



Universidad de las Ciencias Informáticas



Visor tridimensional de entornos interiores para la herramienta Andrómeda

Autores: Javier Bandomo Ruíz

Orlando Torres Saavedra

Tutores: MSc. Ismael Armando Nodarse Mora

Ing. Osdalme Fuentes Colina

La Habana, junio de 2012

Declaramos que somos los únicos autores del trabajo “Visor tridimensional de entornos interiores para la herramienta Andrómeda” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores:

Orlando Torres Saavedra

Javier Bandomo Ruíz

Tutores:

**MSc. Ismael Armando Nodarse
Mora**

Ing. Osdalme Fuentes Colina

“La naturaleza nos ha dado las semillas del conocimiento, no el conocimiento mismo.”

Lucio Anneo Séneca

Dedicatoria

Orlando:

“Dedico especialmente este trabajo de diploma a mi tesoro más sagrado, mis padres.
A mi hermana por su incondicional amistad. A mi sobrinita bella.
A mi familia, amigos y a mi novia”.

Javier:

“A toda mi familia y amigos.”

Agradecimientos

Orlando:

Quiero agradecer de manera muy especial a los protagonistas de estos resultados alcanzados en el largo periodo de estudiante de universidad, a mis queridos padres. A ti mamá, gracias por ser la luz de mi camino cuando apenas puedo mirar en la oscuridad de los momentos difíciles de la vida. A ti papa, gracias por ser mi guía profesional y mostrarme que solo es cuestión de proponerse tus propios logros y el tiempo se encarga de mostrarte los resultados. A ambos gracias por confiar en mis decisiones, por su apoyo incondicional y por todo el amor que me han dado, los amo. A mi hermana por ser tan especial en mi vida y ocupar gran parte de mi corazón. A mi sobrina bella, a mi familia, gracias a todos por preocuparse siempre por mí y ayudarme a hacer realidad mis sueños. A mi novia gracias por ser la luna de vida, por ayudarme a construir un futuro juntos, por regalarme amor con solo tu mirada, por ser mi musa, mi inspiración, por tu confianza, gracias por todo amor. A mis amigos inseparables Román y Pablo por permitirme ser siempre árbitro en sus riñas cotidianas, por su confianza y amistad en todos estos años, gracias amigos. A Yasiel, Javier, Raiko, Omar, a los miembros del “Croquet Team”, a mis compañeros de aula, a mis tutores por guiarnos en este reto, a los profesores que contribuyeron con mi formación profesional, a todos ustedes de corazón, mil gracias.

Javier:

A mis padres y mi hermana: Por estar siempre, por ser personas especiales. Por darme el privilegio de vivir en una familia, una FAMILIA ESPECIAL, de la cual me siento orgulloso, orgulloso porque tengo un PADRE que vive por sus hijos, y una MADRE que incontable son las noches que no ha conseguido sueño por velarnos el nuestro. A mi hermana por ser mi faro, mi meta, tal vez nunca la alcance, pero sé que siempre va a estar ahí. Por enseñarme y educarme, por hacerme ser quien soy.

A mis amigos: Por sus consejos, su apoyo, por estar siempre, incluso cuando no se les necesita ;)

A mis tutores: Primero por el reto y luego por la ayuda.

En fin, a todo aquel que apporto su “bit”.

Resumen

Las Redes Inalámbricas de Sensores y Actuadores, WSAN por sus siglas en Inglés, son un conjunto de dispositivos autónomos distribuidos en un área. Esta tecnología de red tiene gran aceptación por su aplicación en los Sistemas de Control de Edificios debido a las ventajas que brinda. El presente trabajo consiste en el desarrollo de un visor de entornos interiores para el diseño del despliegue de WSAN, de forma tal que se pueda interactuar con un modelo tridimensional lo más cercano posible a la realidad.

El desarrollo de la propuesta fue guiado por la metodología XP, UML como lenguaje de modelado y Java para la implementación, se empleó la librería Java 3D para el modelado tridimensional, se utilizó el NetBeans como Entorno de Desarrollo Integrado y Visual Paradigm for UML como herramienta CASE. En su primera versión, el visor es capaz de visualizar modelos IFC en tres dimensiones que hayan sido generados por herramientas de Modelado de Información para la Edificación, permitiendo además su edición y gestión de componentes de WSAN en el entorno tridimensional. La investigación finaliza con las pruebas funcionales del software, las cuales demostraron la correcta implementación de todas sus funcionalidades.

Palabras claves: IFC, modelo 3D, Redes Inalámbricas de Sensores y Actuadores, Sistemas de Control de Edificios, visor.



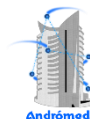
Índice

Introducción	1
1. Fundamentación Teórica.....	7
1.1 Visores IFC.....	7
1.1.1 DDS-CAD Viewer.....	7
1.1.2 Solibri Model Viewer	8
1.1.3 Tekla BIM sight	9
1.1.4 Nemetschek IFC Viewer	10
1.2 Lenguaje de programación.....	11
1.3 Librerías 3D para el desarrollo	12
1.3.1 JOGL.....	13
1.3.2 JGL.....	13
1.3.3 Java3D	13
1.3.4 Selección de la librería.....	14
1.4 Librería para cargar ficheros IFC	14
1.5 Entorno de desarrollo	15
1.6 Metodología de desarrollo de software	16
1.7 Herramienta para el modelado.....	16
1.8 Conclusiones.....	17
2. Descripción de la solución propuesta	18
2.1 Usuarios del sistema	18
2.2 Lista de reservas del producto	18
2.3 Aspectos no funcionales del sistema	20
2.4 Exploración.....	21
2.4.1 Historias de Usuario.....	21
2.5 Planificación	25
2.5.1 Iteraciones.....	26
2.5.2 Plan de Entregas.....	27
2.6 Conclusiones.....	27
3. Implementación y Pruebas.....	28
3.1 Descripción de la arquitectura.....	28
3.1.1 Plataforma Andr6meda	32
3.1.2 M6dulos del sistema	32
3.1.3 Ventanas del Visor.....	32
3.1.4 Paquetes del sistema.....	33
3.1.5 Diagrama de clases del sistema	34
3.2 Implementación	36
3.2.1 Primera iteraci6n	36
3.2.2 Segunda Iteraci6n	41
3.2.3 Tercera Iteraci6n	42
3.2.4 Cuarta Iteraci6n.....	46
3.3 Descripci6n del entorno.....	48
3.4 Requerimientos m6nimos.	51
3.5 Pruebas	51
Conclusiones	64
Recomendaciones.....	65
Referencias Bibliogr6ficas	66
Bibliograf6a.....	67
Glosario	69



Índice de figuras

Figura 1. Visualización de un modelo IFC en la herramienta DDS-CAD Viewer	8
Figura 2. Visualización de un modelo IFC en la herramienta Solibri Model Viewer	9
Figura 3. Visualización de un modelo IFC en la herramienta Tekla BIM sight	10
Figura 4. Visualización de un modelo IFC en la herramienta Nemestchek IFC Viewer	11
Figura 5. Diagrama de Componentes del módulo 3D-Model	29
Figura 6. Diagrama de Componentes del módulo Viewer	30
Figura 7. Diagrama de Componentes del módulo IfcModel	31
Figura 8. Interfaz de usuario del visor	33
Figura 9. Diagrama de paquetes del visor	34
Figura 10. Diagrama con las clases empleadas por el visor	35
Figura 11 Interfaz del visor	48

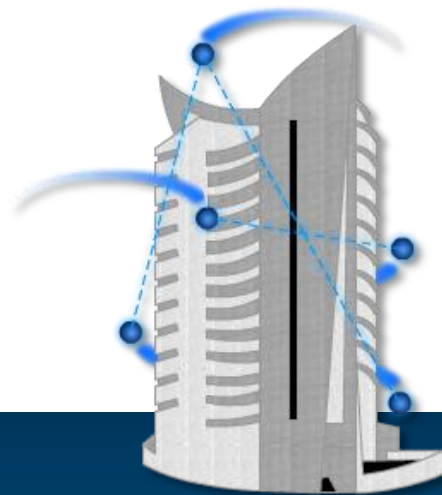


Índice de tablas

Tabla 1. HU 1: Importar modelo IFC	22
Tabla 2. HU 2: Visualizar modelo IFC	22
Tabla 3. HU 3: Permitir la selecci3n de los elementos del modelo	22
Tabla 4. HU 4: Visualizar propiedades de elementos del modelo	23
Tabla 5. HU 5: Opciones de visualizaci3n del modelo3D	23
Tabla 6. HU 6: Permitir el desplazamiento en el modelo 3D	23
Tabla 7. HU 7: Gestionar elemento no estructural	24
Tabla 8. HU 8: Gestionar 1rea de sensado	24
Tabla 9. HU 9: Gestionar dispositivo de sensado	24
Tabla 10. HU 10: Exportar imagen bidimensional del modelo	25
Tabla 11. HU 11: Gestionar proyecto	25
Tabla 12. Plan de Iteraciones	26
Tabla 13. Plan de Entrega	27
Tabla 14. Tarea 1: Estudiar librería OpenIfcTool	37
Tabla 15. Tarea 2: Crear clase Loader.	37
Tabla 16. Tarea 3: Crear clases Project, Building, Storey y Element	37
Tabla 17. Tarea 4: Diseñar Interfaz de usuario del Browser y Viewer	38
Tabla 18. Tarea 5: Implementar el parseo de elementos IFC	38
Tabla 19. Tarea 6: Implementar la vista s3lida	38
Tabla 20. Tarea 7: Implementar la vista lineal	39
Tabla 21. Tarea 8: Implementar la selecci3n de elementos del 1rbol	39
Tabla 22. Tarea 9: Implementar la selecci3n de elementos del visor tridimensional	39
Tabla 23. Tarea 10: Diseñar interfaz Propiedades	40
Tabla 24. Tarea 11: Mostrar propiedades de elementos desde el 1rbol	40
Tabla 25. Tarea 12: Mostrar propiedades de elementos desde el visor	40
Tabla 26. Tarea 13: Diseñar iconos del visor	41
Tabla 27. Tarea 14: Implementar seis vistas.	42
Tabla 28. Tarea 15: Implementar rotaci3n, translaci3n y acercar o alejar	42
Tabla 29. Tarea 16: Implementar el desplazamiento dentro del modelo	42
Tabla 30. Tarea 17: Adicionar elemento no estructural	43
Tabla 31. Tarea 18: Eliminar elemento no estructural	44
Tabla 32. Tarea 19: Editar propiedades del elemento no estructural	44
Tabla 33. Tarea 20: Adicionar 1rea de sensado	44
Tabla 34. Tarea 21: Eliminar 1rea de sensado	45
Tabla 35. Tarea 22: Editar propiedades del 1rea de sensado	45
Tabla 36. Tarea 23: Adicionar dispositivo de sensado	45
Tabla 37. Tarea 24: Eliminar dispositivo de sensado	46
Tabla 38. Tarea 25: Editar propiedades del dispositivo de sensado	46
Tabla 39. Tarea 24: Exportar imagen bidimensional del modelo	47
Tabla 40. Tarea 25: Crear nuevo proyecto	47
Tabla 41. Tarea 26: Abrir proyecto	47
Tabla 42. Tarea 23: Guardar proyecto	48
Tabla 43. CP Importar modelo IFC	52
Tabla 44. CP Visualizar modelo IFC en un 1rbol jer1rquico seg3n su estructura	53
Tabla 45. CP Visualizar un modelo IFC en el 1rea de trabajo	53
Tabla 46. CP Cambiar tipo de visualizaci3n del modelo IFC a s3lida	53
Tabla 47. CP Cambiar tipo de visualizaci3n del modelo IFC a lineal	54
Tabla 48. CP Seleccionar los elementos del modelo IFC desde el 1rbol jer1rquico	54
Tabla 49. CP Seleccionar los elementos del modelo IFC desde la vista tridimensional	54
Tabla 50. CP Visualizar las propiedades del elemento seleccionado desde el 1rbol jer1rquico ..	55
Tabla 51. CP Visualizar las propiedades del elemento seleccionado desde la vista 3D	55
Tabla 52. CP Permitir la rotaci3n de la vista 3D	56
Tabla 53. CP Permitir acercar y alejar la vista 3D	56
Tabla 54. CP Mover la vista 3D	56



Tabla 55. CP Permitir el desplazamiento en el modelo 3D.....	57
Tabla 56. CP Adicionar elementos no estructurales al modelo IFC visualizado.....	57
Tabla 57. CP Eliminar elemento no estructural desde la vista tridimensional	57
Tabla 58. CP Eliminar elemento no estructural desde el 6rbol jer6rquico	58
Tabla 59. CP Editar propiedades del elemento no estructural.....	58
Tabla 60. CP Adicionar el 6rea de sensado en el modelo IFC visualizado	58
Tabla 61. CP Eliminar 6rea de sensado desde la vista tridimensional	59
Tabla 62. CP Eliminar 6rea de sensado desde el 6rbol jer6rquico	59
Tabla 63. CP Editar propiedades del 6rea de sensado.....	59
Tabla 64. CP Adicionar dispositivo de sensado al modelo IFC visualizado	60
Tabla 65. CP Eliminar dispositivo de sensado desde la vista tridimensional	60
Tabla 66. CP Eliminar dispositivo de sensado desde el 6rbol jer6rquico	60
Tabla 67. CP Editar propiedades del dispositivo de sensado	61
Tabla 68. CP Exportar imagen bidimensional del modelo	61
Tabla 69. CP Crear nuevo proyecto	62
Tabla 70. CP Abrir proyecto existente.....	62
Tabla 71. CP Guardar proyecto existente	62



Andrómeda
Andrómeda

Introducción



Introducción

En las últimas décadas la humanidad ha sido favorecida por los avances tecnológicos en distintas áreas de la ciencia, como la biotecnología, nanotecnología e infotecnología. Institutos famosos y acreditados en esta temática como el Instituto Tecnológico de Massachusetts (MIT) han identificado lo más novedoso en tecnología e investigación, destacándose las Redes de Sensores Inalámbricos (Wireless Sensor Networks).

El desarrollo de las redes de sensores tiene su origen cerca de los años 80 con el proyecto Distributed Sensor Networks (DSN) de la Agencia Militar de Investigaciones Avanzadas de Estados Unidos (Defense Advanced Research Projects Agency, DARPA). A finales de los años 90 comienzan las investigaciones fuera del ámbito militar en el proyecto Smartdust, su objetivo era desarrollar un sensor de comunicación cuyo tamaño no superara un milímetro cúbico. Los desarrolladores de este proyecto utilizan por primera vez el término “mote” (mota) para referirse a los nodos sensores. Este proyecto finaliza en el 2001, dando lugar a nuevos proyectos sobre esta línea de investigación.

Con la evolución de esta tecnología surgen los nodos actuadores, que a diferencia de los nodos sensores, toman decisiones y ejecutan acciones sobre el entorno según la información recogida de los sensores sobre el medio físico. Este tipo de redes se conoce como Redes Inalámbricas de Sensores y Actuadores (WSAN según sus siglas en inglés), las cuales poseen características propias tales como: (1)

- Mientras que los sensores son dispositivos pequeños, baratos, con capacidad de comunicación y procesamiento limitados; los actuadores habitualmente consumen más energía y son más caros.
- Se introducen mecanismos de coordinación entre los sensores y los actuadores para lograr una mejor capacidad de reacción ante eventos en tiempo real.
- El número de sensores suele ser muy superior al número de actuadores.

El uso de dispositivos sensores ha contribuido a que formen parte del entorno habitual, debido a su capacidad de medir gran cantidad de fenómenos físicos (humedad, temperatura, luminosidad, ultrasonido, entre otros), evidenciándose mediante su aplicación en diferentes sectores tecnológicos. A continuación se muestran algunos ejemplos: (2)



En la agricultura, combinando sensores como humedad, temperatura y luminosidad se pueden detectar riesgo de heladas, posibles enfermedades de las plantas o la necesidad de riego según el nivel de humedad de la tierra. Con ello se puede controlar las condiciones de viveros e invernaderos y monitorizar al detalle los cultivos de alto rendimiento o delicados, como los viñedos o los frutos tropicales, donde el mínimo cambio en la climatología puede afectar la producción.

En la ganadería una red de sensores cerca de los animales ayudaría a optimizar sus condiciones de crianza. Es posible monitorizar la temperatura a la que se encuentran las crías para mantenerla en los niveles adecuados; medir los niveles de los gases producidos por el ganado como metano (CH₄), amoníaco (NH₃) y sulfuro de hidrógeno (SH₂); o incluso controlar el nivel de estrés de los animales monitorizando la agitación del rebaño con sensores de vibración y movimiento.

A través de una red de sensores se pueden detectar y prevenir incendios forestales, lo cual ayuda a la preservación del medio ambiente. Para ello es posible utilizar sensores de infrarrojos y ultravioleta, para la detección de llamas, calor y gases.

En instituciones hospitalarias la instalación de una red de sensores de comunicación en el entorno cercano a un paciente o persona con limitaciones, permite una atención y control constante. Las enfermedades no son ciencias exactas y surgen imprevistos que pueden evitarse con un control más exhaustivo del paciente. Pueden ser usados sensores de medición de parámetros biométricos, capaces de monitorizar los latidos, la respiración, movimientos, entre otros.

Otra aplicación importante de las WSN es el empleo en Sistemas de Control de Edificaciones (SCE) debido a las ventajas que brinda, dentro de las más significativas se encuentran (2):

- Ahorro energético y supervisión del consumo eléctrico.
- Gestión del personal del edificio.
- Supervisión en tiempo real de eventos.
- Aviso de averías y mantenimiento preventivo.
- Telegestión remota.
- Aumento del confort de los usuarios y estética.
- Detección y gestión eficaz de la seguridad en el complejo.



Lamentablemente, el uso de WSAN en algunos tipos de escenarios puede conllevar al surgimiento de complejos problemas, ejemplo de ello es el despliegue de estas redes dentro de una edificación. El campo eléctrico dentro de un edificio está integrado por un número superior de componentes indirectos que en el caso de un entorno abierto, por lo que en entornos de interior los niveles de señal de Radio Frecuencia (RF) fluctúan en mayor medida que en entornos de exterior; por consiguiente, la cobertura inalámbrica se caracteriza por ser compleja y muchas veces puede resultar incontrolable en el interior de una edificación.

En la actualidad existen varias investigaciones que enfrentan el problema antes mencionado. Algunas soluciones se basan en el empleo de herramientas software que faciliten, de forma efectiva, el despliegue de sensores en el interior de edificaciones. Sin embargo, a pesar de que estas propuestas ofrecen relevantes soluciones en este ámbito de exploración, aún existen enfoques que no han sido abordados y deficiencias sin resolverse, los cuales representan campos de investigación abiertos a nuevas innovaciones. (3)

Las principales deficiencias de las herramientas actuales son:

- Incorrecta selección de los modelos de propagación de la señal de RF.
- En las simulaciones no se tiene en cuenta el tipo de material y los obstáculos presentes en las edificaciones.
- Solo tienen en cuenta la cobertura y la conectividad interna de la red a la hora de generar las propuestas de despliegue.
- Las topologías de red obtenidas no satisfacen los diseños requeridos por los SCE actuales.

Como consecuencias a las deficiencias anteriores y otros problemas identificados en las herramientas actuales, es necesario crear una herramienta software que genere de forma automática el número y la posición de los dispositivos de una WSAN, permitiendo la obtención de topologías de red de tipo estrella, árbol y malla. Esto posibilitará, entre otras cosas, que el diseñador de la red no necesite un elevado conocimiento de comunicaciones inalámbricas y diseño de WSAN, para realizar el despliegue dentro de una edificación. Dicha herramienta llevará por nombre Andrómeda.

Visor tridimensional de entornos interiores para la herramienta Andrómeda

Para llevar a cabo el desarrollo de la herramienta propuesta, se identificaron las principales funcionalidades que la misma debe incluir. Entre las cuales se encuentra la visualización y edición del entorno donde se va a realizar la propuesta de diseño del despliegue de la WSN, por lo que se hace necesario que la herramienta cuente con un visor tridimensional que sea capaz de mostrar el área de despliegue, logrando así un diseño de despliegue que se acerque más a las condiciones reales del lugar, planteándose el siguiente **problema a resolver**:

¿Cómo lograr la visualización y edición de entornos interiores para el diseño del despliegue de Redes Inalámbricas de Sensores y Actuadores?

El **objeto de estudio** se centra en las herramientas para la visualización y edición de entornos tridimensionales.

El **campo de acción** está encaminado a los visores para la visualización y edición de entornos tridimensionales de interiores.

Con el fin de dar solución a la problemática antes mencionada se define como **objetivo general** desarrollar un visor tridimensional de entornos interiores para el diseño del despliegue de Redes Inalámbricas de Sensores y Actuadores, del cual derivan los siguientes **objetivos específicos**.

- Analizar el estado del arte, la definición de requerimientos, metodologías y tecnologías.
- Visualizar modelos tridimensionales de entornos interiores generados por herramientas de Modelado de Información para la Edificación.
- Permitir la edición de los modelos tridimensionales importados al visor.
- Gestionar componentes de Redes Inalámbricas de Sensores y Actuadores en el modelo tridimensional.
- Validar la propuesta de solución.

Para dar cumplimiento a estos objetivos, se plantean las siguientes **tareas de la investigación**:

- Realización del análisis del estado del arte sobre visores tridimensional.
- Selección de la metodología más adecuada para desarrollar el visor.
- Selección de la tecnología a ser usada en la implementación del visor.



- Estudio de los modelos tridimensionales de entornos interiores generados por herramientas de Modelado de Información para la Edificación y seleccionar los que serán importados por el visor.
- Importación de los modelos tridimensionales seleccionados.
- Visualización de los modelos tridimensionales importados.
- Modificación de propiedades de los objetos que componen el modelo tridimensional.
- Agregación de nuevos elementos al modelo tridimensional.
- Adición y eliminación de componentes de Redes Inalámbricas de Sensores y Actuadores en el modelo tridimensional.
- Modificación de propiedades de los componentes de Redes Inalámbricas de Sensores y Actuadores.
- Delimitación del área de sensado en el modelo tridimensional.
- Realización de las tareas establecidas por la metodología de desarrollo seleccionada para validar la propuesta de solución.

Finalmente, para guiar el desarrollo de la solución se plantea la siguiente **idea a defender**: Si se desarrolla un visor capaz de visualizar y editar modelos tridimensionales de entornos interiores generados por herramientas de Modelado de Información para la Edificación, permitirá al diseñador tener una representación virtual del entorno para el cual desea realizar el diseño del despliegue de la Red Inalámbrica de Sensores y Actuadores.

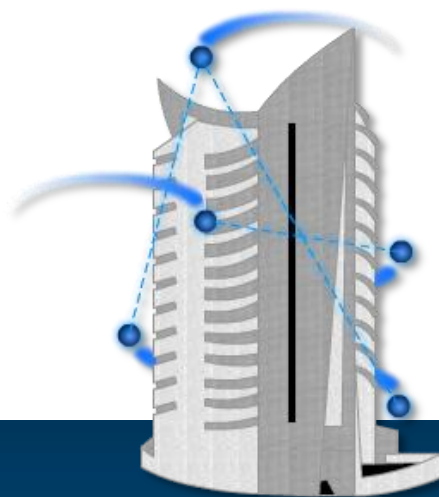
El presente trabajo está constituido por 3 capítulos, los cuales se resumen a continuación:

Capítulo 1: Fundamentación Teórica. Este capítulo trata algunos elementos teóricos necesarios para la realización de todo el trabajo en general, tales como: estado del arte, análisis de la solución existente, así como la descripción y selección de las herramientas y metodología a utilizar.

Capítulo 2: Descripción de la solución propuesta. En este capítulo se hace una descripción del entorno, se definen los requisitos funcionales y no funcionales conformando las Historias de Usuario, las cuales se organizan en iteraciones según la prioridad definida por el cliente, se estima la velocidad del proyecto y se acuerda el Plan de Entrega.



Capítulo 3: Implementación y Pruebas. En este capítulo se describe la arquitectura del sistema, detallando los diagramas UML utilizados para el desarrollo de la solución, así como las Tareas de Ingeniería y el resultado de los Casos de Prueba de Aceptación.



Andr6meda
Andr6meda

Capítulo 1

fundamentación Te6rica



1. Fundamentación Teórica

En el presente capítulo se expone el estado del arte de las soluciones existentes sobre la temática abordada y se analizan características y funcionalidades de los visores estudiados. 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 Visores IFC

Entre los formatos de archivos que destacan para la visualización 3D de herramientas de software para la construcción se encuentra el IFC, por la gran variedad de elementos constructivos que tiene en cuenta, además de sus características, los cuales están concebidos para todo el ciclo de vida del proyecto, desde su concepción hasta su construcción, describiendo de forma total la edificación virtual. Por las razones antes expuestas en el presente acápite se hace un estudio de algunos visores IFC, presentando sus características y funcionalidades. Estos visores serán usados como referencia en el desarrollo de la solución propuesta.

1.1.1 DDS-CAD Viewer

Este visor se distribuye de forma gratuita. Contiene funciones para ayudar a corregir algunos errores típicos del modelado. Las relaciones entre los objetos y espacios pueden ser cambiados, así como sus propiedades y el resultado puede ser guardado en un fichero IFC compatible con versiones futuras de IFC2x4. (4)

Entre las funciones que destacan de este visor se encuentran: zoom, órbita, caminar, gravedad, rayos X, ascensor, recorte, representación, representación de animaciones, asignación de materiales, transparencia, perspectiva, navegar por la estructura y propiedades de los objetos; detección de colisiones, puede abrir múltiples modelos en una misma vista, cambiar relaciones de objetos, agregar y editar propiedades, selecciones múltiples, tipos e instancias de objetos. Soporta formatos de archivo View IFC, IFCZip, gbXML, DWG, DXF y exporta IFC, IFCZip, DWG, DXF, DWS y 3DS. Soporta JavaScript para poder analizar y obtener información de los modelos IFC, también posee una interfaz .NET y COM para escribir pequeños complementos. (4)

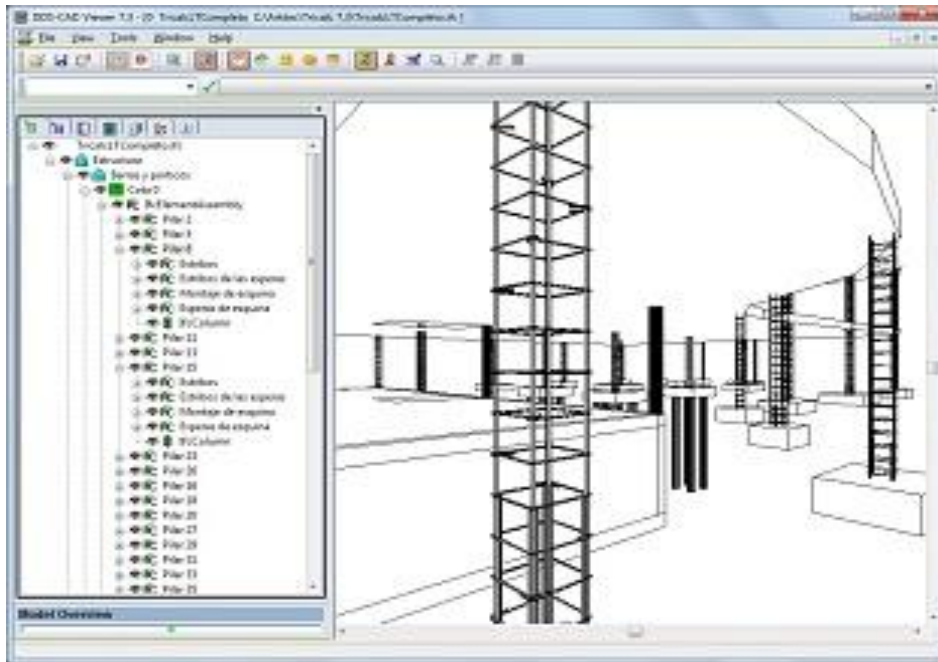


Figura 1. Visualización de un modelo IFC en la herramienta DDS-CAD Viewer

1.1.2 Solibri Model Viewer

Es un software libre desarrollado para visualizar archivos IFC en plataformas como Windows y Macintosh. Este producto puede hacer un análisis basado en normas de la calidad de sus archivos BIM y diseños. Los modelos de análisis de estos resultados y las presentaciones pueden ser compartidos con todo el equipo de diseño, el propietario y los futuros usuarios del edificio. (5)

Algunas de sus características principales son:

- Se ejecuta en Windows, Mac OS X y algunas distribuciones Linux.
- Importa archivos IFC 2.0, IFC 2x, IFC 2x2 e IFC 2x3.
- Importa archivos Solibri Model Checker, incluidos los resultados y presentaciones.

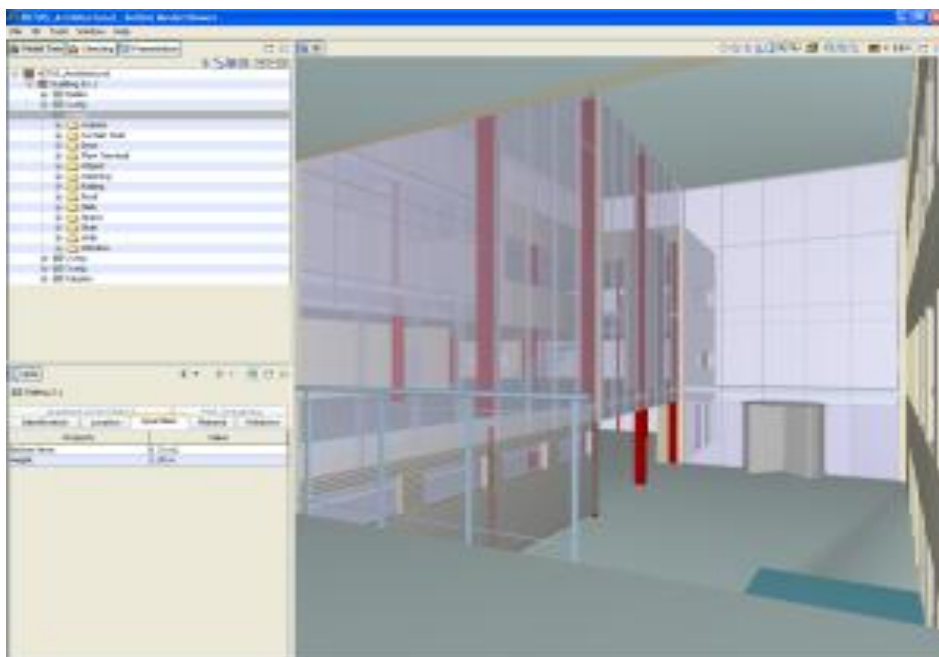


Figura 2. Visualización de un modelo IFC en la herramienta Solibri Model Viewer

1.1.3 Tekla BIM sight

Es un software BIM gratis que se encuentra disponible para instalar desde Internet. Es una aplicación de fácil acceso que presenta el proyecto completo, incluyendo toda la información necesaria para las distintas disciplinas implicadas en la construcción. (6)

Tekla BIM sight, puede importar modelos IFC prácticamente desde cualquier aplicación BIM. Los usuarios pueden realizar controles visuales a los problemas de diseño y factibilidad de construcción, automatizar la detección de choques y marcar el modelo multi-disciplinario con las notas y líneas rojas. (6)

Funcionalidades que se destacan:

- Combinar múltiples modelos y formatos de fichero de una gran variedad de programas BIM dentro de un proyecto.
- Compartir la información de construcción para la coordinación entre distintos sectores y proveedores.
- Identificar y comunicar las zonas problemáticas: verificar choques, gestionar cambios, aprobar comentarios y propuestas, y asignar trabajos en 3D guardando un historial de distintas vistas y descripciones localizadas en el modelo.
- Medir distancias directamente en el modelo para verificar los requerimientos en el diseño y tolerancias constructivas.

Visor tridimensional de entornos interiores para la herramienta Andrómeda

- Controlar la visualización y la transparencia de distintos tipos de partes en el modelo para facilitar la comprensión en zonas complejas y congestionadas del proyecto.
- Consultar propiedades tales como el perfil, la calidad del material, la longitud y el peso de las partes.

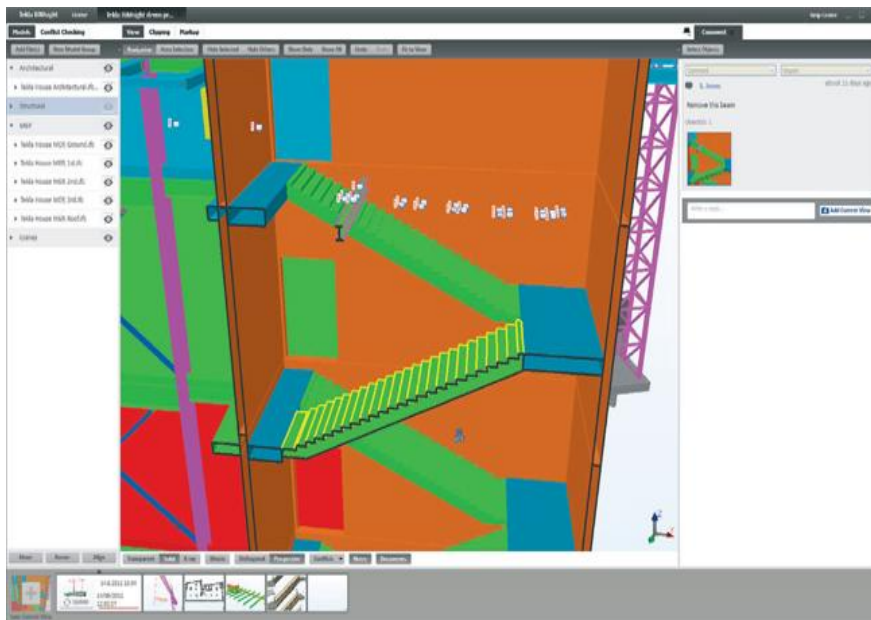


Figura 3. Visualización de un modelo IFC en la herramienta Tekla BIM sight

1.1.4 Nemetschek IFC Viewer

Permite la visualización de modelos IFC independientemente del software donde fueron creados los ficheros. Es una herramienta ideal para la integración de todos los involucrados en el proceso de planificación, incluidos los clientes, de una manera sencilla e intuitiva. (7)

Garantiza la visualización interactiva y la investigación de la estructura de la construcción de los modelos IFC. El edificio, plano 2D, modelos 3D y las propiedades geométricas y alfanuméricas de la construcción de secciones o elementos individuales de construcción se pueden ver intuitivamente. El modelo de datos se puede presentar de las siguientes formas: (7)

- La vista en 3D del modelo completo y las partes seleccionadas de construcción.
- La sección transversal de cada historia.
- La jerarquía de árbol de la estructura del edificio en 3D.

Visor tridimensional de entornos interiores para la herramienta Andrómeda

- Hace una lista de elementos de construcción (pared, techo, ventana, etc.)
- Lista de capas.

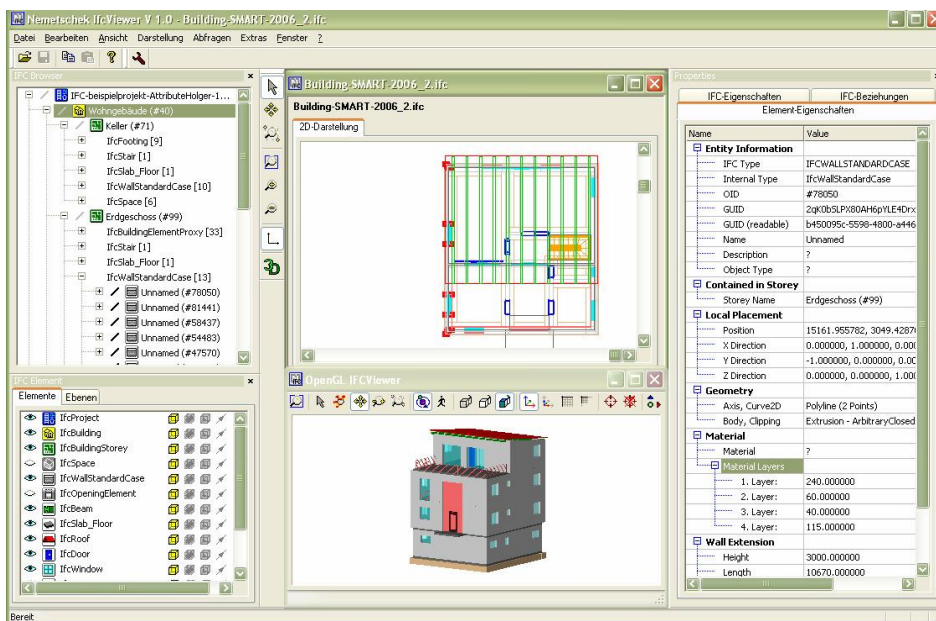


Figura 4. Visualización de un modelo IFC en la herramienta Nemestchek IFC Viewer

1.2 Lenguaje de programación

Para la selección del lenguaje de programación de la herramienta no solo se tienen en cuenta los requisitos relacionados con la visualización 3D del entorno, debido a la complejidad de los algoritmos que serán usados para generar la propuesta de despliegue de la red, es necesaria la utilización de técnicas de inteligencia artificial. Basado en estas características el equipo de arquitectos de la herramienta definió la utilización de Java como lenguaje de implementación.

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

Es un lenguaje robusto, las aplicaciones creadas con este lenguaje es poco probable que posean errores, principalmente porque la gestión de memoria y punteros es realizada por el propio lenguaje y no por el programador, además posee estructuras para la detección de excepciones y permite obligar al programador a escribir código

fiable mediante la declaraci6n de excepciones posibles para una determinada clase reutilizable. (8)

Java incorpora una t6cnica llamada *Multi-Hilo* para permitir la ejecuci6n de varias tareas concurrentes dentro de un mismo programa, lo cual mejora en gran medida la capacidad de respuesta de las aplicaciones.

Las aplicaciones desarrolladas con Java pueden ser ejecutadas indistintamente en muchos equipos, ya que son independientes de la arquitectura, es un lenguaje de desarrollo p6blico y se puede tener acceso gratis al conjunto de herramientas de desarrollo de aplicaciones de Java (Java Development's Kit, JDK seg6n sus siglas en Ingl6s).

Una característica importante que se debe tener en cuenta para el desarrollo de la herramienta propuesta, es el trabajo con modelos tridimensionales, para el cual Java presenta gran variedad de librerías libres con buena potencia gráfica como: *JOGL*, *JGL*, *Java3D*, las cuales ser6n abordadas m6s adelante. Adem6s, se caracteriza por ser un lenguaje muy utilizado en la implementaci6n de aplicaciones que incorporen t6cnicas de inteligencia artificial. Otra de las características que hace de Java el lenguaje de programaci6n m6s usado en los 6ltimos a6os es la abundante documentaci6n que aparece disponible en la red.

Sin embargo, resulta importante se6alar que los programas realizados en Java suelen ser un poco lentos respecto a programas desarrollados con C++. Esto es debido a que las aplicaciones escritas en Java son interpretadas por una m6quina virtual, adem6s para el manejo de c6digo de bajo nivel, se deben utilizar m6todos nativos del sistema, disminuyendo de esta forma su portabilidad.

1.3 Librerías 3D para el desarrollo

Para la representaci6n y el modelado en tercera dimensi6n, Java posee potentes librerías gráficas que facilitan el trabajo y la interacci6n con los objetos 3D, dentro de las principales librerías se encuentran: *JOGL*, *JGL* y *Java3D*. Las cu6les ser6n objeto de estudio para seleccionar la m6s adecuada en el desarrollo del visor.



1.3.1 JOGL

JOGL (Java/OpenGL) encapsula dentro de unas pocas clases Java toda la potencia de la biblioteca OpenGL. De hecho, la API JOGL es básicamente una envoltura JNI alrededor de la biblioteca nativa escrita en C para cada plataforma gráfica. Su utilización no implica usar únicamente ficheros .jar sino que también es necesario hacer uso de bibliotecas de enlace dinámico propias de cada plataforma.

Uno de los problemas que plantea el uso de esta API es que no es realmente orientada a objetos, pues la biblioteca OpenGL no ha sido remodelada y adaptada a Java, sino sólo es envuelta por una capa JNI, lo que obliga al programador experto en Java a retroceder en el tiempo y volver a la época de la programación modular o estructurada.

1.3.2 JGL

JGL es una biblioteca de gráficos 3D desarrollada para Java con una API similar a la de OpenGL que puede ser ejecutada en la plataforma Java 2 y versiones anteriores. Esta, en su versión actual, no soporta todas las funciones de OpenGL. Cuenta con un cargador de ficheros OBJ¹ que se distribuye bajo licencia MIT/X11². Una de las principales desventajas es que no soporta Programación Orientada a Objetos (POO), lo cual obliga al programador a utilizar programación estructurada. JGL es software libre bajo los términos de la licencia GNU³.

A diferencia de Java 3D u otras bibliotecas, JGL no tiene que ser pre-instalado, todos los códigos necesarios se descargan en tiempo de ejecución. JGL está escrito en Java puro, por lo que es realmente independiente de la plataforma y podría ser ejecutado en cualquier máquina compatible con Java.

1.3.3 Java3D

Java3D es un árbol de clases escritas en Java que sirven como interfaz para sistemas de renderizado de gráficos 3D. Trabaja con constructores de alto nivel para crear y manipular objetos geométricos en 3D. Los detalles de renderizado son manejados

¹ Formato 3D creado por Wavefront (www.wavefrontac.com)

² www.opensource.org/licenses/mit-license.php

³ Generic Public License (<http://www.gnu.org/licenses/gpl.html>)



automáticamente aprovechando los hilos de Java, lo que hace posible hacerlo en paralelo.

Proporciona un paradigma de POO de alto nivel para permitir a los desarrolladores generar sofisticadas aplicaciones y *applets* de forma rápida.

Proporciona soporte a cargadores en tiempo de ejecución. Esto permite que Java 3D se adapte a un gran número de formatos de ficheros, como pueden ser formatos específicos de distintos fabricantes de BIM, formatos de intercambio o VRML 1.0 (Virtual Reality Modelling Language, Lenguaje de Modelado de Realidad Virtual en Español) y VRML 2.0.

Los códigos fuente de los proyectos *j3d-core-utils* y *j3d-examples* están bajo licencia Berkeley Software Distribution (BSD⁴).

El código fuente de los proyectos *j3d-core* y *vecmath* están bajo licencia General Public License (GPL).

1.3.4 Selección de la librería

Teniendo en cuenta las características de las librerías gráficas para el modelado 3D, se demuestra que la más idónea para el propósito del visor sería Java3D dadas las ventajas que esta presenta sobre las librerías antes mencionadas, por ejemplo: la librería JOGL necesita además de los ficheros .jar las librerías de enlaces dinámicos dependiendo de la plataforma donde se ejecute la aplicación, no siendo esto necesario si se utiliza Java3D, ya que no usa librerías de enlace dinámico. Tanto JOGL como JavaGL no están orientadas a objetos lo cual atrasa considerablemente el desarrollo de la aplicación, en cambio Java3D permite a los desarrolladores generar aplicaciones de forma rápida empleando las facilidades que brinda la POO. Además del soporte a los cargadores que permite que se adapte a un gran número de formatos de ficheros.

1.4 Librería para cargar ficheros IFC

Se utilizará para el manejo de ficheros IFC la librería OpenIFCTools⁵. Es un proyecto de código abierto y multiplataforma que utiliza tecnología Java, cuyo propósito es crear

⁴ www.opensource.org/licenses/bsd-license.php

⁵ www.openifctools.com



cuatro m6dulos (4D Schedule Assistant, IFC Open Java Tool Box, BM Boolean Modeler y J3D Java loader for Java3D).

El uso de esta librería permite el desarrollo orientado a objetos. Cada clase del fichero IFC est1 representada por una clase Java con sus atributos correspondientes. En este sentido, es posible crear una instancia y manejar cualquier objeto usando las clases de Java. Es posible leer y escribir ficheros en formato IFC haciendo uso de m6todos sencillos, as1 como registrar *listeners* en el modelo para informar los cambios que ocurren en el mismo, lo cual puede aplicarse a cualquier objeto utilizando los m6todos de acceso en lugar de acceder directamente a los atributos (todos los atributos son p6blicos, lo que permite el acceso a los datos. (9)

En la versi6n disponible para descargar, no se encuentra el m6dulo destinado a la visualizaci6n de los ficheros IFC. Es por ello que la representaci6n visual de los elementos del modelo debe ser implementada por el equipo de desarrollo, haciendo uso de la librería gr1fica Java3D. Solo se utilizar1 la librería OpenIFCTools para importar ficheros IFC a clases java, ya que este es 6nico m6dulo del proyecto que est1 implementado en su totalidad.

1.5 Entorno de desarrollo

Para el desarrollo de la soluci6n el equipo de arquitectura seleccion6 como IDE el NetBeans 7.1⁶. Es libre y de c6digo abierto. Posee todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones m6viles con la plataforma Java, adem1s posee integridad con lenguajes como C/C ++, PHP y JavaScript. (10)

Es conocido por la integridad con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de c6digo, coloreo de sintaxis, permite la utilizaci6n y edici6n de los componentes visuales de forma sencilla. Incluye tambi6n el control de versiones, lo cual representa una ventaja ya que permite administrar las diferentes versiones del c6digo fuente (para el control de versiones del visor se emplea Subversion SVN).

⁶ www.netbeans.org



1.6 Metodología de desarrollo de software

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

A continuación se describen las características más relevantes de esta metodología:

Cortas iteraciones: La entrega continua de resultados compromete a ambas partes en la evolución del proyecto, además de influir directamente de forma positiva en la calidad del producto final. En cada iteración, se obtiene un producto listo para entregar y que tiene valor para el cliente.

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

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

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

1.7 Herramienta para el modelado

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML (VP-UML), una de las líderes del mercado de las llamadas herramientas CASE (Computer Assisted Software Engineering, Ingeniería de Software Asistida por Computadora).

VP-UML soporta los principales estándares de la industria tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo

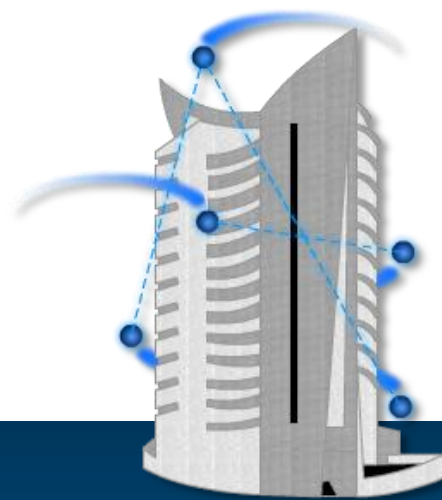


necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos. (11)

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

1.8 Conclusiones

En el presente capítulo se realizó un estudio sobre el estado del arte de las soluciones existentes, así como las herramientas y la metodología a emplear. Siguiendo la línea base que define la arquitectura, se determinó usar como lenguaje de programación Java, haciendo uso de la librería Java 3D para el modelado tridimensional, utilizando como IDE NetBeans, destacándose por su robustez e integridad con el lenguaje. Como metodología de desarrollo XP, siendo la más adecuada teniendo en cuenta las características del equipo de trabajo y el corto plazo de entrega del producto.



Andrómeda
Andrómeda

Capítulo 2

Descripción de la solución propuesta



2. Descripción de la solución propuesta

El objetivo principal de este capítulo es describir las funcionalidades del sistema, es decir, los aspectos funcionales y 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 dos primeras etapas (Exploración y Planificación) con el fin de conocer el alcance, estimar los tiempos de entrega de cada iteración y los artefactos que se generan.

2.1 Usuarios del sistema

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

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

2.2 Lista de reservas del producto

Según define la metodología XP la **lista de reservas del producto** está compuesta por los requisitos funcionales con los que debe contar la aplicación para satisfacer con eficiencia las necesidades requeridas por el cliente. A continuación se muestra el listado de los mismos.

R1. Importar modelo IFC

R2. Visualizar modelo IFC

R2.1. Mostrar elementos del modelo en una estructura jerárquica

R2.2. Mostrar la vista 3D del modelo

R2.3. Cambiar tipo de visualizaci3n

R3. Permitir la selecci3n de los elementos del modelo

R4. Visualizar las propiedades de los elementos del modelo

R5. Opciones de visualizaci3n del modelo 3D

R5.1. Permitir la rotaci3n de la vista 3D

R5.2. Permitir acercar y alejar la vista 3D

R5.3. Mover la vista 3D

R6. Permitir el desplazamiento en el modelo 3D

R7. Gestionar elemento no estructural

R7.1. Adicionar elemento no estructural

R7.2. Eliminar elemento no estructural

R7.3. Editar propiedades del elemento no estructural

R8. Gestionar 1rea de sensado

R8.1. Adicionar 1rea de sensado

R8.2. Eliminar 1rea de sensado

R8.3. Editar propiedades del 1rea de sensado

R9. Gestionar dispositivo de sensado

R9.1. Adicionar dispositivo de sensado

R9.2. Eliminar dispositivo de sensado

R9.3. Editar propiedades del dispositivo de sensado

R10. Exportar imagen bidimensional del modelo.



R11. Gestionar proyecto

R11.1. Crear nuevo proyecto

R11.2. Abrir proyecto

R11.3. Guardar proyecto

2.3 Aspectos no funcionales del sistema

Los requisitos no funcionales del sistema son aspectos a tener en cuenta para hacer que el software sea más agradable, usable, rápido y seguro. Se desglosan en distintos aspectos como:

Software

- El sistema debe ser multiplataforma.

Apariencia o interfaz externa

- La interfaz del visor debe estar compuesta por cuatro ventanas principales. En el centro el área de trabajo, en la parte superior la barra de herramientas, a la izquierda la ventana de navegación y por último a la derecha la ventana de propiedades.
- Debe poseer un área superior al 70% de la pantalla destinada al visor.

Usabilidad

- Las barras de herramientas deben estar organizadas por categorías, así como la ventana de propiedades y librería.
- El visor debe contener elementos representativos (íconos) que identifiquen cada una de las herramientas del visor 3D.
- Dentro del visor 3D el cursor debe poseer un diseño que represente en todo momento la herramienta que ha sido seleccionada.
- Se debe permitir la interacción con los dispositivos que fueron añadidos por la herramienta al modelo IFC importado.

Seguridad

- Confirmación sobre las acciones de eliminación.



Rendimiento

- La herramienta deberá ser capaz de importar los modelos IFC en un lapso de tiempo inferior a 1 minuto.
- Las diferentes herramientas del visor 3D (rotar, mover, etc.) deben ser ejecutadas en tiempos inferiores a 1 segundo.

2.4 Exploración

Como punto de partida en el ciclo de vida de un proyecto, XP propone la Exploración como la primera etapa donde los clientes plantean la propuesta al equipo de trabajo. Este proceso se desarrolla mediante reuniones y tormentas de ideas obteniendo como resultado, a grandes rasgos, las Historias de Usuario más significativas para la primera entrega del producto. De igual manera el equipo de desarrollo en este período se familiariza con las herramientas y tecnologías que serán utilizadas en el proyecto y se investigan las posibilidades de la arquitectura del sistema obteniendo como resultado un prototipo.

2.4.1 Historias de Usuario

Las Historias de Usuario (HU) son una breve descripción de las funcionalidades del sistema. Estas son descritas por el cliente sin uso del lenguaje técnico y representan cada una de las características principales del proyecto. Las HU deben organizarse de forma tal que prioricen las necesidades del cliente, permitiendo al equipo de trabajo identificar las más significativas a desarrollar durante el proceso de implementación de la solución. Se emplean además para estimar el tiempo de vida del proyecto. A continuación se muestran las descripciones de cada una de las HU propuestas por el cliente.

Tabla 1. HU 1: Importar modelo IFC

Historia de Usuario	
No.: 1	Nombre: Importar modelo IFC
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 1 semana	Iteración Asignada: 1
Descripción: El usuario es capaz de seleccionar el modelo IFC e importarlo al visor.	
Información adicional (Observaciones): Da cumplimiento al requisito R1 "Importar modelo IFC".	

Tabla 2. HU 2: Visualizar modelo IFC

Historia de Usuario	
No.: 2	Nombre: Visualizar modelo IFC
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 3 semanas	Iteración Asignada: 1
Descripción: El sistema debe mostrar la estructura jerárquica del modelo en forma de árbol y mostrar la vista tridimensional en el área de trabajo.	
Información adicional (Observaciones): Da cumplimiento al requisito R2 "Visualizar modelo IFC".	

Tabla 3. HU 3: Permitir la selección de los elementos del modelo

Historia de Usuario	
No.: 3	Nombre: Permitir la selección de los elementos del modelo
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 1 semana	Iteración Asignada: 1
Descripción: El usuario debe ser capaz de seleccionar los elementos del modelo, ya sea desde la vista tridimensional o desde la estructura jerárquica.	
Información adicional (Observaciones): Da cumplimiento al requisito R3 "Permitir la selección de los elementos del modelo".	

Tabla 4. HU 4: Visualizar propiedades de elementos del modelo

Historia de Usuario	
No.: 4	Nombre: Visualizar propiedades de elementos del modelo
Usuario: Todos	
Prioridad en el Negocio: Media.	Nivel de Complejidad: Media.
Estimación: 1 semana	Iteración Asignada: 1
Descripción: Una vez seleccionado un elemento del modelo el sistema debe mostrar en una ventana sus propiedades.	
Información adicional (Observaciones): Da cumplimiento al requisito R4 "Visualizar propiedades de elementos del modelo".	

Tabla 5. HU 5: Opciones de visualización del modelo3D

Historia de Usuario	
No.: 5	Nombre: Opciones de visualización del modelo 3D
Usuario: Todos	
Prioridad en el Negocio: Media	Nivel de Complejidad: Alta
Estimación: 2 semanas	Iteración Asignada: 2
Descripción: El sistema permite la rotación de la vista tridimensional, así como acercar, alejar y desplazar; ofreciendo un panorama más detallado del modelo.	
Información adicional (Observaciones): Da cumplimiento al requisito R5 "Opciones de visualización del modelo 3D".	

Tabla 6. HU 6: Permitir el desplazamiento en el modelo 3D

Historia de Usuario	
No.: 6	Nombre: Permitir el desplazamiento en el modelo 3D
Usuario: Todos	
Prioridad en el Negocio: Media	Nivel de Complejidad: Alta
Estimación: 1 semana	Iteración Asignada: 2
Descripción: El usuario puede desplazarse por el modelo haciendo uso de las teclas de navegación.	
Información adicional (Observaciones): Da cumplimiento al requisito R6 "Permitir el desplazamiento en el modelo 3D".	

Tabla 7. HU 7: Gestionar elemento no estructural

Historia de Usuario	
No.: 7	Nombre: Gestionar elemento no estructural
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 1 semana	Iteración Asignada: 3
Descripción: El sistema debe permitir adicionar, eliminar y editar elementos no estructurales en el modelo tridimensional.	
Información adicional (Observaciones): Da cumplimiento al requisito R7 "Gestionar elemento no estructural".	

Tabla 8. HU 8: Gestionar área de sensado

Historia de Usuario	
No.: 8	Nombre: Gestionar área de sensado
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 1 semana	Iteración Asignada: 3
Descripción: El sistema debe permitir adicionar, eliminar y editar áreas de sensado en el modelo tridimensional.	
Información adicional (Observaciones): Da cumplimiento al requisito R8 "Gestionar área de sensado".	

Tabla 9. HU 9: Gestionar dispositivo de sensado

Historia de Usuario	
No.: 9	Nombre: Gestionar dispositivo de sensado
Usuario: Todos	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 2 semanas	Iteración Asignada: 3
Descripción: El sistema debe permitir adicionar, eliminar y editar dispositivos de sensado en el modelo tridimensional.	
Información adicional (Observaciones): Da cumplimiento al requisito R9 "Gestionar dispositivo de sensado".	

Tabla 10. HU 10: Exportar imagen bidimensional del modelo

Historia de Usuario	
No.: 10	Nombre: Exportar imagen bidimensional del modelo
Usuario: Todos	
Prioridad en el Negocio: Baja.	Nivel de Complejidad: Media
Estimaci3n: 1 semana	Iteraci3n Asignada: 4
Descripci3n: El sistema debe permitir exportar la imagen bidimensional del modelo desde la vista 3rea.	
Informaci3n adicional (Observaciones): Da cumplimiento al requisito R10 "Exportar imagen bidimensional del modelo".	

Tabla 11. HU 11: Gestionar proyecto

Historia de Usuario	
No.: 11	Nombre: Gestionar proyecto
Usuario: Todos	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimaci3n: 2 semanas	Iteraci3n Asignada: 4
Descripci3n: El usuario es capaz de realizar operaciones b3sicas, como abrir o cargar un proyecto existente, crear uno nuevo y salvar el actual.	
Informaci3n adicional (Observaciones): Da cumplimiento al requisito R11 "Gestionar proyecto".	

2.5 Planificaci3n

La fase de Planificaci3n se realiza en pocos d3as, no siendo menos importante, ya que su objetivo principal es priorizar las HU. Los programadores estiman el esfuerzo de cada una de ellas, se define el cronograma y se acuerda el alcance de la entrega, el cual debe incluir varias iteraciones. El cliente prioriza las HU seleccionando cuales se realizar3n en cada iteraci3n, logrando en la primera iteraci3n un sistema con la arquitectura de proyecto. Otro aspecto a tener en cuenta durante esta fase es la velocidad del proyecto, la cual se estima utilizando el tiempo empleado en el desarrollo de las iteraciones terminadas. Esta medida debe reevaluarse una vez concluida 3 o 4 iteraciones y en caso de no cumplir el tiempo estimado, debe ser negociando con el cliente un nuevo Plan de Entregas.

2.5.1 Iteraciones

XP enfatiza en el carácter iterativo e incremental del desarrollo, la fase de Planificación es la principal en el ciclo de vida de un proyecto. Las iteraciones agrupan un conjunto de HU a implementar en un período de tiempo, generando al final de cada una un entregable funcional. Son relativamente cortas y no deben exceder las 4 semanas en su desarrollo. La entrega con rapidez de módulos al cliente aumenta la retroalimentación y resulta más provechoso en términos de calidad del producto que una entrega a largo plazo.

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

Tabla 12. Plan de Iteraciones

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo
Iteración 1	HU1. Importar modelo IFC HU2. Visualizar modelo IFC HU3. Permitir la selección de los elementos del modelo. HU4. Visualizar propiedades de elementos del modelo.	6 semanas
Iteración 2	HU5. Opciones de visualización del modelo 3D HU6. Permitir el desplazamiento en el modelo 3D	3 semanas
Iteración 3	HU7. Gestionar elemento no estructural HU8. Gestionar área de sensado HU9. Gestionar dispositivo de sensado	4 semanas
Iteración 4	HU10. Exportar imagen bidimensional del modelo HU11. Gestionar proyecto	2 semanas

2.5.2 Plan de Entregas

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

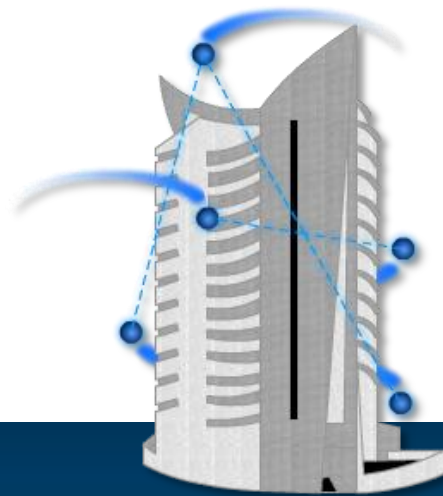
El cronograma de entregas es realizado en dependencia de las estimaciones de tiempos de desarrollo realizadas por los programadores. La estimación es uno de los temas más complicados del desarrollo de un proyecto de software, es por ello que se recomienda luego de algunas iteraciones realizar nuevamente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

Tabla 13. Plan de Entrega

Entregable	Final 1ra iteración 2 ^{da} semana de Marzo de 2012	Final 3ra iteración 5 ^{ta} Semana de Abril de 2012	Final 4ta iteración 2 ^{da} semana de Mayo de 2012
Visor	versión 0.1	versión 0.2	versión 1.0

2.6 Conclusiones

En el presente capítulo se tratan temas referentes a la concepción del producto teniendo en cuenta los requisitos que el mismo debe cumplir, ya sean funcionales o no. Luego de este análisis, se realizó la documentación referente a la primera etapa del ciclo de vida de proyecto siguiendo la metodología seleccionada en la cual se detallan los artefactos como: Historias de Usuario, el Plan de Iteraciones y el Plan de Entregas.



Andrómeda
Andrómeda

Capítulo 3

Implementación y Pruebas



3. Implementación y Pruebas

Debido a que XP no especifica concretamente cuáles artefactos deben ser utilizados en la implementación de la solución, el equipo de trabajo debe tomar la decisión de cuáles diagramas de UML cree necesarios emplear para la documentación y desarrollo de la aplicación. En este capítulo se presenta la descripción de la arquitectura, los diagramas UML utilizados, las Tareas de Ingeniería y el proceso de Pruebas.

3.1 Descripción de la arquitectura

Para el desarrollo de la solución en la arquitectura de la herramienta se seleccionó la plataforma de NetBeans (NetBeans Platform⁷), debido a que su uso permite crear aplicaciones modulares. Haciendo uso del IDE NetBeans, el cual está desarrollado sobre esta plataforma, se facilita considerablemente el desarrollo del tipo de aplicaciones como la que se desea implementar, favoreciendo notablemente las soluciones creadas con dicha tecnología, ya que no se necesita modificar directamente el núcleo de la aplicación para agregar nuevos módulos, lo cual la hace más flexible.

En las Figuras 6, 7 y 8 se muestran los Diagramas de Componentes de los módulos relacionados con el visor.

⁷ netbeans.org/kb/trails/platform.html

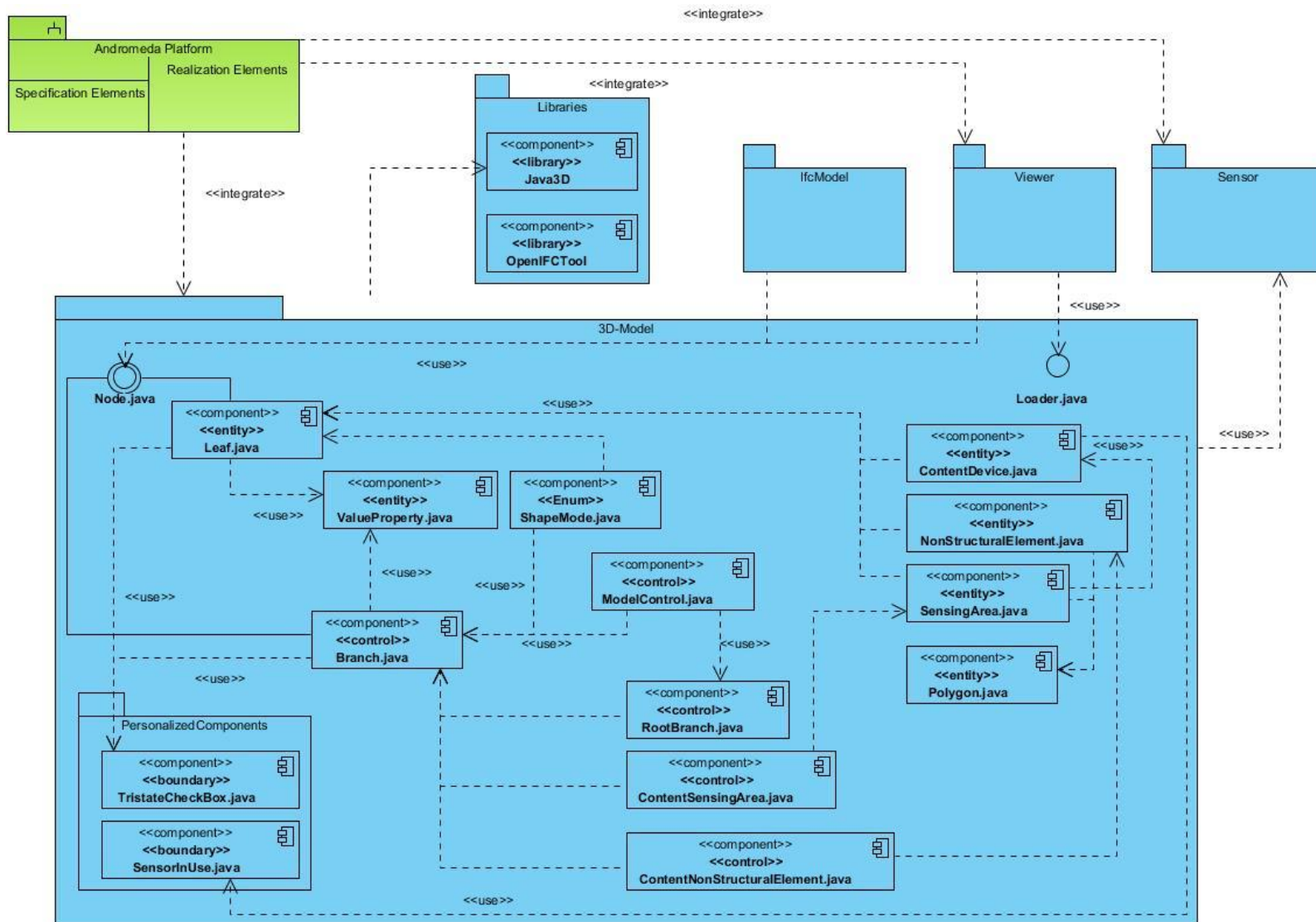


Figura 5. Diagrama de Componentes del módulo 3D-Model

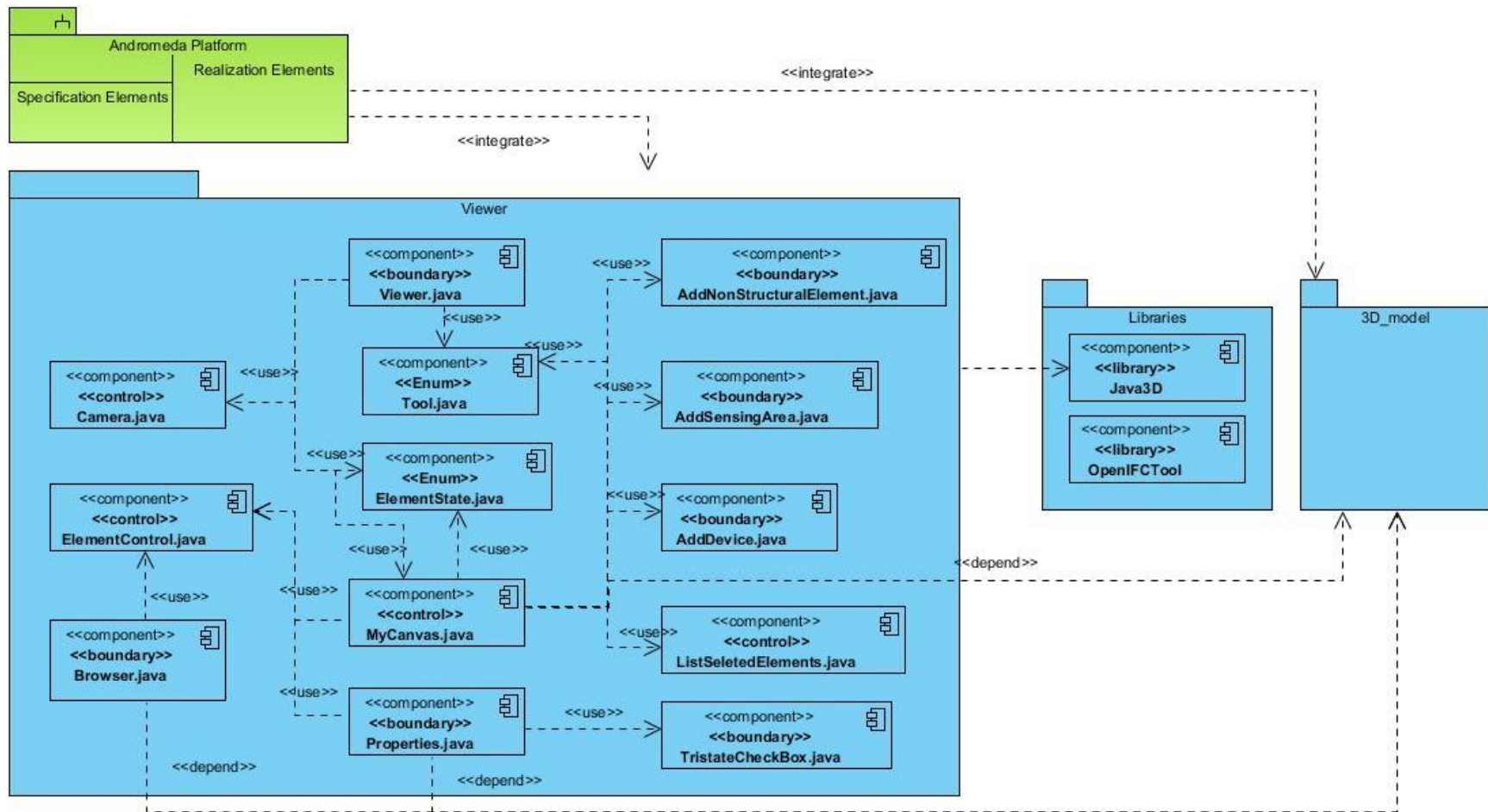


Figura 6. Diagrama de Componentes del módulo Viewer

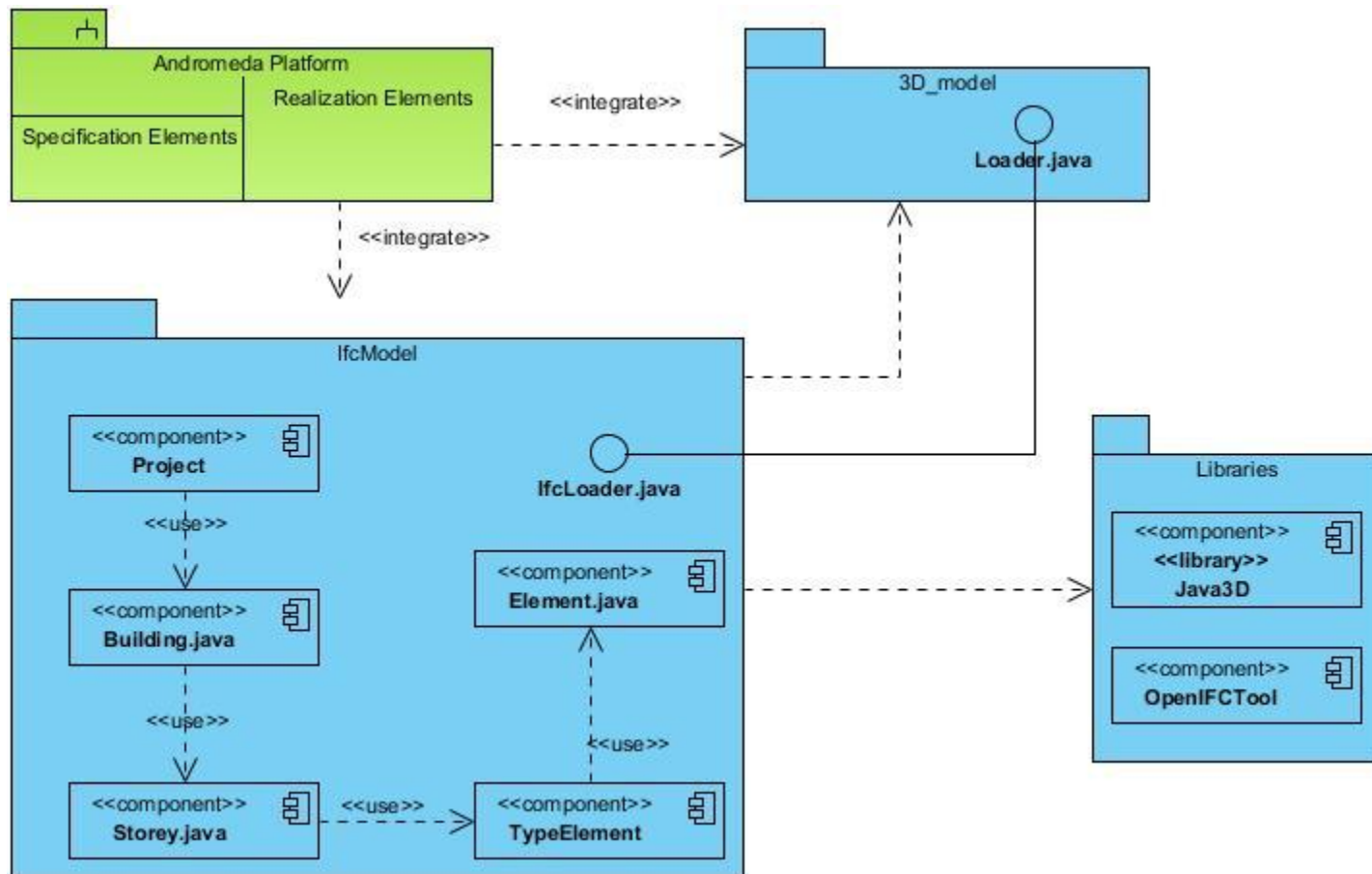


Figura 7. Diagrama de Componentes del módulo IfcModel



3.1.1 Plataforma Andr6meda

Seg6n se defini6 en la arquitectura: “Andr6meda es el subsistema que hace referencia al n6cleo del sistema final a desarrollar. El mismo es una instancia de la Plataforma de NetBeans, por lo cual est6 provisto del conjunto de APIs y clases que provee la plataforma. A su vez es el encargado de la gesti6n de carga de otros m6dulos. Inicia el men6 de la aplicaci6n y carga la biblioteca de sensores a ser usados en el proyecto.”

La Plataforma de NetBeans permite las relaciones entre m6dulos estableciendo dependencias entre ellos. Adem6s posee un mecanismo definido por la plataforma llamado *Lookup*, el cual posibilita que cada m6dulo intercambie informaci6n con los restantes.

3.1.2 M6dulos del sistema

La aplicaci6n del visor est6 compuesta por tres m6dulos. Los m6dulos 3D-Model y Viewer se encargan de la gesti6n y visualizaci6n del modelo 3D respectivamente. Por 6ltimo, el *IfcModel* gestiona espec6ficamente los ficheros IFC. A continuaci6n se describen cada uno de los m6dulos:

- **3D-Model:** M6dulo encargado de gestionar todo lo referente al modelo 3D. Est6 conformado por un conjunto de clases organizadas de forma jer6rquica, que son usadas para la visualizaci6n en el m6dulo Viewer.
- **Viewer:** M6dulo encargado de visualizar el modelo importado tanto en el visor tridimensional como en el 6rbol jer6rquico. Cada formato de archivo especifica su presentaci6n y organizaci6n de forma diferente.
- **IfcModel:** M6dulo encargado de especificar la presentaci6n y organizaci6n de los modelos IFC haciendo uso de las clases del m6dulo **3D-Model**.

3.1.3 Ventanas del Visor

A continuaci6n se muestra la interfaz de usuario del visor:

Visor tridimensional de entornos interiores para la herramienta Andrómeda

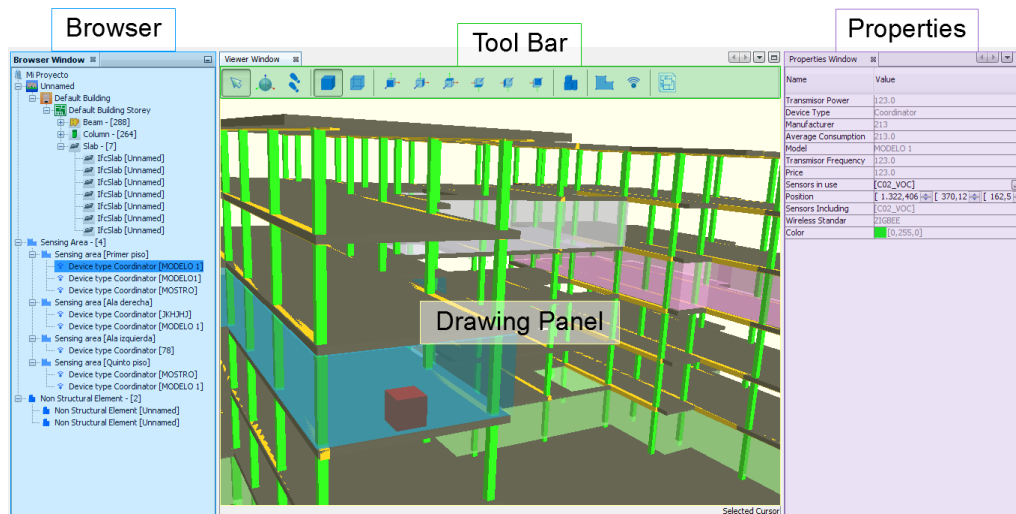


Figura 8. Interfaz de usuario del visor

- **Tool Bar:** Contiene todas las herramientas para operar con el visor tridimensional, se representan en forma de botones y divididas en seis grupos según su funcionalidad: Navegación, visualización, vistas, elementos no estructurales, áreas y dispositivos de sensado, así como exportar imagen bidimensional.
- **Drawing Panel:** Como su nombre indica, es el panel donde se dibuja el modelo tridimensional.
- **Properties:** Es el encargado de visualizar y editar las propiedades de los elementos seleccionados del modelo.
- **Browser:** Muestra de forma jerárquica organizados según su tipo, la estructura del proyecto.

3.1.4 Paquetes del sistema

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

La estructura de paquetes en Java se organiza de la siguiente manera: `paquetepincipal.subpaquete`, por lo que quedan organizados los paquetes que conforman el visor como: `cu.andromeda.viewer`, `cu.andromeda.model3D` y `cu.andromeda.ifcModel`, para más detalles ver la Figura 10.

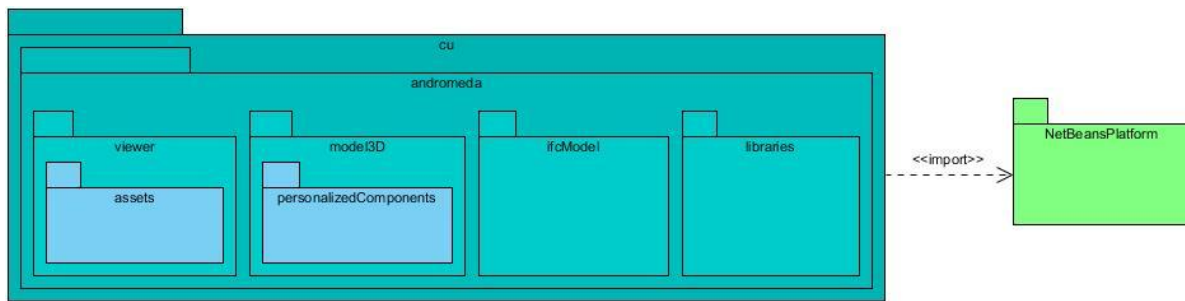


Figura 9. Diagrama de paquetes del visor

A continuación se describen cada uno de los paquetes:

- **cu.andromeda.viewer:** Es el paquete que contiene todo lo referente a la visualización de los modelos.
- **cu.andromeda.model3D:** Este paquete almacena toda la información relacionada con los modelos importados al visor.
- **cu.andromeda.ifcModel:** Este paquete se encarga de especificar la forma lógica de almacenar los datos de los ficheros IFC y define la forma de mostrar los elementos, su organización y características.

3.1.5 Diagrama de clases del sistema

El Diagrama de Clases describe las relaciones entre cada una de estas. En la Figura 11 se muestra el diagrama de clases de la arquitectura del sistema que incluye las clases utilizadas por el visor.

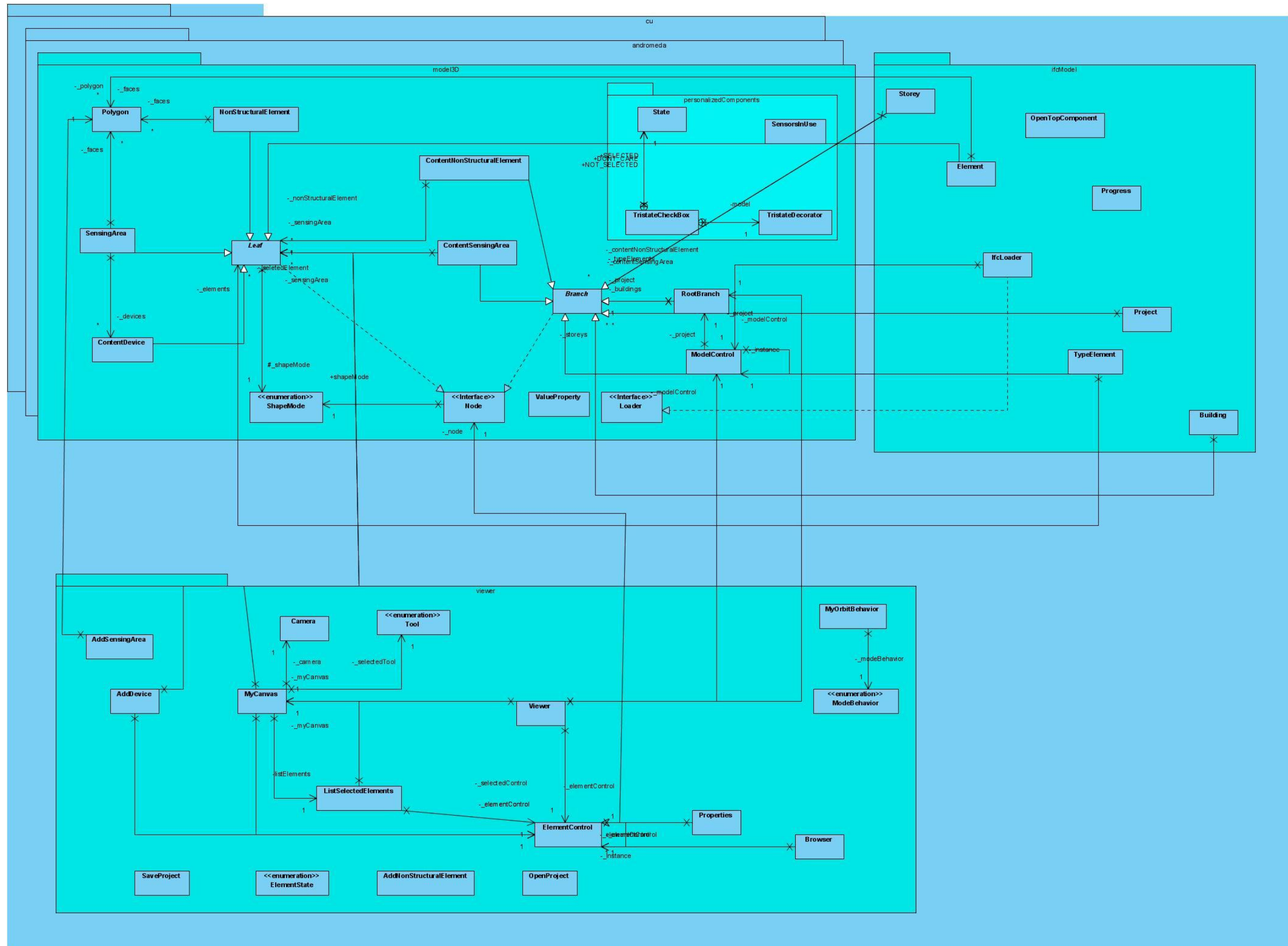


Figura 10. Diagrama con las clases empleadas por el visor



3.2 Implementación

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

3.2.1 Primera iteración

En la primera iteración se pretende desarrollar las primeras cuatro HU, con el objetivo de obtener una primera versión funcional del producto. Con el cumplimiento de estos requisitos se crean las bases para todo el desarrollo de la aplicación.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 1 “Importar modelo IFC”

1. Tarea N° 1: Estudiar librería OpenIfcTool.
2. Tarea N° 2: Crear clase Loader.

Historia de usuario N° 2 “Visualizar modelo IFC”

3. Tarea N° 3: Crear clases Project, Building, Storey y Element.
4. Tarea N° 4: Diseñar Interfaz de usuario del Browser y Viewer.
5. Tarea N° 5: Implementar el parseo de elementos IFC.
6. Tarea N° 6: Implementar la vista sólida.
7. Tarea N° 7: Implementar la vista lineal.

Historia de usuario N° 3 “Permitir la selección de los elementos del modelo”

8. Tarea N° 6: Implementar la selección de elementos del árbol.
9. Tarea N° 7: Implementar la selección de elementos del visor.

Historia de usuario N° 4 “Visualizar las propiedades de los elementos del modelo”

10. Tarea N° 8: Diseñar interfaz Propiedades.
11. Tarea N° 9: Mostrar propiedades de elementos desde el árbol.
12. Tarea N° 10: Mostrar propiedades de elementos desde el visor.

Tabla 14. Tarea 1: Estudiar librería OpenIfcTool

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Estudiar librería OpenIfcTool	
Tipo de tarea: Preparación	Estimación: 3 días
Fecha de inicio: 01 de febrero de 2012	Fecha fin: 03 de febrero de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Estudiar la librería OpenIfcTool con el objetivo de saber cómo es que almacena los datos obtenidos del fichero IFC, así cómo identificar los elementos que son necesarios para visualizar el modelo.	

Tabla 15. Tarea 2: Crear clase Loader.

Tarea	
Número de tarea: 2	Número de HU: 1
Nombre de la tarea: Crear clase Loader	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 06 de febrero de 2012	Fecha fin: 07 de febrero de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar la clase que carga el fichero IFC.	
Paquete: cu.andromeda.ifcModel	

Tabla 16. Tarea 3: Crear clases Project, Building, Storey y Element

Tarea	
Número de tarea: 3	Número de HU: 2
Nombre de la tarea: Crear clases Project, Building, Storey y Element	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 08 de febrero de 2012	Fecha fin: 08 de febrero de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar las clases que almacenan toda la estructura del modelo para que queden organizados jerárquicamente. El proyecto contiene las construcciones, las construcciones contienen los pisos y los pisos los elementos.	
Paquete: cu.andromeda.ifcModel	

Tabla 17. Tarea 4: Diseñar Interfaz de usuario del Browser y Viewer

Tarea	
Número de tarea: 4	Número de HU: 2
Nombre de la tarea: Diseñar Interfaz de usuario del Browser y Viewer	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 09 de febrero de 2012	Fecha fin: 09 de febrero de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Diseñar las interfaces de usuario del Viewer y Browser.	
Paquete: cu.andromeda.viewer	

Tabla 18. Tarea 5: Implementar el parseo de elementos IFC

Tarea	
Número de tarea: 5	Número de HU: 2
Nombre de la tarea: Implementar el parseo de elementos IFC	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 10 de febrero de 2012	Fecha fin: 14 de febrero de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Implementar los métodos necesarios de la clase Element que le permitan ser visualizada en el visor tridimensional.	

Tabla 19. Tarea 6: Implementar la vista sólida

Tarea	
Número de tarea: 6	Número de HU: 2
Nombre de la tarea: Implementar la vista sólida	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 15 de febrero de 2012	Fecha fin: 17 de febrero de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Implementar los métodos necesarios en las clases de los objetos que se van a visualizar para mostrar la vista de polígonos rellenos.	

Tabla 20. Tarea 7: Implementar la vista lineal

Tarea	
Número de tarea: 7	Número de HU: 2
Nombre de la tarea: Implementar vista lineal	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 20 de febrero de 2012	Fecha fin: 22 de febrero de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar los métodos necesarios en las clases de los objetos que se van a visualizar para mostrar la vista de polígonos no rellenos.	

Tabla 21. Tarea 8: Implementar la selección de elementos del árbol

Tarea	
Número de tarea: 8	Número de HU: 3
Nombre de la tarea: Implementar la selección de elementos del árbol	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 23 de febrero de 2012	Fecha fin: 24 de febrero de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar la selección de elementos de la vista de árbol, obteniendo todos sus datos.	

Tabla 22. Tarea 9: Implementar la selección de elementos del visor tridimensional

Tarea	
Número de tarea: 9	Número de HU: 3
Nombre de la tarea: Implementar la selección de elementos del visor tridimensional	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 27 de febrero de 2012	Fecha fin: 29 de febrero de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Implementar la selección de elementos desde la vista tridimensional, obteniendo todos sus datos.	

Tabla 23. Tarea 10: Diseñar interfaz Propiedades

Tarea	
Número de tarea: 10	Número de HU: 4
Nombre de la tarea: Diseñar interfaz Propiedades	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 01 de marzo de 2012	Fecha fin: 02 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Diseñar interfaz de usuario de la ventana de Propiedades.	
Paquete: cu.andromeda.viewer	

Tabla 24. Tarea 11: Mostrar propiedades de elementos desde el árbol

Tarea	
Número de tarea: 11	Número de HU: 4
Nombre de la tarea: Mostrar propiedades de elementos desde el árbol	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 05 de marzo de 2012	Fecha fin: 05 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Mostrar propiedades de los elementos seleccionados desde el árbol.	

Tabla 25. Tarea 12: Mostrar propiedades de elementos desde el visor

Tarea	
Número de tarea: 12	Número de HU: 4
Nombre de la tarea: Mostrar propiedades de elementos desde el visor	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 06 de marzo de 2012	Fecha fin: 07 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Mostrar propiedades de los elementos seleccionados desde el visor.	



3.2.2 Segunda Iteraci3n

En la segunda iteraci3n est3n comprendidas las historias de usuario 5 y 6, “Opciones de visualizaci3n del modelo 3D” y “Permitir el desplazamiento en el modelo 3D” respectivamente; logrando de esta forma una mayor interoperabilidad con el modelo.

Las tareas definidas para esta iteraci3n son las siguientes:

Historia de usuario N° 5 “Opciones de visualizaci3n del modelo 3D”

11. Tarea N° 11: Diseñar iconos del visor
12. Tarea N° 12: Implementar seis vistas (frontal, lateral izquierdo y derecho, trasera, superior e inferior)
13. Tarea N° 13: Implementar rotaci3n, traslaci3n y acercar o alejar.

Historia de usuario N° 6 “Permitir el desplazamiento en el modelo 3D”

14. Tarea N° 14: Implementar el desplazamiento dentro del modelo.

Tabla 26. Tarea 13: Diseñar iconos del visor

Tarea	
Número de tarea: 13	Número de HU: 5
Nombre de la tarea: Diseñar iconos del visor	
Tipo de tarea: Configuraci3n	Estimaci3n: 2 días
Fecha de inicio: 08 de marzo de 2012	Fecha fin: 09 de marzo de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripci3n: Diseñar iconos de la barra de herramientas del visor.	

Tabla 27. Tarea 14: Implementar seis vistas.

Tarea	
Número de tarea: 14	Número de HU: 5
Nombre de la tarea: Implementar seis vistas (frontal, lateral izquierdo y derecho, trasera, superior e inferior)	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 12 de marzo de 2012	Fecha fin: 14 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar las vistas (frontal, lateral izquierdo y derecho, trasera, superior e inferior) del modelo en el visor tridimensional.	

Tabla 28. Tarea 15: Implementar rotación, traslación y acercar o alejar

Tarea	
Número de tarea: 15	Número de HU: 5
Nombre de la tarea: Implementar rotación, traslación y acercar o alejar	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 15 de marzo de 2012	Fecha fin: 21 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar la rotación, traslación y acercar o alejar.	

Tabla 29. Tarea 16: Implementar el desplazamiento dentro del modelo

Tarea	
Número de tarea: 16	Número de HU: 6
Nombre de la tarea: Implementar el desplazamiento dentro del modelo	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 22 de marzo de 2012	Fecha fin: 26 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Implementar el desplazamiento dentro del modelo, de igual manera que un paseo virtual.	

3.2.3 Tercera Iteración

En la tercera iteración están comprendidas las historias de usuario 7, 8 y 9, “Gestionar elemento no estructural”, “Gestionar área de sensado” y “Gestionar dispositivo de

sensado” respectivamente. Con su desarrollo ya se tiene listo el segundo entregable de la aplicación con la mayor parte de las funcionalidades que debe incluir la misma.

Las tareas definidas para esta iteración son las siguientes:

Historia de usuario N° 7 “Gestionar elemento no estructural”

- 15. Tarea N° 15: Adicionar elemento no estructural
- 16. Tarea N° 16: Eliminar elemento no estructural
- 17. Tarea N° 17: Editar propiedades del elemento no estructural

Historia de usuario No 8 “Gestionar área de sensado”

- 18. Tarea No 18: Adicionar área de sensado
- 19. Tarea No 19: Eliminar área de sensado
- 20. Tarea No 20: Editar propiedades del área de sensado

Historia de usuario No 9 “Gestionar dispositivo de sensado”

- 21. Tarea No 21: Adicionar dispositivo de sensado
- 22. Tarea No 22: Eliminar dispositivo de sensado
- 23. Tarea No 23: Editar propiedades del dispositivo de sensado

Tabla 30. Tarea 17: Adicionar elemento no estructural

Tarea	
Número de tarea: 17	Número de HU: 7
Nombre de la tarea: Adicionar elemento no estructural	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 27 de marzo de 2012	Fecha fin: 29 de marzo de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Adicionar elementos no estructurales al modelo.	

Tabla 31. Tarea 18: Eliminar elemento no estructural

Tarea	
Número de tarea: 18	Número de HU: 7
Nombre de la tarea: Eliminar elemento no estructural	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 30 de marzo de 2012	Fecha fin: 30 de marzo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Eliminar elementos no estructurales del modelo, desde la vista tridimensional y la vista jerárquica.	

Tabla 32. Tarea 19: Editar propiedades del elemento no estructural

Tarea	
Número de tarea: 19	Número de HU: 7
Nombre de la tarea: Editar propiedades del elemento no estructural	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 02 de abril de 2012	Fecha fin: 03 de abril de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Editar propiedades del elemento no estructural.	

Tabla 33. Tarea 20: Adicionar área de sensado

Tarea	
Número de tarea: 20	Número de HU: 8
Nombre de la tarea: Adicionar área de sensado	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 04 de abril de 2012	Fecha fin: 09 de abril de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Adicionar área de sensado al modelo.	

Tabla 34. Tarea 21: Eliminar 1rea de sensado

Tarea	
N1mero de tarea: 21	N1mero de HU: 8
Nombre de la tarea: Eliminar 1rea de sensado	
Tipo de tarea: Desarrollo	Estimaci3n: 1 d1a
Fecha de inicio: 10 de abril de 2012	Fecha fin: 10 de abril de 2012
Programador responsable: Orlando Torres Saavedra	
Descripci3n: Eliminar 1rea de sensado del modelo, desde la vista tridimensional y la vista jer1rquica.	

Tabla 35. Tarea 22: Editar propiedades del 1rea de sensado

Tarea	
N1mero de tarea: 22	N1mero de HU: 8
Nombre de la tarea: Editar propiedades del 1rea de sensado	
Tipo de tarea: Desarrollo	Estimaci3n: 2 d1as
Fecha de inicio: 11 de abril de 2012	Fecha fin: 12 de abril de 2012
Programador responsable: Orlando Torres Saavedra	
Descripci3n: Editar propiedades del 1rea de sensado.	

Tabla 36. Tarea 23: Adicionar dispositivo de sensado

Tarea	
N1mero de tarea: 23	N1mero de HU: 9
Nombre de la tarea: Adicionar dispositivo de sensado	
Tipo de tarea: Desarrollo	Estimaci3n: 3 d1as
Fecha de inicio: 13 de abril de 2012	Fecha fin: 24 de abril de 2012
Programador responsable: Javier Bandomo Ru1z	
Descripci3n: Adicionar dispositivo de sensado desde la biblioteca de clases de sensores. Crear clase en el paquete cu.andromeda.model3D : <i>ContenDevice.java</i>	

Tabla 37. Tarea 24: Eliminar dispositivo de sensado

Tarea	
Número de tarea: 24	Número de HU: 9
Nombre de la tarea: Eliminar dispositivo de sensado	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 25 de abril de 2012	Fecha fin: 25 de abril de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Eliminar dispositivo de sensado del modelo, desde la vista tridimensional y la vista jerárquica.	

Tabla 38. Tarea 25: Editar propiedades del dispositivo de sensado

Tarea	
Número de tarea: 25	Número de HU: 9
Nombre de la tarea: Editar propiedades del dispositivo de sensado	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 26 de abril de 2012	Fecha fin: 30 de abril de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Editar propiedades del dispositivo de sensado.	

3.2.4 Cuarta Iteración

En esta iteración están comprendidas las Historias de Usuario 9 y 10, “Delimitar el área de sensado” y “Gestionar proyecto” respectivamente. Esta es la última iteración del desarrollo del producto, por lo cual al finalizar la misma el visor debe satisfacer todos los requisitos establecidos por el cliente.

Las tareas definidas para esta iteración son las siguientes:

Historia de usuario N° 10 “Exportar imagen bidimensional del modelo”

24. Tarea N° 24: Exportar imagen bidimensional del modelo

Historia de usuario No 11 “Gestionar proyecto”

25. Tarea N° 25: Crear nuevo proyecto

26. Tarea N° 26: Abrir proyecto

27. Tarea N° 27: Guardar proyecto

Tabla 39. Tarea 24: Exportar imagen bidimensional del modelo

Tarea	
Número de tarea: 24	Número de HU: 10
Nombre de la tarea: Exportar imagen bidimensional del modelo	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 02 de mayo de 2012	Fecha fin: 02 de mayo de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción: Exportar imagen bidimensional del modelo desde la vista aérea.	

Tabla 40. Tarea 25: Crear nuevo proyecto

Tarea	
Número de tarea: 25	Número de HU: 11
Nombre de la tarea: Crear nuevo proyecto	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 03 de mayo de 2012	Fecha fin: 04 de mayo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Crear nuevo proyecto	

Tabla 41. Tarea 26: Abrir proyecto

Tarea	
Número de tarea: 26	Número de HU: 11
Nombre de la tarea: Abrir proyecto	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 07 de mayo de 2012	Fecha fin: 09 de mayo de 2012
Programador responsable: Orlando Torres Saavedra	
Descripción: Abrir proyecto guardado con anterioridad.	

Tabla 42 Tarea 23: Guardar proyecto

Tarea	
Número de tarea: 23	Número de HU: 11
Nombre de la tarea: Guardar proyecto	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 10 de mayo de 2012	Fecha fin: 11 de mayo de 2012
Programador responsable: Javier Bandomo Ruíz	
Descripción:	
Consiste en guardar el estado actual del proyecto.	

3.3 Descripción del entorno

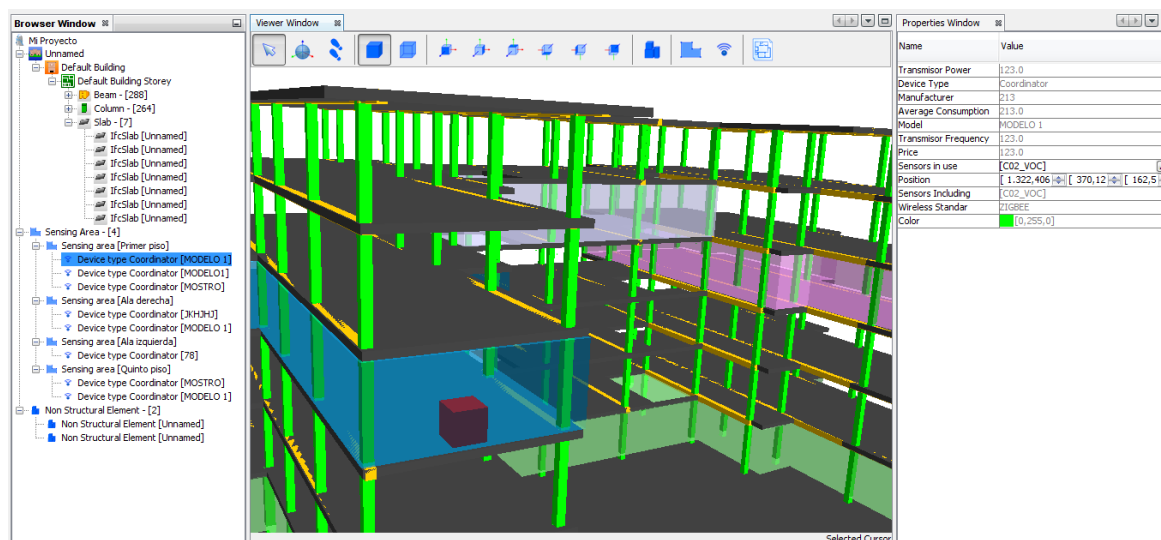


Figura 11 Interfaz del visor

Como todo visor tridimensional, el sistema posee un **área de trabajo (al centro)** en la cual será visualizado el modelo IFC importado, ofreciendo una vista lo más cercana a la realidad del entorno donde se concebirá la propuesta de diseño del despliegue de la WSAN. En ella se realizarán las funciones principales de visualización e interacción con el modelo.

A la izquierda se muestra el **árbol jerárquico de los componentes** que conforman el proyecto, distribuidos en conjuntos según su tipo. Proporciona una vista ordenada en carpetas desde la cual se pueden seleccionar los objetos y mostrar sus características en la **ventana de propiedades**.

La **ventana de propiedades** se encuentra a la derecha del 6rea de trabajo. Como su nombre lo indica, permite la visualizaci6n de las propiedades y edici6n del objeto seleccionado. A continuaci6n se muestran las propiedades editables seg6n el tipo de elemento:

Elementos del modelo IFC importado:

- Visible: Indica si el elemento es visible o no.

Elementos no estructurales⁸:

- Nombre: Nombre del elemento.
- Material: Tipo de material del elemento.
- Posici6n: Posici6n en el eje de coordenadas X, Y y Z.
- Dimensi6n: Alto, ancho y largo.
- Color: Color del elemento.
- Visible: Indica si el elemento es visible o no.

6rea de sensado⁹:

- Nombre: Nombre del elemento.
- Posici6n: Posici6n en el eje de coordenadas X, Y y Z.
- Altura: Altura del 6rea de sensado.
- Color: Color del 6rea de sensado.
- Visible: Indica si el 6rea de sensado es visible o no.

Dispositivo¹⁰:

- Sensores en uso: Sensores que usa el dispositivo
- Posici6n: Posici6n en el eje de coordenadas X, Y y Z.

Las propiedades, tanto de los dispositivos como de los objetos propios importados del modelo IFC, solo se visualizar6n y no ser6n editables, excepto las descritas anteriormente.

⁸ Elementos que afectan la cobertura y conectividad de los dispositivos en el despliegue

⁹ Regi6n de inter6s a sensar en el modelo.

¹⁰ Dispositivos de sensado



En la parte superior se encuentra la **barra de herramientas** con las funcionalidades principales del visor las cuales están divididas en seis grupos, distribuidos según las acciones que se realizan en el visor:

3.3.1 Navegación

- Cursor: Permite la selección de un elemento en el modelo desde la vista tridimensional.
- Orbit: Permite la rotación y traslación, así como acercar y alejar el modelo tridimensional. Cada una de estas acciones se realizan con el mouse (traslación: clic primario, traslación: clic secundario, acercar/alejar: rueda).
- Walk: Paseo virtual por el modelo.

3.3.2 Visualización

- Fill: Visualiza el modelo con los polígonos rellenos.
- Line: Visualiza solo las líneas que componen el modelo sin rellenar los polígonos.

3.3.3 Vistas

- Front: Muestra la vista frontal del modelo con el plano xy al fondo.
- Righth: Muestra la vista derecha del modelo con el plano yz al fondo.
- Top: Muestra la vista aérea del modelo con el plano xz al fondo.
- Down: Muestra el modelo desde el suelo con el plano xz al fondo.
- Left: Muestra la vista izquierda del modelo con el plano yz al fondo.
- Back: Muestra la vista trasera del modelo con el plano xy al fondo.

3.3.4 Elementos no estructurales

- Adicionar elemento: Agrega elementos no estructurales al modelo.

3.3.5 Área de sensado y dispositivos

- Adicionar área de sensado: Permite al usuario delimitar en el modelo el área de interés a sensar.
- Adicionar dispositivo: Incorpora al modelo un dispositivo seleccionado desde la librería de dispositivos que dispone la herramienta.

3.3.6 Exportar imagen

- Exportar imagen: Exporta una imagen en dos dimensiones desde la vista aérea del modelo.

3.4 Requerimientos mínimos.

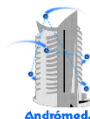
Para lograr la visualización e interacción con el modelo, es preciso definir algunas condiciones necesarias, con el fin de alcanzar un rendimiento óptimo y eficiencia del sistema. A continuación se presentan los requerimientos mínimos identificados:

- Máquina Virtual de Java versión 1.6
- Microprocesador Dual Core 2.0 GHz.
- Memoria RAM de 1 GB.
- Mouse con scroll.

3.5 Pruebas

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

Las técnicas de **caja blanca** están basadas en estudiar el código fuente, verificando su funcionalidad desde el punto de vista del cliente. Para comprobar el funcionamiento de los códigos que se van implementando, uno de los principales pilares de la metodología XP se basa en el Desarrollo Dirigido por Pruebas (12) o también conocido como pruebas unitarias, en inglés Test Driven Development (TDD). Estas pruebas están enfocadas en comprobar los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan realizar pruebas de las fallas que pudieran ocurrir.



TDD es un estilo de desarrollo donde el programador conforma un juego de pruebas antes de comenzar la implementación. Se escriben primero las pruebas y estas determinan el código a implementar. Una vez conformado el conjunto de pruebas, el código no pasa a producción a no ser que los resultados de las mismas sean satisfactorios. Se inicia el proceso de refactorización con el propósito de mejorar la estructura interna del código, aumentar su legibilidad y entendimiento, sin modificar su comportamiento externo.

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

3.5.1 Casos de Prueba de Aceptación para la versión 0.1

Para esta entrega se realizaron 9 Casos de pruebas de aceptación, identificando 2 no conformidades. Debido a la complejidad mínima de las no conformidades, fueron todas resueltas y no quedaron Casos de pruebas pendientes para la siguiente versión.

Tabla 43. CP Importar modelo IFC

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Importar modelo IFC	
Descripción: Prueba para la funcionalidad de Importar un modelo IFC al visor.	
Condiciones de Ejecución: El usuario debe poseer el modelo IFC y conocer la ruta del fichero.	
Entrada/ Pasos de ejecución: Con apoyo de la librería OpenIfcTool se parsea el fichero IFC a clases Java y se crea un nuevo objeto con los atributos necesarios para la visualización.	
Resultado Esperado: El modelo IFC es importado al visor sin errores, en un tiempo de 15 segundos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 44. CP Visualizar modelo IFC en un árbol jerárquico según su estructura

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Visualizar modelo IFC en un árbol jerárquico según su estructura	
Descripción: Prueba para la funcionalidad de Visualizar un modelo IFC en un árbol jerárquico según su estructura.	
Condiciones de Ejecución: El usuario debe haber importado el modelo IFC.	
Entrada/ Pasos de ejecución: Se extraen del objeto RootBranch los elementos necesarios para la generación del árbol y se visualizan en una ventana.	
Resultado Esperado: El modelo IFC es visualizado mediante un árbol jerárquico sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 45. CP Visualizar un modelo IFC en el área de trabajo

Caso de prueba de aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Visualizar un modelo IFC en el área de trabajo	
Descripción: Prueba para la funcionalidad de Visualizar un modelo IFC en el área de trabajo.	
Condiciones de Ejecución: El usuario debe haber importado el modelo IFC.	
Entrada/ Pasos de ejecución: Se extraen del objeto RootBranch los elementos necesarios para la visualización del modelo IFC utilizando la librería Java3D.	
Resultado Esperado: El modelo IFC es visualizado en el área de trabajo sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 46. CP Cambiar tipo de visualización del modelo IFC a sólida

Caso de prueba de aceptación	
Código: HU2_P3	Historia de Usuario: 2
Nombre: Cambiar tipo de visualización del modelo IFC a sólida	
Descripción: Prueba para la funcionalidad cambiar tipo de visualización del modelo IFC a sólida.	
Condiciones de Ejecución: El usuario debe haber importado el modelo IFC.	
Entrada/ Pasos de ejecución: Se selecciona de la barra de herramientas el botón Solid.	
Resultado Esperado: El modelo IFC es visualizado en la vista sólida sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 47. CP Cambiar tipo de visualización del modelo IFC a lineal

Caso de prueba de aceptación	
Código: HU2_P4	Historia de Usuario: 2
Nombre: Cambiar tipo de visualización del modelo IFC a lineal	
Descripción: Prueba para la funcionalidad cambiar tipo de visualización del modelo IFC a lineal.	
Condiciones de Ejecución: El usuario debe haber importado el modelo IFC.	
Entrada/ Pasos de ejecución: Se selecciona de la barra de herramientas el botón Line.	
Resultado Esperado: El modelo IFC es visualizado en la vista lineal sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 48. CP Seleccionar los elementos del modelo IFC desde el árbol jerárquico

Caso de prueba de aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Seleccionar los elementos del modelo desde el árbol jerárquico	
Descripción: Prueba para la funcionalidad de seleccionar los elementos del modelo desde el árbol jerárquico.	
Condiciones de Ejecución: El usuario debe haber importado el modelo IFC.	
Entrada/ Pasos de ejecución: Se selecciona desde el árbol jerárquico con clic izquierdo el elemento deseado.	
Resultado Esperado: Se puede seleccionar cualquier elemento del modelo desde el árbol jerárquico sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 49. CP Seleccionar los elementos del modelo IFC desde la vista tridimensional

Caso de prueba de aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Seleccionar los elementos del modelo desde la vista tridimensional	
Descripción: Prueba para la funcionalidad de seleccionar los elementos del modelo desde la vista tridimensional.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona de la barra de herramientas el botón Cursor y desde la vista tridimensional con clic izquierdo se selecciona el elemento deseado.	
Resultado Esperado: Se puede seleccionar cualquier elemento del modelo desde la vista tridimensional sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 50. CP Visualizar las propiedades del elemento seleccionado desde el 6rbol jer6rquico

Caso de prueba de aceptaci3n	
C3digo: HU4_P1	Historia de Usuario: 4
Nombre: Visualizar las propiedades del elemento seleccionado desde el 6rbol jer6rquico.	
Descripci3n: Prueba para la funcionalidad de visualizar las propiedades de elementos del modelo seleccionado desde el 6rbol jer6rquico.	
Condiciones de Ejecuci3n: El sistema debe haber visualizado el modelo IFC importado en un 6rbol jer6rquico.	
Entrada/ Pasos de ejecuci3n: Se selecciona desde el 6rbol jer6rquico con clic izquierdo el elemento deseado extrayendo del nodo sus propiedades.	
Resultado Esperado: Se visualizan las propiedades del elemento del modelo seleccionado desde el 6rbol jer6rquico en la ventana Propiedades sin errores.	
Evaluaci3n de la Prueba: Prueba satisfactoria.	

Tabla 51. CP Visualizar las propiedades del elemento seleccionado desde la vista tridimensional

Caso de prueba de aceptaci3n	
C3digo: HU4_P2	Historia de Usuario: 4
Nombre: Visualizar las propiedades del elemento seleccionado desde la vista tridimensional.	
Descripci3n: Prueba para la funcionalidad de visualizar las propiedades de elementos del modelo seleccionado desde la vista tridimensional.	
Condiciones de Ejecuci3n: El sistema debe haber visualizado el modelo IFC importado en el 6rea de trabajo.	
Entrada/ Pasos de ejecuci3n: Se selecciona en la barra de herramientas el bot3n Cursor y desde la vista tridimensional con clic izquierdo se selecciona el elemento deseado.	
Resultado Esperado: Se visualizan las propiedades del elemento del modelo seleccionado desde la vista tridimensional en la ventana Propiedades sin errores.	
Evaluaci3n de la Prueba: Prueba satisfactoria.	

3.5.2 Casos de Prueba de Aceptaci3n para la versi3n 0.2

Para esta entrega se realizaron 16 Casos de pruebas de aceptaci3n, identificando 5 no conformidades de las cuales 2 no procedieron. Se le dio soluci3n a las no conformidades, por lo que no quedaron Casos de pruebas pendientes para la siguiente versi3n.

Tabla 52. CP Permitir la rotación de la vista 3D

Caso de prueba de aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Permitir la rotación de la vista 3D	
Descripción: Prueba para la funcionalidad de permitir la rotación de la vista 3D	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Orbita y con el clic izquierdo presionado se rota la vista 3D.	
Resultado Esperado: Se puede rotar la vista 3D sin errores, en un tiempo inferior a un segundo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 53. CP Permitir acercar y alejar la vista 3D

Caso de prueba de aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Permitir acercar y alejar la vista 3D	
Descripción: Prueba para la funcionalidad permitir acercar y alejar la vista 3D.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Orbit y con la rueda del <i>mouse</i> se acerca o aleja la vista 3D.	
Resultado Esperado: Se puede acercar y alejar la vista 3D sin errores, en un tiempo inferior a un segundo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 54. CP Mover la vista 3D

Caso de prueba de aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Mover la vista 3D	
Descripción: Prueba para la funcionalidad Mover la vista 3D.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Orbit y con el clic derecho se mueve la vista 3D.	
Resultado Esperado: Se puede mover la vista 3D sin errores, en un tiempo inferior a un segundo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 55. CP Permitir el desplazamiento en el modelo 3D

Caso de prueba de aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Permitir el desplazamiento en el modelo 3D	
Descripción: Prueba para la funcionalidad permitir el desplazamiento en el modelo 3D.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Walk y mediante las teclas de dirección o el mouse se permite el desplazamiento por el modelo 3D.	
Resultado Esperado: Se permite el desplazamiento en el modelo 3D sin errores, en un tiempo inferior a un segundo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 56. CP Adicionar elementos no estructurales al modelo IFC visualizado

Caso de prueba de aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Adicionar elemento no estructural al modelo IFC visualizado	
Descripción: Prueba para la funcionalidad adicionar elementos no estructurales al modelo IFC visualizado.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Non Structural Element. El cursor identifica la acción y se selecciona en el modelo el lugar donde se quiere insertar el nuevo elemento. Llenar un formulario con los datos requeridos: nombre, material y color.	
Resultado Esperado: Se inserta un nuevo elemento no estructural al modelo IFC sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 57. CP Eliminar elemento no estructural desde la vista tridimensional

Caso de prueba de aceptación	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Eliminar elemento no estructural desde la vista tridimensional	
Descripción: Prueba para la funcionalidad eliminar elemento no estructural desde la vista tridimensional.	
Condiciones de Ejecución: Debe haber insertado al menos un elemento no estructural.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Cursor y desde la vista tridimensional con clic izquierdo se selecciona el elemento no estructural a eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar un elemento no estructural del modelo desde la vista tridimensional sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 58. CP Eliminar elemento no estructural desde el árbol jerárquico

Caso de prueba de aceptación	
Código: HU7_P3	Historia de Usuario: 7
Nombre: Eliminar elemento no estructural desde el árbol jerárquico	
Descripción: Prueba para la funcionalidad eliminar elemento no estructural desde el árbol jerárquico.	
Condiciones de Ejecución: Debe haber insertado al menos un elemento no estructural.	
Entrada/ Pasos de ejecución: Se selecciona del árbol jerárquico el elemento no estructural que desea eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar un elemento no estructural del modelo desde el árbol jerárquico sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 59. CP Editar propiedades del elemento no estructural

Caso de prueba de aceptación	
Código: HU7_P4	Historia de Usuario: 7
Nombre: Editar propiedades del elemento no estructural	
Descripción: Prueba para la funcionalidad editar propiedades del elemento no estructural.	
Condiciones de Ejecución: Debe haber insertado al menos un elemento no estructural.	
Entrada/ Pasos de ejecución: Se puede seleccionar del árbol jerárquico o desde la vista 3D el elemento no estructural y editar sus propiedades.	
Resultado Esperado: Se puede editar las propiedades del elemento no estructural sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 60. CP Adicionar el área de sensado en el modelo IFC visualizado

Caso de prueba de aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Adicionar área de sensado al modelo IFC visualizado	
Descripción: Prueba para la funcionalidad delimitar el área de sensado el modelo IFC visualizado.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Sensing Area. El cursor identifica la acción, se delimita en el modelo el lugar donde se quiere insertar el área de sensado. Llenar un formulario con los datos requeridos: nombre, tipos de sensado y modelo de propagación.	
Resultado Esperado: Se puede adicionar el área de sensado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 61. CP Eliminar área de sensado desde la vista tridimensional

Caso de prueba de aceptación	
Código: HU8_P2	Historia de Usuario: 8
Nombre: Eliminar área de sensado desde la vista tridimensional.	
Descripción: Prueba para la funcionalidad eliminar área de sensado desde la vista tridimensional.	
Condiciones de Ejecución: Debe haber insertado al menos un área de sensado.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Cursor y desde la vista tridimensional con clic izquierdo se selecciona el área de sensado a eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar el área de sensado desde la vista tridimensional sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 62. CP Eliminar área de sensado desde el árbol jerárquico

Caso de prueba de aceptación	
Código: HU8_P3	Historia de Usuario: 8
Nombre: Eliminar área de sensado desde el árbol jerárquico	
Descripción: Prueba para la funcionalidad eliminar área de sensado desde el árbol jerárquico.	
Condiciones de Ejecución: Debe haber insertado al menos un área de sensado.	
Entrada/ Pasos de ejecución: Se selecciona del árbol jerárquico el área de sensado que desea eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar un área de sensado del modelo desde el árbol jerárquico sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 63. CP Editar propiedades del área de sensado

Caso de prueba de aceptación	
Código: HU8_P4	Historia de Usuario: 7
Nombre: Editar propiedades del área de sensado	
Descripción: Prueba para la funcionalidad editar propiedades del área de sensado.	
Condiciones de Ejecución: Debe haber insertado al menos un área de sensado.	
Entrada/ Pasos de ejecución: Se puede seleccionar del árbol jerárquico o desde la vista 3D el área de sensado y editar sus propiedades.	
Resultado Esperado: Se puede editar las propiedades del área de sensado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 64. CP Adicionar dispositivo de sensado al modelo IFC visualizado

Caso de prueba de aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Adicionar dispositivo de sensado al modelo IFC visualizado	
Descripción: Prueba para la funcionalidad adicionar dispositivo de sensado al modelo IFC visualizado.	
Condiciones de Ejecución: Debe haber insertado al menos un área de sensado.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Device. El cursor identifica la acción y se selecciona en el modelo el área de sensado donde se quiere insertar el dispositivo. Llenar un formulario con los datos requeridos.	
Resultado Esperado: Se puede insertar un nuevo dispositivo de sensado al modelo IFC sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 65. CP Eliminar dispositivo de sensado desde la vista tridimensional

Caso de prueba de aceptación	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Eliminar dispositivo de sensado desde la vista tridimensional	
Descripción: Prueba para la funcionalidad dispositivo de sensado desde la vista tridimensional.	
Condiciones de Ejecución: Debe haber insertado al menos un dispositivo de sensado.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Cursor y desde la vista tridimensional con clic izquierdo se selecciona el dispositivo de sensado a eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar el dispositivo de sensado desde la vista tridimensional sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 66. CP Eliminar dispositivo de sensado desde el árbol jerárquico

Caso de prueba de aceptación	
Código: HU9_P3	Historia de Usuario: 9
Nombre: Eliminar dispositivo de sensado desde el árbol jerárquico.	
Descripción: Prueba para la funcionalidad eliminar dispositivo de sensado desde el árbol jerárquico.	
Condiciones de Ejecución: Debe haber insertado al menos un dispositivo de sensado.	
Entrada/ Pasos de ejecución: Se selecciona del árbol jerárquico el dispositivo de sensado que desea eliminar. Luego presionar la tecla Del y confirmar la eliminación.	
Resultado Esperado: Se puede eliminar un dispositivo de sensado del modelo desde el árbol jerárquico sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 67. CP Editar propiedades del dispositivo de sensado

Caso de prueba de aceptación	
Código: HU9_P4	Historia de Usuario: 9
Nombre: Editar propiedades del dispositivo de sensado.	
Descripción: Prueba para la funcionalidad editar propiedades del dispositivo de sensado.	
Condiciones de Ejecución: Debe haber insertado al menos un dispositivo de sensado.	
Entrada/ Pasos de ejecución: Se puede seleccionar del árbol jerárquico o desde la vista 3D el dispositivo de sensado y editar sus propiedades.	
Resultado Esperado: Se puede editar las propiedades del dispositivo de sensado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

3.5.3 Casos de Prueba de Aceptación para la versión 1.0

Para esta entrega se realizaron 4 Casos de pruebas de aceptación, identificando 1 no conformidad. Esta no conformidad se solucionó, dando fin al proceso de pruebas de aceptación obteniendo resultados satisfactorios.

Tabla 68. CP Exportar imagen bidimensional del modelo

Caso de prueba de aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Exportar imagen bidimensional del modelo	
Descripción: Prueba para la exportar imagen bidimensional del modelo.	
Condiciones de Ejecución: El sistema debe haber visualizado el modelo IFC importado en el área de trabajo.	
Entrada/ Pasos de ejecución: Se selecciona en la barra de herramientas el botón Image. Cambia el modelo a la vista lineal, luego presionar la tecla Enter y se selecciona la ruta donde se va guardar de la imagen.	
Resultado Esperado: Se puede exportar imagen bidimensional del modelo sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 69. CP Crear nuevo proyecto

Caso de prueba de aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Crear nuevo proyecto	
Descripción: Prueba para la crear nuevo proyecto.	
Condiciones de Ejecución: -	
Entrada/ Pasos de ejecución: Se selecciona del menú principal Crear nuevo proyecto, se recogen los datos requeridos y se especifica la ubicación del fichero IFC.	
Resultado Esperado: El nuevo proyecto es creado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 70. CP Abrir proyecto existente

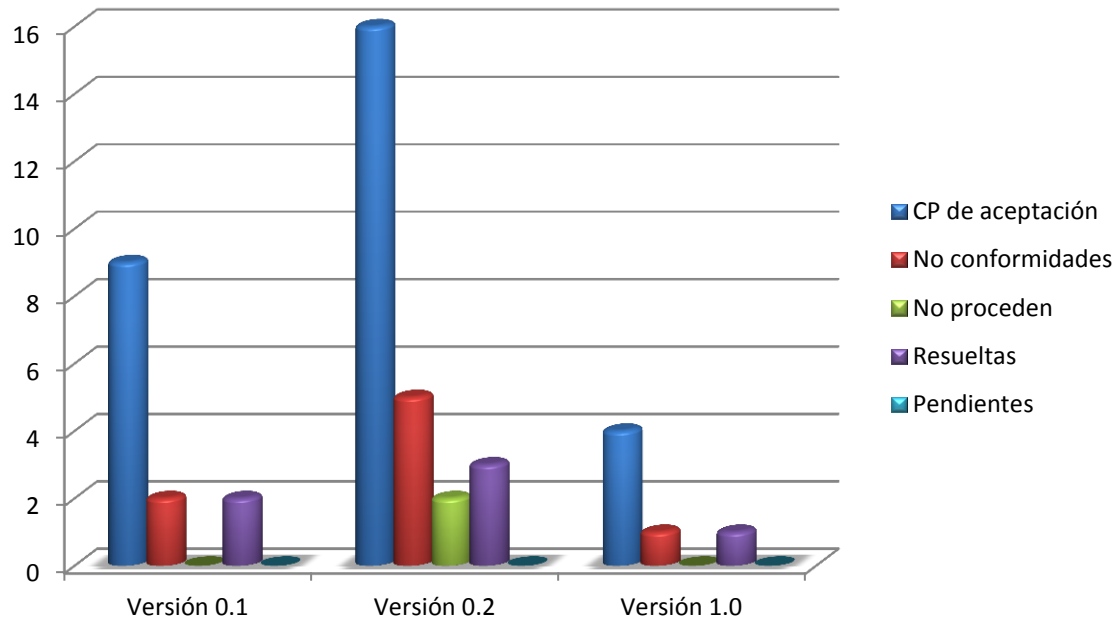
Caso de prueba de aceptación	
Código: HU11_P2	Historia de Usuario: 11
Nombre: Abrir proyecto existente	
Descripción: Prueba para abrir proyecto existente.	
Condiciones de Ejecución: Debe existir un proyecto previamente creado.	
Entrada/ Pasos de ejecución: Se selecciona del menú principal abrir proyecto y se navega a su ubicación.	
Resultado Esperado: El proyecto es cargado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 71. CP Guardar proyecto existente

Caso de prueba de aceptación	
Código: HU11_P3	Historia de Usuario: 11
Nombre: Guardar proyecto existente	
Descripción: Prueba para guardar proyecto existente.	
Condiciones de Ejecución: Debe haber creado el proyecto previamente.	
Entrada/ Pasos de ejecución: Se selecciona del menú principal guardar proyecto.	
Resultado Esperado: El proyecto es guardado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

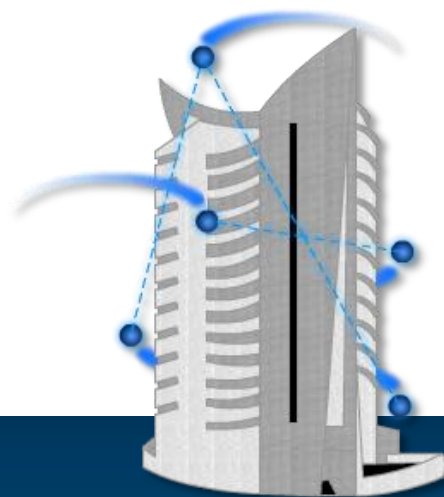
3.6 Resultados generales de las pruebas

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



3.7 Conclusiones

En este capítulo se abordaron todos los temas referentes a la implementación y prueba de la solución propuesta. Primeramente se presentó un diseño detallado de la arquitectura, se describieron las Tareas de Ingeniería al igual que el entorno y por último el proceso de pruebas, concluyendo el proceso con resultados satisfactorios dándole solución a las no conformidades identificadas.



Andr6meda
Andr6meda

Conclusiones y Recomendaciones



Conclusiones

Al término de la presente investigación se concluye lo siguiente:

- Se desarrolló un visor tridimensional de entornos interiores para la herramienta Andrómeda, siguiendo la metodología XP para guiar el proceso de desarrollo, Java como lenguaje de programación, la librería Java 3D para el modelado tridimensional, NetBeans como Entorno de Desarrollo y Visual Paradigm for UML como herramienta de modelado.
- La aplicación desarrollada permite en su primera versión, la visualización de modelos IFC en tres dimensiones generados por herramientas de Modelado de Información para la Edificación y posibilita la gestión de componentes de Redes Inalámbricas de Sensores y Actuadores en el entorno tridimensional.
- Luego de haberse realizado el proceso de Pruebas de Aceptación se demostró la correcta implementación de sus funcionalidades dando cumplimiento a las necesidades del cliente.
- El visor desarrollado le permitirá al diseñador de WSAN tener una representación virtual del entorno para el cual desea realizar el diseño del despliegue de la red.



Recomendaciones

A partir del trabajo realizado y después de haber analizado los resultados obtenidos se recomienda:

- Implementar nuevos módulos para importar otros modelos 3D al visor.
- Implementar una API que brinde mecanismos para interactuar con el modelo 3D importado, lo que permitirá apoyar a la integración de los módulos que conforman la herramienta Andrómeda.



Referencias Bibliográficas

1. **Muñoz Castej6n, Rodrigo.** Interconexi6n de redes de sensores inalámbricos 802.15.4 en localizaciones remotas. Madrid : s.n., 2011.
2. **Libelium Comunicaciones Distribuidas S.L.** Libelium. [En l6nea] <http://www.libelium.com>.
3. **Ismael, Nodarse Mora Armando y Manuel, D6az Rodr6guez.** Utilizaci6n de WSA-Ns en Sistemas de Control de Edificios. 2010.
4. **Data Desing System.** Data Desing System. [En l6nea] <http://www.dds-cad.net>.
5. **Digital Alchemy.** Digital Alchemy. [En l6nea] 2008.
http://www.digitalalchemypro.com/Joomla/index.php?option=com_content&task=view&id=16&Itemid=29.
6. **X3DMedia.** AEC Megazine. [En l6nea] 2000.
<http://aecmag.com/index.php?option=content&task=view&id=439..>
7. **Nemetschek.** Nemetschek IFC Viewer. [En l6nea]
[http://serviceplus.nemetschek.de/de/service/onlinebrochures.nsf/\(files\)/70430114200-2/\\$FILE/nemetschek_ifcviewer_eng.pdf](http://serviceplus.nemetschek.de/de/service/onlinebrochures.nsf/(files)/70430114200-2/$FILE/nemetschek_ifcviewer_eng.pdf).
8. **QTX de M6xico.** Qualitrain. [En l6nea] 2001.
<http://www.qualitrain.com.mx/objeIndirecto/javavsvbasic.htm>.
9. **Tulke, Jan, y otros, y otros.** Open Ifc Tools. [En l6nea] 2010.
http://www.openifctools.org/Open_IFC_Tools/Features.html.
10. **Oracle Corporation.** NetBeans. [En l6nea] <http://netbeans.org>.
11. **Visual Paradigm International.** Visual Paradigm . [En l6nea] 2005. <http://www.visual-paradigm.com>.
12. Los Procesos 6giles: SCRUM y eXtreme Programming. [En l6nea]
<http://procesosagiles.com/index.html>.
13. **Guglielmetti, Marcos, y otros, y otros.** Master Megazine. [En l6nea] 2005.
<http://www.mastermagazine.info/termino/3868.php>.
14. **Graphisoft.** Graphisoft. [En l6nea] <http://www.graphisoft.es/producto/archicad/>.
15. **IG Network.** IG Network. [En l6nea] <http://ignetwork.net/>.
16. **Graphisoft.** Anzix. [En l6nea] <http://www.anzix.com/html/ifc.html>.
17. **Free Software Foundation.** Free Software Foundation. [En l6nea] <http://www.fsf.org/es>.
18. **Albuera, Pablo.** Software Propietario. [En l6nea] <http://www.slideshare.net/pabloalbuera/software-proprietario-1569982>.
19. **Shopping Guide GmbH.** Ciao. [En l6nea] <http://www.ciao.es/>.
20. **Phyton Software Foundation.** Phyton. [En l6nea] <http://www.python.org/>.
21. **G6nzales Duque, Ra6l.** *Phyton para todos*.
22. **Aertia Software.** Aertia Software. [En l6nea] 2004. <http://www.aertia.com/productos.asp?pid=237>.
23. **AForge.net.** AForge.net. [En l6nea] 2008. <http://www.aforgenet.com/>.
24. **Adobe Systems Incorporated.** Adobe. [En l6nea] http://www.adobe.com/actionscript3_overview/.
25. **Agile Alliance.** Programaci6n Extrema. [En l6nea] 2007. <http://www.programacionextrema.org>.



Bibliografía

1. **Muñoz Castej6n, Rodrigo.** Interconexi6n de redes de sensores inal6mbricos 802.15.4 en localizaciones remotas. Madrid : s.n., 2011.
2. **Libelium Comunicaciones Distribuidas S.L.** Libelium. [En l6nea] <http://www.libelium.com>.
3. **Ismael, Nodarse Mora Armando y Manuel, D6az Rodr6guez.** Utilizaci6n de WSA-Ns en Sistemas de Control de Edificios. 2010.
4. **Data Desing System.** Data Desing System. [En l6nea] <http://www.dds-cad.net>.
5. **Digital Alchemy.** Digital Alchemy. [En l6nea] 2008.
http://www.digitalalchemypro.com/Joomla/index.php?option=com_content&task=view&id=16&Itemid=29.
6. **X3DMedia.** AEC Magazine. [En l6nea] 2000.
<http://aecmag.com/index.php?option=content&task=view&id=439..>
7. **Nemetschek.** Nemetschek IFC Viewer. [En l6nea]
[http://serviceplus.nemetschek.de/de/service/onlinebrochures.nsf/\(files\)/70430114200-2/\\$FILE/nemetschek_ifcviewer_eng.pdf](http://serviceplus.nemetschek.de/de/service/onlinebrochures.nsf/(files)/70430114200-2/$FILE/nemetschek_ifcviewer_eng.pdf).
8. **QTX de M6xico.** Qualitrain. [En l6nea] 2001.
<http://www.qualitrain.com.mx/objeIndirecto/javavsvbasic.htm>.
9. **Tulke, Jan, y otros, y otros.** Open Ifc Tools. [En l6nea] 2010.
http://www.openifctools.org/Open_IFC_Tools/Features.html.
10. **Oracle Corporation.** NetBeans. [En l6nea] <http://netbeans.org>.
11. **Visual Paradigm International.** Visual Paradigm . [En l6nea] 2005. <http://www.visual-paradigm.com>.
12. Los Procesos 6giles: SCRUM y eXtreme Programming. [En l6nea]
<http://procesosagiles.com/index.html>.
13. **Guglielmetti, Marcos, y otros, y otros.** Master Magazine. [En l6nea] 2005.
<http://www.mastermagazine.info/termino/3868.php>.
14. **Graphisoft.** Graphisoft. [En l6nea] <http://www.graphisoft.es/producto/archicad/>.
15. **IG Network.** IG Network. [En l6nea] <http://ignetwork.net/>.
16. **Graphisoft.** Anzix. [En l6nea] <http://www.anzix.com/html/ifc.html>.
17. **Free Software Foundation.** Free Software Foundation. [En l6nea] <http://www.fsf.org/es>.
18. **Albuera, Pablo.** Software Propietario. [En l6nea] <http://www.slideshare.net/pabloalbuera/software-proprietario-1569982>.
19. **Shopping Guide GmbH.** Ciao. [En l6nea] <http://www.ciao.es/>.
20. **Phyton Software Foundation.** Phyton. [En l6nea] <http://www.python.org/>.
21. **G6nzales Duque, Ra6l.** *Phyton para todos*.
22. **Aertia Software.** Aertia Software. [En l6nea] 2004. <http://www.aertia.com/productos.asp?pid=237>.
23. **AForge.net.** AForge.net. [En l6nea] 2008. <http://www.aforgenet.com/>.
24. **Adobe Systems Incorporated.** Adobe. [En l6nea] http://www.adobe.com/actionscript3_overview/.
25. **Agile Alliance.** Programaci6n Extrema. [En l6nea] 2007. <http://www.programacionextrema.org>.
26. **Selman, Daniel.** *Java 3D Programing*.
27. **Garc6a de Jal6n, Javier, y otros, y otros.** *Aprenda Java como si estuviera en primero*.
28. **Schildt, Herbert.** *Java 2 Manual de referencias*.
29. Modelado Gr6fico Java 3D. [En l6nea] <http://java.sun.com/products/java-media/3d>.
30. **Apress.** *The Definitive Guide to NetBeans Platform 7*.
31. **Martinez Ju6n, Francisco Javier.** *Gu6a de construcci6n de software en Java con patrones de dise6o*.
32. **Graphisoft.** *IFC 2x3 Reference Guide AC15*.
33. **Liebich, Thomas.** *Scope of IFC Specification*. 2004.
34. **Hietanen, Jiri.** *Wall geometry in IFC R2.0*. 1999.
35. **Liebich, Thomas.** *IFC 2X Edition 2 Model Implementation Guide*. 2004.
36. **Sierra, Kathy y Bates, Bert.** *Head First Java 2nd Edition*.
37. **Karl, Seguin.** *Foundation of Programming* .
38. **S. Wright, Richard y Lipchak, Benjamin.** *OpenGL Superbible Third Edition* . Indianapolis : s.n., 2005.
39. **Barry, Paul y Griffiths, David.** *Head First Programming*. 2009.
40. **Herbert, Schildt.** *Java 2 Manual de Referencia* . Madrid : s.n., 2001.
41. **Ambler, S.W.** *Artifacts for Agile Modeling: The UML and Beyond*. 2011.



42. **Michael, Theiler.** *Documentation and conceptual development of software components for the execution of geometric Boolean set.* 2010.
43. —. *IFC-Import-Funktion für Cademia.* 2009.
44. —. *Interaktives Ermitteln von Bauteilinformationen aus STEP-Dateien.* 2008.
45. **Theiler Michael, Tauscher Eike, Tulke Jan, Riedel Thomas.** *Visualisierung von IFC-Objekten mittels Java3D.* 2009.
46. **Theiler Michael, Matrikel.** *Visualization of the heat flow inside a wall corner with ceiling.* 2009.
47. **Theiler Michael, Tauscher Eike, Tulke Jan, Riedel Thomas.** *Boolesche Operationen für die Visualisierung von IFC-Gebäudemodellen.* 2010.
48. **Theiler, Michael.** *M A S T E R A R B E I T.* 2010.
49. **Katranuschkov, Peter.** *BIM-BASED GENERATION OF MULTI-MODEL VIEWS.* 2010.
50. **Romero Pérez, Victor.** *CREACIÓN DE UN ENTORNO 3D PARA LA SIMULACIÓN DE TRÁFICO URBANO.* 2009.



Glosario

API: Application Program Interface o Interfaz de Programación de Aplicaciones en su traducción al español. Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones. (13)

BIM: El concepto de BIM (*Building Information Modeling* o *Modelado de Información para la Edificación*, en su traducción al español) o Edificio Virtual™ de Graphisoft¹¹, abarca todo el proceso de diseño y gestión de toda la información a lo largo del ciclo de vida del edificio. No sólo es un simple modelo 3D en un ordenador, el Edificio Virtual contiene además con gran detalle información adicional sobre los materiales del edificio y sus características. Es una base de datos tridimensional que hace un seguimiento de todos los elementos que componen el edificio. Esta información puede incluir área y volumen de superficies, propiedades térmicas, descripciones de las habitaciones, precios, información sobre especificaciones del producto, ventanas, puertas y acabados. (14)

Framework: En el desarrollo de software, un *framework* o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, además, provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

IDE: Un entorno de desarrollo integrado, por sus siglas en inglés de *integrated development environment*, es un programa compuesto por un conjunto de herramientas para los programadores. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor

¹¹Graphisoft es pionero y líder en el desarrollo de soluciones de Virtual™ Building.



de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. (15)

IFC: *Industry Foundation Classes (IFC)* es un formato que incluye elementos constructivos como ventanas, puertas, muros, losas, entre otras. Está basado en la idea de objetos que describen de forma total el Edificio Virtual. Estos objetos están definidos para dar continuidad al ciclo de vida del proyecto desde su concepción hasta las etapas de diseño, documentación y construcción. Este archivo está disponible a todos los participantes de la industria de la construcción, para su uso en todos los países, aprovechando este lenguaje común para compartir elementos inteligentes del Edificio Virtual con otras disciplinas. Es un formato de uso general para BIM, siendo una especificación abierta y no está controlada por un único proveedor. (16)

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

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

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

Software libre: Hace referencia a la libertad de los usuarios o desarrolladores de tener acceso libre al código fuente para copiarlo, leerlo, distribuirlo, cambiarlo y mejorarlo a su gusto. Existen diferentes variantes. La principal, promovida por la Free Software Foundation¹², es la más utilizada en el mundo, respaldada legalmente por la licencia GNU/GPL (del inglés “General Public License”). Existen otras variantes de código abierto como Mozilla Public License, L-GPL, etc. (17)

Software privativo: Se refiere a todo aquel software o fragmento de software cuya licencia comercial, de uso o de publicación, no permite compartir el código fuente del mismo o, lo hace, pero restringe las libertades de uso del mismo. (18)

¹²www.fsf.org