



**Universidad de las Ciencias Informáticas**



## **Módulos para la simulación de modelos de propagación en la herramienta Andrómeda**

**Autores:** Lianet Cylwik López

Ariel Llanes Díaz

**Tutor:** MSc. Ismael Armando Nodarse Mora

**Co-tutor:** Ing. Yoennis Garrido Vargas

La Habana, junio de 2014

Declaramos que somos los únicos autores del trabajo "Módulos para la simulación de modelos de propagación en la herramienta Andrómeda" y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Autores:**

**Lianet Cylwik López**

---

**Ariel Llanes Díaz**

---

**Tutores:**

**MSc. Ismael Armando Nodarse Mora**

---

**Co-tutor:**

**Ing. Yoennis Garrido Vargas**

---

*"Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad."*

*Albert Einstein*

*A esas personas importantes en nuestras vidas, que siempre estuvieron listas para brindarnos toda su ayuda, ahora nos toca regresar un poquito de todo lo inmenso que nos han otorgado. Con todo nuestro cariño esta tesis se las dedicamos a ustedes:*

*Nuestros padres, familia y amigos.*

## Agradecimientos

### *Lianet*

- A mi mamá y mi abuela Loida por ser las dos personas más importantes en mi vida, por ser mi apoyo, por tantos años de sacrificio y esfuerzo, por haber dado lo mejor de sí. A ellas les debo quien soy, este trabajo también es de ustedes.
- A mi papá y mi hermana Katia, que aunque vivamos lejos siempre han estado pendiente de mí.
- A mi abuela Hortensia que sé que si pudiera entender que hoy su nieta hembra mayor se convierte en ingeniera estaría muy orgullosa.
- A mi prima Ismaray porque siempre vamos a estar ahí una para la otra.
- A mi tío que es como mi segundo papá porque me enseñó muchísimas cosas de la vida.
- A mi familia en general por su preocupación y comprensión.
- A mis amigos de siempre, especialmente a Dannelis, por todo ese calor de amistad, por aceptarme tal y como soy, a Amet por ayudarme y apoyarme en toda mis locuras en estos últimos años, a Wendy y Ariel por haber sido mis dos primeros amigos en la universidad, por haber compartido tantas cosas juntos, ustedes forman parte inseparable de mis recuerdos.
- A todos los amigos que hice en el nuevo grupo que tuve desde 3ro: Nunca los olvidaré.
- A mi tutor Ismael por tanta preocupación, por ayudarnos y guiarnos siempre, por ser único.
- A los profesores que pusieron su grano de arena para que sea quien soy hoy.
- A todas las personas que de una forma u otra contribuyeron a que hiciera realidad este gran sueño.

## Agradecimientos

### *Ariel*

- A mi mamá por ser tan especial en mi vida, por darme tanto amor y cariño y por ser mi inspiración, gracias por tu ahínco y sacrificio para que yo sea feliz, gracias por ser la mejor del mundo.
- A mi papá, gracias por haberme enseñado a luchar contra todos los obstáculos sin importar lo difícil que fuesen, gracias por ser mi ejemplo a seguir, a ti te debo mi fuerza de voluntad y mis ganas de seguir mejorando cada día más, gracias por luchar por mí, gracias por haber sido sin duda un padre del que siempre me sentiré orgulloso y ojalá donde quiera que estés, también de mí te sientas orgulloso. A ustedes mis padres, va dedicado el resultado de esta tesis.
- A mi hermano, por todo su apoyo y abnegación para que yo llegara hasta aquí, gracias por tu esfuerzo, gracias por ser también un ejemplo en mi vida.
- A toda mi familia por su dedicación y ayuda para poder tener este resultado, muchas gracias por estar siempre a mi lado.
- A mi novia, a esa que tanto amor incondicional me ha brindado durante estos años de estudio, gracias por ser como eres, por preocuparte más por mí que por ti misma, gracias por tu comprensión y por aguantar todos mis momentos de mal humor.
- A mis amigos del barrio, en especial Gaby y Alito, que más que amigos son mis hermanos, gracias por estar siempre ahí para mí.
- A mi compañera de tesis, por su preocupación y responsabilidad, gracias también por su amistad.
- A mi tutor Ismael por ser también nuestro amigo y por tantas horas dedicadas a que saliera bien esta tesis.
- A todos los eternos amigos que hice en esta escuela, que sin duda ocupan un lugar especial en mi vida, gracias Wendy, Odenys, Marlonete, Orelvolo, a todos los que me acompañaron en el grupo 4, a mis compañeros de apartamento, a los del grupo 1, en fin, a todo mi piquete, muchas gracias.

## **Resumen**

Con el incremento tecnológico de la actualidad el desarrollo de la tecnología inalámbrica ha alcanzado un auge significativo, llevando sus aplicaciones incluso a la automatización de edificios. El diseño del despliegue de la red de sensores inalámbricos en una edificación es una compleja tarea que necesita de la simulación del comportamiento de la señal de radio frecuencia dentro de edificaciones. El presente trabajo consiste en la implementación de modelos de propagación que permitan determinar con precisión la conectividad entre dispositivos inalámbricos, de forma tal que se puedan posicionar correctamente en el interior de un edificio. El desarrollo de la propuesta fue guiado por la metodología XP, UML como lenguaje de modelado y Java para la implementación, se utilizó el NetBeans como Entorno de Desarrollo Integrado y Visual Paradigm for UML como herramienta CASE. En su primera versión, la herramienta es capaz de calcular la propagación de ondas utilizando el modelo Multi-Wall y el modelo de Trayecto Dominante. La investigación finaliza con las pruebas unitarias y de aceptación del software, las cuales demostraron la correcta implementación de todas sus funcionalidades.

**Palabras clave:** modelos de propagación, Multi-Wall, Redes Inalámbricas de Sensores y Actuadores, simulación, trayecto dominante.

## Índice

Introducción .....	2
1. Fundamentación Teórica.....	8
1.1 Modelos de propagación empíricos o estadísticos.....	8
1.1.1 Modelo de Propagación en el Espacio Libre.....	9
1.1.2 Modelo <i>Log – Normal Shadowing Path Loss</i> .....	10
1.1.3 Modelo de Pérdida de Trayecto basado en COST 231 .....	11
1.1.4 Modelo basado en el número de muros y suelos.....	12
1.1.5 Modelo ITU-R.....	13
1.1.6 Modelo de Atenuación de Trayecto Linear.....	13
1.1.7 Modelo de Propagación de Doble Pendiente .....	14
1.1.8 Modelo Keenan-Motley .....	15
1.1.9 Modelo Multi-Wall.....	15
1.2 Modelos de propagación teóricos o deterministas. ....	16
1.2.1 Modelos basados en Técnicas de Trazado de Rayos.....	17
1.2.2 Modelo Diferencias Finitas en el Dominio del Tiempo. ....	17
1.2.3 Modelos basados en el método de los momentos.....	18
1.2.4 Modelo de Trayecto Dominante.....	19
1.3 Modelos de propagación seleccionados.....	21
1.4 Herramientas que realizan simulaciones mediante modelos de propagación.....	22
1.4.1 ProMan - WinProp Software Suite.....	22
1.4.2 WISAN Positioning Tool.....	23
1.4.3 Otras herramientas .....	23
1.5 Lenguaje de programación .....	24
1.6 Entorno de desarrollo.....	25
1.7 Librería 3D para el desarrollo .....	25
1.8 Librería para cargar ficheros IFC .....	26
1.9 Metodología de desarrollo de software .....	26
1.10 Herramienta para el modelado .....	27
1.11 Conclusiones parciales .....	28
2. Descripción de la solución propuesta.....	30
2.1 Usuarios del sistema.....	30
2.2 Lista de Reservas del Producto .....	30
2.3 Aspectos no funcionales del sistema .....	31
2.4 Exploración.....	32
2.4.1 Historias de Usuario.....	33
2.5 Planificación .....	35
2.5.1 Iteraciones.....	35
2.5.2 Plan de Entregas.....	36
2.6 Diseño .....	37
2.6.1 Patrones de diseño .....	37
2.6.2 Modelo de datos.....	37
2.6.3 Diagrama de clases .....	39
2.7 Conclusiones parciales.....	39
3 Implementación y pruebas.....	41
3.1 Descripción de la arquitectura.....	41
3.1.1 Plataforma Andr6meda.....	41
3.1.2 M6dulos del sistema .....	42
3.1.3 Paquetes del sistema .....	42
3.1.4 Diagrama de componentes.....	43
3.2 Implementaci6n.....	45
3.2.1 Primera iteraci6n .....	45
3.2.2 Segunda iteraci6n .....	48
3.3 C6lculo de la p6rdida de trayectoria .....	49



## Módulos para la simulación de modelos de propagación en la herramienta Andrómeda

3.3.1	Propagación en el espacio libre .....	50
3.3.2	Detección de colisiones.....	50
3.3.3	Fenómenos físicos y cálculo de los coeficientes.....	53
3.4	Pruebas.....	58
3.4.1	Pruebas experimentales .....	59
3.4.2	Pruebas de caja negra para la versión 1.0.....	60
3.4.3	Pruebas de caja blanca para la versión 1.0.....	60
3.4.4	Pruebas de caja blanca para la versión 2.0.....	61
3.4.5	Resultados generales de las pruebas.....	61
3.5	Conclusiones parciales.....	61
	Conclusiones.....	64
	Recomendaciones.....	65
	Referencias Bibliográficas.....	66
	Glosario.....	68





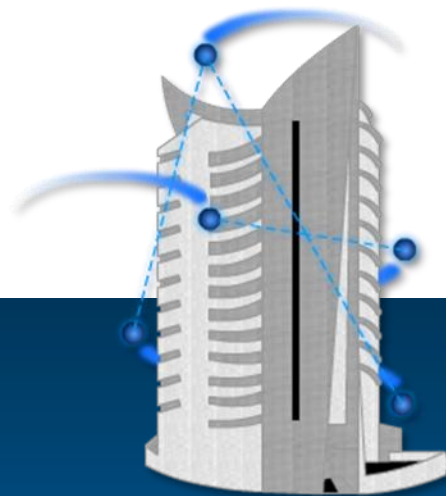
## Índice de figuras

Figura 1. Apertura recibida en una antena (7) .....	9
Figura 2. Pérdidas de trayecto para diferentes valores de $n$ (7) .....	11
Figura 3. Varios modos de propagación (11) .....	21
Figura 4. a) Modelo Multi-Wall, b) Modelo Trayecto Dominante .....	21
Figura 5. Ejemplo de predicción para un entorno interior .....	23
Figura 6. WISAN positioning Tool .....	23
Figura 7. Modelo de Datos .....	38
Figura 8. Diagrama de clases.....	39
Figura 9. Diagrama de paquetes .....	43
Figura 10. Diagrama de componentes correspondiente de los módulo IndoorModel, MultiWallModel y DominantPathModel. ....	44
Figura 11. Representación de una recta. (34) .....	50
Figura 12. Representación de un plano. (34) .....	51
Figura 13. Ejemplo de 3 medios diferentes. $n_1$ y $n_3$ se definen como medios infinitamente anchos.....	54
Figura 14. Efecto de guiado de onda .....	57
Figura 15. Determinación del efecto de guiado de onda. ....	57
Figura 16. Resultado de las pruebas. ....	61



## Índice de tablas

Tabla 1. Variable de pérdida de trayecto (7) .....	11
Tabla 2. Valores medios de los factores de pérdida. ....	12
Tabla 3. Usuarios del sistema.....	30
Tabla 4. HU 1: Visualizar los modelos de propagación implementados en la herramienta Andrómeda. ....	33
Tabla 5. HU 2: Permitir la selección del modelo de propagación a usarse en la simulación. ....	33
Tabla 6. HU 3: Calcular conectividad entre dispositivos inalámbricos usando Multi-Wall. ....	34
Tabla 7. HU 3: Calcular conectividad entre dispositivos inalámbricos usando el modelo Trayecto Dominante. .	34
Tabla 8. Plan de Iteraciones.....	36
Tabla 9. Plan de Entrega.....	37
Tabla 10. Tarea 1: Estudiar el concepto de <i>Lookup</i> en la guía definitiva de la Plataforma de Netbeans 7.....	45
Tabla 11. Tarea 2: Estudio de la arquitectura de la herramienta Andrómeda. ....	46
Tabla 12. Tarea 3: Implementar la clase <i>PropagationModelView</i> .....	46
Tabla 13. Tarea 4: Diseñar Interfaz de selección de los modelos. ....	46
Tabla 14. Tarea 5: Implementar la clase <i>ChoosePropagationModel</i> . ....	47
Tabla 15. Tarea 6: Estudiar la librería OpenIFCtool. ....	47
Tabla 16. Tarea 7: Estudiar la librería Java 3D. ....	47
Tabla 17. Tarea 8: Implementar las clases <i>Plane</i> , <i>Ray</i> , <i>MultiWallModel</i> , <i>SolvingT</i> .....	48
Tabla 18. Tarea 9: Implementar la detección de colisiones. ....	48
Tabla 19. CP Visualizar los modelos de propagación implementados en la herramienta. ....	60
Tabla 20. CP Permitir la selección del modelo de propagación a usarse en la simulación.....	60



Andr<sup>ó</sup>meda  
Andr<sup>ó</sup>meda

# Introducción

## Introducción

En los últimos años las Redes Inalámbricas de Sensores se han convertido en un campo de estudio que se encuentra en continuo crecimiento. Consisten en un conjunto de nodos de pequeño tamaño, de muy bajo consumo, que se comunican entre sí inalámbricamente formando una red y que a su vez se conectan con un sistema central siguiendo diferentes topologías. (1)

La evolución de redes de sensores tiene su origen en iniciativas militares. La investigación comenzó cerca de 1980 con el proyecto *Distributed Sensor Networks* de la Agencia Militar de Investigación Avanzada de Estados Unidos. Las investigaciones fuera del ámbito militar datan de finales de los 90, con el proyecto *Smartdust*. Uno de los objetivos de este proyecto fue crear sensores con comunicación en un milímetro cúbico. Aunque este proyecto finalizó de forma anticipada, dio lugar a muchos otros proyectos de investigación. Los investigadores participantes en estos estudios, apodaron con el término “mote” (mota) a los nodos sensores. (2)

Con la evolución de esta tecnología surgen los nodos actuadores. Mientras los sensores van reuniendo información sobre el medio físico, los actuadores toman decisiones y ejecutan las acciones apropiadas sobre el entorno. Estas redes son conocidas como Redes Inalámbricas de Sensores y Actuadores (WSAN, según sus siglas en inglés). (3)

La gama de aplicaciones y servicios ofrecidos por este tipo de sistemas son muy amplias: seguridad, control de acceso, gestión de la energía (iluminación, climatización, aguas y riegos), monitorización y control (avisos y alarmas, monitorización de la salud, automatización de tareas domésticas), ocio y entretenimiento (equipos multimedia, cultura), comunicación con sistemas y servidores externos, operación y mantenimiento de las propias instalaciones. (3)

Otra importante aplicación de las WSAN, es su utilización en los Sistemas de Control de Edificios (SCE) debido a las ventajas que brinda, dentro de las más significativas se encuentran: (4)

- Ahorro energético y supervisión del consumo eléctrico.
- Gestión del personal del edificio.
- Supervisión en tiempo real de eventos.
- Aviso de averías y mantenimiento preventivo.
- Telegestión remota.



- Aumento del confort de los usuarios y est6tica.
- Detecci3n y gesti3n eficaz de la seguridad en el complejo.

Existen otros factores que han acelerado la utilizaci3n de redes inal6mbricas en los SCE, por ejemplo su empleo puede significar una reducci3n de los costos si se tiene en cuenta el cableado requerido de muchos sistemas de control actuales. Adem6s de la reducci3n de los costos por cableado, la flexibilidad y versatilidad que las caracteriza, representa tambi6n una disminuci3n en los costos de despliegue y mantenimiento. (5)

Lamentablemente, el uso de WSAN en algunos tipos de escenarios puede conllevar al surgimiento de complejos problemas, ejemplo de ello es el despliegue de estas redes dentro de una edificaci3n. El campo el6ctrico dentro de un edificio est6 integrado por un amplio n6mero de componentes indirectos, factores considerables que no se tienen en cuenta en el caso del dise1o de redes para entornos abiertos. Por lo que en entornos interiores los niveles de se1al de Radio Frecuencia (RF) fluct6an en mayor medida que en entornos exteriores; por consiguiente, la cobertura inal6mbrica se caracteriza por ser compleja y muchas veces puede resultar incontrolable en el interior de una edificaci3n.

En la actualidad, existen varias investigaciones que enfrentan el problema antes mencionado. Algunas soluciones se basan en el empleo de herramientas software que faciliten, de forma efectiva, el despliegue de sensores en el interior de edificaciones. Sin embargo, a pesar de que estas propuestas ofrecen relevantes soluciones en este 6mbito de exploraci3n, a6n existen enfoques que no han sido abordados y deficiencias sin resolverse, los cuales representan campos de investigaci3n abiertos a nuevas innovaciones.

Las principales deficiencias de las herramientas actuales son:

- Incorrecta selecci3n de los modelos de propagaci3n de la se1al de RF.
- En las simulaciones no se tiene en cuenta el tipo de material y los obst6culos presentes en las edificaciones.
- Solo tienen en cuenta la cobertura y la conectividad interna de la red a la hora de generar las propuestas de despliegue.
- Las topolog6as de red obtenidas no satisfacen los dise1os requeridos por los SCE actuales.

Como consecuencias de las deficiencias anteriores y otros problemas identificados por (5) en las herramientas actuales, fue necesario crear una herramienta software que genere de forma autom6tica el n6mero y la posici3n de los dispositivos de una WSAN,

que permita la obtención de topologías de red de tipo estrella, árbol y malla que satisfagan el diseño de la red. Dicha herramienta lleva por nombre Andrómeda y está siendo desarrollada por el grupo de investigación del mismo nombre perteneciente a la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI).

Para el desarrollo de Andrómeda, se identificaron funcionalidades importantes que la misma debe incluir para lograr la simulación de la propagación de la señal de RF. Entre las cuales se encuentra: la implementación de métodos numéricos capaces de determinar con precisión la propagación de ondas electromagnéticas en ambientes confinados, donde la onda transmitida encuentra obstáculos en su camino propios de ambientes interiores, antes de llegar al receptor y que pueden producir la pérdida de la señal.

Esto requiere de la valoración de múltiples fenómenos que son dependientes tanto de los materiales como de su configuración geométrica. Como se explicaba anteriormente, la propagación de la señal de RF en entornos interiores se diferencia de la comunicación de radio en exteriores en que la cobertura es mucho menor. Esto se debe a la influencia que tienen las paredes y muros, las puertas y ventanas, los materiales de construcción usados, así como el tipo y el diseño arquitectónico de la edificación.

La propagación de la señal no se comportará igual en un edificio antiguo construido básicamente de ladrillo y madera a un edificio moderno construido de concreto y cristales, con varias oficinas y objetos, es decir, compuesto de gran cantidad de elementos radiantes y radioeléctricamente interferentes. Todos los objetos (mesas, sillas, etc.) dentro de una edificación, causan atenuación en la propagación de la onda electromagnética, además de cambios en su dirección.

Para simular el comportamiento de la señal de RF dentro de edificaciones, resulta importante la utilización de modelos de propagación en interiores. Un modelo de propagación es un conjunto de expresiones matemáticas, diagramas y algoritmos utilizados para representar las características de radio de un ambiente dado (7). La literatura propone diversas variantes de clasificación de estos modelos, de forma general, se pueden clasificar en empíricos o estadísticos y teóricos o deterministas. Los primeros se basan en la extrapolación estadística de resultados a partir de mediciones realizadas sobre el terreno, mientras que los segundos se apoyan en principios físicos en cuanto a la propagación de ondas de radio y fenómenos paralelos.

Producto de lo antes expresado, se plantea el siguiente **problema a resolver**:

¿Cómo simular la propagación de la señal de radio frecuencia para predecir conectividad de dispositivos inalámbricos en la herramienta Andrómeda?

El **objeto de estudio** de la investigación se centra en los modelos de propagación para entornos interiores. De aquí se deriva que el **campo de acción** sea específicamente la simulación de modelos de propagación en la herramienta Andrómeda.

Con el propósito de encontrar una solución al problema planteado se define como **objetivo general**: Desarrollar módulos para la simulación de modelos de propagación en la herramienta Andrómeda.

Teniendo para ello los siguientes **objetivos específicos**:

1. Caracterizar los modelos de propagación para entornos interiores.
2. Diseñar una propuesta para la simulación de modelos de propagación en la herramienta Andrómeda.
3. Implementar módulos para la simulación de modelos de propagación en la herramienta Andrómeda.
4. Validar la solución propuesta.

Para dar cumplimiento a los objetivos específicos y al objetivo general se han propuesto las siguientes tareas de investigación:

1. Análisis del estado del arte de modelos de propagación para entornos interiores, así como las tecnologías a ser utilizadas.
2. Selección de los modelos de propagación a ser implementados.
3. Levantamiento de aspectos funcionales y no funcionales.
4. Realización de las actividades y artefactos que tributen al análisis y diseño de la propuesta de solución, de acuerdo a la metodología de desarrollo utilizada.
5. Estudio de la arquitectura de la herramienta Andrómeda.
6. Realización de las actividades y artefactos que tributen a la implementación de la propuesta de solución, de acuerdo a la metodología de desarrollo utilizada.
7. Implementación de cada modelo de propagación seleccionado.
8. Realización de experimentos que validen los modelos de propagación implementados, así como el ajuste de su capacidad predictiva.
9. Realización de pruebas unitarias y de aceptación.

Finalmente, para guiar el desarrollo de la solución se plantea la siguiente **idea a defender**: si se desarrollan modelos de propagación para entornos interiores, que

permitan simular el comportamiento de la señal de RF, entonces, se podrá predecir la conectividad de dispositivos inalámbricos en la herramienta Andrómeda.

### **Métodos teóricos:**

El método **Analítico-Sintético** se utilizó para estudiar y revisar la documentación referente a los modelos de propagación existentes para entornos interiores, así como para analizar la bibliografía sobre los mecanismos de difracción, reflexión y penetración a través de diversos materiales; obteniendo los elementos necesarios para elaborar conclusiones que brinden solución a la necesidad de la investigación.

El método **Modelación** se utilizó para crear abstracciones de la estructura de los módulos implementados, con vista a explicar la misma a partir de diagramas y así lograr un mayor entendimiento de los requisitos que debe cumplir cada módulo.

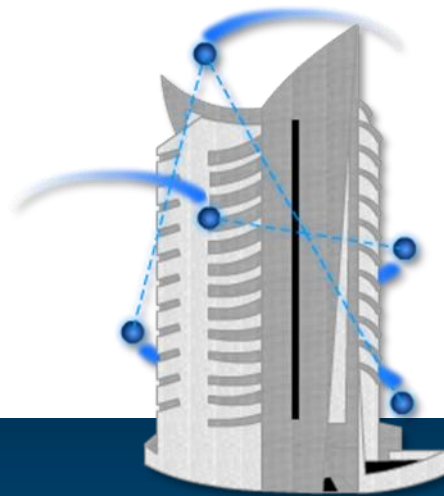
El documento está organizado en 3 capítulos, de la siguiente forma:

**Capítulo 1: Fundamentación Teórica.** Este capítulo aborda algunos elementos teóricos necesarios para la realización del trabajo, tales como: los principales conceptos relacionados con el dominio, estado del arte, así como la descripción y selección de las herramientas y metodología a utilizar.

**Capítulo 2: Descripción de la solución propuesta.** En este capítulo se hace una descripción del entorno, se define la Lista de Reservas del Producto, se conforman las Historias de Usuario, las cuales se organizan en iteraciones según la prioridad definida por el cliente, se estima la velocidad del proyecto y se acuerda el Plan de Entrega.

**Capítulo 3: Implementación y pruebas.** En este capítulo se describe la arquitectura del sistema, se detallan los diagramas UML utilizados para el desarrollo de la solución, las Tareas de Ingeniería y el resultado de los Casos de Pruebas Unitarias y de Aceptación, si como también las experimentales.





Andróneda  
Andróneda

# Capítulo 1

fundamentación Teórica



## 1. Fundamentación Teórica

Con el desarrollo vertiginoso de las tecnologías y sobre todo en el campo de las telecomunicaciones, la conexión de diferentes tipos de dispositivos se realiza de manera inalámbrica, incluso en áreas críticas como lo pueden ser los SCE. Por lo que como mismo se hace imprescindible el buen funcionamiento de los enrutadores, nodos o coordinadores dentro de una red, garantizar la conectividad entre ellos es de gran importancia. Para conocer que tan fuerte es la comunicación o si simplemente existe comunicación entre dispositivos se utilizan los modelos de propagación. En el desarrollo del siguiente capítulo se abordarán las clasificaciones de estos modelos así como algunas características de los mismos y herramientas que los implementan. Se explican detalladamente los aspectos a tener en cuenta para el desarrollo de la solución propuesta, la selección de las herramientas, librerías y metodología a utilizar.

### Modelos de propagación para entornos interiores

Los modelos de propagación permiten predecir el comportamiento de las señales electromagnéticas, en un ambiente determinado, lo cual es indispensable en procesos de planificación de redes inalámbricas y en general en cualquier tipo de sistema de comunicación inalámbrico.

Dependiendo del tipo de sistema de comunicación que se desee estudiar, algunos parámetros tales como: pérdidas de trayecto, longitud del trayecto, lento desvanecimiento, penetración en muros, entre otros; toman mayor o menor importancia, por lo que diversos modelos han sido desarrollados a fin de estimar con mayor facilidad los parámetros pertinentes, en un estudio dado.

Las influencias propias del entorno son tenidas en cuenta de manera implícita en su conjunto, sin ser reconocidas cada una de ellas de manera aislada, siendo esta la principal ventaja de estos modelos. Por el contrario, su precisión depende no solo de la precisión de las medidas sino de la similitud entre el entorno donde fueron llevadas a cabo las medidas y el entorno a analizar. (6) Los mismos se clasifican en empírico o estadísticos y en teóricos o deterministas.

#### 1.1 Modelos de propagación empíricos o estadísticos.

En los últimos años se han desarrollado diferentes modelos de predicción de propagación empíricos para ambientes interiores, en frecuencias que van de los 500Mhz a los 5Ghz.

Cada uno de estos modelos tiene sus ventajas y sus desventajas que ser6n explicadas a continuaci6n.

### 1.1.1 Modelo de Propagaci6n en el Espacio Libre

En espacio libre, la energ6a radiada por una antena omnidireccional se propaga por la superficie de una esfera (Figura 1). El 6rea de la superficie de una esfera de radio  $d$  es  $(4\pi)^2 d^2$ .

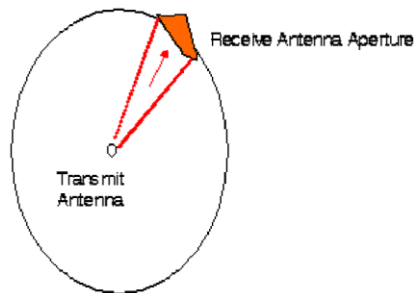


Figura 1. Apertura recibida en una antena (7)

El Modelo de Propagaci6n en Espacio Libre se utiliza para predecir la potencia de la se6al cuando entre el transmisor y el receptor existe una clara l6nea de vista (*line of sight*). Los sistemas de comunicaci6n por sat6lite y los enlaces microondas se pueden modelar como propagaci6n en el espacio libre.

Como la mayor6a de los modelos de propagaci6n a gran escala, este modelo predice que la potencia recibida decae como funci6n de la distancia de separaci6n entre el transmisor y receptor elevada a alguna potencia. La potencia recibida en el espacio libre por una antena receptora, la cual est6 separada de la antena transmisora a una distancia  $d$ , est6 dada por la ecuaci6n de Friis:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

Ecuaci6n 1.1 (7)

Donde:

- $P_r(d)$ : Potencia recibida.
- $P_t$  : Potencia transmitida.

- $G_t$  : Ganancia de antena transmisora.
- $G_r$  : Ganancia de la antena receptora.
- $\lambda$ : Longitud de onda en metros.
- $d$  : Distancia de separaci3n en metros.
- $L$ : P6rdidas del sistema no achacables a la propagaci3n.

### 1.1.2 Modelo Log – Normal Shadowing Path Loss

La mayor parte de los modelos emp6ricos se basan en este modelo seg6n el cual, las p6rdidas en entornos cerrados se pueden expresar por una ecuaci3n en funci3n de la distancia entre transmisor y receptor como:

$$PL(d) = PL(d_0) + 10n \log_{10} \left( \frac{d}{d_0} \right) + X_\sigma$$

**Ecuaci3n 1.2 (7)**

Donde:

- $PL(d_0)$ : P6rdida a distancia cercana de referencia.
- $n$ : Variable de p6rdida de trayecto.
- $d$  : Distancia de separaci3n en metros.
- $X_\sigma$ : Desviaci3n t6pica.

“ $n$ ” es una variable de p6rdida de trayecto que depende del tipo de entorno. La Tabla 1 muestra los valores t6picos de  $n$  para diferentes entornos.  $X_\sigma$  es una variable aleatoria expresada en dB (decibeles) que indica una desviaci3n t6pica de  $\sigma$ .  $PL(d_0)$  se calcula utilizando el Modelo de Propagaci3n en Espacios Libre o realizando medidas de campo.

La Figura 2 muestra las curvas de p6rdida de trayecto para diferentes valores de la variable de p6rdida de trayecto.

Tabla 1. Variable de pérdida de trayecto (7)

Entorno	$n$
Edificios (condiciones de visión directa)	1,6 a 2
Edificios (sin visión directa)	2 a 4
Edificios (sin visión directa, separación de 1 a 3 pisos)	4 a 6

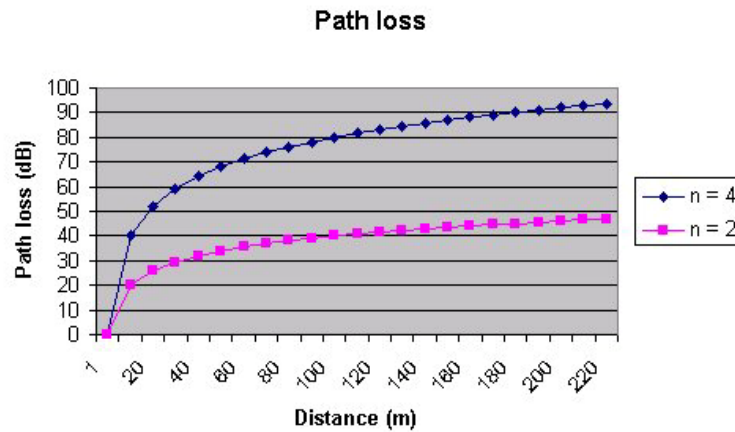


Figura 2. Pérdidas de trayecto para diferentes valores de  $n$  (7)

Este modelo es una técnica de estimación de pérdidas de propagación muy práctica y presenta como principal ventaja que tiene en cuenta todos los factores que influyen en la propagación.

### 1.1.3 Modelo de Pérdida de Trayecto basado en COST 231

Este es el modelo de propagación utilizado en el Sistema Universal de Telecomunicaciones Móviles (UMTS, según sus siglas en inglés). Se ha recogido del ETSI TR-101-112 ch. B.1.8, la cual es una especificación de los procedimientos seleccionados para la elección de las tecnologías de transmisión de radio para UMTS.

El modelo se basa en el modelo COST 231 definido como:

$$L = L_{FS} + L_C + \sum K_{Wi}L_{Wi} + n \left( \frac{n+2}{n+1} - b \right) \times L_f$$

Ecuación 1.3 (7)

Donde:

- $L_{FS}$  : Pérdida en espacio libre entre transmisor y receptor.

- $L_C$  : Constante de pérdida.
- $K_{Wi}$ : Número de paredes de tipo  $i$  penetradas.
- $L_{Wi}$ : Pérdida debida a muro de tipo  $i$ .
- $n$ : Número de suelos penetrados.
- $b$  : Parámetro empírico.
- $L_f$ : Pérdida entre suelos adyacentes.

NOTA:  $L_C$  se fija normalmente en 37 dB.

Tabla 2. Valores medios de los factores de pérdida.

Pérdidas	Descripción	Factor (dB)
$L_f$	Suelos(estructura típica), baldosas, revestimiento de hormigón, espesor típico < 30 cm.	18.3
$L_{W1}$	Muros internos finos, yeso, muros con muchos huecos (ventanas).	3.4
$L_{W2}$	Muros internos, hormigón, ladrillos, mínimo números de huecos.	6.9

El modelo COST 231 es un modelo empírico que tiene en cuenta las pérdidas de propagación en espacio abierto así como las pérdidas introducidas por las paredes, suelos y techos penetrados en el trayecto directo entre transmisor y receptor.

#### 1.1.4 Modelo basado en el número de muros y suelos

Caracteriza la pérdida de trayecto en interior por un exponente fijo de 2 (como en el espacio libre) y unos factores de pérdida relacionados con el número de suelos y muros que atraviesa la línea recta entre emisor y receptor.

$$L = L_1 + 20 \log(r) + n_f a_f + n_w a_w$$

Ecuación 1.4 (7)

Donde:

- $L_1$  : Pérdida de referencia con  $r = 1$  metro.
- $r$ : Distancia en metros en línea recta.

- $n_f$  : Número de suelos que atraviesa.
- $a_f$  : Atenuación por cada suelo que atraviesa.
- $n_w$  : Número de muros que atraviesa.
- $a_w$  : Atenuación por cada muro que atraviesa.

### 1.1.5 Modelo ITU-R

Es similar a los dos anteriores pero solo tiene en cuenta explícitamente el número de suelos. Las pérdidas en el mismo piso por atravesar muros, se incluyen implícitamente cambiando el exponente en la pérdida de trayecto.

$$LT = L_1 + 20 \log(fc[MHz]) + 10 \log(r[m]) + L_f n_f - 28$$

**Ecuación 1.5 (7)**

Donde:

- $L_1$  : Pérdida de referencia con  $r=1$  metro.
- $f$  : Frecuencia.
- $c$  : Velocidad de la luz.
- $r$  : Distancia en metros en línea recta.
- $L_f$  : Atenuación por cada suelo que atraviesa.
- $n_f$  : Número de suelos que atraviesa.

### 1.1.6 Modelo de Atenuación de Trayecto Linear

Para un transmisor y un receptor que se encuentran en la misma planta, Andelman propuso el “Modelo de Atenuación de Trayecto Linear” (7). La pérdida de trayecto medido en dB se obtiene de la pérdida de trayecto en espacio libre ( $PL_{FS}$ ) más un factor que es lineal del rango que experimentalmente se ha obtenido. La fórmula utilizada en este modelo es:

$$PL(d) = PL_{FS} + a \times d$$

**Ecuación 1.6 (7)**

Donde “ $a$ ” es el coeficiente de atenuación lineal y “ $d$ ” es la distancia entre transmisor y receptor. En el caso de un ambiente de oficinas, el coeficiente  $a$  sería 0.47 dB/m. Este modelo es bastante más simple que el modelo *Log-normal Shadowing Path Loss*, y no tiene en consideración efectos de desvanecimiento. Solo toma en consideración la pérdida en espacio libre. Este modelo es susceptible de ser específico de un sitio, debido a que el valor de “ $a$ ” varía de un entorno a otro y debe ser medido con precisión para poder demostrar los beneficios de este modelo. (7)

### 1.1.7 Modelo de Propagación de Doble Pendiente

Este modelo fue desarrollado por Feuerstein y Beyer (8). Ellos observaron que la pérdida de trayecto se comporta de diferentes maneras a distancias cercanas y a distancias largas. Para cuantificar este efecto se trabaja con dos modelos de pérdida de trayecto, uno para distancias cortas y otro para largas. El primero tiene un rango corto y tiene su propio índice de decaimiento. El segundo es función del primero. Para diferenciar la utilización de ambos modelos se introduce la distancia de ruptura  $d_{BR}$ . Las fórmulas utilizadas son las siguientes:

$$PL_{DS1}(d) = 10n_1 \log\left(4\pi d/\lambda\right) - a_0$$

**Ecuación 1.7 (7)**

$$PL_{DS2}(d) = PL_{DS1}(d_{BR}) + 10n_2 \log\left(d/d_{BR}\right)$$

**Ecuación 1.8 (7)**

Donde:

- $d_{BR}$ : Distancia de ruptura.
- $\lambda$ : Longitud de onda.
- $n_1$ : Exponente de pérdida de trayecto antes de  $d_{BR}$  ( $PL_{DS1}$ ).
- $n_2$ : Exponente de pérdida de trayecto después de  $d_{BR}$  ( $PL_{DS2}$ ).
- $a_0$ : Diferencia entre  $PL_{DS1}$  y  $PL_{DS2}$  a la distancia de 1 metro, causado por efectos en las ondas y varía entre 0 y 5 dB.



En las regiones cercanas,  $n_1$  se suele establecer a 2; y para las regiones lejanas  $n_2$  se suele poner un valor de 6 o mayor. Este modelo se puede hacer más específico para un sitio al poder introducir los valores  $n_1$  y  $n_2$ . Además la distancia de ruptura  $d_{BR}$  juega un papel muy importante, debe ser un valor que se conozca lo mejor posible y establecido en base a mediciones.

### 1.1.8 Modelo Keenan-Motley

Motley propuso un modelo que se basa tanto en la teoría, como en las medidas hechas en la práctica. Este modelo mide la pérdida de trayecto mediante la fuerza de la señal recibida y una distancia de referencia. La fórmula utilizada es la siguiente:

$$PL_{KM}(d) = PL_M + 10n \log(d) + K_F PL_{FS}$$

**Ecuación 1.8 (7)**

Donde:

- $PL_M$ : Pérdida de trayecto medido a 1 metro.
- $n$ : Exponente de pérdida de trayecto.
- $d$ : Distancia entre el transmisor y el receptor.
- $PL_{FS}$ : Pérdida de trayecto en espacio libre; incluyendo pérdidas por penetración a través de suelos/techos.
- $K_F$ : Número de suelos/techos penetrados.

La pérdida de trayecto medido es una buena forma de saber el comportamiento del modelo. Sin embargo, este modelo no puede ser utilizado para predecir el nivel de señal recibido, dado el requerimiento de medidas de pérdidas de trayectos reales.

### 1.1.9 Modelo Multi-Wall

Este modelo está caracterizado por un exponente de  $n = 2$ , como pérdida de trayecto en espacio libre; junto con factores de pérdida relacionados con el número de plantas ( $n_f$ ) y muros ( $n_w$ ) que interceptan la línea en visión directa a la distancia ( $d$ ) entre los terminales. La fórmula utilizada es la siguiente:

$$PL_{MW}(d) = PL_1 + 20 \log(d) + n_f a_f + n_w a_w$$

Ecuación 1.9 (9)

Donde:

- $PL_1$ : Pérdida de trayecto a 1 metro.
- $d$ : Distancia entre el transmisor y el receptor.
- $a_f$ : Factor de atenuación de suelos.
- $a_w$ : Factor de atenuación de muros.
- $n_f$ : Número de suelos atravesados.
- $n_w$ : Número de muros atravesados.

## 1.2 Modelos de propagación teóricos o deterministas.

Se basan en principios fundamentales de la física en cuanto a propagación de ondas de radio y los fenómenos que la rodean. Pueden ser aplicados en diferentes entornos sin afectarles su precisión. En la práctica su implementación requiere enormes bases de datos de características relativas al entorno, las cuales son imposibles o inviables de obtener de manera práctica.

Los algoritmos usados por los modelos deterministas son generalmente muy complejos y computacionalmente poco eficientes. Por esta razón su implementación se restringe a pequeñas áreas. Por el contrario, si su implementación es correcta, proporcionan gran precisión en su predicción comparados con los modelos empíricos.

Actualmente son los más utilizados debido a que ofrecen mejores resultados que los estadísticos en cuanto a precisión. Ahora se trabaja íntimamente con todas las variables que rodean el entorno y se emplean principios físicos y matemáticos en la resolución. También se puede emplear uno u otro modelo en función del problema que se enfrente, pudiendo desarrollar modelos híbridos con mejores resultados en cuanto a la relación de tiempo invertido y precisión obtenida. Todas las ventajas anteriores se enfrentan al problema de cálculo comentado arriba que todos estos modelos pueden presentar, ya que las variables pueden ser elevadísimas teniendo que realizar simplificaciones. (7)



### 1.2.1 Modelos basados en Técnicas de Trazado de Rayos

El algoritmo de Trazado de Rayos (*Ray Tracing*), calcula todos los posibles caminos desde el transmisor al receptor. En los modelos básicos de trazado de rayos, la predicción se basa en cálculos de transmisión en espacio abierto complementados con el efecto de reflexión de las paredes, teniendo en cuenta una única reflexión. Los algoritmos de trazado de rayos más complejos incluyen mecanismos de difracción, dispersión difusa y penetración a través de diversos materiales. (8)

En conclusión, el nivel de señal en un punto determinado se obtiene mediante la suma de los componentes de todos los caminos posibles existentes entre transmisor y receptor. Además de las pérdidas de propagación, en estos modelos se pueden predecir de manera fiable la dispersión en el tiempo de la señal.

Hoy en día los modelos de trazado de rayos están considerados entre los más precisos de los modelos de predicción de potencia de campo; sin embargo, requieren una representación muy detallada del área a analizar. La precisión del modelo depende en gran medida de la precisión y completitud/complejidad de la base de datos asociada a la representación. Por otro lado, el tiempo de computación depende en modo exponencial de estos detalles. Así, el tiempo de computación de un área reducida pero muy detallada puede ser mucho mayor que el de un área mayor pero con menor nivel de detalle. (7)

### 1.2.2 Modelo Diferencias Finitas en el Dominio del Tiempo.

Con los modelos que hacen uso de la Óptica Geométrica (GO, según sus siglas en inglés) y de la Teoría Uniforme de la Difracción (UTD, según sus siglas en inglés) se obtienen soluciones relativamente simples para el estudio de la radio propagación. Sin embargo, se sabe que GO obtiene buenos resultados cuando se trabaja con elementos eléctricamente grandes, y que UTD es rigurosa cuando se estudia el caso de conductores perfectos. Enfrentándose a estructuras con pérdidas y de finitas dimensiones, el trazado de rayos no es la solución óptima. Esto cobra mayor importancia cuando en los sistemas de comunicación más complejos tanto las antenas transmisoras como receptoras se instalan próximas a estructuras de materiales complejos para los que no se puede aplicar soluciones asintóticas, aunque sí la solución a las ecuaciones de Maxwell. (10) Por lo tanto, las características de la propagación de radio se pueden derivar resolviendo

directamente las ecuaciones de Maxwell relativas a propagación de ondas electromagnéticas.

El método Diferencias Finitas en el Dominio del Tiempo (FDTD, según sus siglas en inglés) es probablemente el método más utilizado para la obtención de una solución numérica de las ecuaciones de Maxwell. En este método las ecuaciones de Maxwell se aproximan por medio de un conjunto finito de ecuaciones diferenciales. Antes de realizar los cálculos es necesario definir una cuadrícula específica (regular o irregular) sobre el área de interés. Una vez que estas condiciones iniciales han sido definidas, el algoritmo FDTD emplea las diferencias centrales para aproximar las derivadas tanto en el espacio como en el tiempo. En los nodos de la cuadrícula la solución se determina iterativamente, resolviéndose de este modo las ecuaciones de Maxwell directamente.

Al igual que los modelos basados en trazado de rayos, el modelo FDTD es muy exigentes computacionalmente. El tiempo de computación depende proporcionalmente del tamaño del área a ser analizada, pero no significativamente de los detalles incluidos en su descripción. Sin embargo, el número de nodos de la cuadrícula está exponencialmente relacionado con el tamaño del área y la frecuencia de operación.

Las ventajas de este modelo son su exactitud y que facilita una solución completa para todos los puntos de una geometría, lo que puede servir para obtener mapas de cobertura de una determinada área. La precisión del modelo FDTD es comparable a la de los modelos basados en trazado de rayos, y la predicción es tan precisa como pueda serlo la base de datos de descripción del área.

Debido a su complejidad computacional, el modelo FDTD es aconsejables solo en tareas de predicción de campo en pequeñas áreas. Para áreas de mayor tamaño los modelos de trazado de rayos son más aconsejables. (7)

### **1.2.3 Modelos basados en el método de los momentos**

Mientras que los modelos de trazado de rayos ofrecen buenos resultados para predecir coberturas en grandes edificios teniendo un gran número de paredes entre el transmisor y el receptor, si se presenta la situación contraria, el método de los momentos (MoM, por sus siglas en inglés), es la mejor solución si se quiere una mayor precisión. Por supuesto, es posible emplear una combinación de ambos modelos disfrutando de las ventajas que cada uno ofrece. Para casos donde predominan una gran cantidad de pequeños

obstáculos o donde hay trayectorias que no pueden tomarse con el modelo de trazador de rayos se emplearía el modelo basado en MoM.

Con MoM se obtienen soluciones tan exactas al problema como de pequeña sea la segmentación espacial que se emplee para los objetos. Debido a las limitaciones tanto de memoria como de CPU, MoM se suele aplicar para el análisis de objetos del orden de diez veces mayores que la longitud de onda. Sin embargo, en estructuras con dimensiones próximas a unas pocas longitudes de onda, MoM puede ser útil para comprobar y verificar un programa basado en técnicas de trazado de rayos. (7)

#### 1.2.4 Modelo de Trayecto Dominante

El Modelo de Trayecto Dominante determina exactamente el camino dominante entre el transmisor y el receptor. Así que el tiempo de cálculo en comparación con el trazado de rayos se reduce significativamente y la exactitud casi idéntica a dicho método.

Las principales características de este modelo son: (11)

- La dependencia de la precisión de la base de datos de vectores se reduce (en comparación con el trazado de rayos).
- Sólo se considera la trayectoria de propagación más importante, porque este camino ofrece la parte principal de la energía.
- Cortos tiempos de cálculo.
- La precisión alcanza o supera la precisión de los modelos de rayos ópticos

La ecuación utilizada para el cálculo de la pérdida de trayectoria es:

$$L = 20 \log \left( \frac{4\pi}{\lambda} \right) + 10p \log(d) + \sum_{i=0}^n f(\varphi, i) + \sum_{j=0}^m t_j - \Omega$$

**Ecuación 1.10 (7)**

Donde:

- $\lambda$ : Longitud de onda.
- $d$ : Distancia entre el transmisor y el receptor.
- $p$ : Exponente de pérdida de trayectoria, dependiendo de la situación actual de propagación.

- $f(\varphi, i)$  : Función de atenuación individual para cada interacción  $i$  de todas las  $n$  interacciones.
- $t_j$  : Pérdida de transmisión para cada penetración  $j$  a través de una pared.
- $\Omega$ : Ganancia causada por el guiado de ondas para cada píxel a lo largo de una trayectoria de propagación.

Como se describió anteriormente,  $d$  es la longitud de la trayectoria entre el transmisor y el receptor de píxel actual. El exponente de pérdida de trayectoria es  $p$ . La función  $f$  brinda la pérdida (en dB) que es causada por una difracción. Todas las pérdidas de difracción se acumulan a lo largo de una trayectoria de propagación. Las pérdidas de transmisión (penetración)  $t_j$  también se acumulan a lo largo de la trayectoria de propagación, así como los efectos de guía de ondas.

El guiado de ondas se produce si la onda se propaga a través de un largo pasillo y las reflexiones en las paredes aparecen. En este caso una ganancia adicional de guía de ondas (en dB) se determina y se acumula a lo largo de todos los píxeles en la trayectoria de propagación actual.

Este modelo puede distinguir entre varios modos de propagación para definir el exponente de pérdida de trayectoria. La Figura 3 muestra los diferentes modos y las transiciones entre ellos. Los modos son: (12)

- LOS: línea de visión directa entre el transmisor y el receptor.
- OLOS (obstruido LOS): el transmisor y el receptor se encuentran en la misma habitación y el camino no necesita ninguna penetración de la pared, pero no hay una línea de visión directa entre el transmisor y el receptor.
- NLOS (non-LOS): Se requiere al menos una penetración a través de una pared entre el transmisor y el receptor.

Los valores sugeridos para los diferentes modos son: LOS de 2.0 a 2.2, OLOS de 2.1 a 2.3, NLOS de 2.3 a 2.5.

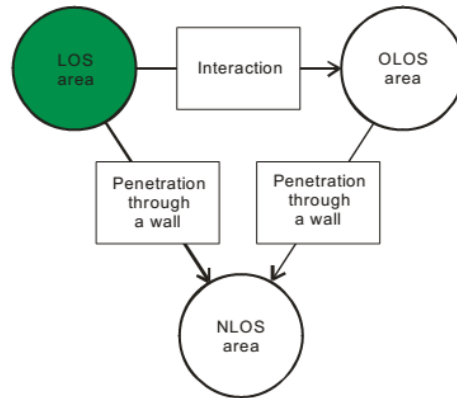


Figura 3. Varios modos de propagación (11)

### 1.3 Modelos de propagación seleccionados

Como se explicaba en secciones anteriores los modelos empíricos, como el Multi-Wall, tienen a su favor su gran eficiencia computacional (13), además, estos modelos consideran las influencias propias del entorno de manera implícita en su conjunto, sin ser reconocidas cada una de ellas de forma aislada. Sin embargo, la exactitud de sus simulaciones, depende de la precisión y similitud de las medidas del entorno donde fueron tomadas y el nuevo entorno donde se va a aplicar el modelo. Además, se basan en el cálculo de la señal transmitida en la trayectoria directa desde el transmisor hacia el receptor, sin considerar mecanismos de propagación en interiores, causantes del problema del multitrayecto de la señal.

En cambio, el modelo Trayecto Dominante (14), se centra en el/los trayecto(s) dominante(s) entre el transmisor y el receptor. Este modelo establece los parámetros que determinan estos trayectos dominantes (longitud del trayecto, las propiedades de los materiales de los objetos con los que interactúa el trayecto, el número y el tipo de interacciones, etc.), los cuales son usados en la predicción.

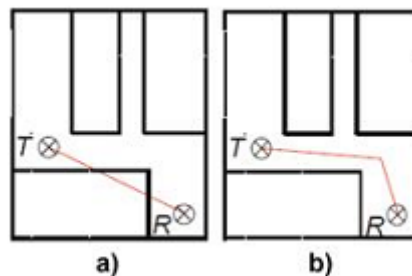


Figura 4. a) Modelo Multi-Wall, b) Modelo Trayecto Dominante

Teniendo en cuenta las ventajas que presentan los modelos Multi-Wall y Trayecto Dominante, los autores de la presente investigación y el cliente consideran que las simulaciones realizadas por la herramienta propuesta en su primera versión deben basarse en el empleo de estos dos modelos de propagación.

#### 1.4 Herramientas que realizan simulaciones mediante modelos de propagación

Existen varias herramientas que incluyen modelos de propagación, a continuación se hace referencia a algunas de ellas.

##### 1.4.1 ProMan - WinProp Software Suite

WinProp es el principal producto de AWE Comunicaciones. Junto con varios socios AWE Comunicaciones fue capaz de desarrollar un paquete de software ideal para la propagación de ondas de radio en diferentes escenarios y para la planificación de redes de diferentes interfaces de aire. ProMan (propagación) está diseñado para predecir la pérdida de trayectoria con precisión entre los transmisores y receptores, incluyendo todos los parámetros importantes del canal de radio móvil. Su estructura es muy modular, condición que garantiza adaptar el paquete de software a los requisitos del cliente. (11) WinProp puede calcular las pérdidas de trayectoria muy rápidamente y las características del ancho de banda de los enlaces por radio en el interior de los edificios. Para hacerlo utiliza diversas combinaciones de modelos de propagación disponibles para entornos interiores. (15)

Se pueden dividir en dos grupos:

**Modelos empíricos:** Modelo Keenan-Motley, Modelo COST 231 y Multi-Wall.

**Modelos deterministas:** Modelo de Trayecto Dominante, Trazado de rayos estándar en 3D y Trazado de rayos inteligente en 3D.

La aplicación incluye también una opción para predicciones en edificios de varias plantas. Esta opción requiere un cálculo sobre cada planta. WinProp puede calcular predicciones en un número arbitrario de alturas y puede exhibir el resultado con vistas en 3D. Así, se pueden predecir fácilmente las interferencias entre los pisos y los problemas de cobertura. La Figura 5 es un ejemplo de predicción del programa: (15)



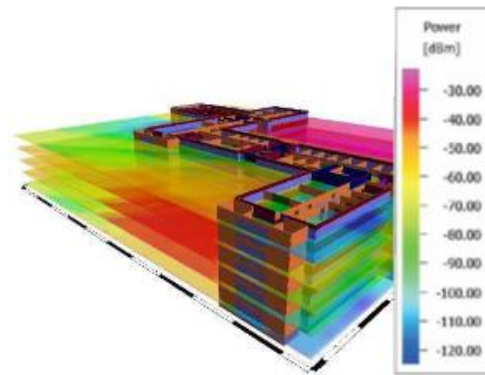


Figura 5. Ejemplo de predicci3n para un entorno interior

### 1.4.2 WISAN Positioning Tool

Herramienta desarrollada por la empresa SoftCrits (de la Universidad de M3laga). Tiene como objetivo la generaci3n de dise1os de despliegues de WISAN para entornos interiores y ser empleada por los dise1adores de estas redes como una herramienta de apoyo. Para la generaci3n de la propuesta dise1o en un primer momento se genera un mapa de posiciones candidatas y posteriormente se basan en un algoritmo de optimizaci3n que es el encargado de generar el dise1o final. Para realizar las predicciones del comportamiento de la se1al de RF esta herramienta provee los modelos de propagaci3n Multi-Wall y el Trayecto Dominante.

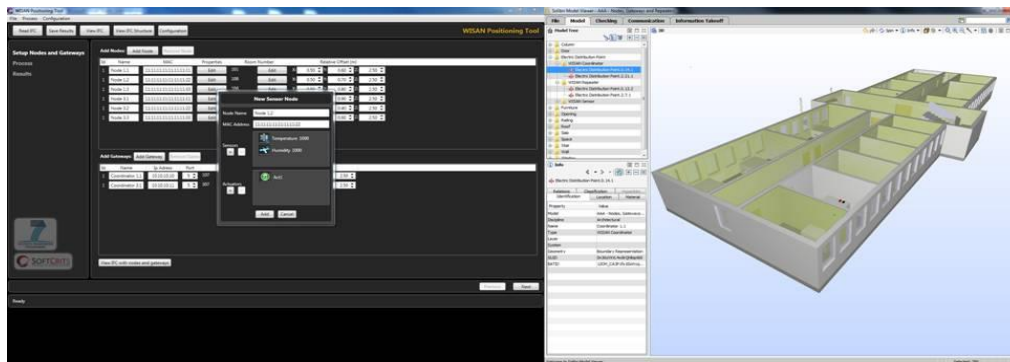


Figura 6. WISAN positioning Tool

### 1.4.3 Otras herramientas

Una propuesta muy interesante es la presentada por Guinard *et al.* (5), la cual tiene el objetivo de facilitar el despliegue de WISAN en SCE. Para ello se basaron en el desarrollo



de una herramienta que incluye un algoritmo de optimización escalable para el diseño de WSAW y una metodología integrada.

Esta herramienta fue creada con la finalidad de ayudar a los diseñadores e integradores de sistemas en una etapa previa al despliegue de la red de sensores, realizando una propuesta automática del número y la ubicación de los dispositivos inalámbricos.

En dicha herramienta, el proceso de optimización y diseño automático de la red se produce en dos fases. Se emplea un algoritmo basado en una red neuronal de tipo Gas Neuronal Creciente, para generar las posiciones candidatas de los dispositivos inalámbricos. Cada dispositivo posee, además de su posición candidata, un mapa de cobertura de su señal de radio correspondiente; para esta predicción la herramienta se basa en un modelo de propagación basado en el método Trazado de Rayos. (5)

### **1.5 Lenguaje de programación**

Para la selección del lenguaje de programación de la herramienta se tuvo en cuenta la complejidad de los algoritmos que serán usados para predecir conectividad de dispositivos inalámbricos, los requisitos relacionados con la visualización 3D del entorno, además de que es necesaria la utilización de técnicas de inteligencia artificial. Basado en estas características el equipo de arquitectos de la herramienta definió la utilización de Java como lenguaje de implementación.

Java fue desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Es un lenguaje robusto, las aplicaciones creadas con este lenguaje es poco probable que posean errores, principalmente porque la gestión de memoria y punteros es realizada por el propio lenguaje y no por el programador, además posee estructuras para la detección de excepciones y permite obligar al programador a escribir código fiable mediante la declaración de excepciones posibles para una determinada clase reutilizable. (16)

Las aplicaciones desarrolladas con Java pueden ser ejecutadas indistintamente en muchos equipos, ya que son independientes de la arquitectura, es un lenguaje de desarrollo público y se puede tener acceso gratis al conjunto de herramientas de desarrollo de aplicaciones de Java (*Java Development's Kit*, JDK según sus siglas en Inglés). Posee gran conectividad con Bases de Datos y otros sistemas.



Una característica importante que se debe tener en cuenta para el desarrollo de la herramienta propuesta, es el trabajo con modelos tridimensionales, aspecto en el que Java tiene un buen desempeño, debido a que presenta gran variedad de librerías libres con buena potencia gráfica. Además, se caracteriza por ser muy utilizado en la implementación de aplicaciones que incorporan técnicas de inteligencia artificial.

Existe además variada documentación y una amplia comunidad sobre este lenguaje. Sin embargo, resulta importante señalar que los programas realizados en Java suelen ser un poco lentos y para el manejo de código de bajo nivel, se deben utilizar métodos nativos, disminuyendo así su portabilidad.

### **1.6 Entorno de desarrollo**

Para el desarrollo de la solución el equipo de arquitectura seleccionó como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) el NetBeans 7.3. Es libre y de código abierto. Posee todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. (17)

Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente.

### **1.7 Librería 3D para el desarrollo**

Para la representación y el modelado en tercera dimensión, Java posee potentes librerías gráficas que facilitan el trabajo y la interacción con los objetos 3D, dentro de las principales librerías se encuentran: JOGL, JGL y Java3D. El equipo de desarrollo del visor de la herramienta Andrómeda, teniendo en cuenta las características de las librerías gráficas para el modelado 3D, seleccionó la librería Java3D dadas las ventajas que esta presenta sobre las librerías antes mencionadas, por ejemplo: la librería JOGL necesita además de los ficheros .jar las librerías de enlaces dinámicos dependiendo de la plataforma donde se ejecute la aplicación, no siendo esto necesario si se utiliza Java3D, ya que no usa librerías de enlace dinámico. Tanto JOGL como JavaGL no están orientadas a objetos lo cual atrasa considerablemente el desarrollo de la aplicación, en

cambio Java3D permite a los desarrolladores generar aplicaciones de forma rápida empleando las facilidades que brinda la Programación Orientada a Objetos. Además del soporte a los cargadores que permite que se adapte a un gran número de formatos de ficheros. (18)

### 1.8 Librería para cargar ficheros IFC

Se utilizará para el manejo de ficheros en formato IFC (Industry Foundation Classes) a la librería OpenIFCTools<sup>1</sup>. Es un proyecto de código abierto y multiplataforma que utiliza tecnología Java.

El uso de esta librería permite el desarrollo orientado a objetos. Cada clase del fichero IFC está representada por una clase Java con sus atributos correspondientes. En este sentido, es posible crear una instancia y manejar cualquier objeto usando las clases de Java. Es posible leer y escribir ficheros en formato IFC haciendo uso de métodos sencillos, así como registrar *listeners* en el modelo para informar los cambios que ocurren en el mismo, lo cual puede aplicarse a cualquier objeto utilizando los métodos de acceso en lugar de acceder directamente a los atributos (todos los atributos son públicos, lo que permite el acceso a los datos). (19)

### 1.9 Metodología de desarrollo de software

En el estudio de las metodologías de desarrollo realizado por el equipo de arquitectura de la herramienta Andrómeda, se decide seleccionar *Programación Extrema* (*eXtremeProgramming*, *XP* según sus siglas en inglés). Debido a que el período de desarrollo es un proceso que se limita solamente a pocos meses de trabajo, el cliente forma parte del equipo de desarrollo y no son grandes las dimensiones del proyecto.

A continuación se describen las características más relevantes de esta metodología: (20) *Cortas iteraciones*: La entrega continua de resultados compromete a ambas partes en la evolución del proyecto, además de influir directamente de forma positiva en la calidad del producto final. En cada iteración, se obtiene un producto listo para entregar y que tiene valor para el cliente.

---

<sup>1</sup> [www.openifctools.com](http://www.openifctools.com)

*Retroalimentación con el cliente o feedback:* Consiste en que al menos un miembro del equipo de desarrollo del proyecto, es un cliente. Lo cual propicia una constante interacción del mismo con el producto en desarrollo.

*Muy flexible a cambios:* Esta es una de las características más reconocidas de XP. Debido a la constante retroalimentación con los clientes, es posible prever futuros cambios en el desarrollo, evitando así llegar a momentos que paralicen el desarrollo del producto.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (21)

Otras características que posee esta metodología y que además ayudan a aumentar la productividad del equipo de desarrollo son: la programación en pares, reuniones diarias y planes de entrega a corto plazo.

#### 1.10 Herramienta para el modelado

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

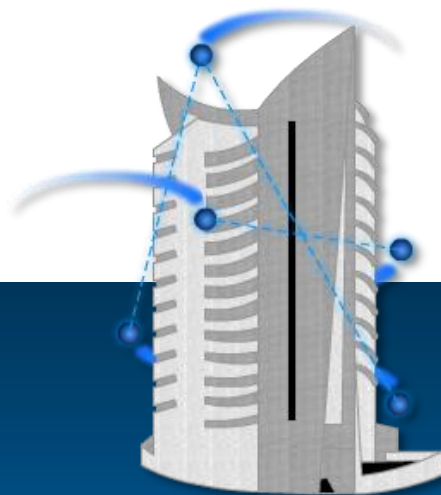
VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos. (22)

Ofrece interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo. Hace posible la generación de código Java desde los diagramas. Uno de los diagramas más utilizados de UML es el Diagrama de Clases del Diseño, VP-UML permite la generación de código a partir de este.



### 1.11 Conclusiones parciales

La investigación realizada en el presente capítulo determinó que la literatura propone diversas variantes de clasificación de los modelos de propagación, de forma general, se pueden clasificar en empíricos o estadísticos y teóricos o deterministas. Los primeros se basan en la extrapolación estadística de resultados a partir de mediciones realizadas sobre el terreno, mientras que los segundos se apoyan en principios físicos en cuanto a la propagación de ondas de radio y fenómenos paralelos. Además, se evidenció la existencia de varios de estos modelos (el Multi-Wall y el Trayecto Dominante fueron los seleccionados a ser implementados) y de herramientas que los utilizan en sus simulaciones, los cuales fueron descritos y analizados. En el resto del capítulo se describieron y seleccionaron las herramientas y tecnologías a ser usadas en el desarrollo de la propuesta de solución: siguiendo la línea base que definió la arquitectura, se determinó usar como lenguaje de programación Java, utilizando como IDE NetBeans v7.3. Las librerías usadas para el desarrollo serán Java 3D y OpenIFCTools. Como metodología de desarrollo se seleccionó a XP y como herramienta CASE Visual Paradigm for UML. Luego de concluir con la fundamentación teórica quedan sentadas las bases para definir las características del sistema a desarrollar.



Andr6meda  
ΑΝΔΡΟΜΕΔΑ

# Capítulo 2

Descripción de la solución propuesta

## 2. Descripción de la solución propuesta

El diseño del despliegue de la red de sensores inalámbricos en una edificación es una compleja tarea que necesita de la simulación del comportamiento de la señal de radio frecuencia dentro de edificaciones. En el presente capítulo se propone una solución al problema de simular la propagación de la señal de radio frecuencia en la herramienta Andrómeda, pues se describen los aspectos funcionales y no funcionales, así como las historias de usuario que permitirán implementar los modelos de propagación seleccionados, los cuales determinan con precisión la conectividad entre dispositivos inalámbricos, de forma tal que se puedan posicionar correctamente en el interior de un edificio. Además se abarcarán las dos primeras etapas de la metodología XP (Exploración y Planificación) con el fin de conocer el alcance, estimar los tiempos de entrega de cada iteración y los artefactos que se generan.

### 2.1 Usuarios del sistema

Los usuarios del sistema son todas aquellas personas o sistemas que interactúan con el mismo con el objetivo de obtener un resultado específico. La herramienta Andrómeda no presenta restricciones de acceso a ciertas funcionalidades para un grupo específico de usuarios, sino que todos tienen los mismos privilegios sobre las funcionalidades internas de la aplicación.

Tabla 3. Usuarios del sistema

Usuarios del sistema	Justificación
Diseñador de WSAN	Puede ser tanto un diseñador de redes con una vasta o con poca experiencia en el diseño de WSAN. En caso de que el usuario posea pocos conocimientos sobre el trabajo con la aplicación, puede acceder a los tutoriales de instrucción en línea para adquirir los conocimientos básicos de trabajo con la misma.

### 2.2 Lista de Reservas del Producto

Según define la metodología XP la **Lista de Reservas del Producto** está compuesta por los aspectos funcionales con los que debe contar la aplicación para satisfacer con



eficiencia las necesidades requeridas por el cliente. A continuación se muestra el listado de los mismos.

- R1. Visualizar los modelos de propagación implementados en la herramienta Andrómeda.
- R2. Permitir la selección del modelo de propagación a usarse en la simulación.
- R3. Calcular conectividad entre dos dispositivos inalámbricos usando el modelo Multi-Wall.
  - R3.1. Calcular distancia entre dos puntos en el espacio.
  - R3.2. Identificar obstáculos existentes en un entorno interior.
  - R3.3. Determinar la variable pérdida de trayecto en dependencia del tipo de entorno.
  - R3.4. Determinar el factor de atenuación de suelos y muros.
- R4. Calcular conectividad entre dos dispositivos inalámbricos usando el modelo Trayecto Dominante.
  - R4.1. Generar el árbol con las paredes del escenario.
  - R4.2. Buscar el mejor camino a través de las paredes.
  - R4.3. Determinar el exponente de pérdida de trayectoria entre varios modos de propagación.
  - R4.4. Calcular la pérdida de iteración acumuladas a lo largo de una trayectoria.
  - R4.5. Determinar la ganancia causada por guía de ondas para cada pixel a lo largo de una trayectoria.
  - R4.6. Calcular la pérdida por cada penetración a través de las paredes.

### **2.3 Aspectos no funcionales del sistema**

Los aspectos no funcionales son propiedades o cualidades que el producto debe tener. Son características del mismo que lo hacen atractivo, usable, rápido y confiable.

#### **Software**

- Se requiere de la Máquina Virtual de Java versión 1.6 o superior.
- Se requiere de la librería Java3D.

#### **Hardware**

- Memoria RAM: 128 MB como mínimo para sistemas Windows y 64 MB para las restantes plataformas.



- Espacio en disco duro: 98 MB mínimo para Windows y más de 58 MB para los restantes.

### **Usabilidad**

- Los módulos serán diseñados para que integren todos los aspectos relacionados con la usabilidad. Esta característica permitirá que el sistema sea eficiente, fácil de aprender con un nivel de abstracción adecuado para la retención de ideas, fácil de usar tanto por diseñadores experimentados como por personas de poca experiencia en el diseño de WSA.

### **Rendimiento**

- La herramienta deberá ser capaz de calcular la conectividad entre dos dispositivos inalámbricos en un lapso de tiempo inferior a 1 minuto, el coste computacional de los cálculos necesarios debe ser aceptable en los sistemas actuales.

### **Portabilidad**

- La herramienta deberá funcionar en los sistemas operativos Windows (XP o superior), Mac, Solaris y las distribuciones de GNU/Linux.

### **Aspectos legales**

- La herramienta y por ende los modelos de propagación que contendrá, una vez desarrollada serán de código abierto y licencia GPL.

## **2.4 Exploración**

Como punto de partida en el ciclo de vida de un proyecto, XP propone la **Exploración** como la primera etapa donde los clientes plantean a grandes rasgos las Historias de Usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (23)

### 2.4.1 Historias de Usuario

Las Historias de Usuario (HU) son un instrumento para el levantamiento de requerimientos para el desarrollo de un software, que ha emergido con la aparición de los nuevos marcos de trabajo de desarrollo ágil, son descripciones cortas de una necesidad de un cliente del software que se esté desarrollando.

Algunas características deseables de las HU son: (24)

- Que sean escritas por el usuario o por un analista de negocio que le represente.
- Frase corta que encaje en una tarjeta de tres por cinco pulgadas.
- Debe describir el rol desempeñado por el usuario en el sistema, descrito de forma explícita.
- Debe describir el beneficio para el área de negocio que representa esta funcionalidad.

A continuación se muestran las descripciones de cada una de las HU propuestas por el cliente.

Tabla 4. HU 1: Visualizar los modelos de propagación implementados en la herramienta Andrómeda.

Historia de Usuario	
<b>No.:</b> 1	<b>Nombre:</b> Visualizar los modelos de propagación implementados en la herramienta Andrómeda.
<b>Usuario:</b> Todos	
<b>Prioridad en el Negocio:</b> Media	<b>Nivel de Complejidad:</b> Media
<b>Estimación:</b> 1 semana	<b>Iteración Asignada:</b> 1
<b>Descripción:</b> La herramienta deberá proporcionarle al usuario varias opciones para realizar el diseño de las WSA, para eso se mostrará un listado de los modelos de propagación implementados para predecir la señal de RF.	
<b>Información adicional (Observaciones):</b> Da cumplimiento a la funcionalidad "Visualizar los modelos de propagación implementados en la herramienta".	

Tabla 5. HU 2: Permitir la selección del modelo de propagación a usarse en la simulación.

Historia de Usuario	
<b>No.:</b> 2	<b>Nombre:</b> Permitir la selección del modelo de propagación a usarse en la simulación
<b>Usuario:</b> Todos	
<b>Prioridad en el Negocio:</b> Alta	<b>Nivel de Complejidad:</b> Baja

<b>Estimaci3n:</b> 1 semana	<b>Iteraci3n Asignada:</b> 1
<b>Descripci3n:</b> El cliente podr3 seleccionar el modelo de propagaci3n con el que quiere realizar la simulaci3n.	
<b>Informaci3n adicional (Observaciones):</b> Debe cumplirse antes la funcionalidad "Visualizar los modelos de propagaci3n implementados en la herramienta". Adem3s solo se podr3 seleccionar un modelo a la vez.	

Tabla 6. HU 3: Calcular conectividad entre dispositivos inal3mbricos usando Multi-Wall.

<b>Historia de Usuario</b>	
<b>No.:</b> 3	<b>Nombre:</b> Calcular conectividad entre dos dispositivos inal3mbricos usando Multi-Wall
<b>Usuario:</b> Todos	
<b>Prioridad en el Negocio:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimaci3n:</b> 4 semanas	<b>Iteraci3n Asignada:</b> 1
<b>Descripci3n:</b> Para calcular la conectividad entre dos dispositivos inal3mbricos usando el modelo de propagaci3n Multi-Wall primero se debe calcular la distancia entre los dos puntos del espacio que ocupan estos dispositivos; despu3s se identifican los obst3culos existentes entre estos puntos, lo pr3ximo es determinar el factor de atenuaci3n de muros y suelos y el valor de la variable p3rdida de trayecto. Una vez que se tienen todos estos datos se procede a calcular la conectividad con la ecuaci3n que ofrece dicho modelo.	
<b>Informaci3n adicional (Observaciones):</b> Tanto el factor de atenuaci3n de muros y suelos como el valor de la variable p3rdida de trayecto van a depender del tipo de entorno (si el entorno tiene condiciones de visi3n directa o no, de su tipo de material, entre otras cosas).	

Tabla 7. HU 3: Calcular conectividad entre dispositivos inal3mbricos usando el modelo Trayecto Dominante.

<b>Historia de Usuario</b>	
<b>No.:</b> 3	<b>Nombre:</b> Calcular conectividad entre dos dispositivos inal3mbricos usando el modelo Trayecto Dominante.
<b>Usuario:</b> Todos	
<b>Prioridad en el Negocio:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimaci3n:</b> 4 semanas	<b>Iteraci3n Asignada:</b> 2

**Descripción:** Para calcular la conectividad entre dos dispositivos inalámbricos usando el modelo de Trayecto Dominante se debe determinar exactamente ese camino dominante entre el transmisor y el receptor, para ello se analizan los tipos de paredes del escenario, se genera un árbol con estas paredes y se busca el mejor camino a través de ellas. Una vez hecho esto se determina el exponente de pérdida de trayecto entre los modos de propagación LOS, OLOS, NLOS. Se determina la función de atenuación para cada iteración causada por difracción, así como la pérdida por transmisión y la ganancia causada por guía de ondas para cada pixel a lo largo de una trayectoria de propagación. Una vez que se tienen todos estos datos se procede a calcular la conectividad con la ecuación que ofrece dicho modelo.

**Información adicional (Observaciones):** La pérdida debido a una interacción (por ejemplo, difracción) a lo largo de una trayectoria de propagación provoca una atenuación adicional, de ahí que la función de atenuación va a depender del ángulo de incidencia y el rayo difractado. El factor de guía de ondas tiene en cuenta, que una onda que se propaga en un largo pasillo, se refleja en las paredes y esto conduce a una menor atenuación en comparación con el espacio libre.

## 2.5 Planificación

La fase de Planificación se realiza en pocos días, no siendo menos importante, ya que su objetivo principal es priorizar las HU. Los programadores estiman el esfuerzo de cada una de ellas, se define el cronograma y se acuerda el alcance de la entrega, el cual debe incluir varias iteraciones. El cliente prioriza las HU seleccionando cuales se realizarán en cada iteración, logrando en la primera iteración un sistema con la arquitectura del proyecto.

Otro aspecto a tener en cuenta durante esta fase es la velocidad del proyecto, la cual se estima utilizando el tiempo empleado en el desarrollo de las iteraciones terminadas. Esta medida debe reevaluarse una vez concluida 3 o 4 iteraciones y en caso de no cumplir el tiempo estimado, debe ser negociando con el cliente un nuevo Plan de Entregas. (23)

### 2.5.1 Iteraciones

XP enfatiza en el carácter iterativo e incremental del desarrollo, la fase de **Planificación** es la principal en el ciclo de vida de un proyecto. Las iteraciones agrupan un conjunto de HU a implementar en un período de tiempo, generando al final de cada una un entregable funcional. Son relativamente cortas. La entrega con rapidez de módulos al cliente aumenta la retroalimentación y resulta más provechoso en términos de calidad del producto que una entrega a largo plazo.

Las HU no cuentan con suficiente detalle como para permitir su análisis y desarrollo sin la presencia del cliente durante la fase. Por lo que se hace necesario al comienzo de cada iteración realizar las tareas necesarias de análisis en conjunto con el cliente, consolidado los datos necesarios para la implementación. La participación del cliente durante esta fase del ciclo es de vital importancia. Culminar sin errores una iteración constituye un avance notable en el desarrollo del proyecto.

**Tabla 8. Plan de Iteraciones.**

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo
Iteración 1	HU1. Visualizar los modelos de propagación implementados en la herramienta. HU2. Permitir la selección del modelo de propagación a usarse en la simulación. HU3. Calcular conectividad entre dos dispositivos inalámbricos usando Multi-Wall.	6 semanas
Iteración 2	HU4. Calcular conectividad entre dos dispositivos inalámbricos usando el modelo de Trayecto Dominante.	8 semanas

### 2.5.2 Plan de Entregas

El Plan de Entregas es el resultado final de la fase de Planificación acordado en una reunión entre todos los actores del proyecto, estableciendo cuáles HU serán agrupadas para conformar una entrega. Es un compromiso que establece el grupo de trabajo con el cliente, aspecto de vital importancia, debido a que entregas tardías podrían tener grandes repercusiones en la economía y moral del equipo de trabajo.

El cronograma de entregas es realizado en dependencia de las estimaciones de tiempos de desarrollo realizadas por los programadores. La estimación es uno de los temas más complicados del desarrollo de un proyecto de software, es por ello que se recomienda luego de algunas iteraciones realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el Plan de Entregas y ajustarlo en caso de ser necesario.

Tabla 9. Plan de Entrega.

Entregable	Final 1ra iteración 3 <sup>ra</sup> semana de Marzo de 2014	Final 2da iteración 3 <sup>ra</sup> semana de Mayo de 2014
Módulos IndoorModel, MultiWallModel y DominantPahtModel de la herramienta Andrómeda	versión 1.0	versión 2.0

## 2.6 Diseño

XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema robusto y reutilizable tratando de mantener su simplicidad, es decir, crear un diseño evolutivo que se va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente.

### 2.6.1 Patrones de diseño

Un patrón de diseño se define como *"... un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos"*. Los patrones de diseño comunican los estilos y soluciones consideradas como "buenas prácticas", que los expertos en el diseño orientado a objetos utilizan para la creación de sistemas.(26)

Se clasifican en patrones GRASP y GoF. Los patrones GRASP describen principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (27). Mientras que los patrones GoF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. (28)

Para el desarrollo de la herramienta Andrómeda se emplearon los patrones de diseño GRASP: **Experto**, **Creador**, **Alta Cohesión** y **Bajo Acoplamiento** y los patrones GoF: **Singleton** y **Observer**. En el [Anexo 1](#) se aborda con ejemplos dónde se emplean cada uno de estos patrones.

### 2.6.2 Modelo de datos

Es la descripción de la organización de una base de datos, constituyéndose en una representación gráfica orientada a la obtención de la estructura de datos mediante

métodos. En un enfoque más amplio, un Modelo de Datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí. El Modelo de Datos está formado por dos componentes: componente estática, relacionada con el lenguaje de definición de datos (LDD) y dinámica, relacionada con el lenguaje de manipulación de datos (LMD). (29)

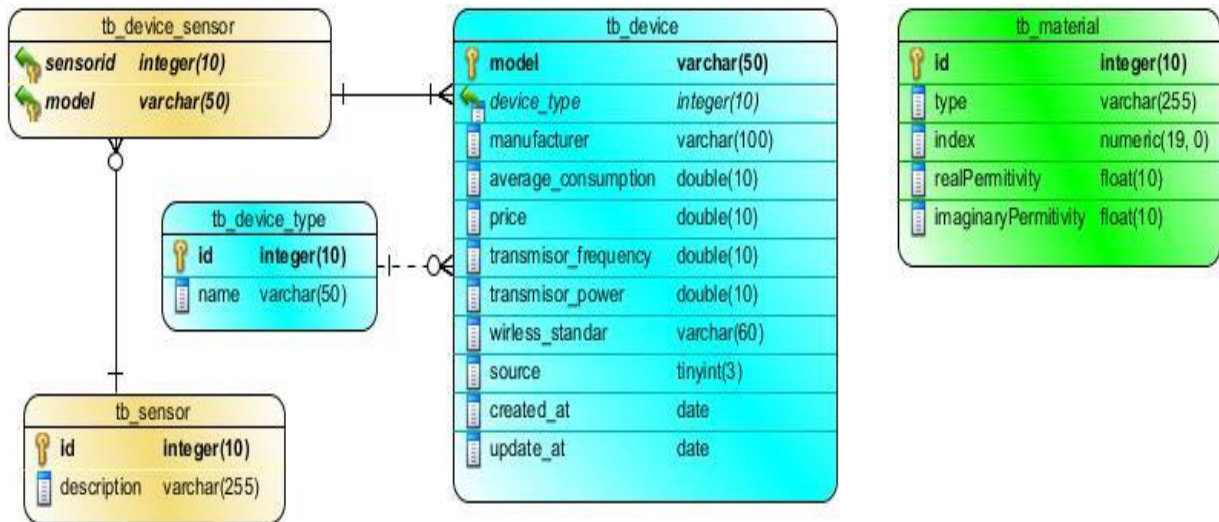


Figura 7. Modelo de Datos

A continuación se muestra la descripción de las tablas que conforman la Base de Datos empleadas por la herramienta para su funcionamiento.

**tb\_device:** Almacena la información persistente de los dispositivos empleados para el trabajo con la herramienta.

**tb\_device\_type:** Esta tabla es un nomenclador que almacena los diferentes tipos de dispositivos que son empleados para el trabajo con la herramienta. Los mismos pueden ser Coordinator, Router y EndDevice.

**tb\_sensor:** Nomenclador que almacena los tipos de sensores que pueden estar contenidos dentro de los dispositivos utilizados en el sensado.

**tb\_device\_sensor:** Esta tabla representa la relación que existe entre los dispositivos y los sensores correspondientes a cada dispositivo.

**tb\_material:** Almacena la información de los diferentes materiales de los elementos no estructurales que pueden estar dentro de las edificaciones.



### 2.6.3 Diagrama de clases

Los Diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.). Son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). (28)

En la Figura 8 se muestra el Diagrama de clases de los módulos implementados.

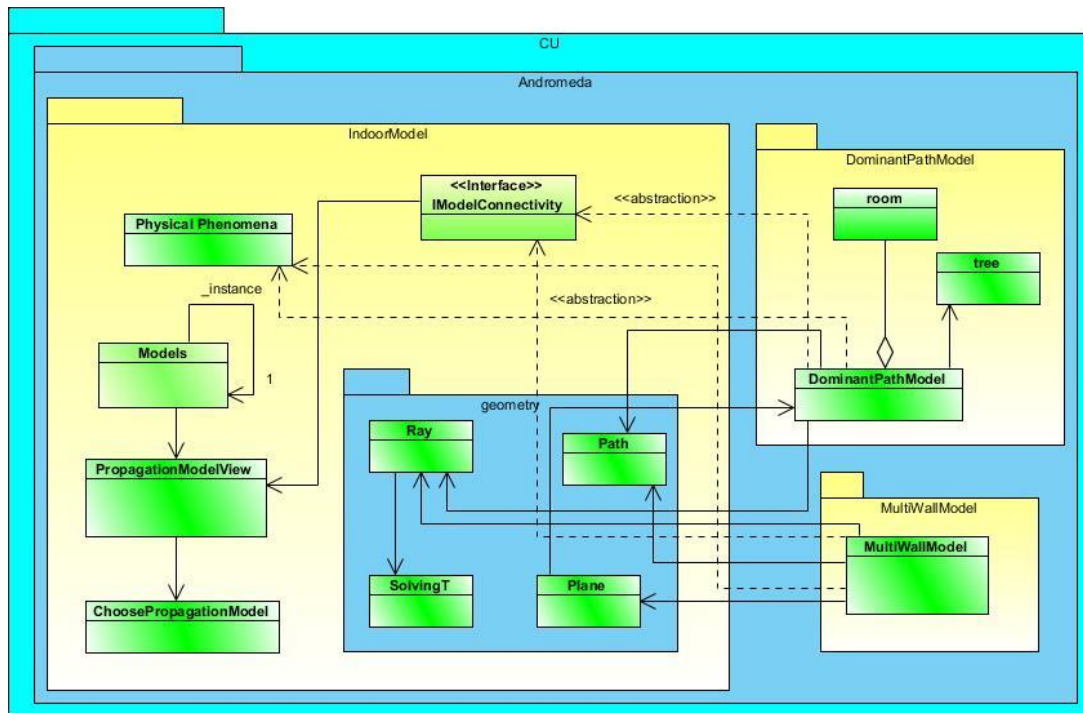
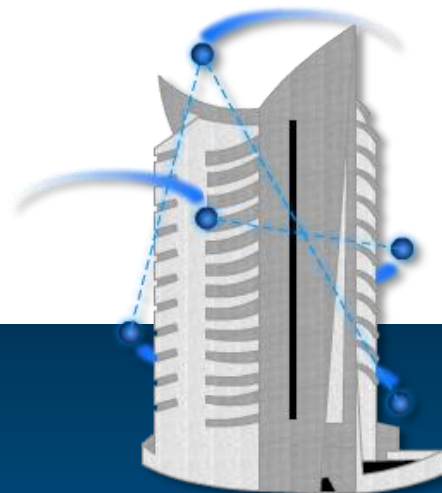


Figura 8. Diagrama de clases

## 2.7 Conclusiones parciales

Las Historias de Usuario descritas permitieron identificar las funcionalidades y características de la propuesta de solución. La fase de Planificación dotó de organización y proyección al equipo de desarrollo, lo cual se evidenció en el Plan de Entregas diseñado. Los artefactos modelados durante el Diseño serán de gran utilidad en la fase de Implementación y en el mantenimiento futuro de la herramienta Andr6meda. La descripción de la solución propuesta en el presente capítulo facilitará la implementación de los modelos de propagación seleccionados.



Andr6meda  
Andr6meda

# Capítulo 3

## Implementación y Pruebas

### 3 Implementación y pruebas

XP no propone concisamente los artefactos a utilizar en la implementación de una solución que utilice dicha metodología. Deja en manos del equipo de desarrollo, dependiendo de sus capacidades de comunicación, la decisión de utilizar tantos tipos de diagramas UML como crea posible y así, facilitar el proceso de desarrollo. En este capítulo se presenta la descripción de la arquitectura, los diagramas UML utilizados y las Tareas de Ingeniería generados por cada historia de usuario y el proceso de pruebas.

#### 3.1 Descripción de la arquitectura

Para el desarrollo de la solución en la arquitectura de la herramienta (29) se seleccionó la Plataforma de NetBeans (NetBeans Platform<sup>2</sup>), debido a que su uso permite crear aplicaciones modulares. Haciendo uso del IDE NetBeans, el cual está desarrollado sobre esta plataforma, se facilita considerablemente el desarrollo del tipo de aplicaciones como la que se desea implementar, favoreciendo notablemente las soluciones creadas con dicha tecnología, debido a que no se necesita modificar directamente el núcleo de la aplicación para agregar nuevos módulos, lo cual la hace más flexible.

##### 3.1.1 Plataforma Andrómeda

Según se definió en la arquitectura (29): “Andr6meda es el subsistema que hace referencia al núcleo del sistema final a desarrollar. El mismo es una instancia de la Plataforma de NetBeans, por lo cual está provisto del conjunto de APIs y clases que provee la plataforma. A su vez es el encargado de la gestión de carga de otros módulos. Inicia el menú de la aplicación y carga la biblioteca de sensores a ser usados en el proyecto.”

La Plataforma de NetBeans permite las relaciones entre módulos estableciendo dependencias entre ellos. Además, posee un mecanismo definido por la plataforma llamado *Lookup*, el cual posibilita que cada módulo intercambie información con los restantes.

---

<sup>2</sup>[netbeans.org/kb/trails/platform.html](http://netbeans.org/kb/trails/platform.html)



### 3.1.2 Módulos del sistema

A la herramienta Andrómeda se le adicionarán tres módulos donde se implementarán los modelos de propagación: **MultiWallModel**, **DominantPathModel** e **IndoorModel**. A continuación se describen cada uno de estos.

- **IndoorModel:** Módulo encargado de contener las clases interfaces que serán implementadas en los restantes módulos, además de contener las clases que gestionarán geoméricamente el modelo 3D.
- **MultiWallModel:** Módulo que implementa el modelo de propagación en interiores Multi-Wall, este contiene solo la clase MultiWallModel que se encarga de buscar todas las paredes y techos que intercepta la señal en su camino directo desde el transmisor (Tx) al receptor (Rx) y con esto calcula la pérdida de trayecto en dB.
- **DominantPathModel:** Módulo que implementa el modelo de propagación en interiores Trayecto Dominante, contiene la clase principal DominantPathModel que se encarga de calcular la pérdida de trayecto por el mejor camino que transita la señal de RF, además contiene el resto de las clases que permiten dividir un modelo 3D en habitaciones para lograr encontrar el trayecto dominante y calcular los diferentes fenómenos físicos.

### 3.1.3 Paquetes del sistema

Siguiendo la estructura convencional de aplicaciones Java para lograr una mejor organización del código, cada módulo de la aplicación se desarrolló centrado en la arquitectura de la plataforma Andrómeda.

La estructura de paquetes en Java se organiza de la siguiente manera: `paqueteprincipal.subpaquete`, por lo que quedan organizados los paquetes que conforman los módulos nuevos como:

- `cu.andromeda.IndoorModel`
- `cu.andromeda.MultiWallModel`
- `cu.andromeda.DominantPathModel`

Para más detalles ver la Figura 9.

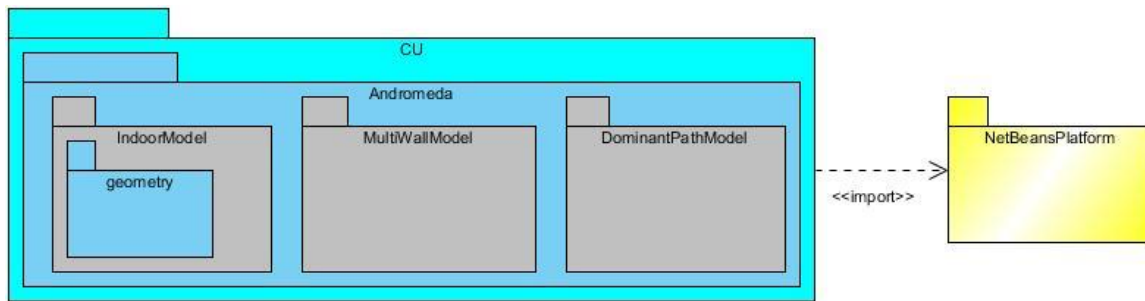


Figura 9. Diagrama de paquetes

A continuación se describen cada uno de los paquetes:

- **cu.andromeda.IndoorModel:** Este paquete almacena las clases e interfaces principales que implementarán los distintos modelos de propagación de la señal de RF.
- **cu.andromeda.IndoorModel.geometry:** Este paquete tiene las clases que definen los elementos geométricos necesarios para implementar los distintos modelos de propagación.
- **cu.andromeda.MultiWallModel:** Este paquete contendrá las clases necesarias para implementar el modelo de propagación Multi-Wall.
- **cu.andromeda.DominatPathModel:** Este paquete contendrá las clases necesarias para implementar el modelo de propagación Trayecto Dominante.

### 3.1.4 Diagrama de componentes

Los Diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación. (30)

En la Figura 10 se muestra el Diagrama de componentes correspondiente a los módulos adicionales.

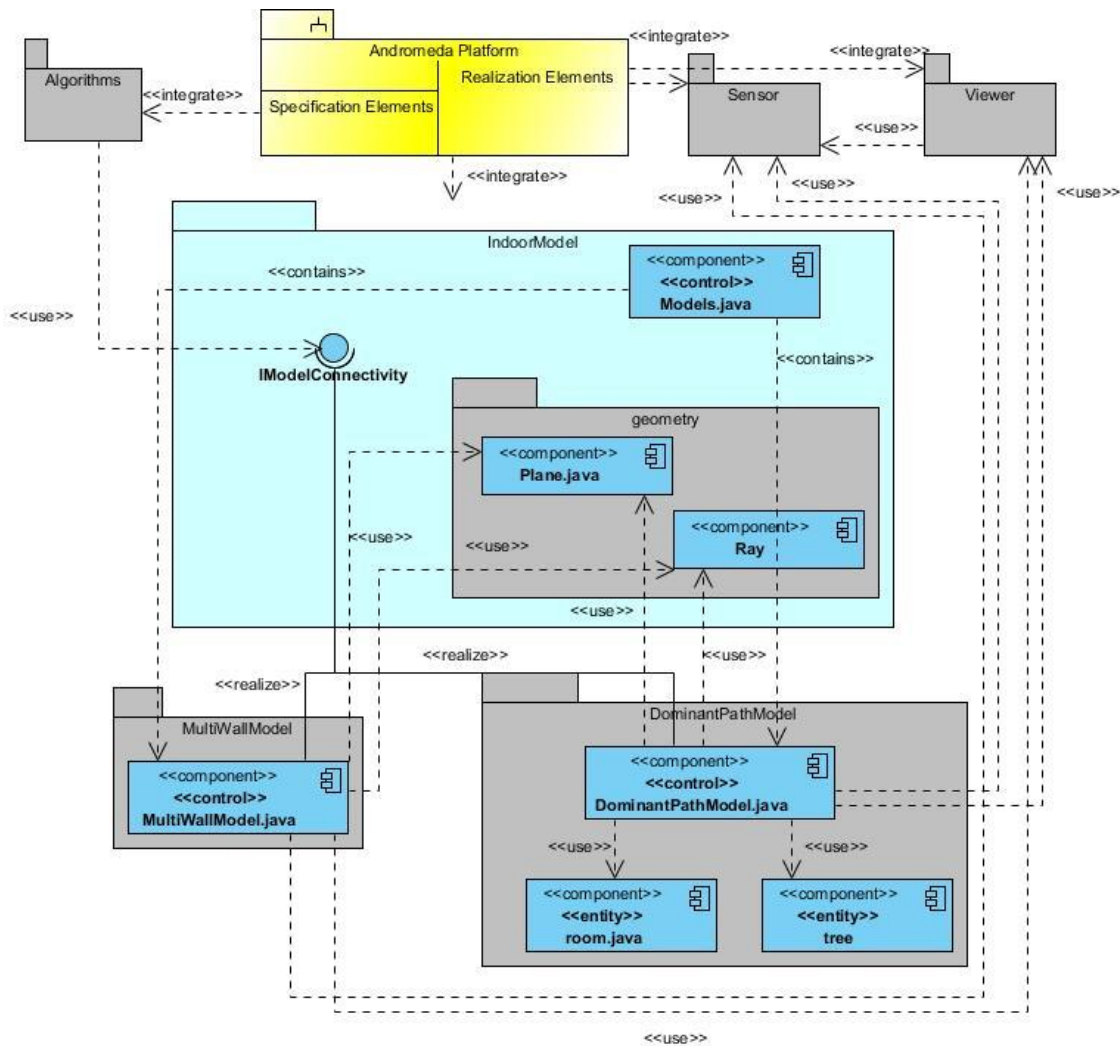


Figura 10. Diagrama de componentes correspondiente de los módulo IndoorModel, MultiWallModel y DominantPathModel.

El módulo **IndoorModel** contendrá los componentes principales que se encargarán de la gestión de los modelos de propagación que empleará la herramienta Andrómeda para su funcionamiento. Cada modelo de propagación de la señal de RF debe implementar la interfaz **IModelConnectivity**, la cual consta de una función encargada de predecir la conectividad entre dos dispositivos. Esta función será utilizada por el subsistema **Algorithms** para comprobar la conectividad entre dos dispositivos.

## 3.2 Implementación

Como buena práctica de desarrollo, XP propone una arquitectura lo más flexible posible para comenzar el desarrollo, evitando grandes cambios para próximas iteraciones provocados por las modificaciones que el cliente por lo general propone.

### 3.2.1 Primera iteración

En la primera iteración se pretende desarrollar las primeras tres HU, con el objetivo de obtener una primera versión funcional del producto.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 1 “Visualizar los modelos de propagación implementados en la herramienta.”

1. Tarea N° 1: Estudiar el concepto de *Lookup* en la guía definitiva de la Plataforma de Netbeans 7.
2. Tarea N° 2: Estudiar la arquitectura de la herramienta Andrómeda.
3. Tarea N° 3: Implementar la clase *PropagationModelView*.

Historia de usuario N° 2 “Permitir la selección del modelo de propagación a usarse en la simulación”

4. Tarea N° 4: Diseñar Interfaz de selección de los modelos.
5. Tarea N° 5: Implementar la clase *ChoosePropagationModel*.

Historia de usuario N°3 “Calcular conectividad entre dos dispositivos inalámbricos usando Multi-Wall”

6. Tarea N° 6: Estudiar la librería OpenIFCTool.
7. Tarea N° 7: Estudiar la librería Java 3D.
8. Tarea N° 8: Implementar las clases *Plane*, *Ray*, *MultiWallModel*, *SolvingT*.
9. Tarea N° 9: Implementar la detección de colisiones.

Tabla 10. Tarea 1: Estudiar el concepto de *Lookup* en la guía definitiva de la Plataforma de Netbeans 7.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Estudiar el concepto de <i>Lookup</i> en la guía definitiva de la Plataforma de Netbeans 7.	
Tipo de tarea: Preparación	Estimación: 3 días

<b>Fecha de inicio:</b> 10 de febrero de 2014	<b>Fecha fin:</b> 12 de febrero de 2014
<b>Programador responsable:</b> Lianet Cylwik López y Ariel Llanes Díaz	
<b>Descripción:</b> Estudiar el concepto de <i>Lookup</i> pues este se utiliza dentro de la plataforma NetBeans para la gestión de instancias de objetos y permite a los módulos comunicarse entre sí, siendo esto necesario para visualizar dinámicamente los modelos implementados en distintos módulos.	

Tabla 11. Tarea 2: Estudio de la arquitectura de la herramienta Andrómeda.

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Estudio de la arquitectura de la herramienta Andrómeda.	
<b>Tipo de tarea:</b> Preparación	<b>Estimación:</b> 6 días
<b>Fecha de inicio:</b> 13 de febrero de 2014	<b>Fecha fin:</b> 20 de febrero de 2014
<b>Programador responsable:</b> Lianet Cylwik López y Ariel Llanes Díaz	
<b>Descripción:</b> Estudiar la arquitectura de la herramienta Andrómeda con el objetivo de saber la organización funcional de la misma y las relaciones entre sus módulos y componentes.	

Tabla 12. Tarea 3: Implementar la clase *PropagationModelView*.

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Implementar la clase <i>PropagationModelView</i> .	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 21 de febrero de 2014	<b>Fecha fin:</b> 21 de febrero de 2014
<b>Programador responsable:</b> Ariel Llanes Díaz y Lianet Cylwik López.	
<b>Descripción:</b> La clase <i>PropagationModelView</i> , muestra a través del servicio <i>Lookup</i> los modelos implementados.	
<b>Paquete:</b> <code>cu.andromeda.IndoorModel</code>	

Tabla 13. Tarea 4: Diseñar Interfaz de selección de los modelos.

Tarea	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 2
<b>Nombre de la tarea:</b> Diseñar Interfaz de selección de los modelos.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 1 día



<b>Fecha de inicio:</b> 24 de febrero de 2014	<b>Fecha fin:</b> 24 de febrero de 2014
<b>Programador responsable:</b> Lianet Cylwik L6pez y Ariel Llanes D6az	
<b>Descripci6n:</b> Dise1nar la interfaces de usuario de selecci6n de los modelos.	
<b>Paquete:</b> cu.andromeda.IndoorModel	

Tabla 14. Tarea 5: Implementar la clase *ChoosePropagationModel*.

Tarea	
<b>N6mero de tarea:</b> 5	<b>N6mero de HU:</b> 2
<b>Nombre de la tarea:</b> Implementar la clase <i>ChoosePropagationModel</i> .	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimaci6n:</b> 1 d6a
<b>Fecha de inicio:</b> 25 de febrero de 2014	<b>Fecha fin:</b> 25 de febrero de 2014
<b>Programador responsable:</b> Ariel Llanes D6az y Lianet Cylwik L6pez	
<b>Descripci6n:</b> Se crea la clase <i>ChoosePropagationModel</i> para colocar el sub men6 "Propagation Model" dentro del men6 <i>Tool</i> con el objetivo de activar y visualizar la ventana de selecci6n del modelo.	
<b>Paquete:</b> cu.andromeda.IndoorModel	

Tabla 15. Tarea 6: Estudiar la librer6a OpenIFCTool.

Tarea	
<b>N6mero de tarea:</b> 6	<b>N6mero de HU:</b> 3
<b>Nombre de la tarea:</b> Estudiar la librer6a OpenIFCTool	
<b>Tipo de tarea:</b> Preparaci6n	<b>Estimaci6n:</b> 3 d6as
<b>Fecha de inicio:</b> 26 de febrero de 2014	<b>Fecha fin:</b> 28 de febrero de 2014
<b>Programador responsable:</b> Lianet Cylwik L6pez y Ariel Llanes D6az	
<b>Descripci6n:</b> Estudiar la librer6a OpenIFCTool para saber c6mo trabajar con ella y para poder acceder a los elementos del modelo IFC	

Tabla 16. Tarea 7: Estudiar la librer6a Java 3D.

Tarea	
<b>N6mero de tarea:</b> 7	<b>N6mero de HU:</b> 2
<b>Nombre de la tarea:</b> Estudiar la librer6a Java 3D	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimaci6n:</b> 3 d6as
<b>Fecha de inicio:</b> 3 de marzo de 2014	<b>Fecha fin:</b> 5 de marzo de 2014

<b>Programador responsable:</b> Ariel Llanes Díaz y Lianet Cylwik López
<b>Descripción:</b> Estudiar la librería Java 3D para poder trabajar geoméricamente con los elementos del modelo IFC.

Tabla 17. Tarea 8: Implementar las clases *Plane*, *Ray*, *MultiWallModel*, *SolvingT*.

Tarea	
<b>Número de tarea:</b> 8	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Implementar las clases <i>Plane</i> , <i>Ray</i> , <i>MultiWallModel</i> , <i>SolvingT</i> .	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 6 días
<b>Fecha de inicio:</b> 6 de marzo de 2014	<b>Fecha fin:</b> 13 de marzo de 2014
<b>Programador responsable:</b> Ariel Llanes Díaz y Lianet Cylwik López	
<b>Descripción:</b> Se implementa la clase <i>Plane</i> donde se definen las ecuaciones de los planos presentes en la edificación. La clase <i>Ray</i> es para definir la ecuación de la recta que se forma entre el transmisor y el receptor. <i>MultiWallModel</i> es la clase que implementa la clase interfaz <i>IModelConnectivity</i> , aquí se calcula la conectividad entre los dispositivos y <i>SolvingT</i> se utiliza para despejar la ecuación de la recta.	
<b>Paquete:</b> cu.andromeda.IndoorModel	

Tabla 18. Tarea 9: Implementar la detección de colisiones.

Tarea	
<b>Número de tarea:</b> 9	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Implementar la detección de colisiones.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 5 días
<b>Fecha de inicio:</b> 14 de marzo de 2014	<b>Fecha fin:</b> 20 de marzo de 2014
<b>Programador responsable:</b> Ariel Llanes Díaz y Lianet Cylwik López	
<b>Descripción:</b> Implementar la detección de colisiones con el objetivo de ver cuántos planos se intersectan con la recta que se forma entre el dispositivo transmisor y el receptor.	

### 3.2.2 Segunda iteración

En la segunda iteración se pretende desarrollar la cuarta HU, con el objetivo de obtener una segunda versión funcional del producto.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 4 “Calcular conectividad entre dos dispositivos inalámbricos usando el modelo Trayecto Dominante.”

10. Tarea N° 10: Estudiar la obtención del trayecto dominante en interiores.
11. Tarea N° 11: Implementar las clases *Tree* y *Room*.
12. Tarea N° 12: Analizar y generar el árbol con las paredes del escenario.
13. Tarea N° 13: Implementar la búsqueda del mejor camino a través de las paredes.
14. Tarea N° 14: Implementar el cálculo de la pérdida de trayectoria en espacio libre.
15. Tarea N° 15: Estudiar los fenómenos físicos: Reflexión, Difracción y Transmisión.
16. Tarea N° 16: Estudiar y almacenar los materiales dieléctricos de diferentes elementos.
17. Tarea N° 17: Implementar el coeficiente de reflexión.
18. Tarea N° 18: Determinar la ganancia causada por guía de ondas.
19. Tarea N° 19: Implementar el coeficiente de transmisión.
20. Tarea N° 20: Determinar la pérdida por cada penetración a través de las paredes.
21. Tarea N° 21: Implementar la pérdida de iteraciones acumuladas.
22. Tarea N° 22: Implementar la clase *DominantPathModel*.

La descripción de cada una de estas tareas se encuentra en el [Anexo 2](#).

### 3.3 Cálculo de la pérdida de trayectoria

Siempre que se propague la señal de RF desde un dispositivo a otro, tanto la trayectoria en el espacio libre como los objetos (mesas, sillas, etc.) presentes dentro de una edificación, causan atenuación en la propagación de la onda electromagnética. Por tanto para lograr predecir con precisión la conectividad entre dispositivos inalámbricos usando algún modelo de propagación es necesario calcular la **pérdida de trayectoria**.

En el modelo Multi-Wall se tienen que conocer la **propagación en el espacio libre** y la cantidad de obstáculos presentes entre el transmisor y receptor, calculados con la **detección de colisiones**.

Para calcular la pérdida de trayectoria en el modelo Trayecto Dominante se tienen que conocer los coeficientes de intersección para la transmisión, reflexión y difracción. Además de la ganancia causada por guiado de ondas para cada pixel a lo largo de una trayectoria de propagación. Para una mejor comprensión ver [Anexo 3](#).

### 3.3.1 Propagaci3n en el espacio libre

Para todos los tipos de propagaci3n de ondas de radio, la potencia de la se1al disminuye con la distancia recorrida. Esta forma de atenuaci3n se conoce como p3rdidas en el espacio libre. En el modelo Multi-Wall la p3rdida en el espacio libre entre el transmisor y receptor ( $L_{FS}$ ) se calcula de la siguiente manera:

$$L_{FS} = 20 \log_{10} \left( 4\pi d / \lambda \right)$$

**Ecuaci3n 1.11** (12)

Donde:

- $d$  : Distancia de separaci3n en metros.
- $\lambda$  : Longitud de onda del transmisor.

### 3.3.2 Detecci3n de colisiones

La detecci3n de colisiones es un t3rmino utilizado en la programaci3n para saber si dos objetos han colisionado. Esto se cumple entre todas las entidades geom3tricas (33). En este caso se analizan las colisiones entre un **segmento de recta** y un **plano**.

#### Ecuaci3n de la recta

Sea  $P(x_0, y_0, z_0)$  un punto de la recta  $r$  y  $\vec{u}$  su vector director, el vector  $\overrightarrow{PX}$  tiene igual direcci3n que  $\vec{u}$ , luego es igual a  $\vec{u}$  multiplicado por un escalar. Obteniendo la ecuaci3n vectorial de la recta. (34)

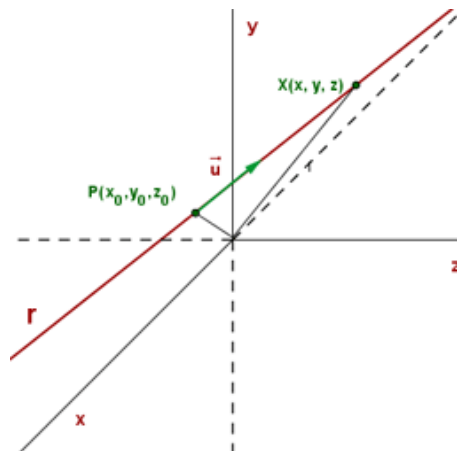


Figura 11. Representaci3n de una recta. (34)

$$(x, y, z) = (x_0, y_0, z_0) + t (U_1, U_2, U_3)$$

Operando en la ecuación vectorial de la recta se llega a la igualdad:

$$(x, y, z) = (x_0 + t * U_1, y_0 + t * U_2, z_0 + t * U_3)$$

Igualando coordenadas se llega a la ecuación paramétrica de la recta:

$$\begin{cases} x = x_0 + t * U_1 \\ y = y_0 + t * U_2 \\ z = z_0 + t * U_3 \end{cases}$$

### Ecuación del plano

Un plano queda determinado por un punto  $P$  y un par de vectores con distinta dirección.

(34)

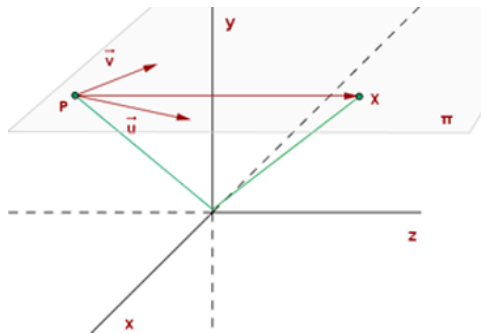


Figura 12. Representación de un plano. (34)

Para que el punto  $P$  pertenezca al plano  $\pi$  el vector  $\overrightarrow{PX}$  tiene que ser coplanario con  $\vec{u}$  y  $\vec{v}$ . Luego se puede expresar como combinación lineal de ambos utilizando los escalares  $\lambda$  y  $\mu$ .

$$\overrightarrow{PX} = \lambda \vec{u} + \mu \vec{v}$$

$$(x - x_0, y - y_0, z - z_0) = \lambda(U_1, U_2, U_3) + \mu(V_1, V_2, V_3)$$

$$(x, y, z) = (x_0, y_0, z_0) + \lambda(U_1, U_2, U_3) + \mu(V_1, V_2, V_3)$$

Partiendo de las ecuaciones anteriores, un punto está en el plano  $\pi$  si tiene solución el sistema:

$$\begin{aligned}x - x_0 &= U_1\lambda + V_1\mu \\y - y_0 &= U_2\lambda + V_2\mu \\z - z_0 &= U_3\lambda + V_3\mu\end{aligned}$$

Este sistema tiene que ser compatible determinado en las inc6gnitas  $\lambda$  y  $\mu$ . Por tanto el determinante de la matriz ampliada del sistema con la columna de los t6rminos independientes tiene que ser igual a cero. (34)

$$\begin{vmatrix}x - x_0 & U_1 & V_1 \\y - y_0 & U_2 & V_2 \\z - z_0 & U_3 & V_3\end{vmatrix} = 0$$

Se desarrolla el determinante.

$$\begin{vmatrix}U_2 & V_2 \\U_3 & V_3\end{vmatrix}(x - x_0) - \begin{vmatrix}U_1 & V_1 \\U_3 & V_3\end{vmatrix}(y - y_0) - \begin{vmatrix}U_1 & V_1 \\U_2 & V_2\end{vmatrix}(z - z_0) = 0$$

Se dan los valores:

$$A = \begin{vmatrix}U_2 & V_2 \\U_3 & V_3\end{vmatrix} \quad B = -\begin{vmatrix}U_1 & V_1 \\U_3 & V_3\end{vmatrix} \quad C = \begin{vmatrix}U_1 & V_1 \\U_2 & V_2\end{vmatrix}$$

Se sustituye:

$$A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$$

Se realizan las operaciones y se le da a  $D$  el valor:

$$D = -Ax_0 - By_0 - Cz_0$$

Se obtiene la ecuaci6n general de plano:

$$Ax + By + Cz + D = 0$$

El vector normal a este plano, es decir, perpendicular al mismo es:  $\vec{n} = (A, B, C)$ .

### Intersecci6n rayo - plano

Despu6s de tener la ecuaci6n del plano  $Ax + By + Cz + D = 0$  se sustituyen  $x, y, z$  por los valores de la recta:

$$\begin{cases}x = x_0 + t * U_1 \\y = y_0 + t * U_2 \\z = z_0 + t * U_3\end{cases}$$

Quedando la ecuaci6n en base a  $t$  e igualada a cero. Si  $t$  da un n6mero entre 0 y 1 entonces la recta corta al plano.

### 3.3.3 Fenómenos físicos y cálculo de los coeficientes

Los modelos de propagación deterministas se apoyan en principios físicos en cuanto a la propagación de onda y en fenómenos paralelos como son la reflexión, la transmisión y la difracción. A continuación se muestra cada uno de estos fenómenos y la forma de calcular sus coeficientes.

#### Reflexión

Cuando un rayo se propaga por el aire, solo se consideran las pérdidas del espacio libre. Pero cuando en la trayectoria se producen reflexiones (ver [Anexo 4](#)), las pérdidas que se presentan en las intersecciones con otros medios se tienen que agregar a las pérdidas del espacio libre. Este campo electromagnético restante se determina calculando los coeficientes de reflexión que son dependientes del **ángulo incidente** con el medio, de las **propiedades dieléctricas** del medio y del **espesor de la superficie**. El coeficiente de reflexión es diferente dependiendo de la polarización de la onda de llegada. (7)

La geometría de la reflexión es de cierta manera bastante simple. Usando la *Ley de Snell* (7) de la reflexión para un caso general se demuestra que el **ángulo de incidencia** rasante ( $\Psi$ ) del rayo de llegada es igual que el ángulo del rayo de salida.

El **ángulo de incidencia** se puede calcular fácilmente usando la normal a la superficie  $n$ .

$$\Psi = \sin^{-1} \left( \frac{V_i \cdot n}{|V_i|} \right)$$

Donde  $V_i$  es un vector en la dirección de propagación,  $|V_i|$  está normalizado y  $\cdot$  denota el producto escalar entre vectores.

#### Coeficiente de reflexión

El caso más simple se produce cuando la superficie intersecada es un conductor perfecto. Esto significa que el medio refleja perfectamente la onda electromagnética de llegada. En este caso, el coeficiente de reflexión es  $-1$  para polarización horizontal y  $+1$  para el caso de polarización vertical.

Una fórmula ampliamente utilizada para calcular el coeficiente de reflexión es la fórmula de *Fresnel*. En sus cálculos, la fórmula de *Fresnel* considera la **permitividad compleja**

( $\epsilon^*$ ) y el **ángulo de incidencia** ( $\Psi$ ). Para la polarización horizontal, el coeficiente se calcula mediante: (7)

$$r_h = \frac{\sin(\Psi) - \sqrt{\epsilon^* - \cos(\Psi)^2}}{\sin(\Psi) + \sqrt{\epsilon^* - \cos(\Psi)^2}}$$

Y para la polarización vertical,

$$r_v = \frac{\epsilon^* \sin(\Psi) - \sqrt{\epsilon^* - \cos(\Psi)^2}}{\epsilon^* \sin(\Psi) + \sqrt{\epsilon^* - \cos(\Psi)^2}}$$

Para hacer el modelo más realista se puede introducir el **espesor de la superficie** intersecada. Estas superficies pueden estar formadas por varias capas con materiales y espesores diferentes como se muestra a continuación:

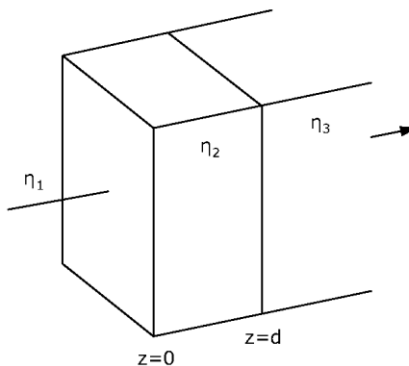


Figura 13. Ejemplo de 3 medios diferentes.  $n_1$  y  $n_3$  se definen como medios infinitamente anchos

La fórmula para calcular el coeficiente de reflexión ( $\rho$ ) de acuerdo a la Figura 13 es:

$$\rho_h = \rho_v = \frac{r_0 + r_d e^{-2kd}}{1 + r_0 r_d e^{-2kd}}$$

Donde  $r_0$  y  $r_d$  son los coeficientes de *Fresnel* cuando  $z = 0$  y  $z = d$  (espesor) respectivamente.  $k$  es el número de onda relativo del material intersecado, dado por:

$$k = \frac{2\pi f \sqrt{\epsilon^* - \sin^2 \Psi}}{c}$$

Donde  $f$  es la frecuencia del transmisor y  $c$  la velocidad de la luz.



## Transmisión

Las intersecciones de transmisión (ver [Anexo 5](#)) se consideran de manera similar a las intersecciones de reflexión y se consideran los mismos parámetros. Igual que para la reflexión, los coeficientes de transmisión se calculan por separado para polarización horizontal y vertical. Debido a que la permitividad en los materiales es más alta en comparación con la permitividad en el espacio libre, la velocidad de las ondas electromagnéticas es más baja y crea retardos. Sin embargo, los retardos en las paredes son tan pequeños que pueden ser despreciados. (7)

## Coeficiente de transmisión

Los coeficientes de *Fresnel* para la transmisión son, para una polarización horizontal:

$$t_h = \frac{2 \sin(\Psi)}{\sin(\Psi) + \sqrt{\epsilon^* - \cos(\Psi)^2}}$$

y para una polarización vertical:

$$t_v = \frac{2 \sin(\Psi) \sqrt{\epsilon^*}}{\epsilon^* \sin(\Psi) + \sqrt{\epsilon^* - \cos(\Psi)^2}}$$

Al tomar un espesor considerable, el procedimiento es igual que para el caso de la reflexión. El coeficiente de transmisión se calcula según la Figura 13 mediante:

$$\rho_h = \rho_v = \frac{t_0 t_d e^{-kd}}{1 + r_0 r_d e^{-2kd}}$$

Donde  $t_0, t_d$  son los coeficientes de transmisión de *Fresnel* cuando  $z=0$  y  $z=d$  respectivamente.

Para el caso particular en el que  $\eta_1 = \eta_3$  la fórmula se reduce a la siguiente expresión:

$$\rho_h = \rho_v = \frac{(1 - r_0^2) e^{-kd}}{1 - r_0^2 e^{-2kd}}$$

Una observación interesante de la fórmula resultante es que el coeficiente de transmisión se puede calcular usando el coeficiente de *Fresnel* de reflexión en vez del de transmisión.

## Difracción

Numerosos trayectos de propagación comprenden un obstáculo o varios obstáculos separados, e interesa calcular la pérdida que estos introducen. Para realizar el cálculo hay que idealizar la forma de tales obstáculos, considerándola bien como de arista de grosor despreciable o como de arista en filo de cuchillo gruesa y lisa, cuyo radio de curvatura en la cima está bien definido.

### Obstáculo único en arista en filo de cuchillo:

En este caso (ver [Anexo 6](#)), todos los parámetros geométricos se agrupan en un solo parámetro sin dimensión, que normalmente se designa por  $v$  y que puede tomar distintas formas equivalentes según los parámetros geométricos elegidos: (35)

$$v = h \sqrt{\frac{2}{\lambda} \left( \frac{1}{d_1} + \frac{1}{d_2} \right)}$$

$$v = \theta \sqrt{\frac{2}{\lambda \left( \frac{1}{d_1} + \frac{1}{d_2} \right)}}$$

$$v = \sqrt{\frac{2h\theta}{\lambda}} (v \text{ tiene el mismo signo que } h \text{ y } \theta)$$

$$v = \sqrt{\frac{2d}{\lambda} * \alpha_1 * \alpha_2} (v \text{ tiene el mismo signo que } \alpha_1 \text{ y } \alpha_2)$$

Donde:

$h$  : Altura de la cima del obstáculo sobre la recta que une los dos extremos del trayecto.

Si la cima queda por debajo de esa línea  $h$  es negativa.

$d_1, d_2$ : Distancias desde los dos extremos del trayecto a la cima del obstáculo.

$d$ : Longitud del trayecto.

$\lambda$  : Longitud de onda del transmisor.

$\theta$ : Ángulo de difracción (rad); tiene el mismo signo que  $h$ . Se supone que el ángulo  $\theta$  es inferior a unos 0,2 rad, o sea, aproximadamente  $12^\circ$ .

$\alpha_1, \alpha_2$ : Ángulos bajo los que, a partir de un extremo, se ven la cima del obstáculo y el extremo opuesto; tienen el mismo signo que  $h$  en las ecuaciones anteriores.

La pérdida (dB) causada por la presencia del obstáculo, en función de  $v$ . Un valor aproximado puede obtenerse de la expresión:

$$J(v) = 6.9 + 20\log\left(\sqrt{(v - 0.1)^2 + 1} + v - 0.1\right)$$

### Guiado de Onda

El efecto de guiado de onda (ver Figura 14) para cada píxel a lo largo de una trayectoria de propagación va a depender de la orientación de las paredes, de la pérdida por reflexión de las paredes y de la distancia entre las paredes y la ruta. (11)

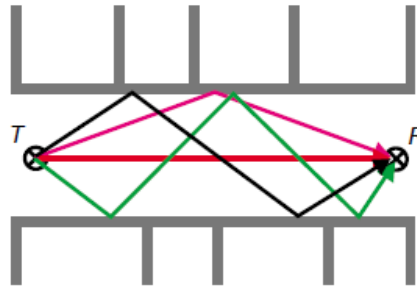


Figura 14. Efecto de guiado de onda

La Figura 15 muestra el cálculo del efecto de guiado de onda de una sola pared.

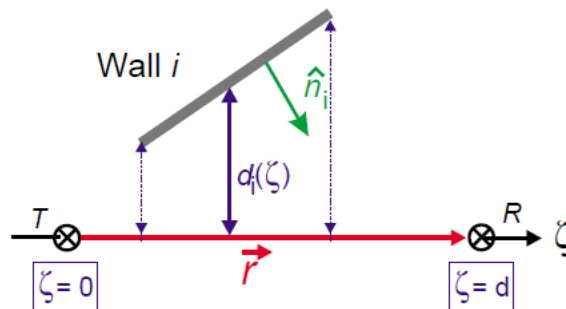


Figura 15. Determinación del efecto de guiado de onda.

El factor ( $w_i$ ) representa el guiado de onda de la pared  $i$ ; para obtener el mismo se combinan el parámetro de orientación ( $\alpha_i$ ) y la distancia normalizada del rayo a la pared ( $d_{n_i}$ ) con la pérdida por reflexión ( $L_R$ ). A continuación se muestran las ecuaciones para este cálculo:

$$\alpha_i = \cos^{-1} \left( \frac{\hat{n}_i \vec{r}}{|\vec{r}|} \right)$$

Si  $\cos \alpha_i \leq \cos \alpha_{max}$  entonces:

$$o_i = 1 - \frac{\cos \alpha_i^2}{\cos \alpha_{max}^2}$$

$$d_{n_i}(\zeta) = 1/2 - 1/2 \tanh(d_i(\zeta) - d_{max})$$

$$w_i(\zeta) = d_{n_i} e^{((1-o_i/1.35)^{L_R/2})}$$

Si  $\cos \alpha_i \geq \cos \alpha_{max}$  entonces  $w_i(\zeta) = 0$ .

Nota:  $\alpha_{max} = 70^\circ$  y  $d_{max} = 3$  metros.

### 3.4 Pruebas

Entre las técnicas más comunes para la realización de pruebas, se encuentran las pruebas de **caja blanca** y **caja negra**, efectuando diferentes tareas en la forma de aplicarlas, ya sean unitarias, integración, aceptación o sistema, según la metodología de desarrollo utilizada.

Las técnicas de **caja blanca** están basadas en estudiar el código fuente, verificando su funcionalidad desde el punto de vista del cliente. Para comprobar el funcionamiento de los códigos que se van implementando, uno de los principales pilares de la metodología XP se basa en el Desarrollo Dirigido por Pruebas (20) o también conocido como pruebas unitarias, en inglés Test Driven Development (TDD). Estas pruebas están enfocadas en comprobar los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan realizar pruebas de las fallas que pudieran ocurrir.

TDD es un estilo de desarrollo donde el programador conforma un juego de pruebas antes de comenzar la implementación. Se escriben primero las pruebas y estas determinan el código a implementar. Para el desarrollo de las mismas se usó JUnit, el cual es un framework para automatizar las pruebas unitarias de aplicaciones Java. Evaluando así que el funcionamiento de cada uno de los métodos de las clases se comportan como se espera.

Finalmente el cliente da paso a realizar las pruebas de **caja negra** o aceptación en conjunto con un representante del equipo de desarrollo. Estas pruebas tienen una visión externa del producto de software final y se centran en analizar la funcionalidad del sistema. Su principal objetivo es verificar el cumplimiento de la Lista de Reservas del Producto desde la perspectiva del usuario final.

### 3.4.1 Pruebas experimentales

Para medir el nivel de precisión de los resultados generados por los modelos de propagación implementados, en cuanto a la pérdida de señal en dB, se realizaron pruebas experimentales en un escenario real (ala izquierda del segundo piso del Docente 5 de la UCI, ver Anexo 9). Para la realización de dichas pruebas se utilizaron dispositivos del tipo XBee-Pro S2 haciendo uso del protocolo de comunicación ZigBee (operando en la banda de frecuencia de 2.4 GHz), estos dispositivos poseen un umbral de sensibilidad de -102 dBm.

Para la realización de las pruebas se fijó el dispositivo transmisor en un punto cualquiera del escenario, a partir de este se comenzó a trasladar el dispositivo receptor (XBee-Pro S2) hacia varias direcciones, tomando la posición donde el receptor indica a través del LED RSSI (apagado si no recibe señal alguna) que no existe conexión y en donde se supone existe una pérdida de -102 dBm. Luego, se colocó la posición del transmisor y los puntos donde se pierde la conexión con el receptor en la herramienta Andrómeda y se compara el resultado con el valor supuesto. Este experimento se realizó en varias ocasiones variando la posición del transmisor y del receptor, arrojando como resultado que los modelos de propagación implementados para la herramienta proveen resultados con una exactitud aproximada de  $\pm 4$  dBm para el Trayecto Dominante y de  $\pm 10$  dBm para el Multi-Wall.

### 3.4.2 Pruebas de caja negra para la versi6n 1.0

Se realizaron dos Casos de pruebas de aceptaci6n, no se identificaron no conformidades, por lo que no quedaron Casos de pruebas pendientes.

Tabla 19. CP Visualizar los modelos de propagaci6n implementados en la herramienta.

<b>Caso de prueba de aceptaci6n</b>
<b>C6digo:</b> HU1_P1 <b>Historia de Usuario:</b> 1
<b>Nombre:</b> Visualizar los modelos de propagaci6n implementados en la herramienta.
<b>Descripci6n:</b> Prueba para la funcionalidad de visualizar los modelos de propagaci6n.
<b>Condiciones de Ejecuci6n:</b> El usuario debe haber importado el modelo IFC.
<b>Entrada/ Pasos de ejecuci6n:</b> El usuario debe dar clic en la opci6n "Tool" del men6, ah6 elige la opci6n "Propagation Model".
<b>Resultado Esperado:</b> La herramienta muestra una ventana con los modelos de propagaci6n implementados, en un tiempo menor de 3 segundos.
<b>Evaluaci6n de la Prueba:</b> Prueba satisfactoria.

Tabla 20. CP Permitir la selecci6n del modelo de propagaci6n a usarse en la simulaci6n.

<b>Caso de prueba de aceptaci6n</b>
<b>C6digo:</b> HU2_P2 <b>Historia de Usuario:</b> 2
<b>Nombre:</b> Permitir la selecci6n del modelo de propagaci6n a usarse en la simulaci6n.
<b>Descripci6n:</b> Prueba para la funcionalidad de seleccionar el modelo de propagaci6n a usarse en la simulaci6n.
<b>Condiciones de Ejecuci6n:</b> El usuario debe haber importado el modelo IFC y visualizado los modelos de propagaci6n.
<b>Entrada/ Pasos de ejecuci6n:</b> El usuario debe dar clic en la opci6n "Tools" del men6, ah6 elige la opci6n "Propagation Model". Se selecciona de la ventana que visualiza los modelos de propagaci6n implementados el deseado para la simulaci6n.
<b>Resultado Esperado:</b> Se puede seleccionar cualquier modelo desde la ventana que los visualiza.
<b>Evaluaci6n de la Prueba:</b> Prueba satisfactoria.

### 3.4.3 Pruebas de caja blanca para la versi6n 1.0

Se realizaron tres Casos de pruebas unitarias, identific6ndose dos no conformidades, las cuales proceden. Se le dio soluci6n a las no conformidades, por lo que no quedaron Casos de pruebas pendientes. Estos Casos de pruebas se describen en el [Anexo 8](#).

### 3.4.4 Pruebas de caja blanca para la versión 2.0

Se realizaron cinco Casos de pruebas unitarias, identificándose una no conformidad, la cual procede. Se le dio solución a la no conformidad, por lo que no quedaron Casos de pruebas pendientes. Estos Casos de pruebas se describen en el [Anexo 9](#).

### 3.4.5 Resultados generales de las pruebas

El siguiente gráfico muestra el resumen del resultado de las pruebas realizadas.

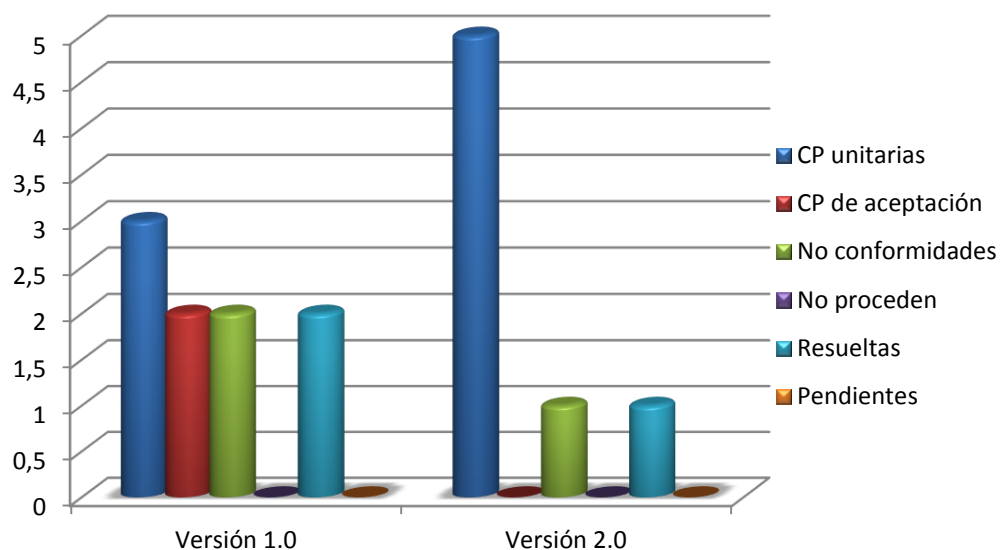


Figura 16. Resultado de las pruebas.

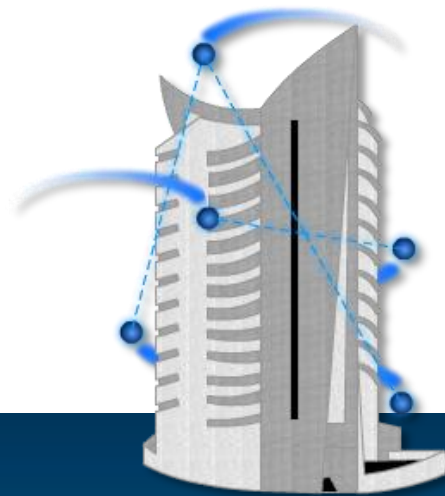
## 3.5 Conclusiones parciales.

Teniendo en cuenta que XP no establece de forma concisa los artefactos a emplear en la implementación de un sistema, en el presente capítulo se decidió en primer lugar realizar una descripción de la arquitectura sobre la cual se sustentan los módulos que se desean implementar, lo cual permitió conocer cómo estaba estructurada la plataforma Andrómeda y qué relación existía entre sus componentes. Posteriormente se utilizaron los Diagramas de Paquetes y Componentes para modelar la integración de los módulos a ser implementados con dicha plataforma. Siguiendo la metodología, se planificaron las iteraciones en las que iban a ser desarrolladas las Historias de Usuario, lo cual permitió la obtención a corto plazo de versiones funcionales de la solución propuesta, además, en



el capitulo se evidenci6 que para lograr predecir con precisi6n la conectividad entre dispositivos inal6mbricos usando alg6n modelo de propagaci6n es necesario calcular la p6rdida de trayectoria (c6lculo que fue descrito en el capitulo). Por 6ltimo, mediante el empleo de pruebas de caja negra, caja blanca y pruebas experimentales fue visible el nivel de calidad de la soluci6n propuesta.





Andrómada  
Andrómada

# Conclusiones y Recomendaciones

## Conclusiones

Al término de la presente investigación se concluye lo siguiente:

- La investigación realizada estableció que en la literatura se proponen diversas variantes de clasificación de los modelos de propagación, de forma general, se pueden clasificar en empíricos o estadísticos y teóricos o deterministas. Los primeros se basan en la extrapolación estadística de resultados a partir de mediciones realizadas sobre el terreno, mientras que los segundos se apoyan en principios físicos en cuanto a la propagación de ondas de radio y fenómenos paralelos. Además, se evidenció la existencia de varios de estos modelos y de herramientas que los utilizan en sus simulaciones, los cuales fueron descritos y analizados.
- Para el desarrollo de la propuesta de solución, siguiendo la línea base que definió la arquitectura, se usó como lenguaje de programación Java, utilizando como IDE NetBeans v7.3. Las librerías usadas para el desarrollo fueron Java 3D y OpenJFCTools. XP se empleó como metodología de desarrollo y Visual Paradigm for UML v8.0 como herramienta CASE.
- Con los módulos implementados la herramienta Andrómeda, en su primera versión, tendrá la capacidad de predecir la conectividad entre dos dispositivos inalámbricos mediante los modelos de propagación Multi-Wall o Trayecto Dominante. Esto le permitirán al diseñador de WSAN, determinar con precisión la propagación de onda entre dos dispositivos y así posicionarlos correctamente en el interior de un edificio.
- Al término del proceso de pruebas realizado a la solución propuesta, mediante el empleo de pruebas de caja blanca, de caja negra y las experimentales, se demostró la correcta implementación de sus funcionalidades, dando cumplimiento a las necesidades y expectativas del cliente.



## Recomendaciones

A partir del trabajo realizado y después de haber analizado los resultados obtenidos se sugiere a los futuros desarrolladores de la herramienta las siguientes recomendaciones que aportarían un mayor valor de uso en la misma:

- Implementar nuevos módulos con otros modelos de propagación.
- Implementar un banco de pruebas el cual incluya un número significativo de modelos de propagación y permita a la comunidad científica realizar experimentos y comparaciones según su nivel de precisión en diferentes tipos de entornos.

## Referencias Bibliográficas

1. **Buendía, Manuel , Vera, Jose y Losilla, Fernando .** Redes de Sensores y Actuadores (WSAN) en domótica. Cartagena, España : s.n., 2010.
2. **O.I.d.S.d.I. y T.d.I.I.** *Redes de sensores. Aplicaciones para control automático de edificios.* 2010.
3. **Muñoz Castejón, Rodrigo.** Interconexión de redes de sensores inalámbricos 802.15.4 en localizaciones remotas. Madrid : s.n., 2011.
4. **Libelium Comunicaciones Distribuidas S.L.** Libelium. [En línea] 2011. <http://www.libelium.com>.
5. **Ismael, Nodarse Mora Armando y Manuel, Díaz Rodríguez.** Utilización de WSANs en Sistemas de Control de Edificios. 2010.
6. **Aguiar, Jhon Jairo Padilla.** *Modelo de propagacion en interiores.* Ingenieria Telemática. 2010.
7. **Landstorfer, F. M.** *Modelo de Cobertura para Redes Inalámbricas de Interiores.* 2012.
8. **F.M.Landstorfer.** *Modelos de propagación de rayos en interior.* 2013.
9. **Michael Galetzka, Jürgen Haufe, Michael Lindig, Uwe Eichler, Peter Schneider.** *Challenges of Simulating Robust Wireless Sensor Network Applications.* Germany : s.n., 2012.
10. **Arfken, G.** *Ecuaciones De Maxwell.* España : s.n., 2013.
11. **Woffle, G.** AWE-Communications. [En línea] 2012. [Citado el: 26 de febrero de 2014.] <http://www.awe-communications.com/>.
12. **GmbH, AWE Communications.** *Indoor Scenarios.* 2012.
13. **Núñez López, Pablo Andrés.** *Estudio de la Propagación de señales Electromagneticas en edificios en banda de Wi-fi y WiMax.* 2009.
14. **Landstorfer, G. Woffle and F. M.** *Dominant paths for the field strength prediction.* University of Stuttgart. Germany : s.n.
15. **Communications, AWE.** *Herramienta de planificación radioeléctrica.* 2012.
16. **QTX de México.** Qualitrain. [En línea] 2012. <http://www.qualitrain.com.mx/objeIndirecto/javavsvbasic.htm>.
17. **Oracle Corporation.** NetBeans. [En línea] 2010. <http://netbeans.org>.
18. **Bandomo Ruíz, Javier y Torres Saavedra, Orlando.** *Visor tridimensional de entornos interiores para la herramienta Andrómeda.* La Habana : s.n., 2012.
19. **Tulke, Jan, y otros.** *Open Ifc Tools.* [En línea] 2010. [http://www.openifctools.org/Open\\_IFC\\_Tools/Features.html](http://www.openifctools.org/Open_IFC_Tools/Features.html).
20. **Beck, Kent.** *Los Procesos Ágiles: SCRUM y eXtreme Programming.* [En línea] 2012. <http://procesosagiles.com/index.html>.
21. **Letelier, Patricio y Penadés, Maria Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* Camino de Vera s/n, 46022 Valencia : s.n., 2012.
22. **Visual Paradigm International.** Visual Paradigm. [En línea] 2007. <http://www.visual-paradigm.com>.
23. **Donovan Wells, J.** *Ciclo de vida de un proyecto XP.* [En línea] 2012. [Citado el: 11 de diciembre de 2013.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
24. *¿Qué son las historias de usuario?: 7 preguntas y respuestas - La Oficina de Proyectos de Informática.* [En línea] 24 de 4 de 2013. [Citado el: 11 de 12 de 2013.] <http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>.
25. **Larman, Craig.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* s.l. : Pearson Educación, 2003. ISBN 9788420534381.
26. **L, Craig.** *UML y patrones.* s.l. : Prentice Hall., 2004.
27. **Gamma, E.** *Design Patterns: Elements of Reusable Object-Oriented Software.* . s.l. : Addison-Wesley Professional. 416, 1994.
28. **Data Desing System.** *Data Desing System.* [En línea] 2005. <http://www.dds-cad.net>.
29. [En línea] enero de 2014. <http://www.slideshare.net/nedowwhaw/diagrama-de-clases-16208245>.
30. **Molina Toledo, Pablo y Rodríguez Estévez, Henry.** *Arquitectura de la herramienta Andrómeda.* 2012.
31. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.* 6.
32. **Juan José , Jiménez Delgado.** *Detección de colisiones.* [En línea] [Citado el: 3 de 1 de 2014.] [http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Opengl/tut\\_opengl/colisiones.htm](http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Opengl/tut_opengl/colisiones.htm).
33. **Weisstein, Eric W.** *Ecuaciones de la recta en el espacio.* 2013.



34. **P.526-7, Recomendación UIT-R.** *Propagación por difracción.* 1978-1982-1992-1994-1995-1997-1999-2001.
35. **IG Network.** *IG Network.* [En línea] 2012. <http://ignetwork.net/>.
36. **Graphisoft.** *Graphisoft.* [En línea] 2009. <http://www.graphisoft.es/producto/archicad/>.
37. **Agile Aliance.** *Programación Extrema.* [En línea] 2007. <http://www.programacionextrema.org>.



## Glosario

**Modelo de Propagación:** Un modelo de propagación es un conjunto de expresiones matemáticas, diagramas y algoritmos utilizados para representar las características de la propagación de la RF en un determinado entorno.

**IDE:** Un entorno de desarrollo integrado, por sus siglas en inglés de *integrated development environment*, es un programa compuesto por un conjunto de herramientas para los programadores. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solos o pueden ser parte de aplicaciones existentes. (36)

**Radiofrecuencia:** La radiofrecuencia es en efecto un tipo de onda electromagnética que es muy semejante a la energía luminosa, y tiene la misma velocidad que la luz.

**SCE:** Sistema de Control de Edificación es la integración total de elementos y servicios del edificio en un sistema de automatización.

**Sensor:** Es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos dispositivos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas.

**Propagación:** Conjunto de fenómenos físicos que conducen a las ondas del transmisor al receptor. Esta propagación puede realizarse siguiendo diferentes fundamentos físicos, cada uno más adecuado para un rango de frecuencias de la onda a transmitir.

**Atenuación:** Se le denomina a la pérdida de potencia sufrida por la señal, sea esta acústica, eléctrica u óptica al transitar por cualquier medio de transmisión.

**IFC:** *Industry Foundation Classes (IFC)* es un formato que incluye elementos constructivos como ventanas, puertas, muros, losas, entre otras. Está basado en la idea de objetos que describen de forma total el Edificio Virtual. Estos objetos están definidos para dar continuidad al ciclo de vida del proyecto desde su concepción hasta las etapas de diseño, documentación y construcción. Este archivo está disponible a todos los



participantes de la industria de la construcción, para su uso en todos los países, aprovechando este lenguaje común para compartir elementos inteligentes del Edificio Virtual con otras disciplinas. Es un formato de uso general para BIM, siendo una especificación abierta y no está controlada por un único proveedor. (37)

**Polarización de ondas:** Es la vibración del campo eléctrico (o magnético) en una única dirección. En la onda, el campo eléctrico no tiene por qué estar siempre vibrando en un plano. Ese plano puede ir rotando o estar fijo, o ser aleatorio, como ocurre en la mayoría de las ondas. La polarización indica que el campo vibra en una única dirección.

**Permitividad:** Es una constante de proporcionalidad entre el desplazamiento eléctrico y la intensidad del campo eléctrico. En el espacio libre, la permitividad es  $\epsilon_0 = 8.854 \cdot 10^{-12}$  Faradios por metro (F/m). La permitividad se expresa a menudo como permitividad relativa,  $\epsilon_r = \epsilon / \epsilon_0$  que es la permitividad del medio ( $\epsilon$ ) en relación con la del espacio libre.

**Conductividad:** Es una medida de la facilidad con la que una corriente eléctrica fluye a través de un medio, medida en Siemens por metro (S/m), igual a la inversa de la resistencia. La conductividad del espacio libre es  $\sigma_0 = 0$ . Cuando  $\sigma_0 = \infty$ , el medio es definido como conductor perfecto.