



Trabajo de Diploma en opción al título de Ingeniero en Ciencias
Informáticas

Módulo para experimentos y comparaciones de los modelos
de propagación en la herramienta informática
“Andrómeda”.

Autor: Renides Leyva Ramírez

Tutor: Ing. Arcel Labrada Batista

La Habana, junio del 2015

Declaración de autoría

Declaro que soy el único autor del trabajo “Módulo para experimentos y comparaciones de los modelos de propagación en la herramienta informática Andrómeda” y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

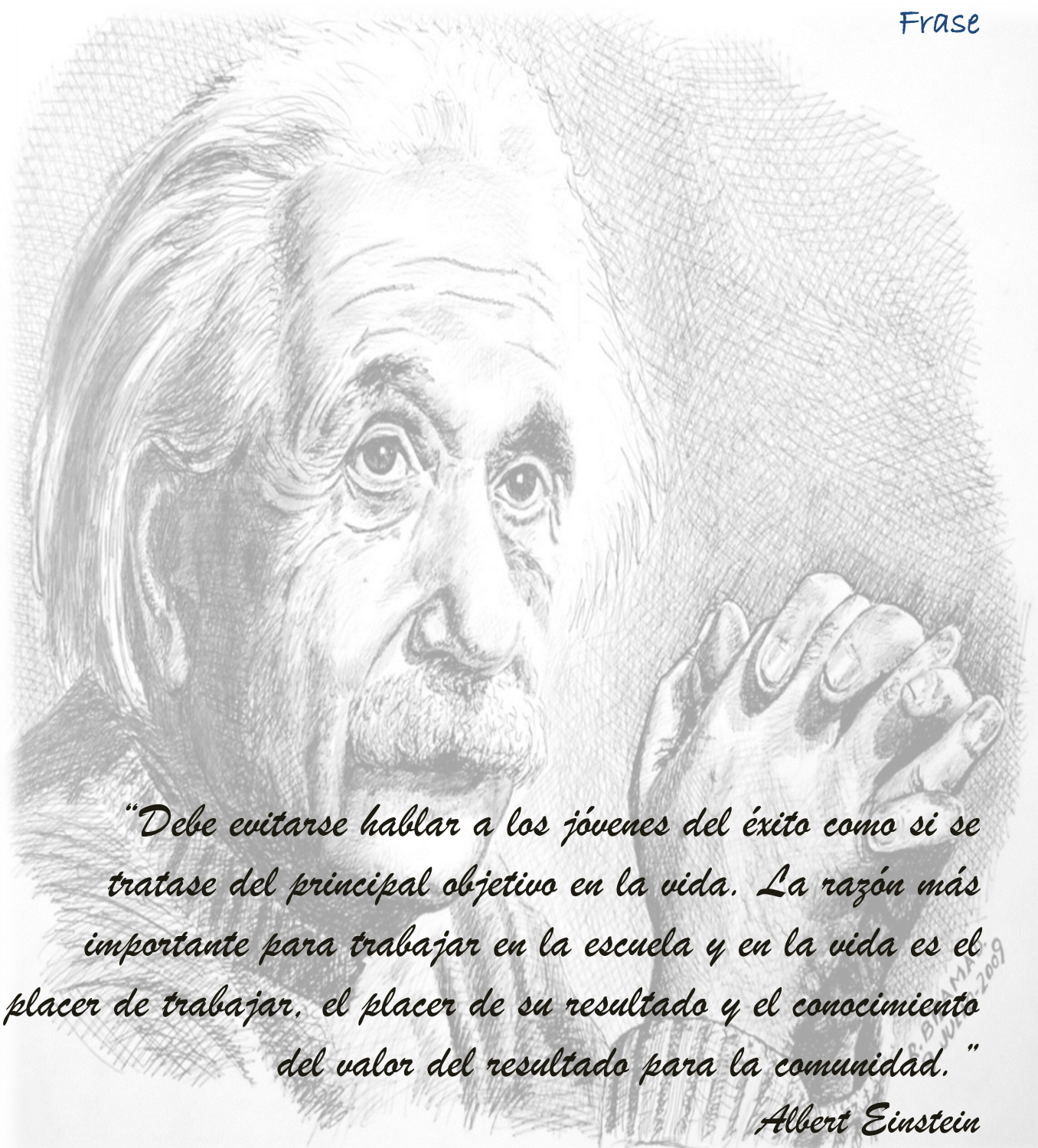
Renides Leyva Ramírez

Firma del Autor

Ing. Arcel Labrada Batista

Firma del Tutor

Frase



“Debe evitarse hablar a los jóvenes del éxito como si se tratase del principal objetivo en la vida. La razón más importante para trabajar en la escuela y en la vida es el placer de trabajar, el placer de su resultado y el conocimiento del valor del resultado para la comunidad.”

Albert Einstein



Dedicatoria

Dedico especialmente el presente trabajo de diploma a las personas que más quiero en el mundo; mis padres, porque sin su ayuda no hubiese podido alcanzar mis objetivos y lograr ser la persona que soy hoy.

A todos los que hicieron posible que mi sueño se hiciera realidad.

Agradecimientos Personales

A la Revolución Cubana y a nuestro eterno Comandante en Jefe, por haber creado un proyecto de tal magnitud; como lo es hoy la UCI y permitirme ser parte de él.

A mi segunda escuela; la FEU, por todas las oportunidades brindadas, por ayudarme a ser la persona que soy hoy. Por la familia que logré hacer gracias a ella, y la satisfacción de haber sido útil.

A mis padres, que gracias a ellos hoy se hace realidad uno de mis grandes sueños; el convertirme en Ingeniero en Ciencias Informáticas. Por señalarme el buen camino en todo momento, pero dejarme elegirlo por mí mismo.

A una persona muy especial, por aparecer en mi vida en uno de los momentos más difíciles e imborrables; a la profe Nely Morera, por el cariño, por el afecto, por brindarme tanta seguridad y tantas atenciones al mismo tiempo. Por el apoyo y la confianza.

A mis mejores amigos; Fernando, Eiler, Ariel, Alejandro, Yoelkys, Eliecer, Daldis. A ustedes que más que mis amigos han sido mis hermanos durante más de 10 años.

A mis amigos de la FEU; Lijandy, Yohanny, Sandra, Dayana, Yero, Niurka, Maikel, Julio Cesar, Angelito, Yosvany, Yoandry, Ernesto, Lester, Lily, Lucio Lázaro. Gracias a todos.

A mí aguerrido y querido amigo, compañero de tantas batallas. A Manuel Reina, por además de buen dirigente, ser buen amigo, buen compañero. Muchas gracias amigo, eres grande. Nadie podía quitarnos el derecho de seguirlo demostrando.

A un amigo muy especial; Javier Díaz Machado, por su paciencia, su dedicación, su cariño, su amistad. Por ser esa persona que tienes como meta a seguir en tu vida. Por su profesionalismo, por el amor a su trabajo, por ser siempre Javier.

A mi amigo Danier Ernesto, que aunque hoy esté lejos de aquí ha sabido acompañarme en todo momento, de brindarme sus consejos, sus criterios.

Agradecimientos Profesionales

A mi tutor, porque además de guiarme en esta última etapa, fue uno de los mejores profesores que tuve a lo largo de toda mi carrera, por su conocimiento, por sus enseñanzas, y sobre todo por haber asumido esto que hoy estamos logrando.

A mi oponente, por sus geniales consejos, sus criterios, su sabiduría. Por ayudar a perfeccionar este trabajo. Por la experiencia obtenida.

Al tribunal, por la guía certera de este proceso tan hermoso que es graduarse, por sus regaños en el momento oportuno, por la crítica cuando fue necesario, por ayudarme a ser más fuerte como estudiante, y a ser mejor profesional.



Resumen

El incremento tecnológico con que se presentan las Tecnologías de la Informática y las Comunicaciones (TIC) en la actualidad ha potenciado el desarrollo y utilización de la tecnología inalámbrica, alcanzando un auge significativo y abarcando una amplia gama de aplicaciones, llegando incluso a la automatización de edificios. El diseño del despliegue de una red de sensores inalámbricos en una edificación es una compleja tarea donde influye sustancialmente la correcta selección del modelo de propagación de la señal según las características de estas edificaciones.

El presente trabajo propone una solución basada en la implementación de una herramienta que, haciendo uso de la regresión lineal como método matemático de análisis estadístico, permitirá la obtención de gráficos comparativos de los diferentes modelos de propagación para interiores; facilitando a los diseñadores de redes el proceso de selección según las características del lugar donde se vayan a ubicar.

El desarrollo de la propuesta fue guiado por la metodología XP, UML como lenguaje de modelado, Java para la implementación y la librería JFreeChart para facilitar el graficado, también se utilizó el NetBeans como Entorno de Desarrollo Integrado y como herramienta CASE: Visual Paradigm. En la entrega final, la herramienta es capaz de comparar los modelos de propagación por la pérdida de señal entre el transmisor y el receptor. Se finaliza con la fase de pruebas, desarrollándose pruebas unitarias y de aceptación, demostrando así la correcta implementación de las funcionalidades propuestas.

Palabras clave: análisis estadístico, difracción de la señal, métodos matemáticos, modelos de propagación, pronóstico estadístico, propagación de la señal, regresión lineal.



Índice de contenido

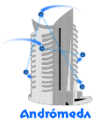
| | |
|--|-----------|
| Resumen | VI |
| Introducción | 1 |
| 1. Fundamentación Teórica | 8 |
| 1.1 Conceptos asociados a la investigación | 8 |
| 1.1.1 Análisis estadístico..... | 8 |
| 1.1.2 Canal de radio..... | 8 |
| 1.1.3 Difracción de la señal..... | 9 |
| 1.1.4 Dispersión de la señal | 9 |
| 1.1.5 Estadística | 9 |
| 1.1.6 Reflexión de la señal | 9 |
| 1.1.7 Pérdidas en espacio libre..... | 10 |
| 1.1.8 Pronóstico estadístico | 10 |
| 1.1.9 Propagación en espacio libre..... | 10 |
| 1.2 Modelos de propagación para entornos interiores (Indoor) | 11 |
| 1.2.1 Modelos de propagación empíricos | 11 |
| 1.2.2 Modelos de propagación teóricos o deterministas | 15 |
| 1.3 Métodos matemáticos empleados en los pronósticos estadísticos | 20 |
| 1.3.1 Métodos de carácter cualitativo..... | 21 |
| 1.3.2 Métodos de pronóstico causales..... | 21 |
| 1.3.3 Modelos de serie de tiempos | 26 |
| 1.4 Herramientas y Tecnologías | 28 |
| 1.4.1 Lenguaje de programación | 28 |
| 1.4.2 Entorno Integrado de Desarrollo | 29 |
| 1.4.3 Librería para el Desarrollo | 29 |
| 1.4.4 Sistema Gestor de Base de Datos | 31 |
| 1.5 Selección de la Metodología de Desarrollo de Software | 32 |
| 1.5.1 Rational Unified Process (RUP)..... | 33 |
| 1.5.2 Programación Extrema | 34 |
| 1.5.3 SCRUM | 35 |
| 1.5.4 Metodología Seleccionada | 36 |
| 1.6 Herramienta CASE | 36 |
| 1.7 Conclusiones parciales | 37 |
| 2. Descripción de la solución propuesta | 39 |
| 2.1 Introducción | 39 |
| 2.2 Propuesta de solución y representación gráfica del sistema | 39 |
| 2.2.1 Usuarios del Sistema..... | 41 |
| 2.3 Lista de Reserva del Producto | 41 |
| 2.4 Aspectos no Funcionales del Sistema | 42 |
| 2.5 Fase de Exploración | 43 |
| 2.5.1 Historias de Usuarios | 43 |



| | |
|--|-----------|
| 2.6 Fase de Planificación | 45 |
| 2.6.1 Estimación de esfuerzos por historias de usuario | 45 |
| 2.6.2 Iteraciones | 46 |
| 2.6.3 Plan de Entregas | 47 |
| 2.7 Diseño | 48 |
| 2.7.1 Patrones de diseño | 48 |
| 2.7.2 Modelo de datos..... | 50 |
| 2.7.3 Diagrama de Clases..... | 52 |
| 2.8 Conclusiones parciales | 53 |
| 3. Implementación y pruebas | 54 |
| 3.1 Introducción | 54 |
| 3.2 Descripción de la arquitectura | 54 |
| 3.2.1 Plataforma Andrómeda | 54 |
| 3.2.2 Diagrama de Componentes | 55 |
| 3.2.3 Método de Regresión Lineal simple | 56 |
| 3.2.4 Ajuste por Mínimos Cuadrados | 58 |
| 3.3 Implementación | 60 |
| 3.3.1 Primera Iteración | 60 |
| 3.3.2 Segunda Iteración | 61 |
| 3.3.3 Tercera Iteración | 62 |
| 3.4 Pruebas | 63 |
| 3.4.1 Pruebas de aceptación para la Iteración 1 | 64 |
| 3.4.2 Pruebas de aceptación para la Iteración 2 | 64 |
| 3.4.3 Pruebas de aceptación para la Iteración 3 | 64 |
| 3.4.5 Resultados generales de las pruebas..... | 65 |
| 3.5 Conclusiones parciales | 66 |
| Conclusiones | 67 |
| Recomendaciones | 68 |
| Referencias Bibliográficas | 69 |
| Anexos | 72 |
| Anexo 1. Fenómenos de difracción, dispersión y reflexión de la señal. | 72 |
| Anexo 2.Fases de desarrollo de la Metodología SCRUM | 73 |
| Anexo 3. Historias de Usuario. | 73 |
| Anexo 4. Patrones de Diseño..... | 77 |
| Anexo 5. Diagrama de dispersión luego de aplicar regresión lineal simple | 79 |
| Anexo 6. Descripción de tareas Iteración 1. | 79 |
| Anexo 7. Descripción de tareas Iteración 2. | 82 |
| Anexo 8. Descripción de tareas Iteración 3. | 85 |
| Anexo 9. Datos de muestra para aplicar la regresión lineal. | 88 |
| Anexo 10. Casos de prueba de aceptación para la Iteración 1..... | 88 |
| Anexo 11. Casos de pruebas de aceptación para la Iteración 2. | 90 |
| Anexo 12. Casos de pruebas de aceptación para la Iteración 3. | 91 |
| Anexo 13. Resultados de pruebas unitarias para Andromeda_Grapher..... | 93 |

Índice de figuras

| | |
|--|----|
| <i>Figura 1 Pérdidas de Trayecto para diferentes valores de n. (15)</i> | 12 |
| <i>Figura 2 Principios del modelo de Motley-Keenan. (16)</i> | 13 |
| <i>Figura 3 Principios del modelo Multi-Wall. (16)</i> | 15 |
| <i>Figura 4 Principios del modelo de Trazado de Rayos. (19)</i> | 17 |
| <i>Figura 5 Representación del Trayecto Dominante. (19)</i> | 19 |
| <i>Figura 6 Ejemplo de una regresión lineal con una variable dependiente y una variable independiente.</i> | 23 |
| <i>Figura 7 Modelo de arquitectura del módulo.</i> | 40 |
| <i>Figura 8 Representación gráfica de la propuesta de solución.</i> | 40 |
| <i>Figura 9 Modelo de Datos.</i> | 51 |
| <i>Figura 10 Diagrama de clases del diseño.</i> | 52 |
| <i>Figura 11 Diagrama de paquetes del módulo implementado.</i> | 55 |
| <i>Figura 12 Diagrama de componentes del módulo AndromedaGrapher.</i> | 56 |
| <i>Figura 13 Método lineal de la clase regresión en la aplicación.</i> | 57 |
| <i>Figura 14 Obtener valores que mejor ajuste tienen aplicando el ajuste.</i> | 58 |
| <i>Figura 15 Recta que mejor se ajusta a las N medidas.</i> | 59 |
| <i>Figura 16 Método para obtener el valor de correlación.</i> | 59 |
| <i>Figura 17 Resultados de las pruebas</i> | 65 |
| <i>Figura 18 Fenómeno de la Difracción de la señal.</i> | 72 |
| <i>Figura 19 Fenómeno de la Dispersión de la señal.</i> | 72 |
| <i>Figura 20 Fenómeno de la Reflexión de la señal.</i> | 72 |
| <i>Figura 21 Flujos de trabajo para la metodología SCRUM.</i> | 73 |
| <i>Figura 22 Ejemplo de código de la clase Element.</i> | 77 |
| <i>Figura 23 Fragmento de código de la clase Device.</i> | 77 |
| <i>Figura 24 Fragmento de código de la clase MultiWallModel.</i> | 77 |
| <i>Figura 25 Ejemplo de código de la clase Andromeda.</i> | 78 |
| <i>Figura 26 Ejemplo de código de la clase ModelControl.</i> | 78 |
| <i>Figura 27 Ejemplo de código de la clase regresión</i> | 78 |
| <i>Figura 28 Dispersión con regresión lineal.</i> | 79 |
| <i>Figura 29 Tabla de datos muestrales.</i> | 88 |
| <i>Figura 30 Resultados de prueba unitaria a la clase SQLiteconexion</i> | 93 |
| <i>Figura 31 Resultados de prueba unitaria a la clase regresión</i> | 93 |



Índice de Tablas

| | |
|---|----|
| Tabla 1. Variable de pérdida de trayecto (15) | 12 |
| Tabla 2. Usuarios del sistema | 41 |
| Tabla 3. HU 1: Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite. | 44 |
| Tabla 3. Estimación por esfuerzo para el desarrollo de la solución. | 46 |
| Tabla 4. Plan de Iteraciones..... | 47 |
| Tabla 5. Plan de Entrega..... | 48 |
| Tabla 6. Tarea 1: Estudiar cómo utilizar el Lookup en la guía definitiva de la Plataforma de Netbeans 7. .. | 61 |
| Tabla 7. HU 2: Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado. | 73 |
| Tabla 8. HU 3: Obtener gráfico de barras partir de la selección del usuario. | 74 |
| Tabla 9. HU 4: Obtener gráfico lineal a partir de la selección del usuario. | 74 |
| Tabla 10. HU 5: Obtener gráfico de dispersión a partir de la selección del usuario. | 74 |
| Tabla 11. HU 6: Exportar gráficos generado en formato .PNG..... | 75 |
| Tabla 12. HU 7: Exportar gráficos generado en formato .SVG. | 75 |
| Tabla 13. HU 8: Exportar gráficos generados en formato .PDF. | 76 |
| Tabla 14. HU 9: Personalizar los gráficos generados..... | 76 |
| Tabla 15. Tarea 2: Estudio de la arquitectura de la herramienta Andrómeda. | 79 |
| Tabla 16. Tarea 3: Diseñar interfaz gráfica del módulo. | 79 |
| Tabla 17. Tarea 4: Implementar clase SQLManager_Conexion..... | 80 |
| Tabla 18. Tarea 5: Estudiar cómo obtener gráficos de barras con la librería JFreeChart. | 80 |
| Tabla 19. Tarea 6: Diseñar e implementar la vista Vista_graficar. | 80 |
| Tabla 20. Tarea 7: Implementar la clase g_barras. | 81 |
| Tabla 21. Tarea 8: Implementar la funcionalidad Graficar. | 81 |
| Tabla 22. Tarea 9: Estudiar cómo obtener gráficos lineales con la librería JfreeChart..... | 81 |
| Tabla 23. Tarea 10: Implementar la clase gráficoLineal..... | 82 |
| Tabla 24. Tarea 11: Implementar la funcionalidad Graficar. | 82 |
| Tabla 25. Tarea No 12: Estudiar el método de la regresión lineal simple por mínimos cuadrados. | 82 |
| Tabla 26. Tarea No 13: Implementar las clases regresión..... | 83 |
| Tabla 27. Tarea No 14: Diseñar e implementar la vista Vista_comparar_modelos. | 83 |
| Tabla 28. Tarea No 15: Estudiar cómo obtener gráficos de dispersión con la librería JfreeChart. | 83 |
| Tabla 29. Tarea No 16: Implementar la clase gráficoDispersión. | 84 |
| Tabla 30. Tarea No 17: Implementar la funcionalidad Graficar. | 84 |
| Tabla 31. Tarea No 18: Implementar la funcionalidad Graficar y calcular..... | 84 |
| Tabla 32. Tarea No 19: Estudiar como exportar gráficos en formato .PNG con la librería JfreeChart. | 85 |
| Tabla 33. Tarea No 20: Implementar la funcionalidad que permita guardar gráficos en el formato .PNG. .. | 85 |
| Tabla 34. Tarea No 21: Estudiar como exportar gráficos en formato .SVG con la librería JfreeChart. | 85 |
| Tabla 35. Tarea No 22: Implementar la funcionalidad que permita guardar gráficos en el formato .SVG. ... | 86 |
| Tabla 36. Tarea No 23: Estudiar en la librería JFreeChart como exportar en formato .PDF..... | 86 |
| Tabla 37. Tarea No 24: Implementar la funcionalidad que permita guardar gráficos en el formato .PDF. ... | 86 |
| Tabla 38. Tarea No 25: Estudiar en la librería JfreeChart las opciones de personalización disponibles para los gráficos generados. | 87 |
| Tabla 39. Tarea No 26: Implementar las opciones de personalización de los gráficos. | 87 |



Índice de ecuaciones

| | |
|---|----|
| <i>Ecuación 1. Pérdida en entornos cerrados. (16)</i> | 12 |
| <i>Ecuación 2. Cálculo de la atenuación de la señal. (17)</i> | 13 |
| <i>Ecuación 3. Ecuación simplificada de la atenuación de la señal. (17)</i> | 14 |
| <i>Ecuación 4. Pérdida de señal para Multi-Wall Model (18).</i> | 14 |
| <i>Ecuación 5. Cálculo de pérdida de trayecto para modelo Trayecto Dominante (20).</i> | 19 |
| <i>Ecuación 6. Función de evaluación en regresión lineal.</i> | 22 |
| <i>Ecuación 7. Ecuación para evaluación en regresión lineal múltiple.</i> | 23 |
| <i>Ecuación 8. Ecuación de regresión de la muestra.</i> | 23 |
| <i>Ecuación 9. Modelo de regresión Polinomial.</i> | 24 |
| <i>Ecuación 10. Aproximación en la regresión lineal con un conjunto n de datos.</i> | 24 |
| <i>Ecuación 11. Suma de las desviaciones verticales.</i> | 25 |
| <i>Ecuación 12. Expresión para minimizar los residuos al cuadrado.</i> | 25 |
| <i>Ecuación 13. Reemplazo de la ecuación para minimizar residuos.</i> | 25 |
| <i>Ecuación 14. Derivación parcial de una función en mínimos cuadrados.</i> | 26 |
| <i>Ecuación 15. Modelo básico de suavización exponencial.</i> | 27 |
| <i>Ecuación 16. Cálculo de pronóstico para el período 2 en suavización exponencial.</i> | 27 |
| <i>Ecuación 17. Ecuaciones para encontrar la pendiente y la ordenada.</i> | 56 |
| <i>Ecuación 18. Ecuaciones para encontrar los valores de las S.</i> | 57 |
| <i>Ecuación 19. Aplicación de los mínimos cuadrados a la función de evaluación.</i> | 59 |



Introducción

El extraordinario desarrollo experimentado en el mundo de la informática y las comunicaciones ha hecho posible adelantos científicos que protagonizan el cambio tecnológico de la presente época, enmarcando el camino de las comunicaciones y el intercambio de información hacia un ambiente de constante evolución, caracterizado por nuevos paradigmas tecnológicos, que hacen contar hoy en día con soluciones novedosas en diferentes esferas de la sociedad.

La conexión mediante cables ha sido una de las soluciones más impactantes en el marco comunicacional a nivel mundial, llegándose a destacar dentro de las principales soluciones tecnológicas la producción y uso de los cables coaxiales o la fibra óptica. En la actualidad ha incrementado significativamente el interés de muchos en cuanto a cómo poder comunicar diferentes equipos de cómputo sin la necesidad de utilizar redes cableadas, es decir, de qué forma se pueden comunicar dos dispositivos de manera inalámbrica. Fundamentalmente, esto se debe a la movilidad que brinda a los usuarios, y se presenta como una solución perfectible para aquellos que no tienen una red en lugares remotos o inaccesibles, para los cuales las redes cableadas es algo impensable. Factores estos que propiciaron que el uso de las Redes de Área Local Inalámbricas (WLAN, del inglés *Wireless Local Area Network*) haya comenzado a formar parte indisoluble del mundo de las comunicaciones (1).

El perfeccionamiento de esta tecnología ha permitido que las transmisiones inalámbricas constituyan una eficaz herramienta para la transferencia de voz y datos sin el obstáculo de tener que utilizar cables para ello. El uso de ondas electromagnéticas para realizar este intercambio de información brinda un grupo de ventajas importantes, dentro de las que destacan movilidad y flexibilidad; además de representar una disminución en los costos de despliegue y mantenimiento. Por lo que la tecnología inalámbrica ha pasado a convertirse en elemento esencial para muchas personas y empresas (2).

Al profundizar en este sentido se aprecia como muchas instituciones y empresas que requieren de un intercambio continuo de información se han apoderado del uso de las WLAN, aprovechando que estos sitios son generalmente espacios reducidos y que en estos marcos no es necesario el acceso a internet. Dichos factores han incidido



decisivamente en la puesta en práctica de las Redes Inalámbricas de Sensores (RIS o WSN del inglés; *Wireless Sensor Networks*) y convirtiéndose además, en cantera para nuevas investigaciones científicas sobre el tema. En febrero de 2003 el Instituto de Tecnología de Massachusetts identificó las 10 tecnologías emergentes que cambiarán el mundo, las WSN aparecen en primera posición. Después, producto a su gran interoperabilidad con otros dispositivos, se definen como Redes Inalámbricas de Sensores y Actuadores (3).

Una Red Inalámbrica de Sensores y Actuadores (WSAN, del inglés; *Wireless Sensor and Actuator Networks*, y en lo adelante WSAN), es una red de pequeños computadores (nodos), equipados con sensores que trabajan con un fin común, ya sea en el monitoreo y/o control. Están formadas por un grupo de sensores con ciertas capacidades sensitivas y de comunicación inalámbrica los cuales permiten formar redes Ad hoc¹ sin infraestructura física preestablecida ni administración central conectada a un sistema (3).

Las WSAN tienen el potencial de revolucionar los complejos sistemas de control u observación, tal y como hoy se conocen. Al referirse a este aspecto, **Nodarse (2012)**, puntualiza que dentro de las aplicaciones que tienen estas redes, se destaca su utilización en los Sistemas de Control de Edificios (SCE). Factores externos como el ahorro energético, emparejado a las ventajas que aporta esta tecnología a los SCE actuales, han incidido en su gran aceptación en este campo de aplicación (4).

Lamentablemente, el uso de WSAN en este tipo de escenarios, trae consigo el surgimiento de ciertos problemas, fundamentalmente en la fase de despliegue. A causa de que el campo eléctrico dentro de una edificación está integrado por un amplio número de componentes indirectos y factores considerables que no se tienen en cuenta en el caso del diseño de redes para entornos abiertos; por lo que en entornos interiores los niveles de señal de Radio Frecuencia (RF), que es la encargada de portar los datos de cada entorno en que están situados los sensores, fluctúan en mayor medida que en entornos exteriores. Consecuentemente, la cobertura inalámbrica se caracteriza por ser compleja y muchas veces puede resultar incontrolable.

¹ Una red "Ad hoc", consiste en un grupo de ordenadores que se comunican cada uno directamente con los otros a través de las señales de radio sin usar un punto de acceso. Las configuraciones "Ad hoc", son comunicaciones de tipo punto a punto. (74)



Estos factores y la necesidad de hacer un modelado tan preciso como sea posible ha dado paso a que investigadores y analistas propongan modelos de propagación capaces de detectar, predecir y corregir las afectaciones a las que puede estar sometida la señal de RF en relación con los obstáculos presentes en los lugares donde se utilicen las WSAN.

Observaciones de **García Fernández (2006)** y **Korkalainen (2010)** en relación con la propagación de la señal en una edificación revelan que 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, lo cual da surgimiento al problema del multitrayecto de la señal. En este contexto la radio propagación afecta de manera directa las señales que se difunden, debido a los fenómenos de reflexión, difracción, absorción y dispersión, impactando significativamente en la cobertura y desempeño de las redes (5) (6).

Para poder simular el comportamiento de la señal de RF dentro de edificaciones, resulta importante la utilización de los modelos de propagación². Existe una amplia bibliografía sobre el tema. En la misma se proponen diversas variantes de clasificación de dichos 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.

La motivación que se tiene para emplear un modelo de propagación para interiores puede ser de variada índole. En el ambiente de las empresas, tiene por objetivo predecir una cobertura de servicio inalámbrico; en el mundo académico y de la investigación, el desarrollar herramientas que ayuden en la toma de decisiones para cubrir un área de servicio con la menor inversión posible. A nivel mundial las principales soluciones se basan en el empleo de herramientas que faciliten de forma efectiva el despliegue de

² 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).



sensores en el interior de edificaciones, mediante el uso de simulaciones digitales de la propagación de la señal de RF desde el transmisor y hasta el receptor.

Una de las principales deficiencias de estas herramientas es la incorrecta selección de los modelos de propagación de la señal de RF a la hora de realizar la propuesta para el despliegue en el interior de una edificación, debido a que en las simulaciones no se tienen en cuenta el tipo de material y los obstáculos presentes en la edificación, donde solo se centran en la cobertura y la conectividad interna de la red. Estos modelos resultan de gran importancia en el diseño de redes y sistemas, pues dada su gran diversidad, es necesario determinar los modelos más eficaces para cada entorno. Estos dependen esencialmente de datos como la pérdida de la señal a través de los pisos, el número de paredes entre el transmisor y el receptor, el número de pisos que atraviesa la señal, el número de tipo de paredes atravesadas por la señal, entre otros muchos datos que se generan de acuerdo al tipo de modelo de propagación que se vaya a emplear.

Como consecuencia a las deficiencias anteriores y otros problemas identificados en los sistemas actuales, fue necesario crear una herramienta software que genere de forma automática, el número y la posición de los dispositivos que forman parte de la WSN que satisfagan el diseño de la red a desplegar en el interior de edificaciones. Dicha herramienta lleva por nombre Andrómeda y está siendo desarrollada por el grupo de investigación del mismo nombre perteneciente al Centro FORTES de la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI).

Sin embargo, tanto Andrómeda -que aún está en fase de desarrollo- como las actuales herramientas de apoyo más usadas para la modelación de las WSN, no cuenta con ninguna aplicación que sugiera mediante análisis de datos o gráficas, cuál o cuáles serían los modelos de propagación más convenientes a emplear en un momento determinado, según las características de la edificación en la que se desee desplegar una red inalámbrica de sensores, y cuál o cuáles se ajustan mejor según la pérdida de la señal de RF producida desde el transmisor y hasta el receptor.

Teniendo en cuenta la problemática antes mencionada, se plantea el siguiente **problema a resolver**: ¿Cómo analizar el comportamiento de los modelos de propagación de la señal de radio frecuencia para contribuir a su mejor selección en la herramienta Andrómeda?

Es por ello que el **objeto de estudio** de la presente investigación se centra en los modelos de propagación para entornos interiores, definiendo como **campo de acción** la simulación de modelos de propagación empleados en la herramienta Andrómeda.

Para darle solución al problema antes mencionado se propone como **objetivo general**: Desarrollar un módulo para el análisis del comportamiento de los modelos de propagación en la herramienta Andrómeda.

Para el cumplimiento de los objetivos propuestos se proponen los siguientes **objetivos específicos**:

- ⚙ Caracterizar los métodos para la comparación de modelos matemáticos.
- ⚙ Diseñar una propuesta para la realización de experimentos y comparaciones entre los modelos de propagación de la herramienta Andrómeda.
- ⚙ Implementar un módulo que permita el análisis del comportamiento de los modelos de propagación en la herramienta Andrómeda.

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

- ⚙ Análisis del estado del arte de los métodos para comparar modelos matemáticos, así como las tecnologías a ser utilizadas.
- ⚙ Selección de los métodos matemáticos para análisis estadísticos ajustados a la investigación.
- ⚙ Levantamiento de las funcionalidades y especificaciones necesarias.
- ⚙ 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.
- ⚙ Estudio de la arquitectura de la herramienta Andrómeda.
- ⚙ Realización de las actividades y artefactos que tributen a la implementación de la propuesta de solución de acuerdo a la metodología de desarrollo.
- ⚙ Implementación del método matemático que se ajuste al desarrollo de la investigación, así como las funcionalidades necesarias para mostrar los resultados obtenidos.
- ⚙ Realización de pruebas unitarias y de aceptación.

Finalmente, para guiar el desarrollo de la solución se plantea la siguiente **hipótesis**: la implementación y desarrollo de un módulo que permita analizar el comportamiento de las señales de Radio Frecuencia en los modelos de propagación implementados contribuirá a su mejor selección en la herramienta Andrómeda.

Para el desarrollo de las tareas de investigación se tienen en cuenta los siguientes **métodos teóricos**, que posibilitan el conocimiento del estado del arte del problema, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado:

Analítico-Sintético: este método 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 dispersión a través de diversos materiales obteniendo los elementos necesarios para elaborar conclusiones que brinden solución a la necesidad de la investigación.

Histórico-Lógico: se utilizó para estudiar la evolución y desarrollo de los Modelos de Propagación de la señal de Radio Frecuencia en el interior de edificaciones, así como los diferentes métodos matemáticos de análisis estadísticos que a nivel mundial se utilizan y comprender lógicamente cuáles son las tendencias actuales.

Inductivo-Deductivo: utilizado para el estudio de los diferentes métodos matemáticos que son empleados en el análisis estadístico de datos con el fin de definir cuál o cuáles de los métodos encontrados pueden ser utilizados en la implementación del módulo.

La presente investigación se estructura en introducción, resumen, índice general, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. A continuación se resume el contenido de los tres capítulos:

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, estudio de los diferentes modelos de propagación y de los métodos matemáticos más utilizados para el análisis estadístico, así como la descripción y selección de las herramientas y metodología a utilizar.



Capítulo 2: Descripción de la solución propuesta. Se hace una descripción del entorno en el que se desarrolla la propuesta, se conforman las historias de usuario, las cuales se organizan en iteraciones según la prioridad definida por el cliente, además 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 realizados al producto final.



1. Fundamentación Teórica

El auge con que en la presente década se presentan las redes de comunicación inalámbrica requiere de rapidez y efectividad, dada la relevancia de las aplicaciones y servicios soportados por estas redes (7). Con el fin de agilizar el proceso de diseño y realizarlos de una manera eficiente, vinculado además a su rápido crecimiento, los investigadores desarrollan herramientas que permiten la valoración del medio a partir de modelos de propagación de la señal de RF (8).

En el desarrollo del presente capítulo se abordarán los principales conceptos y principios teóricos que sustentan la investigación, los cuales estarán avalados por referencias bibliográficas actualizadas. Se especificarán además, las clasificaciones fundamentales de los diferentes modelos de propagación, así como los principales métodos matemáticos de análisis estadísticos, caracterizándolos y referenciando las 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.

1.1 Conceptos asociados a la investigación

Con la finalidad de facilitar la comprensión de este trabajo, a continuación se enuncian los conceptos fundamentales asociados a la investigación.

1.1.1 Análisis estadístico

Se define análisis estadístico como el conjunto de métodos, técnicas y procedimientos para el manejo de datos. Su ordenación, presentación, descripción, análisis e interpretación, favorecen el estudio científico de problemas planteados en distintos sectores y a la toma de decisiones (9).

1.1.2 Canal de radio

Es el canal de comunicación a través del cual viajan las ondas electromagnéticas entre el receptor y el transmisor estableciéndose el proceso de comunicación. El transmisor es el encargado de procesar la información y posteriormente radiarla, mientras que el receptor es el encargado de captar la señal, procesarla y así poder recuperar la información (10).



1.1.3 Difracción de la señal

En óptica se entiende por difracción a la desviación del rayo luminoso al rozar el borde de un cuerpo opaco. Las ondas difractadas se forman cuando el camino de propagación de la onda de radio es obstruido por una superficie que tiene irregularidades o bordes puntiagudos o angulados. La difracción ocurre cuando los obstáculos son impenetrables por las ondas de radio. Basándose en el Principio de Huygen³, el resultado son ondas secundarias alrededor y detrás del obstáculo, incluso en zonas sin visibilidad directa entre transmisor y receptor (11). ([Anexo 1](#))

1.1.4 Dispersión de la señal

La dispersión ocurre cuando en el camino la señal se encuentra con objetos cuyas dimensiones son pequeñas con relación a la longitud de onda. El resultado es que el frente de onda se rompe o dispersa en múltiples direcciones. Las ondas dispersas son producidas por superficies desiguales, pequeños objetos y otras irregularidades presentes en el canal. La mayoría de las construcciones modernas contienen vigas de hierro forjado en su estructura además de conductos para los servicios eléctricos y de tuberías. En la práctica, el follaje, señales de tráfico o farolas pueden provocar dispersión en sistemas de comunicaciones inalámbricos (11). ([Anexo 1](#))

1.1.5 Estadística

La estadística es la ciencia que tiene como objetivo reunir la información cuantitativa concerniente a individuos, grupos y series de hechos; y deducir de ello, basado en el análisis de estos datos, significados precisos o previsiones para el futuro. Harald Cramér se limita a establecer que *“el principal objeto de la teoría estadística consiste en la investigación de la posibilidad de obtener inferencias válidas a partir de los datos estadísticos, y en la construcción de métodos para realizar dichas inferencias”* (12).

1.1.6 Reflexión de la señal

El fenómeno de la reflexión ocurre cuando una onda electromagnética que se propaga por el aire incide contra un objeto de grandes dimensiones en comparación con la longitud de onda de la señal. El resultado puede ser que la señal sea absorbida, reflejada o una combinación de ambas. Los conductores perfectos reflejarán la totalidad de la señal.

³ Todo punto alcanzado por una onda se comporta como un emisor de ondas. El método explica cómo se puede construir un frente de ondas conociendo el frente en un instante anterior.

Otros materiales reflejarán solo una parte de la energía incidente y transmitirán el resto. La cantidad exacta de transmisión y reflexión depende igualmente del ángulo de incidencia así como del grosor y propiedades dieléctricas del material (11). ([Anexo 1](#))

1.1.7 Pérdidas en espacio libre

Es el más simple mecanismo de propagación. Considera línea de vista entre las antenas, y asume que la separación entre antenas es tal, que ninguna está en el campo cercano de la otra. De este modo la potencia recibida puede calcularse directamente en función de la distancia entre antenas; si se conoce la potencia transmitida (13).

1.1.8 Pronóstico estadístico

Un pronóstico en el mundo estadístico no es más que el proceso de estimación en situaciones de incertidumbre. Son procesos críticos y continuos que son necesarios para obtener buenos resultados durante la planificación de un proyecto. El pronóstico basado en el análisis estadístico es el procedimiento más exacto siempre que exista una relación entre el pasado y el futuro. En realidad el pasado brinda la mejor base para las decisiones referentes a la acción futura (14).

1.1.9 Propagación en espacio libre

A la propagación cuando no existen obstáculos en el camino entre el transmisor y el receptor se le denomina propagación en espacio libre. La pérdida de señal que se produce está en función principalmente de la distancia que les separa, interviniendo otros factores como el tipo y diseño de las antenas y su patrón de radiación (10).

Los modelos de propagación para ambientes interiores mejoran con el paso del tiempo, en conjunto con los instrumentos de simulación que se requieren para lograr predicciones seguras, exactas, y hasta cierto punto computacionalmente eficiente, intentando en todo momento vencer las pérdidas de señal en la propagación de RF en interiores. A continuación se describen y detallan elementos importantes de algunos de los modelos de propagación más usados, que son hoy objeto de estudio de muchos investigadores, y algunos de los cuales son implementados en la herramienta informática Andrómeda.

1.2 Modelos de propagación para entornos interiores (Indoor)

Ampliando la clasificación que se realizó en la introducción, los modelos de propagación para ambientes interiores (del inglés; *indoor propagation models*) se pueden clasificar en dos grupos (15): modelos empíricos y modelos deterministas. Los modelos empíricos por su parte se clasifican en modelos empíricos de banda estrecha y modelos empíricos de banda ancha.

Los modelos empíricos de banda estrecha se expresan como una ecuación matemática simple, mediante la cual se obtienen las pérdidas de trayecto de la señal. Mientras que los modelos empíricos de banda ancha permiten predecir las características del ancho de banda del canal. Los modelos deterministas, por su parte, son métodos de cálculo que simulan físicamente la propagación de ondas de radio. Estos modelos brindan información tanto en banda ancha como en banda estrecha del canal dentro del edificio.

Todos los modelos requieren de una base de datos que describa el ambiente de propagación. Cada elemento debe ser clasificado en diferentes categorías (muros, pisos, puertas, ventanas) y se deben conocer sus coordenadas así como las propiedades de sus materiales (espesor, conductividad, permisividad) (13).

1.2.1 Modelos de propagación empíricos

Los modelos de propagación empíricos o estadísticos basan su funcionamiento en la extrapolación estadística de resultados a partir de medidas del fenómeno realizadas sobre el terreno. La principal ventaja que presenta este tipo de modelo es que se tienen en cuenta de forma implícita las influencias propias del entorno en su conjunto, sin ser reconocidas cada una de ellas de manera aislada. La precisión de estos modelos 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.

A continuación se describen algunos de estos modelos, dando detalles que permitan su caracterización y mejor comprensión.

➤ Modelo Log – Normal Shadowing Path Loss

Para este modelo empírico las pérdidas en entornos cerrados se pueden expresar por una ecuación en función de la distancia entre transmisor y receptor como:

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

Ecuación 1. Pérdida en entornos cerrados. (15)

Donde:

- $PL(d_0)$: Pérdida a distancia cercana de referencia.
- n : Variable de pérdida de trayecto.
- d : Distancia de separación en metros.
- X_σ : Desviación típica

La variable de pérdida de trayecto “ n ” depende del tipo de entorno. La Tabla 1 muestra los valores típicos de n para diferentes entornos. Por su parte, X_σ es una variable aleatoria expresada en dB (decibelios) que indica una desviación típica de σ . Mientras que $PL(d_0)$ se calcula utilizando el Modelo de propagación en espacios libres o realizando medidas de campo.

Tabla 1. Variable de pérdida de trayecto (15)

| 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 |

La siguiente figura (fig. 1) muestra las curvas de pérdida de trayecto para diferentes valores de la variable n .

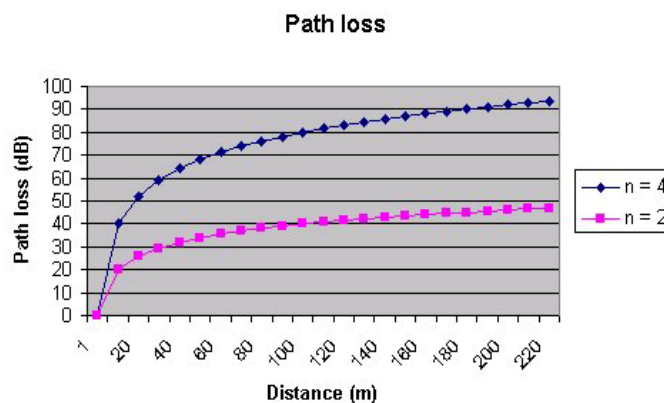


Figura 1 Pérdidas de Trayecto para diferentes valores de n . (15)

Log – Normal Shadowing Path Loss es una técnica de estimación de pérdidas de propagación muy práctica y como principal ventaja destaca el tomar en cuenta en su totalidad a los factores que influyen en la propagación.

➤ **Modelo COST 231 Keenan y Motley**

Este modelo tiene la peculiaridad de haber sido desarrollado exclusivamente para propagación en ambientes interiores. COST 231 es un modelo empírico que considera la pérdida en el espacio libre en conjunto con la pérdida adicional que se produce cuando la señal directa entre transmisor y receptor atraviesa diferentes paredes y pisos (16).

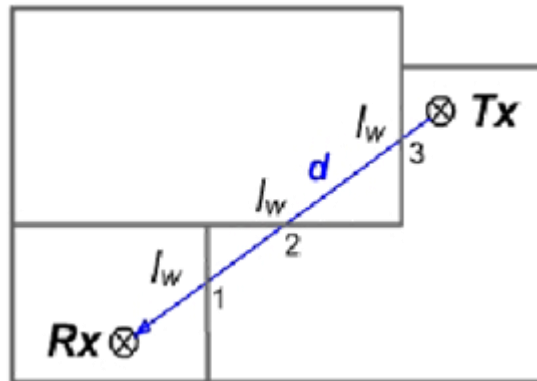


Figura 2 Principios del modelo de Motley-Keenan. (16)

Para su aplicación es necesario conocer un gran volumen de datos, y la atenuación de la señal se determina a través de la siguiente ecuación matemática:

$$L(dB) = L_0 + 10\gamma \log d + \sum_{i=1}^I N_{f,i} L_{f,i} + \sum_{j=1}^J N_{w,j} L_{w,j}$$

Ecuación 2. Cálculo de la atenuación de la señal (16).

Donde:

- L_0 : Pérdidas de propagación a un (1) metro de la antena transmisora, en dB.
- $L_{f,i}$: Pérdidas de propagación de la señal a través de los pisos, en dB.
- $N_{f,i}$: Número de pisos con las mismas características.
- $L_{w,j}$: Pérdidas de propagación de la señal a través de paredes, en dB.
- $N_{w,j}$: Número de paredes con las mismas características.

- I : Número de tipos de pisos atravesados por la seal.
- J : Número de tipos de paredes atravesadas por la seal.

En caso de no contar con la informaci3n suficiente sobre los tipos de paredes y pisos, se aplica versi3n simplificada del modelo, con un nico tipo de piso y dos tipos de paredes:

$$L(dB) = 37 + 20\gamma \log d + N_f L_f + \sum_{j=1}^2 N_{w,j} L_{w,j}$$

Ecuaci3n 3. Ecuaci3n simplificada de la atenuaci3n de la seal (16).

Donde:

N_f : Nmero de pisos atravesados.

L_f : Prdidas a travs de los pisos, en dB.

$N_{w,j}$: Prdidas en paredes ligeras de madera, puertas, etc., en dB.

$L_{w,j}$: Prdidas en paredes gruesas, tabiques de ladrillo, cemento, etc., en dB.

➤ **Modelo COST 231 Multi-Wall**

El modelo Multi-Wall recogido en el COST 231, se considera como una evoluci3n del anteriormente mencionado modelo de Keenan y Motley. En dicho modelo se toma la simplificaci3n respecto del anterior, de considerar todos los suelos como el mismo tipo, sin hacer distinciones. Sin embargo, esta aproximaci3n es menos significativa que la aparici3n de un nuevo parmetro emprico que ampla el nmero de estos tipos de suelos y permite una mejor aproximaci3n (17).

Se expone a continuaci3n la ecuaci3n del Multi-Wall:

$$L(dB) = L_{f,s}(d, n) + L_c + \sum_{i=1}^N K_{w,i} L_{w,i} + K_f L_f \left(\frac{K_f + 2}{K_f - 1} - b \right)$$

Ecuaci3n 4. Prdua de seal para Multi-Wall Model (17).

Donde $L_{f,s}$ depende de la distancia d y del parmetro emprico n , L_c es un factor utilizado para ajustar las prdidas debidas a los cables y otros scatters, $K_{w,i}$ y $L_{w,i}$ estn relacionados con cada uno de los tipos de paredes, K_f y L_f con el nmero de suelos

atravesados y su parámetro empírico asociado y por último b , que es el nuevo parámetro que aparece respecto al modelo anterior.

Se debe tener presente que, además de tener en cuenta las atenuaciones en paredes y suelos, estos parámetros empíricos engloban las pérdidas en el mobiliario del edificio. El modelo es considerado uno de los más adecuados por su eficiencia y precisión como modelo empírico que es, necesitando una campaña previa de medición para acumular datos que permitan optimizar sus parámetros.

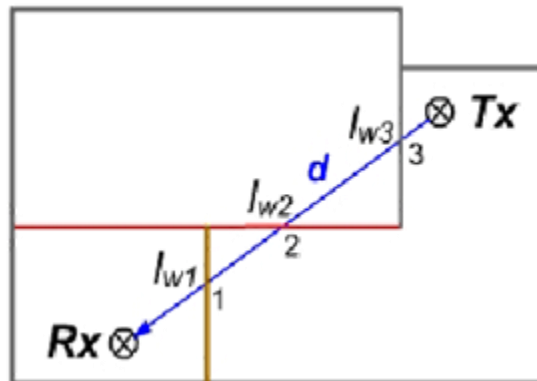


Figura 3 Principios del modelo Multi-Wall. (16)

1.2.2 Modelos de propagación teóricos o deterministas

Los modelos de propagación clasificados como teóricos o deterministas son aquellos que 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. Para ser implementados requieren de enormes bases de datos de características relativas al entorno, las cuales son imposibles de obtener de manera práctica.

Los algoritmos usados por estos modelos son generalmente de gran complejidad y considerados 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. Los modelos deterministas son actualmente los más utilizados debido a que ofrecen mejores resultados que los estadísticos en cuanto a precisión.

También se puede emplear uno u otro modelo en función del problema al que se hace referencia, pudiendo desarrollar modelos híbridos con mejores resultados en cuanto a la

relación de tiempo invertido y precisión obtenida. Todas las ventajas expuestas se enfrentan al problema de cálculo comentado anteriormente que estos modelos presentan, ya que las variables del entorno pueden ser elevadísimas teniendo que realizar simplificaciones (18).

➤ **Modelos basados en técnica de trazados de rayos (Ray Tracing)**

El algoritmo de trazado de rayos (del inglés; *Ray Tracing*), calcula todos los posibles caminos desde el transmisor hacia el receptor. En los modelos básicos de trazados de rayos la predicción se basa en cálculos de transmisión en espacios abiertos complementados con el efecto de reflexión de las paredes, teniendo en cuenta una única reflexión. Los de trazados de rayos más complejos incluyen mecanismos de difracción, dispersión difusa y penetración a través de diversos materiales (19).

Los argumentos expuestos por **Landstorfer (2013)** demuestran que 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.

En la actualidad 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 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 (18).

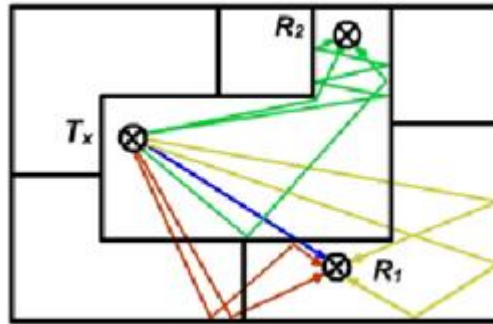


Figura 4 Principios del modelo de Trazado de Rayos. (19)

➤ Modelo Diferencias Finitas en el dominio del Tiempo (FDTD)

Se pueden obtener soluciones relativamente simples para el estudio de la radio propagación con aquellos modelos que hacen uso de la Óptica Geométrica (GO) y de la Teoría Uniforme de la Difracción (UTD). **Arfken (2013)**, al profundizar sobre este particular, explica que la 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, por lo que no se pueden aplicar soluciones asintóticas, aunque sí la solución a las ecuaciones de Maxwell. 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 (20).

El método de Diferencias Finitas en el Dominio del Tiempo (FDTD) 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 definidas estas condiciones iniciales, el algoritmo en cuestión 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 rayos, los modelos FDTD son 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 al tamaño del área y la frecuencia de operación.

Dicho modelo permite especificar el material en todos los puntos dentro del dominio computacional. Sin embargo, FDTD requiere que todo este dominio sea cuadrículado, y la desratización espacial de la red debe ser lo suficientemente fina para resolver tanto la menor longitud de onda electromagnética como la característica geométrica más pequeña en el modelo, convirtiéndose esto en un atenuante de su uso (21).

Si se consideran las valoraciones dadas por **Arfken (2013)** se puede constatar que, debido a su complejidad computacional, el modelo FDTD es aconsejable solo en tareas de predicción de campo en pequeñas áreas. Para áreas de mayor tamaño los modelos de trazado de rayos son más convenientes (20).

➤ **Modelo de Trayecto Dominante**

El Modelo de Trayecto Dominante (DPM, del inglés; *Dominant Path Model*) determina el camino dominante entre el transmisor y el receptor. En cambio, calcula diferentes trayectos que son representativos para ciertos grupos de rayos con características similares y además, son independientes de la variación del canal (movimiento de personas u objetos). Por lo que el tiempo de cálculo en comparación con el trazado de rayos se reduce significativamente y la exactitud es casi idéntica a dicho modelo.

Los modelos empíricos (como COST 231) consideran solamente la trayectoria directa entre un transmisor y un receptor. Mientras que DPM determina la ruta más relevante, lo cual conduce a la minimización de los tiempos de cálculos.

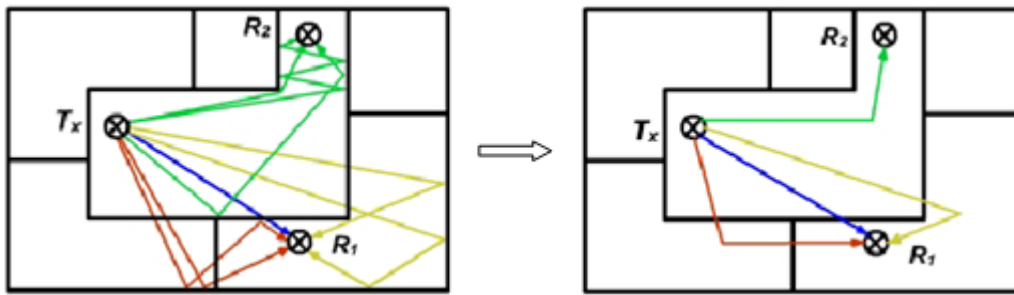


Figura 5 Representación del Trayecto Dominante. (19)

Las principales características de este modelo son (19):

- ⚙ 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.
- ⚙ 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 5. Cálculo de pérdida de trayecto para modelo Trayecto Dominante (19).

Donde:

- λ : 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.
- Ω : Ganancia causada por el guiado de ondas para cada píxel a lo largo de una trayectoria de propagación.

Como se expone anteriormente, d es la longitud de la trayectoria entre el transmisor y el receptor del 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.

Al profundizar sobre el modelo de Trayecto dominante **Wolfle (2012)** considera que, con él se puede distinguir entre varios modos de propagación para definir el exponente de pérdida de trayectoria. Los modos son (22):

- ⚙ 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.

1.3 Métodos matemáticos empleados en los pronósticos estadísticos

Los pronósticos son una de las herramientas fundamentales para la toma de decisiones en empresas y en diferentes ámbitos sociales. Pronosticar es el arte y la ciencia de predecir los eventos futuros. Generalmente los pronósticos son tanto intuitivos como subjetivos, esto es debido a que la toma de decisiones se lleva a cabo con un buen modelo matemático y el juicio de quien los esté utilizando. En el contexto de esta investigación se hace necesario el uso de métodos matemáticos de pronóstico estadístico, cuyo objetivo fundamental es el de comparar y brindarle a los expertos elementos que les permitan realizar proyecciones futuras sobre el comportamiento de los diferentes modelos de propagación en escenarios interiores donde se vayan a desplegar las WSA.



Los argumentos expuestos por **Sapag Chain (2000)** demuestran que la estimación del comportamiento futuro de algunas variables puede realizarse utilizando diversas técnicas de pronóstico. Cada una de las técnicas de proyección tiene una aplicación de carácter especial que hace de su selección un problema de decisión influido por diversos factores, como por ejemplo, la validez y disponibilidad de los datos históricos, la precisión deseada del pronóstico, el costo del procedimiento, los beneficios del resultado, los períodos futuros que se desee pronosticar y el tiempo disponible para hacer el estudio, entre otros (23).

Existen diversas técnicas para realizar pronósticos. Una forma de clasificarlas consiste hacerlo en función de su carácter, definiéndose así métodos de carácter cualitativo, modelos causales y modelos de series de tiempo. Seguidamente se brinda una caracterización de estos métodos.

1.3.1 Métodos de carácter cualitativo

Estos se basan principalmente en opiniones de expertos. Su uso es frecuente cuando el tiempo para elaborar el pronóstico es escaso, cuando no se dispone de todos los antecedentes mínimos necesarios o cuando los datos disponibles no son confiables para predecir el comportamiento futuro. Resulta difícil emitir un juicio sobre la eficacia de sus estimaciones finales (24).

1.3.2 Métodos de pronóstico causales

Los métodos de pronóstico causales parten del supuesto de que el grado de influencia de las variables que afectan al comportamiento de los indicadores que se estén analizando permanezca estable, para luego construir un modelo que relacione ese comportamiento con las variables que se estime que son las causantes de los cambios que se observan en el ambiente (24). Algunos métodos causales son el análisis de la regresión, el modelo Auto regresivo de Media Móvil (ARMA), el modelo ARIMA y la econometría.

Uno de los métodos causales más conocidos es el análisis de la regresión. Esta es una rama de la teoría estadística de gran utilidad en casi todas las disciplinas científicas. Es considerada además, como la técnica básica para “medir” o “estimar” la relación existente entre variables. Los modelos de regresión se clasifican en modelos de regresión lineal y

no lineal. Esta clasificación se establece con respecto a la linealidad entre los parámetros de la ecuación y no entre las variables.

➤ **Regresión Lineal Simple**

El análisis de regresión se basa principalmente en investigar la relación que existe entre una variable respuesta o dependiente (Y) y una variable explicativa o independiente (X). El propósito es obtener una función sencilla de la variable explicativa o independiente, que sea capaz de describir lo más ajustado posible la variación de la variable respuesta o dependiente. La función más eficaz es aquella que describe la variable dependiente con la menor diferencia entre los valores observados y predichos.

La diferencia entre estos valores observados y predichos se denomina variación residual o residuos. Para evaluar los parámetros en la función se utiliza el ajuste por mínimos cuadrados, en el cual la suma de los cuadrados de las diferencias entre los valores observados y esperados sea menor, lo que es necesario que los residuos o variación residual o errores estén distribuidos normalmente y que varíen de modo similar a lo largo de todo el rango de valores de la variable dependiente. Cuando la variable dependiente es cuantitativa y la relación entre ambas variables sigue una relación recta la función es:

$$\hat{Y} = a + bx$$

Ecuación 6. Función de evaluación en regresión lineal.

Donde:

- *a*: ordenada en el origen(constante)
- *b*: pendiente de la recta

Las referencias de **Cole (2005)** avalan que cuando solo existe una variable independiente, esto se reduce a una línea recta. Donde los coeficientes *a* y *b* son parámetros que definen la posición e inclinación de la recta. El parámetro *a*, conocido como la ordenada en el origen indica cuánto es Y cuando X= 0. El parámetro *b*, conocido como “pendiente”, indica cuánto aumenta Y por cada aumento de una unidad en X. El problema consiste en obtener estimaciones de estos coeficientes a partir de una muestra de observaciones sobre las variables **Y** y **X** (25).

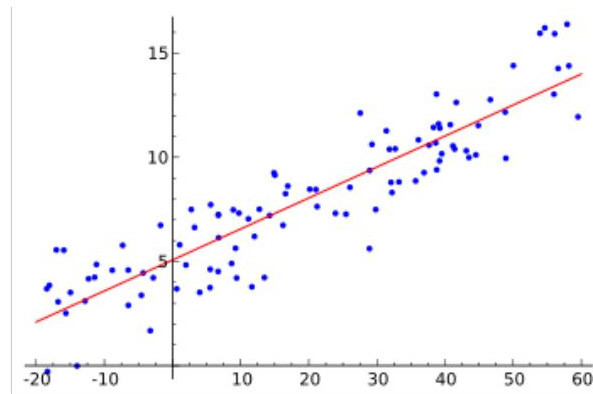


Figura 6 Ejemplo de una regresión lineal con una variable dependiente y una variable independiente.

➤ Regresión Lineal Múltiple

En la mayor parte de los problemas de investigación donde se aplica el análisis de regresión se necesita más de una variable independiente en el modelo de regresión. La complejidad de la mayor parte de los mecanismos científicos es tal que para ser capaces de predecir una respuesta importante se necesita un modelo de regresión múltiple. Cuando este modelo es lineal en los coeficientes se denomina modelo de regresión lineal múltiple. Para el caso de k variables independientes X_1, X_2, \dots, X_k , la media de $Y | X_1, X_2, \dots, X_k$ está dada por el modelo de regresión lineal múltiple.

$$\mu_{Y|X_1, X_2, \dots, X_k} = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

Ecuación 7. Ecuación para evaluación en regresión lineal múltiple.

Y la respuesta estimada se obtiene de la ecuación de regresión de la muestra:

$$\gamma = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

Ecuación 8. Ecuación de regresión de la muestra.

Donde:

Y es la variable dependiente y X_i son las variables independientes o predictoras. Las letras griegas *beta* indicadas representan los parámetros del modelo. Como en el caso de una sola variable independiente, el modelo de regresión lineal múltiple a menudo puede ser una representación adecuada de una estructura más complicada dentro de ciertos rangos de las variables independientes (26).

Las técnicas de mínimos cuadrados similares también se pueden aplicar al estimar los coeficientes cuando el modelo lineal involucra potencias y productos de las variables independientes. Por ejemplo, cuando $k = 1$, el experimentador puede pensar que las medias $\mu_{Y|x}$ no caen en una línea recta pero que se describen de forma más apropiada con el modelo de regresión polinomial.

$$\mu_{Y|x} = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_r x^r$$

Ecuación 9. Modelo de regresión Polinomial.

➤ Modelos de regresión no paramétrica

De acuerdo con las valoraciones de **Olaya (2002)** se puede decir que los objetivos que persigue una regresión no paramétrica son los mismos de su contrapuesta paramétrica, vale decir; estimar y probar las características de la función de regresión.

La idea fundamental en los modelos de regresión consiste en aproximar a partir de un conjunto de n datos (x_i, y_i) la función f en la siguiente expresión:

$$y_i = f(x_i) + \varepsilon_i, i = 1, 2, \dots, n$$

Ecuación 10. Aproximación en la regresión lineal con un conjunto n de datos.

Donde Y es continua, denominada generalmente variable respuesta; $X = (X_1, X_2, \dots, X_p)$, denota p variables continuas denominadas covariables, variables predictoras, variables explicativas o variables independientes; $f(x)$ es la respuesta media y referida en la literatura como función de regresión o curva de regresión; y ε se conoce comúnmente como error o ruido con $E[\varepsilon_i] = 0$ y $Var[\varepsilon_i] = \sigma^2 < \infty, \forall i$.

En el análisis de regresión paramétrica, el investigador presupone una forma de la función de regresión, de la cual solo se desconoce el valor de los parámetros asociados a la misma. Por el contrario, el análisis de regresión no paramétrica no asume un comportamiento del fenómeno a priori, sino que concibe la forma de la curva o función “a partir de lo que los datos digan”. La curva se escoge de entre un conjunto de curvas con ciertas propiedades de continuidad y diferenciabilidad.

➤ **Método de Mínimos Cuadrados**

El método de los mínimos cuadrados tiene como objetivo encontrar la ecuación de una recta a partir de los datos experimentales, es decir, utilizando solamente las mediciones experimentales se obtendrá la pendiente y la ordenada al origen de la recta que mejor se ajuste a tales mediciones. Es considerado además, como el procedimiento más objetivo para ajustar una recta a un conjunto de datos presentados en un diagrama de dispersión. La recta resultante presenta dos características importantes (27):

- ⚙ Es nula la suma de las desviaciones verticales de los puntos a partir de la recta de ajuste.

$$\sum (Y - \hat{Y}) = 0.$$

Ecuación 11. Suma de las desviaciones verticales.

- ⚙ Es mínima la suma de los cuadrados de dichas desviaciones. Ninguna otra recta daría una suma menor de las desviaciones elevadas al cuadrado:

$$\sum (Y - \hat{Y})^2 \rightarrow 0 \text{ (mínima).}$$

El procedimiento consiste entonces en minimizar los residuos al cuadrado C_i^2 .

$$\sum C_i^2 = \sum (Y^o - \hat{Y})^2$$

Ecuación 12. Expresión para minimizar los residuos al cuadrado.

Re-emplazando \hat{Y} queda:

$$\sum C_i^2 = \sum (Y^o - (a + bx))^2$$

Ecuación 13. Reemplazo de la ecuación para minimizar residuos.

La obtención de los valores de a y b que minimizan esta función es un problema que se puede resolver recurriendo a la derivación parcial de la función en términos de a y b : denominada G la función que se va a minimizar:

$$G = \sum (y - a - bx)^2$$

Ecuación 14. Derivación parcial de una función en mínimos cuadrados.

Por su importancia los mínimos cuadrados son utilizados con gran frecuencia. Es necesario señalar que el problema de mínimos cuadrados es conocido bajo diferentes nombres en varias ramas; por ejemplo en estadística se le conoce por regresión lineal por mínimos cuadrados, mientras que en ingeniería se le conoce como estimación de parámetros, filtraje o identificación de procesos (28).

1.3.3 Modelos de serie de tiempos

Una serie de tiempo es un conjunto de datos numéricos que se obtienen en períodos regulares a través del tiempo (29). Estos datos pueden ser muy variados, generalmente son usados para evaluar el comportamiento de las ventas de una empresa, o para evaluar el comportamiento de los índices de precio de un país o de un tipo de producto pero en general pueden aplicarse a cualquier negocio o área. Este comportamiento puede tener características de tipo estacional o cíclico.

Generalmente las empresas lo que hacen es evaluar periódicamente el comportamiento de su actividad y/o productos a fin de pronosticar que va a suceder en el futuro en base a lo que ha venido ocurriendo en el pasado, está sucediendo en el presente y tiene la tendencia a comportarse de la misma manera en el futuro. Este comportamiento se debe a 4 componentes: la tendencia, la variación cíclica, la variación estacional y la variación irregular (30).

La tendencia es aquella disposición a largo plazo sin alteraciones de una serie de tiempo. Esta tendencia pudiera ser de tipo lineal o no lineal, así como también creciente o decreciente, o bien una combinación de alguna de las anteriores. La segunda componente es la variación cíclica en la que a través del período de tiempo analizado se producen ascensos y descensos en varias oportunidades. Este tipo de comportamiento es muy asociado a variaciones de carácter económico (30).

La variación estacional, por su parte, tiene como característica de variación regular dentro de un año y que a su vez se repite cada año, casos típicos son la producción de algunas

frutas y/o comestibles o ventas asociadas a productos como ropa de temporada. La última componente es la componente irregular que adiciona las características anteriores pero además tiene comportamientos extraños e imprevisibles que se dan generalmente en el corto plazo.

Para poder pronosticar cuál es el comportamiento futuro de una variable en función a estas características de comportamiento, es necesario poder representarlo matemáticamente. Existen métodos llamados “métodos de suavizamiento” porque su objetivo es suavizar la variación causada por el componente irregular de la serie de tiempo. Estos métodos son: promedios móviles, promedios móviles ponderados y suavizamiento exponencial. A continuación se describe uno de ellos.

➤ **Suavizamiento Exponencial Simple**

El suavizamiento exponencial emplea un promedio ponderado de la serie de tiempo pasada como pronóstico; es un caso especial del método de promedios móviles ponderados en el cual sólo se selecciona un peso o factor de ponderación: el de la observación más reciente (31). En la práctica comenzamos haciendo que F_1 , el primer valor de la serie de valores uniformados, sea igual a Y_1 , que es el primer valor real de la serie. El modelo básico de suavizamiento exponencial es el siguiente:

$$F_{t+1} = \alpha Y_t + (1 - \alpha)F_t$$

Ecuación 15. Modelo básico de suavización exponencial.

Donde:

- F_{t+1} = pronóstico de la serie de tiempo para el período t+1
- Y_t = valor real de la serie de tiempo en el período t
- F_t = pronóstico de la serie de tiempo para el período t
- α = constante de suavizamiento, $0 \leq \alpha \leq 1$

En base a lo anterior, el pronóstico para el período dos se calcula de la siguiente manera:

$$F_2 = \alpha Y_1 + (1 - \alpha)F_1$$

Ecuación 16. Cálculo de pronóstico para el período 2 en suavización exponencial.



Después de analizados y caracterizados varios de los métodos matemáticos para el análisis estadístico, el autor de la presente investigación y el cliente consideran que, los pronósticos para analizar los datos referentes a los modelos de propagación en la herramienta informática Andrómeda deben basarse en el método causal de regresión lineal simple, ajustado por mínimos cuadrados.

1.4 Herramientas y Tecnologías

En el desarrollo de aplicaciones informáticas son necesarias un conjunto de herramientas que apoyen las fases de la metodología seleccionada. Para el desarrollo exitoso del módulo Andromeda_Grapher se propone continuar con la línea base de desarrollo ofrecido por la herramienta Andrómeda. Seguidamente se realiza una breve descripción de estas herramientas:

1.4.1 Lenguaje de programación

Para la implementación de Andromeda_Grapher se empleará Java debido a que es un lenguaje de desarrollo de propósito general, para realizar todo tipo de aplicaciones. Dentro de las numerosas cualidades que avalan su selección destaca el hecho de facilitar la rapidez del desarrollo, así como garantizar la reutilización de códigos y reducir el proceso de depuración del mismo. Además es el lenguaje base sobre el cual ha sido desarrollado la herramienta Andrómeda.

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 (32).

Uno de los principales elementos que hacen diferente a Java de los demás lenguajes de programación es su portabilidad, pues permite que la misma aplicación escrita con este lenguaje se pueda ejecutar en cualquier máquina, independientemente del sistema operativo y de la configuración de hardware, gracias a la máquina virtual sobre la que se ejecutan los programas. Java es apoyado además por un gran número de librerías que son soportadas por diversas empresas como IBM y Oracle, además de encontrarse otra



gran cantidad desarrolladas por terceros, simplificando el desarrollo y la implementación de las aplicaciones (33).

1.4.2 Entorno Integrado de Desarrollo

Un Entorno de Desarrollo Integrado, traducido del inglés *Integrated Development Environment* (IDE), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien poder utilizarse para varios. Así Dev C++, es un IDE para el lenguaje de programación C++; WebDevStudio, lo es en línea para el lenguaje de programación C/C++ (34).

Para el desarrollo de la solución se seleccionó como IDE el NetBeans v7.4. Dentro de las características más importantes de esta versión destacan notables mejoras en el tratamiento del control de versiones con herramientas como Subversion y Git. Propone mejoras en el soporte de HTML 5 y su integración con Java EE, así como compatibilidad con las nuevas características de JDK 8. Posee además, todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, además posee integridad con lenguajes como C/C ++, PHP y JavaScript.

También influyó sobre su selección, el hecho de ser una herramienta libre y de código abierto y permitir la integridad con el lenguaje Java. Esto facilita el desarrollo utilizando funcionalidades como; completamiento de código, coloreo de sintaxis, utilización y edición de los componentes visuales de forma sencilla e incluye también el control de versiones, lo cual representa una ventaja ya que permite administrar las diferentes versiones del código fuente (22).

1.4.3 Librería para el Desarrollo

En el proceso de desarrollo de un software es necesario mostrar datos con mucha frecuencia, ya que se recopila demasiada información, que a veces es imposible entender solamente a base de textos y números. El uso de gráficos fáciles de entender y que permitan la obtención de conclusiones certeras a partir de ellos es una solución a esta problemática. Para realizar este proceso existen disímiles librerías gráficas, estas tienen la capacidad de generar imágenes para facilitar los fines antes descritos en gráficas de



barras, líneas, pastel, área y dispersión. Seguidamente se describen varias librerías y se selecciona la que se emplea en esta investigación.

➤ **Librería JpGraph**

JpGraph, es una librería que permite crear gráficos matemáticos y estadísticos de manera sencilla. Con JpGraph se pueden obtener gran cantidad de tipos de gráficas para todas las necesidades. Dispone de una amplia documentación y la existencia de muchos tutoriales para aprender a manejarla. Posee opciones configurables como distintos tipos de colores, leyendas, tipologías e imágenes de fondo. Una de las facilidades de su uso es poseer licencia LGPL (*Licencia Pública General Reducida de GNU, del inglés GNU Lesser General Public License*) permitiendo usar la herramienta en cualquier software propietario. Sin embargo, esta librería solo está disponible para el lenguaje PHP, no siendo este el que se utiliza en la presente investigación (35).

➤ **Librería FusionCharts Free**

Esta librería se presenta como un recurso para graficar todo tipo de datos usando componentes Flash. Con ella se pueden crear múltiples gráficos animados. Soporta lenguajes de programación tales como PHP, Python y HTML. FusionCharts Free no solo ofrece amplia variedad de gráficos, sino que incluye en su paquete la incorporación de una ayuda muy completa con diversos ejemplos y demostraciones. Su empleo está pensando para páginas web (36).

➤ **Librería JfreeChart**

JFreeChart⁴ esta implementada en el lenguaje Java y posee una licencia LGPL, lo que permite que el software derivado pueda licenciarse de forma libre o privativa y se distribuye gratuitamente conjuntamente con su documentación.

Esta librería no es un sistema, sino una biblioteca que los programadores pueden utilizar en sus programas para la construcción de variados tipos de gráficas (gráficos de barra, de pastel, gráficos de línea, diagramas de dispersión, gráficas de tiempo, etc.). Es 100% libre para Java, lo que hace que sea más fácil para los desarrolladores mostrar gráficos

⁴ <http://www.jfree.org/jfreechart/>

de calidad en sus aplicaciones, ya sean web o de escritorio. Entre las características principales de esta biblioteca tenemos:

- ⚙ Posee una API consistente y bien documentada con soporte para un amplio rango de tipos de gráficos.
- ⚙ Permite exportar gráficos a diferentes formatos y a la hora de su uso la personalización de las gráficas a mostrar cambiando la orientación de los ejes, el tamaño de las unidades de medida, entre otras muchas funcionalidades.
- ⚙ Producto del lenguaje en el que fue implementado puede utilizarse tanto en aplicaciones web como en aplicaciones de escritorio.

JFreeChart es Open Source⁵, más específicamente, Software Libre, haciendo uso de Jcommon, que no es más que una colección de clases de utilidad la cual incorpora una gran cantidad de clases de apoyo para la creación de gráficas, que pueden ser muy ventajosas a la hora de realizar los trabajos con las mismas. Ha sido ampliamente utilizado por numerosas compañías, por lo que forma parte de gran cantidad de productos disponibles en la red (37).

➤ **Selección de la librería**

Considerando las características actuales de la herramienta Andrómeda, como aplicación Desktop, a la cual se le adicionarán nuevas funcionalidades, específicamente la de comparar modelos de propagación a través del uso de gráficos, se escoge la librería JfreeChart debido a que no hace falta cambiar absolutamente nada en la estructura actual del sistema al estar implementados en el mismo lenguaje de programación. JpGraph y FusionCharts Free se presenta como soluciones útiles para otro tipo de aplicaciones, principalmente para la obtención de aplicaciones web.

1.4.4 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el

⁵ **Open Source** (Pronunciado: open-so-urs) son programas o aplicaciones el cual el código original es revelado y disponible al público en general. (71)

almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación (38).

SQLite es un sistema de gestión de bases de datos relacional contenida en una relativamente pequeña biblioteca en C. SQLite creado por D. Richard Hipp, es un dominio gratuito, público, herramienta de código abierto visual utilizada para crear, diseñar y editar archivos de base de datos. La biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción (39).

Partiendo de la premisa de que el cúmulo de información que maneja el módulo es relativamente pequeño, se usará SQLite como gestor de datos local para el almacenamiento de la información entre el sistema y la base de datos. Dentro de los beneficios que, en el marco del presente trabajo se le confieren a SQLite, destacan el no requerir configuración, no necesitar el uso de servidores, su portabilidad y su compatibilidad con el lenguaje Java y casi cualquier Sistema Operativo.

SQLite garantiza la persistencia de los datos ya que permite guardar como un solo fichero toda la información asociada y cargarlo en la memoria de la máquina, permitiendo realizar de manera efectiva todas las peticiones de los usuarios. Es una herramienta desarrollada con tecnologías libres, multiplataforma y responde a las peticiones que pretende llevar a cabo Cuba; para garantizar la plena soberanía tecnológica. Estas son algunas de las razones que sirvieron de soporte para elegir SQLite como el gestor de base de datos, ya que constituye la herramienta apropiada para el manejo y control de la información que se desea almacenar (39).

1.5 Selección de la Metodología de Desarrollo de Software

Todo proceso de creación de un software debe estar basado en una metodología de desarrollo para que su implementación tenga un resultado exitoso, de no ser así, se tendrá como resultado final un producto que de seguro no contará con la calidad esperada y los clientes se mostraran insatisfechos con la solución dada. Según la Real Academia de la Lengua Española una metodología de desarrollo de software no es más que un

conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal (40).

Al considerar las valoraciones de **Beck (2012)** se puede constatar que las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Dentro de las más conocidas se encuentran las metodologías tradicionales, centradas específicamente en el control del proceso. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo para aquellos proyectos de gran tamaño (respecto a tiempo y recursos).

Como respuesta a la problemática de que las metodologías tradicionales no ofrecen soluciones factibles para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, surgieron otras metodologías que tratan de adaptarse a la realidad del desarrollo de software; las metodologías ágiles. Estas son menos orientadas al documento, exigiendo menor cantidad de documentación para una tarea dada. De muchas maneras son más bien orientados al código, siguiendo un camino que dice que la parte importante de la documentación es el código fuente (41).

A continuación se realizará una explicación breve de algunas de las metodologías más utilizadas en la Universidad de las Ciencias Informáticas (UCI).

1.5.1 Rational Unified Process (RUP)

RUP es una metodología de desarrollo de software formal, orientadas a objetos, con un ciclo de vida espiral. El Proceso Racional Unificado (*RUP*, por sus siglas en inglés) divide en 4 fases el desarrollo del software (42):

- ⚙ **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- ⚙ **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.
- ⚙ **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ⚙ **Transición**, El objetivo es llegar a obtener la liberación (release ⁶) del proyecto.

⁶ Release: Liberación, estreno, lanzamiento. Versión de un software que es distribuida a los clientes.



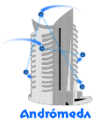
Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

RUP se caracteriza por ser dirigido por casos de uso, donde estos definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio. Se centra en la arquitectura, peculiaridad esta que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes para comprenderlo, desarrollarlo y producirlo de una forma eficaz. Además, es iterativo e incremental, donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo (43).

1.5.2 Programación Extrema

Programación Extrema (XP, del inglés *eXtreme Programming*, y en lo adelante XP) es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, equipos pequeños y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya peculiaridad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito. XP está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. 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.

Esta metodología promueve el trabajo en equipo y la igualdad y cooperación entre sus integrantes. La propiedad compartida del código permite a cualquier persona del equipo modificar código, mejorarlo o extenderlo, para facilitar esta tarea los programadores cumplen con un estándar de codificación que al mismo tiempo mejorará el mantenimiento del código en el futuro. La integración continua provee un método para que todos los integrantes siempre trabajen en la última versión del código de manera que puedan aprovechar los nuevos desarrollos y garantizar que las mejoras y adiciones aún cumplen con las pruebas (44).



XP define historias de usuario (en inglés, *User Stories*) como la base del software a desarrollar. Estas historias son escritas por el cliente y describen las interacciones entre los clientes y el sistema, por lo general son complementadas con otro tipo de descripción. A partir de ellas y de la arquitectura que se utilizará se planifican las entregas, así como los objetivos de cada una y las iteraciones con las cuales contará.

Una característica que distingue a la Extreme Programming es la programación en parejas, con el objetivo de que el código sea revisado y validado antes de ser escrito. Con esta posibilidad se facilita la transferencia de conocimientos de programación entre los miembros del equipo. Los programadores conversan mejorando así el flujo de información y la dinámica del equipo (45) (46).

1.5.3 SCRUM

Scrum es una de las más conocidas metodologías ágiles para la gestión de proyectos. Surge a mediados de los 80 y se desarrolla originalmente en el sector de las Tecnologías de la Información y las Comunicaciones, pero es aplicable en cualquier proyecto en el que exista una lista de funcionalidades o bloques de trabajos por realizar, un entorno complejo con requisitos cambiantes y un equipo de desarrollo asignado a dicha tarea. Esta metodología se basa en un enfoque iterativo, donde cada iteración se denomina Sprint. La diferencia con las iteraciones en cascada es que al final de cada Sprint obtenemos un producto entregable que se va incrementando en sucesivos Sprints, priorizándose aquellos aspectos que aportan mayor funcionalidad y valor al dueño del producto (47).

Otra de las características más importantes son las reuniones a lo largo de la ejecución del proyecto. Son conocidas como *Daily Scrum Meeting*, o reuniones diarias. Se trata de una reunión operativa, informal y ágil, que no ha de durar más de 30 minutos en la que cada integrante del equipo ha de responder 3 preguntas: ¿Qué tarea ha realizado desde la última reunión? ¿Sobre qué va a trabajar en el día actual?, y la Identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance del proyecto, traducida a ¿Qué ayuda necesito? (48).

Con esta metodología el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. De igual forma le permite en cualquier momento



realignar el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema. Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades (47). En el [Anexo 2](#) se puede observar las fases de desarrollo de esta metodología.

1.5.4 Metodología Seleccionada

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas, como las mencionadas anteriormente, que inciden en distintas dimensiones del proceso de desarrollo. Aunque es válido señalar que toda metodología ha de ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.).

RUP es conocido por la robustez de su proceso de desarrollo a largo plazo, mientras SCRUM propone una metodología ágil que enfatiza en reuniones diarias, algo que es innecesario debido a la pequeña dimensión del equipo de desarrollo que siempre está en contacto. Se decide seleccionar XP, algunas de las más relevantes características de esta metodología, y que sirvieron de base para su selección son las que se exponen a continuación:

- ⚙ Las dimensiones del proyecto son pequeñas.
- ⚙ El período de desarrollo es corto, limitado a solamente 5 meses de trabajo.
- ⚙ XP asume que la planificación nunca será perfecta.
- ⚙ El cliente forma parte del equipo de desarrollo.
- ⚙ Promueve un entorno físico con un ambiente que permite la comunicación y colaboración entre todos los miembros del equipo.

1.6 Herramienta CASE

Las aplicaciones informáticas que facilitan el trabajo dentro del ciclo de desarrollo del software son conocidas como herramientas CASE (del inglés: *Computer Aided Software Engineering*) y se emplean para aumentar la productividad del desarrollo del software disminuyendo los tiempos de construcción y el costo de los mismos.



Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm para UML (VP-UML v8.0), una de las líderes del mercado de las herramientas CASE. Esta es una herramienta multiplataforma que utiliza UML⁷ como lenguaje de modelado, muy fácil de usar; su ambiente gráfico agradable brinda muchas facilidades al usuario. Visual Paradigm permite la integración con MS Visio, integración IDE con Eclipse, NetBeans y otros. Entre sus nuevas características se incluyen el modelado colaborativo con CVS y Subversión e interoperabilidad con modelos UML a través de XML⁸. Incorpora soporte para trabajo en equipo, lo que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros (49).

La herramienta ofrece un conjunto completo de instrumentos que pone en manos de 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. Facilita además, un ahorro considerable de tiempo en la ejecución del proyecto, debido a la interoperabilidad que existe entre sus diagramas, ya que es capaz de exportarlos de un modelo a otro con mucha facilidad; además de permitir 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 (49).

1.7 Conclusiones parciales

Se puede constatar con el estudio realizado en el presente capítulo, que existen diversas clasificaciones de los modelos de propagación. De forma general, se clasifican 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. Se estudiaron además, un grupo importante de

⁷ El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. (75)

⁸ XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

métodos matemáticos para el análisis estadístico, y se evidenció que para el objetivo que se persigue en esta investigación el modelo causal de regresión lineal simple es el adecuado para su implementación, avalado por referencias actualizadas en esta área de la ciencia moderna.

En el resto del capítulo se describieron y seleccionaron las herramientas y tecnologías a ser usadas en el desarrollo de la propuesta de solución: se determinó usar como lenguaje de programación Java, utilizando como IDE NetBeans v7.4 y como sistema gestor de base de datos SQLite; además de seleccionar JfreeChart como la librería gráfica apropiada para conseguir de manera óptima el objetivo general de la investigación. El desarrollo será controlado y dirigido mediante la metodología XP y como herramienta CASE se utilizará Visual Paradigm.



2. Descripción de la solución propuesta

2.1 Introducción

El objetivo fundamental del presente capítulo es describir las funcionalidades de la propuesta de solución, es decir, determinar las propiedades o cualidades que deberá tener el módulo para darle cumplimiento a dichas funcionalidades, por lo que se identifican los requerimientos no funcionales que darán paso a la conformación de las historias de usuarios a implementar. Para dar cumplimiento a este objetivo haciendo uso de la metodología XP se abarcarán las etapas de Exploración y Planificación con el fin de conocer el alcance, estimar los tiempos de entrega de cada iteración y los artefactos que se generan.

2.2 Propuesta de solución y representación gráfica del sistema

Como resultado de la investigación realizada se propone la implementación de un grupo de funcionalidades para la herramienta Andrómeda, cuyo objetivo fundamental es facilitar la toma de decisiones con respecto a la ubicación de las WSA en el interior de edificios. La misma utilizará uno de los datos generados por los modelos de propagación implementados en la herramienta, que es la distancia existente entre transmisor y receptor. Con este dato se podrá realizar el análisis, que facilitará a los diseñadores de WSA, la obtención de predicciones y conclusiones sobre el comportamiento de los diferentes modelos de propagación.

Debido a que la arquitectura del software aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los elementos definidos en los pasos posteriores que forman parte del diseño, a continuación y para una mejor comprensión de lo que se quiere realizar se propone un diagrama de la arquitectura del módulo. Su objetivo fundamental es aportar elementos que ayuden a la toma de decisiones. Para lograr este objetivo la arquitectura de software construye abstracciones, materializándolas en forma de diagramas comentados.

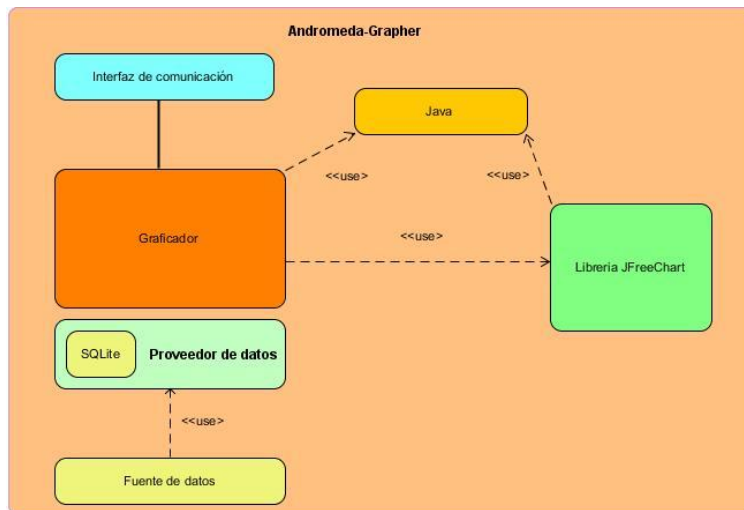


Figura 7 Modelo de arquitectura del módulo.

En la implementación se ha de utilizar un contenedor principal, haciendo uso de la librería JFreeChart, ambos se han de comunicar, utilizando Java como lenguaje en el que será implementado el módulo. Los datos se proveerán desde SQLite como fuente de datos. Mientras que la interfaz facilitará la interacción con el componente, ya que será la puerta de entrada para interactuar con el módulo.

Finalmente, y para facilitar la comprensión de lo que se pretende se muestra una representación gráfica de la propuesta (Fig. 8).

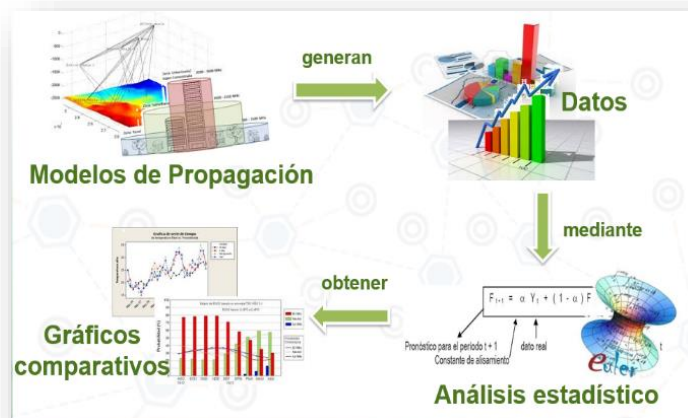


Figura 8 Representación gráfica de la propuesta de solución.

2.2.1 Usuarios del Sistema

Los usuarios del sistema son todas aquellas personas o sistemas que interactúan con este, con el objetivo de obtener un resultado específico. Andrómeda, como aplicación de escritorio que es, no presenta restricciones de acceso a ciertas funcionalidades para un grupo específico de usuarios. Todos tienen los mismos privilegios sobre las funcionalidades internas de la aplicación. Los externos, como es lógico, dependen de las restricciones clásicas que imponen los sistemas operativos, tales como el acceso a documentos de otros usuarios, las preferencias independientes para cada sesión, etc.

Tabla 2. Usuarios del sistema

| Usuarios del sistema | Justificación |
|----------------------|---|
| Diseñador de WSAN | <p>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.</p> <p>Se encarga de manejar los datos que serán objetos del proceso de obtención de los gráficos.</p> <p>Es el responsable del proceso de graficar los datos de las distintas formas.</p> <p>Es el responsable de guardar la imagen de los gráficos en un archivo.</p> <p>El usuario podrá sacar conclusiones y podrá realizar predicciones de los comportamientos de los diferentes modelos de propagación de acuerdo al resultado obtenido del proceso de graficado.</p> |

2.3 Lista de Reserva del Producto

La lista de reservas del producto, según lo define la metodología XP, está compuesta por los requerimientos 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.

1. Permitir seleccionar el tipo de gráfico para el análisis.
2. Importar datos desde el servidor de base de datos usando SQLite.

3. Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado.
4. Obtener gráfico a partir de la selección del usuario.
 - 4.1 Obtener gráfico de barras.
 - 4.2 Obtener gráfico lineal.
 - 4.3 Obtener gráfico dispersión.
5. Exportar gráficos generados en diferentes formatos para su posterior análisis.
 - 5.1 Exportar gráfico en formato .PNG.
 - 5.2 Exportar gráfico en formato .PDF.
 - 5.3 Exportar gráfico en formato .SVG.
 - 5.4 Personalizar los gráficos generados.

2.4 Aspectos no Funcionales del Sistema

Los aspectos no funcionales son propiedades o cualidades que debe tener el sistema. Establecen los servicios que se espera que el sistema proporcione y las restricciones bajo las que funcionará. Son aquellas características que permiten que el producto final sea usable, rápido, o qué software o hardware es necesario para que funcione. Pueden agruparse de acuerdo a diversas clases como rendimiento, usabilidad, seguridad, software o hardware.

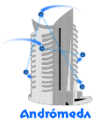
Para lograr una calidad máxima en el sistema se listaron las siguientes cualidades y propiedades.

Portabilidad: Necesidad de que el sistema funcione en diferentes sistemas operativos. Desde las distribuciones de Windows hasta las distribuciones de GNU/Linux.

Usabilidad: El diseño del módulo se guiará por los caminos de la integración de los aspectos relacionados con la usabilidad. Esta concepción permitirá al sistema ganar en eficiencia y podrá ser usado lo mismo por un diseñador experimentado en el tema, que por una persona de poca experiencia en el diseño de WSAN.

Rendimiento: La herramienta deberá ser capaz de graficar los datos referentes a los modelos de propagación en un lapso de tiempo inferior a 1 minuto. Se hace necesario que la velocidad de procesamiento de la información sea rápida.

Aspectos Legales: La herramienta una vez desarrollada será de código abierto y licencia GPL.



Software:

- ⚙ La herramienta deberá funcionar en los sistemas operativos Windows (XP o superior), MAC, Solaris y las distribuciones de GNU/Linux.
- ⚙ Se requiere de la Máquina Virtual de Java versión 1.6 o superior.
- ⚙ Se requiere de la librería JfreeChart, y esta a su vez hace uso de JCommon.

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.

Apariencia o interfaz externa:

- ⚙ El diseño de la interfaz debe ser sencillo e intuitivo para el diseñador.
- ⚙ El tamaño de la fuente debe ser fácil de leer.

Diseño e Implementación:

- ⚙ La herramienta debe ser implementada usando el lenguaje de programación Java en su versión 1.6.
- ⚙ Hacer uso de la biblioteca JfreeChart en la visualización de los gráficos.
- ⚙ Para el desarrollo de la herramienta se empleará la versión 7.4 del IDE NetBeans.

2.5 Fase de Exploración

En esta fase de la metodología XP, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Una vez que los clientes entregan su propuesta al equipo de trabajo, comienza el análisis en grupo, las horas en los pizarrones, las tormentas de ideas y la conceptualización del software. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (45).

2.5.1 Historias de Usuarios

El sistema es desarrollado para el cliente, por tal motivo el usuario es quien decide que tareas realizará la aplicación. Las historias de usuarios (HU) son utilizadas como

herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que realizará el sistema; la redacción de los mismos se realiza bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles (50).

XP y otras metodologías ágiles prefieren comunicación cara a cara sobre una amplia documentación y una rápida adaptación al cambio en lugar de la fijación en el problema. Las HU representan pequeños trozos de valor del negocio que se pueden implementar en un período de días o semanas, por lo que es más fácil estimar el esfuerzo de desarrollo. Además de ser adecuadas en proyectos donde los requerimientos sean volátiles y poco conocidos (51).

Seguidamente se describe una de las HU que el cliente propone para la solución informática de la presente investigación. El resto de las HU generadas se encuentran detalladas en el [Anexo 3](#).

Tabla 3. HU 1: Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite.

| Historia de Usuario | |
|--|--|
| No.: 1 | Nombre: Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite. |
| Usuario: Todos | |
| Prioridad en el Negocio: Alta | Nivel de Complejidad: Media |
| Estimación: 2 días | Iteración Asignada: 1 |
| <p>Descripción: La herramienta deberá proporcionarle al usuario una opción que le permita seleccionar el tipo de gráfico que desee visualizar para realizar el análisis. Los cuales podrán ser gráficos de línea, de barra y de dispersión.</p> <p>Además de permitir importar los datos almacenados en la BD sobre estas características de los modelos, con los que se realizará el análisis.</p> | |
| <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Permitir seleccionar el tipo de gráfico para el análisis" y a la funcionalidad "Importar datos desde el servidor de BD SQLite". Se puede seleccionar uno, dos, o los tres tipos de gráficos para visualizar.</p> | |



2.6 Fase de Planificación

Para la metodología XP, el proceso de planificación es aquel en el que el cliente establece la prioridad de cada HU, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días, aunque no por eso ha de restársele importancia.

Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las HU que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. 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. (52)

2.6.1 Estimación de esfuerzos por historias de usuario

En la fase de planificación el artefacto que se genera es la estimación por esfuerzo. Es aquí donde los clientes establecen la prioridad de cada una de las HU y es donde el programador, o el grupo de programadores realizan una estimación del esfuerzo que se puede tomar en implementarlas. En esta fase se recomienda tener en cuenta los requerimientos del sistema para realizar una correcta estimación del tiempo de desarrollo del producto.

La estimación por esfuerzo consiste en asignarle puntos a las HU y cada uno de estos puntos son equivalentes a una semana. Para realizar esta estimación los programadores se basan en la complejidad que puedan tener las HU y en el tiempo hábil para el desarrollo de la aplicación, para la implementación de las funcionalidades se analizarán cada una de las HU para la correcta estimación.

Tabla 3. Estimación por esfuerzo para el desarrollo de la solución.

| Iteraciones | Orden de las Historias de Usuario a implementar | Cantidad de tiempo de trabajo (semanas) |
|--------------------|---|--|
| 1 | Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite. | 1 |
| 1 | Obtener gráfico de barras partir de la selección del usuario. | 1 |
| 1 | Obtener gráfico lineal a partir de la selección del usuario. | 1 |
| 2 | Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado. | 3 |
| 2 | Obtener gráfico de dispersión a partir de la selección del usuario. | 1 |
| 2 | Exportar gráficos generado en formato.png. | 1 |
| 3 | Exportar gráficos generados en formato .svg. | 1 |
| 3 | Exportar gráficos generados en formato .pdf. | 1 |
| 3 | Personalizar los gráficos generados. | 1 |
| Total | | 11 |

2.6.2 Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las HU asignadas a la iteración. Como las HU no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance (53).

Los elementos que deben tomarse en cuenta durante la elaboración del plan de la iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada



una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores (51).

Tabla 4. Plan de Iteraciones.

| Iteración | HU a implementar en cada iteración | Cantidad de tiempo de trabajo |
|------------------|--|--------------------------------------|
| Iteración 1 | <p>HU 1: Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite.</p> <p>HU 3: Obtener gráfico de barras partir de la selección del usuario.</p> <p>HU 4: Obtener gráfico lineal a partir de la selección del usuario.</p> | 12 días |
| Iteración 2 | <p>HU 2: Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado.</p> <p>HU 5: Obtener gráfico de dispersión a partir de la selección del usuario.</p> <p>HU 6: Exportar gráficos generado en formato .PNG.</p> | 19 días |
| Iteración 3 | <p>HU 7: Exportar gráficos generados en formato .SVG.</p> <p>HU 8: Exportar gráficos generados en formato .PDF.</p> <p>HU 9: Personalizar los gráficos generados.</p> | 14 días |

2.6.3 Plan de Entregas

El cronograma de entregas establece qué historias de usuarios serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión "Juego de planeamiento" (*Planning game*), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planeamiento, *Planning meeting* o *Planning workshop*) Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuarios. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario (53).

Tabla 5. Plan de Entrega.

| Entregable | Final 1ra iteración 4 ^{ra} semana 29 de marzo de 2015 | Final 2da iteración 3 ^{ra} semana de abril de 2015 | Final 3era iteración 1 ^{ra} semana de mayo de 2015 |
|--------------------------|--|---|---|
| Módulo Andromeda_Grapher | versión 1.0 | versión 2.0 | versión 3.0 |

2.7 Diseño

Con la guía de la metodología XP se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: *“supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos”*. Sin dudas, el objetivo fundamental en XP es el establecimiento de prácticas especializadas que incidirán sustancialmente en el resultado final, al lograr tratando de mantener su simplicidad, un sistema robusto y reutilizable (53).

2.7.1 Patrones de diseño

Los patrones de diseño son propuestas de soluciones a problemas que surgen con frecuencia en el proceso de desarrollo de un software. Estos brindan una solución a situaciones similares que se encuentren probadas y bien descritas (54). Entre los principales objetivos de los patrones de diseño de encuentran:

- ⚙ Brindar elementos reusables en el diseño de un software.
- ⚙ Evitar intentar solucionar un problema ya conocido o previamente solucionado.
- ⚙ Estandarizar el modo en que se realiza el diseño de software.

Los patrones de diseño se pueden agrupar en dos grandes grupos; los patrones GRASP (del inglés; *General Responsibility Assignment Software Patterns*), que son patrones generales de software para asignación de responsabilidades y los patrones GOF (del

inglés; *Gang of Four*), que describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño con modelado UML. Estos patrones se agrupan en las siguientes categorías: creacionales, estructurales y de comportamiento (55).

Para el desarrollo de la herramienta Andrómeda, y en particular del módulo *Andromeda_Grapher* se emplearon los patrones de diseño GRASP: **Experto**, **Creador**, **Alta Cohesión** y **Bajo Acoplamiento** y los patrones GoF: **Singleton** y **Observer**. Los ejemplos se pueden apreciar en el [Anexo 4](#).

Patrón Experto: Este patrón resuelve el problema de asignar una responsabilidad al experto en información o sea a la clase que tiene la información necesaria para realizar la responsabilidad (56). Un ejemplo de este patrón en la herramienta Andrómeda se evidencia en las clases **Element**, **Device**, **Material** (*Fig. 22*, *Fig. 23*), la cual tiene asociado un conjunto de atributos y funcionalidades que las relacionan de forma directa con la entidad que representa. Esta clase contiene toda la información necesaria de la tabla que representan en la base de datos.

Patrón Creador: El patrón creador resuelve el problema de asignar responsabilidades relacionadas con la creación de objetos (56). Su propósito fundamental es encontrar un creador que debe conectar con el objeto producido en cualquier evento. Una de las clases de la herramienta que implementa dicho patrón es la clase **MultiWallModel** (*Fig. 24*).

Patrón Alta cohesión: Este patrón resuelve el problema de asignar una responsabilidad de manera que la cohesión permanezca alta. La cohesión es una medida de la fuerza con que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión (56). Para el diseño de clases en la herramienta y el módulo *Andromeda_Grapher* se tuvo en cuenta este patrón.

Patrón Bajo acoplamiento: Este patrón resuelve el problema de asignar una responsabilidad de manera que el acoplamiento permanezca bajo. El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, a otros elementos. Un elemento con bajo (o débil) acoplamiento no depende

de demasiados elementos (56). El diseño de clases en el módulo y la herramienta de forma general tuvo en cuenta este patrón.

Patrón Singleton (instancia única): El uso de este patrón del tipo creacional, garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Unas de las clases de la herramienta que implementa dicho patrón es la clase controladora **Andromeda** y **ModelControl** (Fig. 25, Fig. 26). Esta clase contiene la información de los dispositivos a emplear en el diseño, así como la instancia de la clase para gestionar la información de la Base de Datos. En el módulo **Andromeda_Grapher** particularmente se puede evidenciar el uso de este patrón en la clase **regresión** (Fig. 28). El uso de este patrón garantiza que en todo momento se trabaje con la misma instancia de dicha clase en toda la aplicación.

Patrón Observer (observador): Este patrón del tipo comportamiento, define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Este patrón se ve reflejado en la herramienta de la siguiente forma: La clase **Andrómeda** hereda de la clase **Observable** y es la encargada de almacenar y notificar a los observadores asociados a ella cada vez que ocurra un cambio en la misma. Ejemplos de los observadores que están asociados a esta clase se encuentran la clase **SensorLibraryTopComponent**, **ExportDevices** y **DeviceManagement**, estas clases visualizan la información de los dispositivos a partir de la clase **Observable** **Andrómeda**.

2.7.2 Modelo de datos

En informática se define como un modelo de base de datos al conjunto de ideas lógicas utilizadas para representar la estructura de datos y las relaciones entre ellos dentro de la base de datos. Estos modelos se pueden agrupar en dos categorías: modelos conceptuales y modelos de ejecución. “*El modelo conceptual se enfoca en la naturaleza lógica de la representación de datos. Por consiguiente, este modelo está comprometido con lo representado en la base de datos, y no en cómo está representado. Los modelos conceptuales incluyen el modelo de Entidad-Relación (E-R) y el modelo orientado a objetos*” (57).

Dittrich (1994) lo define como “conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras, tipos de datos, operaciones y restricciones” (58).

Dos componentes conforman el modelo de datos; un componente estático, el cual se relaciona con el lenguaje de definición de datos (LDD) y un componente dinámico, el cual facilitará la manipulación de instancias de las estructuras de la base de datos definidas con los componentes estáticos (serán el soporte de los lenguajes de manipulación de datos LMD o DML (del inglés; *Data Manipulation Languages*) que permitirán escribir transacciones). La dinámica ofrecerá operadores o expresiones para realizar acciones sobre los datos (58).

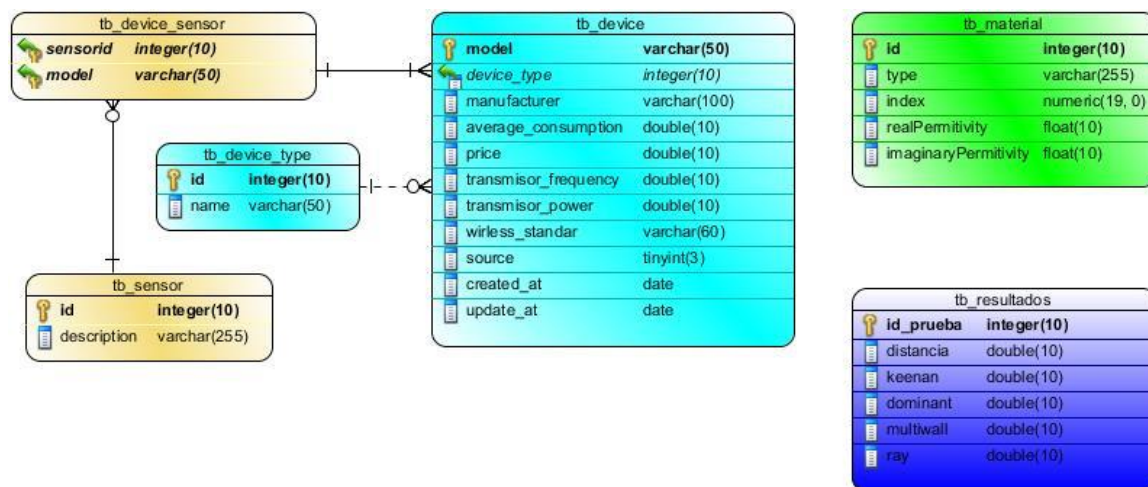


Figura 9 Modelo de Datos.

Seguidamente se realiza la descripción de las tablas que conforman la base de datos empleadas por la herramienta Andrómeda 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 censado.

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.

tb_resultados: almacena la información de la pérdida de señal para los modelos de propagación implementados en la herramienta y con los cuales se realizará el graficado en el módulo implementado.

2.7.3 Diagrama de Clases

Los diagramas de clases describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Se convierte así en la técnica más potente para el modelado conceptual de un software. El diagrama de clases de más alto nivel, será lógicamente un dibujo de los paquetes que componen el sistema, las clases se documentan con una descripción de su propósito, sus objetos que intervienen en la relación y su opcionalidad. Los diagramas de clases son el pilar básico del modelado UML; siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño) (59). En la figura 14 se muestra el diagrama de clases de diseño del módulo a implementar.

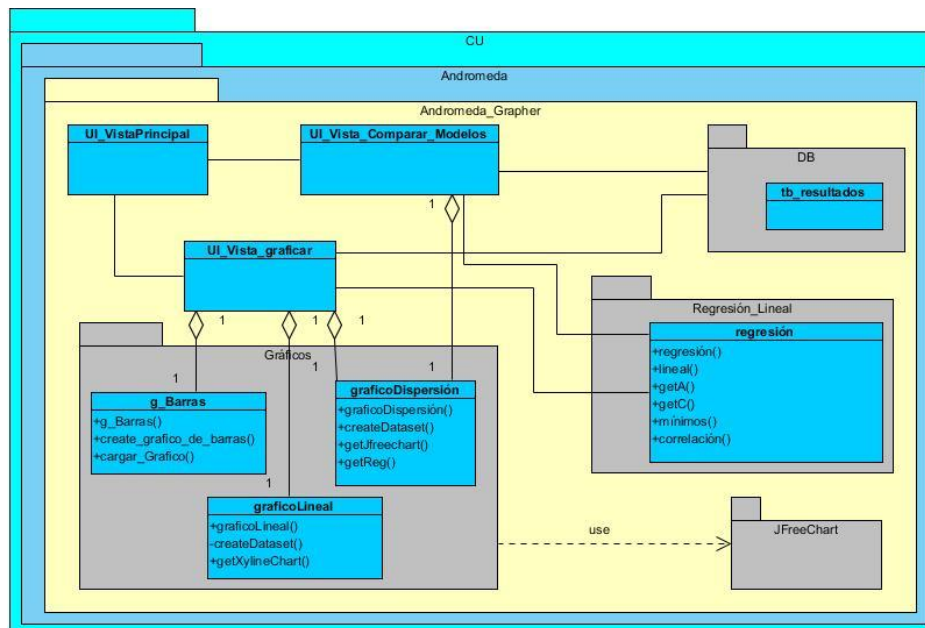


Figura 10 Diagrama de clases del diseño.



2.8 Conclusiones parciales

En el presente capítulo se abordaron los elementos más importantes de la propuesta de solución. Se presentó una descripción de esta propuesta a través del planteamiento de la lista de reserva del producto y los aspectos no funcionales del mismo, los cuales permiten definir de manera más clara y precisa las restricciones que tendrá el módulo implementado. Además se representaron aspectos como el diagrama de clases del diseño, se describieron los patrones de diseño utilizados; que son mecanismos efectivos para la reutilización del código permitiendo que se obtenga una aplicación que cumpla con las funcionalidades identificadas. Se diseñó el plan de entregas del producto, lo cual permitió una óptima organización al autor de la presente investigación y el perfeccionamiento de proyecciones futuras para el trabajo a realizar. Al concluir el capítulo se tiene un conocimiento de lo que lo que hará el módulo sin entrar en detalles de cómo lo hará, y queda el camino listo para pasar a la implementación del mismo.

3. Implementación y pruebas

3.1 Introducción

De acuerdo a la metodología seleccionada, las diferentes tecnologías y herramientas presentadas en la fundamentación teórica, en el presente capítulo se procede a describir la arquitectura de software que guiará el desarrollo de la propuesta de solución. XP, por su parte, no especifica concretamente los artefactos que se han de utilizar en la implementación de una solución que se rija por dicha metodología. Más concretamente, XP pone en manos del equipo de desarrolladores, dependiendo de sus capacidades de comunicación, la decisión de utilizar tantos tipos de diagramas de UML como crean posibles, en aras de facilitar el proceso de desarrollo. Se describe además el diseño del módulo en cuestión, los diagramas utilizados, las tareas de ingeniería generadas para cada HU y el proceso de pruebas.

3.2 Descripción de la arquitectura

La arquitectura que se emplea en el desarrollo de esta solución se encuentra basada en la que ofrece la Plataforma Netbeans (*NetBeans Platform*), ya que su aplicación permite la creación de aplicaciones modulares (60). El IDE NetBeans, desarrollado sobre la plataforma antes mencionada facilita el desarrollo de aplicaciones con las características como la que se desea implementar. Una de las numerosas ventajas que ofrece el uso de esta plataforma resulta el poder agregar módulos nuevos sin la necesidad de modificar directamente el núcleo de la aplicación, haciéndola más flexible.

3.2.1 Plataforma Andrómeda

La plataforma Andrómeda, según se define en la arquitectura (60) : *“Andrómeda 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 APIs⁹ 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.”*

⁹ **API**: del inglés: *Application Programming Interface*, es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

La plataforma NetBeans facilita las relaciones entre módulos estableciendo dependencias entre ellos. Para ello se hace uso de un mecanismo, definido con este objetivo, el cual se conoce como *Lookup*, que posibilita que cada módulo intercambie información con el resto.

El presente trabajo propone la incorporación de un nuevo módulo a la herramienta Andrómeda, que a continuación se describe:

- **Andromeda_Grapher:** Módulo encargado de realizar comparaciones entre los modelos de propagación implementados en Andrómeda, de acuerdo a características específicas de estos modelos. Permitirá mediante gráficos estadísticos poder predecir cuál o cuáles modelos se ajustan mejor de acuerdo a ciertas características.

El módulo se desarrolló enfocado a la arquitectura de la plataforma Andrómeda, lo cual facilita una mejor organización del código. La estructura de paquetes en Java se organiza: `paquetepincipal.subpaquete`, quedando conformado para el nuevo módulo el paquete:

`cu.andromeda.AndromedaGrapher`: En este paquete se almacenan las clases e interfaces que implementaran el método de regresión lineal, y la obtención de los gráficos con los datos generados por la herramienta.

El diagrama de paquetes del módulo presentado se expone a continuación:

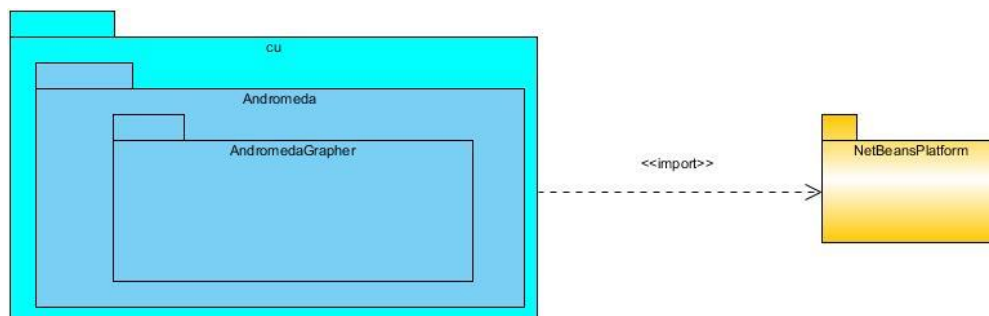


Figura 11 Diagrama de paquetes del módulo implementado.

3.2.2 Diagrama de Componentes

El diagrama de componentes muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Se utiliza para modelar

la vista estática de un sistema y muestra la organización y las dependencias lógicas entre un conjunto de componentes de software; sean estos componentes de código fuente, librerías, binarios o ejecutables.

Seguidamente se muestra el diagrama de componentes para la implementación del módulo resultado de esta investigación.

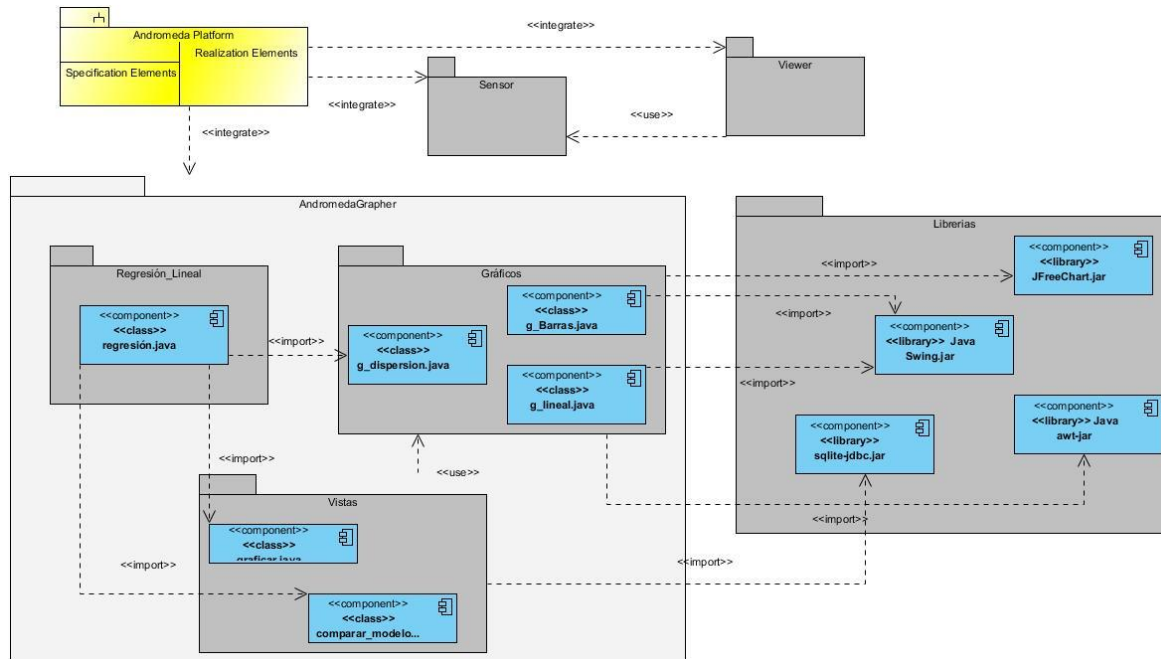


Figura 12 Diagrama de componentes del módulo AndromedaGrapher.

3.2.3 Método de Regresión Lineal simple

Según el método de regresión lineal simple, las puntuaciones de los sujetos en dos variables - una de ellas considerada como variable predictora (X) y la otra como variable de respuesta (Y) - vienen representadas por la ecuación de una línea recta. En el contexto de este trabajo se utiliza la función $f(x) = ax + c$. Los dos parámetros de la ecuación de regresión línea simple, a y c, son conocidos como el origen (o constante) y la pendiente del modelo, respectivamente. En conjunto reciben el nombre de coeficientes de la ecuación de la regresión. A continuación se muestran las ecuaciones necesarias para encontrar estos valores.

$$a = \frac{NS_{xy} - S_x S_y}{NS_{xx} - S_x S_x} \quad c = \frac{S_{xx} S_y - S_x S_{xy}}{NS_{xx} - S_x S_x}$$

Ecuación 17. Ecuaciones para encontrar la pendiente y la ordenada.

$$S_x = \sum_{i=1}^N X_i \quad S_y = \sum_{i=1}^N Y_i \quad S_{xx} = \sum_{i=1}^N X_i^2 \quad S_{xy} = \sum_{i=1}^N X_i Y_i$$

Ecuación 18. Ecuaciones para encontrar los valores de las S.

En la implementación se obtienen estos valores mediante el método *lineal* (). A continuación un fragmento del código:

```
public void lineal() {  
    sxy = sx = sy = sx2 = sy2 = 0.0;  
    for (int i = 0; i < n; i++) {  
        sx += x[i];  
        sy += y[i];  
        sx2 += x[i] * x[i];  
        sy2 += y[i] * y[i];  
        sxy += x[i] * y[i];  
    }  
    a = (n * sxy - sx * sy) / (n * sx2 - sx * sx);  
    c = (sx2 * sy - sx * sxy) / (n * sx2 - sx * sx);  
}
```

Figura 13 Método lineal de la clase regresión en la aplicación.

Una vez que sean conocidos los valores de a y c del modelo de regresión lineal simple, este puede ser utilizado como modelo predictivo, es decir, para realizar predicciones de los valores que tomará la variable de respuesta para determinados valores de la variable explicativa. Bastará para ello con sustituir en la ecuación de regresión el valor concreto de X que se quiera (X_i). Al hacerlo, se obtendrá el valor predicho para Y según la ecuación de regresión para aquellos casos que en la variable X tome el valor (X_i). Este valor es conocido de forma genérica como puntuación predicha, siendo representado simbólicamente como Y_i' .

Para realizar el análisis inicial que presenta la investigación se utilizó una base de datos que contiene más de 50 medidas de pérdidas de señal entre el transmisor (Tx) y el receptor (Rx) a diferentes distancias y con los modelos de propagación implementados en la herramienta Andrómeda. Obtenidas con dispositivos reales de medición de la señal para estos escenarios. En la tabla ([Anexo 9](#)) se muestra el conjunto de datos utilizados para la aplicación del método de regresión lineal a una muestra de estos.

Si la relación lineal de dependencia entre X y Y fuera exacta, las observaciones se situarían a lo largo de una recta. De ser así se estaría contando con el mejor ajuste de la

recta, y por tanto con una solución correcta, o más cercana a la realidad. Pero si la dependencia de X y Y es estocástica, o sea, no existe una dependencia funcional perfecta, entonces las observaciones no se alinearan a lo largo de una recta, sino que formarán una nube de puntos, mejor representados en el diagrama de dispersión. ([Anexo 5](#))

En este caso se pueden visualizar las estimaciones de a y c como las ordenadas en el origen y la pendiente de una recta próxima a los puntos. El problema que se plantea entonces sería el de encontrar unos estimadores $\widehat{\beta}_1$ y $\widehat{\beta}_2$ tales que la recta que pase por los puntos (X_t, \widehat{Y}_t) se ajuste lo mejor posible a los puntos (X_t, Y_t) . Esto supone aplicar un procedimiento de cálculo que permita, a partir de los datos disponibles, obtener los coeficientes de la ecuación de la línea recta que represente óptimamente la distribución conjunta de las variables modeladas. Pudiéndose formular las siguientes interrogantes: ¿Cuál es la línea recta que representa óptimamente a una línea de puntos?, de otra forma, ¿Cuál es la que ofrece mejor bondad de ajuste? Uno de estos métodos de ajuste es el de los mínimos cuadrados.

3.2.4 Ajuste por Mínimos Cuadrados

La finalidad de este método, como se ha dicho con anterioridad es encontrar la recta que más se acerque al conjunto de datos dados, a la cual se le conoce como “recta de regresión”. Para proceder con este cálculo lo primero que se debe tener son las medidas, o datos de muestra. O sea un grupo de N pares de medidas $(x_1, y_1); (x_2, y_2); (x_N, y_N)$. Para encontrar los valores que mejores se ajustan a la función lineal se sustituyen N pares de datos en la función obtenida. Desde la implementación se obtendría así:

```
reg = new regresión(x, y);
getReg().lineal();
double a = getReg().getA();
double c = getReg().getC();

for (int i = 0; i < 50; i++) {
    List<Double> val = datos.get(i);
    Double fx = a*val.get(0) + c;
    linea.add(val.get(0), fx);
}
}
```

Figura 14 Obtener valores que mejor ajuste tienen aplicando el ajuste.

La pregunta entonces que se debe realizar es ¿Qué valores de a y c hacen mínimo x^2 ?
Definiendo: ¿Cuál es la recta que mejor se ajusta a las N medidas?

$$x^2(c, a) = \sum_{i=1}^N (y_i - ax_i - c)^2$$

Ecuación 19. Aplicación de los mínimos cuadrados a la función de evaluación.

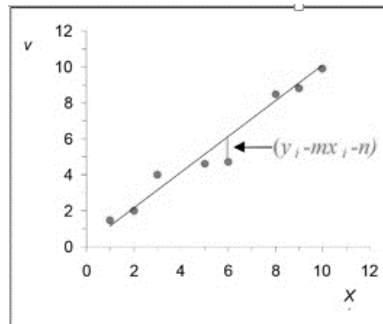


Figura 15 Recta que mejor se ajusta a las N medidas.

El siguiente paso supone saber cuan bueno es el comportamiento lineal de los N pares de datos. Para esto se debe obtener el valor o coeficiente de correlación lineal, denominado r , el cual es el encargado de medir el grado de correlación entre x y y . El mismo debe cumplir la siguiente condición (28):

- $|r| < 1$

Dependiendo del resultado obtenido se puede comprobar que:

- $|r| = 1$ existe correlación total.
- $|r| < 0$ No hay correlación.

Vale aclarar, según la bibliografía consultada que se suele considerar buen ajuste a partir de $|r| \geq 0.7$ aproximadamente (28). En la implementación se obtiene el valor de correlación a través del método de mismo nombre. A continuación un fragmento del código:

```
public double correlacion() {  
    double r = 0;  
    r = (n*sxy - sx*sy) / (Math.sqrt(n*sx2 - sx*sx) * Math.sqrt(n*sy2 - sy*sy));  
    return Math.abs(r);  
}
```

Figura 16 Método para obtener el valor de correlación.

3.3 Implementación

XP propone comenzar la implementación partiendo de una arquitectura lo más flexible posible para evitar grandes cambios en próximas iteraciones y en los cambios que frecuentemente el cliente propone. En el marco de la investigación se toma como punto de partida el análisis anteriormente realizado y los diagramas y especificaciones generados, dando paso a realizar la primera iteración.

3.3.1 Primera Iteración

El principal objetivo de la primera iteración es desarrollar las HU 1, 3 y 4. Esta iteración finaliza con la obtención de gráficos sencillos de barra y de líneas con datos experimentales comparando los diferentes modelos de propagación implementados en la herramienta Andrómeda. Para lograr este propósito se definieron las siguientes tareas ingenieriles:

HU N.1: “Permitir seleccionar el tipo de gráfico para el análisis e importar datos desde el servidor de BD, usando SQLite.”

1. Tarea N° 1: Estudiar cómo utilizar el *Lookup* en la guía definitiva de la Plataforma de Netbeans 7.
2. Tarea N° 2: Estudio de la arquitectura de la herramienta Andrómeda.
3. Tarea N° 3: Diseñar interfaz gráfica del módulo.
4. Tarea N° 4: Implementar clase *SQLManager_Conexion*.

HU N.3: “Obtener gráfico de barras partir de la selección del usuario.”

5. Tarea N° 5: Estudiar cómo obtener gráficos de barras con la librería JFreeChart.
6. Tarea N° 6: Diseñar e implementar la vista *Vista_graficar*.
7. Tarea N° 7: Implementar la clase *g_barras*.
8. Tarea N° 8: Implementar la funcionalidad *Graficar*.

HU N.4: “Obtener gráfico lineal a partir de la selección del usuario.”

9. Tarea N° 9: Estudiar cómo obtener gráficos lineales con la librería JfreeChart.
10. Tarea N° 10: Implementar la clase *gráficoLineal*.
11. Tarea N° 11: Implementar la funcionalidad *Graficar*.

Tabla 6. Tarea 1: Estudiar cómo utilizar el 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 cómo utilizar el <i>Lookup</i> en la guía definitiva de la Plataforma de Netbeans 7. | |
| Tipo de tarea: Preparación | Estimación: 6 días |
| Fecha de inicio: 9 de marzo de 2015 | Fecha fin: 14 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar todo lo que se refiere a <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. | |

La descripción del resto de las tareas de la Iteración 1 se encuentra detallada en el [Anexo 6](#).

3.3.2 Segunda Iteración

En la segunda iteración se desarrollan las HU 2,5 y 6, con el objetivo de obtener nuevas funcionalidades para una segunda versión del producto. Las tareas que darán cumplimiento a esta iteración se definen a continuación:

HU N.2: “Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado.”

12. Tarea N° 12: Estudiar el método de la regresión lineal simple por mínimos cuadrados.
13. Tarea N° 13: Implementar las clase *regresión*.
14. Tarea N° 14: Diseñar e implementar la vista *Vista_comparar_modelos*.

HU N.5: “Obtener gráfico de dispersión a partir de la selección del usuario.”

15. Tarea N° 15: Estudiar cómo obtener gráficos de dispersión con la librería JfreeChart.
16. Tarea N° 16: Implementar la clase *gráficoDispersión*.
17. Tarea N° 17: Implementar la funcionalidad *Graficar*.
18. Tarea N° 18: Implementar la funcionalidad *Graficar y calcular*.

HU N.6: “Exportar gráficos generado en formato .PNG.”

19. Tarea N° 19: Estudiar como exportar gráficos en formato .PNG con la librería JfreeChart.
20. Tarea N° 20: Implementar la funcionalidad que permita guardar gráficos en el formato .PNG.

La descripción de cada una de estas tareas se encuentra en el [Anexo 7](#).

3.3.3 Tercera Iteración

En la tercera y última iteración se desarrollan las HU 7, 8 y 9, con el objetivo de complementar las funcionalidades antes desarrolladas y obtener nuevas funcionalidades que darán fin a la implementación del módulo.

Las tareas definidas por HU para esta iteración son las siguientes:

HU N.7: “Exportar gráficos generado en formato .SVG.”

21. Tarea N° 21: Estudiar como exportar gráficos en formato .SVG con la librería JfreeChart.
22. Tarea N° 22: Implementar la funcionalidad que permita guardar gráficos en el formato .SVG.

HU N.8: “Exportar gráfico generado en formato .PDF.”

23. Tarea No 23: Estudiar en la librería JFreeChart como exportar en formato .PDF.
24. Tarea N° 24: Implementar la funcionalidad que permita guardar gráficos en el formato .PDF.

HU N.9: “Personalizar los gráficos generados.”

25. Tarea N° 25: Estudiar en la librería JfreeChart las opciones de personalización disponibles para los gráficos generados.
26. Tarea N° 26: Implementar las opciones de personalización de los gráficos.

La descripción de cada una de estas tareas se encuentra en el [Anexo 8](#).

3.4 Pruebas

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación (61). Las pruebas de Caja Blanca y Caja Negra destacan dentro de las técnicas más comunes para la realización de pruebas, efectuando diferentes tareas en la manera en que se aplican, ya sean unitarias, integración, aceptación o sistema, en relación con la metodología de desarrollo utilizada.

La metodología XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (62) .

Comprobaciones manuales o automatizadas desarrolladas por los programadores de software conforman las llamadas pruebas de unidad o pruebas unitarias. Las cuales se realizan para verificar que el código correspondiente a un módulo concreto se comporte de la manera que se espera. Dichas pruebas aportan un grupo importante de beneficios, dentro de los que destacan (62):

- ⚙ Ofrecer al programador una inmediata retroalimentación de cómo está realizando su trabajo.
- ⚙ El programador puede realizar cambios de forma segura respaldadas por casos de pruebas.
- ⚙ Permiten saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.

Al módulo resultado de esta investigación se le realizaron las pruebas unitarias correspondientes a las principales funcionalidades y clases; estas pruebas fueron realizadas por el equipo de desarrollo a medida que se fue implementando el módulo por lo que no son muchos los registros que se conservan de las mismas debido a que las deficiencias encontradas fueron solucionadas durante el proceso de ejecución de estas.

[\(Anexo 13\)](#)

Referente a las pruebas de aceptación las conclusiones a las que han llegado diferentes autores plantean que son consideradas como las más adecuadas, pues confieren el

grado de satisfacción que tenga el cliente con el producto (62). El cliente y un representante del equipo de desarrollo se encargan de realizarlas. Con estas pruebas se persigue el objetivo de obtener una visión externa del producto final y se centran en analizar la funcionalidad del sistema. Se logra, desde la perspectiva del usuario verificar el cumplimiento de la lista de reservas del producto.

3.4.1 Pruebas de aceptación para la Iteración 1

Las pruebas de caja negra o de aceptación se centran principalmente en lo que se quiere de un “módulo” o sección específica de un software, es decir, es una manera de encontrar casos específicos en ese módulo que transgreden a su especificación (63). Estas pruebas se limitan a que el probador o desarrollador que se encuentre realizando los pruebas, experimente con datos de entrada y estudie como salen.

Debido a las características de Andromeda_Grapher, puede confirmarse que las pruebas de aceptación tienen un mayor peso que las pruebas unitarias, ya que el módulo desarrollado se enmarca en el trabajo con interfaces, y el éxito del resultado del mismo se encuentra en función de la aceptación del cliente realizando este tipo de prueba. Estas pruebas marcan el final de una iteración y el comienzo de la siguiente.

Para la primera iteración de pruebas se realizaron 3 pruebas de aceptación, identificándose 5 no conformidades. La complejidad mínima de las no conformidades contribuyó a que todas fueran resueltas, por lo que no quedaron casos de prueba pendientes. La descripción de los casos de prueba se encuentra en el [Anexo 10](#).

3.4.2 Pruebas de aceptación para la Iteración 2

Se realizaron 3 casos de pruebas para esta iteración, identificándose 4 no conformidades, las cuales proceden. Por lo que no se cuenta con casos de prueba pendientes. En el [Anexo 11](#) se describen los casos de pruebas en cuestión.

3.4.3 Pruebas de aceptación para la Iteración 3

En esta ocasión se realizaron 4 casos de pruebas de aceptación, donde se identificó 1 no conformidad, la cual procede. No quedaron casos de prueba pendientes. Estos casos de pruebas son descritos en el [Anexo 12](#).

3.4.5 Resultados generales de las pruebas

La fase final de pruebas de la aplicación se desglosó en 3 iteraciones. Las pruebas fueron aplicadas a las funcionalidades más importantes. Las no conformidades detectadas se clasificaron en dos grupos; las significativas, que agrupan todas aquellas no conformidades orientadas a la funcionalidad y validación y las no significativas que abarcan la interfaz de usuario (errores ortográficos).

En la primera iteración se encontraron 1 no conformidad significativa y 4 no significativas, en la segunda iteración se detectaron 2 no conformidades relacionadas con ortografía y 2 correspondiente a las funcionalidades. En la última iteración fue encontrada 1 no conformidad significativa.

Tabla 9. No conformidades por iteración de prueba.

| | Iteración 1 | Iteración 2 | Iteración 3 | TOTAL |
|----------------------------|-------------|-------------|-------------|-------|
| Interfaz de usuario | 4 | 2 | 0 | 6 |
| Funcionalidad | 1 | 1 | 0 | 2 |
| Validación | 0 | 1 | 1 | 2 |
| TOTAL | 1/4 | 2/2 | 1/0 | 10 |

El siguiente gráfico muestra un resumen de los resultados arrojados por las pruebas realizadas al módulo Andromeda_Grapher.

Resultados de las pruebas.

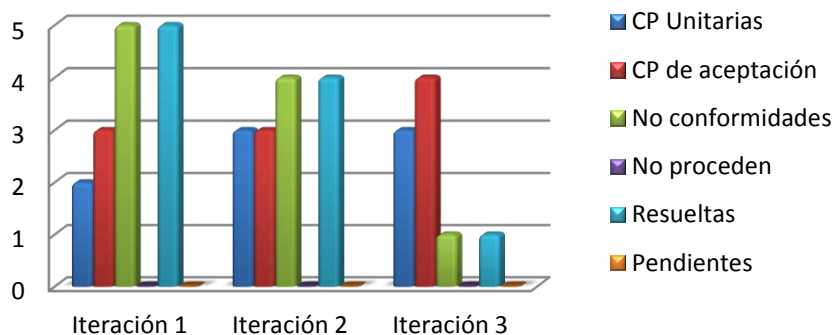
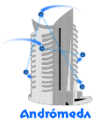


Figura 17 Resultados de las pruebas



3.5 Conclusiones parciales

En el presente capítulo fue presentado el módulo a partir de sus componentes de implementación, tomando como guía la característica de la metodología XP de no establecer claramente los artefactos a emplear en el desarrollo de una aplicación. Se decidió en primera instancia realizar una descripción de la arquitectura sobre la cual se sustenta el módulo que se desarrolló, facilitando detalles de la estructura de la plataforma Andrómeda y que relación existía entre sus componentes. Se presentaron los diagramas de paquetes y de componentes del módulo desarrollado. Se cumplió con el plan de iteraciones establecido para realizar las entregas de cada una de las versiones del producto. Se estudió el método de la regresión lineal por mínimos cuadrados y se describió su utilización en la implementación de la solución propuesta. Una vez concluida la fase de implementación se procedió a la realización de pruebas para comprobar la calidad del módulo. Se realizaron pruebas unitarias y de aceptación resultando satisfactorias.



Conclusiones

Finalizada la investigación y con el resultado del producto de software, se le da solución al problema planteado y cumplimiento a los objetivos propuesto para el presente trabajo de diploma. Al término del mismo, se concluye...

- Con el estudio realizado y el análisis del estado del arte referente a los diferentes modelos de propagación se estableció que existen diversas clasificaciones de dichos modelos, pero de forma general, se pueden clasificar en empíricos o estadísticos y teóricos o deterministas.
- A partir del estudio de los métodos matemáticos de análisis estadístico, se estableció que estos se presentan como una eficaz herramienta para el análisis de variables y brindan la facilidad de obtener proyecciones futuras sobre el comportamiento de estas.
- Se determinó usar como lenguaje de programación Java, utilizando como IDE NetBeans v7.4 y como sistema gestor de base de datos SQLite. La librería usada para el desarrollo fue JfreeChart, ya que esta aporta la experiencia de obtener potentes gráficos de diversas categorías. Guiándose todo el proceso por la metodología XP y como herramienta CASE se utilizó Visual Paradigm para UML v8.0.
- Con el módulo Andromeda_Grapher implementado, en su primera versión, se podrán obtener gráficos comparativos referentes a los modelos de propagación implementados en la herramienta Andrómeda, permitiendo comprobar, atendiendo a la pérdida de la señal entre Tx y Rx, cuál se ajusta óptimamente en el diseño de una WSA.
- Durante el proceso de pruebas se demostró la correcta implementación de las funcionalidades propuestas. Para ello se emplearon las pruebas unitarias y de aceptación.



Recomendaciones

Una vez finalizado este trabajo investigativo, donde se hizo referencia a los métodos de análisis de datos y cómo emplearlo con los modelos de propagación generados por la herramienta Andrómeda, se observa que la aplicación de métodos matemáticos de análisis estadístico en modelos de propagación en entornos cerrados es compleja. En este trabajo se ha tenido en cuenta solamente la pérdida de señal entre el transmisor y el receptor. Sin embargo, existe otro grupo importante de elementos que no fueron considerados y que se pueden añadir para perfeccionar el módulo resultante.

De ahí que se propone:

1. Profundizar en el estudio de las propiedades de propagación de los modelos para entornos interiores considerando siempre factores como el efecto Multi-Path y los fenómenos físicos de reflexión y difracción de la señal.
2. Ampliar la base de datos de medidas utilizando diferentes edificaciones que permitan un mejor y más correcto ajuste de los modelos para posteriores estudios con diversificada cantidad de edificios.
3. Implementar nuevas funciones matemáticas que avalen y justifiquen el correcto uso de los modelos de propagación en determinadas circunstancias.

Referencias Bibliográficas

1. **Felpeto, Ernesto Melian.** *Análisis forense en la red inalámbrica del Docente 1.* 2009.
2. **BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales.** *BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales.* [En línea] [Citado el: 18 de enero de 2015.] <http://www.eumed.net/libros-gratis/2007c/312/CONCLUSION%20LA%20SOCIEDAD%20MOVIL%20EN%20RED.htm>.
3. **Abruzzese, D.** *Long life monitoring of historical monuments via Wireless Sensors Network.* in *Wireless Communication Systems.* [ISWCS 2009] s.l. : 6th International Symposium on. 2009., 2009.
4. **Mora, Ismael Armando Nodarse y Díaz Rodríguez, Manuel.** *Utilización de WSAWs en Sistemas de Control de Edificios.* 2010.
5. **Fernández, Alexander García.** *Web oficial de la CUJAE.* [En línea] 2007. [Citado el: 18 de enero de 2015.] <http://rii.cujae.edu.cu/index.php/revistaind/article/view/84>.
6. **A Survey of RF-Propagation Simulation Tools for Wireless Sensor Networks.** Korkalainen, Marko y Sallinen, Mikko . s.l. : 2010 Fourth International Conference on Sensor Technologies and Applications, 2010.
7. **Serrano, A.G.** *Redes Wi-Fi.* España : Editorial Anaya, 2008. 1ra edición.
8. **Roldan, D.** *"Comunicaciones inalámbricas".* España : Editorial Microinformática, 2004. 1era. Edición.
9. **OpenCourseWare de la Universidad de Sevilla.** [En línea] 9 de 2008. [Citado el: 7 de 2 de 2015.] http://ocwus.us.es/metodos-de-investigacion-y-diagnostico-en-educacion/analisis-de-datos-en-la-investigacion-educativa/Bloque_I/page_03.htm/.
10. **Olivares, José Luis Camargo.** *Modelo de cobertura para redes inalámbricas de interiores.* Sevilla : s.n., 2009.
11. **Fernández, Néstor García.** *Modelo de cobertura en redes inalámbricas.* Oviedo : s.n., 2006.
12. **Cramér, Harald.** *Mathematical Methods of S.* 1951.
13. **Cano, José Antonio Martínez.** *Estudio Teórico Experimental de un Sistema Distribuido MIMO en la Banda de Frecuencias de WIFI.* Cartajena : Universidad de Cartajena, 2006.
14. **Mail Mail .** *Pronóstico. Definición, importancia y tipos de pronósticos.* [En línea] 2012. [Citado el: 5 de 3 de 2015.] http://www.mailxmail.com/pronostico-definicion-importancia-tipos-pronostico_h.
15. **G., Wofle, y otros.** *Extensions to the field strength prediction technique based on dominant paths between transmitter and receiver in indoor wireless communications.* Conference Bonn : s.n., Septiembre 1997.
16. **García, Nelson Pérez.** *Telematique. A New Propagation Model for WLAN Operating at 2.4 GHz, in Indoor Environments.* [En línea] 21 de 11 de 2010. [Citado el: 21 de 1 de 2015.] <http://publicaciones.urbe.edu/index.php/telematique/>.
17. **Lidón, Manuel Ballester.** *Aplicación del modelo de propagación Multi-Wall para la estimación de cobertura de femtoceldas LTE en interiores.* Valencia : s.n., 2011.
18. **Landstorfer, F. M.** *Modelo de Cobertura para Redes Inalámbricas de Interiores.* 2012.
19. **F.M.Landstorfer.** *Modelos de propagación de rayos en interior.* Malaga : s.n., 2013.
20. **Arfken, G.** *Ecuaciones De Maxwell.* España : s.n., 2013.
21. **Campodocs.com.** *Método de tiempo-dominio de diferencias finitas, Historia, Modelos y métodos FDTD, Popularidad.* [En línea] 8 de 3 de 2013. [Citado el: 22 de 1 de 2015.] http://campodocs.com/articulos-utiles/article_118809.html.
22. **Wofle, G.** *AWE-Communications.* [En línea] 2012. [Citado el: 26 de febrero de 2014.] <http://www.awe-communications.com/>.
23. **Chain, Sapag .** *Preparación y evaluación de proyectos.* s.l. : Mc-Graw Hill, 2000.
24. **Nojek, S., y otros.** *Pronóstico de Ventas: Comparación de Predicción basada en Redes Neuronales versus Método Estadístico.* Buenos Aires : Instituto Tecnológico de Buenos Aires, 2010.
25. **COLE, J. H.** *Enciclopedia Multimedia Virtual Interactiva."Nociones de Regresión Lineal".* 2005.
26. **Aguilar, Joenith , Bacallao-Guerra, Jorge y Morales, Estela .** *Estimación del riesgo de descendencia con aberraciones cromosómicas.* La Habana : s.n., 2011.
27. **WALPOLE, Myers.** *Probabilidad y Estadística para Ingenieros.* Mexico : rentice Hall, 1998.

28. **Vanegas, Alberto Patiño.** *ANÁLISIS GRÁFICO DE DATOS EXPERIMENTALES*. [En línea] 2014. [Citado el: 18 de 4 de 2015.] http://www.fisica.ru/dfmg/teacher/archivos/LabFis1_Lectura_de_apoyo2_graficas.pdf.
29. **Berenson, M. y Levine, D.** *Estadística Básica para Administración*. s.l. : Editorial Prentice Hall., 1996. 6ta edición.
30. **Lind, D., Marchal, W. & Mason, R.** *Estadística para Administración y Economía*. México : Alfa Omega Grupo editor., 2004.
31. **UNAD. Universidad Nacional Abierta y a Distancia.** *Lección 4: SUAVIZACIÓN EXPONENCIAL*. [En línea] 2014. [Citado el: 15 de 3 de 2015.] http://datateca.unad.edu.co/contenidos/104561/Metodos_Probabilisticos_2013/MODULO_2013_ACTAUALIZADO/leccin_4_suavizacin_exponencial.html.
32. **QTX de México.** Qualitrain. [En línea] 2011. [Citado el: 25 de 1 de 2015.] <http://www.qualitrain.com.mx/objeIndirecto/javavsvbasic.htm>.
33. **Arnold, K., J. Gosling, and D. Holmes.** *Java™ Programming Language*. 2005. 4 ed..
34. **Oracle Corporation.** NetBeans. [En línea] <http://netbeans.org>.
35. **Unijimpe. unijimpe.** [En línea] 2009. [Citado el: 12 de 5 de 2015.] <http://blog.unijimpe.net/jpgraph-graficos->
36. **Internet y Tecnología.** [En línea] 2011. [Citado el: 14 de 5 de 2015.] <http://pixelco.us/blog/10-poderosos-recursos-para-graficar-datos/>.
37. **JFreeChart.** [En línea] 31 de 7 de 2014. [Citado el: 6 de 2 de 2015.] <http://www.jfree.org/jfreechart/>.
38. **SALAZAR TERRERO, Lirisandra y GONZÁLEZ RODRÍGUEZ, Raúl.** *Solución informática para el portal web “PREPARACIÓN PARA LA DEFENSA FACULTAD # 4.”*. La Habana : Universidad de las Ciencias Informáticas, 2012.
39. **SQLite.org.** About SQLite. . [En línea] 2011. [Citado el: 27 de 1 de 2015.] from: <http://www.sqlite.org/about.html>.
40. **Española R.A.** Metodología. [En línea] [Citado el: 22 de 1 de 2015.] http://buscon.rae.es/draeI/SrvltGUIBusUsual?TIPO_HTML=2&TIPO_BUS=3&LEMA=metodolog%C3%ADa.
41. **Beck, Kent.** Los Procesos Ágiles: SCRUM y eXtreme Programming. [En línea] 2012. [Citado el: 25 de 1 de 2015.] <http://procesosagiles.com/index.html>.
42. **GALVEZ, JORGE.** Fundamentos de la Metodología RUP. [En línea] 16 de 9 de 2007. [Citado el: 5 de 2 de 2015.] <http://www.scribd.com/doc/297224/RUP>.
43. **Martinez, Ruben. Prezi .** "METODOLOGIA RUP". [En línea] 10 de 4 de 2014. [Citado el: 5 de 2 de 2015.] <https://prezi.com/gzlcbljmwhcq/metodologia-rup/>.
44. **Britto Montoya, Jaime Andrés y Hurtado Gil, Sandra Victoria.** Repositorio Institucional - Universidad Autónoma de Manizales. *Adaptación de un proceso de desarrollo de software basado en buenas prácticas*. [En línea] 22 de 1 de 2015. [Citado el: 2015 de 2 de 2015.] <http://repositorio.autonoma.edu.co/jspui/bitstream/11182/838/1/ADAPTACION%20DE%20UN%20PROCESO%20DE%20DESARROLLO%20DE%20SOFTWARE%20BASADO%20EN%20BUENAS%20PR%20CTICAS%20-%20JAJIME%20BRITTO.pdf>.
45. **Penadés, M.C. y Torres, P.O.L.** *Métodologías ágiles para el desarrollo de software. EXtreme Programming (XP)*. 2006. 05.
46. **Caracheo, Eduardo Aguirre.** Repositorio Institucional UAQ. [En línea] 10 de 2014. [Citado el: 5 de 2 de 2015.] <http://ri.uaq.mx/bitstream/123456789/2394/1/RI001928.pdf>.
47. **Proyectalis.** [En línea] 2015. [Citado el: 24 de 1 de 2015.] <http://www.proyectalis.com/servicios/formacion/scrum/>.
48. **Serrano, Jorge.** Explicando Scrum a mi abuela. [En línea] 2014. [Citado el: 24 de 1 de 2015.] <http://geeks.ms/blogs/jorge/archive/2007/05/09/explicando-scrum-a-mi-abuela.aspx>.
49. **www.visual-paradigm.com.** [En línea] [Citado el: 2 de 5 de 2015.] <http://www.visual-paradigm.com/aboutus>.
50. **Echeverry, Luis Miguel y Delgado Carmona, Luz Elena.** *Caso práctico de la metodología agil XP al desarrollo de software*. Pereira : Universidad Tecnológica de Pereira. , 2007.
51. **Docsetools.com.** [En línea] 2014. [Citado el: 18 de 02 de 2015.] http://docsetools.com/articulos-educativos/article_11461.html.



52. **CI-4713 Ingeniería de Software II** . [En línea] 2009. [Citado el: 18 de 2 de 2015.] <http://ldc.usb.ve/~abianc/materias/ci4713/metodologiasagiles.pdf>.
53. **Joskowicz, José**. *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.
54. **Tedeschi, Nicolás**. ¿Qué es un Patrón de Diseño? [En línea] 2010. [Citado el: 3 de 3 de 2015.] <https://msdn.microsoft.com/es-es/es-s/library/bb972240.aspx>.
55. **Hernández, Pedro Veloso**. *Uso de patrones de arquitectura*. 2011.
56. **Craig, Larman**. *UML y patrones*. s.l. : Prentice Hall, 2004.
57. **Rob, Peter and Coronel, Carlos**. *Sistemas de bases de datos: diseño, implementación y administración*. 2006.
58. **Universidad de Sevilla**. Bases de Datos. Modelos de datos. [En línea] 2005. [Citado el: 3 de 3 de 2015.] <http://www.lsi.us.es/docencia/get.php?id=3186>.
59. **Prezi**. *Diagrama de Clases*. [En línea] 26 de 8 de 2013. [Citado el: 3 de 3 de 2015.] <https://prezi.com/ajiesi7miczi/diagrama-de-clases/>.
60. **Molina Toledo, Pablo y Rodríguez Estévez, Henry**. *Arquitectura de la herramienta Andrómeda*. La Habana : s.n., 2012.
61. **Pressman, Roger S**. *Ingeniería del Software. Un enfoque práctico*. Madrid : Concepción Fernández , 2002. 0-07-709677-0.
62. **Javier J. Gutierrez, M. J. Escalona, M. Mejías, J. Torres**. *Sistemas de Programación Extrema*. [En línea] 2013. [Citado el: 18 de 5 de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf.
63. **Globe Testing**. [En línea] 14 de 8 de 2012. [Citado el: 21 de 4 de 2015.] <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>.
64. **Landstorfer, F. M**. *Modelo de Cobertura para Redes Inalámbricas de Interiores*. 2012.
65. **Torrecillas, Valeriano Campillo**. *Programación de una herramienta de trazado de rayos 3D basada en GO y UTD*. Cartagena : s.n., 2011.
66. **Insituto Tecnológico Superior de Teziutlán**. PLATAFORMAS DE CLIENTE RICO EN EL DESARROLLO DE APLICACIONES DE ESCRITORIO CON JAVA . [En línea] 12 de 7 de 2012. [Citado el: 5 de 2 de 2015.] http://www.itsteziutlan.edu.mx/site2010/index.php?option=com_content&view=article&id=773:plataformas-de-cliente-rico-en-el-desarrollo-de-aplicaciones-de-escritorio-con-java&catid=27:artlos&Itemid=288.
67. **Bock, H**. *The Definitive Guide to NetBeans Platform 7*. s.l. : Apress., 2011.
68. **Modelos de Pronósticos**. [En línea] 2011. <http://www.paginasprodigy.com/sylsr/ingenierias/pronosticos/pronostic...>
69. **Barrientos, Andrés Felipe, Olaya , Javier y González, Víctor Manuel** . *Un modelo spline para el pronóstico de la demanda de energía eléctrica*. Colombia : Revista Colombiana de Estadística, 2007. 2.
70. **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.
71. **DigitalLika**. [En línea] 2010. [Citado el: 16 de 3 de 2015.] <http://www.digitalika.com/2010/12/open-source-definicion-de-hoy/>.
72. **Molina, Gabriel y F. Rodrigo, María** . *Estadística descriptiva en Psicología*. Valencia : s.n., 2010.
73. **Rodríguez, Edgar Romero**. *Unisangil-Métodos Numéricos*. 2007.
74. **Mercado, Armando, Berríos, Rafaelgil y Chan Ye, Paul**. *Redes Inalámbricas Ad hoc*. [En línea] 2010. [Citado el: 28 de 5 de 2015.] <http://facultad.bayamon.inter.edu/cgonzalezr/elen4618/adhoc.pdf>.
75. **Histchfeld K., Nancy y Salinas Caro, Patricio** . *Tutorial de UML*. [En línea] 2012. [Citado el: 23 de 5 de 2015.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.

Anexos

Anexo 1. Fenómenos de difracción, dispersión y reflexión de la señal.

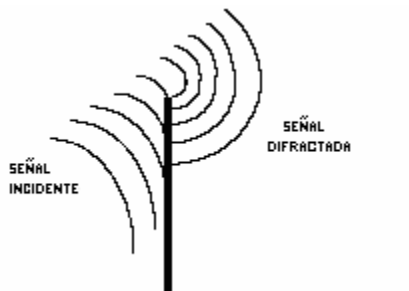


Figura 18 Fenómeno de la Difracción de la señal.

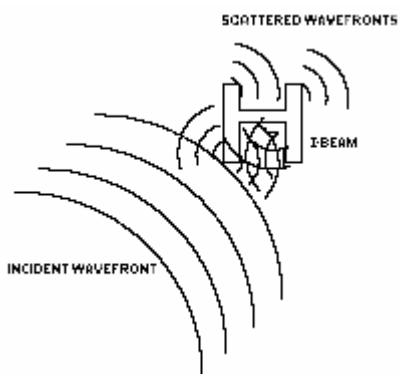


Figura 19 Fenómeno de la Dispersión de la señal.

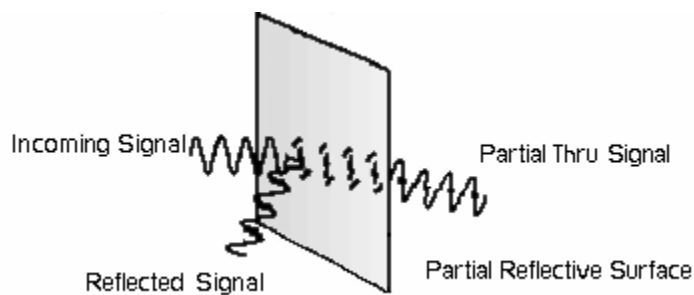


Figura 20 Fenómeno de la Reflexión de la señal.

Anexo 2. Fases de desarrollo de la Metodología SCRUM

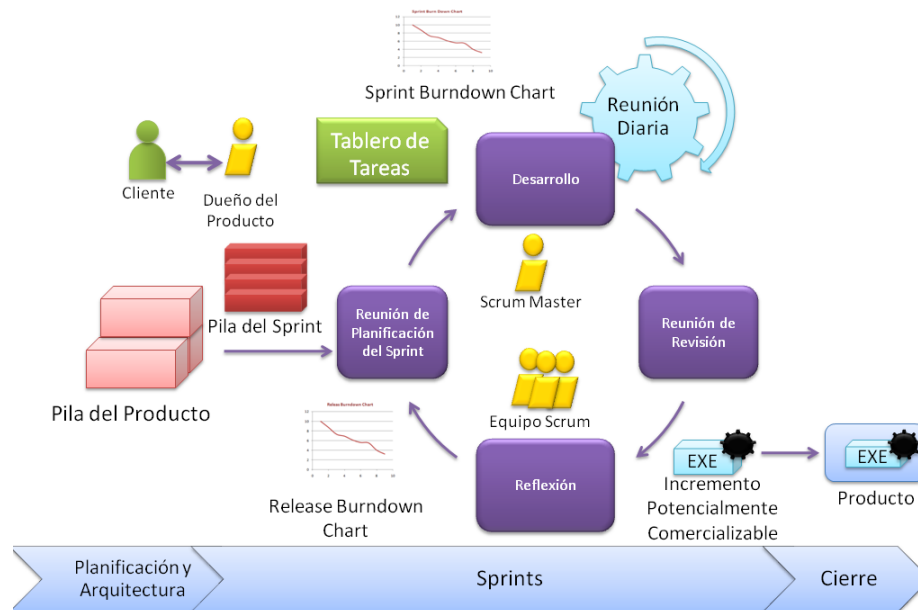


Figura 21 Flujos de trabajo para la metodología SCRUM.

Anexo 3. Historias de Usuario.

Tabla 7. HU 2: Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado.

| Historia de Usuario | |
|--|---|
| No.: 2 | Nombre: Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado. |
| Usuario: Todos | |
| Prioridad en el Negocio: Alta | Nivel de Complejidad: Alta |
| Estimación: 3 semanas | Iteración Asignada: 2 |
| <p>Descripción: El análisis de la regresión lineal se aplicará después de tener definido la variable dependiente y la(s) variable(s) independiente sobre la cual se realizará este análisis. Se implementa la clase <i>regresión</i> la cual utiliza los datos de muestra obtenidos para aplicar los pasos que conforman la regresión y obtener el grupo de puntos que se mostraran en el gráfico de dispersión, así como la recta de mejor ajuste y el valor de correlación obtenido para el análisis realizado.</p> | |
| <p>Paquete: cu.andromeda.Regresión_Lineal</p> | |
| <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad “Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado.” Solo se puede realizar el análisis con un tipo de dato en específico y en un rango determinado. Como se especificó inicialmente que se utilizaría regresión lineal simple, solo se usa una sola función para establecer la aproximación.</p> | |

Tabla 8. HU 3: Obtener gráfico de barras partir de la selección del usuario.

| Historia de Usuario | |
|---|---|
| No.: 3 | Nombre: Obtener gráfico de barras a partir de la selección del usuario. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 1 semana | Iteración Asignada: 1 |
| <p>Descripción: El usuario podrá escoger la opción de obtener gráfico de barras.</p> <p>Gráficos de barras: Este gráfico es una representación gráfica bidimensional en que los objetos gráficos elementales son un conjunto de rectángulos dispuestos paralelamente de manera que la extensión de los mismos es proporcional a la magnitud que se quiere representar. La gráfica de barras permite comparar y tener una representación gráfica de la diferencia de frecuencia o de intensidad de la característica numérica de interés que se esté evaluando.</p> <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad “Obtener gráfico a partir de la selección del usuario.” En este caso la obtención del gráfico de barras.</p> | |

Tabla 9. HU 4: Obtener gráfico lineal a partir de la selección del usuario.

| Historia de Usuario | |
|---|--|
| No.: 4 | Nombre: Obtener gráfico lineal a partir de la selección del usuario. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 1 semana | Iteración Asignada: 1 |
| <p>Descripción: El usuario podrá obtener el gráfico lineal luego de haber seleccionado la opción correspondiente.</p> <p>Gráficos Lineales: La gráfica de líneas muestra una serie con un conjunto de puntos conectados mediante una línea. Los valores se representan por el alto de los puntos con relación al eje y. Las etiquetas de las categorías se representan en el eje X.</p> <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad “Obtener gráfico a partir de la selección del usuario.” En este caso la obtención del gráfico lineal.</p> | |

Tabla 10. HU 5: Obtener gráfico de dispersión a partir de la selección del usuario.

| Historia de Usuario | |
|---------------------|---|
| No.: 5 | Nombre: Obtener gráfico de dispersión a partir de la selección del usuario. |
| Usuario: Todos | |

| | |
|---|------------------------------------|
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 1 semana | Iteración Asignada: 3 |
| <p>Descripción: El usuario podrá elegir la opción de obtener gráfico de dispersión. Esta es la representación gráfica más útil para describir el comportamiento conjunto de dos variables. El gráfico tiene dos ejes de valores para mostrar un conjunto de datos numéricos en el eje horizontal (valor) y otro en el eje vertical (valor). El gráfico muestra puntos en la intersección de un valor numérico x e y, combinando estos valores en puntos de datos únicos. Estos puntos de datos pueden distribuirse de forma uniforme o desigual a lo largo del eje horizontal, en función de los datos.</p> <p>Paquete: cu.andromeda.Gráficos</p> | |
| <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Obtener gráfico a partir de la selección del usuario." En este caso la obtención del gráfico de dispersión.</p> | |

Tabla 11. HU 6: Exportar gráficos generado en formato .PNG.

| Historia de Usuario | |
|---|--|
| No.: 6 | Nombre: Exportar gráficos generado en formato .PNG. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 2 días | Iteración Asignada: 2 |
| <p>Descripción: La herramienta debe permitir exportar los gráficos generados en formato .PNG. Esto se logra usando una de las características de la propia librería que se emplea para el proceso de graficado. Tiene que haberse generado un gráfico para poderlo salvar.</p> | |
| <p>Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Exportar gráficos generados según el comportamiento de los modelos de propagación."</p> | |

Tabla 12. HU 7: Exportar gráficos generado en formato .SVG.

| Historia de Usuario | |
|---------------------------------------|--|
| No.: 7 | Nombre: Exportar gráficos generado en formato .SVG. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 2 días | Iteración Asignada: 2 |

Descripción: La herramienta debe permitir exportar los gráficos generados en formato .SVG. Esto se logra usando una de las características de la propia librería que se emplea para el proceso de graficado. Tiene que haberse generado un gráfico para poderlo salvar.

Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Exportar gráficos generados según el comportamiento de los modelos de propagación."

Tabla 13. HU 8: Exportar gráficos generados en formato .PDF.

| Historia de Usuario | |
|---|--|
| No.:8 | Nombre: Exportar gráficos generado en formato .pdf. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 1 semana | Iteración Asignada: 3 |
| Descripción: La herramienta debe permitir exportar los gráficos generados en formato .PDF. Para ello hará uso de las facilidades que brinda la propia librería JfreeChart para exportar los gráficos. Tiene que haberse generado un gráfico para poderlo salvar. | |
| Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Exportar gráficos generados según el comportamiento de los modelos de propagación." | |

Tabla 14. HU 9: Personalizar los gráficos generados.

| Historia de Usuario | |
|--|---|
| No.:9 | Nombre: Personalizar los gráficos generados. |
| Usuario: Todos | |
| Prioridad en el Negocio: Media | Nivel de Complejidad: Media |
| Estimación: 4 días | Iteración Asignada: 3 |
| Descripción: La herramienta deberá proporcionarle al usuario varias opciones para personalizar los parámetros de los gráficos creados, para eso al dar clic derecho sobre el gráfico se mostrará un menú donde el usuario podrá seleccionar esas opciones. Pudiéndose modificar el nombre del gráfico, el color de fondo de estos, los rangos de valores, los nombres de los ejes. Además de permitirle acercar o alejar la visualización para una mejor comprensión. | |
| Información adicional (Observaciones): Da cumplimiento a la funcionalidad "Personalizar los gráficos generados." | |

Anexo 4. Patrones de Diseño

```
public class Element extends Leaf {  
  
    private Transform3D _transform;  
    private String _type;  
    private String _name;  
    private String _storey;  
    private String _id;  
    private int _lineNumber;  
    private Vector3f _position;  
    private Vector3f _positionGeometry;  
    private Vector3f _direction;  
    private Color _color;  
}
```

Figura 22 Ejemplo de código de la clase Element.

```
public class Device implements Serializable{  
    private String _model;  
    private DeviceType _deviceType;  
    private String _manufacturer;  
    private Number _averageConsumption;  
    private Number _price;  
    private String _wirelessStandar;  
    private Number _transmisiorPower;  
    private Number _transmisiorFrequency;  
    private boolean _source;  
    private boolean _isActive;  
    private Date _createdAt;  
    private Date _updatedAt;  
    private Date _deletedAt;  
    private List<SensorType> _sensorsIncluding;  
    private List<SensorType> _sensorsInUse;  
    private Point3f _position;  
}
```

Figura 23 Fragmento de código de la clase Device.

```
@ServiceProvider(service = IModelConnectivity.class)  
public class MultiWallModel implements IModelConnectivity {  
  
    //Model name to implement  
    private static String _modelName = "Multi-Wall Model";  
    //Class to control the model  
    private ModelControl _modelControl = ModelControl.getInstance();  
    //Geometry Class  
    private Geometry _geometry = Geometry.getInstance();  
    private float _receivedSignal;  
    private Path _path = new Path();  
}
```

Figura 24 Fragmento de código de la clase MultiWallModel.

```
public static Andromeda getInstance() {  
    if (_instance == null) {  
        _instance = new Andromeda();  
    }  
    return _instance;  
}
```

Figura 25 Ejemplo de código de la clase Andromeda.

```
public static ModelControl getInstance() {  
    if (_instance == null) {  
        _instance = new ModelControl();  
    }  
    return _instance;  
}
```

Figura 26 Ejemplo de código de la clase ModelControl.

```
public static regresión getInstance() {  
    Object instance = null;  
    if(instance==null){  
        _instance=new regresión();  
    }  
    return _instance;  
}
```

Figura 27 Ejemplo de código de la clase regresión

Anexo 5. Diagrama de dispersión luego de aplicar regresión lineal simple.

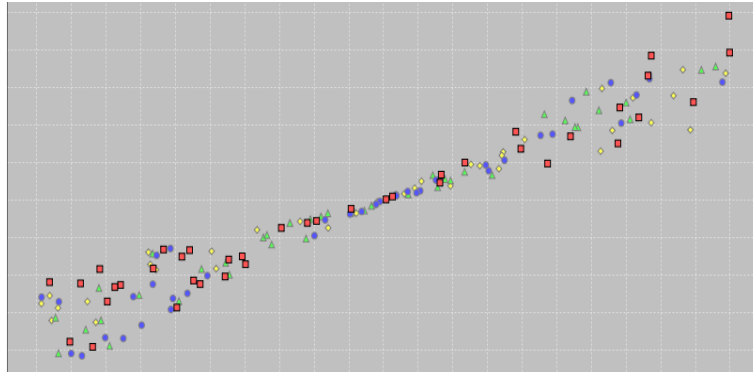


Figura 28 Dispersión con regresión lineal.

Anexo 6. Descripción de tareas Iteración 1.

Tabla 15. Tarea 2: Estudio de la arquitectura de la herramienta Andrómeda.

| Tarea | |
|---|--------------------------------|
| Número de tarea: 2 | Número de HU: 1 |
| Nombre de la tarea: Estudio de la arquitectura de la herramienta Andrómeda. | |
| Tipo de tarea: Preparación | Estimación: 6 días |
| Fecha de inicio: 15 de marzo de 2015 | Fecha fin: 21 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: | |
| 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 16. Tarea 3: Diseñar interfaz gráfica del módulo.

| Tarea | |
|--|--------------------------------|
| Número de tarea: 3 | Número de HU: 1 |
| Nombre de la tarea: 1. Diseñar interfaz gráfica del módulo. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha de inicio: 23 de marzo de 2015 | Fecha fin: 25 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: | |
| Diseñar la interfaz gráfica del módulo de selección de las opciones. | |
| Paquete: cu.andromeda.Vistas | |

Tabla 17. Tarea 4: Implementar clase SQLManager_Conexion.

| Tarea | |
|--|---------------------------------------|
| Número de tarea: 4 | Número de HU: 1 |
| Nombre de la tarea: 1. Implementar clase SQLManager_Conexion | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha de inicio: 23 de marzo de 2015 | Fecha fin: 25 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar la clase SQLManager_Conexion para garantizar el acceso a los datos de muestra que se utilizaran para el graficado y que se encuentran en la tabla correspondiente en la base de datos utilizada por la herramienta Andrómeda. | |
| Paquete: cu.andromeda.BD | |

Tabla 18. Tarea 5: Estudiar cómo obtener gráficos de barras con la librería JFreeChart.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 5 | Número de HU: 3 |
| Nombre de la tarea: Estudiar cómo obtener gráficos de barras con la librería JFreeChart. | |
| Tipo de tarea: Preparación | Estimación: 2 días |
| Fecha de inicio: 25 de marzo de 2015 | Fecha fin: 27 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar la librería JfreeChart y como obtener gráficos de barras con ella, con el objetivo de aplicarla en la implementación. | |

Tabla 19. Tarea 6: Diseñar e implementar la vista *Vista_graficar*.

| Tarea | |
|--|---------------------------------------|
| Número de tarea: 6 | Número de HU: 3 |
| Nombre de la tarea: Diseñar e implementar la vista <i>Vista_graficar</i> . | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha de inicio: 28 de marzo de 2015 | Fecha fin: 29 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Se implementa la vista <i>Vista_graficar</i> , la cual contendrá los tres tipos de gráficos que la aplicación generará (barras, lineal, dispersión), y el usuario podrá elegir cuál o cuáles, desea visualizar. | |
| Paquete: cu.andromeda.Vistas | |

Tabla 20. Tarea 7: Implementar la clase *g_barras*.

| Tarea | |
|---|--------------------------------------|
| Número de tarea: 7 | Número de HU: 3 |
| Nombre de la tarea: Implementar la clase <i>g_barras</i> . | |
| Tipo de tarea: Preparación | Estimación: 3 días |
| Fecha de inicio: 29 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Se crea la clase <i>g_barras</i> . La misma es la encargada de generar el gráfico de barras con las propiedades inicialmente establecidas, que posteriormente el usuario podrá visualizar desde la vista <i>Vista_graficar</i> . | |
| Paquete: <code>cu.andromeda.Gráficos</code> | |

Tabla 21. Tarea 8: Implementar la funcionalidad *Graficar*.

| Tarea | |
|---|--------------------------------------|
| Número de tarea: 8 | Número de HU: 3 |
| Nombre de la tarea: Implementar la funcionalidad <i>Graficar</i> . | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha de inicio: 29 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Se implementa en la vista <i>Vista_graficar</i> la funcionalidad <i>Graficar</i> , donde se cargaran los gráficos. Inicialmente se implementa para el gráfico de barras, estableciendo las dimensiones del gráfico, así como los nombres de los ejes, y el nombre del gráfico. | |

Tabla 22. Tarea 9: Estudiar cómo obtener gráficos lineales con la librería *JfreeChart*.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 9 | Número de HU: 4 |
| Nombre de la tarea: Estudiar cómo obtener gráficos lineales con la librería <i>JfreeChart</i> . | |
| Tipo de tarea: Preparación | Estimación: 2 días |
| Fecha de inicio: 28 de marzo de 2015 | Fecha fin: 29 de marzo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar en la librería <i>JfreeChart</i> como obtener los gráficos lineales que se desean implementar en la aplicación. | |

Tabla 23. Tarea 10: Implementar la clase *gráficoLineal*.

| Tarea | |
|---|--------------------------------------|
| Número de tarea: 10 | Número de HU: 4 |
| Nombre de la tarea: Implementar la clase <i>gráficoLineal</i> . | |
| Tipo de tarea: Preparación | Estimación: 3 día |
| Fecha de inicio: 29 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Se crea la clase <i>gráficoLineal</i> . La misma es la encargada de generar el gráfico lineal con las propiedades inicialmente establecidas, que posteriormente el usuario podrá visualizar desde la vista <i>Vista_graficar</i> . | |
| Paquete: cu.andromeda.Gráficos | |

Tabla 24. Tarea 11: Implementar la funcionalidad *Graficar*.

| Tarea | |
|---|--------------------------------------|
| Número de tarea: 11 | Número de HU: 4 |
| Nombre de la tarea: Implementar la funcionalidad <i>Graficar</i> . | |
| Tipo de tarea: Desarrollo | Estimación: 3 día |
| Fecha de inicio: 29 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Se implementa en la vista <i>Vista_graficar</i> la funcionalidad <i>Graficar</i> , donde se cargaran los gráficos. Se implementa para el gráfico lineal, estableciendo las dimensiones del gráfico, así como los nombres de los ejes, y el nombre del gráfico. | |

Anexo 7. Descripción de tareas Iteración 2.

Tabla 25. Tarea No 12: Estudiar el método de la regresión lineal simple por mínimos cuadrados.

| Tarea | |
|---|--------------------------------------|
| Número de tarea: 12 | Número de HU: 2 |
| Nombre de la tarea: Estudiar el método de la regresión lineal simple por mínimos cuadrados. | |
| Tipo de tarea: Preparación | Estimación: 5 días |
| Fecha de inicio: 2 de abril de 2015 | Fecha fin: 6 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar cómo funciona el método matemático que se utilizará para implementar la solución propuesta y saber cómo funciona el mismo. | |

Tabla 26. Tarea No 13: Implementar las clases regresión.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 13 | Número de HU: 2 |
| Nombre de la tarea: Implementar las clases regresión. | |
| Tipo de tarea: Desarrollo | Estimación: 12 días |
| Fecha de inicio: 6 de abril de 2015 | Fecha fin: 18 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar la clase <i>regresión</i> , que va a necesitar los valores de la X (variable predictora) y la Y (variable de respuesta) para aplicar cada uno de los pasos que conforman la regresión lineal por mínimos cuadrados. Estos datos son obtenidos de la base de datos. En la clase regresión se aplican los ajustes del método matemático por la ecuación definida $y = ax + c$. | |
| Paquete: cu.andromeda.Regresión_Lineal | |

Tabla 27. Tarea No 14: Diseñar e implementar la vista Vista_comparar_modelos.

| Tarea | |
|--|---------------------------------------|
| Número de tarea: 14 | Número de HU: 2 |
| Nombre de la tarea: Diseñar e implementar la vista Vista_comparar_modelos. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha de inicio: 19 de abril de 2015 | Fecha fin: 21 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Diseñar e implementar la vista donde se obtendrán los resultados de la aplicación del método matemático y el gráfico comparativo de los diferentes modelos, así como la recta trazada la cual definirá cuál de los modelos tiene mejor ajuste. | |
| Paquete: cu.andromeda.Vistas | |

Tabla 28. Tarea No 15: Estudiar cómo obtener gráficos de dispersión con la librería JfreeChart.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 15 | Número de HU: 5 |
| Nombre de la tarea: Estudiar cómo obtener gráficos de dispersión con la librería JfreeChart. | |
| Tipo de tarea: Preparación | Estimación: 2 días |
| Fecha de inicio: 21 de abril de 2015 | Fecha fin: 22 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar en la librería JfreeChart como obtener los gráficos lineales que se desean implementar en la aplicación. | |

Tabla 29.Tarea No 16: Implementar la clase *gráficoDispersión*.

| Tarea | |
|---|--------------------------------|
| Número de tarea: 16 | Número de HU: 5 |
| Nombre de la tarea: Implementar la clase <i>gráficoDispersión</i> . | |
| Tipo de tarea: Desarrollo | Estimación:2 días |
| Fecha de inicio:23 de abril de 2015 | Fecha fin: 24 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| <p>Descripción: Se crea la clase <i>gráficoDispersión</i>. La misma es la encargada de generar el gráfico de dispersión con las propiedades inicialmente establecidas, que posteriormente el usuario podrá visualizar desde la vista <i>Vista_graficar</i>, y en este caso, también en la vista <i>Vista_Comparar_modelos</i>.</p> | |
| Paquete: cu.andromeda.Gráficos | |

Tabla 30.Tarea No 17: Implementar la funcionalidad *Graficar*.

| Tarea | |
|--|--------------------------------|
| Número de tarea: 17 | Número de HU: 5 |
| Nombre de la tarea: Implementar la funcionalidad <i>Graficar</i> . | |
| Tipo de tarea: Desarrollo | Estimación:2 días |
| Fecha de inicio:24 de abril de 2015 | Fecha fin: 26 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| <p>Descripción: Se implementa en la vista <i>Vista_graficar</i> la funcionalidad <i>Graficar</i>, donde debe mostrarse el gráfico de dispersión generado, estableciendo las dimensiones del gráfico, así como los nombres de los ejes, y el nombre del gráfico.</p> | |

Tabla 31.Tarea No 18: Implementar la funcionalidad *Graficar y calcular*.

| Tarea | |
|---|--------------------------------|
| Número de tarea: 18 | Número de HU: 5 |
| Nombre de la tarea: Implementar la funcionalidad <i>Graficar y calcular</i> . | |
| Tipo de tarea: Desarrollo | Estimación:2 días |
| Fecha de inicio:26 de abril de 2015 | Fecha fin: 27 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| <p>Descripción: Se implementa en la vista <i>Vista_Comparar_modelos</i>. Esta funcionalidad es la encargada de mostrar los resultados obtenidos del análisis estadístico, y además de mostrar la recta de ajuste, la cual permitirá obtener el mejor ajuste.</p> | |

Tabla 32.Tarea No 19: Estudiar como exportar gráficos en formato .PNG con la librería JfreeChart.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 19 | Número de HU: 6 |
| Nombre de la tarea: Estudiar como exportar gráficos en formato .PNG con la librería JfreeChart. | |
| Tipo de tarea: Preparación | Estimación: 1 días |
| Fecha de inicio: 27 de abril de 2015 | Fecha fin: 28 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Uno de los requerimientos del sistema es importar los gráficos en formato .PNG. Se hace necesario estudiar las utilidades que brinda la librería JfreeChart para utilizarlo en la implementación. | |

Tabla 33.Tarea No 20: Implementar la funcionalidad que permita guardar gráficos en el formato .PNG.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 20 | Número de HU: 6 |
| Nombre de la tarea: Implementar la funcionalidad que permita guardar gráficos en el formato .PNG. | |
| Tipo de tarea: Desarrollo | Estimación: 1 días |
| Fecha de inicio: 28 de abril de 2015 | Fecha fin: 28 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar la funcionalidad que permita guardar el gráfico en el formato .PNG, con el objetivo de brindarle esta facilidad al usuario. | |

Anexo 8. Descripción de tareas Iteración 3.

Tabla 34.Tarea No 21: Estudiar como exportar gráficos en formato .SVG con la librería JfreeChart.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 21 | Número de HU: 7 |
| Nombre de la tarea: Estudiar como exportar gráficos en formato .SVG con la librería JfreeChart. | |
| Tipo de tarea: Preparación | Estimación: 1 días |
| Fecha de inicio: 28 de abril de 2015 | Fecha fin: 29 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Uno de los requerimientos del sistema es importar los gráficos en formato .SVG. Se hace necesario estudiar las utilidades que brinda la librería JfreeChart para utilizarlo en la implementación. | |

Tabla 35.Tarea No 22: Implementar la funcionalidad que permita guardar gráficos en el formato .SVG.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 22 | Número de HU: 7 |
| Nombre de la tarea: Implementar la funcionalidad que permita guardar gráficos en el formato .SVG. | |
| Tipo de tarea: Desarrollo | Estimación: 1 días |
| Fecha de inicio: 29 de abril de 2015 | Fecha fin: 30 de abril de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar la funcionalidad que permita guardar el gráfico en el formato .SVG, con el objetivo de brindarle esta facilidad al usuario. | |

Tabla 36.Tarea No 23: Estudiar en la librería JFreeChart como exportar en formato .PDF.

| Tarea | |
|---|---------------------------------------|
| Número de tarea: 23 | Número de HU: 8 |
| Nombre de la tarea: Estudiar en la librería JFreeChart como exportar en formato .PDF. | |
| Tipo de tarea: Preparación | Estimación: 1 días |
| Fecha de inicio: 30 de abril de 2015 | Fecha fin: 1ro de mayo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar las opciones que brinda la librería JFreeChart para exportar los gráficos generados en la aplicación en formato .PDF | |

Tabla 37.Tarea No 24: Implementar la funcionalidad que permita guardar gráficos en el formato .PDF.

| Tarea | |
|---|-------------------------------------|
| Número de tarea: 24 | Número de HU: 8 |
| Nombre de la tarea: Implementar la funcionalidad que permita guardar gráficos en el formato .PDF. | |
| Tipo de tarea: Desarrollo | Estimación: 1 día |
| Fecha de inicio: 2 de mayo de 2015 | Fecha fin: 3 de mayo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar la funcionalidad que permita guardar el gráfico en el formato .PDF, con el objetivo de brindarle esta facilidad al usuario. | |

Tabla 38. Tarea No 25: Estudiar en la librería JfreeChart las opciones de personalización disponibles para los gráficos generados.

| Tarea | |
|--|-------------------------------------|
| Número de tarea: 25 | Número de HU: 9 |
| Nombre de la tarea: Estudiar en la librería JfreeChart las opciones de personalización disponibles para los gráficos generados. | |
| Tipo de tarea: Preparación | Estimación: 1 días |
| Fecha de inicio: 3 de mayo de 2015 | Fecha fin: 4 de mayo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Estudiar las opciones que brinda la librería JfreeChart para ofrecerle opciones de personalización de los gráficos a los usuarios. | |

Tabla 39. Tarea No 26: Implementar las opciones de personalización de los gráficos.

| Tarea | |
|---|-------------------------------------|
| Número de tarea: 26 | Número de HU: 9 |
| Nombre de la tarea: Implementar las opciones de personalización de los gráficos. | |
| Tipo de tarea: Desarrollo | Estimación: 1 días |
| Fecha de inicio: 4 de mayo de 2015 | Fecha fin: 5 de mayo de 2015 |
| Programador responsable: Renides Leyva Ramírez | |
| Descripción: Implementar las opciones de personalización de los gráficos que se generen con la herramienta. La personalización consiste en poder cambiar los colores, tipos de letra, tamaño, nombre de los gráficos, de los ejes e intervalos del ajuste. | |

Anexo 9. Datos de muestra para aplicar la regresión lineal.

| x | directa | multwall | Dominant Path | Ray Tracing | keenan-molloy | Prop. Directa | Prop. Multi-Wall | Prop. Domin Path | Prop. Ray Tracing | Keenan-Molloy |
|-------|---------|----------|---------------|-------------|---------------|---------------|------------------|------------------|-------------------|---------------|
| 0,21 | -30,71 | -30,75 | -31,12 | -17,46 | -30,71 | -31,71 | -30,76 | -31,12 | -17,46 | -30,71 |
| 1,21 | -36,11 | -36,05 | -36,71 | -22,75 | -36,11 | -36,1 | -36,04 | -36,71 | -22,75 | -36,1 |
| 2,21 | -40,50 | -40,35 | -41,45 | -27,06 | -40,50 | -40,5 | -40,35 | -41,45 | -27,06 | -40,5 |
| 3,21 | -43,56 | -43,36 | -44,91 | -30,06 | -43,56 | -45,56 | -43,35 | -44,9 | -30,06 | -43,56 |
| 4,21 | -45,87 | -45,62 | -47,63 | -32,32 | -45,87 | -46,87 | -45,62 | -47,63 | -32,32 | -45,87 |
| 5,21 | -47,71 | -47,42 | -49,90 | -35,02 | -47,71 | -47,71 | -47,42 | -49,9 | -35,02 | -47,71 |
| 6,21 | -49,24 | -48,92 | -51,86 | -39,52 | -49,24 | -51,24 | -48,92 | -51,86 | -39,51 | -49,24 |
| 7,21 | -50,55 | -50,20 | -53,61 | -43,36 | -50,55 | -51,55 | -50,2 | -53,61 | -43,36 | -50,55 |
| 8,21 | -51,69 | -51,32 | -55,20 | -46,71 | -51,69 | -51,69 | -51,32 | -55,2 | -46,71 | -51,69 |
| 9,21 | -52,70 | -52,31 | -56,66 | -49,69 | -52,70 | -53,7 | -52,31 | -56,66 | -49,68 | -52,7 |
| 10,21 | -53,61 | -53,20 | -58,01 | -52,36 | -53,61 | -54,61 | -53,2 | -58,01 | -52,35 | -53,61 |
| 11,21 | -54,43 | -54,01 | -59,29 | -54,78 | -54,43 | -55,43 | -54,01 | -59,29 | -54,78 | -54,43 |
| 12,21 | -55,19 | -54,75 | -60,50 | -57,00 | -55,19 | -56,18 | -54,75 | -60,5 | -57 | -55,18 |
| 13,21 | -55,88 | -55,43 | -61,65 | -59,04 | -55,88 | -56,88 | -55,43 | -61,65 | -59,04 | -55,88 |
| 14,21 | -56,53 | -56,06 | -62,75 | -60,94 | -56,53 | -57,52 | -56,06 | -62,75 | -60,94 | -56,52 |
| 15,21 | -57,13 | -56,65 | -63,81 | -62,71 | -57,13 | -57,13 | -56,65 | -63,81 | -62,7 | -57,13 |
| 16,21 | -57,69 | -57,20 | -64,83 | -64,36 | -57,69 | -57,69 | -57,2 | -64,83 | -64,36 | -57,69 |
| 17,21 | -58,22 | -57,72 | -65,82 | -65,92 | -58,22 | -59,22 | -57,72 | -65,82 | -65,92 | -58,22 |
| 18,21 | -58,72 | -58,21 | -66,78 | -67,39 | -58,72 | -60,72 | -58,21 | -66,78 | -67,39 | -58,72 |
| 19,21 | -59,19 | -58,68 | -67,71 | -68,78 | -59,19 | -61,19 | -58,68 | -67,71 | -68,78 | -59,19 |
| 20,21 | -59,64 | -59,12 | -68,62 | -70,10 | -59,64 | -59,64 | -59,12 | -68,62 | -70,1 | -59,64 |
| 21,21 | -60,07 | -59,54 | -69,51 | -71,35 | -60,07 | -60,07 | -59,54 | -69,51 | -71,35 | -60,07 |
| 22,21 | -60,47 | -59,94 | -70,38 | -72,55 | -60,47 | -62,47 | -59,94 | -70,38 | -72,55 | -60,47 |
| 23,21 | -60,86 | -60,32 | -71,23 | -73,70 | -60,86 | -60,86 | -60,32 | -71,23 | -73,7 | -60,86 |
| 24,21 | -61,24 | -60,68 | -72,07 | -74,80 | -61,24 | -61,24 | -60,68 | -72,07 | -74,8 | -61,24 |
| 25,21 | -61,60 | -61,04 | -72,89 | -75,85 | -61,60 | -62,6 | -61,03 | -72,89 | -75,85 | -61,6 |
| 26,21 | -61,94 | -61,37 | -73,70 | -76,87 | -61,94 | -61,94 | -61,37 | -73,7 | -76,86 | -61,94 |
| 27,21 | -62,27 | -61,70 | -74,49 | -77,84 | -62,27 | -63,27 | -61,7 | -74,49 | -77,84 | -62,27 |
| 28,21 | -62,59 | -62,01 | -75,27 | -78,78 | -62,59 | -63,59 | -62,01 | -75,27 | -78,78 | -62,59 |
| 29,21 | -62,90 | -62,31 | -76,05 | -79,69 | -62,90 | -64,9 | -62,31 | -76,05 | -79,69 | -62,9 |
| 30,21 | -63,20 | -62,61 | -76,81 | -80,56 | -63,20 | -63,2 | -62,61 | -76,81 | -80,56 | -63,2 |
| 31,21 | -63,49 | -62,89 | -77,56 | -81,41 | -63,49 | -64,49 | -62,89 | -77,56 | -81,41 | -63,49 |
| 32,21 | -63,77 | -63,16 | -78,30 | -82,23 | -63,77 | -64,77 | -63,16 | -78,3 | -82,23 | -63,77 |
| 33,21 | -64,04 | -63,43 | -79,04 | -83,03 | -64,04 | -64,04 | -63,43 | -79,04 | -83,03 | -64,04 |
| 34,21 | -64,30 | -63,69 | -79,77 | -83,80 | -64,30 | -64,3 | -63,68 | -79,77 | -83,8 | -64,3 |
| 35,21 | -64,55 | -63,94 | -80,49 | -84,55 | -64,55 | -64,55 | -63,94 | -80,49 | -84,55 | -64,55 |
| 36,21 | -64,80 | -64,18 | -81,20 | -85,28 | -64,80 | -65,8 | -64,18 | -81,2 | -85,28 | -64,8 |
| 37,21 | -65,04 | -64,41 | -81,91 | -85,99 | -65,04 | -65,04 | -64,41 | -81,91 | -85,99 | -65,04 |
| 38,21 | -65,28 | -64,65 | -82,61 | -86,68 | -65,28 | -66,28 | -64,65 | -82,61 | -86,68 | -65,28 |
| 39,21 | -65,51 | -64,87 | -83,30 | -87,36 | -65,51 | -67,51 | -64,87 | -83,3 | -87,35 | -65,51 |
| 40,21 | -65,73 | -65,09 | -83,99 | -88,01 | -65,73 | -66,73 | -65,09 | -83,99 | -88,01 | -65,73 |
| 41,21 | -65,95 | -65,30 | -84,67 | -88,65 | -65,95 | -65,95 | -65,3 | -84,67 | -88,65 | -65,95 |
| 42,21 | -66,16 | -65,51 | -85,35 | -89,28 | -66,16 | -68,16 | -65,51 | -85,35 | -89,28 | -66,16 |
| 43,21 | -66,37 | -65,71 | -86,02 | -89,89 | -66,37 | -66,37 | -65,71 | -86,02 | -89,88 | -66,37 |
| 44,21 | -66,57 | -65,91 | -86,69 | -90,48 | -66,57 | -68,57 | -65,91 | -86,69 | -90,48 | -66,57 |
| 45,21 | -66,77 | -66,11 | -87,36 | -91,06 | -66,77 | -68,77 | -66,11 | -87,36 | -91,06 | -66,77 |
| 46,21 | -66,96 | -66,30 | -88,02 | -91,63 | -66,96 | -66,96 | -66,3 | -88,02 | -91,63 | -66,96 |
| 47,21 | -67,15 | -66,48 | -88,67 | -92,19 | -67,15 | -67,15 | -66,48 | -88,67 | -92,19 | -67,15 |
| 48,21 | -67,34 | -66,66 | -89,33 | -92,74 | -67,34 | -67,34 | -66,66 | -89,32 | -92,74 | -67,34 |
| 49,21 | -67,52 | -66,84 | -89,97 | -93,27 | -67,52 | -68,52 | -66,84 | -89,97 | -93,27 | -67,52 |

Figura 29 Tabla de datos muestrales.

Anexo 10. Casos de prueba de aceptación para la Iteración 1.

Tabla 40. CP Visualizar gráficos de barras con datos insertados en la base de datos.

| Caso de prueba de aceptación | |
|----------------------------------|---|
| Código: | HU3_P1 Historia de Usuario: 3 |
| Nombre: | Visualizar gráficos de barras con datos insertados en la base de datos. |
| Descripción: | Prueba para la funcionalidad de visualizar los resultados de una consulta a la base de datos, con los datos obtenidos de la muestra insertada en la base de datos de la aplicación. |
| Condiciones de Ejecución: | Debe contarse con datos dentro de la tabla a la cual se realiza la consulta de la aplicación. |

| |
|--|
| Entrada/ Pasos de ejecución: El usuario debe desplegar el menú “ <i>Opciones</i> ”, seleccionar el submenú “ <i>Grafica</i> ”. Ya en la vista el usuario debe seleccionar el JCheckBox “ <i>barras</i> ”, activando la casilla que se encuentran en la parte superior izquierda de la vista Graficar. |
| Resultado Esperado: La herramienta muestra una ventana con el gráfico generado en un tiempo menor a 5 segundos. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 41. CP Visualizar gráficos lineales con datos insertados en la base de datos.

| Caso de prueba de aceptación |
|--|
| Código: HU3_P2 Historia de Usuario: 3 |
| Nombre: Visualizar gráficos lineales con datos insertados en la base de datos. |
| Descripción: Prueba para la funcionalidad de visualizar los resultados de una consulta a la base de datos, con los datos obtenidos de la muestra insertada en la base de datos de la aplicación. |
| Condiciones de Ejecución: Debe contarse con datos dentro de la tabla a la cual se realiza la consulta de la aplicación. |
| Entrada/ Pasos de ejecución: El usuario debe desplegar el menú “ <i>Opciones</i> ”, seleccionar el submenú “ <i>Grafica</i> ”. Ya en la vista el usuario debe seleccionar el JCheckBox “ <i>lineal</i> ”, activando la casilla que se encuentran en la parte superior izquierda de la vista Graficar. |
| Resultado Esperado: La herramienta muestra una ventana con el gráfico generado en un tiempo menor a 5 segundos. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 42. CP Permitir seleccionar el tipo de gráfico a visualizar por el usuario.

| Caso de prueba de aceptación |
|---|
| Código: HU1_P3 Historia de Usuario: 1 |
| Nombre: Permitir seleccionar el tipo de gráfico a visualizar por el usuario. |
| Descripción: Prueba para la funcionalidad “ Permitir seleccionar el tipo de gráfico para el análisis. ” |
| Condiciones de Ejecución: El usuario debe tener la vista Graficar en ejecución. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic sobre el menú “ <i>Opciones</i> ”, luego en el submenú “ <i>Grafica</i> ”. Con la vista Graficar en ejecución el usuario podrá seleccionar él, o los gráficos que desee se muestren en pantalla. |
| Resultado Esperado: Se puedan visualizar los gráficos con los que cuenta la aplicación. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Anexo 11. Casos de pruebas de aceptación para la Iteración 2.

Tabla 43. CP Visualizar gráfico de dispersión con datos insertados en la base de datos.

| Caso de prueba de aceptación |
|---|
| Código: HU5_P4 Historia de Usuario: 5 |
| Nombre: Visualizar gráfico de dispersión con datos insertados en la base de datos. |
| Descripción: Prueba para la funcionalidad “ Obtener gráfico de dispersión a partir de la selección del usuario. ” |
| Condiciones de Ejecución: Debe contarse con datos dentro de la tabla a la cual se realiza la consulta de la aplicación. |
| Entrada/ Pasos de ejecución: El usuario debe desplegar el menú “ <i>Opciones</i> ”, seleccionar el submenú “ <i>Graficar</i> ”. Ya en la vista el usuario debe seleccionar el JCheckBox “ <i>dispersión</i> ”, activando la casilla que se encuentran en la parte superior izquierda de la vista Graficar. |
| Resultado Esperado: La herramienta muestra una ventana con el gráfico generado en un tiempo menor a 5 segundos. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 44. CP Permitir exportar gráfico en formato .PNG.

| Caso de prueba de aceptación |
|--|
| Código: HU6_P5 Historia de Usuario: 6 |
| Nombre: Permitir exportar gráfico en formato .PNG. |
| Descripción: Prueba para la funcionalidad “ Exportar gráficos generado en formato .PNG. ” |
| Condiciones de Ejecución: El usuario debe contar con al menos un gráfico generado por la herramienta. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic en el menú “ <i>Opciones</i> ”, seguidamente se despliegan los submenús correspondientes y debe seleccionar “ <i>Graficar</i> ” o “ <i>Comparación de los modelos</i> ”. El usuario debe dar clic derecho sobre el gráfico generado, se despliega un menú donde se encuentra la opción “ <i>Save as</i> ” y escoge el formato .PNG. |
| Resultado Esperado: Se puede exportar cualquier gráfico generado por la herramienta en el formato seleccionado. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 45. CP Obtener comparación de los modelos de propagación con ajuste por mínimos cuadrados.

| Caso de prueba de aceptación |
|---|
| Código: HU2_P6 Historia de Usuario: 2 |
| Nombre: Obtener comparación de los modelos de propagación con ajuste por mínimos cuadrados. |
| Descripción: Prueba para la funcionalidad “ Aplicar regresión lineal por mínimos cuadrados al parámetro seleccionado. ” |
| Condiciones de Ejecución: El usuario debe contar con un gráfico de dispersión creado. |

| |
|--|
| Entrada/ Pasos de ejecución: El usuario debe desplegar el menú “ <i>Opciones</i> ”, seleccionar el submenú “ <i>Comparación de los modelos</i> ”. Luego de estar en la vista seleccionada dar clic sobre el botón “ <i>Graficar y calcular</i> ”. |
| Resultado Esperado: La herramienta muestra en la ventana, el gráfico de dispersión obtenido, la línea producto de aplicar el método de análisis con los datos de muestra y los valores de mínimos cuadrados de cada modelo de propagación, así como el valor de correlación de la recta encontrada. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Anexo 12. Casos de pruebas de aceptación para la Iteración 3.

Tabla 46. CP Permitir exportar gráfico en formato .SVG.

| Caso de prueba de aceptación |
|--|
| Código: HU7_P7 Historia de Usuario: 7 |
| Nombre: Permitir exportar gráfico en formato .SVG. |
| Descripción: Prueba para la funcionalidad “ Exportar gráficos generado en formato .SVG ” |
| Condiciones de Ejecución: El usuario debe contar con un gráfico generado por la herramienta. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic en el menú “ <i>Opciones</i> ”, seguidamente se despliegan los submenús correspondientes y debe seleccionar “ <i>Graficar</i> ” o “ <i>Comparación de los modelos</i> ”. El usuario debe generar al menos un gráfico, luego dar clic derecho encima de este, se despliega un menú donde se encuentra la opción “ <i>Save as</i> ” y escoge el formato .SVG. |
| Resultado Esperado: Se puede exportar cualquier gráfico generado por la herramienta en el formato seleccionado. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 47. CP Permitir exportar gráfico en formato .PDF.

| Caso de prueba de aceptación |
|--|
| Código: HU8_P8 Historia de Usuario: 8 |
| Nombre: Permitir exportar gráfico en formato .PDF. |
| Descripción: Prueba para la funcionalidad “ Exportar gráficos generado en formato .PDF ” |
| Condiciones de Ejecución: El usuario debe contar con un gráfico generado por la herramienta. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic en el menú “ <i>Opciones</i> ”, seguidamente se despliegan los submenús correspondientes y debe seleccionar “ <i>Graficar</i> ” o “ <i>Comparación de los modelos</i> ”. El usuario debe generar al menos un gráfico, luego dar clic derecho encima de este, se despliega un menú donde se encuentra la opción “ <i>Save as</i> ” y escoge el formato .PDF. |
| Resultado Esperado: Se puede exportar cualquier gráfico generado por la herramienta en el formato seleccionado. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 48. CP Permitir Acercar o Alejar todos o uno de los ejes, de acuerdo a selección del usuario.

| Caso de prueba de aceptación |
|--|
| Código: HU9_P9 Historia de Usuario: 9 |
| Nombre: Permitir acercar o alejar todos o uno de los ejes, de acuerdo a selección del usuario. |
| Descripción: Prueba para la funcionalidad “ Personalizar los gráficos generados. ” |
| Condiciones de Ejecución: El usuario debe contar con un gráfico generado por la herramienta. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic en el menú “ <i>Opciones</i> ”, seguidamente se despliegan los submenús correspondientes y debe seleccionar “ <i>Graficar</i> ” o “ <i>Comparación de los modelos</i> ”. El usuario debe generar al menos un gráfico, luego dar clic derecho sobre el gráfico generado, se despliega un menú donde se encuentran las opciones “ <i>Acercar</i> y “ <i>Alejar</i> ”. Cada una de estas con las Opciones de “ <i>Todos los ejes</i> ”/ “ <i>Eje horizontal</i> ”/ “ <i>Eje vertical</i> ” |
| Resultado Esperado: De acuerdo a la selección se permitirá acercar o alejar, todos los ejes, o simplemente el eje horizontal o el eje vertical. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Tabla 49. CP Permitir personalizar el gráfico generado.

| Caso de prueba de aceptación |
|---|
| Código: HU9_P10 Historia de Usuario: 9 |
| Nombre: Permitir personalizar el gráfico generado. |
| Descripción: Prueba para la funcionalidad “ Personalizar los gráficos generados. ” |
| Condiciones de Ejecución: El usuario debe contar con al menos un gráfico generado por la herramienta. |
| Entrada/ Pasos de ejecución: El usuario debe dar clic en el menú “ <i>Opciones</i> ”, seguidamente se despliegan los submenús correspondientes y debe seleccionar “ <i>Graficar</i> ” o “ <i>Comparación de los modelos</i> ”. El usuario debe generar al menos un gráfico, se despliega un menú donde se encuentra la opción “ <i>Propiedades</i> ”. La ventana de propiedades debe permitirle al usuario modificar el título, el trazo de ambos ejes (x, y), así como el color de fondo y de los trazos del gráfico. |
| Resultado Esperado: La herramienta muestra una ventana con todas las propiedades del gráfico, en un tiempo menor de 3 segundos. |
| Evaluación de la Prueba: Prueba satisfactoria. |

Anexo 13. Resultados de pruebas unitarias para Andromeda_Grapher.

La siguiente imagen muestra las pruebas realizadas a la clase *SQLiteconexion*, la cual contiene los métodos necesarios para importar los datos necesarios para la comparación de los modelos y la aplicación del método matemático de regresión línea simple. Los resultados fueron satisfactorios.

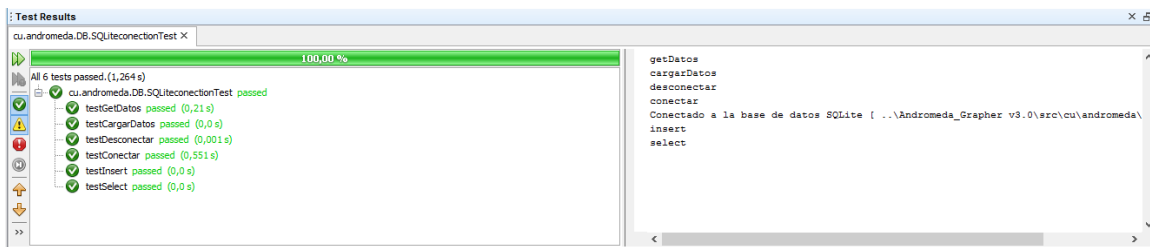


Figura 30 Resultados de prueba unitaria a la clase SQLiteconexion

La siguiente imagen muestra los resultados de las pruebas realizadas a la clase *regresión*, la cual contiene los métodos necesarios para realizar el proceso de análisis estadístico de los datos de muestra con los que cuenta el módulo. Los resultados fueron satisfactorios.

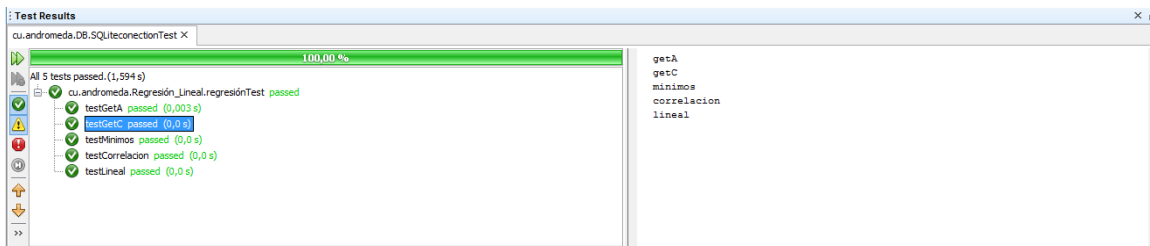


Figura 31 Resultados de prueba unitaria a la clase regresión