



Facultad 5

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Título: Plugin para exportar animaciones y mallas poligonales de Autodesk 3ds Max a Blender.

Autor: Yunior Frómeta Carbonell

Tutor: Ing. Minardo Gollún González López
Ing. Adriana Santos Lebeque

Cotutor: Ing. Manuel Alberto Avila Solarana

La Habana

Junio, 2012

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al tutor Minardo Gollón González López, a la tutora Adriana Santos Lebeque y al Centro de Informática Industrial (CEDIN), de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

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

Yunior Frómeta Carbonell

Firma del Autor

Ing. Minardo Gollón González López

Firma del Tutor

Ing. Adriana Santos Lebeque

Firma del Tutor

Ing. Manuel Alberto Avila Solarana.

Firma del Cotutor

Datos de Contacto:

Tutor

Nombre y apellidos: Minardo Gollún González López

Edad: 29 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente: Asistente

E-mail: mgonzalezl@uci.cu

Graduado de la UCI, con ocho años de experiencia en el tema de Realidad virtual, y jefe del proyecto Paseos Virtuales en el Centro de Desarrollo de Informática Industrial.

Tutor

Nombre y apellidos: Adriana Santos Lebeque

Edad: 26 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente:

Ingeniero en Ciencias informáticas, graduado en la Universidad de las Ciencias Informáticas, con 5 años de experiencia en el tema.

Tutor

Nombre y apellidos: Manuel Alberto Ávila Solarana

Edad: 25 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente:

E-mail: alebeque@uci.cu

Ingeniero en Ciencias informáticas, graduado en la Universidad de las Ciencias Informáticas, con 6 años de experiencia en el tema.

Agradecimientos:

Quisiera agradecer ante todo a mis padres, por ser tan especiales y dedicados conmigo.

Agradecer a mis amigos, Franklin, Yuri, Dariel, Lauren, Carlos, Yoan, Claudia, Parra, Lester, Gerandys y otros muchos que quizás ahora mismo no me acuerde, pero que siempre estarán presentes para mí. A los compañeros del lab, a los compañeros y amigos del apto, a los freakys del café, en fin a todos los que compartieron algún momento de sus vidas conmigo. Agradecer a los que me apoyaron cuando las cosas se me tornaron difíciles y me aconsejaron: Abel, David, Omar. A todos gracias.

Agradecer también a mis tutores, Adriana y Minardo, quienes han sido más que guías, gracias por su desinteresada amistad y ayuda. Gracias a mi cotutor Malbers por el apoyo ofrecido.

Agradecer a mi familia, mis abuelas, mis tías, mis primos, a todos por el amor y el cariño: gracias.

Por último, agradecer a Julián por haberme guiado profesionalmente en un principio y gracias a Blender Foundation por persistir.

Dedicatoria:

A mi papa, mi chiquitico, por ser el mejor padre del mundo y luchar por mí.

A mi mamá, mi nena, por siempre ser tan especial para mí y amarme tanto.

A los seres que más quiero, está dedicado este trabajo.

Resumen:

Entre los datos que se exportan de los software de creación de contenido 3D se destacan las animaciones y las mallas poligonales. Actualmente en el Instituto Cubano de Arte e Industria Cinematográfica (ICAIC) no realizan la exportación de animaciones y mallas poligonales del software privativo Autodesk 3ds Max al Software Libre Blender, ya que de forma manual genera atrasos en la producción debido al volumen de datos implicado y la posibilidad de introducir errores en los archivos creados, tales como pérdida de datos y deformaciones incorrectas en las mallas. Se carece de una forma rápida de exportar las animaciones y mallas poligonales de un software a otro.

La presente investigación tiene como objetivo la implementación de un *plugin* que permita exportar animaciones y mallas poligonales de 3ds Max a Blender, haciendo uso de ficheros con formatos tridimensionales (3D), aptos para el intercambio de información. Para la realización del trabajo se utilizará Programación Extrema como metodología de desarrollo de software y las siguientes herramientas: Visual Paradigm para UML y MAXScript como lenguaje de desarrollo.

Se tuvo como resultado un *plugin* que permite a 3ds Max exportar animaciones de sistemas basados en huesos, en *morphing* y simulaciones hacia Blender. Se permite además la configuración del nombre y directorio de los archivos exportados, así como la configuración de los parámetros del tiempo. Al incorporarle al 3ds Max este *plugin* para exportar los datos de animación y mallas poligonales hacia Blender, se logra un incremento en la rapidez de este proceso.

Palabras claves: animaciones, contenido 3D, exportar, mallas poligonales, *plugin*.

Índice

Introducción..... 1

Capítulo 1 . Fundamentación Teórica. 4

 Introducción..... 4

 1.1. Animación 3D. 4

 1.1.1. Animación de personajes 3D. 5

 1.1.2. Basado en huesos. 5

 1.1.3. Basado en Morphing..... 7

 1.1.4. Basado en Simulaciones. 7

 1.2. Herramientas de Animación 3D. 8

 1.2.1. Autodesk 3ds Max. 8

 1.2.2. Blender. 9

 1.2.3. Autodesk Maya. 10

 1.2.4. Lightwave. 11

 1.3. Formatos de almacenamiento 3D. 12

 1.4. Características de los formatos de almacenamiento 3D. 12

 1.4.1. 3D Studio (.3DS)..... 12

 1.4.2. Wavefront (OBJ). 12

 1.4.3. AutoCad (DXF). 13

 1.4.4. Lightwave object (LWO)..... 13

 1.4.5. Collada (DAE)..... 13

 1.5. Formatos de animación de vértice. 14

 1.5.1. Formato Motion Designer Data (MDD)..... 15

 1.5.2. Formato Point Cache 2 (PC2). 15

 1.6. Herramientas de para exportar Animaciones 3D en el software 3ds Max. 16

 1.6.1. XMesh. 17

 1.6.2. SuperMesher. 17

 1.6.3. Point Oven..... 18

 1.7. Metodologías de desarrollo de software..... 18

 1.7.1. Programación Extrema (XP). 18

 1.7.2. Rational Unified Process (RUP)..... 21

1.8.	Lenguajes de programación y herramientas para la implementación.....	22
1.8.1.	SDK de Autodesk 3ds Max.....	22
1.8.2.	MAXScript.....	22
1.8.3.	Python.....	23
1.9.	Herramienta CASE. Visual Paradigm para UML.....	24
	Consideraciones del capítulo.....	25
Capítulo 2 .	Solución propuesta.....	26
	Introducción.....	26
2.1.	Selección de los formatos 3D a utilizar.....	26
2.2.	Propuesta de la solución.....	27
2.3.	Modelo de dominio.....	28
2.4.	Requisitos funcionales.....	29
2.5.	Requisitos no funcionales.....	29
2.6.	Fase de Exploración.....	30
2.6.1	Historias de Usuario.....	30
2.7.	Fase de Planificación.....	33
2.7.1	Estimación de esfuerzo por Historia de Usuario.....	33
2.7.2	Plan de iteraciones.....	34
2.7.3	Plan de duración de las iteraciones.....	34
2.7.4	Plan de Entrega.....	35
2.8.	Fase de diseño.....	35
2.8.1	Arquitectura del sistema.....	36
	Consideraciones parciales.....	38
Capítulo 3.	Implementación y Prueba. Resultados de la propuesta.....	39
	Introducción.....	39
3.1	Fase de implementación.....	39
3.1.1	Iteración 1.....	39
3.1.2	Iteración 2.....	43
3.1.3	Descripción de las funciones y eventos.....	45
3.2	Fase de pruebas.....	47
3.2.1	Pruebas de aceptación.....	47

3.3 Resultados de la solución.....	51
Consideraciones parciales.....	53
Conclusiones.....	54
Recomendaciones.....	55
Referencias Bibliográficas.....	56
Anexos.....	59
Glosario de términos.....	63

Índice de Tablas.

Tabla 1. Estructura de la cabecera del formato Point Cache 2.....	16
Tabla 2. Selección de los formatos de intercambio a usar.....	27
Tabla 3. HU Exportar animación .pc2.....	31
Tabla 4. HU Exportar archivo Collada .dae.....	31
Tabla 5. HU Administrar la selección de objetos.....	32
Tabla 6. HU Configurar opciones de animación .pc2.....	32
Tabla 7. HU Configurar nombre y ruta de los archivos a exportar.....	33
Tabla 8. Estimación del esfuerzo por Historia de Usuario.....	34
Tabla 9. Plan de duración de las iteraciones.....	35
Tabla 10. Plan de Entrega.....	35
Tabla 11. Tiempo de la implementación por HU de la iteración 1.....	40
Tabla 12. Tareas por HU en la iteración 1.....	40
Tabla 13. Tarea 1 de la HU 1.....	41
Tabla 14. Tarea 2 de la HU 1.....	41
Tabla 15. Tarea 3 de la HU 1.....	41
Tabla 16. Tarea 1 de la HU 2.....	42
Tabla 17. Tarea 1 de la HU 3.....	42
Tabla 18. Tarea 2 de la HU 3.....	42
Tabla 19. Tarea 3 de la HU 3.....	43
Tabla 20. Tiempo de la implementación por HU de la iteración 2.....	43
Tabla 21. Tareas por HU en la iteración 2.....	43
Tabla 22. Tareas 1 de la HU 4.....	44
Tabla 23. Tarea 2 de la HU 4.....	44

Tabla 24. Tarea 1 de la HU 5	44
Tabla 25. Tarea 2 de la HU 5	45
Tabla 26. Tarea 3 de la HU 5	45
Tabla 27. Funciones implementadas en el <i>script</i>	46
Tabla 28. Eventos implementados en el <i>script</i>	46
Tabla 29. Caso de prueba de Aceptación "Adicionar objetos 3D".....	48
Tabla 30. Caso de prueba de Aceptación "Eliminar objetos 3D".....	48
Tabla 31. Caso de prueba de Aceptación "Actualizar el rango de animación".....	49
Tabla 32. Caso de prueba de Aceptación "Configurar nombre y ruta de los archivos a exportar".....	49
Tabla 33. Caso de prueba de Aceptación "Exportar archivo Collada .dae".....	50
Tabla 34. Caso de prueba de Aceptación "Exportar animación .pc2".....	51
Tabla 36. Tiempos alcanzados en la exportación de animación de complejidad media.....	52
Tabla 35. Tiempos alcanzados en la exportación de animación de complejidad alta.....	52
Tabla 38. Evaluación del tiempo dedicado a la tarea de exportar animación de complejidad media.....	52
Tabla 37. Evaluación del tiempo dedicado a la tarea de exportar animación de complejidad alta.....	53

Índice de Figuras.

Figura 1. Secuencia de <i>frame</i> que conforman una animación.....	4
Figura 2. Malla poligonal de "Sintel [3]" personaje 3D.....	5
Figura 3. Partes de un personaje controlado por huesos. (a) huesos deformadores. (b) controladores. (c) malla poligonal. (d) sistema completo con restricciones incluidas, de "Sintel [3]".....	6
Figura 4. Morph de una cara.....	7
Figura 5. Simulación de una bandera cubana.....	8
Figura 6. Modelo de dominio.....	28
Figura 7. Estructura del <i>script</i>	37
Figura 8. Diagrama de clases del diseño.....	38

Índice de Anexos.

Anexo 1. Administrar selección de objetos.....	59
Anexo 2. Configurar opciones de animación .pc2.....	59
Anexo 3. Configurar nombre del archivo y directorio.....	59

Anexo 4. Exportar Animación .pc2 y Exportar fichero .dae.....	60
Anexo 5. IDE utilizado para el desarrollo del <i>plugin</i>	60
Anexo 6. Uso de la solución propuesta, <i>plugin</i> InterAnim.....	61
Anexo 7. Interfaz de la solución propuesta.....	61
Anexo 8. Opinión del usuario ICAIC.....	62

Introducción.

En la industria del cine y el entretenimiento se hace uso de múltiples software de creación de contenidos 3D (Swcc3D), para la realización de animados, películas, anuncios publicitarios, cortometrajes y audiovisuales en general, con el objetivo de recrear e instruir al hombre. Los Swcc3D poseen disímiles herramientas que permiten a los artistas y técnicos recrear virtualmente elementos del mundo real, darles vida o añadirles efectos especiales, dependiendo de los resultados que se esperan y hacia donde estén enfocados.

El proceso de darle vida a los elementos que se desarrollan en los Swcc3D, se denomina animación. Para llevar a cabo este proceso en el mundo existe una multitud de software que siguen paradigmas colaborativos distintos (Software Libre y privativo), sin sacrificar la calidad de los resultados que son capaces de ofrecer. Entre los software privativos más usados para este propósito se encuentran el Autodesk 3ds Max (3ds Max) y el Autodesk Maya (Maya). Por otro lado, de los que se encuentran dentro de la filosofía del Software Libre, el más destacado es el Blender.

En la realización de productos de la industria del cine y el entretenimiento se planifica una serie de procesos en los cuales se incluye el proceso de animación. Sin embargo en ocasiones, por cuestiones legales o de organización, se necesita tomar decisiones para salvar recursos y trabajos realizados en uno de estos procesos previos a una etapa determinada y esto suele dar lugar a conflictos de herramientas. Un ejemplo real es la exportación de los datos que se utilizan para producir la película Meñique, la cual ha sido realizada con Maya, pero ha tenido que ser exportada al software privativo Lightwave. Esta decisión fue tomada por el Instituto Cubano de Arte e Industria Cinematográfica (ICAIC) con el fin de que se pueda concluir la película usando un software del cual poseen la licencia.

El proceso de exportar datos de un Swcc3D a otro se puede realizar a través de *plugin* de forma automatizada o de forma manual. Los *plugin* son funcionalidades o componentes que se le agregan al sistema del Swcc3D a través de lenguajes *script* usualmente definidos por el mismo software. La exportación de datos a través de *plugin* de forma manual es usada generalmente en casos donde el volumen de datos que se maneja es pequeño, sin embargo cuando el volumen de datos es grande no es recomendable realizar la exportación mediante esta vía. En estos casos es más factible el uso de *plugin* de forma automatizada, pues agiliza y simplifica la exportación.

Actualmente en el ICAIC, lugar donde se desarrollan proyectos audiovisuales utilizando 3ds Max, también existe un creciente interés en Blender debido a su condición de Software Libre y la posibilidad de lograr un producto de calidad aceptable mediante su uso. Sin embargo, en este instituto la exportación de

animaciones y mallas poligonales de 3ds Max a Blender no se realiza, ya que de forma manual genera atrasos en la producción debido al gran volumen de datos que implica la exportación de muchos objetos animados, además que se puede introducir errores en los archivos creados, tales como pérdida de datos y deformaciones incorrectas en las mallas. Tampoco cuentan con una forma rápida de exportar las animaciones y mallas poligonales, lo cual constituye la **situación problemática** en este trabajo.

Partiendo de la situación antes expuesta se plantea el siguiente **problema de investigación**: ¿Cómo exportar animaciones y mallas poligonales desde el software Autodesk 3ds Max a Blender?

A partir del problema anterior se define como **objeto de estudio**: Tecnologías para exportar contenidos 3D. Limitando el **campo de acción** a: tecnología de exportación de datos de animación 3D. Para dar respuesta al problema planteado se formula como **objetivo general**: implementar un *plugin* que permita exportar animaciones y geometrías de Autodesk 3ds Max a Blender

Se tiene como **idea a defender** que con la creación de un *plugin* en Autodesk 3ds Max se podrá agilizar el proceso de exportación de animaciones y mallas poligonales hacia el Swcc3D Blender, con lo cual instituciones como el ICAIC y la UCI se verán beneficiadas a la hora de exportar sus datos del software Autodesk 3ds Max al Blender.

Para darle cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

1. Elaboración del marco teórico de la investigación a partir del estado de las tecnologías de exportación de animación y geometría.
2. Identificación de los requisitos y diseño de la solución.
3. Identificación de las herramientas y lenguajes para el desarrollo de la solución.
4. Implementación de la solución propuesta.
5. Validación de la propuesta en el Instituto Cubano de Arte e Industria Cinematográficos.

En el progreso de la investigación científica se emplearon los siguientes métodos:

Métodos del nivel teórico:

- ❖ El método histórico lógico se empleó para realizar un estudio valorativo de la evolución de los diferentes lenguajes, metodologías y software posibles a utilizar en el desarrollo de la herramienta y de sus funcionalidades.
- ❖ El método inductivo-deductivo es utilizado en la investigación para sintetizar información de interés relacionada con el trabajo que se presenta.

- ❖ El método analítico–sintético se utilizó al analizar toda la información relacionada con el tema de tesis, para extraer los elementos más importantes de cada documento consultado.

Métodos del nivel empírico:

- ❖ Análisis de todas las fuentes de información relacionadas con el tema.
- ❖ La entrevista para obtener información sobre las herramientas de diseño más utilizadas, procesos de desarrollos, entre otros.

El contenido del trabajo está estructurado de la siguiente forma.

Capítulo 1. Fundamentación Teórica: Incluye un estado del tema tratado, los distintos formatos 3D son estudiados basado en las características que poseen. Se precisan un conjunto de conceptos y definiciones que le ofrecen al lector una panorámica del contexto de la investigación, incluye un estudio de las animaciones basadas en huesos, *morphing* y simulaciones, además de los software que exportan animaciones usados en la actualidad. Por último, se hace referencia a las tecnologías y metodologías seleccionadas, que serán de utilidad para su desarrollo práctico.

Capítulo 2. Solución Propuesta: En este capítulo se presenta la descripción de la solución propuesta para el desarrollo de la herramienta usando la metodología de desarrollo XP, partiendo de las funcionalidades y conceptos principales a usar. Se desarrollan las fases de exploración, planificación y diseño dando cumplimiento a las tareas de investigación número uno, dos y tres.

Capítulo 3. Implementación y Prueba. Evaluación: Plantea las tareas de ingeniería necesarias para implementar los requisitos funcionales y comprueba que el sistema cuente con todas las funcionalidades requeridas por el cliente. Se presenta una evaluación realizada a la solución propuesta. Se da cumplimiento a las tareas de investigación número cuatro y cinco.

Anexos: Muestra imágenes de la aplicación desarrollada.

Glosario de Términos: Contiene los términos importantes de la investigación, con el objetivo de facilitar la comprensión del lenguaje utilizado.

Capítulo 1 . Fundamentación Teórica.

Introducción.

En el presente capítulo, se exponen conceptos fundamentales sobre los formatos de almacenamiento 3D y herramientas usadas para el intercambio de mallas poligonales y animación, así como algunas técnicas usadas para animar objetos 3D y las principales herramientas de animación que se utilizan en el mundo. Se exponen características sobre las metodologías ágiles y pesadas, así como de los lenguajes más utilizados por las herramientas 3D.

1.1. Animación 3D.

Según el Diccionario de la Lengua Española, “...la animación es el proceso de dar vida a cosas inanimadas, comunicarles mayor vigor e intensidad.” [1]. La animación 3D, es el arte de dar vida a objetos inanimados haciendo uso de un software informático, que permita interpolar una secuencia de imágenes entre los cuadros claves de una animación, para crear la ilusión de movimiento [2]. Esta animación 3D o digital obedece a los 12 conceptos principales que rigen la animación clásica.

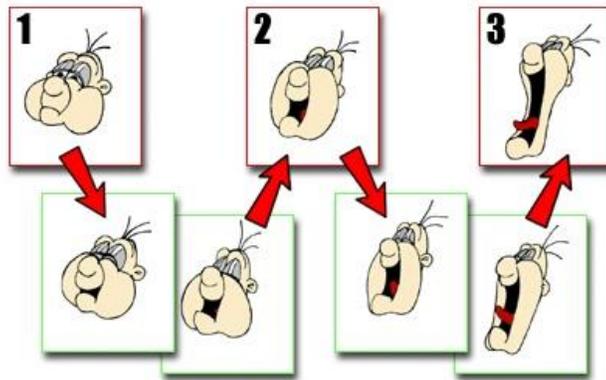


Figura 1. Secuencia de *frame*¹ que conforman una animación.

En la animación 3D se realizan transformaciones visuales a los distintos elementos 3D, ya sean personajes, lámparas, materiales, texturas, parámetros o efectos, con el fin de lograr movimiento en los elementos que se desee dar vida. Los objetos 3D pueden sufrir transformaciones básicas en sus valores de traslación, rotación y escala. Además de estas transformaciones básicas pueden ser animadas sus

¹ *Frame*: Cuadro o imagen, que conforman la animación.

formas, a través de deformadores, esqueletos o dinámica. Una buena animación es determinante en la calidad de una escena 3D, ya que trae a la vida todo el conjunto de elementos que la componen.

1.1.1. Animación de personajes 3D.

Los personajes creados en las herramientas 3D pueden ser considerados sistemas que según su complejidad poseen una serie de parámetros que pueden o no ser animados. Básicamente están constituidos por las mallas poligonales 3D o modelo, el sistema de huesos, controladores y deformadores, y su material, el cual incluye las texturas y demás detalles visuales que lo acompañen. La animación de personajes consiste en las transformaciones que sufren los controladores y deformadores que conforman el sistema de animación, que orientados por un animador pueden crear la ilusión de vida en los modelos 3D.

Para animar el personaje, es necesario llevar a cabo una serie de tareas destinadas a prepararlo para que pueda ser animado. Inicialmente, hay que darle forma dentro del software a la concepción que se tiene del personaje. Esto se logra mediante las técnicas de modelado, que dan como resultado una malla poligonal con la forma definida del personaje [3].

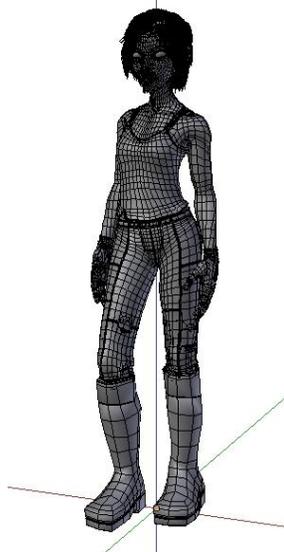


Figura 2. Malla poligonal de “Sintel [3]” personaje 3D.

1.1.2. Basado en huesos.

Los sistemas de animación basados en huesos son de los más útiles y usados por el animador pues permiten controlar y modificar los personajes 3D a gusto de este. Son un conjunto de modificadores conocidos como huesos, que conforman una estructura jerárquica. Sirven para generar expresiones

corporales o faciales [4]. La estructura lógica de un sistema basado en hueso se describe brevemente a continuación:

- ❖ **Controladores:** Son los encargados de dirigir a los huesos deformadores, el animador los manipula para lograr la animación deseada.
- ❖ **Huesos deformadores:** Estos huesos son los encargados de deformar la malla según un mapa de peso asignado a esta, el cual indica sobre que parte de la malla debe ejercer mayor o menor efecto la deformación.
- ❖ **Huesos ayudantes:** No son más que huesos que colaboran en la deformación, a través de Restricciones.
- ❖ **Restricciones:** Son limitaciones que se le imponen a los huesos con el fin de restringir algunas de las transformaciones que puede sufrir.

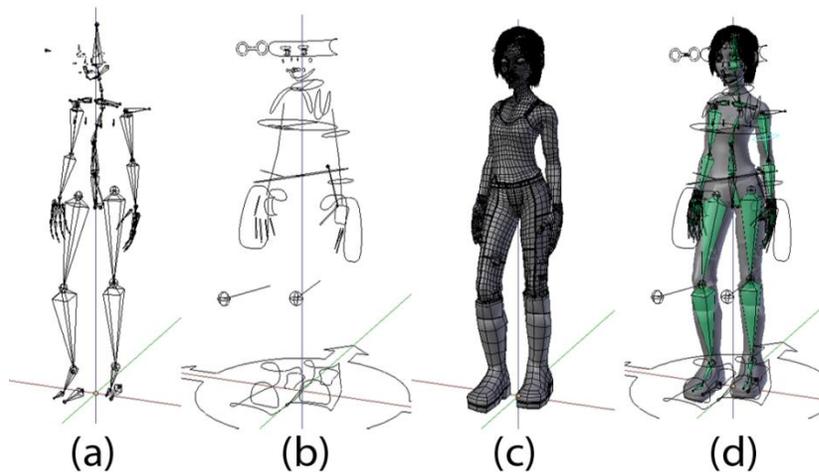


Figura 3. Partes de un personaje controlado por huesos. (a) huesos deformadores. (b) controladores. (c) malla poligonal. (d) sistema completo con restricciones incluidas, de “Sintel [3]”.

En general los sistemas de huesos dependen de la topología del personaje, pues muchas herramientas ya cuentan con sistemas para generarlos de forma automática. En el caso del 3ds Max cuenta con el *Biped*, el cual permite crear un esqueleto bípedo humanoide con propiedades especiales que le permiten ser animado instantáneamente. En el caso de Blender cuenta con el *plugin* Rigify², el cual permite generar completamente o por partes un sistema de huesos humanoide también.

² Rigify: Componente de Blender que permite crear un sistema de huesos.

1.1.3. Basado en *morphing*.

El *morphing* o *blendshape* es el sistema más usado para simular expresiones faciales. Se basa en crear una serie de copias de las mallas poligonales que se interpolan durante la animación, logrando de esta forma los cambios deseados. Comúnmente se utiliza para sincronización labial o expresiones faciales de personajes [5].

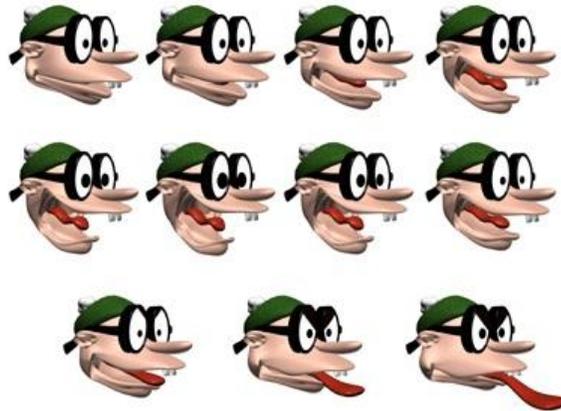


Figura 4. Morph de una cara.

Tiene como desventaja, que si se hacen muchas copias de las mallas poligonales en una escena, puede ralentizar el funcionamiento del software. Además, por la cantidad de modificadores que es necesario añadir para lograr las transiciones, a menudo ocurren errores en el modificador de deformación de la piel durante la animación, debido principalmente, a cambios en las referencias de los vértices de las mallas poligonales a los huesos [6].

1.1.4. Basado en simulaciones.

Los sistemas simulados son de gran importancia en los Swcc3D, ya que permiten realizar animaciones que para el animador son difíciles de lograr. Los principales Swcc3D incluyen un motor físico que permite realizar simulaciones, ya sea de fluidos, cuerpos blandos, rígidos o ropas.



Figura 5. Simulación de una bandera cubana.

Las simulaciones calculadas por los motores físicos que poseen los Swcc3D generalmente poseen parámetros que se pueden animar en muchos casos, con el fin de lograr una mayor eficiencia y que la necesidad de cálculo computacional sea menor [7].

1.2. Herramientas de Animación 3D.

Actualmente existen disímiles herramientas para la creación de animaciones 3D, las cuales brindan múltiples posibilidades que facilitan el trabajo de los animadores y logran una mayor credibilidad en los resultados que se obtienen de ellas. Entre las aplicaciones privativas destacan por su calidad y uso en el ICAIC: 3ds Max y Maya de Autodesk. De las herramientas libres el Blender es el mayor exponente. Estos software permiten exportar mallas poligonales 3D en disímiles formatos, sin embargo no todos poseen *plugin* o exportadores para formatos de animación de vértice.

1.2.1. Autodesk 3ds Max.

Es una suite completa e integrada de modelado 3D, animación, renderizado y composición para desarrolladores de juegos, artistas de efectos visuales y artistas gráficos en movimiento, junto con otros profesionales creativos que trabajan en la industria del diseño de los medios de comunicación.

Este Swcc3D dispone de una sólida capacidad de edición, una poderosa arquitectura de *plugin* y una larga tradición en plataformas Microsoft Windows. 3ds Max es utilizado en mayor medida por los desarrolladores de videojuegos, aunque también en el desarrollo de proyectos de animación como películas o anuncios de televisión, efectos especiales y en la arquitectura.

Desde la primera versión 1.0 hasta la 4.0 el programa pertenecía a Autodesk con el nombre de 3D Studio. Más tarde, Kinetix³ compró los derechos del programa y lanzó tres versiones desde la 1.0 hasta la 3.0 bajo el nombre de 3D Studio MAX. Más tarde, la empresa Discreet compró los derechos, retomando la familia empezada por Autodesk desde la 4.0 hasta 7.0 esta vez bajo el nombre de 3ds Max. Finalmente, Autodesk retomó el programa desarrollándolo desde la versión 8.0 hasta la actualidad bajo el nombre de Autodesk 3ds Max [8]. Su costo por licencia es 3495 dólares [9]. Para el uso empresarial su precio es superior. El 3ds Max posee una serie de *plugin* para exportar contenido 3D, destacando entre los principales los siguientes: Collada (.dae), 3D Studio (.3ds), Wavefront (.obj), AutoCaD (.dxf), Point Cache2 (.pc2), Autodesk (.fbx).

1.2.2. Blender.

Blender es una suite de código libre para la creación de contenido 3D, disponible para los principales sistemas operativos bajo la Licencia Pública General (GPL) GNU. Está orientado a artistas y profesionales del diseño y multimedia, puede ser usado para crear visualizaciones 3D estáticas o vídeos de alta calidad.

Blender permite modelar, texturizar, animar, renderizar, simular efectos de partículas, editar, sincronizar audio y video en forma no lineal y crear aplicaciones 3D como videojuegos o paseos virtuales. Originalmente desarrollado por la compañía “Not a Number” (NaN), el programa inicialmente fue distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta aunque posteriormente pasó a ser Software Libre. Blender es ahora desarrollado como 'Software Libre', con el código fuente disponible bajo la licencia GNU GPL.

Tiene una característica que a menudo es pasada por alto. Blender utiliza un intérprete de Python funcional. Esto le permite a cualquier usuario añadir funcionalidades a Blender escribiendo algún *script* de Python. Posee un paquete de creación integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D. Es multiplataforma, con una interfaz unificada para todas las plataformas basada en *Open Graphics Library* (OpenGL), listo para ser usado en todas las versiones de *Windows* (98, NT, 2000 y XP), Linux, OSX, FreeBSD, Irix y Sun, y otros sistemas operativos [10].

Entre las características de Blender destacan:

- ❖ Cuenta con una arquitectura 3D de alta calidad permitiendo un rápido y eficiente desarrollo, reestructurada en la serie 2.5x y 2.6x.

³ Antigua división de Autodesk, unida posteriormente a Inc. Discreet Logic.

- ❖ Ofrece herramientas de simulación avanzadas tales como dinámicas de cuerpos rígidos, de cuerpos suaves y de fluidos, herramientas de animación de personajes, un sistema de simulación física avanzado, y un sistema de composición de materiales basado en nodos.
- ❖ Posee un motor de 3D en tiempo real que permite la creación de contenido interactivo que puede ser reproducido independientemente.
- ❖ Tiene capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS (modelo matemático para representar curvas y superficies).
- ❖ Junto a las herramientas de animación se incluyen cinemática inversa, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- ❖ Posibilita el renderizado interno versátil y la integración externa con el potente trazador de rayos o “raytracer” libre de YafRay (motor de Render que trata de imitar la iluminación real, genera imágenes bastante realistas).
- ❖ Presenta modificadores apilables, para la aplicación de transformación no destructiva sobre mallas.
- ❖ Cuenta con un sistema de partículas estáticas para simular cabellos y pelajes, al que se han agregado nuevas propiedades entre las opciones de sombreado para lograr texturas realistas.
- ❖ Recientemente incluye entre sus motores de procesamiento gráfico al *Cycles* con el cual se alcanza un mayor realismo en las imágenes.

Además Blender posee una gran cantidad de *plugin* exportadores e importadores que le permite el intercambio con una variedad de software que interpretan contenido 3D. Entre ellos se puede nombrar el Ogre, SketchUp⁴, Lightwave, Maya y 3ds Max.

1.2.3. Autodesk Maya.

Autodesk Maya (también conocido como Maya) es un programa informático dedicado al desarrollo de gráficos en 3D, efectos especiales y animación. Surgió a partir de la evolución de *Power Animator*⁵ y de la fusión de Alias y Wavefront, dos empresas canadienses dedicadas a los gráficos generados por ordenador. Más tarde *Silicon Graphics*⁶ (ahora SGI), el gigante informático, absorbió a Alias-Wavefront, que finalmente ha sido absorbida por Autodesk.

Maya se caracteriza por su potencia y las posibilidades de expansión y personalización de su interfaz y herramientas. MEL (*Maya Embedded Language*) es el código que forma el núcleo de Maya, y gracias al cual se pueden crear scripts y personalizar el paquete [11].

⁴ El software de comunicación en 3D para los profesionales del diseño y la ingeniería. sketchup.google.com

⁵ Software predecesor de Maya.

⁶ Proveedor de soluciones de alto rendimiento a sistemas computarizados. www.sgi.com

Autodesk Maya ofrece a los artistas un flujo de trabajo creativo completo, con todas las herramientas para realizar animación, modelado, simulación, efectos visuales, renderización, rastreo de movimiento y composición en 3D dentro de una plataforma de producción sumamente ampliable. Toda esta funcionalidad se reúne en una única aplicación que aporta un valor excepcional a los artistas gráficos.

1.2.4. Lightwave.

Lightwave es un paquete de software que se utiliza para representar las imágenes en 3D, tanto animadas como estáticas. Incluye un motor de renderizado que soporta características avanzadas tales como la reflexión y la refracción realista, radiosidad y productos cáusticos. El componente de modelado 3D es compatible con el modelado de polígonos y superficies de subdivisión. El componente de la animación tiene características tales como la cinemática inversa y hacia adelante para los sistemas de animación de personajes y la dinámica de partículas. Los programadores pueden ampliar la capacidad de Lightwave, utilizando un SDK incluido, el cual ofrece LScript scripting (un lenguaje de programación propietario) y las interfaces comunes del lenguaje C.

Lightwave se compone de dos programas separados: *Modeler* y *Layout*. Cada programa está diseñado específicamente para proporcionar un espacio de trabajo dedicado para tareas específicas. Layout contiene el sistema de animación y de render que ofrece al usuario varias opciones, incluyendo opciones de seguimiento de rayos, multihilo, iluminación global, y parámetros de salida.

LScript es el lenguaje *script* de Lightwave. También proporciona un conjunto completo de funciones predefinidas que puede utilizar. Lightwave CORE hace uso de Python como su motor de scripting [12].

Entre las funcionalidades del Lightwave se pueden encontrar:

- ❖ Dinámicas: Posee sistemas de simulación para cuerpos blandos, cuerpos rígidos, ropas, colisiones y gravedad.
- ❖ Hypervoxels: Generan efectos de partículas animadas que luego pueden ser renderizados.
- ❖ Material: Lightwave viene con un editor de texturas nodal que viene con una colección de los sombreados de materiales para usos especiales.
- ❖ Física Bullet: Bullet es un rápido motor físico orientado al trabajo en tiempo real y motores de juego. Bullet cuenta con dinámicas 3D de cuerpos rígidos originalmente creadas por Coumans Erwin.
- ❖ Python: Se utiliza como un lenguaje de *script*, pero también se utiliza en una amplia gama de contextos de secuencias de comandos.

1.3. Formatos de almacenamiento 3D.

Los ficheros 3D son contenedores de datos, habitualmente de un escenario en tres dimensiones, ordenados mediante un formato de almacenamiento. Conteniendo generalmente datos como la posición de los objetos en la escena, materiales, texturas, orientación de las normales de las caras, posición y orientación de las luces, estados de render, así como todas las restantes informaciones relacionadas con un escenario 3D. Dependiendo del formato de almacenamiento que se utilice, influirá en la complejidad del fichero, facilidad de entendimiento y tiempo de carga.

1.4. Características de los formatos de almacenamiento 3D.

Los formatos de almacenamiento 3D son un conjunto de reglas (algoritmos) que definen la manera correcta de almacenar datos en memoria. Actualmente se utilizan diversos formatos para el almacenamiento de la información en ficheros, siendo los comunes el binario y el ASCII.

1.4.1. 3D Studio (3DS).

Es un formato de archivo binario creado por Autodesk para el 3ds Max. Originalmente se creó como formato de archivo nativo para 3D Studio DOS (versiones 1-4). Debido a su fiabilidad y solidez, se ha convertido en un estándar industrial para el intercambio de modelos 3D entre programas de diseño 3D. Fue reemplazado en versiones posteriores por el formato .MAX, también de Autodesk [13].

Ventajas: Permite el intercambio de mallas poligonales 3D, colores de vértices y las coordenadas UV de las texturas.

Desventajas: No permite el intercambio de sistemas de huesos, ni de materiales con un nivel medio de complejidad. Tampoco permite almacenar algún tipo de animación.

1.4.2. Wavefront (OBJ).

El formato de archivo .obj es un formato de archivo de objetos 3D estándar creado para el uso con Wavefront's Advanced Visualizer. Es recomendable para objetos estáticos. Contiene mallas poligonales que presentan puntos, aristas y caras, para definir un objeto o mallas poligonales de objetos de forma libre que contienen curvas y superficies, aunque en lo que más se usa es en objetos poligonales. No necesita ningún tipo de encabezado, aunque es usual poner un comentario al inicio del fichero [14].

Ventajas: Permite el intercambio de mallas poligonales 3D, colores de vértices y las coordenadas UV de las texturas. Buena integración con el Blender y 3ds Max.

Desventajas: Permite el intercambio de sistemas de huesos, pero en forma de malla poligonal 3D, permite el intercambio de materiales con un nivel medio de complejidad, pero generalmente contiene errores. No permite almacenar algún tipo de animación. Puede contener animación, pero generalmente con errores.

1.4.3. AutoCad (DXF).

Formato de datos desarrollado por Autodesk y usado por imágenes vectoriales de herramientas de diseño asistido por computadoras (CAD, por sus siglas en inglés). Fue creado como un formato universal de forma que los documentos de AutoCAD pudiesen ser abiertos más fácilmente por otros programas [15].

Ventajas: Permite el intercambio de mallas poligonales 3D.

Desventajas: No permite el intercambio de sistemas de huesos, no permite el intercambio de materiales con un nivel medio de complejidad o coordenadas UV de texturas. Tampoco permite almacenar algún tipo de animación.

1.4.4. Lightwave object (LWO).

Es el formato nativo de los objetos 3D en Lightwave. LWO es el acrónimo de objeto de Lightwave. Fue desarrollado por la empresa NewTek, la cual es la propietaria del Lightwave.

Posee un formato binario, el cual permite almacenar colecciones de polígonos y sus propiedades. Aunque es un formato privativo es usado por muchos software y aplicaciones 3D [16].

Ventajas: Permite el intercambio de mallas poligonales 3D, colores de vértices y las coordenadas UV de las texturas. Buena integración con el Blender.

Desventajas: No permite el intercambio de sistemas de huesos, permite el intercambio de materiales con un nivel medio de complejidad, pero generalmente contiene errores. Tampoco permite almacenar algún tipo de animación.

1.4.5. Collada (DAE).

Collada define un esquema basado en XML para que sea fácil de transportar activos 3D entre aplicaciones que permite creación de diversas herramientas para el contenido 3D y de perfeccionamiento que se combinan en una línea de producción.

Sin embargo, la Collada no es sólo una tecnología, como la tecnología por sí sola no puede resolver este problema de comunicación. Collada ha logrado proporcionar una zona neutral donde los competidores trabajan juntos en el diseño de una especificación común. Esto crea un nuevo paradigma en el que se

apoya el formato directamente por la creación de contenido digital los proveedores. Cada uno de ellos escribe y es compatible con su propia implementación del importador Collada y herramientas exportadoras. El esquema de Collada soporta todas las características de los modernos Swcc3D, incluidos los efectos de sombreado programables y de simulación de la física. También se puede extender fácilmente por los usuarios finales para sus propios fines específicos. En otras palabras, Collada no está diseñado para ser un mecanismo de transporte de datos temporal, sino más bien a ser el esquema para los datos de origen para sus activos digitales. No está diseñado como un mecanismo de entrega, pero para ser titular por cualquier plataforma de destino. El documento de esquemas XML Collada es de acceso público en Internet para la validación de contenido en línea.

El lenguaje intermedio proporciona codificación completa de las escenas visuales, incluyendo: las mallas poligonales, los sombreados y efectos, la física y la animación. Collada permite a los principales Swcc3D trabajar con eficacia para crear aplicaciones de sombreado, efectos y empaquetado usando OpenGL Lenguaje de sombreado [17].

Ventajas: Permite el intercambio de mallas poligonales 3D, colores de vértices y las coordenadas UV de las texturas. Buena integración con el Blender y 3ds Max.

Desventajas: Permite el intercambio de sistemas de huesos, generalmente con errores, permite el intercambio de materiales con un nivel medio de complejidad, pero generalmente contiene errores. Permite almacenar animación 3D, pero a nivel básico, solo rotaciones, traslaciones y escalas de objetos.

1.5. Formatos de animación de vértice.

Los formatos de animación de vértices permiten almacenar datos de deformación, almacenando la posición de cada vértice de las mallas poligonales 3D para cada *frame*. Estos formatos permiten simplificar las escenas en algunos Swcc3D, ya que permiten almacenar las animaciones de deformaciones de objetos 3D, evitando tener en la escena complejos sistemas de huesos o simulaciones que generalmente sobrecargan al software. Son usados también para almacenar animaciones de objetos simulados, por ejemplo las ropas o cuerpos blandos. Los funcionamientos de estos sistemas generalmente implican cálculos algorítmicos bastante complejos que sobrecarga el software, sin embargo si esos datos son almacenados en algunos de los formatos de animación de vértice, pues la carga sobre el software sería mucho menor.

1.5.1. Formato Motion Designer Data (MDD).

Es un formato privativo desarrollado por NewTek [18]. Fue el primer formato de almacenamiento de animación de vértices, se utiliza en muchos paquetes de 3D y proporciona una manera fácil de mover a los personajes animados, simulaciones dinámicas, y las deformaciones geométricas entre los paquetes. El formato .mdd es utilizado para almacenar datos de deformación de los vértices. Cuando un archivo de .mdd se exporta este guarda la posición de cada punto de las mallas elegidas para cada fotograma de la animación específica. Una vez almacenadas, las mallas pueden ser despojadas de todos los deformadores y animaciones. Luego pueden ser leídos los datos para el fotograma especificado y deformar las mallas a la posición correcta. Para una resolución de malla muy alta esto pueden ser muchos datos para leer o escribir. Los datos en el formato .mdd se almacenan como binario para que la velocidad y la eficiencia sean mayores, no se puede leer en un visor de texto [18].

1.5.2. Formato Point Cache 2 (PC2).

Point Cache 2 es un formato de animación de vértices, creado por John Burnett en el año 2000 para la empresa Autodesk, la cual lo incluyó en el 3ds Max y posteriormente en el Maya. Permite almacenar en un archivo los datos de la animación, grabando para cada vértice de las mallas poligonales seleccionada su posición en el espacio. Ayudando en la aceleración de la reproducción de complejas deformaciones realizadas por sistemas de huesos o simulaciones.

La animación almacenada de un fichero Point Cache 2 no se puede editar, pero precisamente puede soportar cualquier tipo de animación, en cualquier tipo de objeto, siempre y cuando la topología o vértice número u orden no cambie entre fotogramas. El formato de caché sólo contiene puntos de posiciones de los vértices, de manera que para llevar la animación a otro Swcc3D como 3ds Max, también se tendrá que exportar la propia malla en otro formato, como .obj o .dae [19].

El formato de archivo Point Cache 2 está compuesto de una cabecera con la estructura que se expone a continuación [20]:

Tipo de dato	Descripción
char	Firma Cache
int	Versión del archivo
int	Número de puntos
float	<i>Frame</i> de Inicio

float	Índice de la muestra
int	Número de muestra

Tabla 1. Estructura de la cabecera del formato Point Cache 2

Firma Cache: Primer dato de la cabecera, contiene los caracteres POINTCACHE2. Este dato es la rúbrica del archivo .pc2, identificándolo como el formato que es.

Versión del archivo: Constante entera que siempre tiene valor uno.

Número de puntos: Cantidad de vértices por muestra.

Frame de Inicio: Indica el *frame* inicial a partir del cual se archivará la información contenida en el bloque de Dato.

Índice de muestra: Índice que determina cada cuantas muestras se realizará el proceso de grabar los vértices. Así que podrá realizar la muestra cada cierto número de *frames* salvando espacio en el dispositivo de almacenamiento si no es necesario hacer cada muestra a todos los *frames*. Este factor es útil para realizar el proceso de Desenfoque de movimiento⁷.

Número de muestra: Cantidad de *frame* total a la que se realizara el proceso.

Tras la cabecera están almacenadas todas las muestras de caché, que son instantáneas de todas las posiciones de puntos para un objeto. Cada muestra se almacena una tras otra como una matriz de plana de dato numérico de punto flotante (float), de la forma [x, y, z] para cada punto y cada muestra es calculada por la siguiente fórmula:

$$(\text{Número_de_puntos} * \text{sizeof(float)} * 3) \text{ bytes}$$

La mayor desventaja que posee este formato es que no permite la pérdida de información. Es decir que las mallas poligonales no pueden perder o ganar vértices durante el desarrollo de la animación que se desarrolle.

1.6. Herramientas de para exportar Animaciones 3D en el software 3ds Max.

Actualmente existen herramientas que facilitan la exportación de animaciones de un Swcc3D a otro. Estas aplicaciones se basan en el trabajo con los distintos formatos de intercambio de mallas poligonales 3D y de animación de vértice. Su objetivo principal es que a partir de una selección de distintos elementos

⁷ Efecto de render que se utiliza cuando hay movimientos rápidos o bruscos dentro de la escena 3D. Se utiliza para lograr un efecto realista.

geométricos o malla de la escena, previamente animados con sistemas de huesos o simulaciones, se pueda crear un archivo o caché de animación. La mayoría de estos son aplicaciones o *plugin* privativos que desarrollan empresas que se dedican al trabajo con contenido 3D. A continuación se analizan las principales.

1.6.1. XMesh.

XMesh es un conjunto de herramientas para guardar mallas poligonales de una escena animada en una secuencia de archivo externo y para la carga de estos datos más tarde para el render. Thinkbox Software Inc. es una compañía canadiense de propiedad de Thinkbox Inc., una empresa de EE.UU, propietaria del software. La herramienta está dividida en dos sub-herramientas principales: el XMesh Loader la cual se distribuye de forma libre y el XMesh Saber la cual requiere licencia [21].

Las principales características favorables de esta herramienta son:

- ❖ Soporte para topología modificable. Lo cual permite la perdida de vértices durante la animación.
- ❖ Soporte para los 100 canales de animación de 3ds Max.
- ❖ Soporte para exportar múltiples objetos como una malla en el Espacio del Mundo 3D.

Las desventajas son:

- ❖ Herramienta privativa perteneciente a Estados Unidos y Canadá.
- ❖ Una licencia por equipo.
- ❖ Exporta en una formato privativo, limitando a que la lectura del fichero solo se pueda realizar en 3ds Max.

1.6.2. SuperMesher.

SuperMesher es un *plugin* para 3ds Max que permite almacenar en caché las mallas poligonales animada, independientemente de su topología y el número de objetos, incluyendo los sistemas de partículas. La caché se almacena en un archivo separado que se puede cargar en cualquier versión de 3ds Max 9 o superior. Debido a que el resultado es una sola malla, SuperMesher genera automáticamente identificadores apropiados para los materiales [22]. Seguidamente se muestran algunas ventajas y desventajas de este *plugin*.

Ventaja:

- ❖ Permite acelerar los sistemas animados, ya que una vez guardada la animación no hace falta tener sistemas de huesos, lo cual hace que la escena esté menos cargada.

Desventajas:

- ❖ Herramienta privativa perteneciente a Estados Unidos.
- ❖ Exporta en un formato privativo, limitando a que la lectura del fichero solo se pueda realizar en 3ds Max.

1.6.3. Point Oven.

Point Oven es una suite de *plugin* comerciales diseñados para almacenar los datos de animaciones vértices, para agilizar los flujos de trabajo, y para transferir datos entre diferentes aplicaciones 3D. Los *plugin* se pueden utilizar para simplificar las escenas en 3ds Max, así como deformaciones complejas, o para compartir datos con otros usuarios que tienen el Point Oven, pero pueden utilizar una aplicación de 3D diferente. Point Oven es compatible actualmente con XSI⁸, Maya, 3ds Max y Lightwave [23].

Las ventajas de este *plugin* son:

- ❖ Buena Integración con el 3ds Max.
- ❖ Exporta en varios formatos, incluyendo el formato de animación de vértice .mdd.
- ❖ Está disponible para varios Swcc3D.

Contando con las siguientes desventajas:

- ❖ Herramienta privativa a nombre de Mark Wilson.

1.7. Metodologías de desarrollo de software.

Las metodologías y estándares utilizados en el desarrollo de software proporcionan una guía de buenas prácticas, con lo cual se asegura la calidad del producto final, así como el cumplimiento en la entrega dentro de un tiempo estipulado.

Es una guía que indica qué hacer y cómo actuar en cada momento del ciclo de desarrollo del software, y su finalidad es mantener la eficacia y eficiencia durante todo el proceso de desarrollo para obtener un sistema con calidad. Hoy en día existen varias metodologías, las que se clasifican en dos tipos principales: las metodologías robustas y las metodologías ágiles.

1.7.1. Programación Extrema (XP).

La Programación Extrema es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Esta metodología

⁸ XSI: Software de creación de contenido 3D, propiedad de Autodesk, su nombre completo es Autodesk Softimage XSI.

surge a partir de los problemas existentes en el desarrollo de software que presentan cambios en sus requerimientos, se plantea como una metodología a emplear en proyectos de riesgo elevando la productividad de los mismos.

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes y jefes de proyecto). El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres. Este mecanismo indica que, por ejemplo, si el cliente establece el alcance y la calidad, y el jefe de proyecto el precio, el grupo de desarrollo tendrá libertad para determinar el tiempo que durará el proyecto [24].

XP se plantea dos objetivos fundamentales:

- ❖ La satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.
- ❖ Potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

En el documento Agile Manifiesto se definen los planes en XP diferenciándolo de las metodologías tradicionales en tres aspectos fundamentales.

- ❖ Simplicidad del plan. No se espera que un plan requiera complicados sistemas de gerenciamiento de proyectos.
- ❖ Los planes son realizados por las mismas personas que desarrollarán el trabajo.
- ❖ Los planes no son predicciones del futuro, sino la mejor estimación de cómo saldrán las cosas. Los planes son útiles, pero necesitan ser cambiados cuando las circunstancias lo requieren. De otra manera, se termina en situaciones en las que el plan y la realidad no coinciden, y en estos casos, el plan es totalmente inútil.

Los conceptos básicos de esta planificación son los siguientes:

Historias de usuarios

Sustituyen los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de

especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

Plan de entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes). XP denomina a esta reunión *Planning workshop*, o como se estime convenientemente dependiendo del tipo de empresa y trabajo que realiza. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores.

Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario. XP se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito:

❖ Comunicación

Muchos de los problemas que existen en proyectos de software (así como en muchos otros ámbitos) se deben a problemas de comunicación entre las personas. La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.

❖ Simplicidad

XP, como metodología ágil, apuesta a la sencillez, en su máxima expresión. Sencillez en el diseño, en el código, en los procesos. La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas.

❖ Retroalimentación

La retroalimentación debe funcionar en forma permanente. El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración; y para comprender, cada vez más, sus necesidades.

Los resultados de las pruebas unitarias son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de su trabajo.

❖ Coraje

Cuando se encuentran problemas serios en el diseño o en cualquier otro aspecto, se debe tener el coraje suficiente como para encarar su solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuanto tiempo se ha invertido previamente en el mismo.

1.7.2. Rational Unified Process (RUP).

Se encuentra dentro de las Metodologías tradicionales, las cuales teniendo en cuenta la filosofía de desarrollo, son aquellas con mayor énfasis en la planificación y control del proyecto, así como en la especificación precisa de requisitos y modelado.

RUP sin importar la magnitud del proyecto que se llevara a cabo está estructurado y permite adecuarse a cualquier entorno. También permite el modelado de sistemas con tecnología orientada a objetos, destacándose en el aseguramiento de la planificación enfocada a la producción de software de calidad ajustándose a las exigencias de cada usuario o cliente final basándose en costos y calendarios predecibles [25].

El proceso de software propuesto por RUP tiene tres características esenciales:

- ❖ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios necesitan y se representa a través de los requerimientos. Los Casos de Uso son una herramienta para especificar los requisitos del sistema y a su vez guían su diseño, implementación y prueba.
- ❖ Centrado en la Arquitectura: La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de líneas base y se va modificando en dependencia de las necesidades del producto final.
- ❖ Iterativo e Incremental: El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto final. Para esto, la

estrategia que se propone en RUP es tener un proceso iterativo e incremental donde el trabajo se divide en partes más pequeñas o mini proyectos dependiendo de las dimensiones del proyecto y así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales).

1.8. Lenguajes de programación y herramientas para la implementación.

Las herramientas de desarrollo de software desempeñan un papel primordial en la creación de aplicaciones. A lo largo de la vida de un proyecto de desarrollo de software, el programador puede utilizar diversas tecnologías y herramientas, las cuales muchas veces son definidas por supervisores o seleccionada por expertos.

Para la elección de las herramientas a utilizar se realizó un estudio para seleccionar las más adecuadas con las características de desarrollo.

1.8.1. SDK de Autodesk 3ds Max.

El SDK (*Software Development Kit* o Paquete de Desarrollo de Software), provee un entorno para que los programadores puedan añadir funcionalidades y herramientas al 3D Studio Max. Es una buena opción cuando es necesario efectuar una gran cantidad de cálculo con rapidez.

SDK permite desarrollar complejas herramientas para casi cualquier aspecto del 3D Studio Max, haciendo uso de la programación orientada a objetos, el lenguaje de alto nivel C++ y el IDE⁹ (*Integrated Development Environment*) de desarrollo de Microsoft, Visual Studio.

El SDK es el preferido cuando el rendimiento es lo más importante; en general cuando el cálculo es el principal propósito de una herramienta. Pero los conocimientos para desarrollar e implementar una herramienta o funcionalidad para el 3ds Max utilizando el SDK ya no serían solo los del diseñador, haría falta tener conocimientos amplios del IDE de desarrollo y de las funciones con las que se va a trabajar.

1.8.2. MAXScript.

Es el lenguaje de programación que viene integrado al 3ds Max para usuarios con un nivel de experiencia adecuado y se emplea con el objetivo de automatizar procesos cotidianos. Es orientado a objetos y tiene varias características especiales como los constructores que reflejan conceptos de alto nivel en la interfaz de 3ds Max, permitiéndole al usuario la creación ágil y óptima de MacroScript que reflejen acciones conjuntas para lograr un mismo objetivo.

⁹ Entorno de desarrollo que le provee las facilidades integrales a los programadores para el desarrollo del software.

La sintaxis de MAXScript es simple de usar para los usuarios del 3D Studio Max, basta con dominar el trabajo manual para que la programación sea tan fácil como describir con lenguaje común lo que se quiere hacer.

Un *script* es un archivo de texto y no necesita compilador, las líneas de código o declaraciones son interpretadas una por una y ejecutadas a su vez. Lo cual significa que MAXScript no domina nada sobre el resto del código, cada vez que el *script* se ejecuta, MAXScript debe interpretar el código completo otra vez [26].

Proporciona la capacidad de:

- ❖ Crear *script* para todos los aspectos del uso del programa, como el modelado, la animación, los materiales, la renderización, etc.
- ❖ Controlar el programa interactivamente mediante una ventana de línea de comandos.
- ❖ Construir herramientas de importación/exportación personalizadas con la E/S (entrada/salida) de archivos interna.
- ❖ Crear controladores de procedimiento que puedan acceder a todo el estado de la escena. Construir herramientas de proceso por lotes, como *script* de renderización por lotes.
- ❖ Configurar interfaces vivas con sistemas externos mediante Automatización OLE.

La sencillez de la sintaxis permite que lo utilicen usuarios no programadores, ya que sus normas de puntuación y formato mínimas son muy simples.

MAXScript brinda algunos métodos de alto nivel que pueden ser encontrados en el SDK(C++) de 3ds Max y presenta nuevas potencialidades que no están presentes en el SDK. Sí la funcionalidad es soportada por el MAXScript, pero no por el SDK, entonces el MAXScript será la única opción. En particular el 3ds Max brinda potencialidades vía el OLE/ActiveX que hace más fácil codificar en MAXScript que con el SDK.

MAXScript puede ser conveniente para crear prototipos de *plugin*, para desarrollar *plugin* de simple complejidad y para escribir pruebas de pequeños fragmentos de funcionalidades.

1.8.3. Python.

El lenguaje de Python fue creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de *script*, con tipado dinámico, multiplataforma y orientado a objetos.

Python es considerado como la “oposición leal” a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a este mucho más limpio y elegante para programar. Este permite dividir el programa en módulos reutilizables desde otros programas. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaz gráfica de usuario a (GUI), entre otros.

Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa [27].

Características de Python.

Es un lenguaje de programación fácil de aprender y potente. Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz. La elegante sintaxis de Python y su naturaleza interpretada hacen de él el lenguaje ideal para guiones (scripts) y desarrollo rápido de aplicaciones; en muchas áreas y en la mayoría de las plataformas, o sea, está expresamente diseñado para ser usado como una extensión para las aplicaciones que necesiten una interfaz programable, y esto es por lo que Blender lo utiliza. De las dos maneras que tiene de extender Blender, siendo la otra los *plugin* (complemento) binarios; los *script* de Python, es la más poderosa, versátil, más sencilla de comprender y robusta. Proporciona un equilibrio muy bueno entre lo práctico y lo conceptual puesto que es un lenguaje interpretado, o sea, suele ocurrir que se vuelve algo lento en la ejecución de los programas, ya que en lugar de compilar los programas escritos en este lenguaje para obtener ficheros binarios y bibliotecas, se ejecutan sobre la marcha.

1.9. Herramienta CASE. Visual Paradigm para UML.

Visual Paradigm para UML (VP-UML) es una herramienta de diseño UML y herramienta CASE diseñada para la ayuda al desarrollo de software. VP-UML soporta estándares de la industria claves, tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI, ofrece un completo conjunto de herramientas de los equipos de desarrollo de software necesario para la captura de requisitos, la planificación de programas, la planificación de controles, la clase de modelado, modelado de datos entre otros [28].

Consideraciones del capítulo.

Con este capítulo se logró estudiar un conjunto de conceptos y caracterizaciones que permiten una mejor comprensión del estado del arte y de los contenidos que se mostrarán en los próximos capítulos del presente trabajo. Con el fin de obtener los criterios necesarios para seleccionar una metodología de desarrollo se estudiaron y caracterizaron algunas de las más usadas. Se profundizó en las principales ventajas y desventajas de los formatos de almacenamiento 3D y los principales formatos de animación de vértice, así como en los lenguajes de programación MAXScript y Python. Se realizó un estudio de las herramientas que existen a nivel internacional y nacional, lo que permite asegurar que en las fuentes consultadas no existe evidencia de que se haya desarrollado en Cuba alguna herramienta con este fin.

Capítulo 2 . Solución propuesta.

Introducción.

En el presente capítulo se presentan las características principales de la aplicación y se dará una propuesta de la solución. Se explica la dinámica del proyecto a través de las historias de usuario, plan de iteraciones y otros modelos generados durante el desarrollo.

2.1. Selección de los formatos 3D a utilizar.

Para seleccionar los formatos para el intercambio de mallas poligonales y animación entre los Swcc3D 3ds Max y Blender fue necesario indagar sobre cuales formatos realizaban estas funciones de manera correcta. Sin embargo el equipo de desarrollo no identificó documentación alguna que expusiera de forma consistente esta información. El equipo de desarrollo decidió realizar pruebas con los formatos estudiados para poder encontrar cuales permitía de forma correcta exportar las mallas poligonales y animación.

A continuación se exponen una serie de características que permitieron comparar los formatos 3D estudiados de acuerdo a las necesidades planteadas por el cliente. Las características a tener en cuenta en los formatos son:

- ❖ Intercambio de mallas poligonales 3ds Max a Blender (IGeo)
- ❖ Intercambio de animación 3ds Max a Blender (IAnim)
- ❖ Intercambio de sistema de huesos 3ds Max a Blender (ISistH)
- ❖ Intercambio de coordenadas UV 3ds Max a Blender (IUUV)

Teniendo en cuentas estas características se podrán valorar cuál o cuáles formatos son los indicados para lograr un eficiente intercambio de mallas poligonales y animación del software 3ds Max a Blender. El formato LWO se omitió del análisis pues al no ser un formato incluido en el 3ds Max no se puede tener acceso para trabajar con él. Las evaluaciones serán generadas por el equipo de desarrollo en base a la capacidad del formato de contener los datos de animaciones, mallas poligonales, coordenadas UV y sistemas de huesos. Las evaluaciones son:

- ❖ **Aceptable (1):** Indica que el proceso se realiza de forma correcta, los datos fueron exportados íntegramente.
- ❖ **Parcial (0):** Indica que el proceso se realiza de forma incompleta, no todos los datos se encuentran almacenados de forma correcta.

- ❖ **Incorrecto (-1):** Indica que el proceso no se realiza de forma correcta, los datos almacenados poseen error en el formato o pérdida de información.
- ❖ **Negativo (--):** Indica que este proceso no se realiza en el formato, no está definido dentro de él almacenar este tipo de información.

Formatos:	IGeo	IAnim	ISistH	IUV
.dae	1	0	0	1
.3ds	0	--	--	1
.obj	0	0	0	1
.dxf	-1	--	--	1
.pc2	--	1	--	--
.mdd	--	1	--	--

Tabla 2. Selección de los formatos de intercambio a usar

Luego de una valoración establecida por el equipo de desarrollo y basándose en los resultados que se resumen en la anterior tabla los formatos seleccionados fueron el .pc2 para la animación de vértice y el .dae para el intercambio de mallas poligonales. La estructura del formato .pc2 es conocida y documentada, mientras que el formato .dae está suficientemente integrado al Blender, permitiendo la importación de objetos 3D de forma correcta, pues mantiene el orden lógico de las partes de las mallas poligonales importadas.

2.2. Propuesta de la solución.

Para darle cumplimiento al desarrollo de la propuesta se decidió utilizar la metodología de desarrollo de software XP, pues el cliente solicitó participar activamente en el desarrollo del *plugin*, lo que permitió al equipo de desarrollo contar con su presencia durante el todo el ciclo de desarrollo. XP brindó además una forma rápida de guiar el desarrollo de este *plugin*, permitió centrar las fuerzas en la implementación y solo documentar los artefactos más importantes. Como lenguaje de implementación se decidió utilizar el MAXScript, lenguaje de implementación del propio software 3ds Max con un IDE propio de la herramienta. Como lenguaje de marcado se utilizará XML.

La solución de la herramienta se basa en la creación de un *plugin* en 3ds Max, el cual creará los ficheros Point Cache 2 (.pc2) y Collada (.dae) necesarios para obtener la animación y el modelo de una escena animada previamente. Los datos creados serán legibles en el software Blender. El funcionamiento del *plugin* se basa en una serie de pasos a seguir:

- ❖ Selección de los objetos 3D que conforman al personaje u objeto animado previamente en el visor 3D.
- ❖ Luego serán añadidos a una lista.
- ❖ A elección del usuario se configurarán los parámetros de animación .pc2, el nombre y directorio de los ficheros a generar.
- ❖ Se exportará su animación en formato .pc2
- ❖ Por último exportar el objeto con las mallas poligonales que conforman todas las partes del personaje unidas, en formato .dae.

El *plugin* se encontrará en la categoría¹⁰ *File* de la barra de herramientas del 3ds Max con el nombre InterAnim y se almacenará en la locación del disco duro donde 3ds Max copia los *plugin* instalados por el usuario.

2.3. Modelo de dominio.

Un modelo de dominio o modelo conceptual es “una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés”. En otras palabras, explica los conceptos significativos de un problema, facilitando la interpretación del sistema y el entorno donde está enmarcado.

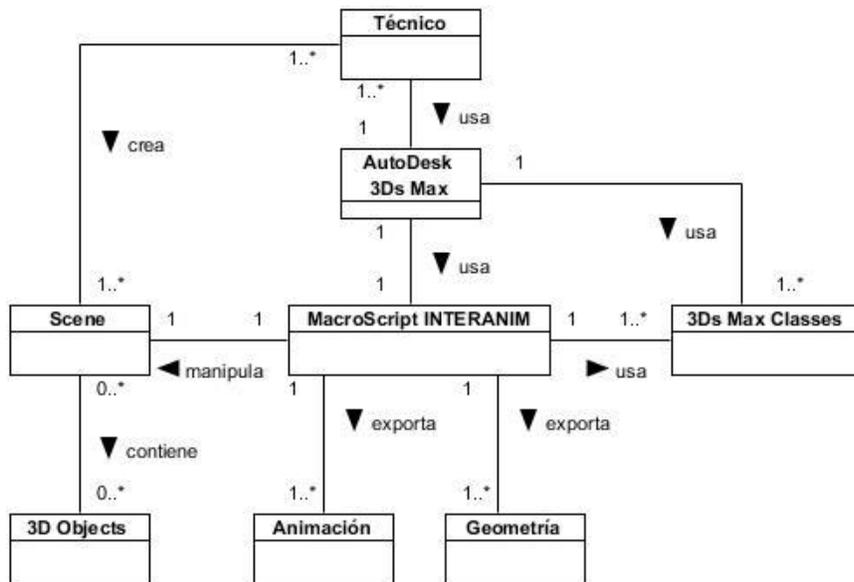


Figura 6. Modelo de dominio

¹⁰Categoría: En 3ds Max se definen una serie de categorías para ubicar las herramientas según su función.

A continuación se describen los conceptos representados en el modelo anterior:

- ❖ **Técnico:** encargado de desarrollar la labor de exportar, posee conocimientos en el trabajo con el software 3ds Max.
- ❖ **Autodesk 3ds Max:** Software privativo 3D donde se desempeñan las tareas de exportar la animación y los modelos.
- ❖ **Scene:** Conjunto de todos los objetos que contiene el 3ds Max, esto incluye los objetos geométricos malla, sistemas de partículas, cámaras, ayudadores, etc...
- ❖ **3D Objects:** Objetos 3D presentes en la escena
- ❖ **MacroScript INTERANIM:** Herramienta que posibilita exportar las animaciones de los objetos 3D y sus respectivos modelos.
- ❖ **3ds Max Classes:** Conjunto de todas las clases usadas para implementar la interfaz, funciones y eventos de la herramienta en aras de lograr su objetivo.
- ❖ **Archivo .dae:** Archivo con el formato Collada que contiene los datos de las mallas poligonales y UV del objeto exportado.
- ❖ **Archivo .pc2:** Archivo con el formato Point Cache 2 que contiene los datos de la animación del objeto u objetos seleccionado(s).

2.4. Requisitos funcionales.

“Los requerimientos funcionales describen lo que el sistema debe hacer [29]”, que en otras palabras son las acciones que se pueden realizar. Dentro de las funcionalidades que la herramienta debe brindar están:

- RF1.** Administrar la selección de objetos 3D
- RF2.** Configurar los parámetros de animación .pc2
- RF3.** Configurar la salida de archivos de la aplicación
- RF4.** Exportar animación en formato .pc2
- RF5.** Exportar mallas poligonales en formato .dae

2.5. Requisitos no funcionales.

Los requisitos no funcionales se refieren a las propiedades o cualidades del software (24), las que hacen del mismo un producto atractivo, usable, rápido y confiable. Seguidamente, se explican los requerimientos no funcionales de la aplicación a desarrollar:

- ❖ Soporte: Contará con un video tutorial.
- ❖ Hardware: La PC deberá contar con 1 Gigabyte de memoria RAM como mínimo.
- ❖ Software: La PC deberá tener sistema operativo: Windows XP, Vista, Seven.

2.6. Fase de Exploración.

La metodología de desarrollo XP comienza con su fase de exploración. Durante esta fase se realiza el proceso de identificación de las Historias de Usuario (HU o UH, del inglés *User Histories*). Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología [24].

2.6.1 Historias de Usuario.

La Historia de Usuario es una técnica utilizada para especificar los requisitos del software, se utiliza específicamente en XP. Las mismas son tarjetas de papel donde el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Estas Historias de Usuario tienen un tratamiento muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [31].

Los programadores estiman el esfuerzo asociado y las dependencias entre ellas. A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Historia de Usuario	
Número:1	Nombre: Exportar animación .pc2.
Usuario: Todos los usuarios que interactúen con la herramienta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
El usuario exporta el archivo .pc2. Archivo que contendrá la animación de los objetos 3D	

seleccionados en la escena. Se generarán los archivos .pc2 correspondientes a cada objeto 3D seleccionado y luego se ensamblarán todos creando uno solo.

Observaciones:

Tabla 3. HU Exportar animación .pc2.

Historia de Usuario	
Número:2	Nombre: Exportar archivo Collada .dae
Usuario: Todos los usuarios que interactúen con la herramienta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
<p>El usuario exporta el archivo .dae. Archivo que contendrá las mallas poligonales de los objetos 3D seleccionados en la escena. Se creará un objeto temporal para obtener el pivote en el eje de coordenadas [0, 0, 0], luego se unirán las distintas partes de las mallas poligonales que componen al personaje u objeto animado. Haciendo uso del exportador FBX Collada se exportarán las mallas poligonales previamente seleccionada.</p>	
Observaciones:	

Tabla 4. HU Exportar archivo Collada .dae.

Historia de Usuario	
Número:3	Nombre: Administrar la selección de objetos
Usuario: Todos los usuarios que interactúen con la herramienta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada:1	
<p>Descripción: El usuario selecciona las partes que componen al personaje u objeto animado, solo aquellas que sean geometría o malla. Si en la selección se comete el error de seleccionar algún objeto que no sea malla o geometría, el <i>plugin</i> debe enviar un mensaje advirtiendo del error en la selección, adicionando a la lista solo los objetos malla o geometría. Debe mostrarse la cantidad de objetos seleccionados.</p>	

Se debe permitir agregar cuantos elementos tenga un personaje, mientras cumplan con la condición anterior. Se debe permitir al usuario limpiar la lista completa, así como eliminar de uno en uno los elementos seleccionados en caso que haya sido agregada una malla o geometría que no pertenezca a las partes del personaje.

Observaciones:

Tabla 5. HU Administrar la selección de objetos.

Historia de Usuario	
Número:4	Nombre: Configurar opciones de animación .pc2
Usuario: Todos los usuarios que interactúen con la herramienta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
Descripción: El usuario configura los siguientes parámetros:	
<ul style="list-style-type: none"> ❖ <i>Frame</i> inicio ❖ <i>Frame</i> final ❖ Factor de Muestra 	
Además de estos parámetros deberá dar la posibilidad de actualizar los valores de inicio y final del <i>frame</i> .	
Observaciones:	

Tabla 6. HU Configurar opciones de animación .pc2.

Historia de Usuario	
Número:5	Nombre: Configurar nombre y ruta de los archivos a exportar
Usuario: Todos los usuarios que interactúen con la herramienta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Iteración Asignada: 1	
El usuario debe escribir el nombre del archivo que desea exportar en un edit.	

Además de la posibilidad de escoger el directorio o raíz donde quiera que esté el archivo.

Debe dársele la posibilidad de crear una nueva carpeta en caso que lo requiera.

El nombre especificado será el del archivo .pc2 y .dae, para evitar confusiones.

Observaciones:

Tabla 7. HU Configurar nombre y ruta de los archivos a exportar.

2.7. Fase de Planificación.

En esta fase se establece la prioridad de cada HU, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociadas a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según el alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación. [32]

2.7.1 Estimación de esfuerzo por Historia de Usuario.

Durante la fase de planificación se realiza la estimación del esfuerzo que costará implementar cada historia de usuario. Esto se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la

historia de usuario, por ejemplo, las pruebas unitarias, la integración y refactorización del código, y la preparación y ejecución de las pruebas de aceptación.

A continuación se muestra mediante una tabla la planificación de la estimación del esfuerzo para las diferentes historias de usuario:

No.	Historia de Usuario	Puntos de Estimación (Semanas)
1	Exportar animación .pc2	2
2	Exportar archivo Collada .dae	2
3	Administrar la selección de objetos	2
4	Configurar opciones de animación .pc2	1.5
5	Configurar nombre y ruta de los archivos a exportar	1.5

Tabla 8. Estimación del esfuerzo por Historia de Usuario.

2.7.2 Plan de iteraciones.

Después de identificar y detallar las historias de usuario, se procede a la elaboración del plan de iteraciones, donde se definen qué historias de usuario serán implementadas en cada iteración. Para el desarrollo del presente software se estableció la realización de dos iteraciones:

Iteración 1: se realizan las historias de usuario de mayor rango para el cliente, obteniendo la primera versión funcional de la aplicación, que sirve para captar nuevos requerimientos y para realizar pruebas eficaces.

Iteración 2: se desarrollan las historias de usuario de menor prioridad en el negocio, además se corrigen los errores y no conformidades de la anterior iteración. Indistintamente se prueba cada una de las funcionalidades efectuadas.

2.7.3 Plan de duración de las iteraciones.

El plan de duración de las iteraciones muestra cuanto se estima que dure cada iteración a efectuar, además del orden en que serán implementadas las historias de usuario en cada una de las iteraciones.

Iteración	Historias de usuarios	Duración total(semanas)
-----------	-----------------------	----------------------------

	Exportar animación .pc2	
1	Exportar archivo Collada .dae	6
	Administrar la selección de objetos	
	Configurar opciones de animación .pc2	
2	Configurar nombre y ruta de los archivos a exportar	3

Tabla 9. Plan de duración de las iteraciones.

2.7.4 Plan de Entrega.

Una vez que el cliente culmina la elaboración de las HU, se comienza con la creación del Plan de Entregas. El mismo se hace con la intención de que los programadores obtengan una estimación de dichas historias en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una.

Historias de usuario	Iteración 1 (02 /4/2012)	Iteración 2 (07/5/2012)
Exportar animación .pc2		
Exportar archivo Collada .dae	V1.0	Terminado
Administrar la selección de objetos		
Configurar opciones de animación .pc2		
Configurar nombre y ruta de los archivos a exportar	No empezado	V2.0

Tabla 10. Plan de Entrega.

2.8. Fase de diseño.

El diseño adecuado para el software es aquel que supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación del equipo de desarrollo y tiene el menor número posible de clases y métodos [33].

La metodología XP no requiere la descripción del sistema por medio de diagramas de clases utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y

comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

2.8.1 Arquitectura del sistema.

La arquitectura es la base estructural de un sistema informático, lo describe desde varios puntos de vista; contiene lineamientos muy importantes que establecen esquemas a los que se deben ajustar los componentes del sistema local variables.

El MAXScript es un lenguaje orientado a objetos que permite extender las posibilidades del software 3ds Max. La empresa Autodesk recomienda la estructura básica que deben de tener las Macros que se desarrollan en MAXScript (Ver Anexo 5) [34].

1. User-interface items
2. Rollouts
3. Functions
4. Event handlers

User interface ítems: Donde se definen los elementos de la interfaz de usuario.

Rollouts: Se utilizan como contenedoras de ayuda visual (botones, contadores, selectores de color, casillas y botones radiales de verificación) para que el usuario no tenga que trabajar directamente con el código. Las persianas pueden ir en ventanas flotantes o en el panel de comandos del 3ds Max como parte de las utilidades [35].

Functions: En las funciones se engloban un conjunto determinado de acciones las cuales pueden o no utilizar parámetros externos. Retornan valores que pueden ser almacenados en variables o arreglos y pasados a otras funciones como argumentos. Incluyen en su cuerpo variables, otras funciones y expresiones [36]. Las funciones creadas y utilizadas se describen en el epígrafe **3.1.3**.

Event handlers: Es una definición de la función local que despliega un servicio público o que se proporciona para manejar el procesamiento que ocurre cuando un usuario realiza una acción en un elemento de interfaz de usuario. Por ejemplo, los controladores de eventos se llaman cuando el usuario presiona un botón, abre o cierra la utilidad, o cambia de tamaño o se mueve una ventana flotante desplegada [37]. Los eventos creados y utilizados se describen en el epígrafe **3.1.3**

La propuesta se realizó siguiendo la Arquitectura definida por Autodesk para crear *plugin* para su software 3ds Max.

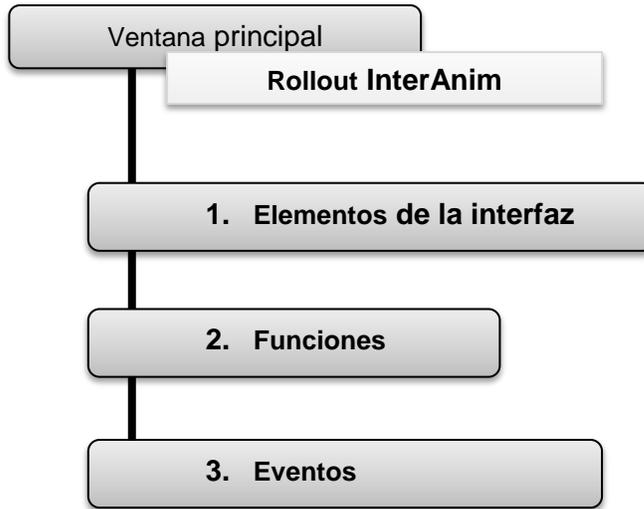


Figura 7. Estructura del script.

Un diagrama de clases del diseño “representa las especificaciones de las clases e interfaz del software” [38]. La Figura 8 muestra el diagrama de clases del diseño correspondiente al sistema desarrollado usando las clases del sistema 3ds Max.

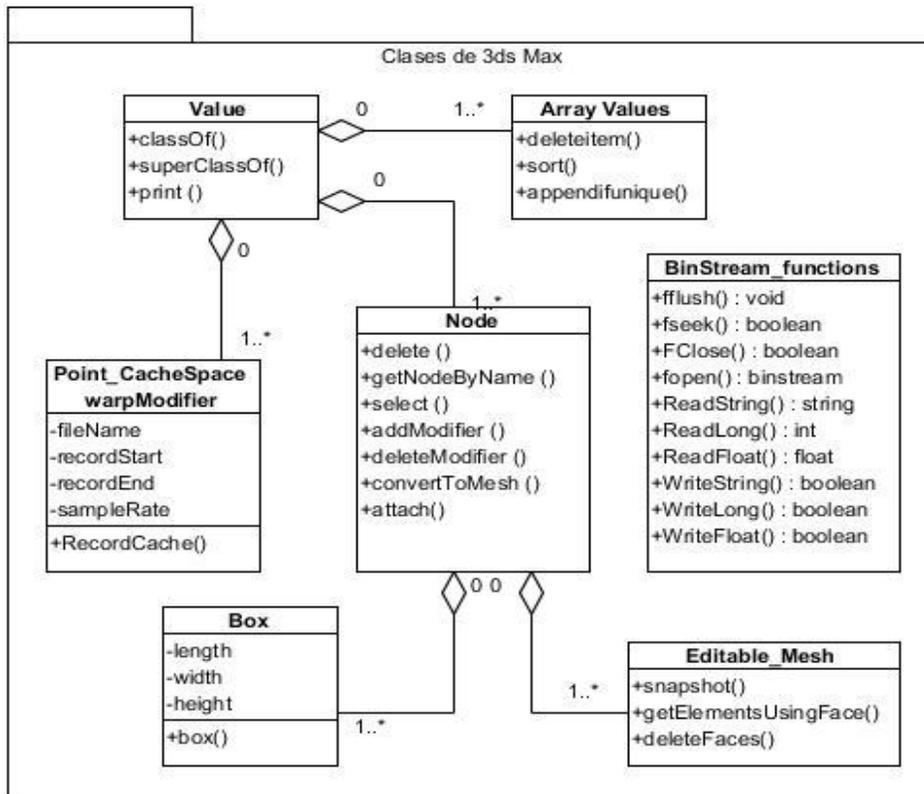


Figura 8. Diagrama de clases del diseño.

Consideraciones parciales.

Conociendo como se lleva a cabo el flujo actual de eventos, se elaboró una propuesta del sistema. Se realizó una descripción de las HU precisando por el cliente la prioridad de cada una y definiendo así el orden en el que serán implementadas las mismas. Se cuenta con cinco HU que serán implementadas en dos iteraciones. Quedó elaborado el modelo necesario para llevar a cabo la implementación del sistema, mediante la descripción de la arquitectura utilizada. Los formatos Point Cache 2 y Motion Designer Data son adecuados para realizar la exportación de 3ds Max a Blender, siendo usados para almacenar los datos de la solución propuesta.

Capítulo 3 . Implementación y Prueba. Resultados de la propuesta.

Introducción.

El objetivo de este capítulo es implementar la propuesta de solución elaborada anteriormente y comprobar que el sistema desarrollado cumple con cada requerimiento planteado por el cliente. Para ello, se llevan a cabo las tareas de ingeniería y se realizan las pruebas de aceptación.

3.1 Fase de implementación.

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta etapa, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto [39].

En la primera iteración se puede intentar establecer una arquitectura del sistema, que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor del negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Como parte de este plan se descomponen las HU en tareas de ingeniería, asignando a un grupo o una persona como responsable de su implementación. Estas tareas son para el uso estricto de los programadores, por lo que pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente. A continuación se detallan las tareas del desarrollo realizadas en cada una de las iteraciones.

3.1.1 Iteración 1.

La primera iteración tendrá como objetivo darle cumplimiento a las HU 1, 2 y 3 que representan un mayor valor para el cliente, pues con las mismas se conformará la base de la estructura del negocio. Estas recogen funcionalidades de gran importancia para el proyecto, pues a través de ellas se definen aspectos que serán utilizados luego por las demás funcionalidades.

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

No	Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
1	Exportar animación .pc2	2	2.5
2	Exportar archivo Collada .dae	2	2
3	Administrar la selección de objetos	2	2

Tabla 11. Tiempo de la implementación por HU de la iteración 1.

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo, que se realizaron para cumplir el propósito de la misma.

Historia de usuario	Tareas por HU
Exportar animación .pc2	<ol style="list-style-type: none"> 1) Guardar archivo Point_cache2 WSM con los parámetros definidos 2) Gestionar trabajo con archivo Point_cache2 abierto 3) Crear nueva cabecera y datos a partir de los datos almacenados.
Exportar archivo Collada .dae	<ol style="list-style-type: none"> 1) Exportar la selección de archivo activa.
Administrar la selección de objetos	<ol style="list-style-type: none"> 1) Añadir objetos de la selección. 2) Eliminar objetos de la selección 3) Comprobar tipo de objeto añadido.

Tabla 12. Tareas por HU en la iteración 1.

Tarea de Ingeniería	
Número de la tarea:1	Número de HU: 1
Nombre de la tarea: Guardar archivo Point_cache2 WSM con los parámetros definidos.	

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Realizar el guardado de los archivos .pc2 de las distintas partes de los objetos 3D aplicando un modificador Point Cache 2.	

Tabla 13. Tarea 1 de la HU 1

Tarea de Ingeniería	
Número de la tarea:2	Número de HU: 1
Nombre de la tarea: Gestionar trabajo con archivo Point_cache2 abierto.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Abrir ficheros .pc2 y leer la información, teniendo en cuenta que los parámetros sean correctos.	

Tabla 14. Tarea 2 de la HU 1

Tarea de Ingeniería	
Número de la tarea:3	Número de HU: 1
Nombre de la tarea: Crear nueva cabecera y datos a partir de los datos almacenados.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Se crea una nueva cabecera a partir de los datos leídos y se va actualizando la cabecera y los datos del archivo activo.	

Tabla 15. Tarea 3 de la HU 1

Tarea de Ingeniería	
Número de la tarea:1	Número de HU: 2
Nombre de la tarea: Exportar la selección de archivo activa.	

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: A través del exportador FXB Collada, exportar la selección activa que debe ser el conjunto de mallas que compone al personaje u objeto 3D.	

Tabla 16. Tarea 1 de la HU 2

Tarea de Ingeniería	
Número de la tarea:1	Número de HU: 3
Nombre de la tarea: Añadir objetos de la selección.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Los objetos 3D que han sido seleccionados deben ser añadidos al arreglo con el que se trabajará.	

Tabla 17. Tarea 1 de la HU 3

Tarea de Ingeniería	
Número de la tarea:2	Número de HU: 3
Nombre de la tarea: Eliminar objetos de la selección.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Permitir eliminar un objeto de la selección añadida actualmente. O eliminar la selección completa.	

Tabla 18. Tarea 2 de la HU 3

Tarea de Ingeniería	
Número de la tarea:3	Número de HU: 3
Nombre de la tarea: Comprobar tipo de objeto añadido.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Prevenir la selección de algún objeto que no sea de la clase geometría. Verificar que el objeto sea único y no se nombre como el archivo a exportar.	

Tabla 19. Tarea 3 de la HU 3

3.1.2 Iteración 2.

La segunda iteración está centrada en desarrollar la HU 4, 5 que son de gran importancia también para el cliente, aunque incluyen una menor complejidad de desarrollo.

No	Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
4	Configurar opciones de animación .pc2	1.5	1.5
5	Configurar nombre y ruta de los archivos a exportar	1.5	1

Tabla 20. Tiempo de la implementación por HU de la iteración 2

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo, que se realizaron para cumplir el propósito de la misma.

Historia de usuario	Tareas por HU
Configurar opciones de animación .pc2	<ol style="list-style-type: none"> 1) Actualizar el rango de animación. 2) Pasar los parámetros al archivo.
Configurar nombre y ruta de los archivos a exportar	<ol style="list-style-type: none"> 1) Guardar archivo con el nombre especificado. 2) Guardar en directorio especificado. 3) Comprobar permiso y previa existencia de archivos.

Tabla 21. Tareas por HU en la iteración 2

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Tarea de Ingeniería	
Número de la tarea:1	Número de HU: 4
Nombre de la tarea: Actualizar el rango de animación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Actualizar los datos de rango de inicio y fin, según se ha dispuesto en la herramienta 3ds Max.	

Tabla 22. Tareas 1 de la HU 4

Tarea de Ingeniería	
Número de la tarea:2	Número de HU: 4
Nombre de la tarea: Pasar los parámetros de animación al archivo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Pasar al archivo a exportar los datos de animación de .pc2, factor de muestra, rango de inicio y rango de fin.	

Tabla 23. Tarea 2 de la HU 4

Tarea de Ingeniería	
Número de la tarea:1	Número de HU: 5
Nombre de la tarea: Guardar archivo con el nombre especificado.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Pasar al archivo a crear el nombre especificado por el usuario.	

Tabla 24. Tarea 1 de la HU 5

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Tarea de Ingeniería	
Número de la tarea:2	Número de HU: 5
Nombre de la tarea: Guardar en directorio especificado.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Permitir elegir el directorio donde se guarda la información o crear carpeta donde se desee.	

Tabla 25. Tarea 2 de la HU 5

Tarea de Ingeniería	
Número de la tarea:3	Número de HU: 5
Nombre de la tarea: Comprobar permiso y previa existencia de archivos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Programador(es) responsable(s): Yunior Frómeta Carbonell	
Descripción: Comprobar la existencia de un archivo con el mismo nombre y los permisos del usuario a escribir en el directorio que se eligió o creó.	

Tabla 26. Tarea 3 de la HU 5

Las iteraciones de desarrollo sobre el sistema, permitieron que al finalizar se obtuviera un producto con todas las características deseadas por el cliente.

Cabe destacar que el desarrollo de la solución se dividió en dos iteraciones para ordenar la implementación, realizándose pequeñas entregas al cliente al final de cada una de ellas, para recibir la aceptación del mismo, sin embargo el despliegue del producto en su entorno real de explotación no se realizará hasta terminada la última iteración.

3.1.3 Descripción de las funciones y eventos.

En esta herramienta no se implementaron clases, por lo que en este epígrafe se muestra la tabla generada en Visual Paradigm, que contiene las funciones implementadas en el MacroScript, cada una con su respectiva descripción.

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Nombre de la función	Descripción
function UpdateUI	Actualiza los elementos de la interfaz.
function ProveObjList	Verifica que el nodo sea único.
function ProveFileDir	Verifica el directorio elegido, el derecho de escritura y la previa existencia de archivos.
function SavePC2	Salva un archivo .pc2 WSM por objeto.
function JoinMesh	Ensambla un objeto 3D a partir de la lista de objetos, poniendo el pivote en el origen de coordenada.
function AssemblingPC2	Ensambla una lista de archivos .pc2 en uno solo.
function AssemblingAniMesh	Ensambla una lista de objetos, crea y ensambla sus Point Cache, aplica a la malla unificada el modificador con el .pc2 creado.

Tabla 27. Funciones implementadas en el *script*.

Nombre del evento	Descripción
btnAddObjectsToList	Añade los objetos seleccionados en el visor 3D al arreglo con el que se trabajara.
btnClearList	Limpia la lista de objetos seleccionados.
btnDelObjFromList	Elimina el objeto seleccionado.
btnRefreshRange	Actualiza el rango de inicio y fin de la herramienta.
btnSelectPathButton	Selecciona el directorio donde se guardara el archivo a crear y con qué nombre.
btnSalvarGeo	Exporta las mallas poligonales 3D haciendo uso del exportador FBX Collada.
btnBakeAnimation	Exporta el archivo .pc2.
btnHelp	Muestra una ayuda básica.

Tabla 28. Eventos implementados en el *script*.

3.2 Fase de pruebas

XP propone la realización de las pruebas unitarias y las pruebas de aceptación. Las primeras son desarrolladas por el programador para verificar que el código funciona, mientras que las segundas son utilizadas para probar que la aplicación contiene las funcionalidades deseadas por el cliente.

Por tener mayor interés para la investigación solo se expondrán las pruebas de aceptación, las cuales demostraron que el *plugin* posee las funcionalidades que eran de utilidad para el cliente.

3.2.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario y significan el grado de satisfacción del cliente con el producto desarrollado. Para la presente aplicación, se realizarán seis casos de pruebas.

Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas pruebas es garantizar que los requisitos han sido cumplidos y que el sistema es aceptable [31]. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación [40].

Caso de prueba de Aceptación	
Número: 1	Historia de Usuario: 3
Nombre de la persona que realiza la prueba: Yunior Frómata Carbonell	
Nombre de la prueba: Adicionar objetos 3D (Ver Anexo 1).	
Descripción: El caso de prueba permite al usuario adicionar objetos 3D de la selección al arreglo de objetos con el que se trabajará.	
Condiciones de ejecución: Solo deben ser seleccionadas mallas 3D.	
Entradas/Pasos de ejecución: <ul style="list-style-type: none">• El usuario escoge en el visor 3D los objetos a los que desea realizarle el proceso.• El <i>plugin</i> debe adicionar solo los tipos de objetos malla 3D y advertir en caso que no lo sean.	
Resultado esperado: El <i>plugin</i> debe visualizar en el listbox los objetos adicionados	

Evaluación de la prueba: Satisfactorio

Tabla 29. Caso de prueba de Aceptación "Adicionar objetos 3D".

Caso de prueba de Aceptación	
Número: 2	Historia de Usuario: 3
Nombre de la persona que realiza la prueba: Yunió Fróméta Carbonell	
Nombre de la prueba: Eliminar objetos 3D.	
Descripción: El caso de prueba permite al usuario eliminar objetos 3D de la selección realizada.	
Condiciones de ejecución: Puede eliminar uno a uno o todos de una vez.	
Entradas/Pasos de ejecución: <ul style="list-style-type: none">• El usuario escoge en el listbox el objeto a eliminar y presiona "-" para eliminar.• El <i>plugin</i> debe eliminar solo el objeto seleccionado.• El usuario presiona el botón "Limpiar" para eliminar todos los objetos.• El <i>plugin</i> debe eliminar todos los objetos del listbox	
Resultado esperado: El <i>plugin</i> debe visualizar en el listbox los objetos eliminados.	
Evaluación de la prueba: Satisfactorio	

Tabla 30. Caso de prueba de Aceptación "Eliminar objetos 3D".

Caso de prueba de Aceptación	
Número: 3	Historia de Usuario: 4
Nombre de la persona que realiza la prueba: Yunió Fróméta Carbonell	
Nombre de la prueba: Actualizar el rango de animación (Ver Anexo 2).	
Descripción: El caso de prueba permite al usuario actualizar el rango de animación	
Condiciones de ejecución: El rango de animación del software 3ds Max es distinto al de la aplicación.	

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Entradas/Pasos de ejecución:

- El usuario presiona el botón “Actualizar” para que el rango de animación del software 3ds Max y el del *plugin* sean los mismos.
- El *plugin* debe actualizar el rango de la aplicación, reiniciando el *plugin*.

Resultado esperado: El *plugin* debe visualizar el rango de animación actual.

Evaluación de la prueba: Satisfactorio

Tabla 31. Caso de prueba de Aceptación "Actualizar el rango de animación".

Caso de prueba de Aceptación	
Número: 4	Historia de Usuario: 5
Nombre de la persona que realiza la prueba: Yunior Frómeta Carbonell	
Nombre de la prueba: Configurar nombre y ruta de los archivos a exportar (Ver Anexo 3).	
Descripción: El caso de prueba permite al usuario configurar el nombre y directorio de los archivos a exportar.	
Condiciones de ejecución:	
<ul style="list-style-type: none">• El nombre del archivo no debe ser el mismo que el de alguno de los objetos 3D seleccionados.• El usuario debe tener privilegios administrativos para crear archivos o directorios.	
Entradas/Pasos de ejecución:	
<ul style="list-style-type: none">• El usuario nombra al archivo con el nombre que desee, excepto el de algún objeto existente en la selección.• El usuario escoge el directorio donde desea exportar los ficheros o puede crearlo.• El <i>plugin</i> advierte en caso que no tenga privilegios administrativos a la hora de crear carpeta.	
Resultado esperado: El <i>plugin</i> debe crear la carpeta que el usuario indicó, para posteriormente escribir el archivo.	
Evaluación de la prueba: Satisfactorio	

Tabla 32. Caso de prueba de Aceptación "Configurar nombre y ruta de los archivos a exportar".

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Caso de prueba de Aceptación	
Número: 5	Historia de Usuario: 2
Nombre de la persona que realiza la prueba: Yunió Fróméta Carbonell	
Nombre de la prueba: Exportar archivo Collada .dae (Ver Anexo 4).	
Descripción: El caso de prueba permite al usuario exportar las mallas poligonales 3D del objeto creado a partir de la unión de los objetos seleccionados previamente.	
Condiciones de ejecución: <ul style="list-style-type: none">• Este proceso puede ocurrir una vez generado el archivo con la animación .pc2 y la unión de los objetos 3D seleccionados.	
Entradas/Pasos de ejecución: <ul style="list-style-type: none">• El usuario exporta la animación .pc2.• El <i>plugin</i> automáticamente une las mallas poligonales de los objetos 3D seleccionados.• El <i>plugin</i> invoca el exportador FBX Collada para exportar el fichero .dae con la nueva malla poligonal 3D.	
Resultado esperado: El <i>plugin</i> debe crear el archivo .dae en el directorio indicado previamente y con el nombre indicado por el usuario.	
Evaluación de la prueba: Satisfactorio	

Tabla 33. Caso de prueba de Aceptación "Exportar archivo Collada .dae".

Caso de prueba de Aceptación	
Número: 6	Historia de Usuario: 1
Nombre de la persona que realiza la prueba: Yunió Fróméta Carbonell	
Nombre de la prueba: Exportar animación .pc2 (Ver Anexo 4).	
Descripción: El caso de prueba permite al usuario exportar la animación .pc2 de los objetos 3D seleccionados previamente.	
Condiciones de ejecución:	

- Este proceso solo se le realiza a mallas poligonales 3D que no pierdan información durante el trascurso de la animación.

Entradas/Pasos de ejecución:

- El usuario exporta la animación .pc2.
- El *plugin* ensambla los archivos .pc2 previamente exportados, ensamblando en uno.

Resultado esperado: El *plugin* debe crear el archivo .pc2 en el directorio indicado previamente y con el nombre indicado por el usuario.

Evaluación de la prueba: Satisfactorio

Tabla 34. Caso de prueba de Aceptación "Exportar animación .pc2".

Las pruebas de aceptación arrojaron como resultados que todos los requisitos han sido cumplidos satisfactoriamente.

3.3 Resultados de la solución.

Con el objetivo de hacer una evaluación de los resultados de las pruebas de aceptación diseñadas se analizaron los tiempos necesarios para realizar la exportación de dos escenas animadas de complejidad media y alta respectivamente (Ver Anexo 6). La determinación del grado de complejidad de la escena animada fue ofrecido por dos especialistas en animación [41] [42]. La escena animada de complejidad alta pertenece al ICAIC, específicamente al proyecto audiovisual que realiza la serie animada titulada "Cuentos para una abuela" [43].

Para realizar la evaluación de la solución propuesta se solicitó a cinco técnicos realizar el proceso de exportación de animación de 3ds Max a Blender de forma manual y luego utilizando el *script* desarrollado. La unidad de tiempo utilizada fue en minutos (min). En las siguientes tablas se muestran los tiempos alcanzados.

Usuario	Usando <i>plugin</i> InterAnim	Manualmente
Técnico 1	1	3,8
Técnico 2	0,9	3,9
Técnico 3	0,85	4,1
Técnico 4	1	4,25
Técnico 5	0,95	3,95

Capítulo 3. Implementación y Prueba. Evaluación de la propuesta.

Tabla 35. Tiempos alcanzados en la exportación de animación de complejidad media.

Usuario	Usando <i>plugin</i> InterAnim	Manualmente
Técnico 1	6,7	22,8
Técnico 2	6,3	20,2
Técnico 3	7	21,5
Técnico 4	6	19,6
Técnico 5	6,5	23,4

Tabla 36. Tiempos alcanzados en la exportación de animación de complejidad alta.

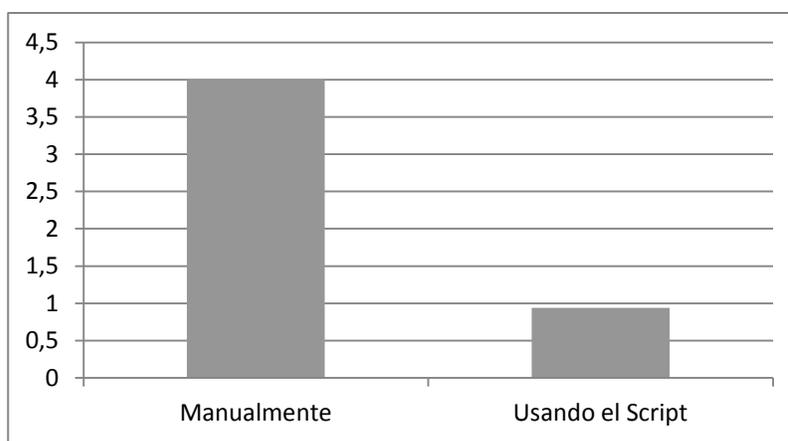


Tabla 37. Evaluación del tiempo dedicado a la tarea de exportar animación de complejidad media.

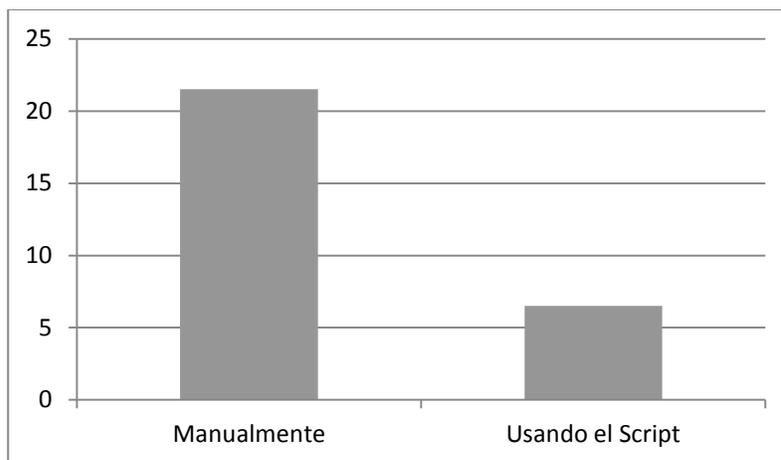


Tabla 38. Evaluación del tiempo dedicado a la tarea de exportar animación de complejidad alta.

Los resultados arrojan que con el *plugin* desarrollado el proceso de exportar animaciones y mallas poligonales de 3ds Max a Blender promedia 0,94 min en una escena de complejidad media y 6,5 min en una escena de complejidad alta. Mientras que el mismo proceso realizado de forma manual promedia 4 min para escenas animadas de complejidad media y 21,5 min para escenas animadas de complejidad alta. Estos resultados evidencian la rapidez con que el *plugin* realiza el proceso, con un incremento de la velocidad de exportación en un 76,5% para escenas animadas de complejidad media y un 69,8% para escenas animadas de complejidad alta.

El *plugin* fue probado en el ICAIC y recibido por el representante de la entidad (Ver Anexo 8). Comprobando los objetivos trazados, se emitió una carta de aceptación donde se expresa la satisfacción con el producto.

Consideraciones parciales

Se implementó el sistema y se efectuaron las pruebas correspondientes para comprobar que cumple con los requisitos planteados. Se realizó una evaluación al *plugin* desarrollado arrojando resultados positivos, por lo que puede ser utilizado en el proceso de exportación de animaciones y mallas poligonales.

Conclusiones

Con la terminación del trabajo se logró implementar un *plugin* que permite exportar animaciones y mallas poligonales de objetos animados del software Autodesk 3ds Max a Blender, con lo que se cumplió el objetivo general de la investigación. Al realizar pruebas con la solución propuesta usando animaciones de complejidad media y alta:

- ❖ Los resultados obtenidos no presentaron ningún error al ser probados en Blender.
- ❖ Se logró disminuir el tiempo necesario para realizar la exportación en más de un 72% en comparación con el tiempo utilizado al realizar la exportación de forma manual.

Recomendaciones

Se recomienda como parte del soporte del *plugin*:

1. Implementar las funcionalidades que permitan exportar la animación de objetos 3D tales como: cámaras, luces o ayudantes.
2. Implementar las funcionalidades que permitan exportar en el formato .mdd.
3. Utilizar y continuar ampliando el plugin desarrollado, en el Departamento de Visualización y Realidad Virtual del CEDIN.

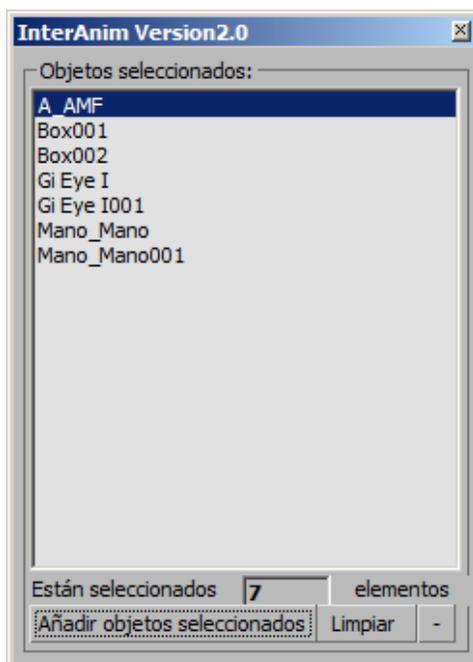
Referencias Bibliográficas

1. WordReference.com - Diccionario de la lengua española. [En línea] <http://www.wordreference.com/definicion/humanoide>.
2. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Autodesk 3ds Max Help > Animation > Animation Concepts and Methods > Animation Concepts.
3. **CG COOKIE**. *Animation Training Series*. [DVD] s.l. : Blender Cookie, 2011.
4. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Creating Geometry > Systems > Bones System.
5. —. Autodesk 3ds Max® 2011 Help. Modifiers > Object-Space Modifiers > Morpher Modifier.
6. *CG Academy - Intermediate Rigging - Head Setup*. 2006.
7. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Autodesk 3ds Max Help > Modifiers > Object-Space Modifiers > Cloth and Garment Maker Modifiers > .
8. —. History of Autodesk 3ds Max. [En línea] 2012. [Citado el: 12 de 4 de 2012.] <http://area.autodesk.com/maxturns20/history>.
9. —. Autodesk 3ds Max® 2013. [En línea] 2012. [Citado el: 28 de 5 de 2012.] <http://store.autodesk.com/store/adskus/pd/productID.247426800>.
10. **Blender Foundation**. Blender. *Blender*. [En línea] Blender Foundation, 23 de Marzo de 2011. [Citado el: 23 de Marzo de 2011.] <http://www.blender.org/>.
11. **3D Profesional**. www.3dprofesional.com. *software*. [En línea] 3D Profesional, 2011. [Citado el: 14 de Marzo de 2011.] <http://www.3dprofesional.com/software/index.html>.
12. **New Tek**. LightWave. [En línea] New Tek, 8 de 1 de 2011. [Citado el: 18 de 4 de 2012.] www.newtek.com/lightwave/.
13. **The Labs - Design & Functionality For The Net**. 3D Studio File Format (3ds). [En línea] 5 de 10 de 1999. [Citado el: 16 de 4 de 2012.] <http://www.the-labs.com/Blender/3dsspec.html>.
14. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Autodesk 3ds Max Help > Managing Scenes and Projects > Geometry File Formats > Wavefront (OBJ) Files > .
15. —. Autodesk 3ds Max® 2011 Help. Autodesk 3ds Max Help > Managing Scenes and Projects > Geometry File Formats > AutoCAD Interchange (DXF) Files > .
16. **Wolfram Research, Inc.** reference.wolfram.com. [En línea] Wolfram Research, Inc., 2012. [Citado el: 4 de 5 de 2012.] <http://reference.wolfram.com/mathematica/ref/format/LWO.html>.

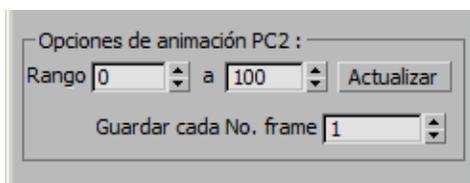
17. **Khronos Group**. Collada.org. [En línea] collada.org, 10 de 10 de 2011. [Citado el: 8 de 5 de 2012.] https://collada.org/mediawiki/index.php/COLLADA_-_Digital_Asset_and_FX_Exchange_Schema.
18. **NewTek**. www.newtek.com. [En línea] NewTek, 2012. [Citado el: 9 de 5 de 2012.] www.newtek.com/solutions/3d-animation/lw-feature-film.html.
19. **Blender Foundation**. A look at point cache. *code.blender.org*. [En línea] 2012. [Citado el: 10 de 4 de 2012.] <http://code.blender.org/index.php/2011/01/a-look-at-point-cache/>.
20. **John Burnett** . www.maxplugins.de. [En línea] Autodesk, 2000. [Citado el: 12 de 3 de 2012.] http://www.maxplugins.de/r6_files/burnett/PointCache2_R6.zip.
21. **Thinkbox Software**. www.thinkboxsoftware.com. [En línea] Thinkbox Software, 2012. [Citado el: 15 de 4 de 2012.] <http://www.thinkboxsoftware.com/xmesh/>.
22. **Boomer Labs**. www.boomerlabs.com. [En línea] Boomer Labs, 2012. [Citado el: 15 de 5 de 2012.] <http://www.boomerlabs.com/cart/home.php?cat=6>.
23. **ef9.com**. www.ef9.com. [En línea] ef9, 2012. [Citado el: 18 de 5 de 2012.] http://www.ef9.com/ef9/PO1.5/PointOven_DocIndex.html.
24. **Letelier, P y Penadés, M C**. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia : s.n.
25. **Figueroa, R G, Solís, C J y Cabrera, A A**. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. Universidad Técnica Particular de Loja : s.n.
26. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Basic Concepts and Methods.
27. **python.org**. www.python.org. [En línea] 2012. [Citado el: 19 de 5 de 2012.] <http://www.python.org/>.
28. **Visual Paradigm**. *Ayuda en línea de Visual Paradigm*. [En línea] 2011. [Citado el: 15 de 11 de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.
29. **I. Jacobson, R. Booch, J. Rumbaugh**. *El proceso unificado de desarrollo de software*. 2000. ISBN: 84-7829-036-2.
30. **Garcerant, Iván**. *Tecnología y Synergix*. [En línea] 5 de 8 de 2008. [Citado el: 09 de 11 de 2011.] <http://synergix.wordpress.com/2008/08/05/un-buen-requisito-debe-ser-atomico/>.
31. **BECK, K**. *Planeando en Programación Extrema*. 2000.
32. **Letelier, P y Penadés, M C**. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* . Universidad Politécnica de Valencia : s.n.
33. **Beck, K**. *Extreme Programming Explained. Embrace Change*. s.l. : Pearson Education Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.

34. **Autodesk®**. Autodesk 3ds Max® 2011 Help. Visibility of Locals, Functions, Structures, and User-Interface Items in Rollout Code.
35. —. Autodesk 3ds Max® 2011 MAXScript Reference. MAXScript Tools and Interaction with 3ds max > Creating MAXScript Tools > Scripted Utilities and Rollouts.
36. —. Autodesk 3ds Max® 2011 MAXScript Reference. MAXScript Language Reference > Creating Functions.
37. —. Autodesk 3ds Max® 2011 Help. MAXScript Language Reference > Rollout Clauses.
38. **Larman, Craig**. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Prentice Hall, 2006. pág. 590.
39. **Letelier, Patricio**. *Metodologías Ágiles en el Desarrollo de Software*. 2008.
40. **Joskowicz, J**. *Reglas y Prácticas en eXtreme Programming*. 2008.
41. **Solarana, Manuel Alberto**. *Especialista en animación*. [entrev.] Yunior Frómeta Carbonell. La Habana, 1 de 6 de 2012. "Escenas 3D animadas de baja, media y alta complejidad."
42. **Pérez González, Jerzy** . *Especialista en animación*. [entrev.] Yunior Frómeta Carbonell. La Habana, 2012. "Escenas 3D animadas de baja, media y alta complejidad."
43. **ICAIC**. Estudios de Animación. ICAIC. [En línea] ICAIC, 2011. [Citado el: 12 de 5 de 2012.] <http://cubacine.cult.cu/dibujosanimados/filmes/muestrafilme.php?CodigoFilme=0350>.
44. **Levy, Colin**. *Sintel*. [digital]. [prod.] Ton Roosendaal. The Durian Open Movie Project; Blender Foundation, 2010.

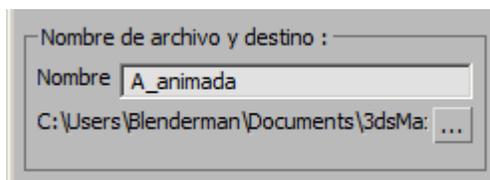
Anexos



Anexo 1. Administrar selección de objetos.



Anexo 2. Configurar opciones de animación .pc2.



Anexo 3. Configurar nombre del archivo y directorio.

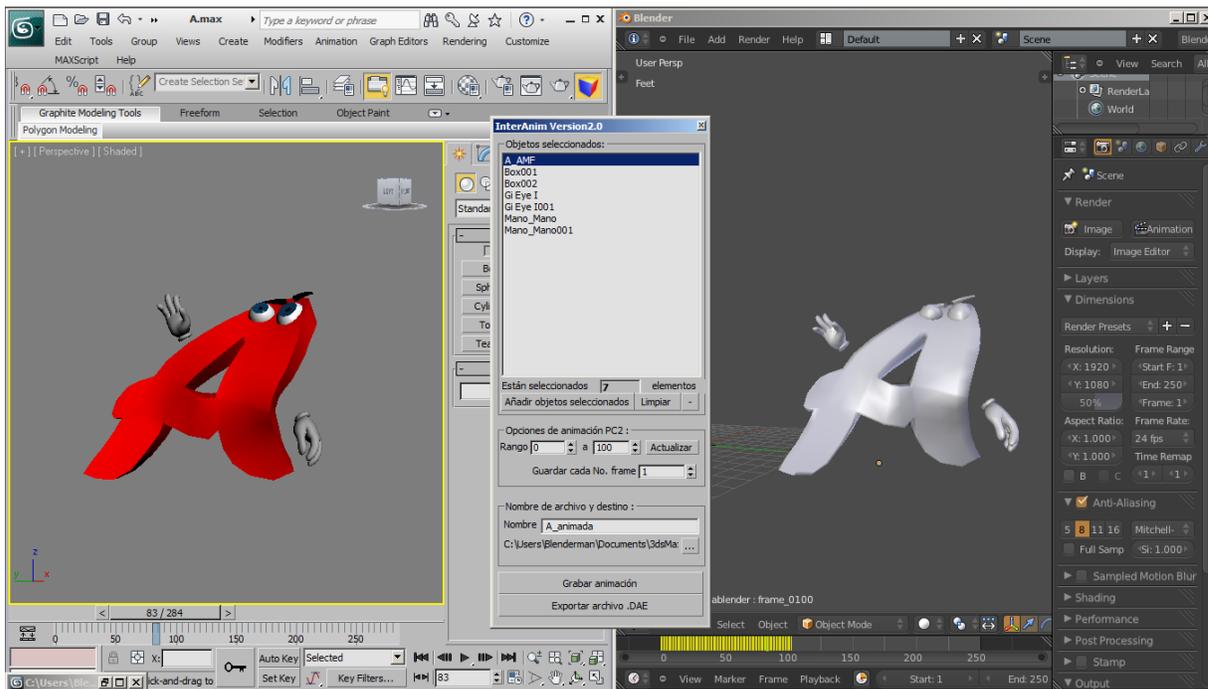


Anexo 4. Exportar Animación .pc2 y Exportar fichero .dae.

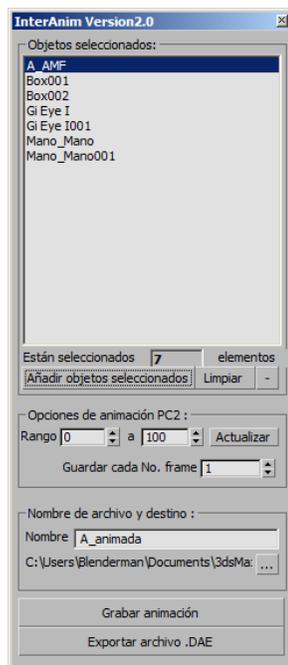
```

C:\Users\Blenderman\AppData\Local\Autodesk\3dsMax\2012 - 64bit\enu\UI\usermacros\InterAnim-InterAnim.mcr * MAXScript
File Edit Search View Tools Options Language Windows Help
InterAnim-InterAnim.mcr *
14
15 macroScript InterAnim
16 category:"File"
17 tooltip:"InterAnim"
18 icon:#("BodyObjects",2)
19 (
20
21 --VAR
22
23 global rollInterAnim
24 --global rollInterAnim
25
26
27 rollout rollInterAnim "InterAnim Version2.0" width:240 height:550
28 (
29
30 -----
31 --VARIABLES LOCALES
32 -----
33
34 local rolloutAbierto = false -- indica si el rollout está abierto
35 local cont
36 local ObjsABake = #() -- objetos a los que hacer bake
37
38 local RangoInicio = animationrange.start
39 local RangoFin = animationrange.end
40 local DirText = getDir #export
41 local ArchivoText = "AnimacionBaked"
42
43 -----
44 --//////////////////// Interface //////////////////////////////////////
45 -----
46
47 groupbox GpAnnadirObjs "Lista de Objetos seleccionados:" pos:[5,5] width:232 height:310
48 listbox LxObjs3d "" pos:[8,20] width:225 height:19
49 label lblCant1 "Estan seleccionados" pos:[10,275] width:100 height:16
50 edittext editcantobj pos:[115,275] width:30 height:18 readOnly:true
51 label lblCant2 "elementos" pos:[145,275] width:75 height:16
52 button btnAddObjsToList "Añadir objetos seleccionados" pos:[8,290] width:150 height:20 toolTip:"Añadir los elementos se
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589

```



Anexo 6. Uso de la solución propuesta, *plugin* InterAnim.



Anexo 7. Interfaz de la solución propuesta.

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “**Plugin para exportar animaciones y mallas poligonales de AutoDesk 3DsMax a Blender**”, fue desarrollado para aplicarse en los Estudios de Animación del ICAIC. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
 Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma permitirán a la entidad disponer de esta herramienta para exportar animaciones y mallas poligonales del software AutoDesk 3DsMax hacia el software Blender en los formatos 3D Point Cache 2 y Collada respectivamente de forma ágil y segura.

Y para que así conste, se firma la presente a los:

21 días del mes de junio del año 2012.

Armando Alba Noguera
 Representante de la entidad

Subdirector de Producción
 Cargo

[Firma]
 Firma



[Cuño]
 Cuño

Glosario de términos.

Animación: Procedimiento de diseñar los movimientos de los personajes o de los objetos y elementos.

Biped: Herramienta del 3ds Max que permite crear una jerarquía esquelética para personajes bípedos, que pueden ser animados rápidamente usando una amplia variedad de métodos.

Bípedo: Dícese de los seres que poseen dos pies y estructura simétrica con respecto al centro.

IDE: *Integrated Development Environment*. Es un sistema integrado que permite la generación, compilación y depuración de código.

Malla poligonal: (del inglés: *polymesh* o *mesh*) es una superficie creada mediante un método tridimensional generado por sistemas de vértices posicionados en un espacio virtual con datos de coordenadas propios.

Morph (morphing o blendshape): términos derivados de metamorfosis, lo que significa cambiar la forma física de un objeto. Es usado generalmente para crear expresiones Facial.

Nodos: Todos los objetos que se representan en la memoria del *3ds Max* están formados por nodos. Contienen geometrías, transformaciones y proveen los bloques de construcción para los vínculos jerárquicos.

Simulación: Se trata de la representación simplificada, mediante un modelo, de la realidad de un proceso.

Script: secuencia de instrucciones para automatizar una tarea. Son piezas de texto que contienen instrucciones codificadas para una aplicación particular.

Renderizado: Proceso de crear la imagen en un *Swcc3D*, utilizando un motor de render y los componentes de la escena: modelos, materiales, UV, texturas, simulaciones, etc...

Rigify: *Plugin* de Blender que permite crear sistemas de huesos humanoide. Genera el sistema completo, huesos deformadores, ayudantes, controles y restricciones.

Vértice: Es un punto en el espacio cuya única propiedad es su posición. Forman las estructuras básicas de creación de los objetos geométricos