datos junto con sensores para ayudar a hacer cumplir la medida de distanciamiento social en toda la comunidad, así como para la detección temprana de nuevos brotes. Sin embargo, dichas infraestructuras y tecnologías compatibles necesitarán una inversión a largo plazo y una asociación entre entidades públicas y privadas para permitir la interacción de cientos de dispositivos inteligentes entre sí, estos costos de infraestructura y mantenimiento no están al alcance de algunos países que se encuentran en vías de desarrollo [2].

Con el fin de cumplir esta medida de distanciamiento social y que pueda ser implementada a un costo bajo en Ecuador, en este trabajo se pretende establecer una metodología de localización usando un prototipo telemático sin incurrir en costos de infraestructura.

En primera instancia se realizará un estudio comparativo de las diferentes tecnologías inalámbricas en el contexto de la detección de personas cercanas para facilitar el distanciamiento social. El estudio evaluará la factibilidad de uso de tecnologías con componentes disponibles en el mercado como UWB, Bluetooth LE, WiFi, RFID y NFC para la construcción de dispositivos y el costo de los “wearables” que permitan el rastreo y detección de personas en aplicaciones de distanciamiento social [6].

Finalmente usando simulaciones se evaluarán factores como precisión y latencia, estas son una de las características más importantes de un sistema de posicionamiento o localización. Los ambientes interiores debido a la presencia de obstáculos y efectos de múltiples trayectorias muchas veces requieren un procesamiento de señal y eliminación de ruido con el fin de que los sistemas puedan estimar las distancias con precisión. Además, la localización en tiempo real requiere que el sistema pueda informar la ubicación y las coordenadas del usuario sin ningún retraso notable, esto significa que el sistema debería poder ubicar al usuario con un pequeño número de señales de referencia y debería realizar operaciones complejas con una granularidad de milisegundos [4].

Con este estudio de factibilidad se desea contribuir al mercado nacional, en el desarrollo de pequeños dispositivos electrónicos que faciliten el control del distanciamiento social, a cada usuario que posea un “wearable”, ayudando de esta forma a contener la propagación de este virus altamente contagioso en lugares cerrados y edificaciones [7].

### **1.3 Objetivos**

#### **1.3.1. Objetivo general**

- Desarrollar una metodología de distanciamiento social mediante la evaluación de diferentes tecnologías de posicionamiento en interiores con la finalidad de construir dispositivos “wearables” de bajo costo.

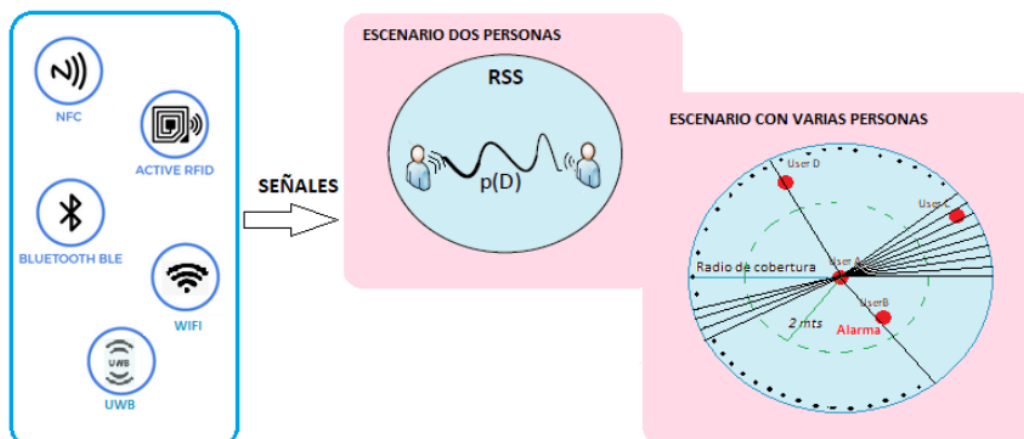
#### **1.3.2. Objetivos específicos**

- Analizar la arquitectura de las diferentes tecnologías inalámbricas de posicionamiento en interiores para realizar simulaciones usando código abierto.
- Establecer el modelo de RSSI y ToF para calcular la distancia entre dos objetos usando el simulador OMNeT++.
- Realizar un estudio de factibilidad de los componentes disponibles en el mercado para la construcción de dispositivos “wearables” de bajo costo.
- Evaluar el uso de las diferentes tecnologías para distanciamiento social en base a los resultados de las simulaciones y del estudio de factibilidad.

### **1.4 Metodología**

La metodología que se va a desarrollar consiste en utilizar las diferentes tecnologías inalámbricas tales como: WiFi, Bluetooth LE, RFID, UWB y NFC para calcular la distancia entre dos personas, si la distancia es menor a 2 metros, los dispositivos deben notificar a los usuarios y alentarlos a practicar el distanciamiento social. La distancia entre dos personas va a ser estimada usando los niveles de RSSI o ToF, estos valores se procesarán mediante un filtrado o promedio para lograr una mejor precisión, este procedimiento se realizará mediante un barrido circular delimitando el radio de cobertura para el caso de existir más personas que posean el “wearable” en el lugar [8].





**Figura 1.1. Escenarios para determinar la distancia entre dos personas**

Los datos estimados de distancia se obtienen realizando simulaciones usando las diferentes tecnologías inalámbricas en interiores disponibles como base para determinar los requerimientos del prototipo “wearable” que se desea implementar. Al finalizar en el estudio se mostrará que tecnología es factible considerando costos, precisión de localización y elementos disponibles en el mercado de Ecuador, para la construcción del prototipo que permita aplicar correctamente las medidas de distanciamiento social aplicadas para evitar la propagación del COVID-19.

### 1.5 Resultados esperados

Al finalizar este trabajo, basándose en los objetivos planteados se espera tener una visión más profunda de las arquitecturas de diferentes tecnologías inalámbricas en interiores y proponer un prototipo “wearable” de alta precisión y bajo costo que pueda ser aplicado en ambientes laborales en Ecuador para alarmar a los trabajadores que no cumplan el distanciamiento social de 2 metros y además monitorear el estado de salud de los mismos y proceder a notificar a sus jefes inmediatos.

## CAPÍTULO 2

# 2 ESTADO DEL ARTE DE TECNOLOGÍAS DE POSICIONAMIENTO EN INTERIORES.

### 2.1 Introducción

La pandemia del coronavirus ha acarreado graves consecuencias en el área de la salud y en la economía nacional. En Ecuador alrededor de un millón de personas han perdido sus empleos en tan solo 4 meses (de marzo a junio de 2020). En tal ambiente urge una necesidad de brindar soluciones para frenar la propagación de la enfermedad.

Una de estas medidas es la práctica del distanciamiento social que ayuda a reducir la propagación de este virus. Las tecnologías surgen como un papel clave para estas medidas, los sistemas de posicionamiento en interior contribuyen a que se respete el distanciamiento social entre las personas.

En este documento, aportamos un estudio completo sobre las tecnologías emergentes y cuáles son los impactos que favorecen al distanciamiento social en la práctica.

Las tecnologías inalámbricas como WiFi, Bluetooth LE, RFID, UWB y NFC nos permitirán desarrollar modelos básicos con lo que podremos discutir sobre sus aplicaciones, ventajas y practicas viables en su implementación para el distanciamiento social.

### 2.2 Tecnologías inalámbricas para distanciamiento social

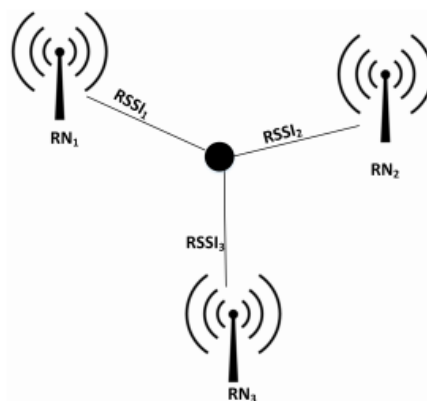
#### 2.2.1 WiFi

Una de las tecnologías que puede brindar una solución muy prometedora al distanciamiento social en ambientes interiores es la tecnología WiFi, ampliamente implementada justamente donde otras tecnologías como GPS o celular no puedan tener disponibilidad o poca precisión en el rastreo de las personas [8].

Una de las aplicaciones de la tecnología WiFi es el posicionamiento, el cual permite al usuario detectar multitudes dentro de un lugar y así forzar un distanciamiento especialmente en estos eventos donde existen brotes de enfermedades [8].

En primer plano la instalación de los dispositivos en zonas primordiales para determinar el posicionamiento de los usuarios, se lo puede realizar a muy bajo costo. En segundo plano, los avances en la tecnología WiFi para posicionamiento en interiores nos proporcionan una solución en servicios de localización haciendo posible que los usuarios cumplan con el distanciamiento social [4].

La distancia social entre personas va a ser determinada continuamente usando los niveles de RSSI en tiempo real, para decidir el modelo a implementar entre los puntos de acceso y los dispositivos de los usuarios adoptando el sistema de navegación para potenciar la precisión de los niveles RSSI, y así poder tomar las acciones necesarias para que las personas dentro de establecimiento realicen el distanciamiento social o informar a otros usuarios no ingresar al lugar [8].



**Figura 2.1. Localización con WiFi basada en RSSI [4]**

Otro método es la implementación de las huellas dactilares (fingerprints), con una base de datos ya construida previamente se puede tener accesos a estos usuarios donde las señales inalámbricas son muy afectadas por los obstáculos. Existe una desventaja en el uso de esta recopilación de

datos, es que requiere mucho tiempo y es muy laborioso especialmente en lugares donde existe mucha dinámica de los usuarios [4].

Otro procedimiento para localizar y rastrear a los usuarios en donde se puedan detectar multitudes potenciales es WiVi que significa (WiFi Visión), aquí se puede notificar al usuario que al lugar donde acude existe demasiada gente. Una de sus ventajas es la anulación de interferencias por el uso de MIMO y solo se centra en el usuario en movimiento [8].

La tecnología basada en WiFi es usada en ambientes interiores por lo que se requiere tener varios puntos de acceso, y con los mejoramientos de localización, a los usuarios se los pueden alentar a cumplir el distanciamiento social [8].

### **2.2.2 Bluetooth LE (BLE)**

Bluetooth 4.0, también conocido como Bluetooth Smart o Bluetooth Low Energy, es un protocolo de comunicación por radio especialmente diseñado para Internet de las cosas. BLE es una implementación totalmente diferente de las versiones anteriores de Bluetooth (1.0, 2.0 y 3.0), que estaban dedicadas a la transmisión de audio o datos, y como una sustitución inalámbrica de las comunicaciones RS-232. El objetivo principal de BLE es lograr la detección de dispositivos y la conectividad con un bajo consumo de batería, incluso a costa de velocidades de comunicación más bajas [9].

BLE usa ondas de radio UHF en la banda ISM de 2.4 a 2.485 GHz, y asigna en esa banda 40 canales, cada uno de 2 MHz de ancho. De estos 40 canales, 3 canales se utilizan para mensajes publicitarios y los 37 canales restantes se utilizan para la transmisión de datos utilizando un esquema de salto de frecuencia. La tasa de bits máxima es de 1 Mbps y la potencia de transmisión máxima es de 10 mW. A la potencia de transmisión máxima, el rango de transmisión BLE puede ser superior a 100 metros. Durante la fase de conexión, los dispositivos intentan conectarse a intervalos constantes, lo que se denomina intervalo de conexión. Este valor se puede establecer en cualquier lugar entre 7,5

milisegundos y 4 segundos. Un esquema de salto de canal cambia a un nuevo canal en cada intervalo de conexión [9].

El protocolo BLE utiliza mensajes de muy corta duración para publicidad. Esta baliza para descubrimiento también puede incluir alguna carga útil con otra información relevante, como la lectura del sensor (temperatura, presión, humedad, etc.).

BLE tiene varias ventajas sobre las balizas basadas en WiFi. Cuando se usa un teléfono para registrar las señales WiFi mediante escaneo pasivo, la duración mientras el teléfono está esperando en el Identificador de conjunto de servicios (SSID) puede ser tan grande como 4 segundos, por lo que la velocidad de escaneo podría ser inferior a 0,25 Hz. BLE permite tasas de publicidad de hasta 10 Hz y la intensidad de la señal recibida (RSSI) se obtiene en tiempo real sin almacenamiento en búfer. Además, las etiquetas BLE duran varios meses con baterías pequeñas, y la lectura de mensajes de transmisión del teléfono consume menos energía que en un escaneo WiFi. Por último, las etiquetas BLE se pueden implementar fácilmente ya que son pequeñas y funcionan con baterías [9].

El rendimiento de rango de punto a punto con BLE es bajo y similar al que se puede lograr con WiFi o balizas RFID, es decir, alrededor de 1 a 2 metros en rangos cortos (menos de 5 metros), pero el rango no se puede estimar con precisión para distancias más grandes o cuando personas u objetos bloqueadores atenúen la señal de radio. Sin embargo, cuando se utilizan varias etiquetas BLE como balizas en cooperación, el rendimiento de localización puede estar entre 2 y 5 metros en áreas más grandes. Encontrar un objeto utilizando el rango BLE de punto a punto es un desafío y otras fuentes de información, como el movimiento del usuario, podrían ser beneficiosas [10].

En los sistemas BLE, hay cinco estados de operación a nivel de la capa de enlace, los mismos que se muestran en la Tabla 2.1.

**Tabla 2.1 Estados de operaciones de BLE**

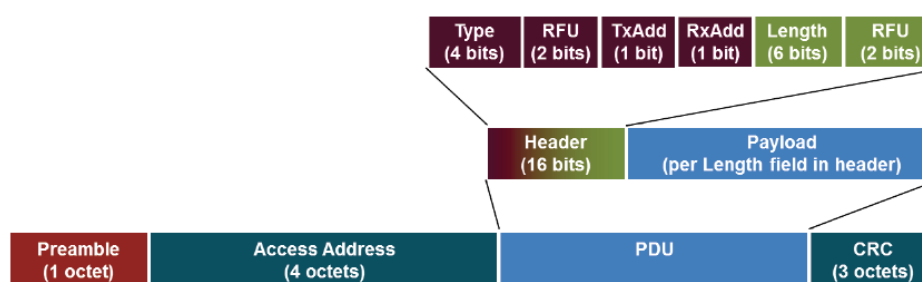
<b>Estado</b>	<b>Descripción</b>
Espera	No transmite, ni recibe paquetes
Anuncio	Emite anuncios mediante los canales publicitarios.
Escaneo	Busca anunciantes.
Inicio	Inicia la conexión con el anunciante
Conexión	<p><b>Rol Maestro:</b> se comunica con el dispositivo que tiene el rol de esclavo y define tiempos de transmisión.</p> <p><b>Rol esclavo:</b> se comunica con un solo dispositivo que esté en el rol de maestro.</p>

La capa BLE GAP define cuatro roles: emisor, observador, periférico y central. El emisor envía periódicamente mensajes de anuncio con datos, está optimizado para aplicaciones que distribuyan datos regularmente. El observador está optimizado para aplicaciones que solamente quieren recopilar datos de los dispositivos emisores, escucha los datos incluidos en los mensajes de publicidad. El periférico es un dispositivo de bajo consumo que expone información y puede realizar conexiones, utiliza paquetes publicitarios conectables para transmitir información que cualquier otro dispositivo BLE dentro del alcance puede escuchar. Un dispositivo central suele ser un dispositivo alimentado, que incluye una batería recargable y con una mayor potencia de procesamiento con respecto a los periféricos (por ejemplo, un teléfono inteligente o una tableta). Los dispositivos centrales implementan una modalidad de escáner, en la que escuchan los anuncios e inician la solicitud de conexión [11].

Cuando un periférico quiere transmitir, inicia un evento publicitario, donde el mismo paquete se transmite secuencialmente en cada uno de los tres

canales publicitarios. Los dispositivos que funcionan como escáneres detectarán uno de estos y pasarán la información que contiene a la aplicación y la pila de protocolos de nivel superior. Aunque el objetivo principal de los paquetes publicitarios dentro de la especificación es permitir el descubrimiento de dispositivos y establecer una conexión segura, también permiten que se transmitan pequeñas cantidades de datos para que otros dispositivos los escuchen [11].

El formato de los paquetes publicitarios se muestra en la Figura 2.1. El paquete completo se define como Preámbulo, Dirección de acceso, Unidad de paquete de datos (PDU) y Campo de verificación de redundancia cíclica (CRC). El preámbulo (0xaa) se utiliza para la sincronización de frecuencia y el entrenamiento de control automático de ganancia (AGC). La dirección de acceso (0x8e89bedd6) está diseñada para la detección de paquetes. CRC se calcula sobre la PDU para la verificación de errores. La PDU se compone de carga útil y encabezado. En el encabezado, se definen el tipo de paquete, el tipo de dirección TX/RX, la longitud de la carga útil y el campo reservado para uso futuro [11].



**Figura 2.2. Formato del paquete publicitario [11]**

Debido a la naturaleza de la banda ISM, BLE debe superar la interferencia de otros sistemas haciendo uso de la técnica de espectro ensanchado por salto de frecuencia (FHSS). Esto distribuye la potencia de RF a través del espectro, lo que reduce la interferencia y la densidad de potencia espectral. FHSS ocurre mientras está en una conexión. La frecuencia

salta entre 37 canales de datos según los algoritmos de selección de canales. La capa de enlace del maestro clasificará los canales de datos en canales usados y canales no usados que se denominan mapa de canales. El esclavo recibirá el mapa de canales del maestro en la solicitud de conexión. El maestro puede actualizar el mapa de canales mediante un mensaje de actualización de canales.

En este trabajo se utilizará BLE para determinar la distancia entre dos personas usando dispositivos wearables, la estimación de distancia será basada en niveles RSSI. Cabe señalar que la tecnología Bluetooth puede permitir que un dispositivo se conecte a varios dispositivos al mismo tiempo. Por lo tanto, el dispositivo puede detectar simultáneamente distancias a múltiples dispositivos en su cobertura. Si la distancia es inferior a un umbral determinado, por ejemplo, 2 metros los dispositivos pueden advertir y/o animar a los usuarios a practicar el distanciamiento social [12].



**Figura 2.3. Distanciamiento entre dos personas que usan BLE [12]**

### 2.2.3 Identificación por radio frecuencia (RFID)

En los últimos años se han desarrollado muchas tecnologías, la identificación por radiofrecuencia es una de ellas, RFID utiliza campos electromagnéticos para identificar y rastrear automáticamente las etiquetas adheridas a los objetos. Las etiquetas contienen datos acumulados electrónicamente.

Las etiquetas RFID se utilizan ampliamente en la identificación de bienes o personas, a una distancia mínima del lector. Las etiquetas también se



pueden colocar en vehículos, que se pueden escanear a distancia. La etiqueta se puede escanear si se pasa cerca de un lector, incluso si está cubierta por el objeto o no es visible. Las limitaciones de RFID son la dependencia de un sistema de RFID que incluye hardware primario como etiquetas, lector y software que dificulta el acceso al sistema. La precisión del sistema depende del tipo de etiquetas, el lector y la proximidad o la cercanía de la etiqueta y el lector [12].

RFID juega un papel clave en la localización y seguimiento de objetos en tiempo real. Un sistema de localización RFID generalmente consta de tres componentes principales: lectores RFID, etiquetas RFID y un sistema de procesamiento de datos. El lector RFID lee los datos emitidos por las etiquetas. Los lectores y las etiquetas RFID utilizan frecuencias definidas y un protocolo para la transmisión y recepción de datos [13].

Normalmente, las etiquetas RFID se pueden clasificar en dos tipos: etiquetas activas y etiquetas pasivas. Una etiqueta RFID pasiva puede funcionar sin necesidad de ninguna fuente de alimentación y está alimentada por el campo electromagnético generado por el lector RFID. La señal que llega desde los lectores es suficiente para que el circuito integrado del tag funcione, sin embargo, el radio de cobertura es limitado se encuentra entre 1 a 2 metros. Estos sistemas pasivos pueden funcionar en 4 bandas de frecuencia: LF (125 kHz), HF (13,56 MHz), UHF (433 o 868-915 MHz) y microondas (2,45 GHz o 5,8 GHz) [13].

Por el contrario, una etiqueta RFID activa tiene su propia fuente de energía, por ejemplo, una batería, y transmite continuamente sus propias señales. Son más fiables, el rango de cobertura va desde decenas hasta cientos de metros y los rangos de frecuencia son similares a las etiquetas pasivas. Las etiquetas RFID activas se utilizan normalmente en sistemas de localización. Así, la tecnología RFID puede considerarse una tecnología potencial para practicar el distanciamiento social [12].

### 2.2.4 Banda ultra ancha (UWB)

Como una tecnología prometedora, UWB tiene un sistema de posicionamiento en interiores más preciso en niveles de corto a mediano rango. Por el oportuno ancho de banda de las señales UWB (500MHz), este tiene una gran ventaja de generar pulsos de muy poca duración, lo que permite a la señal propagarse sin necesidad de una portadora adicional [6].

Una de las mayores ventajas que tiene esta tecnología es la baja potencia que tienen los transmisores UWB, por lo tanto, se obtiene mayor precisión y seguimiento en tiempo real en los dispositivos de posicionamiento en interior mediante el uso de técnicas de triangulación. Los objetivos de seguimiento necesitan comunicarse con los puntos de accesos, por lo que el dispositivo puede ser rastreado en tiempo real definiendo la diferencia entre la respuesta al impulso del nuevo canal, con el dispositivo anterior [8].

La detección de los dispositivos en aéreas donde existe mayor recurrencia de personas se vuelve un poco limitado el trabajo del rastreo, porque dichos dispositivos tendrían que conectarse a los puntos de accesos sin revelar información personal, de este modo las autoridades pueden rastrear con exactitud y tomar las acciones adecuadas para tener el distanciamiento social [12].

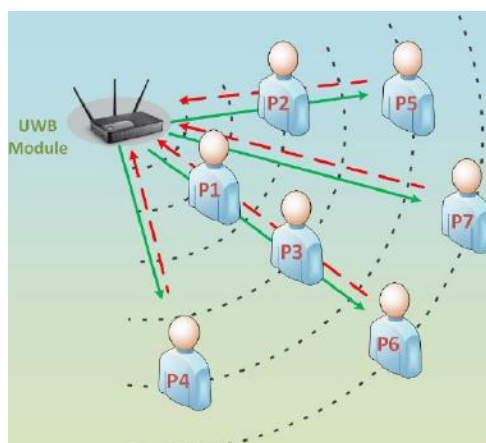


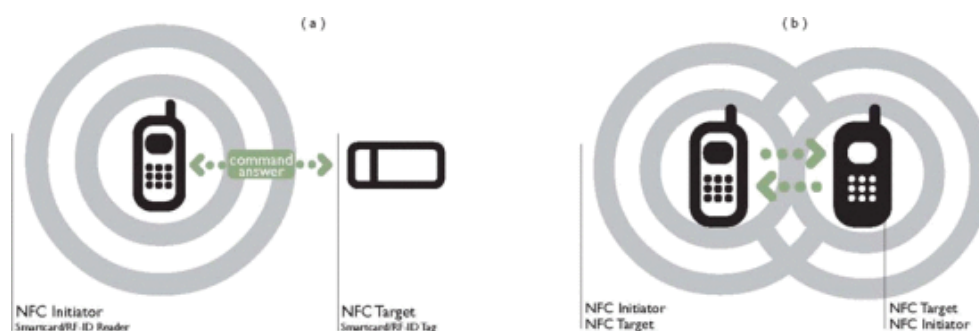
Figura 2.4. Sistema de conteo usando UWB [12]

La tecnología UWB se puede considerar como una destacada solución para el manejo del distanciamiento social, tanto en ambientes de interior como de exterior. Estos sistemas se pueden manejar para detectar y rastrear los dispositivos con un bajo costo en su implementación, otra aplicación que tiene esta tecnología también es que se puede monitorear la posición de la gente que está aislada y ver si incumplen su cuarentena [8].

### 2.2.5 Comunicación de campo cercano (NFC)

NFC utiliza una banda de alta frecuencia de hasta 13,56 MHz, con una velocidad de transmisión de datos de 424 kbps y un alcance de 10 centímetros. Los sistemas NFC constan de dos elementos: el iniciador; como su nombre lo indica inicia y controla el intercambio de información (llamado lector en RFID); y el target es el dispositivo que responde al requerimiento del iniciador (llamado etiqueta en RFID) [14].

En un sistema NFC hay dos modos de comunicación: activo y pasivo. En el activo, ambos dispositivos generan su propio campo de radiofrecuencia para transmitir datos (peer to peer). En el pasivo, solo uno de estos dispositivos genera el campo de radiofrecuencia, mientras que el otro se utiliza para cargar transferencias de datos [14].



**Figura 2.5. Modos de comunicación: (a) Pasivo (b) Activo [14]**

Los modos de funcionamiento de NFC se definen en función del emparejamiento de los dispositivos y del modo de comunicación. En el modo lector/escritor el teléfono móvil habilitado para NFC activo es capaz de leer o escribir una etiqueta pasiva, se utiliza principalmente para

prestación de servicios, por ejemplo: publicidad interactiva, acceso a información y entrega de contenidos. En el modo peer to peer permite la comunicación entre dos dispositivos NFC activos, la interfaz de comunicación RF está estandarizada por ISO/IEC 18092 lo que permite el modelo de solicitud-respuesta. En el modo de emulación de tarjeta el dispositivo NFC se comunica con lectores RFID conocidos, es posible utilizar infraestructura existente, por ejemplo: para pagos o control de acceso [15].

### **2.3 Técnicas de ranging en ambientes interiores**

Las técnicas de ranging se pueden clasificar en enfoques basados en indicadores de intensidad de señal recibida (RSSI), basados en tiempo y basados en fases. En las soluciones basadas en RSSI, el receptor de la señal calculará su distancia al transmisor basándose en la atenuación de la señal transmitida a una distancia. Las soluciones basadas en RSSI sufren de desvanecimiento de múltiples rutas y muchos investigadores utilizan la huella dactilar a menudo en combinación con el filtro Kalman para mitigar este problema. En la solución basada en el tiempo, la distancia entre dos dispositivos se calcula en función del retardo de propagación. Las soluciones basadas en el tiempo se pueden clasificar en hora de llegada (ToA), diferencia horaria de llegada (TDoA) y tiempo de vuelo (ToF). Sin embargo, las soluciones basadas en el tiempo requieren un gran ancho de banda de señal para tener una precisión aceptable en entornos de múltiples rutas, lo cual es incompatible con BLE.

#### **2.3.1 Indicador de fuerza de la señal recibida (RSSI)**

Para la técnica basada en RSSI la distancia o ranging se mide en base a la atenuación introducida por la propagación de la señal desde el nodo transmisor al nodo receptor. Esta técnica se usa comúnmente en sistemas de bajo costo como las redes de sensores inalámbricos (WSN) o WiFi debido a la fácil disponibilidad de este tipo de medición y no se requiere sincronización entre nodos.

El mapeo entre el RSSI medido y la distancia entre los nodos de transmisión y recepción se realiza típicamente utilizando modelos de pérdida de trayectoria teóricos y/o empíricos. El inconveniente más relevante de RSSI es que, en entornos complicados, los fenómenos de propagación hacen que la atenuación de la señal se correlacione deficientemente con la distancia, particularmente en condiciones de canal sin línea de vista (NLOS), lo que resulta en estimaciones de distancia inexactas [16].

Por tanto, basándose en el nivel RSSI de las señales recibidas, el punto de acceso puede estimar la posición del dispositivo del usuario en entornos interiores.

### 2.3.2 Diferencia de tiempo de arribo (TDoA)

Este método usa dos tipos de señales de transmisión con diferentes velocidades y se calcula la distancia  $D$  basada en la diferencia entre ellas, como se muestra en la Ecuación 1 [12].

$$\frac{D}{c_1} - \frac{D}{c_2} = t_1 - t_2 \quad \text{Ecuación 1}$$

$c_1$ ,  $c_2$ ,  $t_1$   $t_2$  son las velocidades y el tiempo de llegada de las dos señales respectivamente.

Los sistemas que utilizan TDoA no se basan en estimaciones de distancia absoluta entre pares de nodos. Estos sistemas suelen emplear uno de dos esquemas. De acuerdo con el primer esquema, se transmiten múltiples señales desde nodos ancla sincronizados y el nodo móvil mide la TDoA (esta técnica es similar a la adoptada por las tecnologías GNSS). Según el segundo esquema, el nodo móvil emite una señal de referencia y es recibida por varios anclajes. Los nodos anclas comparten su ToA estimado y calculan el TDoA. Cada esquema requiere que los nodos anclas estén estrechamente sincronizados a través de una red [16].

### 2.3.3 Ángulo de arribo (AoA)

Las técnicas basadas en ángulos estiman la posición de un nodo móvil midiendo el ángulo de llegada (AoA) de las señales que llegan al nodo de medición mediante un conjunto de antenas. Con medidas perfectas, el problema de posicionamiento se puede resolver geoméricamente encontrando la intersección de varias líneas rectas que representan la señal AoA [16].

Por lo general, la geometría se puede lograr con antenas direccionales o con una serie de antenas. El principio fundamental detrás de la medición AoA a través de conjuntos de antenas consiste en que las diferencias en los tiempos de llegada de una señal entrante en diferentes elementos de antena incluyen la información del ángulo dado que se conoce la geometría del conjunto, además AoA no requiere sincronización entre nodos [17].

Esta técnica tiene sus limitaciones, es muy sensible a rutas múltiples y NLOS a veces no es adecuado para localización en interiores, sin embargo, se puede combinar con otras técnicas para mejorar su precisión. Además, estos algoritmos tienen una mayor complejidad en comparación a otros métodos y requieren antenas adicionales con capacidad para medir los ángulos, lo cual aumenta el costo de todo el sistema [17].

### 2.3.4 Tiempo de arribo (ToA)

Para calcular la distancia entre un nodo transmisor y receptor se puede usar el retardo de propagación y velocidad de la señal.

$$D = \tau \times c \quad \text{Ecuación 2}$$

Donde  $c$  es la velocidad a la que viaja la señal ( $3 \times 10^8 m/s$ ),  $\tau$  es el tiempo de propagación desde el nodo transmisor al receptor y  $D$  es la distancia entre el nodo transmisor y receptor [17].

Se puede utilizar ToA junto con la tecnología UWB para lograr una mayor precisión, esta técnica puede manejar mejor la resolución de tiempo con

la ayuda de la tecnología UWB. Si bien la técnica ToA tiene un requisito restringido para la sincronización, esta ineficiencia puede ser compensada por UWB debido a su naturaleza de sensibilidad al tiempo. La tecnología UWB utiliza una duración de pulso corta para filtrar las señales causadas por la reflexión para mejorar el rendimiento general. Como consecuencia, la combinación de estas dos tecnologías es predominante en el campo de la localización en interiores en la actualidad [17].

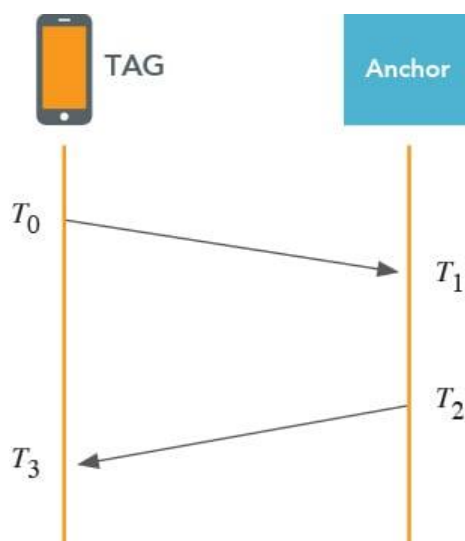
### 2.3.5 Tiempo de vuelo (ToF)

Esta técnica surge con el objetivo de resolver el problema de sincronización. ToF para determinar la distancia entre pares mide el tiempo de propagación que tarda la señal de RF en viajar entre el transmisor y el receptor. Esta medida de tiempo multiplicada por la velocidad de la luz determina la distancia con gran precisión. En comparación con otras tecnologías que se basan en la intensidad de la señal medida para evaluar la distancia, UWB proporciona un mayor grado de precisión y seguridad.

El proceso de medición de ToF entre pares de UWB se denomina rango de dos vías (TWR), porque los pares intercambian una serie de mensajes de solicitud y respuesta con marcas de tiempo integradas. El tiempo de ida y vuelta (RTT) se usa para determinar el ToF como se describe en las Ecuación 3.

$$D = \frac{(RTT - Reply) \times c}{2} \quad \text{Ecuación 3}$$

Donde  $RTT$  es el tiempo de todo el viaje de la señal ida y vuelta ( $T_3 - T_0$ ),  $Reply$  es el retardo de tiempo predeterminado entre el momento en que el nodo objetivo recibe la señal y la envía de vuelta ( $T_2 - T_1$ ), y  $c$  es la velocidad de la señal de transmisión [17].



**Figura 2.6. Envío de mensajes entre dos dispositivos UWB**

Una desventaja común de los métodos ToA y TDoA es que requieren relojes sincronizados en el nodo y el objetivo. Eso puede ser costoso de implementar, ya que requiere calibraciones frecuentes para mantener la precisión [17].

#### **2.4 Modelo para calcular el RSSI en ambientes interiores**

Hoy en día, la tecnología de posicionamiento al aire libre se ha utilizado ampliamente. Pero el canal de propagación de radio en interiores es muy diferente al del exterior, se ve fuertemente afectado por los materiales de construcción, la estructura interna, la distribución de la habitación y otros factores, que causan la compleja estructura de múltiples trayectos.

Cuando la onda electromagnética se encuentra con un objeto, como pisos y paredes, cuyo tamaño es mayor que su longitud de onda, se producirá la reflexión. La difracción ocurre cuando la ruta de radio entre el transmisor y el receptor está bloqueada por los bordes afilados. La dispersión ocurre cuando las ondas encuentran una superficie rugosa cuyas dimensiones son del mismo orden que la longitud de onda. La dispersión generalmente se ignora en ambientes interiores y la reflexión debe tenerse en cuenta con la propagación de la línea de vista.



Se requieren modelos de propagación para utilizar la técnica RSSI, actualmente, el modelo shadowing se utiliza ampliamente para modelar la pérdida de propagación de la señal inalámbrica y se expresa como se muestra en la Ecuación 4 [18].

$$RSSI(d) = P_r(d_o) - 10 \times n \times \log_{10}\left(\frac{d}{d_o}\right) + X_\sigma \quad \text{Ecuación 4}$$

Donde  $d$  y  $d_o$  representan la distancia real y la distancia de referencia respectivamente;  $P_r(d)$  denota la potencia de la señal recibida en  $dBm$  a la distancia real;  $P_r(d_o)$  es la potencia de la señal recibida en  $dBm$  a la distancia de referencia;  $n$  es el exponente path-loss y  $X_\sigma$  es una variable aleatoria que representa el ruido en la medida de  $P_r(d)$ . Para aplicaciones en ambientes interiores  $d_o = 1 \text{ metro}$  y  $P_r(d_o)$  es calculado mediante la fórmula de pérdida de trayectoria en espacio libre. El ruido  $X_\sigma$  se modela como una variable aleatoria con distribución Gaussiana con media cero y varianza  $\sigma^2$ . El exponente path-loss  $n$  está determinado por las variables ambientales y la estructura del ambiente interior.

El exponente path-loss varía dentro de un cierto rango dependiendo de las características del medio ambiente como se muestra en la Tabla 2.2.

**Tabla 2.2 Exponentes path-loss para diferentes ambientes [18]**

Ambiente	Exponente path-loss
Oficina	1.4 ~ 2.5
Corredor	1.9 ~ 2.5
Escalera y balcones	1.4 ~ 2.4
Parque	2.7 ~ 3.4
Cercas	4.6 ~ 5.1
Callejones	2.1 ~ 3.0

Dado que el error del cálculo de distancia es causado principalmente por la influencia del medio ambiente, se utiliza el modelo shadowing simplificado como se muestra en la Ecuación 5 [18].

$$RSSI(d) = A - 10 \times n \times \log_{10} \left( \frac{d}{d_0} \right) \quad \text{Ecuación 5}$$

Donde  $RSSI(d)$  es la potencia de la señal recibida a una distancia  $d$  y  $A$  es la potencia de la señal recibida a una distancia de 1 metro.

Basada en la Ecuación 5 se puede determinar la distancia  $d$  entre dos nodos.

$$d = 10^{\left( \frac{|A - RSSI|}{10n} \right)} \quad \text{Ecuación 6}$$

## 2.5 Procesamiento de la señal RSSI

Dada la Ecuación 6 para el cálculo de la distancia entre dos nodos se puede notar que este valor está estrechamente relacionado con los valores de RSSI tomados durante la adquisición de la señal inalámbrica. La propagación de la señal en un ambiente interior es muy compleja, con diversos factores de interferencia, ocasionando que el RSSI recibido sea fluctuante y poco preciso, por tal motivo si se desea obtener una distancia precisa se debe realizar un procesamiento de valores de RSSI medidos utilizando métodos de filtrado tales como: filtrado de Kalman, filtro Gaussiano, filtro promedio, etc. [19].

### 2.5.1 Filtrado Gaussiano

Algunos datos de RSSI recibidos pueden tener valores de probabilidades pequeñas, y otros pueden encontrarse dentro de un rango razonable de fluctuaciones, siendo similares a los valores reales. Los valores de RSSI de alta frecuencia son seleccionados y los valores de RSSI de error se eliminan a través de un ajuste de Gauss. La función de ajuste de Gauss se muestra en la Ecuación 7 [20].

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{Ecuación 7}$$

Dado:

$$\mu = \frac{1}{N} \sum_{i=1}^N RSSI_i \quad \text{Ecuación 8}$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (RSSI_i - \mu)^2} \quad \text{Ecuación 9}$$

Debido a las características no lineales de la función de ajuste, el tiempo de procesamiento de una gran cantidad de datos es largo. Para seleccionar datos rápidamente, se adopta el método de ajuste cercano a Gauss, que es usar el valor medio de los datos de origen como punto de referencia y establecer la desviación estándar para la selección de datos, la fórmula se muestra en la Ecuación 10 [20].

$$|RSSI_i - \mu| < k\sigma \quad \text{Ecuación 10}$$

Donde  $\mu$  es la media del RSSI,  $\sigma$  es la desviación estándar y  $k$  es el número de valores de RSSI tomados durante un intervalo, este valor puede ser determinado según el porcentaje de retención de datos  $P$  [20].

$$P = \frac{k}{N} \times 100\% \quad \text{Ecuación 11}$$

Donde  $N$  es el tiempo de las lecturas, si  $P$  es demasiado pequeño se puede cometer errores, en este caso es necesario aumentar  $k$  para eliminar los datos incorrectos rápida y eficazmente [20].

### 2.5.2 Filtrado de Kalman

El filtrado de Kalman se puede utilizar para datos dinámicos logrando así que los valores de RSSI sean uniformes. Este filtrado se puede dividir en dos etapas predicción y actualización.

La etapa de predicción está dada por la Ecuación 12 y 13 [19].

$$x(k|k-1) = x(k-1|k-1) \quad \text{Ecuación 12}$$

$$P(k|k-1) = P(k-1|k-1) + Q \quad \text{Ecuación 13}$$

La etapa de actualización está dada por la Ecuación 14, 15 y 16 [19].

$$Kg(k) = \frac{P(k|k-1)}{P(k|k-1) + R} \quad \text{Ecuación 14}$$

$$x_{(k|k)} = x(k|k-1) + Kg(k)[y(k) - x(k|k-1)] \quad \text{Ecuación 15}$$


$$P_{(k|k)} = (1 - Kg(k))P(k|k-1) \quad \text{Ecuación 16}$$

Donde  $x(k-1|k-1)$  es el valor de RSSI del momento  $k-1$ ;  $x_{(k|k)}$  es el valor de RSSI del momento  $k$ ;  $x(k|k-1)$  es el valor de predicción de RSSI del momento  $k$ ;  $P(k-1|k-1)$  corresponde al error de covarianza de  $x(k-1|k-1)$ ;  $P(k|k-1)$  corresponde al error de covarianza de  $x(k|k-1)$ ;  $Q$  es la covarianza de ruido del estado;  $R$  es el ruido de covarianza medido;  $Kg(k)$  es la ganancia de Kalman del momento  $k$ ; y  $y(k)$  es el valor medido de RSSI en el momento  $k$  [19].

## 2.6 Estudio de mercado de dispositivos de rastreo o distanciamiento social en el ámbito internacional y nacional

Diferentes países han adoptado varios enfoques tecnológicos para lograr el rastreo de contactos y aplicar el distanciamiento social en ambientes laborales. La distancia entre dos personas, así como la duración de su proximidad, influye significativamente en el riesgo de infección. A continuación, se mencionan tecnologías basadas en wearables o aplicaciones móviles usadas con el fin de detener la propagación del virus en el ámbito internacional y nacional:


**Tabla 2.3 Wearables disponibles en el mercado internacional.**

Producto	País	Características
Estimote [21] 	USA	<ul style="list-style-type: none"> <li>• Tecnología Bluetooth.</li> <li>• Privacidad de los usuarios.</li> <li>• Informes en tiempo real del estado de salud del usuario.</li> <li>• Base de datos local y segura.</li> </ul>

<p>TraceTogether [22]</p> 	<p>Singapur</p>	<ul style="list-style-type: none"> <li>• Tecnología Bluetooth.</li> <li>• Rastreo de contactos.</li> <li>• Captura datos de proximidad a través de Bluetooth y no del GPS.</li> <li>• Datos son cifrados.</li> <li>• Batería con duración de 9 meses sin recarga.</li> </ul>
<p>Tsingoal [23]</p> 	<p>China</p>	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Precisión de <math>\pm 0,1</math> metros.</li> <li>• Cifrado de datos.</li> <li>• Almacenamiento en la nube.</li> <li>• Batería recargable en promedio cada 24 horas.</li> <li>• Luz de alarma y vibración con distancias menores a 6 pies.</li> </ul>
<p>Safe Spacer [24]</p> 	<p>Italia</p>	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Alerta de vibración, luz y sonido.</li> <li>• Resistente al agua.</li> <li>• Privado y seguro.</li> </ul>
<p>Rombit [25]</p> 	<p>Bélgica</p>	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Alertas visuales.</li> <li>• Batería hasta 18 horas de duración.</li> <li>• Resistente al agua.</li> </ul>

<p>Proxii Contact [26]</p> 	<p>Canadá</p>	<ul style="list-style-type: none"> <li>• Tecnología Bluetooth 4.0</li> <li>• Alertas a los usuarios.</li> <li>• Protección contra el polvo y agua.</li> <li>• Datos cifrados.</li> <li>• Capacidad de actualización del firmware.</li> </ul>
<p>Samsung [27]</p> 	<p>Corea del Sur</p>	<ul style="list-style-type: none"> <li>• Sensores de actividad, movimiento y frecuencia cardíaca.</li> <li>• Alertas como vibraciones y mensajes codificados por colores.</li> <li>• Preconfigurado con una aplicación de Radiant RFID.</li> </ul>
<p>eqWave [28]</p> 	<p>USA</p>	<ul style="list-style-type: none"> <li>• Tecnología UWB y Bluetooth.</li> <li>• Cubre 360 grados entre personas en ambientes sin línea de vista.</li> <li>• Alerta visual y vibratoria.</li> <li>• Batería de larga duración.</li> <li>• Rastreo de contactos.</li> </ul>
<p>Tended [29]</p> 	<p>UK</p>	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Alta precisión y baja latencia.</li> <li>• Vibraciones como alertas.</li> <li>• Privacidad de datos del usuario.</li> <li>• Dashboard en un monitor.</li> <li>• Información del estado físico del usuario.</li> </ul>

<p>Proximity Trace [30]</p> 	USA	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Alarma visual y audible.</li> <li>• Rastreo de contactos.</li> <li>• Informes de seguimiento de contactos.</li> <li>• Batería recargable.</li> </ul>
<p>SafeSpace [31]</p> 	UK	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Rastreo de contactos.</li> <li>• Duración de batería más de 14 horas.</li> <li>• Interacciones de usuario rastreables con información cifrada.</li> <li>• Alertas de audio, visuales o vibratorias.</li> </ul>
<p>RightCrowd [32]</p> 	Australia	<ul style="list-style-type: none"> <li>• Tecnología BLE 4.2.</li> <li>• Alertas visuales en las tarjetas digitales.</li> <li>• Rastreo de contactos.</li> <li>• Batería de litio, 80 horas de uso continuo.</li> <li>• Aplicación móvil que emite advertencias de audio.</li> </ul>
<p>AlertTrace [33]</p> 	USA	<ul style="list-style-type: none"> <li>• Tecnología BLE.</li> <li>• Cifrado de extremo a extremo.</li> <li>• Historial de contactos de un usuario.</li> <li>• Batería de larga duración de hasta 6 meses.</li> <li>• No necesita infraestructura.</li> </ul>

<p>Losant [34]</p> 		<ul style="list-style-type: none"> <li>• Tecnología BLE.</li> <li>• Interacciones con otros tags.</li> <li>• Guarda 100 interacciones.</li> <li>• Reporte y datos históricos enviados a la nube.</li> <li>• No se necesita infraestructura de servicios.</li> <li>• No captura la información de posición.</li> </ul>
<p>Kinexon SafeTag [35]</p> 	Alemania	<ul style="list-style-type: none"> <li>• Tecnología UWB.</li> <li>• Permite gestionar diferentes zonas dentro de una empresa.</li> <li>• Advertencias audibles personalizables.</li> <li>• Se puede integrar con una infraestructura de TI existente.</li> <li>• Registra distancia de los usuarios y duración del contacto.</li> </ul>

**Tabla 2.4 Aplicaciones móviles disponibles en el mercado internacional.**

Aplicación móvil	País	Características
ASI Ecuador [36]	Ecuador	<ul style="list-style-type: none"> <li>• Sin consumo de datos móviles.</li> <li>• Acceso a contactos a través de Bluetooth.</li> <li>• No utiliza geolocalización.</li> <li>• No accede a ningún dato personal del teléfono.</li> <li>• Notificaciones son enviadas a los usuarios que han estado en contacto</li> </ul>



		con usuarios que tienen síntomas del virus.
Corona Warn App [37]	Alemania	<ul style="list-style-type: none"> <li>• Transmite un identificador y buscar identificadores de otros teléfonos que utilizan BLE.</li> <li>• Identificadores son válidos durante 10 a 20 minutos.</li> <li>• Claves cambian cada 24 horas.</li> <li>• Servidor agrega las claves del usuario a la lista de SARS-CoV-2</li> </ul>
Aarogya Setu [38]	India	<ul style="list-style-type: none"> <li>• Se registra con un número de teléfono.</li> <li>• Versión Android 5 o superior, o versión de IOS 10.3 o superior.</li> <li>• Rastreo de contactos.</li> <li>• Mapeo sindrómico y autoevaluación.</li> <li>• Utiliza las funciones de GPS y Bluetooth.</li> </ul>
StopCovid	Francia	<ul style="list-style-type: none"> <li>• Almacena identificadores anónimos denominados EphID.</li> <li>• Los usuarios deben verificar sus EphID para verificar si están en riesgo.</li> <li>• Utiliza la tecnología Bluetooth.</li> </ul>

## CAPÍTULO 3

### 3 DISEÑO DEL WEARABLE PARA DISTANCIAMIENTO SOCIAL

Como se ha expuesto anteriormente existen diferentes tecnologías inalámbricas de posicionamiento en interiores, que debido a limitaciones de precisión y alcance son aplicadas en un entorno definido y concreto. Dentro de las diferentes tecnologías estudiadas se ha definido la utilización de WiFi, UWB, BLE y RFID, para evaluar la precisión al momento de estimar distancias entre wearables, se ha descartado NFC ya que esta tecnología tiene un alcance de máximo 10cm.

Para la estimación de distancia o ranging se va a medir la intensidad de la señal recibida RSSI y el tiempo de vuelo ToF, estas son formas sencillas de medir distancia entre dos puntos y tiene un costo bajo de implementación, se toman varios datos de este parámetro para proceder a realizar un promedio y así lograr mejorar la precisión del wearable.

Los wearables permitirán reforzar el distanciamiento social de forma fiable indicando si el usuario se encuentra a cierta distancia de otro usuario. Los expertos en enfermedades infecciosas y las agencias de salud global están de acuerdo en que el distanciamiento social es una de las formas más efectivas de detener o frenar la propagación del COVID-19, sin embargo, en el mercado de Ecuador aún no se han desarrollado estos prototipos, al momento solo se dispone de una aplicación móvil llamada ASI Ecuador, donde se registran las personas que presentan síntomas o que han dado positivo en una prueba COVID.

Del estudio realizado sobre las diferentes aplicaciones móviles desarrolladas a nivel internacional y nacional se determinó que las principales preocupaciones están relacionadas con la gestión de datos del usuario, casos de falsos positivos y negativos, problemas de seguridad y privacidad. Además del consumo excesivo de batería de los dispositivos móviles, ya que estas aplicaciones intercambian continuamente mensajes con el servidor. Otro inconveniente es el tener que

desarrollar una aplicación que funcione en todos los modelos de teléfonos inteligentes considerando los diferentes sistemas operativos Android e iOS.

En este capítulo se abordará los requerimientos, arquitectura y algoritmos para el diseño e implementación de los wearables, se han desarrollado simulaciones usando el software OMNeT++ con el fin de evaluar la precisión de las tecnologías UWB, RFID, WiFi y BLE. Además, se han realizado pruebas en un ambiente indoor con módulos WiFi, BLE y UWB.

### **3.1 Requerimientos de la implementación de wearables para distanciamiento social**

Para el diseño e implementación de wearables aplicados en distanciamiento social es necesario definir los requerimientos de software y hardware que se usaron con el fin de evaluar la precisión de distancias usando diferentes tecnologías. Primero se utilizó el software OMNeT++ junto INET que es un simulador de redes inalámbricas y móviles, en este simulador se evaluaron las tecnologías: UWB, RFID, WiFi y BLE. Finalmente se realizaron pruebas con hardware ESP01, Arduino NANO 33 BLE y módulos UWB que corresponden a tecnologías WiFi, BLE y UWB respectivamente.

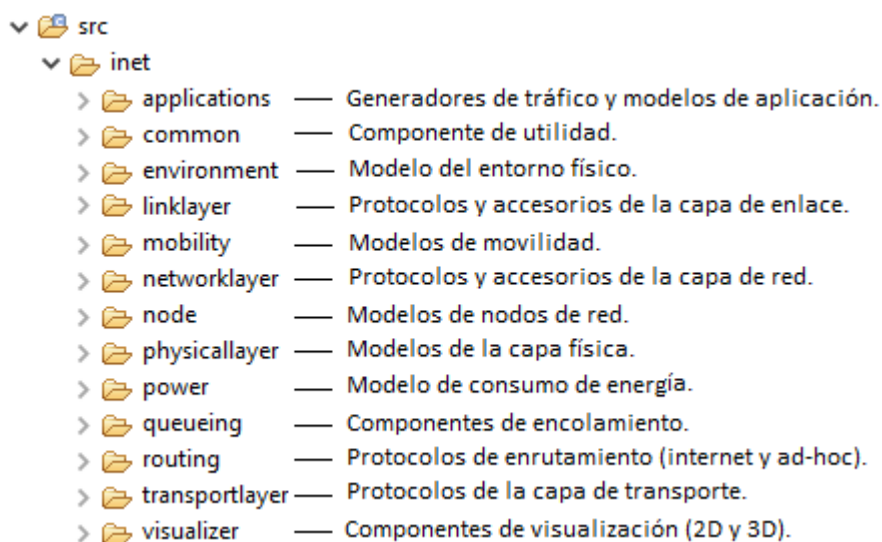
#### **3.1.1 Software (Simulador OMNeT++)**

El uso de simuladores es el enfoque más común para desarrollar o evaluar prototipos, sin la necesidad de tener bancos de pruebas, ya que esto suele ser costoso en términos de tiempo y esfuerzo. También se entiende que el modelo de simulación no puede representar el 100% de un hardware real. Una de las herramientas de simulación disponibles es OMNeT++, siendo este un simulador de eventos discretos basado en C++ para modelar redes de comunicación, multiprocesadores, redes de colas, etc. Esta herramienta es de código abierto y compatible con varios modelos y frameworks entre ellos INET. OMNeT++ proporciona una arquitectura de componentes para modelos, los componentes se programan en C++ y luego se ensamblan en componentes y modelos más grandes utilizando un lenguaje de alto nivel (NED). Al crear un proyecto en OMNeT++ se crean los siguientes archivos:

- Archivo **.ned**, se definen los componentes y las redes de simulación.
- Archivo **.cc**, se guardan los archivos fuente C++ de los componentes.
- Archivo **.msg**, descripción de mensajes que son traducidos automáticamente a clases de C++.
- Archivo **\_m.cc** y **\_m.h**, archivos creados a partir del **.msg** durante el proceso de compilación.
- Archivo **.ini**, setear parámetros y opciones de configuración para la simulación.

OMNeT++ proporciona un framework llamado INET admite una amplia clase de redes de comunicación, incluidas redes cableadas, inalámbricas, móviles, ad hoc y sensores. Contiene modelos para la capa de Internet, capa de enlace, capa física, protocolos de enrutamiento y varios modelos de aplicación. Además, soporta movilidad para nodos, visualización avanzada y emulación de redes.

INET se basa en OMNeT++ y utiliza módulos que se comunican entre si mediante el paso de mensajes. Los módulos están organizados como se muestra en la Figura 3.1.



**Figura 3.1 Módulos del framework INET 4.2.1**

Para realizar la simulación en INET se necesita definir el módulo que representa la red. Posteriormente se definen módulos para el medio de transmisión, entorno físico, visualizadores y otros relacionados con la infraestructura.

Los nodos de red se comunican a nivel de red intercambiando mensajes que son representaciones abstractas de señales físicas en el medio de transmisión. Las señales se envían a través de conexiones en el caso cableado o se envían directamente a la puerta del nodo de red receptor en el caso inalámbrico. Las señales encapsulan paquetes específicos de INET que representan los datos digitales transmitidos. Los paquetes se dividen además en fragmentos que proporcionan representaciones alternativas para datos más pequeños (por ejemplo, encabezados de protocolo, datos de aplicación).

Los modelos de nodo se ensamblan a partir de otros módulos que representan aplicaciones, protocolos de comunicación, interfaces de red, tablas de enrutamiento, modelos de movilidad, modelos de energía y otras funciones.

- Capa de aplicación suelen modelar el comportamiento del usuario y el protocolo de la capa de aplicación (por ejemplo, HTTP ). Las aplicaciones suelen utilizar protocolos de capa de transporte (TCP/UDP ), pero también pueden utilizar directamente protocolos de capa inferior (IP o Ethernet ) a través de sockets.
- Los protocolos de enrutamiento se proporcionan como módulos separados: OSPF, BGP o AODV. Estos módulos utilizan TCP, UDP e IPv4 y manipulan rutas en el módulo ***Ipv4RoutingTable***.
- Los protocolos de la capa de transporte están conectados a aplicaciones y protocolos de la capa de red. Se representan mediante módulos simples, actualmente son compatibles con TCP , UDP y SCTP .
- Los protocolos de la capa de red están conectados a los protocolos de la capa de transporte y las interfaces de red. Por lo general, se

modelan como módulos compuestos: ***Ipv4NetworkLayer*** para IPv4, e ***Ipv6NetworkLayer*** para IPv6. El módulo ***Ipv4NetworkLayer*** contiene varios módulos de protocolo: ***Ipv4***, ***Arp***, ***Icmp*** y ***Icmpv6***.

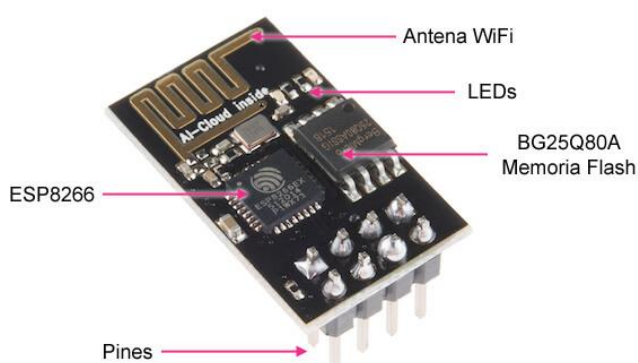
- Las interfaces de red están representadas por módulos compuestos que están conectados a los protocolos de la capa de red y otras interfaces de red en el caso de cableado. A menudo se modelan como módulos compuestos que contienen módulos separados para colas, clasificadores, protocolos MAC y PHY.
- Los protocolos de capa de enlace suelen ser módulos simples que se encuentran en módulos de interfaz de red. Algunos protocolos, por ejemplo, IEEE 802.11 MAC, se modelan como un módulo compuesto debido a la complejidad del protocolo.
- Los protocolos de la capa física son módulos compuestos que también forman parte de los módulos de interfaz de red.
- La tabla de interfaz mantiene el conjunto de interfaces de red (por ejemplo, eth0, wlan0) en el nodo de red. Las interfaces se registran dinámicamente durante la inicialización de las interfaces de red.
- Las tablas de enrutamiento mantienen la lista de rutas para el protocolo de red correspondiente (por ejemplo, ***Ipv4RoutingTable*** para ***Ipv4***). Las rutas se agregan mediante configuradores de red automáticos o protocolos de enrutamiento. Los protocolos de red utilizan las tablas de enrutamiento para encontrar la mejor ruta coincidente para los datagramas.
- Los módulos de movilidad son responsables de moverse por el nodo de la red en la escena simulada. El modelo de movilidad es obligatorio para las simulaciones inalámbricas incluso si el nodo de la red está estacionario. El módulo de movilidad almacena la ubicación del nodo de red que se necesita para calcular la propagación inalámbrica y la pérdida de ruta.

- Los módulos de energía modelan los mecanismos de almacenamiento de energía, el consumo de energía de los dispositivos y los procesos de software, la generación de energía de los dispositivos y los procesos de gestión de la energía que apagan y encienden los nodos de la red.

### 3.1.2 Hardware

Para la implementación existen diferentes modelos de hardware con tecnologías WiFi y BLE en el mercado internacional, sin embargo, en el mercado nacional se logró conseguir los que se detallarán a continuación con el fin de efectuar pruebas y evaluar el desempeño de las tecnologías inalámbricas en un entorno real.

**ESP-01** está basado en el System on Chip (SoC) ESP8266 que ofrece soluciones WiFi altamente integrada de uso eficiente de energía, diseño compacto y rendimiento confiable en la industria de IoT. La plataforma ESP8266 permite el desarrollo de aplicaciones en diferentes lenguajes como: Arduino, Lua, MicroPython, C/C++, Scratch, etc. Además, el ESP8266 integra interruptores de antena, balun RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros y módulos de administración de energía.

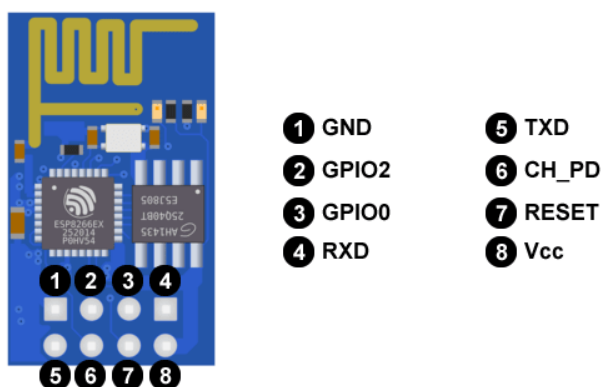


**Figura 3.2 Placa microcontroladora ESP-01 [39]**

Entre las características principales se destacan las siguientes: protocolos soportados 802.11 b/g/ n, Switch TR integrado, balun, LNA,

amplificador de potencia, red de adaptación, soporte de red WiFi 2.4 GHz soporta WPA / WPA2, antena con ganancia 3dBi y patrón de radiación omnidireccional con polarización vertical, stack TCP/IP integrado, potencia de salida 0.15watts y +19.5dBm en modo 802.11b, consumo de energía menor a 10 uA, procesador integrado de 32 bits, comunicación serial y UART, transferencia de paquetes menor a 2ms y 17 pines GPIO que pueden ser asignados a circuitos externos [39].

Los pines de salida del GPIO se muestran en la Figura 3.4 y son los siguientes: GND Tierra, GPIO2 entrada o salida de propósito general, GPIO0 entrada o salida de propósito general, RXD receptor serial, también se puede utilizar como pin digital y corresponde al GPIO3, TXD transmisor serial, también se puede utilizar como pin digital y corresponde al GPIO1, CP\_PD pin para apagar o encender el ESP-01, RESET permite resetear el ESP-01, en bajo se resetea el módulo y VCC [39].



**Figura 3.3. Pines GPIO del ESP-01 [39]**

**Arduino Nano 33 BLE** es una placa compacta construida con el módulo NINA B306 basado en el Nordic nRF52840. El dispositivo viene integrado con sensores:

- Sensor inercial de 9 ejes ideal para dispositivos portátiles.
- Sensor de humedad y temperatura.
- Sensor barométrico.



- Micrófono para captura y analizar el sonido en tiempo real.
- Sensor de gestos, proximidad, color de la luz e intensidad de la luz.



**Figura 3.4 Arduino Nano 33 BLE [40]**

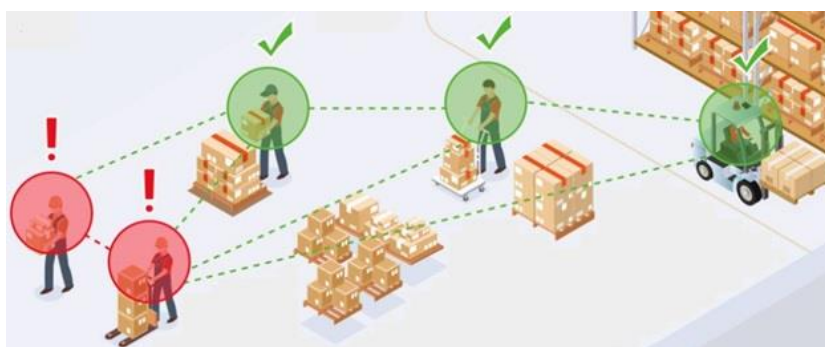
Las características técnicas del Arduino Nano 33 BLE son las siguientes: microcontrolador nRF52840, sensibilidad de -95 dBm en 1Mbps de BLE, potencia del transmisor de -20 a +8dBm, antena sobre el chip balun con RSSI (1dB de resolución), 14 pines de entrada/salida, comunicación UART, SPI e I2C, y 8 pines de entrada analógica (ADC 12 bits 200kmuestras) [40].

El transceptor de radio de 2.4 GHz es compatible con múltiples estándares de radio tales como: 1Mbps, 2Mbps y BLE. El modo IEEE 802.15.4 de 250 kbps es totalmente compatible, así como los modos de operación patentados de 1 Mbps y 2 Mbps por Nordic's. El transceptor incluye un indicador de intensidad de la señal recibida (RSSI) y un contador de bits. El contador de bits genera eventos si se ha enviado o recibido un número preconfigurado de bits [40].

Un paquete de radio contiene los siguientes campos: PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD y CRC. El orden de bytes enviados al medio es siempre el byte menos significativo primero para los campos ADDRESS y PAYLOAD, y el byte más significativo para el campo CRC. El campo ADDRESS se transmiten y reciben desde el bit menos significativo. Los tamaños de los campos S0 y S1 se pueden configurar individualmente a través de S0LEN, LFLEN y S1LEN en PCNF0 respectivamente [40].

### 3.2 Arquitectura del sistema usando las diferentes tecnologías inalámbricas

En esta sección se analizan las arquitecturas de posicionamiento en interior, con el fin de poder establecer una estrategia sólida y sostenible que permita validar el desempeño de diferentes tecnologías inalámbricas para llevar a cabo el distanciamiento social entre las personas, el estudio permite tener un panorama completo para la implementación de un “wearable” de alta precisión desarrollado en el país.

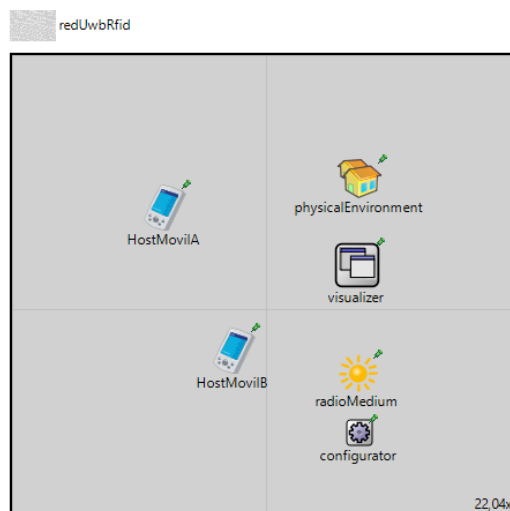


**Figura 3.5. Tags para determinar el distanciamiento social**

Para desarrollar la arquitectura en la simulación se utilizó el programa OMNeT++ 5.6.2 y el framework INET 4.2.1, este framework cuenta con el modelo para el estándar 802.15.4 y 802.11. Se ha creado en **examples** una carpeta con el nombre **aaSimuTec**, donde se encuentran los archivos **red.ned**, **paredes.xml** y **omnetpp.ini**

En el archivo **red.ned** se ha definido la arquitectura para las diferentes tecnologías inalámbricas, para el caso de UWB y RFID se utilizó el modelo del estándar 802.15.4 de INET donde se especifica la capa física y MAC. En el diseño de la red se han colocado los siguientes submódulos: **HostMovilA** y **HostMovilB** que representan los wearables en diferentes posiciones implementan un nodo con interfaz inalámbrica 802.15.4, en **physicalEnvironment** se especifica la dimensión del ambiente para colocar los diferentes objetos creados en el archivo **paredes.xml**, **visualizer** permite visualizar en la consola los diferentes resultados, **radioMedium** para la capa física en este caso se ha utilizado **IEEE802154NarrowbandScalarRadioMedium** donde se va a definir el modelo

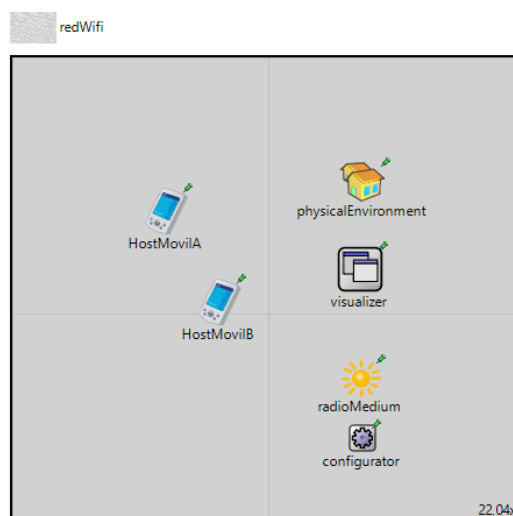
de propagación, fuentes de ruidos e interferencias, y **configurator** este módulo asigna direcciones IPv4 y configura enrutamiento estático.



**Figura 3.6. Arquitectura del sistema con tecnología UWB y RFID**

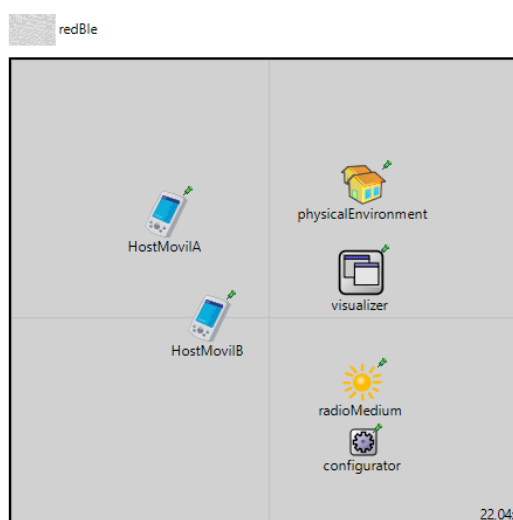
En el caso de la tecnología WiFi se utilizó el estándar 802.11 desarrollado en INET. En el diseño de la red se han colocado los submódulos que se indicaron en la red UWB haciendo los siguientes cambios: **HostMovilA** y **HostMovilB** modelan un host con tarjeta inalámbrica 802.11 en modo de infraestructura y **radioMedium** utiliza el modelo de radio **ieee80211ScalarRadioMedium** con una potencia de transmisión escalar en la representación analógica.

En el caso de la tecnología WiFi se utilizó el estándar 802.11 desarrollado en INET. En el diseño de la red se han colocado los submódulos que se indicaron en la red UWB haciendo los siguientes cambios: **HostMovilA** y **HostMovilB** modelan un host con tarjeta inalámbrica 802.11 en modo de infraestructura y **radioMedium** utiliza el modelo de radio **ieee80211ScalarRadioMedium** con una potencia de transmisión escalar en la representación analógica.



**Figura 3.7. Arquitectura del sistema con tecnología WiFi**

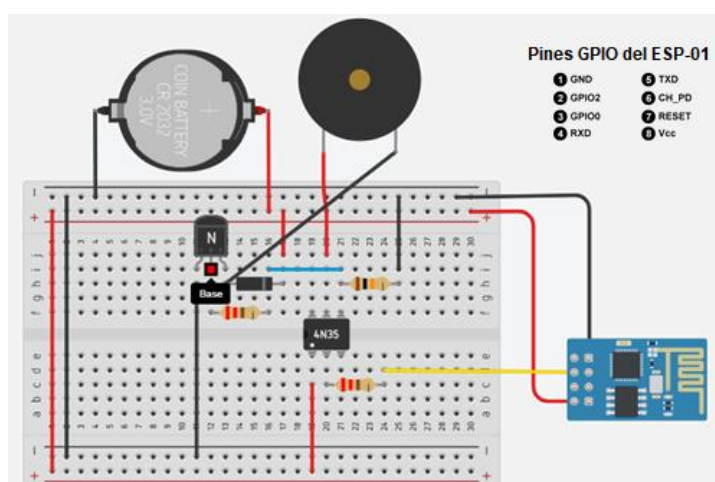
Para la tecnología BLE se desarrolló en INET el estándar 802.15.1 a nivel de la capa física y enlace de datos, el código fuente se muestra en el **Ápndice # 1**. En el diseño de la red se han colocado los submódulos que se indicaron en UWB haciendo los siguientes cambios: **HostMovilA** y **HostMovilB** modelan un host con tarjeta inalámbrica 802.11 en modo de infraestructura y **radioMedium** utiliza el modelo de radio **IRadioMedium** proporciona un comportamiento de capa física muy simple pero rápido y predecible. Debe utilizarse junto con el modelo **IdealRadio**.



**Figura 3.8. Arquitectura del sistema con tecnología BLE**

Se ha descrito el diseño de las redes desarrolladas en el archivo *red.ned*, además en el *Ápndice # 2* se ha colocado el código fuente. En cada una de las redes creadas se ha definido las ubicaciones de los submódulos y las diferentes librerías que se utilizarán del framework INET, para posteriormente inicializarlos en el archivo *omnetpp.ini* con los parámetros correspondientes a cada tecnología.

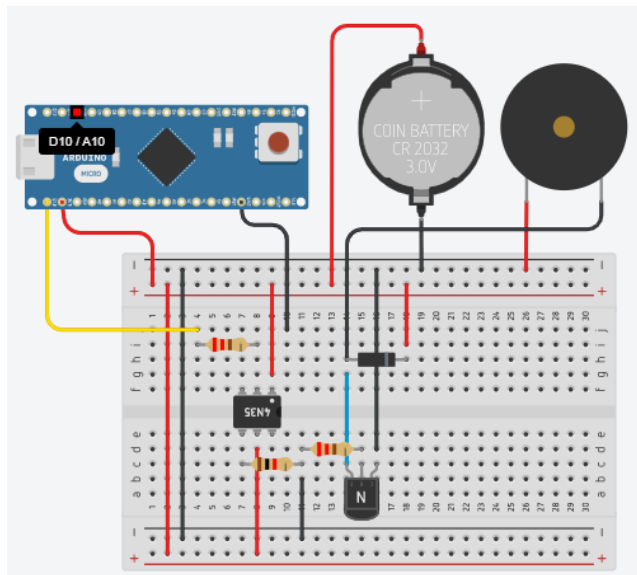
Finalmente se logró implementar dos prototipos con hardware para estimar distancias entre usuarios y validar la precisión de dos tecnologías WiFi y BLE, para esto se utilizó los esquemas de conexión que se muestran en la Figura 3.9 y 3.10 con módulos ESP-01 y Arduino Nano 33 BLE. El hardware utilizado para pruebas son módulos adquiridos en el país y de bajo costo, el microcontrolador fue programado utilizando el lenguaje de programación de Arduino basado en C++.



**Figura 3.9. Diagrama de conexiones con el ESP-01**

El módulo ESP01 se configuran en modo estación con un nombre de red y una contraseña, estos datos son seteados iguales para los dos prototipos. El puerto GPIO 2 está asignado como un pin digital de salida, que en este caso envía un HIGH o LOW hacia el circuito para activar el zumbador. En el caso de obtener un valor de distancia menor a 2mts., el pin GPIO 2 envía un bajo y activa el zumbador durante un segundo a una frecuencia de 10KHz. El

integrado 4N35 es un optoacoplador que permite separar las etapas del módulo WiFi del circuito que activa el zumbador.



**Figura 3.10. Diagrama de conexiones con el Arduino Nano 33 BLE**

El Arduino Nano 33 BLE se configura en modo central y se escanea el valor del Uuid de servicio del módulo. El puerto D13 está asignado como un pin digital de salida, que en este caso envía un HIGH o LOW hacia el circuito para activar el zumbador. En el caso de obtener un valor de distancia menor a 2mts., el pin D13 envía un bajo y activa el zumbador durante un segundo a una frecuencia de 10KHz.

Los prototipos han sido energizados con una batería de 3.3V y se han implementado un par de ellos utilizando la tecnología WiFi y un par utilizando BLE. Además, la cobertura es de 360 grados lo cual implica que al colocar un tercer prototipo no habría inconveniente, ya que el escaneo de redes y la captura del RSSI lo hace en ese radio de cobertura.

### **3.3 Algoritmos de estimación de distancia para las diferentes tecnologías inalámbricas**

Una vez creada las diferentes redes se procede a configurar los parámetros en el archivo **omnetpp.ini** a nivel de la capa de aplicación , física y enlace de datos.

### 3.3.1 Configuraciones de la capa de aplicación

En esta capa para la red UWB, RFID, WiFi y BLE, los diferentes Hosts se inicializan con el módulo **UdpApp** que genera tráfico para una aplicación a través del puerto destino y origen 1000. Además, se indica la comunicación del **HostMovilA** con el **HostMovilB** y viceversa, el tráfico se generará a través de ráfagas con el módulo **BurstyPacketProducer**.

```
*.*.numApps = 1
*.Host*.app[0].typename = "UdpApp"
*.Host*.app[0].io.localPort = 1000
*.HostMovilA.app[0].io.destAddress = "HostMovilB"
*.HostMovilB.app[0].io.destAddress = "HostMovilA"
*.Host*.app[0].io.destPort = 1000
*.Host*.app[0].source.typename = "BurstyPacketProducer"
*.Host*.app[0].source.normalProducer.productionInterval = 1s
*.Host*.app[0].source.normalProducer.packetLength = 100B
*.Host*.app[0].source.burstProducer.productionInterval = 1s
*.Host*.app[0].source.burstProducer.packetLength = 1000B
*.Host*.app[0].source.scheduler.waitIntervals = "40 4"
*.Host*.app[0].source.scheduler.transitionProbabilities="0 1 10"
```

Para el movimiento del **HostMovilB** se utiliza el módulo **LinearMobility**, con una velocidad de 0.4mps dentro del área definida.

```
**.*.updateInterval = 1s
**.*.HostMovilB.mobility.typename = "LinearMobility"
**.*.HostMovilB.mobility.initFromDisplayString = false
**.*.HostMovilB.mobility.speed = 0.4mps
**.*.HostMovilB.mobility.angle = 0deg # degrees
**.*.constraintAreaMinX = 5.5m
**.*.constraintAreaMinY = 5.5m
**.*.constraintAreaMinZ = 0m
**.*.constraintAreaMaxX = 11.5m
**.*.constraintAreaMaxY = 9.5m
**.*.constraintAreaMaxZ = 0m
```

### 3.3.2 Configuraciones de la capa física

La capa más baja en los protocolos de comunicación tiene como principal tarea inicializar modelos tanto analógicos como decisores, dichos modelos se encuentran en el framework INET.

Para la red UWB se realizan las configuraciones del puerto inalámbrico de los hosts utilizando el módulo

**Ieee802154NarrowbandScalarRadioMedium**, donde se definen parámetros tales como: potencia de ruido de fondo, tipo de transmisor y receptor, frecuencia central, ancho de banda, sensibilidad del receptor, potencia de transmisión, SNIR, número de antenas, tipo de modulación. Además, se ha definido el medio inalámbrico con un modelo pathloss (Ecuación 4) llamado **LogNormalShadowing** con propagación constante **ConstantSpeedPropagation** y obstáculos dieléctricos. Para calcular el valor del RSSI se utiliza el módulo **ScalarAnalogModel**.

```
*.mediumType = "Ieee802154NarrowbandScalarRadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 6.48GHz
*.Host*.wlan[0].radio.typename =
"Ieee802154NarrowbandScalarRadio"
*.Host*.wlan[0].radio.transmitter.typename =
"Ieee802154NarrowbandScalarTransmitter"
*.Host*.wlan[0].radio.receiver.typename =
"Ieee802154NarrowbandScalarReceiver"
*.Host*.wlan[0].radio.centerFrequency = 6.48GHz
*.Host*.wlan[0].radio.bandwidth = 500MHz
*.Host*.wlan[0].radio.transmitter.power = 20uW
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"
*.Host*.wlan[0].radio.antenna.numAntennas = 1
*.Host*.wlan[0].radio.*.headerLength = 54b
*.Host*.wlan[0].radio.receiver.errorModel.typename =
"ApskErrorModel"
*.Host*.wlan[0].radio.transmitter.modulation = "BPSK"
*.Host*.wlan[0].radio.receiver.modulation = "BPSK"

*.physicalEnvironment.config = xmldoc("paredes.xml")
*.radioMedium.obstacleLoss.typename = "DielectricObstacleLoss"
*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

*.radioMedium.pathLoss.typename = "LogNormalShadowing"
**.propagation.typename = "ConstantSpeedPropagation"
**.analogModel.typename = "ScalarAnalogModel"
```

Para la red RFID se realizan las configuraciones del puerto inalámbrico de los hosts utilizando el módulo **Ieee802154NarrowbandScalarRadioMedium**, donde se definen



parámetros tales como: potencia de ruido de fondo, tipo de transmisor y receptor, frecuencia central, ancho de banda, sensibilidad del receptor, potencia de transmisión, SNIR, número de antenas, tipo de modulación. Se ha utilizado el mismo medio inalámbrico de la red UWB.

```
*.mediumType = "Ieee802154NarrowbandScalarRadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 865MHz
*.Host*.wlan[0].radio.typename =
"Ieee802154NarrowbandScalarRadio"
*.Host*.wlan[0].radio.transmitter.typename =
"Ieee802154NarrowbandScalarTransmitter"
*.Host*.wlan[0].radio.receiver.typename =
"Ieee802154NarrowbandScalarReceiver"
*.Host*.wlan[0].radio.centerFrequency = 865MHz
*.Host*.wlan[0].radio.bandwidth = 500kHz
*.Host*.wlan[0].radio.transmitter.power = 1.99W
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"
*.Host*.wlan[0].radio.antenna.numAntennas = 1
*.Host*.wlan[0].radio.*.headerLength = 54b
*.Host*.wlan[0].radio.receiver.errorModel.typename =
"ApskErrorModel"
*.Host*.wlan[0].radio.transmitter.modulation = "BPSK"
*.Host*.wlan[0].radio.receiver.modulation = "BPSK"
```

Para la red WiFi se realizan las configuraciones del puerto inalámbrico de los hosts utilizando el módulo ***ieee80211ScalarRadioMedium***, donde se defines parámetros tales como: tasa de transmisión, número de antenas, potencia de transmisión, frecuencia central, sensibilidad del receptor y SNIR. Se ha utilizado el mismo medio inalámbrico definido para la red UWB.

```
**wlan*.bitrate = 11Mbps
**wlan[*].radio.antenna.numAntennas = 8
**wlan[*].radio.transmitter.power = 100mW
**.centerFrequency = 5GHz
**wlan[*].radio.receiver.sensitivity = -85dBm
**wlan[*].radio.receiver.snirThreshold = 4dB
```

Para la red BLE se realizan las configuraciones del puerto inalámbrico de los hosts utilizando el módulo ***ieee802151BLERadioMedium***, donde

se definen parámetros tales como: potencia de ruido, tipo de radio, tipo de transmisor y receptor, número de antenas, tipo de antena, ancho de banda, potencia de transmisión, frecuencia central, sensibilidad del receptor, SNIR y tipo de modulación. Se ha utilizado el mismo medio inalámbrico definido para la red UWB.

```
*.mediumType = "Ieee802151BLERadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 2.45GHz
*.Host*.wlan[0].radio.typename = "Ieee802151BLERadio"
*.Host*.wlan[0].radio.transmitter.typename =
"Ieee802151BLETransmitter"
*.Host*.wlan[0].radio.receiver.typename =
"Ieee802151BLEReceiver"
*.Host*.wlan[0].radio.centerFrequency = 2.45GHz
*.Host*.wlan[0].radio.bandwidth = 2MHz
*.Host*.wlan[0].radio.transmitter.power = 1mW
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"
*.Host*.wlan[0].radio.antenna.numAntennas = 1
*.Host*.wlan[0].radio.*.headerLength = 54b
*.Host*.wlan[0].radio.receiver.errorModel.typename =
"ApskErrorModel"
*.Host*.wlan[0].radio.transmitter.modulation = "GFSK"
*.Host*.wlan[0].radio.receiver.modulation = "GFSK"
```

En el archivo ***paredes.xml*** se ha colocado los siguientes objetos dieléctricos, se define el material, tamaño, posición, orientación y estructura.

```
<environment>
  <object position="min 5 5 0" orientation="0 0 0" shape="cuboid
  0.3 5 3" material="concrete" fill-color="203 65 84"
  opacity="0.4"/>
  <object position="min 5 5 0" orientation="0 0 0" shape="cuboid
  5 0.3 3" material="concrete" fill-color="203 65 84"
  opacity="0.4"/>
  <object position="min 10 5 0" orientation="0 0 0"
  shape="cuboid 0.3 5 3" material="concrete" fill-color="203 65
  84" opacity="0.4"/>
  <object position="min 5 10 0" orientation="0 0 0"
  shape="cuboid 5 0.3 3" material="concrete" fill-color="203 65
  84" opacity="0.4"/>
```

```

<object position="min 7 5.5 0" orientation="0 0 0"
shape="cuboid 1.5 1 1" material="wood" fill-color="203 65 84"
opacity="0.4"/>
<object position="min 9 7 0" orientation="0 0 0" shape="cuboid
0.5 1.5 1" material="wood" fill-color="203 65 84"
opacity="0.4"/>
<object position="min 7 9 0" orientation="0 0 0" shape="cuboid
1 0.5 1" material="wood" fill-color="203 65 84"
opacity="0.4"/>
<object position="min 8.5 9 0" orientation="0 0 0"
shape="cuboid 1 0.5 1" material="wood" fill-color="203 65 84"
opacity="0.4"/>
<object position="min 5.7 8 0" orientation="0 0 0"
shape="cuboid 1 1 1" material="wood" fill-color="203 65 84"
opacity="0.4"/>
</environment>

```

### 3.3.3 Módulos de capa de enlace de datos

En la capa de enlace es donde se procura las funcionalidades de desencapsulación y encapsulación de los mensajes utilizando el manejo básico en el inferior de la capa. Se declaran parámetros como las tasas de datos, longitud de cabecera, potencia de transmisión y estado de los nodos. También se declaran parámetros de tiempos de conmutación de hardware tales como: tiempo para cambiar de modo de suspensión al modo de recepción, tiempo para cambiar de modo de transmisión al modo de recepción y tiempo para cambiar del modo de recepción al de transmisión.

Para la red UWB se utiliza el módulo *ieee802154Mac* que admite Mac-ACK, así como tiempos de retroceso constantes, lineales o exponenciales, además se define la tasa de datos y longitud de la cabecera.

```

*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "WirelessInterface"
*.Host*.wlan[0].mac.typename = "Ieee802154Mac"
*.Host**.bitrate = 6.8Mbps
*.Host*.wlan[*].mac.useMACAcks = false
*.Host*.wlan[*].mac.rxSetupTime = 0s
*.Host*.wlan[*].mac.headerLength = 16B

```

Para la red RFID se utiliza el módulo *ieee802154Mac*, además se define la tasa de datos y longitud de la cabecera.

```

*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "WirelessInterface"
*.Host*.wlan[0].mac.typename = "Ieee802154Mac"
*.Host**.bitrate = 640kbps
*.Host*.wlan[*].mac.useMACAcks = false
*.Host*.wlan[*].mac.rxSetupTime = 0s
*.Host*.wlan[*].mac.headerLength = 16B

```

Para la red BLE se utiliza el módulo **leee802151BLEMac** y se define la tasa de datos.

```

*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "BMacInterface"
*.Host*.wlan[0].mac.typename = "Ieee802151BLEMac"
*.Host**.bitrate = 0.7Mbps

```

Finalmente, en el **Ápndice # 3** se muestran el archivo fuente completo **omnetpp.ini**.

Usando hardware en este caso módulos WiFi se desarrolló el algoritmo que se muestra en el **Ápndice # 4**, este algoritmo nos permite la toma de los valores de RSSI, en primera instancia a cada nodo se le asignó un nombre de la red inalámbrica (SSID) y una contraseña, además se utilizó el parámetro de path-loss con un valor de 2.5 que es el correspondiente a un ambiente de oficina. El valor de la potencia de señal recibida a una distancia de 1 metro en espacio libre se seteo como una constante de valor -63dBm. Luego se procedió a abrir una conexión serie entre el módulo y la computadora para observar los datos de RSSI sensados. Utilizando las diferentes funciones de la librería ESP8266WiFi disponibles en el IDE de Arduino se procedió a escanear todas las redes WiFi disponibles en este caso se valida que el nombre de la red sea "WearableCOVID", si se encuentran redes disponibles se procede a obtener el valor de RSSI mediante la función WiFi.RSSI(). Con el parámetro de RSSI obtenido se calcula el valor de la distancia estimada utilizando la Ecuación 6. Si la distancia calculada es menor que 2mts se imprime en el terminal del IDE de Arduino un mensaje indicando "Distancia peligrosa" y se coloca un valor LOW en el pin digital GPIO 2 el cual permite activar el zumbador, este se activa con una lógica negativa. Los nodos han sido configurados como estaciones WiFi.

El transceptor de BLE implementa un mecanismo para medir la potencia en la señal recibida. Esta característica se llama indicador de intensidad de la señal recibida (RSSI). El RSSI se mide continuamente y el valor se filtra mediante un filtro IIR unipolar. Después de un cambio de nivel de señal, el RSSI se estabilizará después de 15us. El muestreo de la intensidad de la señal recibida se inicia mediante la tarea RSSISTART. La muestra se puede leer del registro RSSISAMPLE. El período de muestreo del RSSI está definido por 0.25us. El RSSISAMPLE mantendrá el filtrado del RSSI recibido después del período de muestreo. El algoritmo desarrollado en lenguaje de Arduino es el que se muestra en el Apéndice # 4.

## CAPÍTULO 4

### 4 ANÁLISIS DE RESULTADOS

#### 4.1 Entorno de pruebas

Para la simulación se realizaron pruebas en un escenario indoor, se han colocado dos Hosts en diferentes posiciones el **HostMovilA** funcionará como transmisor y el **HostMovilB** como receptor, este host receptor se va a mover linealmente con una velocidad de 0.4mps hacia el **HostMovilA**, a diferentes distancias se tomarán los valores de RSSI adquiridos por el módulo **ScalarAnalogModel**.

Se ha definido un ambiente de oficina que abarca un espacio de 10 metros de largo y 8 metros de ancho donde se han colocado cuatros paredes de concreto y cinco objetos de madera que simularán las mesas o sillas como se muestran en la Figura 4.1.

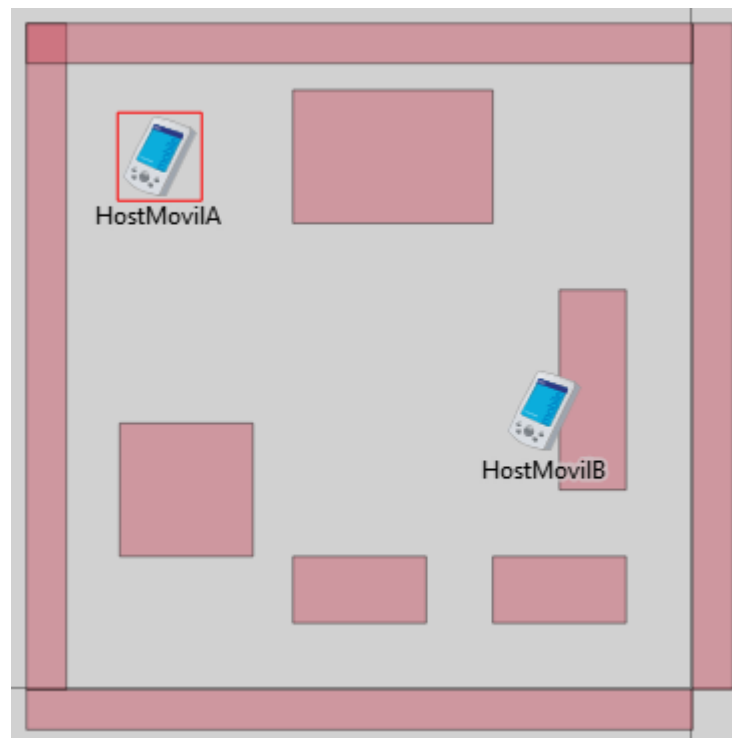
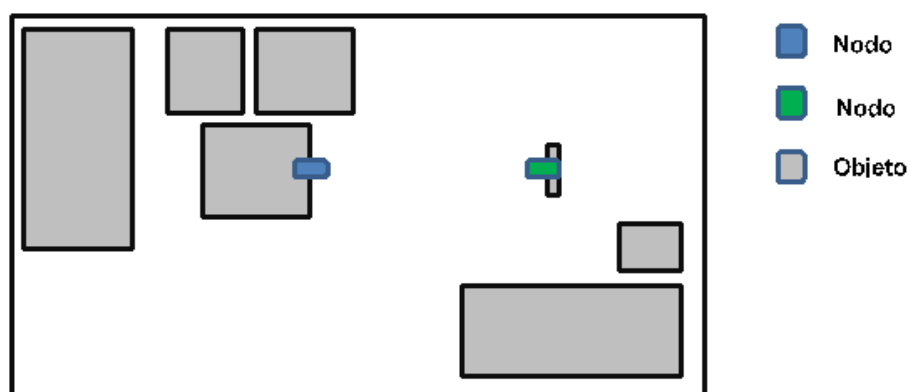


Figura 4.1. Escenario de oficina para la simulación

Para la verificación de la implementación con hardware se utilizó un escenario indoor, en este caso se usó una sala de una casa que abarca un espacio de 8.5 metros de largo y 4 metros de ancho.

En el área que se ha definido para pruebas se encuentran varias estructuras tales como: mesa de madera, sillas metálicas, escritorio y armarios metálicos, que pueden llegar a provocar interferencias en las mediciones del RSSI obtenido por los módulos WiFi, BLE o UWB.

Dentro del escenario definido para pruebas se colocaron dos nodos ya sean WiFi, BLE o UWB como se muestra en la Figura 4.2, estos nodos inicialmente se encuentran a una distancia de 0.6 metros, luego se procedió a alejar el Nodo (verde) cada 0.2 metros, hasta terminar en una distancia de 6 metros de separación entre los dos nodos.

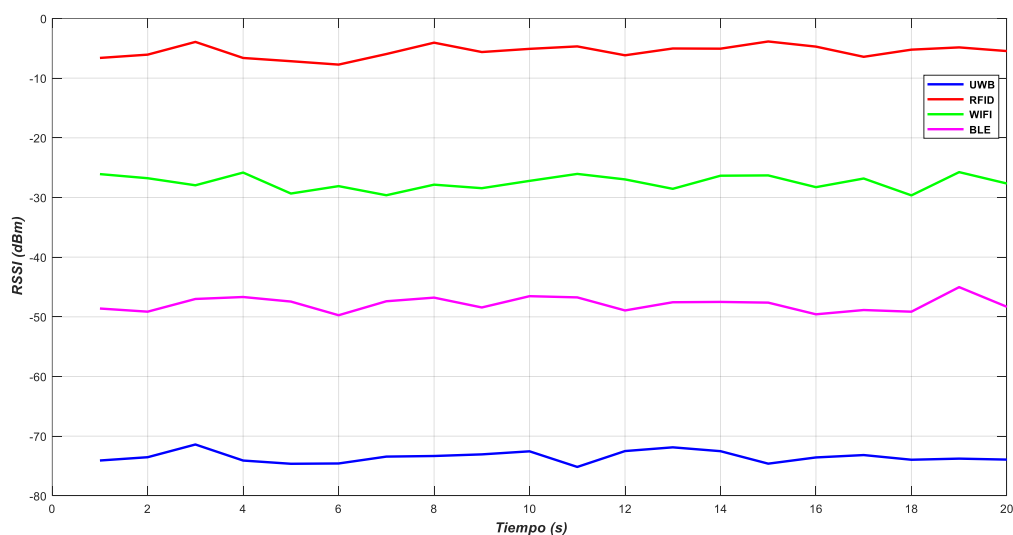


**Figura 4.2. Entorno de pruebas utilizando hardware**

#### 4.2 Estimación de distancias obtenidas mediante simulaciones

Para realizar las simulaciones se procede a ejecutar el archivo **omnetpp.ini** y se toma el valor de RSSI del **HostMovilB**, a una distancia de 2 metros del **HostMovilA**, los valores obtenidos de RSSI fluctúan ya que en el ambiente indoor se tienen paredes, objetos de madera y suelo ocasionando que la señal se refleje, lo que da como resultado que muchas copias del mismo paquete lleguen al transmisor con diferentes intensidades o potencias. En la Figura 4.3 se muestran los datos de fluctuación de RSSI con las tecnologías UWB, RFID, WiFi y BLE a una distancia de 2 metros en un periodo de 20 segundos, al

analizar el conjunto de datos se obtuvo las siguientes desviaciones estándar  $\sigma_{UWB} = 0,94516$  ,  $\sigma_{RFID} = 1,04128$ ,  $\sigma_{WiFi} = 1,22263$  y  $\sigma_{BLE} = 1,18267$ ; con lo cual se puede determinar que la tecnología UWB presenta una menor desviación estándar, los datos se encuentran concentrados alrededor de la media. UWB aparentemente sería una buena opción para la comunicación en presencia de efectos multitrayecto debido a sus menores fluctuaciones de RSSI, pero la importancia de esta métrica debe equilibrarse con la capacidad de relacionar el RSSI con la distancia de manera precisa y confiable.

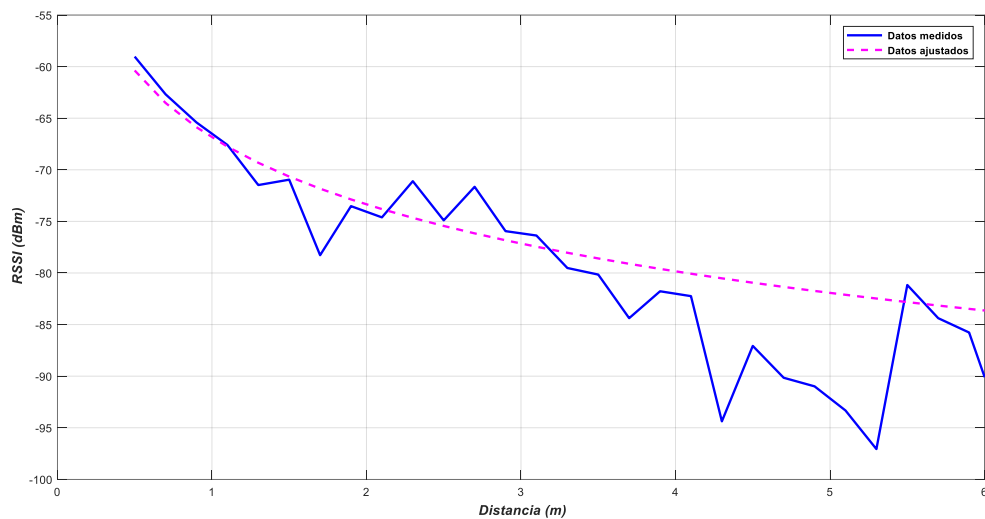


**Figura 4.3. Fluctuaciones de RSSI en un ambiente indoor**

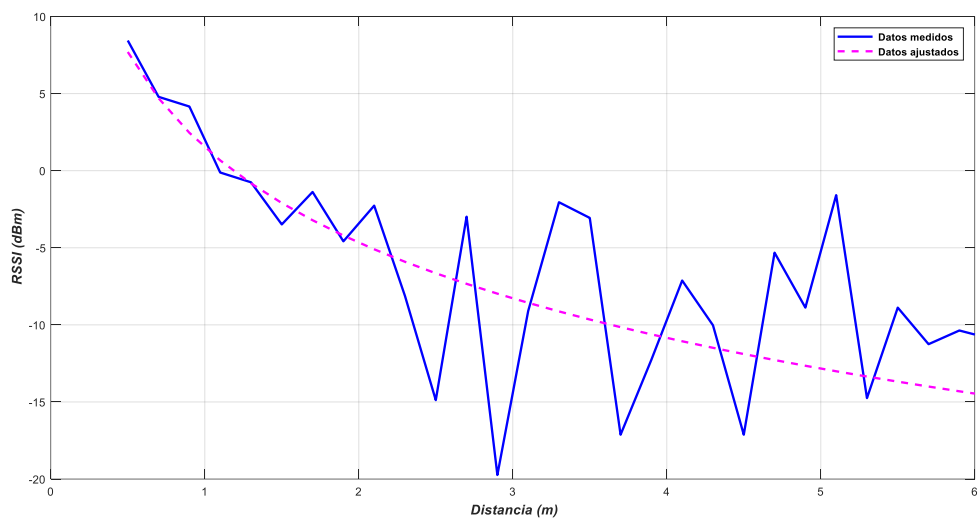
Los valores RSSI medidos suelen diferir de la relación esperada entre la intensidad de la señal y la distancia como se muestra en la Figura 4.4, 4.5, 4.6 y 4.7. El efecto de un entorno interior en la propagación de la señal es la razón principal de que los datos no coincidan con su respectivo modelo teórico. Cualquier paquete puede verse influido por trayectos múltiples a pesar de promediar los resultados. Además, la potencia de transmisión, la sensibilidad del receptor y la orientación de la antena juegan un papel importante en la influencia del RSSI y hacen que los valores de RSSI no coincidan con los valores esperados.



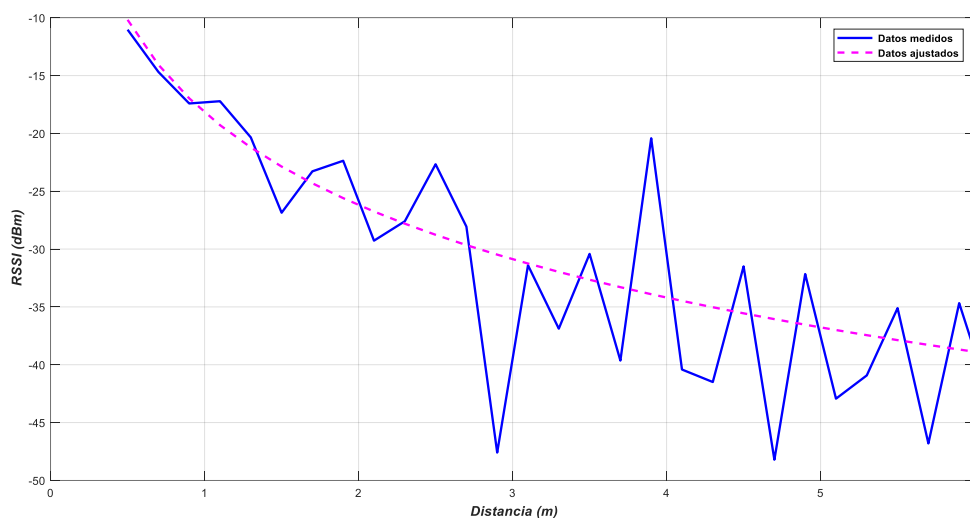
Finalmente, como se mencionó anteriormente, el valor de RSSI fluctúa incluso para distancias fijas entre transmisor y receptor. Sin embargo, estas fluctuaciones varían según la distancia, es decir, cuanto mayor es la distancia entre el transmisor y el receptor, mayores son las variaciones observadas en los valores RSSI medidos, como se muestra en la Figuras 4.4, 4.5, 4.6 y 4.7.



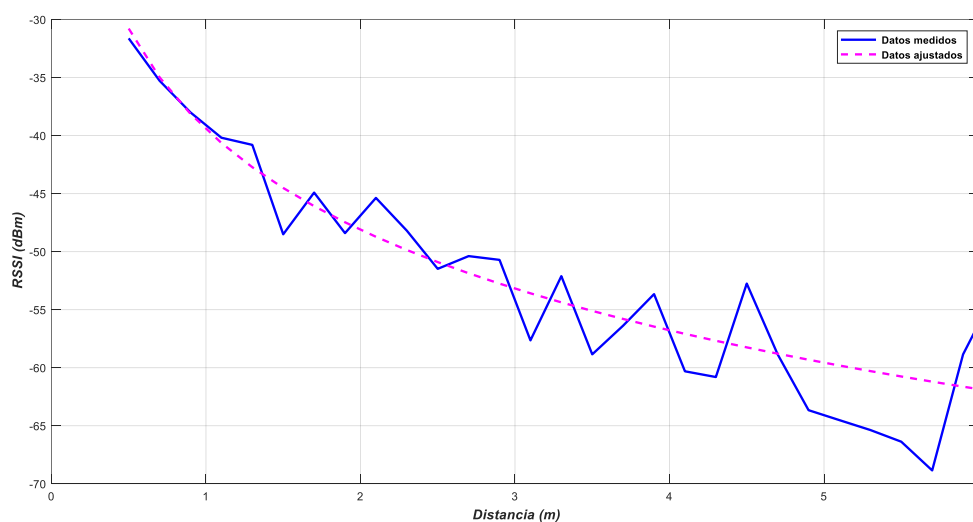
**Figura 4.4. RSSI vs. Distancia usando UWB**



**Figura 4.5. RSSI vs. Distancia usando RFID**



**Figura 4.6. RSSI vs. Distancia usando WiFi**



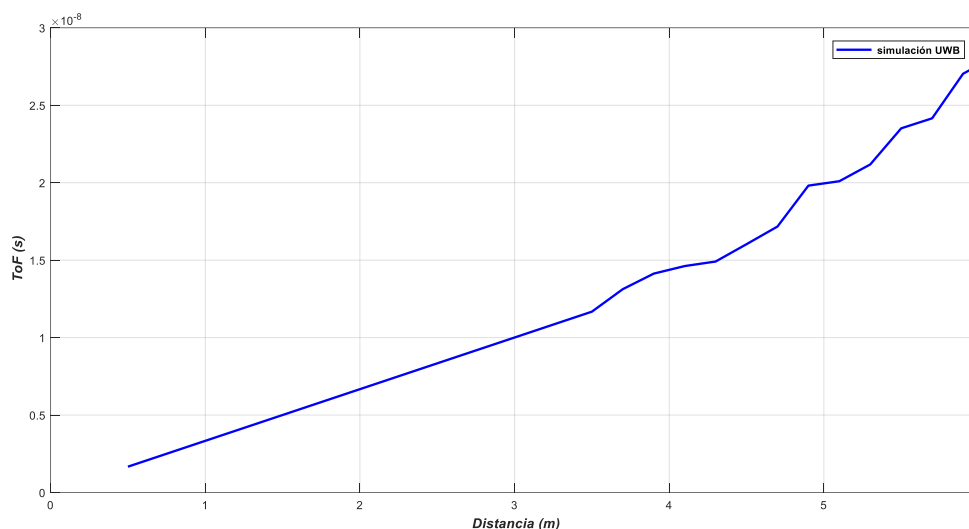
**Figura 4.7. RSSI vs. Distancia usando BLE**

Para cada valor de distancia se tomó el promedio de 10 valores de RSSI, obteniendo los coeficientes de correlación que se muestran en la Tabla 4.1, se puede observar que en este caso las tecnologías UWB y BLE tienen coeficientes de correlación similares cercanos al valor de 1 lo que indica una menor dispersión en los puntos alrededor de la tendencia de la curva, el coeficiente de correlación más bajo se obtuvo con la tecnología RFID mostrando una mayor dispersión de los datos medidos.

**Tabla 4.1 Coeficientes de correlación entre distancia y RSSI**

Tecnología	Coefficiente de correlación
UWB	-0,8945
RFID	-0,6221
WiFi	-0,7922
BLE	-0,8947

Posteriormente se ejecutó la simulación con la tecnología UWB procediendo a tomar los valores de los tiempos de salida y llegada de la señal que viaja del transmisor al receptor y viceversa, con estos datos se calculó el ToF para estimar la distancia de separación de los nodos obteniendo la curva que se muestra en la Figura 4.8, donde se observa una tendencia lineal, conforme aumenta la separación de los nodos también se incrementa el valor de ToF. Para este caso se obtuvo un coeficiente de correlación de 0,9838 lo que indica una menor dispersión de los datos en comparación con la estimación de distancia usando los valores de RSSI.

**Figura 4.8. ToF vs. Distancia usando UWB**

Usando hardware se realizaron varias pruebas a diferentes distancias considerando el entorno descrito en la Figura 4.2. Al colocar los módulos WiFi y BLE a ciertas distancias se procedió a tomar los valores de RSSI, estos valores que se muestran en las Tablas 4.2 y 4.3 son el promedio de 10 muestras. Para calcular la distancia entre módulos se utilizó la Ecuación 6 considerando un valor de coeficiente de path-loss  $n = 2$ .

Posteriormente se utilizó los módulos UWB DWM1001 a diferencia de las tecnologías de radio tradicionales (como Bluetooth o WiFi), UWB opera con el llamado ToF de la señal en lugar de RSSI; en este caso se registraron los valores de ToF como se muestra en la Tabla 4.4 y finalmente para calcular la distancia se utilizó la Ecuación 3. Además, es importante mencionar que para cada valor de distancia de separación de los módulos se tomaron 10 muestras, las mismas que fueron promediadas para lograr una mayor precisión.

En la Tabla 4.2 se puede apreciar que el error de distancia entre los dos prototipos BLE comienza a incrementarse conforme aumenta la separación de los dispositivos a partir 1,4 metros. Usando está tecnología se obtiene un error promedio de 0,38 metros y una desviación estándar de 0.27 metros.

**Tabla 4.2 Distancias estimadas y error usando BLE**

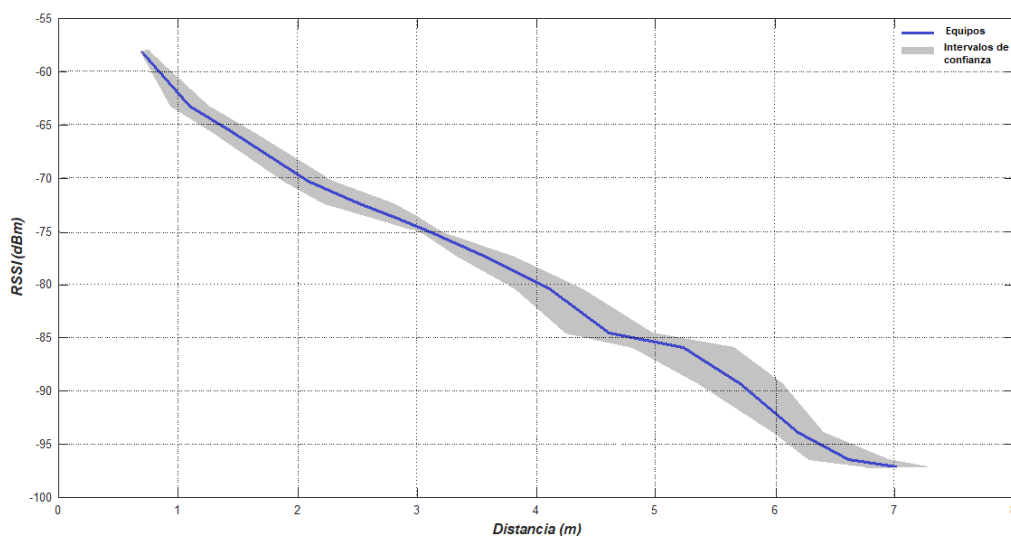
	<b>RSSI promedio (dBm)</b>	<b>Distancia estimada (m)</b>	<b>Distancia real (m)</b>	<b>Error (m)</b>
<b>1</b>	-58,1	0,6918309	0,6	0,0918309
<b>2</b>	-63,2	1,0964782	1,0	0,0964782
<b>3</b>	-65,6	1,4454397	1,4	0,0454397
<b>4</b>	-70,3	2,0892961	1,8	0,2892961
<b>5</b>	-72,4	2,5118864	2,2	0,3118864
<b>6</b>	-73,7	2,8102345	2,6	0,2102345
<b>7</b>	-75,1	3,1267459	3,0	0,1267459
<b>8</b>	-77,3	3,5623101	3,4	0,1623101
<b>9</b>	-80,4	4,1023487	3,8	0,3023487
<b>10</b>	-84,5	4,6198234	4,2	0,4198234
<b>11</b>	-85,9	5,2309198	4,6	0,6309198
<b>12</b>	-89,3	5,7123491	5,0	0,7123491
<b>13</b>	-93,9	6,1937641	5,4	0,7937641
<b>14</b>	-96,4	6,6146212	5,8	0,8146212
<b>15</b>	-97,1	7,0170431	6,2	0,8170431
			<b>Error</b>	<b>0,38833943</b>
			<b>Desviación Estándar</b>	<b>0,27804822</b>

En la Tabla 4.3 se muestra que el error de distancia entre los dos prototipos WiFi comienza a incrementarse conforme aumenta la separación de los mismos. Usando está tecnología se obtiene un error promedio de 0,89 metros y una desviación estándar de 0,41 metros.

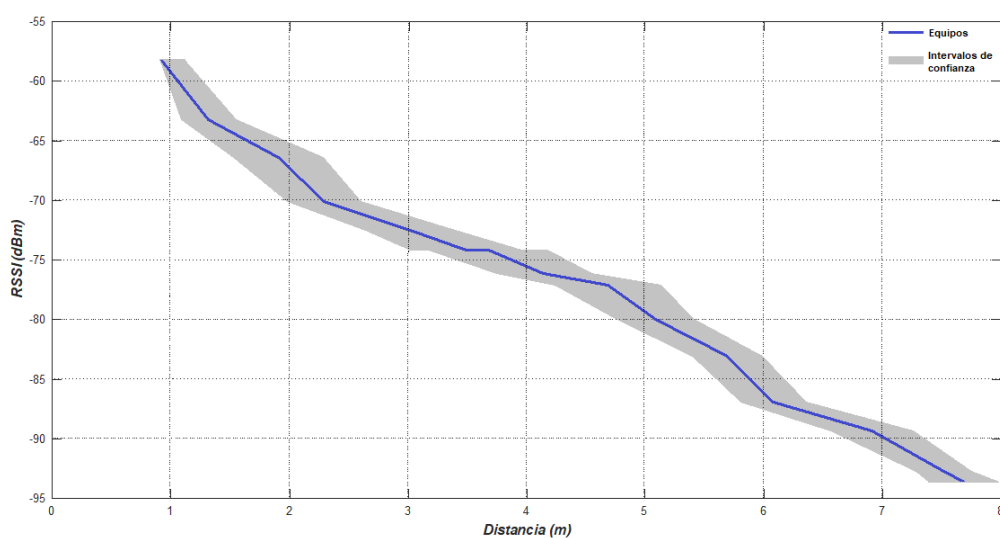
**Tabla 4.3 Distancias estimadas y error usando WiFi**

	<b>RSSI promedio (dBm)</b>	<b>Distancia estimada (m)</b>	<b>Distancia real (m)</b>	<b>Error (m)</b>
<b>1</b>	-58,2	0,9120108	0,6	0,31201084
<b>2</b>	-63,2	1,3182567	1,0	0,31825674
<b>3</b>	-66,4	1,9054607	1,4	0,50546072
<b>4</b>	-70,1	2,2908677	1,8	0,49086765
<b>5</b>	-72,5	3,0199517	2,2	0,81995172
<b>6</b>	-74,2	3,4812345	2,6	0,88123451
<b>7</b>	-74,2	3,6721345	3,0	0,67213451
<b>8</b>	-76,1	4,1456903	3,4	0,74569026
<b>9</b>	-77,1	4,6879245	3,8	0,88792451
<b>10</b>	-80,0	5,0913452	4,2	0,89134521
<b>11</b>	-83,1	5,698234	4,6	1,09823401
<b>12</b>	-86,9	6,0823491	5,0	1,08234911
<b>13</b>	-89,3	6,9024613	5,4	1,50246134
<b>14</b>	-92,7	7,5120456	5,8	1,71204561
<b>15</b>	-93,6	7,6901355	6,2	1,49013551
			<b>Error</b>	<b>0,89400682</b>
			<b>Desviación Estándar</b>	<b>0,41089890</b>

Mediante las pruebas realizadas se logró evaluar las tecnologías BLE y WiFi usando hardware, se puede observar que la desviación estándar es mayor en el caso de WiFi lo cual indica que los datos se encuentran extendidos sobre un rango más amplio de la media o promedio. En la Figura 4.9 y 4.10 se ha graficado las distancias promedio de las 10 muestras con los intervalos de confianza del 95% para cada medición, en este caso se utilizó una distribución t-Student ya que el tamaño de la muestra es inferior a 30. Además, en la Figura 4.9 se puede apreciar que la curva obtenida con las mediciones usando la tecnología BLE, presenta una mejor tendencia, a medida que se aumenta la distancia el valor del RSSI decremente proporcionalmente sin fluctuaciones.



**Figura 4.9. RSSI vs. Distancia para BLE usando hardware**



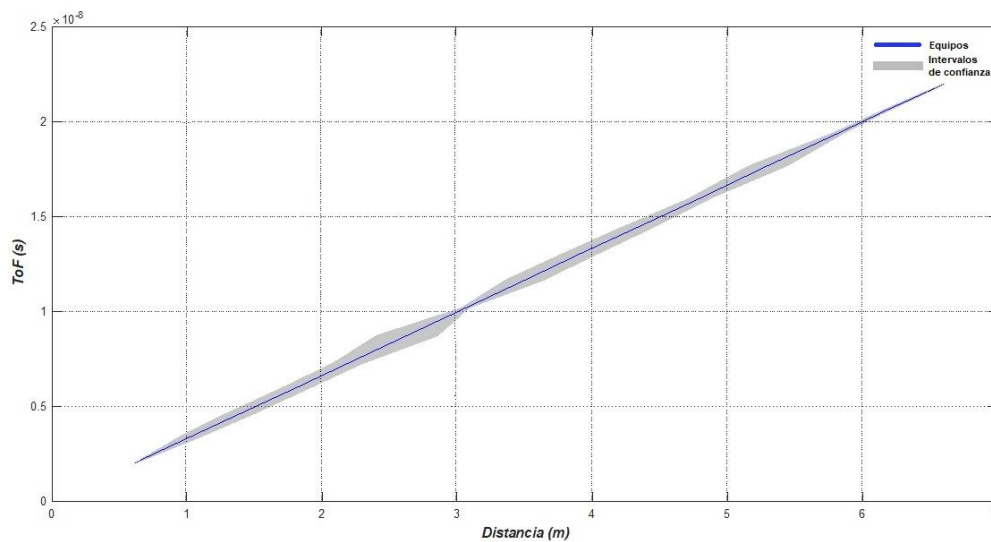
**Figura 4.10. RSSI vs. Distancia para WiFi usando hardware**

En la Tabla 4.4 se puede apreciar que el error promedio entre los dos dispositivos UWB es de 0,14 metros y una desviación estándar de 0,14 metros, lo cual indica que en comparación con las tecnologías anteriores presenta una mejor precisión en estimación de distancias. Finalmente, en la Figura 4.11 se observa una tendencia lineal, a medida que se incrementa la distancia el valor

del ToF disminuye, además se puede observar el intervalo de confianza del 95% alrededor de los datos promedios de distancia.

**Tabla 4.4 Distancias estimadas y error usando UWB**

	ToF calculado (s)	Distancia estimada (m)	Distancia real (m)	Error (m)
<b>1</b>	2,02E-09	0,6066195	0,6	0,00661950
<b>2</b>	3,39E-09	1,01624383	1,0	0,01624383
<b>3</b>	4,63E-09	1,38834566	1,4	0,01165434
<b>4</b>	6,06E-09	1,8167315	1,8	0,01673150
<b>5</b>	7,28E-09	2,18257933	2,2	0,01742067
<b>6</b>	8,74E-09	2,62210239	2,6	0,02210239
<b>7</b>	1,01E-08	3,05004594	3,0	0,05004594
<b>8</b>	1,16E-08	3,49231241	3,4	0,09231241
<b>9</b>	1,30E-08	3,91321367	3,8	0,11321367
<b>10</b>	1,44E-08	4,33456689	4,2	0,13456689
<b>11</b>	1,59E-08	4,79012345	4,6	0,19012345
<b>12</b>	1,76E-08	5,30187645	5,0	0,30187645
<b>13</b>	1,93E-08	5,79123045	5,4	0,39123045
<b>14</b>	2,06E-08	6,19459312	5,8	0,39459312
<b>15</b>	2,20E-08	6,60125678	6,2	0,40125678
			<b>Error promedio</b>	<b>0,14399943</b>
			<b>Desviación Estándar</b>	<b>0,14820661</b>



**Figura 4.11. ToF vs. Distancia usando UWB**

## CAPÍTULO 5

### 5 ANÁLISIS DE FACTIBILIDAD DEL DISEÑO PROPUESTO

#### 5.1 Análisis comparativo de las tecnologías de posicionamiento en interiores en función de la exactitud

Para evaluar la precisión en estimación de distancias usando diferentes tecnologías aplicadas en ambientes interiores se realizaron pruebas en un ambiente de simulación y en un entorno real con hardware.

Para estas pruebas se usaron dos técnicas para medir distancias entre dos dispositivos. Una de las opciones se basa en el RSSI donde la intensidad de la señal disminuye con el aumento de la distancia desde el transmisor de una manera determinista que se basa en fórmulas teóricas, con esta suposición, podemos estimar la distancia entre un receptor y un transmisor. Este enfoque tiene varias desventajas dado que el entorno, canal de radio, cambia constantemente, por lo tanto, también cambia el parámetro RSSI, que a su vez trae inexactitud al sistema. El parámetro RSSI también se puede degradar por la propagación multitrayectoria y otros fenómenos que son bastante comunes para el canal de radio. Las tecnologías como WiFi, BLE y RFID generalmente se basan en este proceso de estimación de distancia, en las pruebas realizadas en el trabajo también se utilizó la tecnología UWB con esta técnica, logrando determinar en la simulación que la tecnología BLE y UWB presentan un desempeño similar ya que el coeficiente de correlación entre las dos variables: distancia y RSSI, muestran una menor dispersión de datos. Con hardware se utilizó la tecnología BLE y WiFi obteniendo un menor error promedio y desviación estándar con módulos Arduino Nano 33 BLE.

Otra opción es utilizar el tiempo de vuelo de la señal en lugar de RSSI. Esto produce resultados mucho más precisos en entornos de línea de vista (LOS) y puede conducir a una precisión de un centímetro según la frecuencia y la naturaleza de la señal. Este es un enfoque que utiliza la tecnología UWB. La



naturaleza de las señales UWB lo convierte en un candidato ideal para utilizar el proceso de estimación de distancia de tiempo de vuelo, ya que utiliza un tren de impulsos en lugar de una onda sinusoidal modulada para transmitir información, esta característica hace que el pulso ocupe una banda de frecuencia tan amplia, su borde ascendente es muy pronunciado y esto permite al receptor medir con mucha precisión el tiempo de llegada de la señal. Además, los pulsos en sí son muy estrechos, por lo general no más de dos nanosegundos, pudiéndose distinguir incluso en entornos ruidosos. También debido a la máscara espectral, la potencia de transmisión se encuentra en el piso de ruido, lo que significa que UWB no interfiere con otros sistemas de comunicación por radio que operan en las mismas bandas de frecuencia similar a las tecnologías de espectro ensanchado (CDMA).

El rendimiento puede verse degradado, por ejemplo, al oscurecer la línea de vista entre dispositivos UWB, sin embargo, la precisión general sigue siendo superior en comparación con RSSI, mediante simulación se logró determinar que el coeficiente de correlación es 0,9838 obteniendo una menor dispersión de datos en comparación con las medidas de RSSI. Finalmente se realizó las pruebas con módulos DW1000 de tecnología UWB que miden el ToF mostrando un error promedio de 0,14 metros; valor menor a las mediciones realizadas con módulos BLE y WiFi utilizando el RSSI, cabe mencionar que los módulos DW1000 utilizan seis canales de radio, esto le da flexibilidad en caso de existir interferencia en un canal se puede cambiar fácilmente a uno diferente.

## **5.2 Análisis comparativo de las tecnologías de posicionamiento en interiores en función de la escalabilidad**

Uno de los factores críticos para el crecimiento de estos sistemas es la escalabilidad, las implementaciones de estas tecnologías tienen como principal objetivo un crecimiento en el uso de los wearables, para lo cual el rendimiento actual no debe verse afectado con el incremento de los mismos.

La inexactitud y la falta de fiabilidad son problemas comunes en las tecnologías inalámbricas que utilizan RSSI, como BLE y WiFi. Es posible

posicionarse a unos pocos metros, pero los objetos en movimiento y otros obstáculos a menudo desvían la precisión de manera significativa. De las revisiones de literatura se obtuvo que para el caso de WiFi una de las limitantes es la escalabilidad debido al límite del espacio de espectro disponible y de los puntos de acceso; esta tecnología puede ser escalable hasta 100 o 1000 tags. BLE tiene un rendimiento similar al de WiFi dado que utiliza mecanismos basados en flooding, lo cual genera complicaciones al momento de incrementar el tamaño de la red. RFID no presenta problemas en cuanto a escalabilidad se puede colocar la cantidad de tags necesarios en un ambiente indoor sin afectar el rendimiento. Finalmente, para UWB los fabricantes de chips DW1000 afirman que poseen una tasa de actualización máxima de 8000 actualizaciones/seg/canal y que el sistema puede localizar con precisión un máximo de 7500 tags por segundo en un radio de 20 metros, sin embargo, esto depende de las configuraciones a nivel de la capa MAC y física.

### **5.3 Análisis comparativo de las tecnologías de posicionamiento en interiores en función de costos**

En esta sección se presentará el costo aproximado de la realización o importación de los wearables para cada tecnología, dicho desglose se lo realizará por etapas, cada etapa será cuantificada para estimar los recursos y determinar un presupuesto aproximado.

Para la realización del trabajo se han identificado cuatro etapas:

- Etapa 1: Estudio del estado del arte.
- Etapa 2: Diseño de la aplicación del wearable.
- Etapa 3: Resultados de la aplicación del wearable.
- Etapa 4: Implementación y documentación del wearable.

Los costos se van a dividir en dos partes: costos de los materiales y del personal, en estos valores se incluye el tiempo invertido.

**Costo de materiales:** se detalla en la Tabla 5.1 los componentes para la realización del wearable usando tecnología UWB obteniendo un costo de materiales de \$79,78, para las demás tecnologías lo que varía es el costo del módulo, en el caso de BLE el módulo tiene un costo de \$30 el costo total sería \$85,88. El módulo WiFi tiene un costo de \$4,85, el costo total sería \$60,73. Finalmente para la tecnología RFID el costo del módulo es aproximadamente \$5,00, el costo total de materiales sería \$60,88.

**Tabla 5.1 Costos de materiales para implementación de wearable UWB**

	<b>Componente</b>	<b>Costo Unitario</b>	<b>Costo Total</b>
1	Módulo DWM1001C	\$23,90	\$23,90
2	LED estándar – SMD (RGB)	\$0,30	\$0,30
3	LED amarillo 0603 SMD	\$0,38	\$0,38
4	LED rojo DIFFUSED 0603 SMD	\$0,46	\$0,46
5	Resistor SMD 1K OHM 5% 1/10W 0603	\$0,10	\$0,10
6	Resistor SMD 100K OHM 1% 1/10W 0603	\$0,10	\$0,10
7	Resistor SMD 140K OHM 1% 1/10W 0603	\$0,10	\$0,10
8	Resistor SMD 0 OHM JUMPER 1/10W 0603	\$0,10	\$0,10
9	Capacitor Cerámico 1UF 10V X7R 0805	\$0,19	\$0,19
10	Capacitor Cerámico 10UF 10V X5R 0603	\$0,36	\$0,36
11	IC Controlador de Motor 2.5V-6.5V SOT23-5	\$2,76	\$2,76
12	Motor de Vibración 9000 RPM 3V	\$2,88	\$2,88
13	Conector SMD 10POS 1.27MM	\$0,74	\$0,74
14	Switch Slide SPDT 300MA 6V	\$0,63	\$0,63
15	Soporte para batería de botón 20MM PC PIN	\$0,78	\$0,78
16	PCB (incluye Lamina FR4, SolderMask, Produconduct)	\$16,00	\$16,00
17	Impresión de carcasa para wearable en PLA	\$30,00	\$30,00
<b>TOTAL</b>			<b>\$79,78</b>


**Costo de personal:** en esta sección se detallan los honorarios de las personas podrían participar en el proyecto. El cual está dirigido por un Jefe de proyecto e Ingenieros encargados del desarrollo de los wearables.

**Tabla 5.2 Costos de personal para desarrollo de wearables**


<b>Ocupación</b>	<b>Cantidad</b>	<b>Costo</b>
Jefe de Proyecto	1	\$2000,00
Ingenieros	4	\$4800,00
<b>TOTAL</b>		<b>\$6800,00</b>

Los precios de los materiales se obtuvieron de proveedores locales. A continuación, se presenta el costo total del wearable por tecnología.


**Tabla 5.3 Costos del wearable para tecnología UWB**

		
Ítem	Costo Unitario	Costo total
Diseño y preparación		\$6800,00
Costos de materiales (10)	\$79,78 c/u	\$ 797,80
Subtotal Inversión		\$7597,80
IVA 12%		\$ 911,74
<b>TOTAL</b>		<b>\$8509,54</b>


**Tabla 5.4 Costos del wearable para tecnología RFID**

		
Ítem	Costo Unitario	Costo total
Diseño y preparación		\$6800,00
Costos de materiales (10)	\$60,88 c/u	\$ 608,80
Subtotal Inversión		\$7408,80
IVA 12%		\$ 889,06
<b>TOTAL</b>		<b>\$8297,86</b>

**Tabla 5.5 Costos del wearable para tecnología WiFi**

		
Ítem	Costo Unitario	Costo total
Diseño y preparación		\$6800,00
Costos de materiales (10)	\$60,73 c/u	\$ 607,30
Subtotal Inversión		\$7407,30
IVA 12%		\$ 888,87
<b>TOTAL</b>		<b>\$8296,17</b>

**Tabla 5.6 Costos del wearable para tecnología BLE**

		
Ítem	Costo Unitario	Costo total
Diseño y preparación		\$6800,00
Costos de materiales (10)	\$85,88 c/u	\$ 858,80
	Subtotal Inversión	\$7658,80
	IVA 12%	\$ 919,06
	<b>TOTAL</b>	<b>\$8577,86</b>

En base a los costos obtenidos para el diseño e implementación de 10 wearables en Ecuador se obtuvo que la tecnología WiFi y RFID serían la más económicas.

#### 5.4 Análisis comparativo de las tecnologías de posicionamiento en interiores en función de componentes disponibles en el mercado

Los componentes que se van a utilizar para el desarrollo del wearable se encuentran disponibles en el mercado con entrega inmediata, sin embargo, para la adquisición de los módulos de las diferentes tecnologías se tienen diferentes plazos de entrega de fábrica, a continuación, se detallan los tipos de módulos.

**Tabla 5.7 Módulo UWB**

	<b>Descripción</b>
	DWM1001C
	<b>Radio Frecuencia</b>
	UWB - 6.5Ghz
	<b>Plazo de entrega de fábrica</b>
14 semanas	

Tabla 5.8 Módulo RFID

	<b>Descripción</b>
	Rc522 tarjeta, lector y llavero
	<b>Radio Frecuencia</b>
	UWB - 6.5Ghz
	<b>Plazo de entrega de fábrica</b>
1 semana	

Tabla 5.9 Módulo WiFi

	<b>Descripción</b>
	ESP-01 -WiFi
	<b>Radio Frecuencia</b>
	2.4 GHz
	<b>Tiempo estimado</b>
Inmediata	

Tabla 5.10 Módulo BLE

	<b>Descripción</b>
	Nano 33 BLE Sense
	<b>Radio Frecuencia</b>
	BLE 2.4 GHz
	<b>Tiempo estimado</b>
2 semanas	

### 5.5 Tabla comparativa de las tecnologías de posicionamiento en interiores para la implementación de wearables

Existen muchas tecnologías inalámbricas que permiten estimar distancias o posiciones en ambientes indoor. Cada una de ellas tienen sus propias ventajas y desventajas en cuanto a precisión, precio, cobertura de señal, latencia, consumo de potencia, escalabilidad, etc. Siempre depende del cliente decidir

qué tipo de rendimiento espera del sistema que desea implementar. Si bien las tecnologías tradicionales como WiFi, Bluetooth LE y RFID activa pueden llevar la precisión a varios metros, UWB es más adecuada para aplicaciones donde la precisión de distancias o posicionamiento es un parámetro crítico.

En la Tabla 5.11 se muestra una comparación entre las diferentes tecnologías usadas para estimar distancias o posicionamiento en ambientes indoor, se puede ver claramente que cada tecnología tiene sus inconvenientes, depende del cliente decidir qué tipo de tecnología es la más adecuada para su aplicación o uso.

**Tabla 5.11 Comparación entre tecnologías usadas en ambientes indoor.**

<b>Tecnología</b>	<b>UWB</b>	<b>RFID</b>	<b>WiFi</b>	<b>BLE</b>
Métodos de medición	ToF	RSSI	RSSI	RSSI
Cobertura	Típico: 70 metros Máx.: 250 metros	Típico: 1 metro Máx.: 5 metros	Típico: 50 metros Máx.: 150 metros	Típico: 15 metros Máx.: 100 metros
Precisión	Centímetros	Centímetros a 1 metros	5 a 15 metros	1 a 5 metros
Fiabilidad	Fuerte inmunidad a las interferencias y multitrayectoria	Penetra objetos sólidos no metálicos; no requiere LOS	Muy sensible a trayectos múltiples, obstrucciones e interferencias	Muy sensible a trayectos múltiples, obstrucciones e interferencias
Velocidad de transmisión de datos	Hasta 27Mbps	----	Hasta 1Gbps	Hasta 2Mbps
Seguridad de capa física	Protocolo delimitado por distancia-tiempo	Se puede falsificar mediante un ataque de retransmisión	Se puede falsificar mediante un ataque de retransmisión	Se puede falsificar mediante un ataque de retransmisión
Escalabilidad	Mayor a 10000 tags	No limitado	100 a 1000 tags	100 a 1000 tags
Potencia	5nJ/b TX – 9nJ/b RX	Pasivo	50nJ/b TX/RX	15nJ/b TX/RX
Costo de los 10 primeros prototipos	\$8509,54	\$8297,86	\$8296,17	\$8557,86

## CONCLUSIONES

Del trabajo realizado mediante el uso del simulador OMNeT++ y pruebas de hardware en un ambiente real se concluye lo siguiente:

- Se analizaron las arquitecturas de las diferentes tecnologías inalámbricas de posicionamiento en interiores determinando que la disponibilidad, el costo, la confiabilidad y la precisión de la localización son los parámetros más importantes al seleccionar un método y tecnología. Entre las tecnologías existentes, BLE y WiFi son opciones populares debido a su disponibilidad (disponible en dispositivos inteligentes modernos) y bajo costo. Sin embargo, tienen una deficiencia significativa en ambientes interiores.
- Se utilizaron las técnicas del RSSI y ToF para medir la distancia entre dos dispositivos en simulación, en este caso se pudo comprobar que la tecnología que presenta un mejor coeficiente de correlación entre las variables distancia y RSSI es BLE seguido por UWB. Al realizar pruebas con hardware en un ambiente indoor se determinó que el error promedio con la tecnología BLE es 38,8 cm y en WiFi es 89,4 cm usando el RSSI, también se utilizaron módulos UWB con la técnica ToF obteniendo un error promedio de 14,3 cm. Los modelos de propagación usando el RSSI se ven gravemente afectados por cualquier objeto atenuador de RF, incluidos los cuerpos humanos, y por la orientación del receptor cuando lo sostiene un ser humano. Por lo tanto, se propone el diseño del wearable con la tecnología UWB y la técnica ToF ya que se obtuvo una alta precisión en estimación de distancias en comparación con la técnica del RSSI y otras tecnologías.
- Del estudio de factibilidad se obtuvo que para la construcción de 10 prototipos de wearables con componentes disponibles en Ecuador se necesita aproximadamente \$8509,54, haciendo uso de módulos UWB que es la tecnología que presenta mejores resultados en términos de precisión, escalabilidad, seguridad y potencia. Sin embargo, los costos se reducirán considerablemente al fabricarlos a escala, como referencia se tiene el wearable comercial Safe Spacer con tecnología UWB a un precio de \$99,99.



## RECOMENDACIONES

Durante el desarrollo del trabajo, se encontraron algunas variables que pueden influir en el diseño de los wearables, por lo que se tiene las siguientes recomendaciones:

- Realizar pruebas con equipos más sensibles, que arrojen valores de RSSI confiables para ajustar un modelo de pérdidas más exacto.
- Validar la escalabilidad y seguridad a nivel de la capa física de las diferentes tecnologías en ambientes indoor.
- Establecer diferentes ambientes indoor con varios módulos UWB en la simulación para validar el error promedio o coeficiente de correlación.
- Investigar el uso de módulos UWB con una mejor precisión y de tal manera que se pueda realizar un wearable con dimensiones inferiores a las que existen en el mercado, para que el usuario final lo pueda portar fácilmente.

## BIBLIOGRAFÍA

- [1] A. Albán, «Coronavirus: Trabajar en Guayaquil, a dos metros de distancia,» *Diario Expreso*, 4 Mayo 2020.
- [2] M. A. S. M. Maanak Gupta, «Enabling and Enforcing Social Distancing Measures using Smart City and ITS Infrastructures: A COVID-19 Use Case,» p. 5, 2020.
- [3] M. Peel y D. Mancini, «One metre or two? Economic pressure stokes distancing debate,» *Financial Times*, 11 Junio 2020.
- [4] F. Zafari, A. Gkelias y K. Leung, «A Survey of Indoor Localization Systems and Technologies,» *IEEE Communications Surveys & Tutorials*, pp. 2568 - 2599, 2019.
- [5] IK Multimedia Production, «Applications. Von Safe Spacer,» Junio 2020. [En línea]. Available: <https://www.safespacer.net/>.
- [6] H. Zhang, Z. Zhang, N. Gao, Y. Xiao, Z. Meng y Z. Li, «Cost-effective wearable indoor localization and motion analysis via the integration of UWB and IMU,» *Sensors*, vol. 20, nº 2, p. 344, 2020.
- [7] M. Girolami, F. Mavilia y F. Delmastro, «Sensing social interactions through BLE beacons and commercial mobile devices,» *Pervasive Mob. Comput.*, p. 101198, 2020.
- [8] D. Thai, H. Nguyen, C. Huynh, N. Saputra, Y. Nguyen, T. Nguyen, D. Vu, T. Dutkiewicz, E. Chatzinotas y Symeon., «Enabling and Emerging Technologies for Social Distancing: A Comprehensive Survey,» 2020.
- [9] B. Islam, M. Uddin, S. Mukherjee y S. Nirjon, «Rethinking Ranging of Unmodified BLE Peripherals in Smart City Infrastructure,» *Proceedings of the 9th ACM Multimedia Systems Conference*, pp. 339-350, Junio 2018.
- [10] A.R. Jimenez y F. Seco, «Finding objects using UWB or BLE localization technology: A museum-like use case,» *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, pp. 1-8, 2017.

- [11] Yongchang He, «Experimental setup for Bluetooth low energy ranging application,» *Embedded Systems, Department of Mathematics and Computer Science*, 2016.
- [12] C. Nguyen, Y. Saputra, N. Huynh, N. Nguyen, T. Khoa, B. Tuan, D. Nguyen, D. Hoang, T. Vu, E. Dutjiewicz, S. Chatzinotas y B. Ottersten, «A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing—Part I: Fundamentals and Enabling Technologies,» *IEEE Access*, vol. 8, pp. 153479-153507, 2020.
- [13] M. Laka Iñurrategi, «Sistema de localización y seguimiento de personas en interiores mediante cámara PTZ basado en las tecnologías Kinect y Ubisense,» Universidad del País Vasco / Euskal Herriko Unibertsitatea.
- [14] G. Chavira, S. W. Nava, R. Hervas, J. Bravo y C. Sanchez, «Combining RFID and NFC Technologies in an Aml Conference Scenario,» *Eighth Mexican International Conference on Current Trends in Computer Science*, pp. 165-172, 2007.
- [15] R. Ramanathan y J. Imtiaz, «NFC in industrial applications for monitoring plant information,» *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-4, 2013.
- [16] D. Dardari, P. Closas y P. M. Djurić, «Indoor Tracking: Theory, Methods, and Technologies,» *IEEE Transactions on Vehicular Technology*, vol. 64, nº 4, pp. 1263-1278, Abril 2015.
- [17] D. Zhang, F. Xia, Z. Yang, L. Yao y W. Zhao, «Localization Technologies for Indoor Human Tracking,» *5th International Conference on Future Information Technology*, pp. 1-6, 2010.
- [18] Y. Huang, J. Zheng, Y. Xiao y M. Peng, «Robust Localization Algorithm Based on the RSSI Ranging Scope,» *International Journal of Distributed Sensor Networks*, pp. 1-8, 2015.
- [19] K. Zhang, Y. Zhang y S. Wan, «Research of RSSI indoor ranging algorithm based on Gaussian - Kalman linear filtering,» *IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1628-1632, 2016.

- [20] P. Wang y Y. Luo, «Research on WiFi Indoor Location Algorithm Based on RSSI Ranging,» *4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1694-1698, Julio 2017.
- [21] «Estimote,» 2012. [En línea]. Available: <https://estimote.com/wearable/>.
- [22] Saira Asher, «BBC News,» 4 Julio 2020. [En línea]. Available: <https://www.bbc.com/news/technology-53146360>.
- [23] «Tsingoa (Beijing) Technology Co,» Social distancing and contact tracing, 2020. [En línea]. Available: [http://www.tsingoa.com/product\\_/](http://www.tsingoa.com/product_/).
- [24] IK Multimedia Production srl, [En línea]. Available: <https://www.safespacer.net/?L=ES>.
- [25] Rombit, [En línea]. Available: <https://rombit.be/romware-wearables/>.
- [26] Corporación de Tecnología PROXXI, [En línea]. Available: <https://www.proxxi.co/contact>.
- [27] Samsung, [En línea]. Available: <https://www.samsung.com/us/business/solutions/topics/wearables>.
- [28] Equivital, [En línea]. Available: <https://www.equivital.com/products/social-distancing-device>.
- [29] Tended, [En línea]. Available: <https://www.tended.co.uk/social-distancing-solution>.
- [30] Triax Technologies, [En línea]. Available: <https://www.triaxtec.com/resource/fact-sheet/proximity-trace/>.
- [31] LociLabs 2020, [En línea]. Available: <https://www.locilabs.co.uk/>.
- [32] RightCrowd - Everyone on Site is Right, [En línea]. Available: <https://www.rightcrowd.com/>.
- [33] VOS Systems LLC Partner Offices, [En línea]. Available: <https://www.alerttrace.com/>.

- [34] Losant IoT 2020, [En línea]. Available: <https://www.losant.com/contact-tracing-with-iot>.
- [35] Kinexon, [En línea]. Available: <https://kinexon.com/safezone>.
- [36] El Universo, «ASI Ecuador, app para notificar contacto con casos positivos de COVID-19, plan piloto comienza en Cuenca,» 3 Agosto 2020.
- [37] Corona Warn-App Open Source Project, [En línea]. Available: <https://www.coronawarn.app/en/>.
- [38] Aarogya Setu, [En línea]. Available: <https://aarogyasetu.gov.in/>.
- [39] Espressif IOT Team, «ESP8266EX Datasheet,» 2020.
- [40] NORDI Semiconductor ASA, «nRF52840 Product Specification v1.1,» 2019.
- [41] L. D. I. C. Rozadillas, «Evaluación de dispositivos UWB para localización en interiores con medidas de distancia». 2017.
- [42] S. D. Molinera, «Luis De la Calle Rozadillas». 2016.
- [43] H. Mari, «Simulation Models for the Performance». 2017.
- [44] Köpke, Andreas & Swigulski, M. & Wessel, K. & Willkomm, Daniel & Haneveld, P. & Parker, T. & Visser, Otto & Lichte, Hermann & Valentin, Stefan, «Simulating wireless and mobile networks in OMNeT++ the MiXiM vision,» *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, SimuTools*, 2008.
- [45] L. D. I. C. Rozadillas, «Evaluación de dispositivos UWB para localización en interiores con medidas de distancia». 2017.

# ÁPENDICES

## Ápndice # 1

En el siguiente apéndice se va a colocar el código fuente desarrollado en INET para la capa de enlace de datos y física del estándar 802.15.1.

### CAPA DE ENLACE DATOS

- Fuente del archivo **ieee802151BLEMac.ned**

```
package inet.linklayer.ieee802151;

import inet.linklayer.lmac.LMac;

module Ieee802151BLEMac extends LMac
{
    //en esta sección se pueden cambiar los parámetros de MAC TDMA
    //para LMAC en relación
    //a los parámetros de LMAC LAYER inet.linklayer.lmac.LMacLayer;
}
```

- Fuente del archivo **ieee802151BLENic.ned**

```
package inet.linklayer.ieee802151;

import inet.queueing.contract.IPacketQueue;
import inet.linklayer.acking.AckingMac;
import inet.linklayer.contract.IWirelessInterface;
import inet.networklayer.common.InterfaceEntry;
import inet.physicallayer.contract.packetlevel.IRadio;

module Ieee802151BLENic extends InterfaceEntry like
IWirelessInterface
{
    parameters:
        string interfaceTableModule;
        string radioType = default("Ieee802151BLERadio");
        *.interfaceTableModule =
default(absPath(interfaceTableModule));
        @display("i=block/ifcard");
    gates:
        input upperLayerIn;
        output upperLayerOut;
        input radioIn @labels(IRadioFrame);
    submodules:
        mac: Ieee802151BLEMac {
            parameters:
                @display("p=100,150");
        }
        radio: <radioType> like IRadio {
```

```

        parameters:
            @display("p=100,250");
    }
    connections:
        upperLayerIn --> mac.upperLayerIn;
        mac.lowerLayerOut --> radio.upperLayerIn;
        mac.upperLayerOut --> { @display("m=n"); } --> upperLayerOut;
        radioIn --> { @display("m=s"); } --> radio.radioIn;
        radio.upperLayerOut --> mac.lowerLayerIn;
    }

```

### CAPA FÍSICA

- Fuente del archivo ***ieee802151BLEReceiver.cc***

```

#include
"inet/physicalayer/ieee802151/packetlevel/Ieee802151BLEReceiver.h"

namespace inet {

namespace physicalayer {

Define_Module(Ieee802151BLEReceiver);

Ieee802151BLEReceiver::Ieee802151BLEReceiver() :
    FlatReceiverBase()
{
}

void Ieee802151BLEReceiver::initialize(int stage)
{
    FlatReceiverBase::initialize(stage);
    if (stage == INITSTAGE_LOCAL) {
        minInterferencePower =
mW(math::dBmW2mW(par("minInterferencePower")));
    }
}

std::ostream& Ieee802151BLEReceiver::printToStream(std::ostream&
stream, int level) const
{
    stream << "Ieee802151BLEReceiver";
    return FlatReceiverBase::printToStream(stream, level);
}

} // namespace physicalayer

} // namespace inet

```

- Fuente del archivo ***ieee802151BLEReceiver.h***

```

#ifndef __INET_IEEE802151BLERECEIVER_H

```

```

#define __INET_IEEE802151BLERECEIVER_H

#include "inet/physicallayer/base/packetlevel/FlatReceiverBase.h"

namespace inet {

namespace physicallayer {

class INET_API Ieee802151BLEReceiver : public FlatReceiverBase
{
protected:
    W minInterferencePower;

public:
    Ieee802151BLEReceiver();

    void initialize(int stage) override;

    virtual std::ostream& printToStream(std::ostream& stream, int
level) const override;

    virtual W getMinInterferencePower() const override { return
minInterferencePower; }
};

} // namespace physicallayer

} // namespace inet

#endif // ifndef __INET_IEEE802151BLERECEIVER_H

```

- Fuente del archivo ***Ieee802151BLETransmitter.cc***

```

#include
"inet/physicallayer/ieee802151/packetlevel/Ieee802151BLETransmitter.h"
#include
"inet/physicallayer/analogmodel/packetlevel/ScalarTransmission.h"
#include "inet/physicallayer/contract/packetlevel/RadioControlInfo_m.h"
#include "inet/mobility/contract/IMobility.h"

namespace inet {

namespace physicallayer {

Define_Module(Ieee802151BLETransmitter);

Ieee802151BLETransmitter::Ieee802151BLETransmitter() :
    FlatTransmitterBase()
{
}

```



```

std::ostream& Ieee802151BLETransmitter::printToStream(std::ostream&
stream, int level) const
{
    stream << "Ieee802151BLETransmitter";
    return FlatTransmitterBase::printToStream(stream, level);
}

const ITransmission *Ieee802151BLETransmitter::createTransmission(const
IRadio *transmitter, const Packet *packet, const simtime_t startTime)
const
{
    W transmissionPower = computeTransmissionPower(packet);
    bps transmissionBitrate = computeTransmissionDataBitrate(packet);
    const simtime_t headerDuration = b(headerLength).get() /
bps(transmissionBitrate).get();
    const simtime_t dataDuration = b(packet->getTotalLength()).get() /
bps(transmissionBitrate).get();
    const simtime_t duration = preambleDuration + headerDuration +
dataDuration;
    const simtime_t endTime = startTime + duration;
    IMobility *mobility = transmitter->getAntenna()->getMobility();
    const Coord startPosition = mobility->getCurrentPosition();
    const Coord endPosition = mobility->getCurrentPosition();
    const Quaternion startOrientation = mobility-
>getCurrentAngularPosition();
    const Quaternion endOrientation = mobility-
>getCurrentAngularPosition();
    return new ScalarTransmission(transmitter, packet, startTime,
endTime, preambleDuration, headerDuration, dataDuration, startPosition,
endPosition, startOrientation, endOrientation, modulation, headerLength,
packet->getTotalLength(), centerFrequency, bandwidth,
transmissionBitrate, transmissionPower);
}

} // namespace physicallayer

} // namespace inet

```

- Fuente del archivo ***ieee802151BLETransmitter.h***

```

#ifndef __INET_IEEE802151BLETRANSMITTER_H
#define __INET_IEEE802151BLETRANSMITTER_H

#include "inet/physicallayer/base/packetlevel/FlatTransmitterBase.h"

namespace inet {

namespace physicallayer {

class INET_API Ieee802151BLETransmitter : public FlatTransmitterBase
{
public:

```

```

Ieee802151BLETransmitter();

    virtual std::ostream& printToStream(std::ostream& stream, int level)
const override;
    virtual const ITransmission *createTransmission(const IRadio *radio,
const Packet *packet, const simtime_t startTime) const override;
};

} // namespace physallayer

} // namespace inet

#endif // ifndef __INET_IEEE802151BLETRANSMITTER_H

```

- Fuente del archivo *ieee802151BLERadio.ned*

```

package inet.physallayer.ieee802151.packetlevel;

import inet.physallayer.base.packetlevel.FlatRadioBase;

module Ieee802151BLERadio extends FlatRadioBase
{
    parameters:
        antennaType.typename = default("IsotropicAntenna");
        transmitterType.typename = default("Ieee802151BLETransmitter");
        receiverType.typename = default("Ieee802151BLEReceiver");

        centerFrequency = default(2540 MHz);

        // B_20dB ATmega256RFR2 (page 564)
        // *.bandwidth = default(2.8 MHz);
        // what is meant by bandwidth here?
        // the B_20dB bandwidth would lead to far too low BERs
        bandwidth = default(0.25 MHz);

        // 802.15.4-2006 (page 28)
        *.bitrate = default(250 kbps);

        // PHY Header, 802.15.4-2006 (page 43)
        // 4 octets Preamble
        // 1 octet SFD
        // 7 bit Frame Length
        // 1 bit Reserved
        *.headerBitLength = default((4*8 + 1*8 + 7 + 1) * 1 b);

        // RSSI sensitivity (ATmega256RFR2, page 566)
        receiver.energyDetection = default(-90dBm);

        // Receiver sensitivity (ATmega256RFR2, page 565)
        // TODO That is not quite true, because sensitivity
        //       is defined as the input signal power that yields
        //       a PER < 1% for a PSDU of 20 octets, but INET handles it

```

```

//      as minimum reception power.
receiver.sensitivity = default(-100dBm);

// There is no fixed boundary, because of the
// DSSS and the capture effect. Taking the sensitivity minus some
// arbitrary value as an approximate guess.
receiver.minInterferencePower = default(-120dBm);

// Minimum SNIR
// -8 dB results into 98% PER for a PSDU of 20 octets
receiver.snirThreshold = default(-8 dB);

// TX Output power (typ. 3.5 dBm, ATmega256RFR2, page 564)
transmitter.power = default(2.24mW);

@class(Radio);
}

```

- Fuente del archivo **ieee802151BLERadioMedium.ned**

```

package inet.physicallayer.ieee802151.packetlevel;

import inet.physicallayer.common.packetlevel.RadioMedium;

module Ieee802151BLERadioMedium extends RadioMedium
{
    parameters:
        propagationType.typename = default("ConstantSpeedPropagation");
        analogModelType.typename = default("ScalarAnalogModel");
        backgroundNoiseType.typename =
default("IsotropicScalarBackgroundNoise");

        // 802.15.4-2006, page 266
        pathLossType.typename = default("BreakpointPathLoss");
        pathLoss.breakpointDistance = 8 m;
        pathLoss.l01 = 40.2;
        pathLoss.alpha1 = 2;
        pathLoss.l02 = 58.5;
        pathLoss.alpha2 = 3.3;

        // Reflects the thermal noise for the receiver sensitivity
        // TODO couple with sensitivity
        backgroundNoise.power = default(-96.616dBm);
        backgroundNoise.dimensions = default("time");
}

```

- Fuente del archivo **ieee802151BLEReceiver.ned**

```

package inet.physicallayer.ieee802151.packetlevel;

import inet.physicallayer.base.packetlevel.NarrowbandReceiverBase;

```

```

module Ieee802151BLEReceiver extends NarrowbandReceiverBase
{
    parameters:
        modulation = default("DSSS-OQPSK-16");
        errorModelType.typename = default("APSKErrorModel");
        double minInterferencePower @unit(dBm);
        @class(Ieee802151BLEReceiver);
}

```

- Fuente del archivo ***ieee802151BLETransmitter.ned***

```

package inet.physicallayer.ieee802151.packetlevel;

import inet.physicallayer.base.packetlevel.NarrowbandTransmitterBase;

module Ieee802151BLETransmitter extends NarrowbandTransmitterBase
{
    parameters:
        modulation = default("DSSS-OQPSK-16");
        @class(Ieee802151BLETransmitter);
}

```

## Ápndice # 2

En el siguiente Ápndice se va a colocar el código fuente de la red creada para las diferentes tecnologías.

```
//Paquete contenedor del archivo NED
package inet.examples.aaSimuTec;
//Importación de Librerías y módulos
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.node.ethernet.Eth100M;
import inet.node.ethernet.EtherSwitch;
import inet.node.inet.Router;
import inet.node.inet.StandardHost;
import inet.node.inet.WirelessHost;
import inet.node.internetcloud.InternetCloud;
import inet.node.wireless.AccessPoint;
import
inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;
import inet.visualizer.integrated.IntegratedCanvasVisualizer;
import inet.common.misc.ThruputMeteringChannel;
//import inet.common.figures.DelegateSignalConfigurator;
import inet.node.inet.INetworkNode;
import inet.physicallayer.contract.packetlevel.IRadioMedium;
import inet.environment.common.PhysicalEnvironment;
import
inet.physicallayer.ieee802154.packetlevel.Ieee802154NarrowbandScalarRadioMedium;
//import inet.common.lifecycle.LifecycleController;
import inet.node.inet.SensorNode;

//Red de La Simulación A del Escenario 1
network redBle
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgb=10,0,greyscale;bg1=3;i=background
    /streetmap");
    @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de los submódulos de la red con su ubicación (x,y)
        y tipo de ícono
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=13.570001,8.28");
        }
        configurator: Ipv4NetworkConfigurator {
```

```

        parameters:
            assignDisjunctSubnetAddresses = false;
            @display("p=13.670657,14.842427;is=s");
    }
    radioMedium: <mediumType> like IRadioMedium {
        parameters:
            @display("p=13.614859,12.498886");
    }
    HostMovilA: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    physicalEnvironment: PhysicalEnvironment {
        @display("p=13.670657,4.8544784");
    }
    HostMovilB: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }
}

connections:
}

//Red de La Simulación A del Escenario 1
network redWifi
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgg=10,0,greys;bg1=3;i=background
    /streetmap");
    @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de Los submódulos de la red con su ubicación (x,y)
        y tipo de ícono
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=13.570001,8.28");
        }
        configurator: Ipv4NetworkConfigurator {
            parameters:
                assignDisjunctSubnetAddresses = false;
                @display("p=13.670657,14.842427;is=s");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=13.614859,12.498886");
        }
}

```

```

    }
    HostMovilA: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    physicalEnvironment: PhysicalEnvironment {
        @display("p=13.670657,4.8544784");
    }

    HostMovilB: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    connections:
}

network redUwbRfid
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgg=10,0,greys;bg1=3;i=background
    /streetmap");
        @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de los submódulos de la red con su ubicación (x,y)
        y tipo de ícono
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=13.570001,8.28");
        }
        configurator: Ipv4NetworkConfigurator {
            parameters:
                assignDisjunctSubnetAddresses = false;
                @display("p=13.670657,14.842427;is=s");
        }
        radioMedium: Ieee802154NarrowbandScalarRadioMedium {
            parameters:
                @display("p=13.614859,12.498886");
        }
        HostMovilA: SensorNode {
            parameters:
                @display("p=6,6;i=device/pocketpc");
        }
}

```

```

    physicalEnvironment: PhysicalEnvironment {
        @display("p=13.670657,4.8544784");
    }

    HostMovilB: SensorNode {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    connections:
}
//Paquete contenedor del archivo NED
package inet.examples.aaSimuTec;
//Importación de librerías y módulos
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.node.ethernet.Eth100M;
import inet.node.ethernet.EtherSwitch;
import inet.node.inet.Router;
import inet.node.inet.StandardHost;
import inet.node.inet.WirelessHost;
import inet.node.internetcloud.InternetCloud;
import inet.node.wireless.AccessPoint;
import
inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;
import inet.visualizer.integrated.IntegratedCanvasVisualizer;
import inet.common.misc.ThruputMeteringChannel;
//import inet.common.figures.DelegateSignalConfigurator;
import inet.node.inet.INetworkNode;
import inet.physicallayer.contract.packetlevel.IRadioMedium;
import inet.environment.common.PhysicalEnvironment;
import
inet.physicallayer.ieee802154.packetlevel.Ieee802154NarrowbandScalarRadioMedium;
//import inet.common.lifecycle.LifecycleController;
import inet.node.inet.SensorNode;

//Red de La Simulación A del Escenario 1
network redBle
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgg=10,0,grey;bgl=3;i=background
    /streetmap");
        @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de Los submódulos de la red con su ubicación (x,y)
        y tipo de ícono

```



```

visualizer: IntegratedCanvasVisualizer {
    parameters:
        @display("p=13.570001,8.28");
}
configurator: Ipv4NetworkConfigurator {
    parameters:
        assignDisjunctSubnetAddresses = false;
        @display("p=13.670657,14.842427;is=s");
}
radioMedium: <mediumType> like IRadioMedium {
    parameters:
        @display("p=13.614859,12.498886");
}
HostMovilA: WirelessHost {
    parameters:
        @display("p=6,6;i=device/pocketpc");
}

physicalEnvironment: PhysicalEnvironment {
    @display("p=13.670657,4.8544784");
}
HostMovilB: WirelessHost {
    parameters:
        @display("p=6,6;i=device/pocketpc");
}

connections:
}

//Red de La Simulación A del Escenario 1
network redWifi
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgg=10,0,greyscale;bg1=3;i=background
    /streetmap");
    @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de Los submódulos de La red con su ubicación (x,y)
        y tipo de ícono
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=13.570001,8.28");
        }
        configurator: Ipv4NetworkConfigurator {
            parameters:
                assignDisjunctSubnetAddresses = false;

```

```

        @display("p=13.670657,14.842427;is=s");
    }
    radioMedium: Ieee80211ScalarRadioMedium {
        parameters:
            @display("p=13.614859,12.498886");
    }
    HostMovilA: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    physicalEnvironment: PhysicalEnvironment {
        @display("p=13.670657,4.8544784");
    }

    HostMovilB: WirelessHost {
        parameters:
            @display("p=6,6;i=device/pocketpc");
    }

    connections:
}

network redUwbRfid
{
    //parámetros y señales a ser desplegadas en el ambiente de
    simulación
    parameters:
        string mediumType = default("IdealRadioMedium");

    @display("bgb=19.920101,18.134544;bgu=m;bgg=10,0,grey;bg1=3;i=background
    /streetmap");
    @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

    types:
    submodules:
        //inclusión de los submódulos de la red con su ubicación (x,y)
        y tipo de ícono
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=13.570001,8.28");
        }
        configurator: Ipv4NetworkConfigurator {
            parameters:
                assignDisjunctSubnetAddresses = false;
                @display("p=13.670657,14.842427;is=s");
        }
        radioMedium: Ieee802154NarrowbandScalarRadioMedium {
            parameters:
                @display("p=13.614859,12.498886");
        }
}

```

```
HostMovilA: SensorNode {  
  parameters:  
    @display("p=6,6;i=device/pocketpc");  
}  
  
physicalEnvironment: PhysicalEnvironment {  
  @display("p=13.670657,4.8544784");  
}  
  
HostMovilB: SensorNode {  
  parameters:  
    @display("p=6,6;i=device/pocketpc");  
}  
  
connections:  
}
```

### Ápndice # 3

En el siguiente apéndice se va a colocar el código fuente del archivo **omnetpp.ini** donde se detallan los parámetros de las diferentes tecnologías.

```
#declaración de nombre de la configuración
[Config BLE]
network = redBle
sim-time-limit = 10s

#####
####CONFIGURACIONES CAPA APLICACION###
#####
*.*.numApps = 1
*.Host*.app[0].typename = "UdpApp"
*.Host*.app[0].io.localPort = 1000
*.HostMovilA.app[0].io.destAddress = "HostMovilB"
*.HostMovilB.app[0].io.destAddress = "HostMovilA"
*.Host*.app[0].io.destPort = 1000
*.Host*.app[0].source.typename = "BurstyPacketProducer"
*.Host*.app[0].source.normalProducer.productionInterval = 1s
*.Host*.app[0].source.normalProducer.packetLength = 100B
*.Host*.app[0].source.burstProducer.productionInterval = 1s
*.Host*.app[0].source.burstProducer.packetLength = 1000B
*.Host*.app[0].source.scheduler.waitIntervals = "40 4"
*.Host*.app[0].source.scheduler.transitionProbabilities = "0 1 1 0"

#####
####CONFIGURACIONES DE MOVILIDAD###
#####
**.updateInterval = 1s
**.HostMovilB.mobility.typename = "LinearMobility"
**.HostMovilB.mobility.initFromDisplayString = false
**.HostMovilB.mobility.speed = 0.4mps
**.HostMovilB.mobility.angle = 0deg # degrees
**.constraintAreaMinX = 5.5m
**.constraintAreaMinY = 5.5m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 11.5m
**.constraintAreaMaxY = 9.5m
**.constraintAreaMaxZ = 0m

#####
####CONFIGURACIONES DE OBSTÁCULOS FÍSICOS Y ENTORNO###
#####
*.physicalEnvironment.config = xmldoc("paredes.xml")
*.radioMedium.obstacleLoss.typename = "DielectricObstacleLoss"
*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

#####
```

```

#####CONFIGURACIONES DE PÉRDIDAS Y MODELO DE PROPAGACIÓN###
#####
*.radioMedium.pathLoss.typename = "LogNormalShadowing"
**.propagation.typename = "ConstantSpeedPropagation" #propagación
constante
**.analogModel.typename = "ScalarAnalogModel" #calcula la potencia de la
señal recibida

#####
#####CONFIGURACIONES INTERFACES DE RED###
#####
*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "BMacInterface"
*.Host*.wlan[0].mac.typename = "Ieee802151BLEMac"
*.Host*.**.bitrate = 0.7Mbps

#####
#####CONFIGURACIONES RADIO###
#####
*.mediumType = "Ieee802151BLERadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 2.45GHz
*.Host*.wlan[0].radio.typename = "Ieee802151BLERadio"
*.Host*.wlan[0].radio.transmitter.typename = "Ieee802151BLETransmitter"
*.Host*.wlan[0].radio.receiver.typename = "Ieee802151BLEReceiver"
*.Host*.wlan[0].radio.centerFrequency = 2.45GHz
*.Host*.wlan[0].radio.bandwidth = 2MHz
*.Host*.wlan[0].radio.transmitter.power = 1mW
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"
*.Host*.wlan[0].radio.antenna.numAntennas = 1
*.Host*.wlan[0].radio.*.headerLength = 54b
*.Host*.wlan[0].radio.receiver.errorModel.typename = "ApskErrorModel"
*.Host*.wlan[0].radio.transmitter.modulation = "GFSK"
*.Host*.wlan[0].radio.receiver.modulation = "GFSK"

[Config Wifi]
network = redWifi
sim-time-limit = 10s

#####
#####CONFIGURACIONES CAPA APLICACION###
#####
*.*.numApps = 1
*.Host*.app[0].typename = "UdpApp"
*.Host*.app[0].io.localPort = 1000
*.HostMovilA.app[0].io.destAddress = "HostMovilB"
*.HostMovilB.app[0].io.destAddress = "HostMovilA"
*.Host*.app[0].io.destPort = 1000

```

```

*.Host*.app[0].source.typename = "BurstyPacketProducer"
*.Host*.app[0].source.normalProducer.productionInterval = 1s
*.Host*.app[0].source.normalProducer.packetLength = 100B
*.Host*.app[0].source.burstProducer.productionInterval = 1s
*.Host*.app[0].source.burstProducer.packetLength = 1000B
*.Host*.app[0].source.scheduler.waitIntervals = "40 4"
*.Host*.app[0].source.scheduler.transitionProbabilities = "0 1 1 0"

#####
####CONFIGURACIONES DE MOVILIDAD###
#####
**.updateInterval = 1s
**.HostMovilB.mobility.typename = "LinearMobility"
**.HostMovilB.mobility.initFromDisplayString = false
**.HostMovilB.mobility.speed = 0.4mps
**.HostMovilB.mobility.angle = 0deg # degrees
**.constraintAreaMinX = 5.5m
**.constraintAreaMinY = 5.5m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 9.5m
**.constraintAreaMaxY = 9.5m
**.constraintAreaMaxZ = 0m

#####
####CONFIGURACIONES DE OBSTÁCULOS FÍSICOS Y ENTORNO###
#####
*.physicalEnvironment.config = xmlDoc("paredes.xml")
*.radioMedium.obstacleLoss.typename = "DielectricObstacleLoss"
*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

#####
####CONFIGURACIONES DE PÉRDIDAS Y MODELO DE PROPAGACIÓN###
#####
*.radioMedium.pathLoss.typename = "LogNormalShadowing "
**.propagation.typename = "ConstantSpeedPropagation" #propagación
constante
**.analogModel.typename = "ScalarAnalogModel" #calcula la potencia de la
señal recibida

**.wlan*.bitrate = 11Mbps
**.wlan[*].radio.antenna.numAntennas = 8
**.wlan[*].radio.transmitter.power = 100mW
**.centerFrequency = 5GHz
**.wlan[*].radio.receiver.sensitivity = -85dBm
**.wlan[*].radio.receiver.snirThreshold = 4dB

[Config UWB]
network = redUwbRfid
sim-time-limit = 10s

```

```

#####
####CONFIGURACIONES CAPA APLICACION###
#####
*.*.numApps = 1
*.Host*.app[0].typename = "UdpApp"
*.Host*.app[0].io.localPort = 1000
*.HostMovilA.app[0].io.destAddress = "HostMovilB"
*.HostMovilB.app[0].io.destAddress = "HostMovilA"
*.Host*.app[0].io.destPort = 1000
*.Host*.app[0].source.typename = "BurstyPacketProducer"
*.Host*.app[0].source.normalProducer.productionInterval = 1s
*.Host*.app[0].source.normalProducer.packetLength = 100B
*.Host*.app[0].source.burstProducer.productionInterval = 1s
*.Host*.app[0].source.burstProducer.packetLength = 1000B
*.Host*.app[0].source.scheduler.waitIntervals = "40 4"
*.Host*.app[0].source.scheduler.transitionProbabilities = "0 1 1 0"

#####
####CONFIGURACIONES DE MOVILIDAD###
#####
**.updateInterval = 1s
**.HostMovilB.mobility.typename = "LinearMobility"
**.HostMovilB.mobility.initFromDisplayString = false
**.HostMovilB.mobility.speed = 0.4mps
**.HostMovilB.mobility.angle = 0deg # degrees
**.constraintAreaMinX = 5.5m
**.constraintAreaMinY = 5.5m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 9.5m
**.constraintAreaMaxY = 9.5m
**.constraintAreaMaxZ = 0m

#####
####CONFIGURACIONES DE OBSTÁCULOS FÍSICOS Y ENTORNO###
#####
*.physicalEnvironment.config = xmlDoc("paredes.xml")
*.radioMedium.obstacleLoss.typename = "DielectricObstacleLoss"
*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

#####
####CONFIGURACIONES DE PÉRDIDAS Y MODELO DE PROPAGACIÓN###
#####
*.radioMedium.pathLoss.typename = "LogNormalShadowing"
**.propagation.typename = "ConstantSpeedPropagation" #propagación
constante
**.analogModel.typename = "ScalarAnalogModel" #calcula la potencia de la
señal recibida

#####
####CONFIGURACIONES INTERFACES DE RED###
#####

```

```

*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "WirelessInterface"
*.Host*.wlan[0].mac.typename = "Ieee802154Mac"
*.Host*.**.bitrate = 6.8Mbps
*.Host*.wlan[*].mac.useMACAcks = false
*.Host*.wlan[*].mac.rxSetupTime = 0s
*.Host*.wlan[*].mac.headerLength = 16B

#####
####CONFIGURACIONES RADIO###
#####
*.mediumType = "Ieee802154NarrowbandScalarRadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 6.48GHz
*.Host*.wlan[0].radio.typename = "Ieee802154NarrowbandScalarRadio"
*.Host*.wlan[0].radio.transmitter.typename =
"Ieee802154NarrowbandScalarTransmitter"
*.Host*.wlan[0].radio.receiver.typename =
"Ieee802154NarrowbandScalarReceiver"
*.Host*.wlan[0].radio.centerFrequency = 6.48GHz
*.Host*.wlan[0].radio.bandwidth = 500MHz
*.Host*.wlan[0].radio.transmitter.power = 20uW
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"
*.Host*.wlan[0].radio.antenna.numAntennas = 1
*.Host*.wlan[0].radio.*.headerLength = 54b
*.Host*.wlan[0].radio.receiver.errorModel.typename = "ApskErrorModel"
*.Host*.wlan[0].radio.transmitter.modulation = "BPSK"
*.Host*.wlan[0].radio.receiver.modulation = "BPSK"

[Config RFID]
network = redUwbRfid
sim-time-limit = 10s

#####
####CONFIGURACIONES CAPA APLICACION###
#####
*.*.numApps = 1
*.Host*.app[0].typename = "UdpApp"
*.Host*.app[0].io.localPort = 1000
*.HostMovilA.app[0].io.destAddress = "HostMovilB"
*.HostMovilB.app[0].io.destAddress = "HostMovilA"
*.Host*.app[0].io.destPort = 1000
*.Host*.app[0].source.typename = "BurstyPacketProducer"
*.Host*.app[0].source.normalProducer.productionInterval = 1s
*.Host*.app[0].source.normalProducer.packetLength = 100B
*.Host*.app[0].source.burstProducer.productionInterval = 1s
*.Host*.app[0].source.burstProducer.packetLength = 1000B

```



```

*.Host*.app[0].source.scheduler.waitIntervals = "40 4"
*.Host*.app[0].source.scheduler.transitionProbabilities = "0 1 1 0"

#####
####CONFIGURACIONES DE MOVILIDAD###
#####
**.updateInterval = 1s
**.HostMovilB.mobility.typename = "LinearMobility"
**.HostMovilB.mobility.initFromDisplayString = false
**.HostMovilB.mobility.speed = 0.4mps
**.HostMovilB.mobility.angle = 0deg # degrees
**.constraintAreaMinX = 5.5m
**.constraintAreaMinY = 5.5m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 9.5m
**.constraintAreaMaxY = 9.5m
**.constraintAreaMaxZ = 0m

#####
####CONFIGURACIONES DE OBSTÁCULOS FÍSICOS Y ENTORNO###
#####
*.physicalEnvironment.config = xmldoc("paredes.xml")
*.radioMedium.obstacleLoss.typename = "DielectricObstacleLoss"
*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

#####
####CONFIGURACIONES DE PÉRDIDAS Y MODELO DE PROPAGACIÓN###
#####
*.radioMedium.pathLoss.typename = "LogNormalShadowing"
**.propagation.typename = "ConstantSpeedPropagation" #propagación
constante
**.analogModel.typename = "ScalarAnalogModel" #calcula la potencia de la
señal recibida

#####
####CONFIGURACIONES INTERFACES DE RED###
#####
*.Host*.numWlanInterfaces = 1
*.Host*.wlan[0].typename = "WirelessInterface"
*.Host*.wlan[0].mac.typename = "Ieee802154Mac"
*.Host**.bitrate = 640kbps
*.Host*.wlan[*].mac.useMACAcks = false
*.Host*.wlan[*].mac.rxSetupTime = 0s
*.Host*.wlan[*].mac.headerLength = 16B

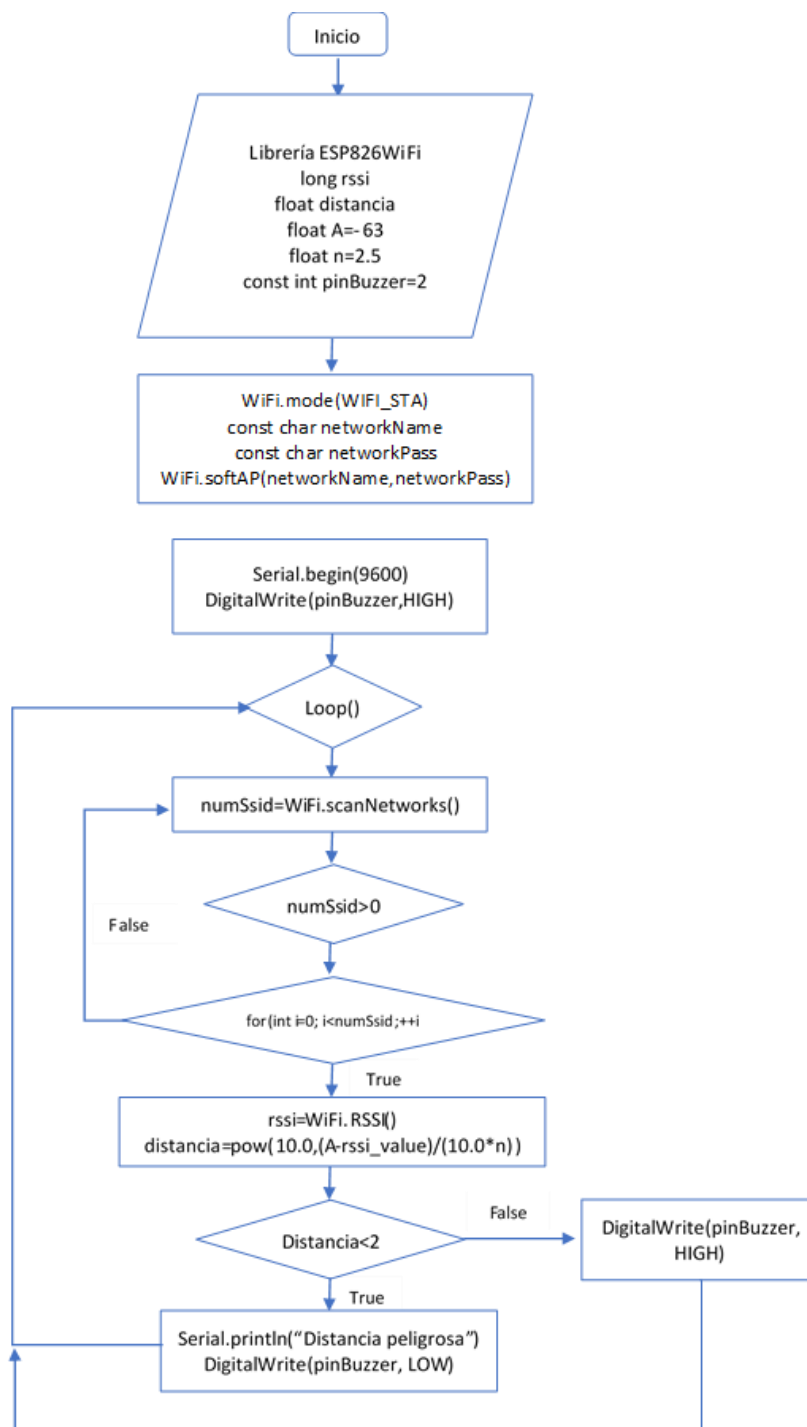
#####
####CONFIGURACIONES RADIO###
#####
*.mediumType = "Ieee802154NarrowbandScalarRadioMedium"
*.radioMedium.backgroundNoise.power = -90dBm
*.radioMedium.mediumLimitCache.centerFrequency = 865MHz

```

```
*.Host*.wlan[0].radio.typename = "Ieee802154NarrowbandScalarRadio"  
*.Host*.wlan[0].radio.transmitter.typename =  
"Ieee802154NarrowbandScalarTransmitter"  
*.Host*.wlan[0].radio.receiver.typename =  
"Ieee802154NarrowbandScalarReceiver"  
*.Host*.wlan[0].radio.centerFrequency = 865MHz  
*.Host*.wlan[0].radio.bandwidth = 500kHz  
*.Host*.wlan[0].radio.transmitter.power = 1.99W  
*.Host*.wlan[0].radio.transmitter.preambleDuration = 10us  
*.Host*.wlan[0].radio.receiver.sensitivity = -87dBm  
*.Host*.wlan[0].radio.receiver.energyDetection = -87dBm  
*.Host*.wlan[0].radio.receiver.snirThreshold = -8dB  
*.Host*.wlan[0].radio.antenna.typename = "IsotropicAntenna"  
*.Host*.wlan[0].radio.antenna.numAntennas = 1  
*.Host*.wlan[0].radio.*.headerLength = 54b  
*.Host*.wlan[0].radio.receiver.errorModel.typename = "ApskErrorModel"  
*.Host*.wlan[0].radio.transmitter.modulation = "BPSK"  
*.Host*.wlan[0].radio.receiver.modulation = "BPSK"
```

## Ápndice # 4

- Algoritmo utilizado en el código C++ cargado en el ESP-01



- Algoritmo utilizado en el código C++ cargado en el Arduino BLE

