



## **Herramienta para el desarrollo de paseos virtuales interactivos con elementos en 3D**

Trabajo final presentado en opción al título de Máster en Informática Aplicada

Autor: Minardo Gollún González López.

Tutores: Dr.C. Ramón Carrasco Velar.

MSc. Lidiexy Alonso Hernández.

La Habana, febrero de 2012

## **DECLARACIÓN JURADA DE AUTORÍA**

Declaro por este medio que yo Minardo Gollún González López con carné de identidad 82080427462, soy el autor principal del trabajo final de maestría “Herramienta para el desarrollo de paseos virtuales interactivos con elementos en 3D”, desarrollada como parte de la Maestría en Informática Aplicada y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, con los mismos derechos del autor.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Minardo Gollún González López**

**Nombre del maestrante**

\_\_\_\_\_

**Firma**

### *Agradecimientos*

- *A mis tutores Ramón Carrasco Velar y Lidiexy Alonso Hernández por todas sus observaciones, sus consejos y su paciencia.*
- *A Andy Torres, Enier Molina, Yanoski R. Camacho, “Grupo de visualización gráfica y realidad virtual” y a todos los que han colaborado de alguna forma a la realización de este trabajo.*
- *A mi familia por comprenderme y aguantarme.*
- *A mi novia que ha sido crucial en los últimos 5 años de mi vida.*

### *Dedicatoria*

- *A toda mi familia y especialmente a mi mamá.*

### Resumen

En la tesis se define el concepto de Centros Expositivos Virtuales como una nueva entidad informática, materializada en una herramienta de escritorio. Se describe cómo se desarrolló la solución y se analiza su inserción en el proceso actual de desarrollo de paseos virtuales, usado actualmente en el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas. La investigación se desarrolló guiada por la metodología RUP y se basó en el estudio de editores de escenarios, herramientas de diseño gráfico y otras tecnologías de desarrollo. La propuesta de inserción en el proceso de desarrollo se analizó mediante la evaluación de los cambios introducidos en la etapa de implementación y cómo estos suponen una simplificación de los elementos que componen el proceso. De este modo se logró una solución destinada fundamentalmente a usuarios sin experiencia en el uso de las técnicas de programación y diseño gráfico. La herramienta permite recrear virtualmente un centro expositivo en el cual se puede crear, editar, visualizar paseos virtuales interactivos con elementos en tres dimensiones y administrar los recursos disponibles.

**Palabras clave:** Centros Expositivos Virtuales, ingeniería del software, paseos virtuales, realidad virtual.

### **Abstract**

In the present scientific research the concept Virtual Exhibition Center is defined as a new term that belongs to the computer sciences area, developed as a desktop tool. The development of the solution is described as well as its insertion in the ongoing development of virtual tours, currently used in Industrial Computing Center at the University of Computer Sciences. The research is carried out following the RUP Methodology and based on the study of scenes editors, graphic design tools and other development technologies. The insertion of this tool in the development process was analyzed and proposed through evaluating changes in the implementation phase and how these represent a simplification of the elements of the process. This proposal is oriented mainly to inexperienced users in the use of programming techniques and graphic design, and to recreate a virtual exhibition center where you can create, edit, view virtual tours, interactive three-dimensional elements, and manage available resources.

**Keywords:** Virtual Exhibition Centers, software engineering, virtual tours, virtual reality.

**ÍNDICE**

**Introducción ..... 1**

**Capítulo 1: Fundamentación teórica..... 7**

    Introducción..... 7

    1.1 Realidad virtual ..... 7

    1.2 Paseos virtuales..... 9

    1.3 Metodologías de desarrollo de software ..... 9

        1.3.1 RUP ..... 10

    1.4 Implementación de un paseo virtual..... 10

    1.5 Herramientas para el diseño de escenarios virtuales en 3D ..... 12

        1.5.1 Autodesk 3D Studio Max..... 13

        1.5.2 Autodesk Maya ..... 13

        1.5.3 Blender ..... 14

    1.6 Motores de juegos..... 15

        1.6.1 G3D ..... 15

        1.6.2 Ogre..... 16

        1.6.3 SceneToolKit (STK) ..... 17

    1.7 Entornos de Desarrollo Integrado (IDE) ..... 17

        1.7.1 Microsoft Visual Studio..... 17

        1.7.2 Qt Creator ..... 18

        1.7.3 Eclipse ..... 18

    1.8 Lenguajes de modelado ..... 19

        1.8.1 Lenguaje Unificado de Modelado ..... 19

    1.9 Framework Qt ..... 19

    1.10 Lenguajes de programación ..... 20

1.10.1	C++ .....	20
1.10.2	Python.....	20
1.11	Editores de escenarios .....	21
1.11.1	Cryengine Sandbox.....	21
1.11.2	Warcraft III World Editor .....	22
1.11.3	Unreal Level Editor.....	23
1.12	Características de los formatos de almacenamiento 3D .....	23
1.12.1	Formato binario.....	24
1.12.2	Formato ASCII .....	26
1.13	Consideraciones parciales.....	27
<b>Capítulo 2: Propuesta de solución.....</b>		<b>29</b>
	Introducción al capítulo.....	29
2.1	Desarrollo de la herramienta CEV .....	29
2.1.1	Especificación de los requisitos de software .....	29
2.1.2	Definición de los casos de uso .....	35
2.1.3	Diagrama de casos de uso.....	36
2.1.4	Modelos conceptuales de la herramienta .....	37
2.1.5	Vista lógica.....	38
2.1.6	Diagramas y descripción de las clases.....	40
2.1.7	Interfaz de usuario .....	43
2.1.8	Modelo de datos.....	46
2.1.9	Modelo de prueba .....	47
2.2	Propuesta de modificaciones al proceso de desarrollo de un paseo virtual .....	47
2.3	Consideraciones parciales.....	50
<b>Capítulo 3: Resultados y discusión .....</b>		<b>51</b>

Introducción.....	51
3.1 Centros Expositivos Virtuales.....	51
3.2 Análisis de la herramienta CEV.....	52
3.3 Comparación entre algunas de las herramientas estudiadas y CEV .....	53
3.4 CEV desde la perspectiva del editor de centros expositivos.....	55
3.4.1 Cuestionario.....	55
3.4.2 Análisis de los resultados de la encuesta.....	56
3.5 CEV desde la perspectiva del informático .....	57
3.6 Consideraciones parciales .....	59
<b>Conclusiones.....</b>	<b>60</b>
<b>Recomendaciones.....</b>	<b>61</b>
<b>Referencias bibliográficas.....</b>	<b>62</b>
<b>Anexo 1: Muestra de los casos de pruebas diseñados .....</b>	<b>68</b>
<b>Anexo 2: Glosario de términos.....</b>	<b>70</b>

## ÍNDICE DE TABLAS

Tabla 1. Descripción de los actores que interactúan con la herramienta. ....	35
Tabla 2 Comparación de algunas herramientas. ....	54
Tabla 3. Encuesta a estudiantes. ....	55
Tabla 4. Resultados de encuesta por objetivos. ....	56
Tabla 5. Diseño de caso de prueba. CU Gestionar modelos del sistema. ....	69
Tabla 6. Diseño de caso de prueba. CU Inicializar Datos. ....	69

## ÍNDICE DE FIGURAS

Figura 1. Diagramas de casos de usos arquitectónicamente significativos. ....	36
Figura 2. Modelo conceptual del módulo Editor. ....	38
Figura 3. Modelo conceptual del módulo Visualizador. ....	38
Figura 4. Vista Lógica de la herramienta. ....	40
Figura 5. Diagrama de clases del diseño del módulo Editor. ....	41
Figura 6. Diagrama de clases del diseño del módulo Visualizador. ....	42
Figura 7. Interfaz Gráfica del Editor. ....	44
Figura 8. Interfaz Gráfica del Visualizador. ....	45
Figura 9. Interfaz Gráfica del Visualizador. ....	46
Figura 10. Formato CEV. ....	47
Figura 11. Representación del proceso de implementación de un PVI en el CEDIN. .	48
Figura 12. Propuesta de modificación al proceso de implementación de un PVI. ....	49
Figura 13. Gráfica de los resultados de la encuesta por objetivos. ....	57
Figura 14. Análisis de las modificaciones a los conceptos y relaciones entre ellos. ....	58

## Introducción

Desde los primeros tiempos la humanidad ha enfrentado dificultades y en la búsqueda continua de soluciones se ha conformado un amplio arsenal de técnicas y tecnologías que forman parte imprescindible de la vida moderna. Con el acelerado desarrollo de las tecnologías de la computación y los avances en campos como el de la realidad virtual (RV), que según Diego Levis es una *“base de datos interactiva capaz de crear una simulación que implique a todos los sentidos, generada por un ordenador, explorable, visualizable y manipulable en tiempo real bajo la forma de imágenes y sonidos digitales, dando la sensación de presencia en el entorno informático”* [1], el humano de hoy ha roto barreras en las que el tiempo y las posiciones geográficas ya no son limitantes para el conocimiento y la recreación.

Una de las áreas de mayor impacto en este campo son los videojuegos, consolidados como el principal impulsor del interés de los usuarios y desarrolladores hacia la realidad virtual, y las primeras en incorporar escenarios virtuales<sup>1</sup> (EV). Algunos videojuegos han logrado reproducir escenarios reales con alta fidelidad, como es el caso del Assasin's Creed II<sup>2</sup>, en el cual el jugador se mueve por la Venecia de 1476.

Al inicio, los escenarios eran suministrados únicamente por las empresas desarrolladoras hasta la aparición, a finales de los años 80 e inicio de los años 90 del siglo pasado, de herramientas que permitían al jugador personalizar un escenario a su gusto y jugar sobre él. Los editores se hicieron populares y la mayoría de las grandes compañías desarrolladoras de videojuegos los adoptaron como estándar. Desde ese momento, el desarrollo de editores propios y especializados se convirtió en una tendencia para las empresas que requieren de la creación y la edición de EV.

Los EV no se limitaron al uso exclusivo en videojuegos y hoy se desarrollan entornos con disímiles aplicaciones que recrean lugares de interés histórico, cultural o económico. Algunos de los ejemplos son la enciclopedia Microsoft Encarta, que ofrece un recorrido por la Atenas del año 432 A.C.; el sitio web *www.vatican.va*, desde donde se puede acceder a recorridos virtuales por las Basílicas y Capillas Papales de la

---

<sup>1</sup> Simulación, mediante la realidad virtual, de un escenario real o ficticio.

<sup>2</sup> Videojuego de acción y aventura, desarrollado por la empresa francesa UbisoftUbisoft Entertainment y disponible en el mercado desde el 17 de noviembre de 2009.

Santa Sede y el Museo de Historia Natural de los E.E.U.U Smithsonian National Museum, que permite recorrer sus salas a través de su sitio web: [www.mnh.si.edu](http://www.mnh.si.edu).

Esta forma de uso de la realidad virtual, que puede encontrarse en disímiles formatos y plataformas, se conoce como Paseos Virtuales (PV). Existen numerosos tipos de paseos virtuales y diferentes técnicas y tecnologías para realizarlos. Además, se pueden clasificar de distintas formas y existen muchos criterios emitidos por varios autores.

Se pueden recrear virtualmente escenarios del mundo real o productos de la imaginación, con lo que se hace posible que personas que se encuentren en cualquier lugar del planeta puedan, al menos virtualmente, “visitar” dicho sitio e interactuar con los objetos del entorno, con lo que se aumenta considerablemente la sensación de realidad y el volumen de la información gráfica disponible.

Como manifestación cultural se encuentra que museos, galerías de arte, acuarios y zoológicos, entre otros centros expositivos permanentes o transitorios, utilizan este tipo de tecnologías para facilitar la difusión de sus contenidos mediante visitas virtuales. En internet se pueden encontrar servicios de visitas en diversas formas, entre los que se encuentran:

- Páginas web con imágenes e información, como la del Palacio de los Capitanes Generales y Templete [2].
- Videos digitales soportados por la web, como Santiago de Cuba - Virtual Tour [3].
- Aplicaciones realizadas en Flash<sup>3</sup>, como la del Acuario Nacional de Cuba, representado en vista aérea y acompañada de información textual de las áreas que lo componen [4].
- Vistas 360, como las que pueden visualizarse del Castillo de los Tres Reyes del Morro, Varadero Cuba, Cayo Coco, Playa Pesquero o Barber Shop Havana Cuba [5].
- Paseos virtuales a diferentes zoológicos como por ejemplo el de Barranquilla y el de San Juan de Aragón en México, o el recorrido virtual de 360 al zoológico de Saint Louis.

---

<sup>3</sup> Herramienta producida por la compañía Adobe Systems Incorporated para realizar animaciones que pueden visualizarse en un reproductor incorporado a un navegador web.

Estos ejemplos muestran de forma general una aplicación web con bajo realismo en cuanto a la cantidad de información que percibe el usuario, limitada por la vista única que expone una imagen o los ángulos definidos para las escenas en un video. En todos los casos el usuario, acostumbrado a un mundo tridimensional, percibe fácilmente el poco realismo del escenario.

Al realizar una búsqueda de las compañías extranjeras desarrolladoras de paseos o visitas virtuales se encontraron varias y entre ellas: Fourth Dimension Solutions<sup>4</sup>, Paseo VIRTUAL<sup>5</sup>, Jasvir Business Data-Systems<sup>6</sup> y muchas otras que pueden encontrarse en sitios como <http://360vr.top1001sites.com/>. Todas ellas realizan visitas interactivas sin elementos en tres dimensiones (3D) utilizando imágenes o videos. Estas empresas emplean variadas herramientas, tal es el caso de: Panotour Pro, 360 Flash Tour, EveryScape y otras. Sin embargo, en la búsqueda realizada no se encontraron herramientas especializadas para la realización de paseos virtuales interactivos con modelos tridimensionales especializadas en crear centros expositivos.

Para obtener de forma tradicional un paseo virtual interactivo (PVI), se transita por un proceso que incluye las siguientes actividades: el Análisis y Diseño de Software, Implementación, Diseño Gráfico y las Pruebas. Cada una de ellas requiere de tiempo para realizarla y de especialistas que pueden intervenir varias veces en el proceso.

Al terminar un PVI, se entrega al cliente como una solución integral. Esto puede convertirse en un inconveniente ya que con frecuencia tienen lugar situaciones en las que es necesario volver a realizar gran parte o la totalidad del proceso de desarrollo. Algunas de estas situaciones pueden ser:

- La aplicación necesita mantenimiento o rectificación de errores.
- La aplicación necesita ser modificada por no satisfacer parcial o totalmente las expectativas del cliente.
- El uso del paseo virtual es experimental y deben probarse varias soluciones.
- Necesidad de exhibir muestras diferentes, sobre todo en los grandes centros expositivos.

---

<sup>4</sup> Disponible en: <http://www.fourthdimensionsolutions.com/interactive-3d-walkthrough.html>

<sup>5</sup> Disponible en: <http://www.paseosvirtuales.com>

<sup>6</sup> Disponible en: <http://www.jasvirsystems.com/da3dmen03.html>

Las situaciones anteriores se tornan problemáticas por diferentes factores como la poca experiencia cubana en la realización de paseos virtuales con elementos en tres dimensiones. A esto se le adiciona la complejidad del proceso de desarrollo y los altos costos de implementación de un producto de este tipo. Estos inconvenientes limitan el uso de la realidad virtual a entidades con suficientes recursos para sufragar los costos de adquisición y mantención. Aplicar la realidad virtual puede ser útil en diferentes funciones como la actividad publicitaria, para contribuir a la conservación y divulgación del patrimonio nacional, entre otras. Un aspecto importante lo constituye el hecho de que no está previsto que los clientes modifiquen de alguna manera sus paseos virtuales, lo cual trae consigo que en caso de necesitar modificaciones a la solución, se tenga que acudir al equipo de desarrollo nuevamente, y esto a su vez representa una desventaja si el cliente en cuestión depende de la variedad de exposiciones para el buen funcionamiento de su negocio. Por tanto, si los clientes pudieran modificar de forma controlada sus paseos virtuales, podría contribuirse a potenciar el desarrollo de este tipo de aplicaciones, generalizando su uso e incrementando la divulgación de los contenidos en los centros de exposición. Sin embargo, esta aspiración se ve frenada por los elevados requerimientos para su propio desarrollo, pues se necesita tener conocimientos de informática en el campo específico de la realidad virtual para ejecutar este tipo de tareas, que como norma, no poseen los usuarios finales.

Hasta el momento, la experiencia cubana en el desarrollo de paseos virtuales se caracteriza principalmente por la baja interactividad del usuario con los elementos que conforman el escenario. Desde el año 2002, se creó el Grupo de Desarrollo de Paseos Virtuales (GDPV) en la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI), que actualmente forma parte del Centro de Informática Industrial (CEDIN). Este grupo ha mantenido la producción de paseos virtuales y el estudio de las herramientas y técnicas adecuadas para su desarrollo durante varios años. Consecuentemente, ha podido identificar ventajas y desventajas en este tipo de soluciones, sobre todo en la etapa de implementación, donde se requieren altos conocimientos de programación.

En los años de trabajo del GDPV se han desarrollado varias aplicaciones entre las que se encuentran:

- Paseo virtual "Maqueta UCI 2003", Suiza, 2003.

- Paseo virtual "Módulo 1 de la IP<sup>7</sup>", realizado en el 2005.
- Paseo virtual "Docente 1 de la UCI", realizado en junio del 2006.
- Paseo virtual "Prado de las esculturas", realizado en el 2009.
- Paseo virtual "Salón virtual de la industria informática cubana 2011", SVIIC 2010<sup>8</sup>, premiado en la Feria de Soluciones Informáticas (FESI 2011) como "producto más popular".

Partiendo de la situación antes expuesta se plantea el siguiente **problema de investigación**:

¿Cómo facilitar a personas sin habilidades en técnicas de programación y diseño gráfico, la creación de paseos virtuales interactivos con elementos en tres dimensiones?

A partir del problema formulado se define como **objeto de investigación** los paseos virtuales interactivos con elementos en tres dimensiones. Para dar solución al problema planteado se define el siguiente **objetivo general**: desarrollar una herramienta informática que permita la creación, edición y visualización de paseos virtuales interactivos con elementos en tres dimensiones, a usuarios inexpertos en el uso de herramientas de diseño gráfico y técnicas de programación. Este objetivo general se subdivide en cuatro **objetivos específicos**:

- Concebir una herramienta para la edición del escenario virtual 3D.
- Garantizar la visualización de los paseos virtuales y la navegación sencilla por ellos.
- Evaluar cualitativamente la influencia de la introducción de la herramienta propuesta dentro del proceso de desarrollo de un PV.
- Reducir los errores de codificación en la implementación de un PV.

Teniendo claros los objetivos la investigación se enmarca en el siguiente **Campo de acción**: las aplicaciones de escritorio para la creación, edición y visualización de paseos virtuales interactivos con elementos en tres dimensiones.

---

<sup>7</sup> Infraestructura Productiva. Edificio en la UCI creado para la Dirección General de Producción.

<sup>8</sup> SVIIC ® en el Centro Nacional de Derecho de Autor de Cuba (CENDA).

## **Contribución intelectual del maestrante**

- Concepción de una entidad informática materializada en una herramienta que permite a personas con escasos conocimientos en programación gráfica, manejo de herramientas de diseño y realidad virtual, realizar paseos virtuales interactivos con elementos en tres dimensiones.
- Creación de una base documental para facilitar la comprensión de la herramienta CEV y describir cómo realizar paseos virtuales interactivos con elementos en tres dimensiones.
- Diseño del formato de fichero \*.cev” a partir de algunos de los formatos más conocidos y la incorporación de nuevas características para responder a las necesidades específicas de la aplicación.
- Concepción de una funcionalidad para la creación y administración de paquetes de visualización.

## **Estructura del documento**

El documento está estructurado en tres capítulos. En el primero se construye un marco teórico que permite la familiarización con conceptos básicos asociados al objeto de estudio. Además, se caracterizan las principales metodologías, lenguajes, herramientas de desarrollo de software, paseos virtuales y formatos de escenas 3D. Con este capítulo se dejan sentadas las bases teóricas para una correcta comprensión de las decisiones tomadas a lo largo del trabajo. En el segundo se dan conocer cada una de las actividades realizadas para lograr la creación de la nueva herramienta y se representan los cambios propuestos a la etapa de implementación del proceso de desarrollo de paseos virtuales usado actualmente en el CEDIN. En el tercer y último capítulo se realizan consideraciones generales sobre la concepción de la herramienta propuesta y se hace una breve comparación con sus predecesores y algunas de las herramientas estudiadas. Asimismo se analizan los comportamientos de los posibles usuarios frente a la herramienta y se describe cualitativamente cómo afecta su uso a la etapa de implementación de paseos virtuales dentro del proceso de desarrollo usado por el GDPV. Por último, se define una nueva entidad informática que se infiere de los resultados alcanzados en el desarrollo de la herramienta propuesta.

# Capítulo 1: Fundamentación teórica

## Introducción

En este capítulo se caracterizan la realidad virtual, los paseos virtuales, los componentes fundamentales que intervienen en la etapa de implementación dentro del proceso de desarrollo de un paseo virtual interactivo con elementos en tres dimensiones y algunos de los formatos de escenas 3D más usados. Así mismo, se describen las principales herramientas, metodologías y lenguajes de programación empleados para la creación de paseos virtuales de forma general. Se agrega además un conjunto de criterios valorativos que fundamentan la selección de las tecnologías a emplear para el logro del objetivo propuesto.

### 1.1 Realidad virtual

Son numerosas las definiciones atribuidas al término realidad virtual, quizás tantas como el número de autores que se han acercado al tema. Al realizar una búsqueda del significado de dicha expresión se pueden encontrar varias similitudes pero al mismo tiempo, características específicas que permiten apropiarse del concepto desde una perspectiva más amplia. A continuación son citadas algunas definiciones:

*“Una simulación por ordenador de un sistema real o imaginario que permite a un usuario realizar operaciones en el sistema simulado y muestra los efectos en tiempo real” [6].*

*“Una forma de interacción persona-ordenador en el que un entorno real o imaginario es simulado y los usuarios pueden interactuar y manipular ese mundo. Los usuarios viajan en el mundo simulado moviéndose hacia donde quiera estar, e interactuar y manipular objetos simulados... [7]”*

Basado en los criterios anteriores el autor considera que un sistema de realidad virtual debe ser capaz, en determinado grado, de estimular y al mismo tiempo confundir los sentidos de las personas a las que va dirigido. Teniendo en cuenta esto, se define que para la presente investigación el concepto de realidad virtual más adecuado es el brindado por Dr. M. Pauline Baker.

## Capítulo 1: Fundamentación teórica

---

Según el nivel de introducción del usuario en el ambiente virtual, se pueden encontrar varias clasificaciones para la realidad virtual tales como: inmersiva, semi-inmersiva o no inmersiva. Teniendo en cuenta que el Diccionario de la Lengua Española en su vigésima segunda edición muestra que el significado de la palabra inmersión es la “acción de introducir o introducirse plenamente alguien en un ambiente determinado”, que para el caso que se aborda se trata de un ambiente virtual. El autor de este trabajo reconoce tres clasificaciones para sistemas de RV:

- Inmersivos: *donde el usuario se siente dentro del mundo virtual que está explorando. Este tipo de sistemas utiliza diferentes dispositivos denominados accesorios, como pueden ser guantes, trajes especiales, visores o cascos, estos últimos le permiten al usuario visualizar los mundos a través de ellos.*
- Semi inmersivos: *también denominados inmersivos de proyección, se caracterizan por ser cuatro pantallas en forma de cubo (tres pantallas forman las paredes y una el piso), las cuales rodean al observador, el usuario usa lentes y un dispositivo de seguimiento del movimiento de la cabeza, de esta manera al moverse el usuario las proyecciones perspectivas son calculadas por el motor de RV para cada pared y se despliegan en proyectores que están conectados a la computadora.*
- No inmersivos: *son aquellos donde el monitor es la ventana hacia el mundo virtual y la interacción es por medio del teclado, micrófono, mouse o joystick<sup>9</sup>, este tipo de sistemas se usa para visualizaciones científicas, entretenimiento y aunque no ofrecen una total inmersión son una buena alternativa de bajo costo [8].*

La presente investigación se apoya para desarrollar el objetivo propuesto en un sistema de RV no inmersivo o de escritorio, mediante el cual se logra la representación digital de un entorno tridimensional donde el usuario pueda interactuar intuitivamente y en tiempo real<sup>10</sup> con los objetos que encuentre dentro de escenario visible.

---

<sup>9</sup> Referido al dispositivo de entrada en forma de palanca o mando para jugar.

<sup>10</sup> Tiempo Real se refiere a la representación en pantalla de las imágenes generadas con una frecuencia no menor a 30 imágenes por segundo.

## 1.2 Paseos virtuales

Los paseos virtuales son aplicaciones con un amplio mercado en todo el mundo. Consisten en la simulación de un entorno empleando tecnología de realidad virtual y pueden crearse para representar áreas existentes del mundo que nos rodea con determinado grado de realismo visual. Permiten además el enriquecimiento de la escena e incluso la representación de áreas totalmente imaginarias.

La creación de un paseo virtual trae consigo muchas bondades, sobre todo la posibilidad de conocer lugares sin necesidad de desplazarse más allá de su escritorio, lo cual resulta muy tentativo y útil. Reproducir en la memoria de una computadora el recorrido por cavernas prehistóricas cuya visita física pondría en riesgo su conservación, mostrar productos en tres dimensiones para su comercialización o darle publicidad a locales con fines diversos, están entre las muchas aplicaciones de los paseos virtuales.

En la actualidad, una gran variedad de industrias utilizan esta tecnología para ayudar a comercializar sus servicios y productos. En los últimos años, la popularidad de estas aplicaciones se ha incrementado considerablemente. En una búsqueda sencilla en Internet del término “paseo virtual” con algunos de los principales buscadores como Google, Altavista, Aol Search, ASK, Yahoo y Bing, se encuentran como promedio más de 25 millones de páginas, lo cual brinda una medida del interés general que despierta este tema en el mundo.

## 1.3 Metodologías de desarrollo de software

Las metodologías se usan para planificar, estructurar y controlar el proceso de desarrollo de software. Se han definido en función de las dimensiones de los proyectos en los que se aplicarán y de sus variables, lo que ha conllevado a un desarrollo constante y fructífero de las mismas. Actualmente existen varias, entre las que se pueden citar el Proceso Unificado de Desarrollo (RUP), Scrum, Common Diagnostic Model (CDM) y Programación Extrema (XP). Esta última se considera una de las más populares dentro de las llamadas metodologías ágiles y aunque faculta a los desarrolladores para responder con seguridad a las necesidades cambiantes de los clientes [9], el volumen de información que describe es menor que el generado por las

metodologías pesadas. El autor considera que para este trabajo no es recomendable usar XP debido a que no permite conformar una documentación tan detallada como otras metodologías. Una de las metodologías pesadas más reconocidas y utilizadas a nivel mundial es RUP.

### 1.3.1 RUP

Es una metodología de desarrollo de software con un conjunto de actividades dirigidas a transformar los requerimientos del cliente en una solución. Aumenta la productividad del equipo de desarrollo y permite a cada miembro compartir un lenguaje común (UML<sup>11</sup>), un proceso y una vista de cómo obtener el producto. RUP describe un proceso configurable, soportado por herramientas que automatizan gran parte del proceso. Además, genera una amplia y detallada documentación de los elementos que la componen y en su modelo define como sus principales elementos a los trabajadores, actividades y artefactos.

Los principios de RUP son: adaptar el proceso, equilibrar las prioridades de los interesados, colaborar con otros equipos, demostrar el valor de forma iterativa, elevar el nivel de abstracción y centrarse continuamente en la calidad. Además, el ciclo de vida de RUP se caracteriza por ser: dirigido por casos de uso, centrado en la arquitectura e Interactivo e incremental [10].

Un aspecto a tener en cuenta para la selección de la metodología que guiará el presente trabajo, es la necesidad de generar una amplia y detallada documentación que garantice a los especialistas la comprensión de la idea general, el diseño y particularmente el código. Si se comprende adecuadamente el diseño y el código se facilita el trabajo de ampliación y actualización de la herramienta.

## 1.4 Implementación de un paseo virtual

La implementación es una de las principales etapas dentro de un proceso de desarrollo de software, tiene como propósito la producción eficaz y eficiente de un producto que reúna los requisitos del cliente. Para obtener un paseo virtual interactivo con elementos

---

<sup>11</sup> Lenguaje Unificado de Modelado.

en tres dimensiones, se realizan las actividades comunes a todo ciclo de desarrollo de un producto informático. El autor de esta investigación estima que el diseño gráfico y la programación de las funcionalidades están incluidas dentro de la etapa de implementación y son actividades realizadas por especialistas con habilidades en técnicas de programación gráfica, optimización de algoritmos y modelado 3D. Los principales conceptos involucrados en la implementación del “proceso de desarrollo de paseos virtuales” (PDPV) [11] usado en el CEDIN se describen a continuación:

- **Diseñador gráfico:** Persona que crea los modelos gráficos y edita los escenarios.
- **Modelos gráficos:** Representación gráfica de objetos y planos mediante el uso de aplicaciones informáticas.
- **Probador:** Persona que realiza las pruebas a los modelos, escenarios, componentes y al sistema.
- **Desarrollador:** Persona que realiza la codificación de los componentes del sistema.
- **Herramientas de diseño gráfico:** Aplicaciones informáticas utilizadas para diseñar escenarios.
- **Herramientas de pruebas:** Aplicaciones utilizadas para apoyar la realización de las pruebas al sistema.
- **IDEs:** Programa informático compuesto por un editor de código, un compilador<sup>12</sup>, un depurador y un constructor de interfaz gráfica que permite al desarrollador implementar un producto.
- **Motor de juego:** Conjunto de programas para manipular gráficos 2D<sup>13</sup> y 3D. Está compuesto por un simulador de fenómenos físicos, que incluye un detector de colisiones. Además, integra sonidos y animaciones, algoritmos de inteligencia artificial, bibliotecas para la transmisión de paquetes de red y componentes de administración de memoria. Con frecuencia incluye también un escenario gráfico virtual.

---

<sup>12</sup> Programa que convierte el lenguaje de programación empleado por el usuario en lenguaje propio del computador.

<sup>13</sup> Abreviatura usada para: bidimensional.

- **Componentes:** Programas o ficheros de código.

A pesar de la experiencia que posee el GDPV del CEDIN, derivada de varios años de trabajo en la realización de paseos virtuales interactivos con elementos en tres dimensiones y de haberse aplicado un perfeccionamiento al proceso de desarrollo, todavía se atrasan las entregas de los productos. Adicionalmente, a medida que se incrementan los compromisos productivos, la posibilidad de contar con personal especializado es cada día menor.

## 1.5 Herramientas para el diseño de escenarios virtuales en 3D

Para el modelado de EV se utilizan varias herramientas de diseño 3D, en dependencia del fin que se persigue. Por ejemplo, en el mercado de los videojuegos es muy popular el Autodesk 3ds Studio Max (3dsMax) mientras que Autodesk Maya (Maya) es de la preferencia de los animadores y productores de cine en general, y por último Blender, una herramienta libre que está ganando seguidores cada día la cual se perfila con un futuro prometedor. El estudio de estas herramientas permite identificar características comunes que podrían ser útiles y fáciles de identificar por una persona que desee diseñar un modelo o escenario en tres dimensiones.

Estas herramientas se han desarrollado con un propósito general, lo que hace que posean grandes paneles de herramientas que provocan que un usuario inexperto<sup>14</sup> se “pierda” en busca de las funcionalidades que realmente necesita, lo que se dificulta el trabajo de edición de escena. En todos los casos, los artistas digitales deben ser especialistas y tener habilidades en el modelado 3D. Un modelo con estas características en manos de un inexperto puede terminar deformado, por lo que el trabajo con este tipo de herramientas se asume exclusivamente por expertos en diseño 3D. En el siguiente epígrafe se detallan las características fundamentales de algunas de las herramientas estudiadas.

---

<sup>14</sup> En documento este término se utiliza para clasificar a personas sin experiencia en el uso de herramientas de diseño gráfico o en técnicas de programación.

### 1.5.1 Autodesk 3D Studio Max

Se anunció oficialmente en SIGGRAPH<sup>15</sup> 1995, y estuvo en manos de los usuarios en abril de 1996. Desde entonces se han entregado varias versiones del producto. Las herramientas de modelado, animación, render<sup>16</sup> y composición del Autodesk 3ds Studio Max están diseñadas para desarrolladores de videojuegos, artistas de efectos visuales, aunque también se usa por productores de televisión y cine. Resalta entre sus características que posee uno de los paneles de herramientas más ricos en contenido de los que se dispone actualmente en la industria. Autodesk 3ds Studio Max facilita la creación de objetos orgánicos y paramétricos con características de modelado de polígonos, spline (línea) y basado en nurbs<sup>17</sup>. Autodesk 3ds Studio Max cuenta con más de 100 herramientas de diseño para el modelado poligonal organizadas en un paquete denominado Graphite. También facilita el control preciso del número de caras y vértices en los objetos con la tecnología ProOptimizer que permite reducir la complejidad de una selección de polígonos hasta un 75% sin pérdida de detalle [12].

### 1.5.2 Autodesk Maya

Es otro producto desarrollado por Autodesk, pero más orientado a la producción audiovisual. La diferencia comienza en la interfaz visual, que está diseñada para esos fines y continúa en las numerosas herramientas que, aunque comparten el mismo productor, tienen equipos de desarrollo dedicados especialmente a mejorar las características del programa en cada nueva versión. A continuación se detallan algunas de sus características:

---

<sup>15</sup> Conferencia organizada por el grupo del mismo nombre, el cual fue fundado en 1974 y reúne a profesionales con intereses en infografía o computación gráfica.

<sup>16</sup> Creación y representación de los modelos gráficos en una escena como una imagen 2D.

<sup>17</sup> Acrónimo inglés de la expresión *Non Uniform Rational B-splines*. Es un modelo matemático utilizado en la computación gráfica para generar y representar curvas y superficies.

**Interfaz de usuario:** Está disponible para los sistemas operativos Mac OS X, Windows de 32 y 64 bits<sup>18</sup> y Linux de 64 bits, con una interfaz que ofrece elementos anclables<sup>19</sup> y editores flexibles.

**Modelado 3D:** Ofrece un enriquecido conjunto de herramientas para el modelado 3D, que incluye nurbs, superficies de subdivisión y polígonos.

**Animación general:** Entrega una amplia gama de herramientas especializadas para la animación por keyframes<sup>20</sup>, procedural y a base de scripts<sup>21</sup> [13].

### 1.5.3 Blender

Es un proyecto de la Fundación Blender, alternativa de software libre con muchos seguidores [14]. A pesar de ser una herramienta relativamente joven, se perfila como una buena opción. Su interfaz posee una disposición de ventanas flexible y configurable. Cuenta además con un editor de texto interno para anotaciones y edición de scripts de Python<sup>22</sup>. Se puede destacar que consta de una interfaz que se mantiene robusta para todas las plataformas [15]. Brinda varias posibilidades para el modelado de objetos y la edición de escenas, entre las que se encuentran:

- Modelado multi-resolución con pinceles procedurales de mapas 2D y 3D (Paint, Smooth, Pinch, Inflate, Grab) con soporte para simetrías.
- Modificadores de deformación de mallas (Lattice, Curve, Armature y Displace).
- Modificador Mirror con clipping para los vértices del centro y eliminación automática de caras interiores.

Esta herramienta posee un motor de juegos propio que permite la creación de pequeñas aplicaciones interactivas con ayuda del editor de texto incluido. Sus

---

<sup>18</sup> La unidad más pequeña de información en notación binaria.

<sup>19</sup> Elemento de la interfaz que puede colocarse (anclarse) en la posición que desee el usuario.

<sup>20</sup> Fotograma clave, término utilizado en la animación de personajes.

<sup>21</sup> Programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

<sup>22</sup> Lenguaje de programación interpretado de alto nivel, multiparadigma, usa tipado dinámico, es fuertemente tipado y multiplataforma.

principales desventajas son que su motor de juegos no está suficientemente desarrollado para soportar grandes escenarios y que al igual que otros motores de juegos requiere que el usuario posea algún conocimiento en lenguajes de programación.

### 1.6 Motores de juegos

El motor de juego (game engine) se utiliza para simplificar el desarrollo de la aplicación en términos de implementación. Ofrece un conjunto de herramientas de desarrollo, además de componentes reutilizables agrupados en subsistemas. Los subsistemas más comunes son: procesamiento de entradas, motor gráfico, animación, audio, comportamiento e inteligencia artificial y conectividad, entre otros. Independientemente de las facilidades que brindan los motores de juego, en todos los casos se requiere de conocimientos en algún lenguaje de programación, por lo que el desarrollo de aplicaciones sobre ellos adiciona cierto grado de complejidad, aún más cuando se trata de aplicaciones con gráficos en tres dimensiones. Estas características representan un obstáculo para personas poco diestras en las tareas de implementación, por lo que les sería difícil implementar un paseo virtual interactivo con elementos en tres dimensiones.

A continuación se describen algunas características de dos de los motores de juegos más empleados. También se hace referencia al SceneToolKit, herramienta nacional desarrollada por el CEDIN.

#### 1.6.1 G3D

Graphics Three Dimensional Engine (G3D) es un motor de juegos multiplataforma escrito en lenguaje C++, con una amplia documentación y numerosos programas de ejemplo. Posee un diseño orientado a objetos y soporta varias extensiones para la manipulación de imágenes (\*.jpg, \*.tga y \*.png) y modelos generados por otras aplicaciones (\*.3ds, \*.ifs, \*.md2, \*.bsp, \*.ply2, \*.off). Es interesante destacar que integra

dentro de sí el trabajo con shaders<sup>23</sup> y diversas técnicas de representación como Shadow Volumes, que aportan mayor realismo al entorno virtual [16].

## 1.6.2 Ogre

Es un motor de juegos de código abierto, multiplataforma escrito en lenguaje C++. Cuenta con una gran comunidad internacional y una amplia documentación, además de permitir el acople de nuevos complementos que incrementan su flexibilidad [17]. Entre sus principales características se encuentran:

### **Texturas:**

- TextureMapping. Cálculo automático de Mip-Maps.
- Texturas multinivel en un solo paso.
- Texturas animadas.
- Cálculo procedural de coordenadas de textura.

### **Iluminación:**

- Luces puntuales, direccionales y de foco.
- Nieblas.
- Luces dinámicas.
- Radiosidad pre-calculada sobre los mapas de luces.

### **Modelado:**

- Soporte para mallas con bajos polígonos.
- Superficies de Bezier.
- Grafo jerárquico de escena.
- Edición externa mediante Autodesk 3ds Studio Max.

---

<sup>23</sup> Procedimiento que permite al artista o programador especificar el render de un vértice (vertex) o de un pixel para manipular la iluminación, el sombreado o crear efectos gráficos como animaciones, efectos de partículas, etc.

- Modelado de terreno.
- Importación de niveles del juego Quake3.

### 1.6.3 SceneToolKit (STK)

Es un paquete de herramientas para desarrollar sistemas de realidad virtual, que actualmente se encuentra registrada y responde a las necesidades más significativas de los proyectos desarrollados en el CEDIN [18]. La SceneToolkit<sup>24</sup> brinda soporte para las Application Programming Interface (APIs) OpenGL y DirectX, por lo que es compatible con los sistemas operativos Windows y Linux, sobre los que facilita la interacción con ventanas y periféricos (ratón, teclado, joystick). Su arquitectura dividida por módulos permite la integración y acoplamiento de nuevos módulos sin necesidad de una reestructuración a gran escala del sistema completo [19].

## 1.7 Entornos de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Normalmente dispone de un editor de texto, un compilador, un intérprete, herramientas para la automatización, un depurador y opcionalmente un sistema de control de versiones [20].

Independientemente de las facilidades que brinda un IDE, se requiere de conocimientos en algún lenguaje de programación, lo que adiciona complejidad para la obtención de un paseo virtual. El estudio, selección y uso de ellos se hace imprescindible para la creación de software que puedan utilizarse posteriormente como herramientas de desarrollo.

### 1.7.1 Microsoft Visual Studio

Es un IDE privativo de la compañía Microsoft que ofrece un conjunto completo de herramientas de desarrollo para la generación de aplicaciones ASP.NET, servicios

---

<sup>24</sup> Parte importante de la información referente a la SceneToolkit es tomada del expediente del proyecto.

web, aplicaciones de escritorio y para dispositivos móviles. Sobre Microsoft Visual Studio se puede programar en los lenguajes Visual Basic, Visual C++, Visual C# y Visual J#, entre otros, lo que permite compartir herramientas y facilita la creación de soluciones en varios lenguajes [21]. Microsoft ofrece hasta el momento una amplia documentación sobre este IDE y garantiza el soporte tecnológico.

### 1.7.2 Qt Creator

Se distribuye bajo tres tipos de licencias: Qt Commercial Developer License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: Linux, Mac OSX; Windows, Windows CE, Symbian y Maemo. Su objetivo principal es el desarrollo de aplicaciones en lenguaje C++ de manera sencilla y rápida. Basado en el framework Qt posee las siguientes características:

- Editor de código C++ y Javascript.
- Diseñador integrado de interfaz de usuario.
- Herramientas de administración y construcción de proyectos.
- Soporte para el control de versiones.
- Simulador para interfaces de dispositivos móviles.
- Soporte para dispositivos móviles y aplicaciones de escritorio [22].

### 1.7.3 Eclipse

Es un IDE de código abierto y multiplataforma. Aprovechando su estructura de complementos se pueden adicionar una amplia variedad de lenguajes de programación, no obstante la versión “Indigo” cuenta en su núcleo con soporte para Java, C/C++, HTML, CSS, entre otros. Brinda funcionalidades avanzadas que incluyen un editor con completamiento de código y resaltador de sintaxis, un motor de búsquedas y un generador de ficheros. No incluye compilador ni debugger (depurador), por lo que es necesario instalarlos de forma independiente [23].

## 1.8 Lenguajes de modelado

Desde el momento en que se concibe un software es necesario establecer una forma transparente de comunicación entre los involucrados con el producto, de forma tal que el equipo obtenga información útil y confiable durante su proceso de desarrollo. Los lenguajes de modelado permiten representar un sistema y describir las actividades que se deben realizar para construir un determinado producto. La presente investigación utiliza el UML para estandarizar y enriquecer la documentación.

### 1.8.1 Lenguaje Unificado de Modelado

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Fue creado por tres de las personalidades más importantes de la ingeniería del software James Rumbaugh, Ivar Jacobson y Grady Booch para reemplazar a modelos como OMT, Booch y Objectory [24]. Está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema, que incluye aspectos conceptuales como procesos de negocios, funciones del sistema o aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Se puede aplicar de varias formas para dar soporte a una metodología de desarrollo de software como el Proceso Unificado Racional, pero no especifica en sí mismo qué metodología o proceso usar [25].

## 1.9 Framework Qt

Qt es un framework (marco de trabajo) multiplataforma para desarrollar interfaces de usuario (UIs, por sus siglas en inglés). A través de él, se pueden escribir aplicaciones web y desplegarlas en entornos de escritorio, en sistemas operativos integrados y como parte de sistemas móviles, sin tener que reescribir el código fuente. Incluye una biblioteca de clases en lenguaje C++ que permite la portabilidad entre los diferentes sistemas operativos. Contiene también un conjunto de herramientas para facilitar el

trabajo de los desarrolladores, se integra al IDE Qt Creator y se destaca por un alto rendimiento cuando se ejecutan aplicaciones en tiempo real [26]. El Framework Qt facilita al programador implementar los componentes visuales, que permiten la interacción de los usuarios con la herramienta que se propone.

### 1.10 Lenguajes de programación

El dominio de los lenguajes de programación es una de las habilidades que se requieren para desarrollar un PVI con elementos en 3D. Para seleccionar un lenguaje es importante conocer sus características. Seguidamente se describen C++ y Python, dos de los lenguajes más usados en la implementación de aplicaciones gráficas.

#### 1.10.1 C++

C++ nace en 1983 y fue creado por B. Stroustrup [27] a partir del C original, desarrollado por Kernighan y Ritchie en 1972. Es un lenguaje basado en el paradigma de la Programación Orientada a Objetos (POO), muy versátil, potente y general [28]. Aporta un nivel de productividad muy superior a C, sin comprometer la flexibilidad, el rendimiento o el control de la codificación. Es altamente compatible con las bibliotecas gráficas Glide, OpenGL, DirectX y G3D, algunas de las más utilizadas en el desarrollo de aplicaciones gráficas, sobre todo en videojuegos [29].

#### 1.10.2 Python

Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90, cuyo nombre está inspirado en el grupo de comediantes ingleses Monty Python. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia que favorece un código legible. Se trata de un lenguaje interpretado o script, con tipado dinámico y multiparadigma, entre los que se cuenta el orientado a objetos. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo y muy rápido. Python no es adecuado sin embargo para la

programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico [30].

### 1.11 Editores de escenarios

Los editores de escenarios, también conocidos como editores de mapas, son las herramientas que se usan para diseñar los escenarios. Permiten que un usuario pueda modificar o personalizar completamente el mapa del juego sin conocimientos de programación [31]. Surgen de la necesidad de brindar mayor libertad a los usuarios de videojuegos para configurar escenarios de forma personalizada. Los productos que se construyen con estos editores tienen como único destino el videojuego al que están asociados, por lo que no se utilizan para la creación de paseos virtuales.

Los Editores de escenarios estudiados permiten deformar las geometrías, esto puede ser un riesgo ya que podrían dañarse objetos que poseen dimensiones y características distintivas. Además cada editor realiza la gestión de sus recursos basados en las características generales del videojuego para el que fueron creados y no con las de los objetos de un centro expositivo tradicional.

A continuación se describen las herramientas Cryengine Sandbox, Warcraft III World Editor y Unreal Level Editor, de las cuales se seleccionaron características comunes, que podrían ser útiles e intuitivas para las personas que necesiten diseñar escenarios en tres dimensiones para un paseo virtual.

#### 1.11.1 Cryengine Sandbox

Es el editor de niveles usado en la línea de juegos CryEngine, producidos por Crytek<sup>25</sup>. Provee herramientas para facilitar el scripting, animación y creación de objetos. Ha sido incluido en varios juegos de Crytek como Crysis y FarCry. Además se utiliza de forma extensiva con propósitos de modelado. El editor puede construir escenas de exteriores e interiores usando su estilo de edición “aditivo”, el cual lo diferencia de su competidor principal Unreal Editor. Los objetos de la escena se adicionan a un entorno global

---

<sup>25</sup> Empresa desarrolladora de videojuegos alemana fundada en 1999 por tres hermanos turcos: Cevat, Avni y Faruk Yerli.

vacío, usando una forma algorítmica para pintar texturas y objetos en el paisaje. Posee varios parámetros para definir la distribución de texturas o los tipos de vegetación en el escenario. El editor es compatible con todas las funcionalidades que brinda el motor de juegos CryEngine como los vehículos, la física, la secuencias de comandos, iluminación avanzada, sombras en movimiento, cinemáticas, shaders, audio 3D, animaciones, sistemas de partículas, mapa de normales, oclusión de mapas, inteligencia artificial, entre otras; todo esto ofrecido en tiempo real para la edición del escenario [32].

## 1.11.2 Warcraft III World Editor

Es el editor de niveles del juego de estrategia Warcraft III: Reign of Chaos y su expansión Warcraft III: The Frozen Throne, hechos por la compañía Blizzard<sup>26</sup>. Permite a los jugadores crear y personalizar sus propios escenarios y campañas. El editor ha sido usado para la creación de mapas personalizados como el popular Defense of the Ancients (DoTA). Soporta el lenguaje de programación JASS usado por el sistema interno de Warcraft III de Blizzard para programar acciones que no son accesibles mediante la interfaz gráfica. Warcraft III World Editor está dividido en varios módulos que permiten al usuario configurar diferentes características de un mapa, lo que facilita el acceso a modelos personalizados, texturas e iconos. Entre estos módulos se pueden señalar:

- Editor de terreno: Brinda una paleta de edición provista de herramientas para la creación de islas, agua, árboles, luces avanzadas, entre otras.
- Editor de objetos: Permite al usuario modificar la mayoría de los atributos de los objetos del juego, así mismo permite crear objetos personalizados basados en otros ya existentes.
- Editor de disparadores (Trigger Editor): Permite crear triggers (disparadores), que se lanzan bajo condiciones específicas dadas por eventos dentro del juego.
- Editor de campañas: Permite editar campañas para exportarlas hacia el videojuego [33].

---

<sup>26</sup> Referido a la empresa “Blizzard Entertainment, Inc.”, desarrolladora y distribuidora de videojuegos estadounidense con sede en Irvine, California y fundada el 1 de junio de 1994.

### 1.11.3 Unreal Level Editor

El Unreal Editor (o UnrealEd) es una herramienta de creación de contenidos que llena los espacios vacíos con modelos hechos en herramientas de modelado 3D y con el contenido del juego. Sus características incluyen:

- Diseño visual de los niveles, colocación, edición y organización de los objetos en el escenario, los artefactos del inventario, non-player character (NPCs), rutas de acceso y luces.
- Diseño arquitectónico del sistema basado en pinceles para creación rápida de escenarios.
- Un conjunto de herramientas para el trabajo de edición, entre las que se encuentran Unreal Matinee, editor de terreno, editor de materiales, editor de mallas, Unreal Facade, FaceFX Studio, editor de post-procesamiento [34].

### 1.12 Características de los formatos de almacenamiento 3D

Un formato de almacenamiento es un conjunto de reglas que definen la manera correcta de almacenar datos en memoria de forma persistente. Es importante identificar un formato que satisfaga las expectativas del programa que se desea desarrollar, o encontrar un conjunto de características comunes que enriquezcan y justifiquen la opción de diseñar un formato propio. En la actualidad se utilizan diversos formatos para el almacenamiento de la información en ficheros, siendo los más comunes el binario y el ASCII<sup>27</sup>.

---

<sup>27</sup> American Standard Code for Information Interchange. Código normalizado en Estados Unidos para el intercambio de información. Forma de representar números, letras y otros símbolos, de manera que puedan ser introducidos en el ordenador a través del teclado.

## 1.12.1 Formato binario

Los ficheros binarios contienen típicamente una secuencia de bytes<sup>28</sup> o grupos ordenados de ocho bits. El formato binario puede incluir múltiples tipos de datos en el mismo archivo, tales como imagen, video, audio y las características de un escenario virtual [35]. Son muy utilizados en juegos, por la rapidez de carga e interpretación del archivo. Entre los ficheros de este tipo más conocidos están el STL (Standard Tessellation Language), .3DS y el .BLEND. A continuación se caracterizan brevemente.

### STL

El formato STL (\*.stl) es nativo de Stereolithography CAD, creado por 3d System. En este formato se realiza una representación basada en caras o polígonos de superficies u objetos sólidos. Este formato tiene una cabecera (header) de 80 caracteres, que generalmente se ignora. Después del header hay un entero sin signo que indica cuántas caras triangulares contiene el fichero. Luego se representan las descripciones de cada triángulo hasta el final del archivo. Cada triángulo se describe mediante 12 números de punto flotante de 32 bits (float<sup>29</sup>), tres para la dirección de la normal<sup>30</sup> de la cara y tres para las coordenadas de cada vértice. Para finalizar el bloque se encuentra un entero sin signo de dos bytes (frecuentemente 0) [36].

### Autodesk 3DS

Creado por Autodesk para el 3D Studio en su versión 1. Fue reemplazado en versiones posteriores por el formato .MAX, aunque ha seguido usándose. Su organización está basada en bloques, en el que cada uno contiene un identificador, la longitud de los datos, y los datos [37]. Tiene una estructura jerárquica. Cada bloque está dividido en sub-bloques y no tiene que existir un orden en el fichero, por ello es que a simple vista es difícil entender su estructura.

---

<sup>28</sup> Una secuencia de ocho bits adyacentes operados como una unidad.

<sup>29</sup> Tipo de dato compuesto por 4 bytes.

<sup>30</sup> Referido a la perpendicular a una superficie.

### BLEND

Creado como formato de archivo nativo para Blender, está diseñado con un alto nivel de abstracción y permite crear proyectos de animaciones complejas dentro de un archivo, además de la reutilización eficiente de los datos y el uso de plantillas. Resulta ser una “base de datos” no estándar, diferente al grafo de escena tradicional, pero basado en la creación de “mundos de datos” 3D donde se pueden construir los grafos de escena que se necesiten. Los bloques de datos en Blender están almacenados en un árbol principal y son como los bloques de construcción del usuario: pueden ser copiados, modificados y vinculados unos con otros. Por ejemplo, el tipo de bloque Object hace aparecer en el escenario 3D un bloque Mesh. El Mesh sólo almacena la ubicación de los vértices y las caras. Otros tipos de bloques, como el Material, pueden vincularse con Meshes [38].

A continuación se muestra un ejemplo de los bloques de datos que conforman el formato de Blender:

**<ccode>**

```
typedef struct ID {
```

```
void *next, *prev; //para insertar en listas
```

```
struct ID *newid; //datos temporales para encontrar links nuevos
```

```
struct Library *lib; //puntero a librería opcional
```

```
char name[24]; //nombre único, comienza con un identificador de 2 bytes
```

```
short us; //cantidad de usuarios en el bloque
```

```
short flag; //banderas de bit para indicar tipos especiales
```

```
} ID;
```

**</ccode>**

## 1.12.2 Formato ASCII

Archivo que utiliza la codificación ASCII, estos se pueden abrir y editar con cualquier editor de texto, frecuentemente utilizan la extensión “.txt” [39]. Los ficheros ASCII o de texto, son menos sensibles a corromperse que los binarios. Un pequeño error en un archivo binario puede hacerlo ilegible, lo que no ocurre con un archivo ASCII. Estos usan un formato estándar, por lo que múltiples programas son capaces de leerlos y editarlos [40]. Entre los ficheros más conocidos que emplean este formato se encuentran el Wavefront OBJ, STL, ASCII Scene Exporter (ASE) y Ogre DotScene, los que a continuación se caracterizan brevemente:

### Wavefront OBJ

Es un formato de archivo de objetos 3D creado para el uso de Wavefront's Advanced Visualizer y recomendable para objetos estáticos. Contiene geometrías poligonales que presentan puntos, aristas y caras para definir un objeto de forma libre, los cuales se derivan en curvas, superficies y objetos poligonales. No necesita ningún tipo de encabezado, aunque es usual poner un comentario al inicio del fichero [41].

### STL

Puede encontrarse en ficheros ASCII o binarios, aunque es más común en el último debido a que los archivos son más pequeños. Es uno de los formatos más simples en cuanto a estructura, fácil de cargar y con poca cantidad de atributos, que sólo toma en cuenta la geometría de los objetos (caras y vértices) [42].

### ASE

Es usado para cargar objetos estáticos y se organiza en forma de jerarquía de clases. Las estructuras de las clases se conforman con un identificador y un bloque de identificadores encerrados entre llaves. Es propio de éste formato que los identificadores comiencen por un asterisco, ejemplo: \*Nombre\_del\_atributo [43].

### OGRE DotScene

Formato de fichero XML<sup>31</sup> estándar con extensión .scene. Creado para usarse en escenas de Ogre. La estructura está basada en nodos, cada uno de los cuales representan los diferentes elementos presentes en la escena. No contiene datos acerca de las mallas, texturas u otro recurso. Los datos de los recursos están contenidos en otro tipo de ficheros, el Dotmesh (\*.mesh) y el Dotmaterial (\*.material), de los cuales el .scene se auxilia para guardar la información de los objetos. En el \*.mesh se almacena la información física de las mallas, como vértices, posición o dirección de las normales y en el \*.material se encapsulan las coordenadas de texturas de los objetos presentes en la escena [44]. Es importante señalar que aunque el \*.material es un fichero ASCII, el \*.mesh es un fichero de tipo binario.

### 1.13 Consideraciones parciales

- El uso de los editores de escenarios estudiados queda descartado ya que los escenarios que se obtienen, son específicos para los videojuegos a los que tributan, lo cual constituye una limitante para la construcción de paseos virtuales interactivos con elementos en tres dimensiones.
- En el diseño de modelos y en la creación de escenarios se deben tener habilidades bien desarrolladas, pues un modelo gráfico o escenario en manos de personal inexperto podría terminar deformado o dañado.
- El estudio de los editores de escenarios y de las herramientas de diseño gráfico permitió identificar algunas formas comunes de presentarle al usuario las herramientas de diseño. Se evaluó la posición en la interfaz o vía de acceso a cada funcionalidad, además de la imagen que la identifica. También se analizaron factores como las formas de realizar la visualización de los objetos en el espacio, entre otros elementos que podrían ser útiles para lograr una propuesta que brinde una interfaz intuitiva para personas sin experiencia en estas tareas y que necesiten diseñar escenarios en tres dimensiones para un paseo virtual.
- La codificación de los componentes necesarios en un paseo virtual, resulta una de las principales dificultades dentro del desarrollo ya que aunque se utilice un IDE o un

---

<sup>31</sup> XML, siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el "World Wide Web Consortium" (W3C).

## Capítulo 1: Fundamentación teórica

---

motor de juego determinado e independientemente de las facilidades que brindan, el desarrollador debe tener dominio de lenguajes y técnicas que son intrínsecamente complejas.

- Las herramientas estudiadas han creado formatos de almacenamiento específicos que responden a sus necesidades particulares y los formatos estudiados no contemplan las particularidades de la herramienta que se propone en el trabajo.

## Capítulo 2: Propuesta de solución

### Introducción al capítulo

En este capítulo se presenta la propuesta de solución y se describen varias de las actividades realizadas para lograr la creación de la herramienta CEV, incluyendo imágenes de los modelos generados durante el desarrollo ingenieril [45]. Se representan y describen también los cambios propuestos a la etapa de implementación del proceso de desarrollo utilizado actualmente en el CEDIN, para la creación de paseos virtuales interactivos con elementos en tres dimensiones.

### 2.1 Desarrollo de la herramienta CEV

Se desarrolla una herramienta para crear, editar y visualizar paseos virtuales interactivos con elementos 3D, que está dirigida a dos tipos de usuarios:

- Los responsables de editar las “áreas expositivas tradicionales”, que por lo general son personas que no poseen experiencia en el manejo de herramientas de diseño gráfico.
- Los miembros del GDPV, que crean sus productos mediante la utilización de un proceso de desarrollo establecido en el CEDIN.

La herramienta está formada por dos módulos, los cuales se describen a continuación:

- El módulo “Editor” permite al usuario crear, editar, almacenar los recursos gráficos y exportar paquetes con los paseos virtuales que desee visualizar posteriormente.
- El módulo “Visualizador” permite al usuario seleccionar el paseo virtual que desee de los contenidos en el paquete que se encuentren en la herramienta.

#### 2.1.1 Especificación de los requisitos de software

Los requisitos son una descripción abstracta de los servicios que el sistema debe proporcionar al cliente y las limitaciones bajo las cuales éste debe operar [46]. Ellos cumplen tres propósitos:

- 1 Permitir a los desarrolladores entender cómo el cliente quiere que trabaje el sistema.

- 2 Especificar a los diseñadores la funcionalidad y las características que el sistema debe tener.
- 3 Especificar a los integrantes del grupo de pruebas lo que deben realizar para verificar que las exigencias del cliente están satisfechas [47].

Varios autores como Dromey [48] y Pressman [49] clasifican los requisitos en Requisitos Funcionales (**RF**) y Requisitos no Funcionales (**RnF**). Basado en el estudio de los editores de escenarios y las herramientas de diseño gráfico, se definieron los requisitos básicos para obtener una herramienta con las características y facilidades esperadas. Se especifican a continuación los requisitos para ambos módulos de la herramienta, a cada uno de los cuales se le asocia un número, el nombre y una breve descripción.

### ➤ **RF - Módulo Editor**

**RF 1** Administrar proyecto: maneja las funcionalidades crear, salvar, eliminar y cargar un paseo virtual.

**RF 1.1** Crear proyecto: crea una carpeta con el nombre del paseo virtual a diseñar, dentro de esta se crea una carpeta que contiene los recursos (modelos gráficos, cámaras, recorridos, luces) del paseo virtual y el fichero principal del paseo. El fichero y la carpeta de los recursos deben tener el mismo nombre.

**RF 1.2** Salvar proyecto: guarda los cambios realizados al paseo cargado en la carpeta creada según la distribución descrita en el requisito RF 1.1.

**RF 1.3** Eliminar proyecto: elimina el proyecto, la carpeta del paseo con el fichero principal y todos sus modelos gráficos.

**RF 1.4** Cargar proyecto: carga el proyecto que seleccione el usuario en el editor para luego realizar cambios.

**RF 2** Administrar modelos del sistema: adiciona o elimina modelos al sistema.

**RF 2.1** Adicionar modelo al sistema: adiciona un modelo al sistema.

**RF 2.2** Eliminar modelo del sistema: elimina un modelo del sistema. Antes de eliminarlo se pregunta al usuario si realmente está seguro de eliminarlo definitivamente del sistema.

**RF 3** Administrar objetos del entorno: maneja las funcionalidades relacionadas con los objetos decorativos o expositivos que se muestran dentro del entorno.

**RF 3.1** Cargar objeto en el entorno: carga un objeto y lo muestra en el área de visualización.

**RF 3.2** Mover objeto del entorno: mover un objeto hacia cualquier área válida en el entorno.

**RF 3.3** Alinear objetos: al mover un objeto se muestra una línea que indica la posición del objeto seleccionado con respecto a otro objeto cercano, lo que permite alinearlo con otro objeto del escenario.

**RF 3.4** Activar sombra: muestra u oculta la sombra de un objeto en el entorno.

**RF 3.5** Cambiar vista del objeto: cambia el ángulo de vista de todos los objetos. Las vistas pueden ser: frente, arriba, derecha, izquierda y libre en perspectiva.

**RF 3.6** Cambiar modo de visualización: cambia la forma de visualizar los objetos gráficos de la escena. La forma de visualización puede ser: sólido, puntos o malla.

**RF 3.7** Eliminar objeto del entorno: elimina el objeto seleccionado del entorno y del área de visualización.

**RF 4** Modificar objeto decorativo: modifica, tanto el tamaño de un objeto decorativo como la cantidad que puede existir en el paseo.

**RF 4.1** Escalar objetos decorativos: aumenta o disminuye el tamaño de un objeto decorativo.

**RF 4.2** Duplicar objetos decorativos: realiza una copia de un objeto decorativo en el entorno.

**RF 5** Modificar información de los objetos: crea o modifica la información referente a un objeto y permite clasificarlo en decorativo, expositivo o delimitador.

**RF 5.1** Actualizar información de objeto: modifica la información asociada a un objeto.

**RF 5.2** Definir tipo de objeto: clasifica el objeto en decorativo, expositivo o delimitador.

**RF 6** Administrar sonido: adiciona un sonido al entorno, permite su reproducción recursiva, asociarlo a un objeto o escenario y da la posibilidad de activarlo o desactivarlo.

**RF 6.1** Adicionar sonido: adiciona un sonido al sistema.

**RF 6.2** Eliminar sonido: elimina un sonido del sistema.

**RF 6.3** Repetir sonido: realiza la reproducción recursiva de un sonido.

**RF 6.4** Asociar sonido a la escena: asigna un sonido a un entorno.

**RF 6.5** Quitar sonido de la escena: quita un sonido asociado a un entorno.

**RF 7** Administrar cámara: adiciona una cámara al paseo, permite modificar su altura y velocidad así como definir una trayectoria y eliminarla del paseo.

**RF 7.1** Adicionar cámara: adiciona una cámara al paseo.

**RF 7.2** Modificar altura: aumenta o disminuye la altura de una cámara.

**RF 7.3** Definir trayectoria: define el recorrido que puede realizar la cámara dentro del paseo.

**RF 7.4** Eliminar cámara: elimina la cámara del paseo.

**RF 7.5** Eliminar Trayectoria: elimina la trayectoria definida por el actor.

**RF 8** Detectar colisiones: detecta la cercanía entre objetos.

**RF 9** Exportar paseo: guarda todos los cambios realizados al paseo en la carpeta del visualizador.

**RF 10** Listar recursos: lista los nombres de los proyectos y los nombres de los modelos que se encuentren en el sistema.

### ➤ **RnF - Módulo Editor**

#### **1. Usabilidad.**

**RnF 1.1** Utilización del sistema: los usuarios deberán poseer conocimientos mínimos en el uso de aplicaciones informáticas.

#### **2. Portabilidad.**

**RnF 2.1** Ejecución de la herramienta: puede ser compilada y ejecutada sobre la plataforma Windows y Linux.

### 3. Restricciones en el diseño y la implementación.

**RnF 3.1** Lenguajes de programación: C++.

**RnF 3.2** Herramientas de desarrollo: el IDE QT Creator y el motor de juego Ogre.

**RnF 3.3** Herramientas de diseño gráfico: Blender.

**RnF 3.4** Implementación del sistema: Basado en el paradigma de programación orientada a objetos (POO).

### 4. Hardware

**RnF 4.1** La computadora deberá tener: memoria RAM de 1 Gb o superior, velocidad de microprocesador a 3.0 GHz o superior, video de 256 Mb o superior.

### 5. Interfaz o apariencia externa.

**RnF 5.1** Interfaz de usuario: las opciones se muestran en una barra de herramienta. Cada funcionalidad tiene un icono identificador y muestra un texto que describe brevemente la función que realiza. Consta de un menú en la parte inferior izquierda para listar los nombres de los proyectos que se encuentren en el sistema. En la parte superior izquierda, se muestra un menú para listar los objetos que se encuentren en el entorno. Los nombres de los modelos que se encuentren en el sistema se muestran en la parte inferior derecha.

### 6. Ayuda y documentación.

**RnF 6.1** Ayuda y documentación: cuenta con un manual de ayuda que describe cada uno de los servicios que brinda la aplicación, además en la pantalla inicial, de forma gráfica, se muestran las principales acciones que puede realizar el usuario.

#### ➤ **RF - Módulo Visualizador.**

**RF 1** Inicializar datos: lista en el visualizador los nombres de los paseos previamente creados en el módulo Editor que se encuentren en la carpeta de proyectos de la aplicación.

**RF 2** Definir modo de paseo: define la forma en que se realiza el recorrido por el paseo virtual, el recorrido puede realizarse de forma libre (para esto se usan los controles del teclado) o dirigido (se muestra el paseo a través del recorrido de la cámara).

**RF 3** Editar cámara: modifica la velocidad y la altura de la cámara mediante los dispositivos de entrada, de esta forma es posible realizar el recorrido por el paseo virtual.

**RF 3.1** Modificar velocidad: disminuye o aumenta la velocidad con que se moverá la cámara en el paseo.

**RF 3.2** Modificar altura: disminuye o aumenta la altura a la que estará la cámara durante el paseo.

**RF 3.3** Pausar paseo: disminuye la velocidad de la cámara a valor cero.

**RF 4** Mostrar información asociada a un objeto: Muestra la información asociada a un objeto en una pequeña ventana al hacer clic con el mouse (ratón) encima del mismo.

**RF 5** Resaltar objetos expositivos: para indicar que un objeto puede ser seleccionado, al pasar el mouse por encima, el puntero cambia de forma.

**RF 6** Comenzar paseo: carga todos los modelos, las luces y recorridos del paseo. Activa la cámara seleccionada y muestra el paseo en pantalla completa.

**RF 7** Cerrar paseo: cierra la vista previa del paseo virtual y se muestran el resto de las ventanas que conforman el editor.

**RF 8** Mostrar ayuda del paseo: muestra información sobre cómo realizar el recorrido por el entorno virtual.

### ➤ **RnF - Módulo Visualizador.**

#### **1. Usabilidad.**

**RnF 1.2** Utilización del sistema: visitantes interesados en observar un paseo virtual determinado con habilidades en la manipulación del teclado y mouse de una computadora.

#### **2. Rendimiento.**

**RnF 2.1** Velocidad de representación de los gráficos en tiempo real.

#### **3. Portabilidad.**

**RnF 3.1** Ejecución de la herramienta: permite ser compilada y ejecutada sobre la plataforma Windows y Linux.

#### **4. Restricciones de diseño.**

**RnF 4.1** Lenguajes de programación: C++.

**RnF 4.2** Herramientas de desarrollo: el IDE QT Creator y el motor de juego Ogre.

**RnF 4.3** Herramientas de diseño gráfico: Blender.

**RnF 4.4** Implementación del sistema: POO.

### 5. **Hardware**

**RnF 5.1** La computadora deberá tener: memoria RAM de 1 Gb o superior, velocidad de microprocesador a 3.0 GHz o superior, video de 128 Mb o superior.

### 6. **Interfaz.**

**RnF 6.1** Interfaz de Usuario: las opciones se muestran en una barra de herramienta, cada funcionalidad tiene un icono identificador y muestra un texto que describe brevemente la función que realiza. Cuenta con un menú para listar los nombres de los proyectos que se encuentren en el sistema en la parte superior izquierda.

### 7. **Ayuda y documentación en línea.**

**RnF 7.1** Ayuda y documentación: cuenta con un manual de ayuda que describe cada uno de los servicios que brinda la aplicación, además en la pantalla inicial, de forma gráfica, se muestran las principales acciones que puede realizar el usuario.

### 2.1.2 **Definición de los casos de uso**

Una vez recopilados los requisitos del sistema, se definen los casos de uso (CU) que facilitan la descripción de las funciones. El caso de uso describe la manera en que los actores interactúan con el sistema [50].

#### **Definición de los actores del sistema (Actores).**

Un actor representa a las personas o dispositivos que juegan su papel como impulsores del sistema. A continuación se describen los actores que pueden interactuar con la herramienta para crear, editar y visualizar paseos virtuales interactivos con elementos en 3D.

<b>Actores</b>	<b>Justificación</b>
Diseñador del Paseo.	Es la persona que interactúa con el módulo "Editor", para crear, editar o exportar uno o varios paseos virtuales.
Usuario.	Es la persona que interactúa con el módulo "Visualizador", para realizar el recorrido por uno de los paseos virtuales disponibles.

**Tabla 1. Descripción de los actores que interactúan con la herramienta.**

### Listados de casos de uso (CU):

#### Módulo Editor.

- CU-1. Administrar proyecto.
- CU-2. Administrar modelos del sistema.
- CU-3. Administrar objetos del entorno.
- CU-4. Modificar objeto decorativo.
- CU-5. Modificar información de los objetos.
- CU-6. Administrar sonido.
- CU-7. Administrar cámara.
- CU-8. Detectar colisiones.
- CU-9. Exportar paseo.
- CU-10. Listar recursos.

#### Módulo Visualizador.

- CU-1. Inicializar datos.
- CU-2. Definir modo de paseo.
- CU-3. Editar cámara.
- CU-4. Mostrar información asociada a un objeto.
- CU-5. Resaltar objetos expositivos.
- CU-6. Comenzar paseo.
- CU-7. Cerrar paseo.
- CU-8. Mostrar ayuda del paseo.

### 2.1.3 Diagrama de casos de uso

La representación de los diagramas de casos de uso permite especificar la funcionalidad y el comportamiento del sistema mediante su interacción con los actores. La figura 1 muestra la vista de casos de uso arquitectónicamente significativos, para ambos módulos.

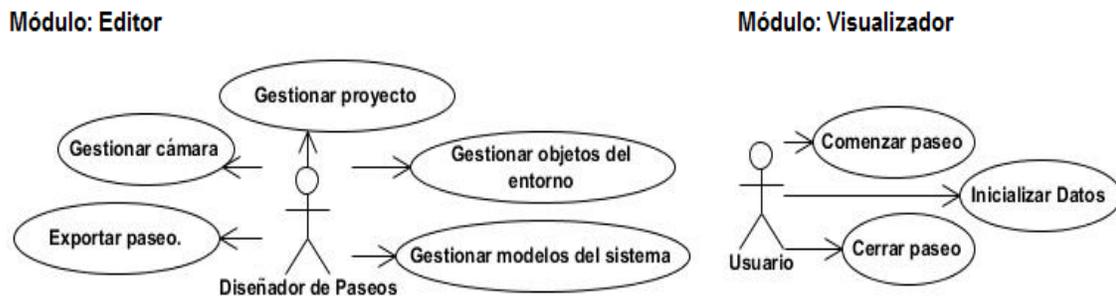


Figura 1. Diagramas de casos de usos arquitectónicamente significativos.

En la implementación de los casos de uso expuestos se tienen en cuenta algunas consideraciones técnicas, producto del estudio del estado del arte y de las condiciones

reales para el desarrollo. Un ejemplo de consideración es la técnica seleccionada para la creación de paseos guiados por el sistema. En este caso se desarrolló un recorrido para la cámara sobre una línea, aunque se estudiaron otras técnicas como por ejemplo, el desplazamiento por una superficie definida por colores y algunas formas de detección de colisiones. Debido fundamentalmente al tiempo y la cantidad de recursos disponibles, el sistema no cuenta todavía con un módulo que implemente estas técnicas.

Para cada caso de uso se realizó una detallada descripción paso a paso, de la secuencia de eventos que los actores realizan para completar un proceso a través del sistema. Se especificaron las acciones que realizan los actores y la respuesta del sistema a cada evento. La realización de los casos de usos se describe durante el análisis y el diseño a través de los diagramas de colaboración y los diagramas de secuencia de los cuales se realizaron más de 30 en cada caso. Esta labor permite contar con una documentación que sirve de guía a la codificación de los componentes y a la mejor comprensión del código por terceros.

### **2.1.4 Modelos conceptuales de la herramienta**

Los modelos que se muestran a continuación exponen los conceptos más significativos asociados a la creación edición y visualización de paseos virtuales. Con lo que se pretende crear las bases para el análisis y diseño. En la propuesta del sistema se definió que la herramienta a desarrollar estará compuesta por dos módulos por lo que se realizó un modelo conceptual para cada uno de ellos.



arquitectónicamente significativos, representada en la figura 1, se realiza el diseño de la vista lógica.

La herramienta se caracteriza por ser una aplicación de escritorio la cual solo necesita de una computadora para su ejecución. En la figura 4 se muestra la organización del sistema en capas mediante el patrón “3 capas” que permite simplificar y comprender el desarrollo del sistema, lo que reduce la dependencia. Esto trae consigo que las capas más bajas no son conscientes de ningún detalle de las superiores. Cada una de las clases se organiza según la capa a la que pertenece.

1. Capa de presentación: el paquete interfaz de usuario, contiene las clases arquitectónicamente significativas encargadas de enviar las peticiones del cliente a las clases que se encuentran en la capa de aplicación.

2. Capa de negocio: el paquete lógica del negocio contiene las clases y funcionalidades de control y ejecución de las peticiones que llegan desde la capa de interfaz de usuario. Sirve como intermediario entre la lógica de presentación y la lógica de negocio.

Paquete de dominio: contiene las clases del negocio y sus funcionalidades.

3. Capa de datos: acceso a datos contiene la clase y funcionalidades de acceso a los datos es decir la información almacena en ficheros. Sirve como intermediario entre la lógica de datos y la lógica de negocio.

CEV Ficheros: recoge las especificaciones del almacenamiento de los datos en ficheros.

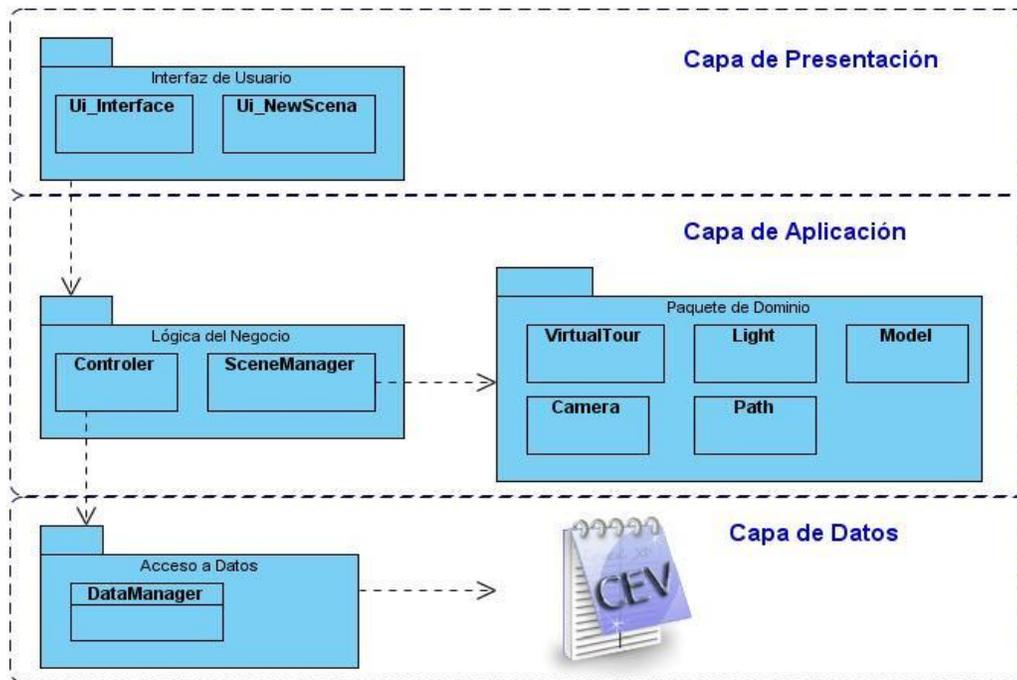


Figura 4. Vista Lógica de la herramienta.

### 2.1.6 Diagramas y descripción de las clases

En estos diagramas se representan las clases y sus relaciones. En el modelo de análisis se utilizan tres estereotipos diferentes sobre clases: clase interfaz, clase de control y clase entidad. Para la especificación detallada de esta etapa se realizaron más de 30 diagramas de colaboración para los dos módulos. El diagrama de clases del diseño incluye más detalles que el diagrama de clases del modelo de análisis lo cual es necesario para la adaptación del diseño al entorno de implementación. En esta etapa se realizaron más de 30 diagramas de secuencia. A continuación se muestra el diagrama de clases del diseño de cada módulo y una breve descripción de las clases arquitectónicamente significativas para la herramienta.

Las figuras 5 y 6 representan las clases que conforman la herramienta para la realización de los casos de uso de forma general.

Leyenda:

- Clases que se encuentran en la capa Presentación.
- Clases que se encuentran en la capa Aplicación.
- Clases que se encuentran en la capa Acceso a datos.

#### Módulo Editor

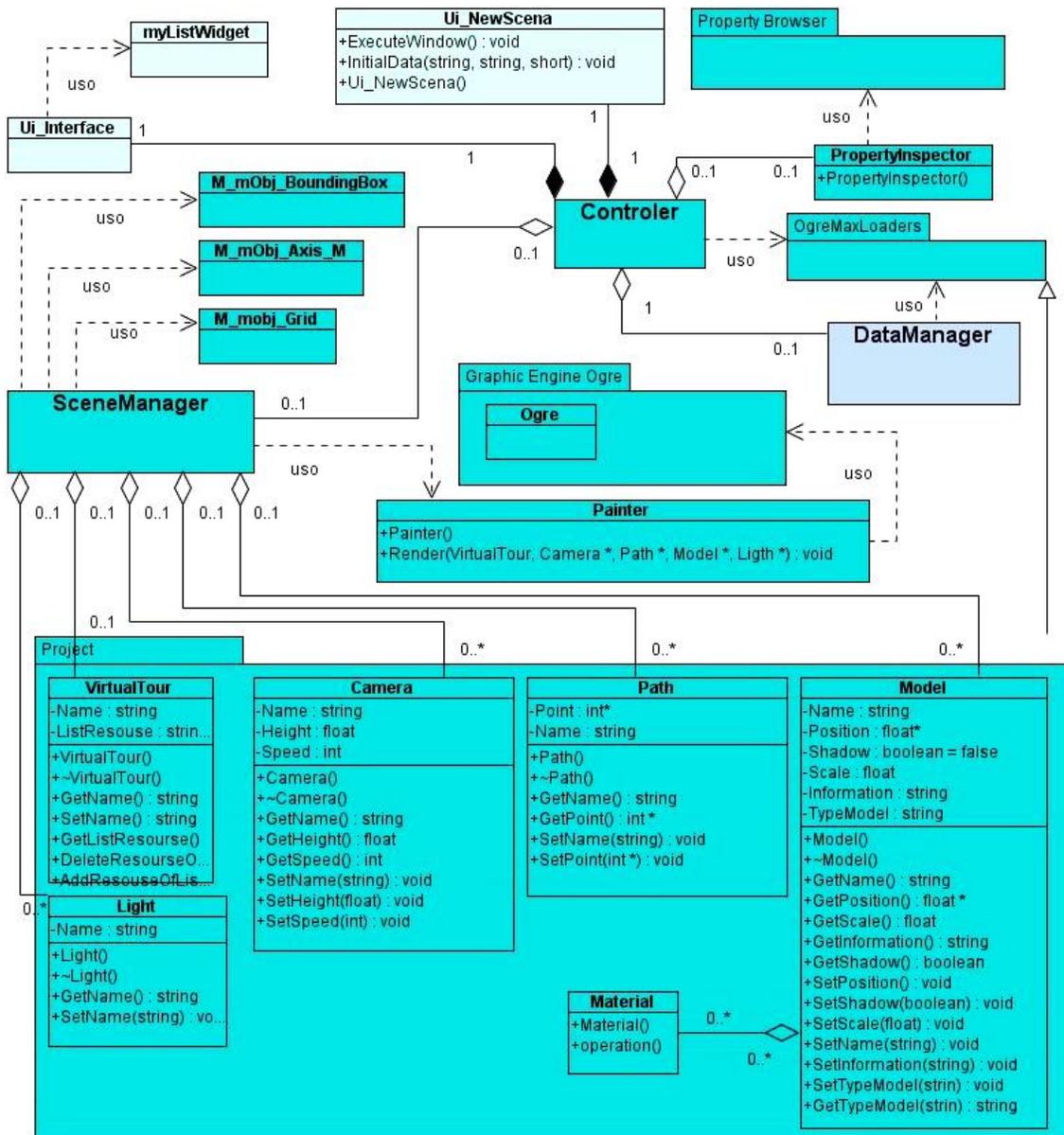


Figura 5. Diagrama de clases del diseño del módulo Editor.

### Clases arquitectónicamente significativas del módulo Editor

SceneManager. Clase controladora que tiene la responsabilidad de administrar los datos de las entidades que maneja el sistema.

“Controler”. Clase controladora que tiene la responsabilidad de enviar las respuestas a las peticiones que realiza el usuario hacia la clase Interfaz. Contiene todas las funcionalidades que dan respuesta a los casos de uso del sistema.

“DataManager”. Clase controladora que tiene la responsabilidad de administrar los ficheros que almacenan los datos persistentes del sistema.

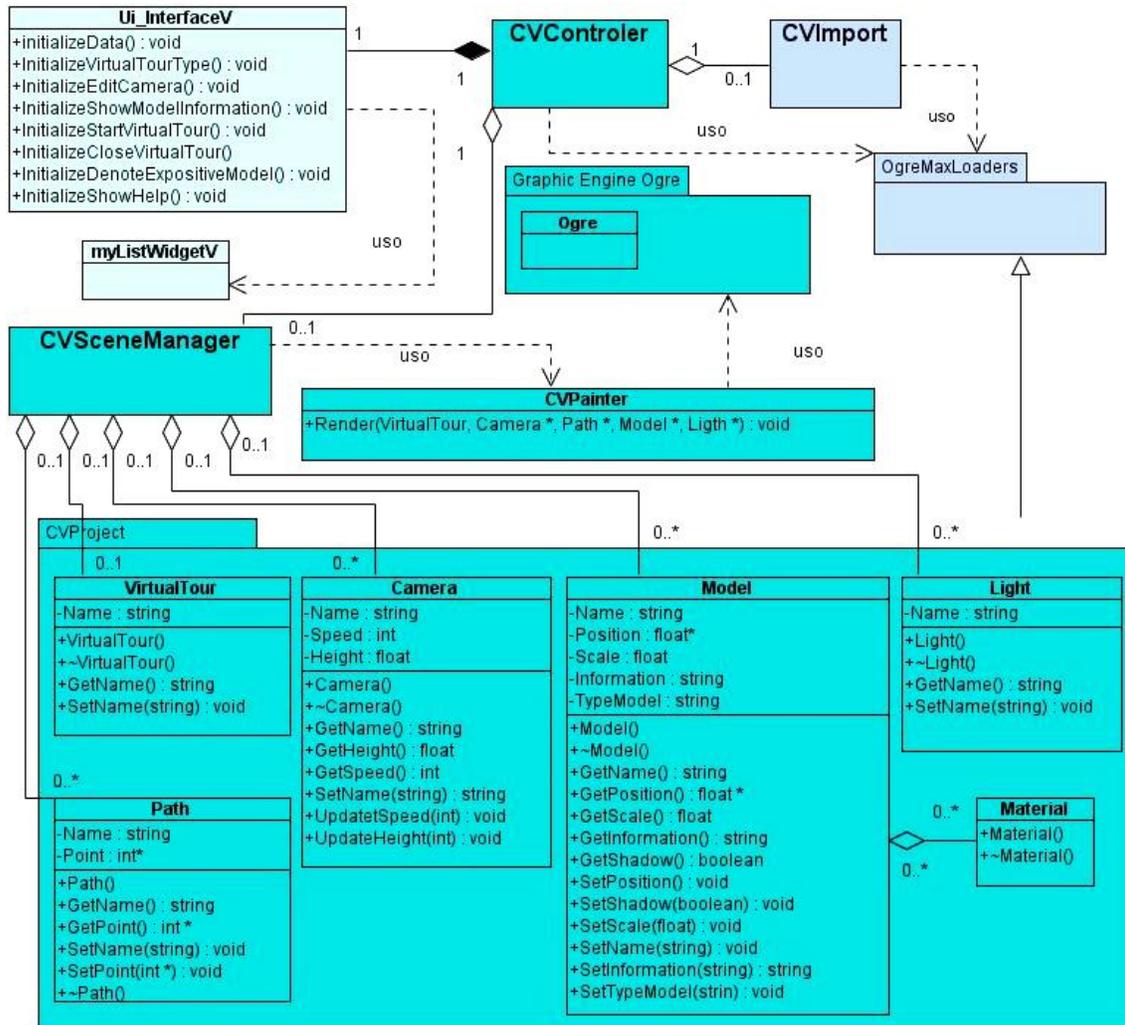


Figura 6. Diagrama de clases del diseño del módulo Visualizador.

### Clases arquitectónicamente significativas del módulo Visualizador

“CVSceneManager”. Clase controladora que tiene la responsabilidad de administrar los datos de las entidades que maneja el sistema.

“CVControler”. Clase controladora que tiene la responsabilidad de enviar las respuestas a las peticiones que realiza el usuario hacia la clase Interfaz. Contiene todas las funcionalidades que dan respuesta a los casos de uso del sistema.

“CVImport”. Clase controladora que tiene la responsabilidad de administrar los ficheros que almacenan los datos persistentes del sistema.

### 2.1.7 Interfaz de usuario

Con el desarrollo de la interfaz se definen un conjunto de elementos que le permiten al usuario interactuar con el sistema. Durante el presente epígrafe se describe la interfaz diseñada para ambos módulos, teniendo en cuenta los requisitos definidos para el sistema. Una de las principales características de la interfaz que se presenta es que las funcionalidades se encuentran directamente a una distancia de no más de dos clics.

En las figuras 7, 8 y 9 las áreas de la interfaz están enmarcadas en rectángulos e identificadas por una letra. Ahí se encuentran los componentes que le dan respuesta a los casos de uso arquitectónicamente significativos.

En la Fig. 7 muestra la interfaz de usuario del módulo “Editor” la cual de forma general está conformada por siete áreas que se describen a continuación:

- Barra de títulos: se muestra el nombre de la herramienta, el proyecto que esté cargado en escena y los botones para maximizar, minimizar y cerrar la herramienta.
- Área “A”: barra de tarea, donde se encuentran elementos de acceso rápido para algunas de las funcionalidades del sistema.
- Área “B”: componente explorer, que permite listar los nombres de los objetos que se encuentren en la escena (área D).
- Área “C”: proyectos, componente que permite listar los nombres de los proyectos que se encuentran en la carpeta del sistema.
- Área “D”: render, área de edición y visualización de la escena.
- Área “E”: propiedades, componente que muestra los atributos de los recursos (modelos, cámara, recorridos, luces) de la escena y permite editarlos.

## Capítulo 2. Propuesta de solución

- Área “F”: mallas, componente que permite listar los modelos gráficos que se encuentren en el sistema.
- Área “G”: cargar proyecto, ventana que permite introducir los datos necesarios para crear o cargar un proyecto.

A continuación se ilustran las áreas donde se encuentran los componentes de la interfaz que dan respuesta a los casos de uso arquitectónicamente significativos del módulo Editor:

CU-1: Administrar proyecto: áreas “A”, “C” y “G”.

CU-2: Administrar modelos del sistema: áreas “A” y “F”.

CU-3: Administrar objetos del entorno: áreas “A”, “B” y “F”.

CU-7: Administrar cámara: área “E”.

CU-9: Exportar paseo: área “A”.

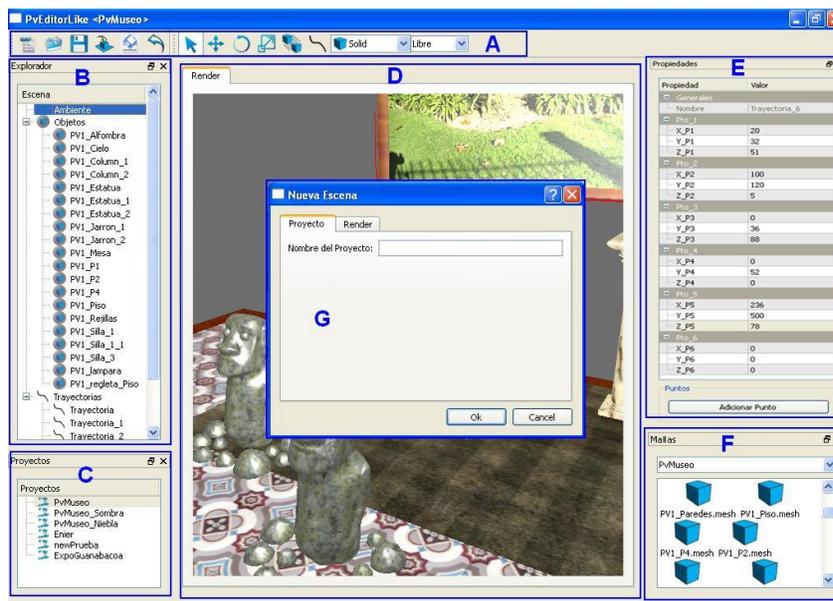


Figura 7. Interfaz Gráfica del Editor.

En las figuras 8 y 9 se muestra la interfaz de usuario del módulo Visualizador. Se describen a continuación los componentes de la interfaz.

- Figuras 8 y 9 - área “A”: barra de tarea, donde se encuentran elementos de acceso rápido para algunas de las funcionalidades del sistema.

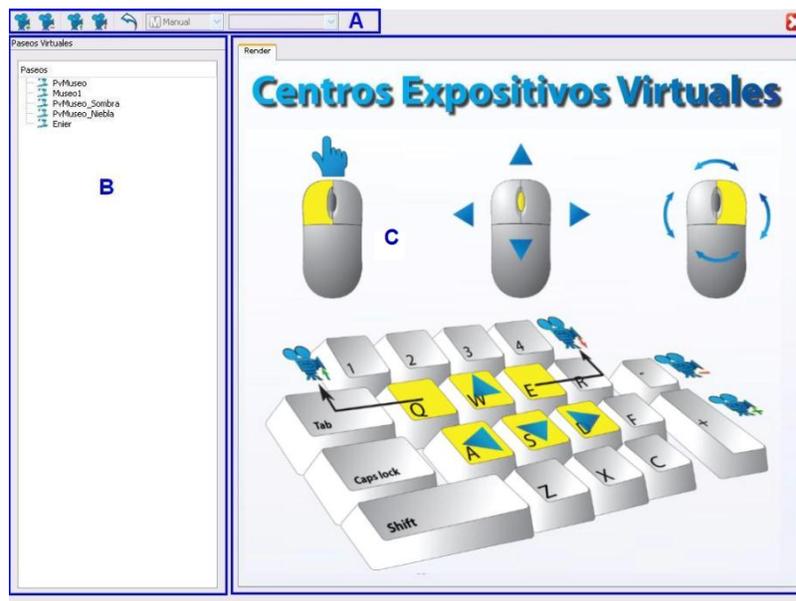
- Figura 8 - área "B": proyectos, componente que permite listar los nombres de los proyectos que se encuentran en la carpeta del sistema.
- Figura 8 - área "C": ayuda, especifica información referente a cómo utilizar el teclado y el mouse para desplazarse por el paseo virtual.
- Figura 9 - área "B": área de visualización, que muestra un paseo virtual.
- Figura 9 - área "C": área para confirmar cerrar un paseo.

A continuación se señalan las áreas donde se sitúan los componentes de la interfaz que dan respuesta a los casos de uso arquitectónicamente significativos del módulo Visualizador:

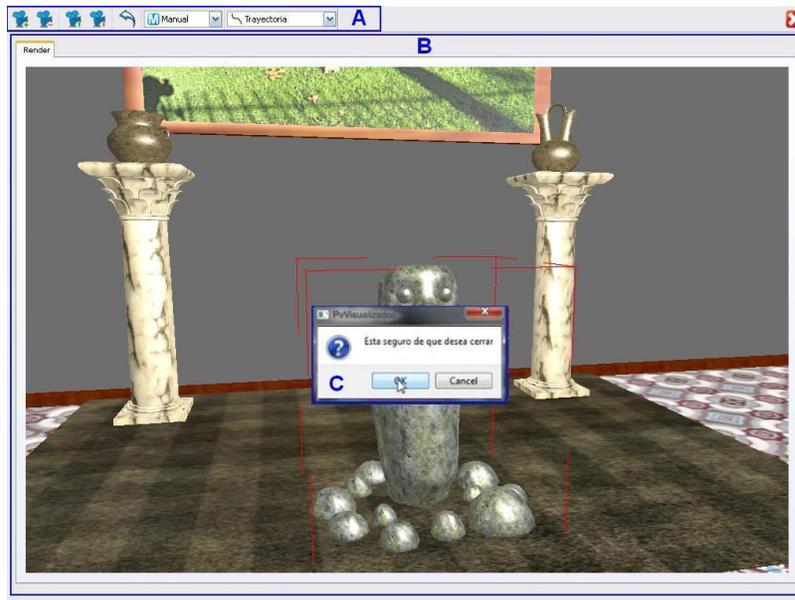
CU-1. Inicializar Datos: Figura 8 - área "B".

CU-6. Comenzar paseo: Figura 8 - área "B".

CU-7. Cerrar paseo: Figuras 8 y 9 - área "A" y figura 9 - área "C".



**Figura 8. Interfaz Gráfica del Visualizador.**



**Figura 9. Interfaz Gráfica del Visualizador.**

### 2.1.8 Modelo de datos

Se diseña un modelo de datos, basado en el formato DotScene de Ogre que almacena los cambios efectuados en el escenario. Al ser descrito a través de etiquetas XML facilita la interpretación de sus bloques de información. Permite también guardar los datos de la escena, los tipos de objetos (“expositivo”, “decorativo” o “delimitador”), la información asociada a cada objeto y un conjunto de puntos que definirán la línea encargada de guiar el paseo virtual una vez que se cargue en el módulo de visualización. La creación de un formato propio asegura la disponibilidad y la posibilidad de realizar modificaciones sobre él, garantiza la introducción de varios niveles de seguridad y permite controlar la disposición de la información para lograr archivos más pequeños. La estructura general del formato se muestra en la figura 10.

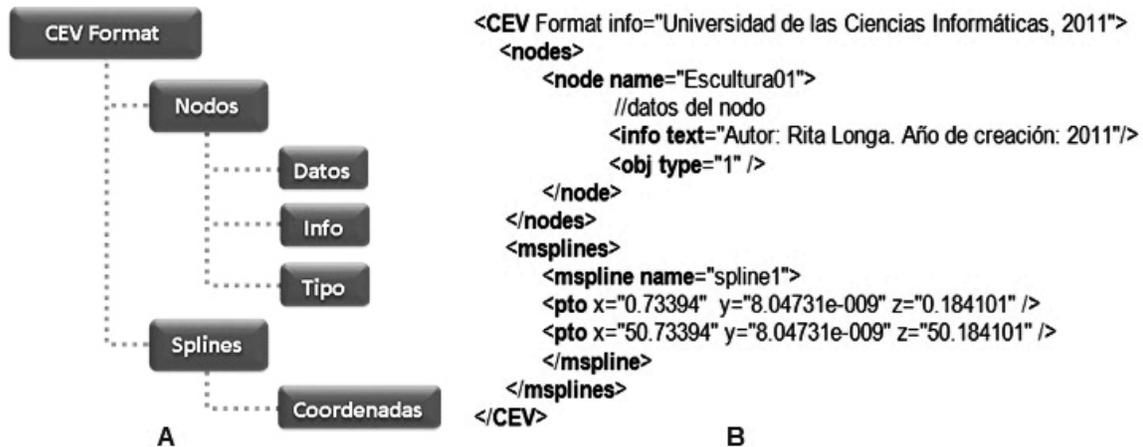


Figura 10. Formato CEV.

Los nodos representan los objetos de la escena con sus datos característicos. Se almacena por cada objeto un texto de información y el tipo de objeto (c para expositivos, 1 para decorativos y 2 para delimitadores). La trayectoria de recorrido de la cámara está definida por una serie de puntos en tres dimensiones con los componentes x, y, z que los caracteriza.

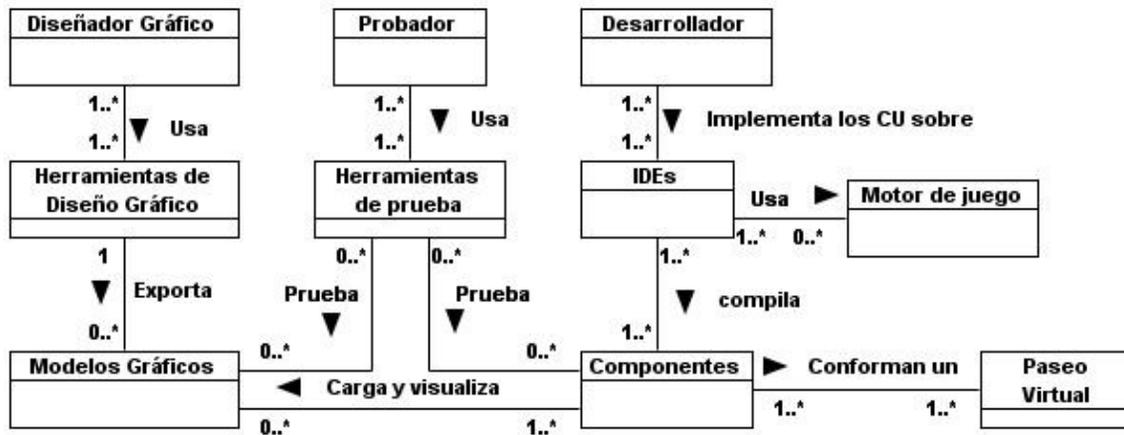
### 2.1.9 Modelo de prueba

Los modelos de pruebas responden a la implementación de los casos de uso más importantes de la aplicación. Para documentarlos se utilizó el formato de los documentos oficiales del CEDIN. Se realizaron pruebas de caja negra, especialmente dirigidas a la interacción del usuario con el sistema y las respuestas que brinda mediante su interfaz. Para el módulo “Editor” se diseñaron 30 casos de pruebas y 10 para el “Visualizador”, obteniendo un 100% de resultados esperados. En el Anexo 1 se puede encontrar una muestra de estos diseños de casos de prueba [52] [53].

## 2.2 Propuesta de modificaciones al proceso de desarrollo de un paseo virtual

Es importante aclarar que la investigación no propone una metodología o proceso completo, sino modificaciones en la etapa de implementación dentro del proceso existente para desarrollar paseos virtuales en el CEDIN. Por tanto, la integración de la

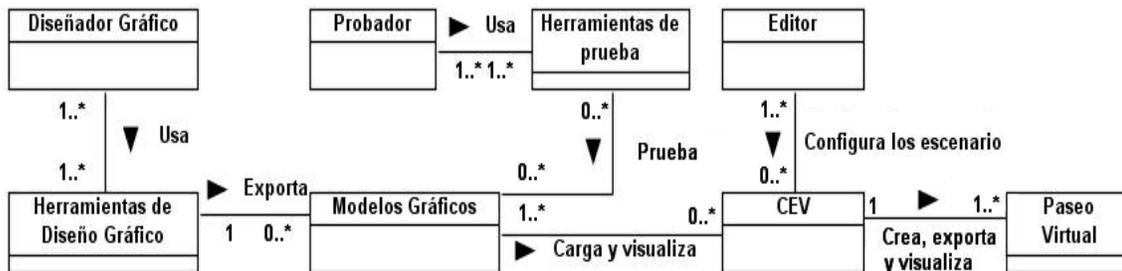
herramienta CEV a este proceso de desarrollo de PVI con elementos en tres dimensiones, cambia considerablemente la representación de los conceptos de la etapa de implementación [54]. Ver la figura 11.



**Figura 11. Representación del proceso de implementación de un PVI en el CEDIN.**

Dicho esto se puede analizar la figura anterior dividiéndola en tres capas desde arriba hacia abajo. En la capa superior se encuentran los especialistas que deben poseer habilidades bien definidas en su trabajo. En la segunda capa se encuentran las herramientas que cada especialista utiliza para obtener los elementos conceptuales que se encuentran representados en la tercera capa. Los Diseñadores Gráficos mediante las herramientas de diseño gráfico, como Blender o 3DsMax, crean los modelos y con ellos editan la escena. Los Desarrolladores haciendo uso de los IDEs y de los Motores de Juego logran codificar los componentes necesarios para realizar la lógica de la aplicación y manipular los componentes gráficos, también son los encargados de compilar la solución para obtener un paseo virtual. Por su parte los Probadores usan diferentes herramientas para probar los elementos gráficos (modelos y el escenario) así como los componentes codificados.

En la figura 12 se observan los resultados que se obtienen después de realizados los cambios al proceso explicado anteriormente. Los conceptos que se manejan se corresponden con los descritos en el epígrafe 1.5.1. En este caso se incorpora el elemento “CEV” que representa a la herramienta propuesta [55].



**Figura 12. Propuesta de modificación al proceso de implementación de un PVI.**

Varios conceptos se ven afectados:

Se cambia totalmente:

- El Desarrollador por Editor, esto se debe a que la actividad de codificar se reduce a cero, ya que desde la interfaz de la herramienta se facilita la manipulación de los elementos gráficos y la creación del paseo virtual. El nuevo concepto Editor será el encargado de conformar el escenario con los objetos necesarios y exportar el paseo virtual que estará disponible para visualizar desde el módulo de visualización y para realizar cualquier modificación en el momento que el usuario desee.

Desaparecen:

La desaparición de los tres conceptos que se relacionan a continuación se debe a que con la propuesta la necesidad de implementar los componentes desaparece, como se explicó anteriormente.

- IDE
- Motor de Juego
- Componentes

Se reducen las responsabilidades de:

- El Diseñador Gráfico, se centrará en la creación de los modelos. la edición de escenarios quedaría entre las funciones del Editor, quien no necesariamente debe tener habilidades en el uso de las herramientas de diseño.
- El Probador, solo realizará labores de prueba para garantizar la calidad de los modelos gráficos del paseo virtual, ya que la lógica y manipulación de los gráficos se garantizan internamente por CEV.

### **2.3 Consideraciones parciales**

- Al concebir, diseñar e implementar una herramienta para crear, editar y visualizar de forma intuitiva paseos virtuales con elementos en tres dimensiones, se facilita a personas inexpertas la configuración personalizada de sus escenarios
- Simplificar la etapa de implementación dentro del proceso de desarrollo de paseos virtuales interactivos con elementos en tres dimensiones, sugiere una reducción en los tiempos de entrega y disminuye la dependencia de especialistas.

### Capítulo 3: Resultados y discusión

#### Introducción

En el presente capítulo se define en qué consiste la nueva entidad informática denominada Centro Expositivo Virtual (CEV), se hace una breve comparación entre CEV y algunas de las herramientas estudiadas, y se analizan los comportamientos de los posibles usuarios frente a la herramienta. Igualmente se describe cómo afecta el uso de CEV a la etapa de implementación de paseos virtuales dentro del proceso de desarrollo usado por el GDPV del CEDIN.

#### 3.1 Centros Expositivos Virtuales

En la bibliografía consultada no se encuentran referencias al término Centros Expositivos Virtuales (CEV). Teniendo en cuenta algunas de las acepciones que brinda la Real Academia de la Lengua Española para cada una de las partes que conforman la expresión, se encuentra:

Centro:

- Punto donde habitualmente se reúnen los miembros de una sociedad o corporación.
- Instituto dedicado a cultivar o a fomentar determinados estudios e investigaciones.
- Lugar en que se desarrolla más intensamente una actividad determinada.

Exposición:

- Presentación pública de artículos de industria o de artes y ciencias, para estimular la producción, el comercio o la cultura.

Virtual:

- Que tiene existencia aparente y no real [56].

Una galería de arte es un espacio destinado a la exhibición de obras de valor económico, cultural o patrimonial y un museo exhibe, conserva, investiga, comunica y adquiere, con fines de estudio, educación y disfrute, la evidencia material de la gente y su medio ambiente. Sergio Talens y José Hernández entienden los museos virtuales como una réplica de los museos tradicionales pero en soporte electrónico: *"Los museos virtuales reciben fundamentalmente esta denominación porque suelen copiar los contenidos de algún otro museo real, siguen la obra de algún artista o tratan un tema especial. Aunque los museos virtuales no reemplazarán nunca las visitas físicas para ver los originales de obras históricas para la humanidad, cuando la distancia o las*

*posibilidades económicas no permiten ir, siempre pueden ser una opción muy válida [57].*

Por todo lo anteriormente expuesto, se define como Centro Expositivo Virtual (CEV) a una entidad informática caracterizada por ser: *“Entidad informática, para usuarios inexpertos en diseño e implementación de paseos virtuales interactivos con elementos en tres dimensiones, que permite administrar y recrear virtualmente un Centro Expositivo a partir de la gestión de los recursos disponibles”*

Desde los CEV's, se pueden crear paseos virtuales y al igual que desarrollados independientemente, recrean lugares que en su interior exponen objetos de valor, abiertos al servicio y desarrollo del conocimiento artístico, histórico, estético, natural y social de la población. Pero al ser modificables por los clientes finales, pueden adaptarse a situaciones nuevas y brindar más opciones de aprendizaje al público que acceda a ellos.

### **3.2 Análisis de la herramienta CEV**

La creación de la herramienta CEV es producto de un trabajo de investigación guiado por la metodología de desarrollo RUP, la documentación generada incluye la definición de un grupo de requisitos a cumplir por una herramienta enfocada a realizar paseos virtuales interactivos con elementos en tres dimensiones. En conjunto, toda la documentación obtenida permite a los programadores la comprensión de la idea general y en particular el código fuente, lo que facilita el trabajo para ampliar o actualizar la herramienta. Un aspecto importante de CEV es la sencillez desde el punto de vista gráfico de la interfaz que ofrece. Mediante íconos conocidos y textos asociados a ellos, se responde a las necesidades básicas de edición del escenario y en todos los casos con no más de dos clics. Estas características reducen la posibilidad de que un usuario inexperto demore en identificar las funciones que necesita para editar su escenario. El menú “Proyectos” y la posibilidad de selección múltiple, son algunas de las opciones útiles para los “editores” ya que pueden crear “paquetes de visualización” con uno o varios paseos virtuales. Estos paquetes se cargan por el módulo “Visualizador”, que permite navegar de forma intuitiva por cualquiera de los paseos virtuales contenidos en el paquete de visualización. Permite

además la administración desde la interfaz de los proyectos en los que se han trabajado previamente, sin tener que abrirlos o buscar la posición donde se encuentran almacenados. Con esta herramienta, debe reducirse el número de programadores en el proceso de desarrollo de un paseo virtual, lo que debe contribuir a la agilización de dicho proceso. De forma general, la creación de la herramienta y del formatos \*.cev, contribuye en alguna medida a lograr la independencia tecnológica.

### **3.3 Comparación entre algunas de las herramientas estudiadas y CEV**

Para valorar cualitativamente las implicaciones de la introducción de la herramienta CEV en la creación de nuevos CEV's se realizó un estudio comparativo entre varias de las herramientas estudiadas en la fundamentación teórica.

Se emplearon como herramientas de comparación: A) Cryengine Sandbox, B) Warcraft III World Editor, C) Unreal Level Editor, D) Autodesk Maya, E) Autodesk 3D Studio Max, F) Blender y G) CEV. Como criterios de comparación se consideraron:

1. Facilidad para mover, escalar y rotar objetos
2. Libertad para la creación y deformación de objetos.
3. Facilidad para Importar objetos realizados en otras herramientas.
4. Ver objetos de la escena.
5. Posibilidad de exportar escenas para ser cargadas desde aplicaciones no asociadas.
6. Conveniencia de poseer un formato de fichero propio para el almacenamiento de escenas.
7. Capacidad para crear paquetes de Visualización.
8. Capacidad para crear y visualizar paseos virtuales interactivos con elementos en 3D.
9. Libres.

## Capítulo 3. Resultados y discusión

En la tabla 2 se presentan los resultados de la comparación entre las herramientas. En el bloque conformado por los cinco primeros criterios se muestra una alta coincidencia, lo que indica similitudes en cuanto a las funciones generales para editar la escena, sin embargo en el criterio dos en la herramienta CEV se evalúa como “Baja”, esto se debe a que en el diseño se tuvo en cuenta que el usuario dispusiera solo de las herramientas necesarias para editar su escenario, con lo que se reduce la complejidad y la posibilidad de que pueda dañar los objetos.

<b>Criterios:</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>
<b>1</b>	Si						
<b>2</b>	Media	Media	Media	Alta	Alta	Alta	Baja
<b>3</b>	Si						
<b>4</b>	Si						
<b>5</b>	Si	Si	Si	Si	Si	Si	No
<b>6</b>	Si						
<b>7</b>	No	No	No	No	No	No	Si
<b>8</b>	No	No	No	No	No	No	Si
<b>9</b>	No	No	No	No	No	Si	Si

**Tabla 2 Comparación de algunas herramientas.**

En el criterio seis hay una coincidencia del 100%, lo cual puede indicar la tendencia a crear formatos propios que respondan a características específicas de cada aplicación. En el séptimo y el octavo criterio la coincidencia es muy pequeña, ya que el resto de las herramientas no cuentan con la capacidad de exportar paquetes de visualización ni están diseñadas para realizar paseos virtuales.

En el criterio nueve, que indica la importancia de si la herramienta es libre o no, solo Blender cumple con este criterio. Sin embargo aunque cumpla con este criterio cualquier herramienta puede en un momento determinado dejar de ser libre o perder el soporte, lo que significa un riesgo. Por estas razones el desarrollo de aplicaciones propias, que en alguna medida garanticen la soberanía tecnológica, es una alternativa mucho más segura.

### 3.4 CEV desde la perspectiva del editor de centros expositivos

CEV no está dirigida solamente a informáticos, sino también a los responsables, encargados o editores de áreas expositivas. Teniendo en cuenta que estas personas no tienen que dominar lenguajes de programación o herramientas de diseño gráfico, y como con el uso de la herramienta propuesta se pretende que puedan editar sus propios escenarios, se hizo necesario conocer el criterio de potenciales usuarios, sobre la herramienta. Con este objetivo, se realizó una encuesta del tipo *entrevista personal*, en la que se redactaron varias preguntas clasificadas según la respuesta como: *cerradas y categorizadas por valoración* y según su contenido, de *opinión* [58]. Se seleccionó una muestra integrada por 20 estudiantes de primer año de la Facultad 5, con características similares a los posibles usuarios de la herramienta. A los encuestados se les presentó el siguiente cuestionario:

#### 3.4.1 Cuestionario

*Existen varias herramientas para la creación, edición de modelos gráficos y escenarios virtuales.*

*En la presente encuesta se le mostrarán las herramientas Blender, CEV y Autodesk 3D Studio Max. Usted deberá identificar dentro de las herramientas varias características y evaluar de 2 a 5 según su criterio. (Leyenda: 5- Muy bueno, 4-Bueno, 3-Regular, 2-Malo)*

Preguntas:	Blender	CEV	3DsMax
Identificar la funcionalidad "Mover objeto"	3,75	4,95	4,75
Identificar la funcionalidad "Escalar objeto"	3,80	4,95	4,50
Identificar la funcionalidad "Rotar objeto"	3,90	4,90	4,90
Identificar la funcionalidad "Ver objetos de la escena"	3,90	4,75	3,50
Más fácil de usar	3,30	4,95	4,10
Presentación	4,60	4,35	4,90

**Tabla 3. Encuesta a estudiantes.**

En la tabla 3 se muestran las calificaciones promedio de la encuesta realizada a los 20 estudiantes. De un total de seis preguntas, las cuatro primeras se dirigieron a que se calificara la edición de la escena a partir de la facilidad de encontrar cuatro funcionalidades básicas. Una quinta se orientó de forma directa a que el encuestado calificara las facilidades de cada una de las herramientas de forma general; el objetivo de esa pregunta era el de contraponer los resultados a las otras cuatro esperando que no existiera contradicción. La sexta se dirigió también de forma directa para conocer la opinión acerca de la presentación.

### 3.4.2 Análisis de los resultados de la encuesta

En la Tabla 4 se muestra el promedio de los resultados de las respuestas a las diferentes preguntas de la encuesta, agrupados en los tres objetivos principales que se perseguían.

<b>Objetivos</b>	<b>Blender</b>	<b>CEV</b>	<b>3DsMax</b>
Presentación	4,60	4,35	4,90
Edición de escena	3,84	4,89	4,41
Usabilidad	3,30	4,95	4,10

**Tabla 4. Resultados de encuesta por objetivos.**

Del análisis de los resultados se infiere que:

- “Presentación”: son desfavorables a CEV. Esto se corresponde con la cantidad de personas, tiempo y recursos que se dedican a desarrollar las restantes.
- “Edición de escena”: son favorables a CEV, debido fundamentalmente a que solamente muestra en la interfaz las funciones básicas, mediante íconos con textos asociados que anuncian la función que realiza el botón.
- “Usabilidad”: son favorables a CEV. Estos resultados se basan en los criterios obtenidos con las preguntas anteriores, aunque de forma más más general.

Si se analizan las diferencias entre los resultados del objetivo *Edición de escena* y *Usabilidad*, no se encuentran contradicciones apreciables en una misma aplicación, pero se observan valores con diferencias de 0,54, 0,06 y 0,31 respectivamente. La pequeña diferencia obtenida en CEV permite afirmar que, en la percepción de los usuarios esta herramienta facilita la edición de los escenarios, debido principalmente a

la reducción de funcionalidades en comparación con las otras. Es indiscutible que el mayor número de funcionalidades implica una potencialidad también mayor, pero a su vez, una mayor complejidad operacional. Dado que el objetivo del trabajo es la creación de una herramienta que cumpla el requisito fundamental de ser útil a usuarios inexpertos, puede afirmarse que este se ha alcanzado.

En la figura13 se presentan estos resultados en forma de gráfica de barras para una rápida visualización de los mismos.

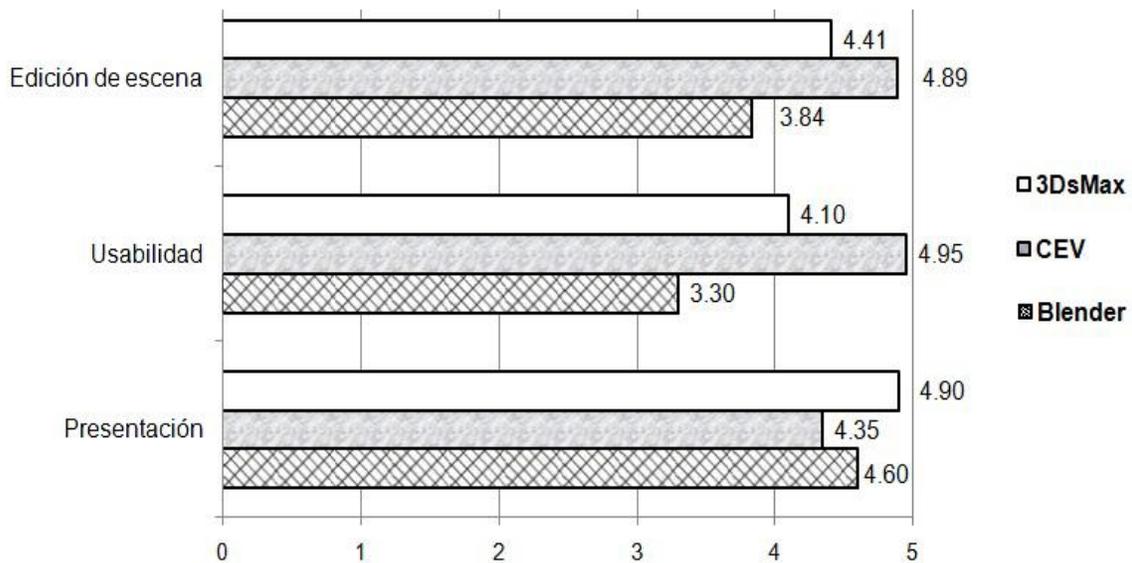
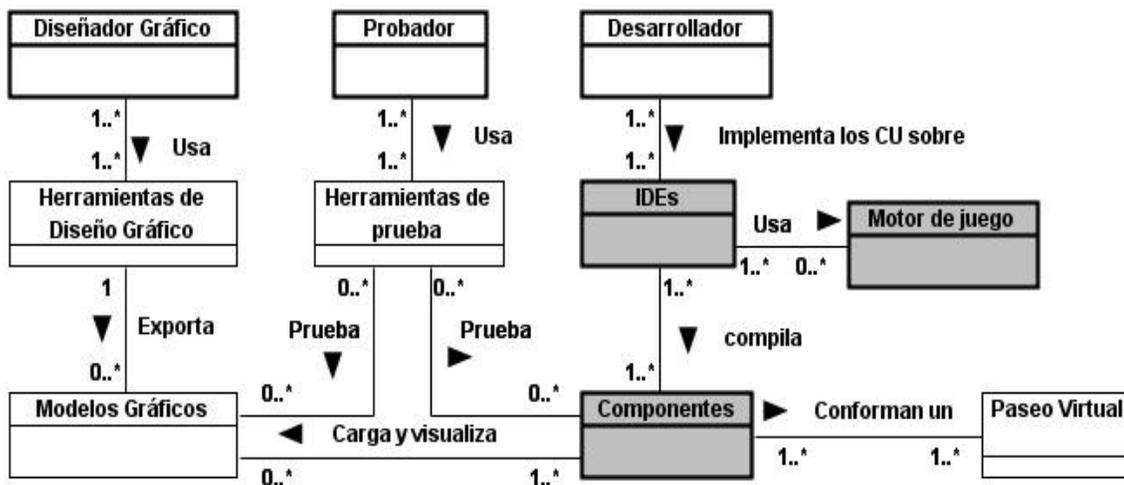


Figura 13. Gráfica de los resultados de la encuesta por objetivos.

### 3.5 CEV desde la perspectiva del informático

Con la integración de la herramienta CEV a la etapa de implementación del proceso de desarrollo usado en el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas, para obtener paseos virtuales interactivos con elementos en tres dimensiones, la representación que se muestra en la figura 11, cambia considerablemente respecto a la que se muestra en la figura 12. Por tanto en la figura 14 se muestran con borde grueso y sombreados los conceptos que desaparecen totalmente y con un borde grueso y fondo blanco, los que se modifican parcial o totalmente.



**Figura 14. Análisis de las modificaciones a los conceptos y relaciones entre ellos.**

Con esta propuesta, la etapa de implementación del proceso de desarrollo usado en el Centro de Informática Industrial para obtener un paseo virtual se simplifica. El análisis que se sigue para llegar a este punto se describe a continuación:

- Desaparecen totalmente conceptos como: “IDE”, “Motor de Juego” y “Componentes” debido a que no es necesario codificar ningún componente informático para el manejo de gráficos, lógica del sistema u otra función, para conformar la aplicación. Por esta razón, la necesidad de contar con especialistas en algún lenguaje o técnica de programación gráfica pudiera reducirse hasta cero. Al eliminar la codificación de componentes para conformar un paseo virtual, la posibilidad de introducir errores de código desaparece, lo que pudiera contribuir a la obtención de un producto final con calidad.
- El “Desarrollador”, que anteriormente realizaba parte de la implementación del sistema, ahora se cambia por el “Editor” y solo editará con elementos gráficos los escenarios y exportará los paseos virtuales o paquetes de visualización creados.
- El uso del editor de CEV, permite a una persona sin preparación en diseño gráfico asumir la edición de los escenarios. Esto limita al “Diseñador gráfico” a crear los modelos que conforman la escena.
- Las actividades del “Probador” también se ven disminuidas ya que no se realizan componentes y centraría su trabajo en los elementos gráficos y en el producto final.

- Desde el punto de vista de los desarrolladores del GDPV del CEDIN, se elimina la necesidad de contar con especialistas en lenguajes y técnicas de programación gráfica. Los especialistas en diseño gráfico se centrará en la creación de los modelos, por tanto la edición de escenarios quedaría entre las funciones del Editor, quien no necesariamente debe tener habilidades en el uso de las herramientas de diseño. Esto garantiza que los Diseñadores Gráficos solo realizarán la mitad de las tareas que cumplían, lo que permite al igual que con los “desarrolladores”, utilizar estos recursos humanos en la realización de otros compromisos productivos del CEDIN o de la Universidad de las Ciencias Informáticas en general. Al simplificar la implementación y el diseño gráfico, es probable que los tiempos para terminar un producto se reduzcan. Cabe aclarar, que este resultado debe ser comprobado mediante la aplicación de los métodos necesarios, pero un estudio de así rebasa el alcance de la presente investigación.

### **3.6 Consideraciones parciales**

- El análisis de los resultados de la encuesta sugiere que el módulo “Editor” de la solución propuesta facilita la edición de escenario, ya que los encuestados identifican de forma más rápida las funcionalidades necesarias para realizar esta actividad.
- Mediante el análisis de los cambios ocurridos con la incorporación de la herramienta CEV en el proceso de desarrollo de paseos virtuales usado por el CEDIN, se muestra la posibilidad de que se disminuya a la mitad la cantidad de actividades necesarias para implementar un paseo virtual, lo que sugiere además una disminución en los tiempos de desarrollo y en el costo final del producto.

### Conclusiones

Terminada la actual investigación, basada en el estudio de los editores de escenarios, las herramientas de diseño gráfico, los diferentes tipos de paseos virtuales, las técnicas y tecnologías para desarrollarlos, y tomando en cuenta la experiencia del autor y su equipo en el desarrollo de paseos virtuales se concluye lo siguiente:

- Al concebir, diseñar e implementar una herramienta para crear, editar y visualizar de forma intuitiva Centros Expositivos Virtuales con elementos en tres dimensiones, se facilita a personas inexpertas la configuración personalizada de sus escenarios virtuales con elementos en tres dimensiones.
- La posibilidad de crear, exportar y posteriormente cargar paquetes de visualización desde una herramienta especializada, que permita la navegación intuitiva, garantiza la disponibilidad inmediata de los paseos virtuales deseados.
- Al valorar cómo influye la introducción de la herramienta propuesta dentro de la etapa de implementación del proceso de desarrollo de paseos virtuales usado en el Centro de Informática Industrial, se evidencia que simplifica la etapa de implementación dentro del proceso, lo que contribuye a reducir los tiempos de entrega y disminuir la dependencia de especialistas.
- Al eliminar la codificación de componentes para conformar un paseo virtual, la posibilidad de introducir errores se reduce.
- Diseñar un formato propio (\*.cev) para almacenar los datos manejados por la herramienta, permitió incorporar elementos particulares de los Centros Expositivos Virtuales como las clasificaciones de los objetos del escenario, los datos asociados a los objetos expositivos y la definición de recorridos.

### **Recomendaciones**

- Desarrollar una metodología para la creación y administración de paseos virtuales generales.
- Realizar una valoración sobre la posible reducción de tiempo y costo que representa la generalización de la aplicación.
- Mantener la investigación y la vigilancia tecnológica sobre el desarrollo de paseos virtuales interactivos con elementos en tres dimensiones, para identificar e incluir nuevas características como la exportación de los paquetes de visualización a otros formatos estándares.

### Referencias bibliográficas

1. **Levis, Diego.** Diego Levis. Comunicación & Educación. *¿Qué es la realidad virtual?* [Online] Copyright © Diego Levis- 1997/2006.

[www.diegolevis.com.ar/secciones/Articulos/Que\\_es\\_RV.pdf](http://www.diegolevis.com.ar/secciones/Articulos/Que_es_RV.pdf). NoComercial 2.5 Buenos Aires, Argentina.

2. **Dirección de Patrimonio Cultural.** Museos y Casas Museos. *Palacio de los capitanes generales y Tempete.* [En línea] Oficina del Historiador de la Ciudad de La Habana, 2010. [Citado el: 2011 de Noviembre de 20.]

[www.ohch.cu/\\_menu\\_1/info.php?cat=Museosycasasmuseos&iditem=56&id\\_Cat=2](http://www.ohch.cu/_menu_1/info.php?cat=Museosycasasmuseos&iditem=56&id_Cat=2).

3. **Dailymotion.** Dailymotion.com. *Santiago de Cuba - Virtual Tour (2da Parte).* [En línea] 2011. [Citado el: 20 de Noviembre de 2011.]

[www.dailymotion.com/video/x8s5pq\\_santiago-de-cuba-virtual-tour-2da-p\\_travel](http://www.dailymotion.com/video/x8s5pq_santiago-de-cuba-virtual-tour-2da-p_travel).

4. **Acuario Nacional de Cuba.** *Acuario Nacional de Cuba.* [En línea] 2011. [Citado el: Noviembre de 20 de 2011.] [www.acuarionacional.cu/visita-virtual/visita-virtual-flash.htm](http://www.acuarionacional.cu/visita-virtual/visita-virtual-flash.htm).

5. **360 Cities.** 360cities.net. [En línea] 2011. [Citado el: 20 de Noviembre de 2011.] [www.360cities.net/area/cuba](http://www.360cities.net/area/cuba).

6. **Dictionary.com LLC's.** Science Dictionary. *Virtual reality.* [En línea] Dictionary.com, LLC, 2011. [Citado el: 28 de 11 de 2011.]

<http://dictionary.reference.com/browse/Virtual+Reality>.

7. **Baker , M.Pauline.** Access Science. *Virtual Reality.* [En línea] The McGraw-Hill Companies, 2011. [Citado el: 28 de 11 de 2011.]

<http://accessscience.com/abstract.aspx?id=757461&referURL=http%3a%2f%2faccessscience.com%2fcontent.aspx%3fid%3d757461>.

8. **Cordero López, Martha Rosa.** *Simulación de escenarios naturales tridimensionales.* [pdf] México, D.F. : Instituto superior politécnico nacional. Centro de innovación y desarrollo tecnológico en cómputo., 2008.

9. **Wells, Don.** extremeprogramming. *Extreme Programming: A gentle introduction*. [En línea] 8 de Noviembre de 2009. [Citado el: 25 de Enero de 2011.]

[www.extremeprogramming.org/](http://www.extremeprogramming.org/).

10. **IBM.** developerWorks. *www.ibm.com*. [En línea] 2011. [Citado el: 3 de Febrero de 2011.]

[www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf).

11. **Rado Naranjo, Lisandra Zaylín.** *Análisis y Diseño de una herramienta para la creación, edición y visualización de Centros Expositivos Virtuales*. La Habana : s.n., 2011.

12. **Autodesk.com.** Autodesk. *Autodesk 3ds Max Products*. [En línea] 2011. <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13567426>.

13. **Autodesk.com.** Autodesk Maya. [En línea] Autodesk, 2011.

<http://usa.autodesk.com/maya/features/>.

14. **Blender Foundation.** Blender. [En línea] Blender Foundation, 23 de Marzo de 2011. [Citado el: 23 de Marzo de 2011.] <http://www.blender.org/>.

15. **Dobarro Ansedo, Alejandro, González de Canales, Eva y Barros Castro, Jesús.** *Sistemas Adaptativos y Bioinspirados en Inteligencia Artificial. Tutorial Blender*. [En línea]

<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Tutoriales/blender/uno.html>.

16. **G3D Innovation Engine.** G3D Innovation Engine. [En línea] 2011. <http://g3d.sourceforge.net/>.

17. **Torus Knot Software Ltd.** Ogre3D Home page. [En línea] 2009. [www.ogre3d.org](http://www.ogre3d.org).

18. **Camacho Román, Yanoski Rogelio.** Biblioteca de la Universidad de las Ciencias Informáticas. *Optimización de la visualización en la herramienta gráfica SceneToolKit*. [En línea] junio de 2009.

[http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_2806\\_09/1/TD\\_2806\\_09.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_2806_09/1/TD_2806_09.pdf).

19. **Quintana López, A y Alonso Monteagudo, Y.** *Motor de Render Genérico*. Ciudad de la Habana : s.n., 2010.
20. **E. U. I. T. I. O.** Sitio web de la E.U. Ingeniería Técnica en Informática de Oviedo. *Entornos de Desarrollo Integrado*. [En línea] 14 de Enero de 2009. <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
21. **Microsoft.** Introducción a Visual Studio. *MSDN*. [En línea] 2011. [http://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=VS.80\).aspx](http://msdn.microsoft.com/es-es/library/fx6bk1f4(v=VS.80).aspx).
22. **Nokia Corporation and/or its subsidiaries.** Qt-Cross-plataform application and UI framework. *Qt Creator IDE and tools*. [En línea] 2011. <http://qt.nokia.com/products/developer-tools>.
23. **The Eclipse Foundation.** Eclipse. [En línea] 2011. [www.eclipse.org/downloads/moreinfo/c.php](http://www.eclipse.org/downloads/moreinfo/c.php).
24. **Schmuller, Joseph.** *Aprendiendo UML en 24 Horas*. México : Prentice Hall, 2000. 968-444-463-X.
25. **RedIRIS.** forja.rediris.es. [En línea] RedIRIS, 29 de 3 de 2008. [Citado el: 20 de Junio de 2011.] <http://forja.rediris.es/docman/view.php/282/444/uml20.pdf>.
26. **Nokia.** Qt Creator IDE and tools. [En línea] Nokia Corporation and/or its subsidiaries, 2011. [Citado el: 23 de Junio de 2011.] <http://qt.nokia.com/products/developer-tools/>.
27. **Stroustrup, Bjarne.** *A History of C++: 1979–1991*. New Jersey : AT&T Bell Laboratories, 2005.
28. **Informatica Full.** Blog de informatica para todos los que deseen aprender mas sobre informática, programacion, redes, etc. *Historia del lenguaje C++*. [En línea] Informatica Full, 16 de 6 de 2009. <http://informatica-full2.blogspot.com/2009/06/historia-del-lenguaje-c.html>.

29. **González Duque, Raúl.** Mundo Geek. *El lenguaje de los grandes programadores.* [En línea] 2008. <http://mundogeek.net/archivos/2011/05/08/el-lenguaje-de-los-grandes-programadores/>.
30. **González Duque, Raúl .** Universidad Nacional de San Luis, Argentina. *Python para todos.* [En línea] 6 de Mayo de 2008. [Citado el: 24 de Enero de 2011.] [www.dirinfo.unsl.edu.ar/jornadas/img/ebooks/pythonparatodos.pdf](http://www.dirinfo.unsl.edu.ar/jornadas/img/ebooks/pythonparatodos.pdf).
31. **Giantbomb.** *In-Game Map Editor.* [En línea] 12 de 2008. [www.giantbomb.com/in-game-map-editor/92-269](http://www.giantbomb.com/in-game-map-editor/92-269).
32. **Crytek GmbH.** SandBox 2 Manual. [En línea] Crytek GmbH, 2008. <http://doc.crymod.com/SandboxManual/frames.html?frmname=topic&frmfile=index.html>.
33. **Jelsoft Enterprises Ltd.** hiveworkshop.com. [En línea] Jelsoft Enterprises Ltd, 25 de 8 de 2010. [www.hiveworkshop.com/forums/tutorials](http://www.hiveworkshop.com/forums/tutorials).
34. **Epic Games, Inc.** UnrealTechnology. *Unreal Editor Features.* [En línea] 2011. [Citado el: 5 de Diciembre de 2011.] [www.unrealengine.com/features/editor](http://www.unrealengine.com/features/editor).
35. **University of Meryland.** Department of Computer Sciences. *Binary vs. ASCII.* [En línea] University of Meryland, 2011. [www.cs.umd.edu/class/spring2003/cm311/Notes/BitOp/asciiBin.html](http://www.cs.umd.edu/class/spring2003/cm311/Notes/BitOp/asciiBin.html).
36. **Quickparts.** Learning Center. *What Is An STL File?* [En línea] Quickparts, 2012. [www.quickparts.com/LearningCenter/WhatIsAnSTLFile.aspx](http://www.quickparts.com/LearningCenter/WhatIsAnSTLFile.aspx).
37. **FileInfo.** FileInfo.com. *.3DS File Extension.* [En línea] FileInfo.com, 22 de 8 de 2011. [www.fileinfo.com/extension/3ds](http://www.fileinfo.com/extension/3ds).
38. **Blender Foundation.** Blende.org. *Blender Architecture Overview.* [En línea] Blender Foundation, 2011. <http://wiki.blender.org/index.php/Dev:Source/Architecture/Overview>.
39. **University of Washington.** University of Washington. *ASCII Format Data Files.* [En línea] University of Washington, 2011. <http://depts.washington.edu/uwxafs/Docs/fileh/fileh-2.html>.

40. R. Geoff Dromey. [book auth.] *Cornering the Chimera*. s.l. : IEEE Software, 1996, pp. 33-43.

[www.computer.org/csdl/mags/so/1996/01/s1033.html](http://www.computer.org/csdl/mags/so/1996/01/s1033.html)

41. **Joswig, Michael y Polthier, Konrad**. EG-Models. *Wavefront and Java3D .obj Format*. [En línea] Noviembre de 2010. [Citado el: 22 de Noviembre de 2011.]

[www.eg-models.de/formats/Format\\_Obj.html](http://www.eg-models.de/formats/Format_Obj.html).

42. **Rypl, Daniel**. Czech Technical University in Prague. *STL File Format*. [En línea] Czech Technical University in Prague, 11 de 2005. <http://mech.fsv.cvut.cz/~dr/papers/Lisbon04/node2.html>.

43. **DownloadAtoZ**. DownloadAtoZ File Extension. *File Extension ASE*. [En línea] DownloadAtoZ, 2011. [Citado el: 6 de Diciembre de 2011.] [www.downloadatoz.com/file-extensions/ase-file-extension.html](http://www.downloadatoz.com/file-extensions/ase-file-extension.html).

44. **Torus Knot Software Ltd**. Ogre3d. *DotScene*. [En línea] Torus Knot Software Ltd, 7 de Septiembre de 2010. [www.ogre3d.org/tikiwiki/DotScene](http://www.ogre3d.org/tikiwiki/DotScene).

45. **González López, Minardo Gollún y Rado Naranjo, Lisandra Zaylín**. *Centros\_Expositivos\_Virtuales: Diagramas UML*. [.vpp] La Habana : s.n., 2011.

46. **Alexander , Ian F. y Stevens, Richard** . *Writing Better Requirements*. Reino Unido : Addison Wesley, 2002. 0-321-13163-0.

47. **Sommerville, Ian**. *Software engineering (8th edition ed.)*. 2007. ISBN 978-0-321-31379-9 ISBN 0-321-31379-8..

48. **Fenton, N y Pfleeger, S**. *Software Metrics: A Rigorous & Practical Approach. Second Edition*. s.l. : PWS Publishing Company, 1997.

49. **Pressman, Roger S**. *Ingeniería del Software. Un enfoque práctico. Quinta edición*. España : McGraw-Hill, 2001. 8448132149.

50. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady**. *El lenguaje unificado de modelado. Manual de referencia*. España : Addison Wesley, 2000. 8478290370.

51. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El proceso unificado de desarrollo de software*. España : Addison Wesley, 2000. 84-7829-036-2.

52. **Molina Ramírez, Enier y González López, Minardo Gollún.** *Implementación de un módulo de edición de escena 3D para una herramienta de creación de Centros Expositivos Virtuales.* La Habana : s.n., 2011.
53. **Torres Utra, Andy y González López, Minardo Gollún.** *Implementación del módulo de persistencia y visualización para una herramienta de creación de Centros Expositivos Virtuales.* La Habana : s.n., 2011.
54. **González López, Minardo Gollún y Rado Naranjo, Lisandra Zaylín.** *Centros\_Expositivos\_Virtuales: Etapa de Implementación del proceso usado en el CEDIN.* [.vpp] La Habana : s.n., 2011.
55. **González López, Minardo Gollún y Rado Naranjo, Lisandra Zaylín.** *Centros\_Expositivos\_Virtuales: Propuesta usando CEV.* [.vpp] La Habana : s.n., 2011.
56. **Real Academia Española.** *Diccionario de la lengua española - Vigésima segunda edición.* [En línea] 2010. [Citado el: 25 de Noviembre de 2011.] <http://buscon.rae.es/drae/>.
57. **Talens, Sergio y Hernández, José.** *Internet: redes de computadores y sistemas de información.* Madrid : Paraninfo, 1996. 8428323348.
58. **Universidad Champagnat.** La encuesta y tipos de encuesta. *GestioPolis.* [En línea] 2002. [www.gestiopolis.com/recursos/documentos/fulldocs/mar/tipencuch.htm](http://www.gestiopolis.com/recursos/documentos/fulldocs/mar/tipencuch.htm).

## Anexo 1: Muestra de los casos de pruebas diseñados

**Editor:**

**Diseño del caso de prueba. CU Gestionar modelos del sistema.**

**Descripción general:**

El diseñador del paseo inicializa el caso de uso cuando selecciona la opción adicionar modelo al sistema. Antes de eliminar un modelo del sistema se pregunta al usuario si realmente está seguro de eliminarlo definitivamente del sistema.

**Condiciones de ejecución:**

No puede existir ningún modelo en el sistema con el mismo nombre del modelo que se va a adicionar.

**Secciones a probar en el caso de uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
<b>SC 1:</b> Adicionar modelo al proyecto.	<b>EC 1.1:</b> Adicionar modelo nuevo al proyecto.	El usuario da clic sobre la opción de adicionar modelo al proyecto. El sistema abre un explorador para que el usuario busque el modelo. Da clic sobre el botón aceptar y el sistema agrega el modelo como un recurso gráfico del proyecto.	Seleccionar opción adicionar modelo.  Buscar el modelo.  Adicionar modelo al proyecto.
	<b>EC 1.2:</b> Adicionar modelo existente al proyecto.	Al seleccionar un modelo que ya existe en el proyecto, el sistema dará la opción de adicionarlo al sistema con un nuevo nombre, que será el nombre del modelo seguido de un “_” y un número para diferenciarlo del ya existente.	Seleccionar modelo ya existente en el proyecto.  Adicionar modelo con nombre nuevo.
<b>SC 2:</b>	<b>EC 2.1:</b>	El usuario selecciona la opción eliminar	Seleccionar opción

Eliminar modelo del proyecto.	Eliminar modelo del proyecto.	modelo, el sistema pide confirmación de la eliminación y elimina el modelo del proyecto.	eliminar modelo. El sistema elimina el modelo del proyecto.
-------------------------------	-------------------------------	--	--

**Tabla 5. Diseño de caso de prueba. CU Gestionar modelos del sistema.**

**Diseño de caso de prueba. CU Inicializar Datos.**

**Descripción general:**

El usuario inicializa el caso de uso cuando ejecuta el Visualizador. El Visualizador carga los paseos previamente creado en el módulo Editor que se encuentren en la carpeta de recursos de la aplicación y los lista para posibilitar la selección del paseo que desee visualizar el usuario.

**Condiciones de ejecución:**

El paseo debe estar creado con el módulo Editor.

**Secciones a probar en el caso de uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
<b>SC 1:</b> Inicializar Datos	<b>EC 1.1:</b> Inicializar Datos	Al ejecutarse el sistema, este carga y muestra los proyectos que actualmente tiene en la carpeta de proyectos, lo que permite que el usuario abra cada uno de ellos.	Ejecutar producto

**Tabla 6. Diseño de caso de prueba. CU Inicializar Datos.**

## Anexo 2: Glosario de términos

- **Aliasing:** Las líneas, especialmente las que están casi horizontalmente o verticales, aparecen dentadas o irregulares debido a su representación por píxeles. A este escalonamiento se le conoce como aliasing.
- **AntiAliasing:** Técnicas que se usan para reducir el Aliasing.
- **API:** (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- **ASCII:** American Standard Code for Information Interchange. Código normalizado en Estados Unidos para el intercambio de información. Forma de representar números, letras y otros símbolos, de manera que puedan ser introducidos en el ordenador a través del teclado.
- **Bézier:** superficies descritas por primera vez en 1972 por el ingeniero francés Pierre Bézier, quién las usó para diseñar cuerpos de automóviles. Las superficies de Bézier pueden ser de cualquier grado.
- **Clic:** Pulsación que se hace en alguno de los botones del ratón de un ordenador.
- **Escena:** Área virtual que contiene los modelos, luces, recorridos, cámara y todos los elementos necesarios para visualizar un paseo.
- **GDPV:** Grupo de Desarrollo de Paseos Virtuales.
- **Material:** Combinación de luces, colores e imágenes usados para definir una apariencia.
- **Modelo:** Representación abstracta, comúnmente estructurado por triángulos, colores y materiales que representan un objeto.
- **Paseos virtuales:** Aplicaciones de la realidad virtual, básicamente son la simulación de un entorno empleando tecnologías de realidad virtual.
- **Path:** Lista de vectores que definen una trayectoria a través de la escena.
- **Píxel:** Abreviatura de “picture element” es la menor unidad homogénea en color que forma parte de una imagen digital.
- **Proyección:** Conversión de coordenadas 3D del mundo a las coordenadas 2D de un plano.

- **Perspectiva:** Referente a “proyección perspectiva”, determina los tamaños de los objetos basándose en la distancia de los objetos al plano de proyección.
- **Plugins:** Del inglés "enchufable", es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.
- **Python:** Es un lenguaje de programación interpretado de alto nivel cuya filosofía hace hincapié en una sintaxis limpia y que favorezca un código legible. Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y en menor medida, programación funcional. Usa tipado dinámico y es multiplataforma.
- **Render:** Corresponde a la creación de una imagen 2D, de la representación de los modelos gráficos en una escena.
- **Shaders:** Procedimiento que permite al artista o programador especificar el render de un vértice (vertex) o de un pixel para manipular la iluminación, el sombreado o crear efectos gráficos como animaciones, efectos de partículas, etc.
- **Script:** Se refiere a archivos de órdenes o de procesamiento por lotes, es un programa usualmente simple, que frecuentemente se almacena en un archivo de texto plano.
- **Textura:** Imagen que sirve de “piel” a los modelos en un mundo virtual.