

“Evolution”, Aplicación Tridimensional Interactiva Visual basada en el Framework XNA.

C. Viteri, N. Ortiz, G. Raura, D. Martínez

Departamento de Ciencias de la Computación, Escuela Politécnica del Ejército, Sangolquí-Ecuador

{cviteri,nortiz,georaura,dmartinez}@espe.edu.ec

RESUMEN: El presente proyecto consiste en el diseño y construcción de una aplicación tridimensional interactiva denominada “Evolution”, cuyo objetivo es crear un videojuego con un enfoque educativo que ayude a crear conciencia sobre el cuidado de la Naturaleza. Evolution simula un medio ambiente en donde los habitantes de un planeta tratan de sobrevivir ante los posibles fenómenos naturales que amenazan su evolución. En este entorno el usuario es capaz de producir y controlar fenómenos naturales de acuerdo al comportamiento de la población. Evolution es un videojuego multi-consola, diseñado con la combinación de 3D Studio Max y Torque Game Engine Advanced. Fue desarrollado mediante Microsoft XNA Framework 2.0, que aplica conceptos como la teoría de juegos, patrones de comportamiento e inteligencia artificial, para definir la conducta de los personajes que conforman el videojuego en un entorno tridimensional controlado.

SUMMARY: This project involves the design and construction of a three-dimensional interactive application called "Evolution", which aims to create a game with an educational approach to help raise awareness about caring for nature. Evolution simulates an environment where the inhabitants of a planet trying to survive against potential natural disasters that threaten their evolution. In this environment the user is able to produce and control natural phenomena according to the behavior of the population. Evolution is a multi-console game, designed with a combination of 3D Studio Max and Torque Game Engine Advanced. It was developed using Microsoft XNA Framework 2.0, which applies concepts such as game theory, behavioral patterns and artificial intelligence, to define the behavior of characters that make up the game in a controlled three-dimensional environment.

1. INTRODUCCION

La programación Los video juegos son una área de las ciencias de la computación que utilizan modelos matemáticos para manipular imágenes sobre una pantalla, estudiar sus interacciones en estructuras formalizadas y llevar a cabo procesos de decisión. Estos modelos son validados mediante la implementación de programas de computación destinados al entretenimiento. En la actualidad existen diversos géneros de video-juegos como aventura, estrategia, lucha, terror, educación, etc. En este último contexto, recientemente, la comunidad científica ha indagado en las relaciones existentes entre la educación y los videojuegos, tanto realizando investigaciones sobre el contexto educativo de los videojuegos, así como sobre los efectos educativos de los mismos.

Sin embargo el emprender con el desarrollo de videojuegos demanda un alto nivel de conocimientos en diversas áreas, pues no sólo se centra en la programación y en el uso de herramientas para creación de software, sino más bien, su concepto se extiende a otras ramas como son el diseño y modelado en 3D. Otro problema no menos importante es el presupuesto que se requiere para llevar a cabo los procesos de pre-producción, producción y post-producción que contemplan las fases de creación y modelado de los elementos visuales, del código y del marketing, donde en conjunto alcanzan cifras millonarias. Así por ejemplo, la producción de Halo 3 costó alrededor de 30 millones de dólares, más otros cerca de 20 millones de publicidad [1], mientras que el costo de Gears of War fue de 10 millones de dólares, siendo ambos juegos de Microsoft para el Xbox 360.

Ante esta problemática, la presente investigación consiste en diseñar e implementar una aplicación tridimensional interactiva denominada “Evolution”, cuyo objetivo primordial es el crear un videojuego con un enfoque educativo orientado a la conservación del medio ambiente. Para llevarlo a cabo se utilizó la herramienta Microsoft XNA Framework 2.0, que aplica conceptos como la teoría de juegos y patrones de comportamiento combinándola con Inteligencia Artificial, para definir la conducta de los personajes que conforman el videojuego, y que en conjunto con las herramientas de diseño 3D Studio Max y Torque Game Engine Advanced [2] contribuyan al diseño de este entorno tridimensional controlado.

La premisa de nuestra contribución es utilizar los video-juegos, de una manera original y divertida, para inculcar valores a los jugadores para que tomen conciencia y conozcan el daño que ocasiona al medio ambiente el abuso y la mala gestión de los recursos naturales y las consecuencias que esto provoca a la humanidad.

El resto del artículo ha sido organizado de la siguiente manera: La sección 2 describe los fundamentos técnicos empleados tanto en el diseño como en el desarrollo del video-juego. La sección 3 detalla el proceso de desarrollo del video-juego dividido por fases. En la sección 4 se detallan los resultados obtenidos tanto en el diseño como en el proceso de desarrollo. En la sección 5 se presentan algunos trabajos relacionados. Finalmente en la sección 6 se presentan las conclusiones y líneas de trabajo futuro sobre la base de los resultados obtenidos.

2. METODOS, TECNICAS Y HERRAMIENTAS

2.1 Teoría de Juegos:

- **Teoría de Juegos:** Permite analizar el comportamiento estratégico de los jugadores que intervienen en el juego, estudiando situaciones que involucran conflictos de intereses, estrategias óptimas, trampas, así como el comportamiento de individuos en el juego.
- **Matemática 3D:** Abarca todo lo referente a lugares, distancias, ángulos precisos y matemática en espacios 3D. El uso más frecuente se encuentra dentro del denominado Sistema Cartesiano Coordinado aplicado en el diseño de personajes en 3D.
- **Vector:** Un vector es representado por una flecha dirigida hacia un punto en el espacio. En Evolution el uso de vectores permite obtener el cambio de posición de un carácter.
- **Forma Normal del Juego:** Definida también como forma estratégica del juego, consiste en una matriz que contiene los jugadores, las estrategias empleadas y las recompensas ofrecidas.
- **Estados:** Representa el comportamiento del personaje en un momento dado. Cada estado reacciona de manera diferente a los eventos y está asociado a una animación diferente.
- **Autómata:** Constituye una máquina que se encuentra en un estado particular, y que ante determinados eventos cambia de un estado a otro. En Evolution cada personaje actuará como un autómata.
- **Máquina de Estados:** Es un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen no sólo de las señales de entradas actuales sino también de las anteriores.

- **Autómata Finito** (FSM Finite State Machine): Es un modelo matemático para una máquina de estado finita. Una FSM es una máquina que, dada una entrada de símbolos, "salta" a través de una serie de estados de acuerdo a una función de transición. Esta función le dice al autómata a qué estado cambiar dados un estado y símbolo determinados.
- **Agentes Inteligentes:** En Evolution es un habitante que se comporta como una entidad que percibe su entorno y actúa en consecuencia decidiendo que hacer por medio de reglas de lógica propias del agente.
- **Skybox:** Tiene como objetivo, encerrar la escena principal del juego, con una figura básica, el cual se encuentra texturizado internamente con imágenes de ambientes exteriores en sus cuatro planos.
- **Height map:** Es una imagen en blanco y negro cuyo objetivo es recrear la superficie terrestre. Como se aprecia en la Figura 1 mediante el uso de difuminación y brochas circulares en el editor gráfico 2D es posible crear elevaciones. Aparentemente la imagen tiene semejanza con una nube pero el motor gráfico transforma la imagen en terreno como se muestra en la Figura 2.

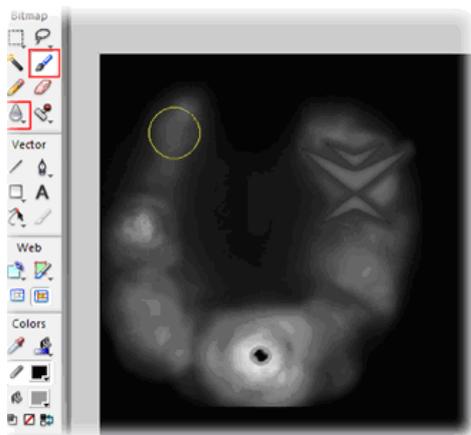


Fig. 1. Heihg map.



Fig. 2. Heigh map implementado

- **Rigging:** Se refiere a la creación y colocación de huesos previo a la animación de los personajes. Para empezar el proceso de animación, se debe tener finalizado el modelo a ser animado.

2.2 Herramientas de Diseño y Desarrollo:

- **XNA:** Es una herramienta y un lenguaje de programación que permiten generar software conectado a .NET para Microsoft Windows [3]. Su sintaxis es similar a la de C++, y su entorno de desarrollo integrado tiene gran flexibilidad y capacidad para dar soluciones en una gran variedad de plataformas y dispositivos.
- **PhotoShop:** Es un programa especializado en la edición de imágenes a nivel profesional, el cual es muy usado en aplicaciones 3D.
- **Fireworks:** Es un programa que esta orientado a la edición de imágenes a un nivel intermedio el cual tiene la mayoría de funciones que se necesitan para esta aplicación en especial.
- **3D Max:** Es la herramienta más popular para el diseño 3D. Sus ventajas son la excelente operabilidad, facilidad de uso, extensa documentación en el mercado, así como la integración con animación, programación, y facilidad de acoplamiento con herramientas externas, tal es el caso de XNA.
- **DirectX SDK** (DirectX Software Development Kit) [4]: Es un conjunto de herramientas de desarrollo, necesario para un juego y para crear aplicaciones basadas en DirectX en VisualBasic.NET, C / C ++, C #.

3. DISEÑO E IMPLEMENTACIÓN

Esta sección ha sido dividida en tres fases principales: fase de preproducción, producción y postproducción.

3.1 Fase de Pre-producción:

Dentro de la fase de preproducción se estructuraron las ideas generales del juego, así como su temática. En el caso de Evolution la fase de preproducción se dividió a su vez en cinco etapas que son: *i*) La idea del Juego: En esta etapa se planteó el tema y el concepto que definen las bases y la modalidad del juego; *ii*) La Historia: En este punto se describieron los objetivos, el rol de los personajes y los niveles en los que se desarrolló el juego. En Evolution el objetivo principal es la supervivencia de una civilización la cual se enfrenta a las amenazas producidas por la naturaleza. El rol de cada habitante es aprender a defenderse de las adversidades que se le presenten, buscar su comida y mantenerse a salvo de cualquier alteración del medio en el que se desenvuelve y así pasar al siguiente nivel del juego; *iii*) Tipo de Juego: El videojuego fue desarrollado en su totalidad en tres dimensiones, teniendo como base el tipo RTS (Real Time Strategy) [5]. El proceso comenzó con el manejo y familiarización de los motores gráficos XNA y TORQUE. Evolution fue integrado en ambos motores gráficos, con el objetivo de analizar cuál ofrece mayores ventajas para crear aplicaciones 3D de calidad y que puedan competir con otros productos existentes en el mercado; *iv*) Sistema de Visualización: Esta parte describió la forma en que el jugador visualiza los escenarios y la ejecución de sus acciones. *v*) Complementos del Juego: Evolution acopló en su modo de juego el control y administración de los eventos naturales; mientras que la civilización de individuos subsiste por sus propios medios sin que el jugador tome el control sobre ellos.

3.2 Fase de Producción:

En esta fase, se emprendió con el diseño y desarrollo del juego. Aquí se plasmaron las ideas que fueron expuestas en la fase de Pre-producción (ver sub-sección 3.1). En el caso de Evolution se tomó en consideración tres sub-etapas que son: Diseño de Arte, Diseño de Mecánicas y Diseño de Producción, que serán expuestas a continuación:

3.2.1 Diseño de Arte

En esta etapa se diseñó la ambientación del juego, para lo cual se usaron elementos como: Skybox, Terreno o Height map, Agua y Partículas. Estos fueron los componentes: -Guión: Permitió relatar la historia del juego en base a bosquejos que ilustraron de manera conceptual la funcionalidad del juego, así como sus caracteres, escenarios e interfases de usuario. Para la creación de los huesos de los personajes de Evolution se utilizó la técnica de rigging con la cual se crea el esqueleto que posteriormente se lo utiliza para dar movilidad al personaje, para lo cual la imagen debe estar previamente texturizada.

En cuanto a la animación se emplearon dos formas para animar un personaje. La primera, fue crear los huesos uno a la vez y conectarlos de acuerdo a su jerarquía, permitiendo crear libremente cualquier estructura. La segunda, fue mediante el uso de un Bípedo (Biped), que es una estructura de huesos similar a las de un ser humano donde se define el número de huesos para luego generar automáticamente su estructura. En cuanto al Sonido, dentro de Evolution los sonidos difieren según el escenario. En cuanto la interfaz, aquí se especificó la lógica y secuencia de menús. Evolution cuenta con menús en 2D para el menú principal y para las opciones de configuración. También cuenta con menús en 3D, destinados a la jugabilidad y otros para tutoriales. En cuanto a los Gráficos, dentro de esta categoría se establecieron los parámetros tanto en modelos de 2D y 3D, los cuales parten de un modelo conceptual previamente establecido. En Evolution todos los elementos tridimensionales están optimizados para un mayor desempeño dentro de la aplicación, ya que fueron estructurados con la menor cantidad de polígonos sin que esto afecte la calidad del diseño (ver Figura 3).

Para integrar todos los caracteres al escenario se seleccionó en el plano cartesiano un punto central, donde después se adjunta la imagen, con la diferencia de que estos puntos no se mueven a lo largo de una curva. Posteriormente se utilizó el concepto de aureola de acción, que son los puntos que rodean al punto central de un personaje (ver Figura 4). La aureola permitió darle la posibilidad al personaje de que se relacione con el escenario (stage) y con otros personajes (enemigos). Para realizar esto se utilizan sentencias como if, else, and, or, etc., y el concepto de distancia entre dos puntos. La distancia entre dos puntos $a=(x_0, y_0, z_0)$ y $b=(x_1, y_1, z_1)$, esta dada por la ecuación (1):

$$d(a,b)=\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2} \quad (1)$$

Con esto se definió los eventos como la muerte del un habitante de Evolution, la captura de un animal por parte de un personaje como alimento y la captura de un habitante por parte de su enemigo para lo cual se utilizó sentencias condicionales en el código del juego.

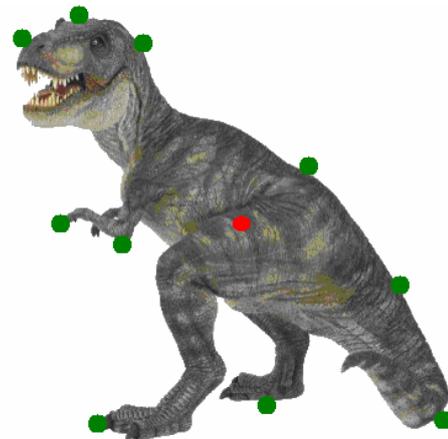
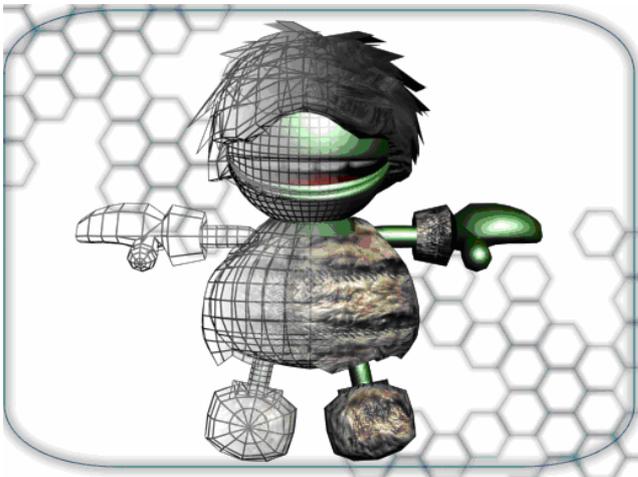


Fig. 3. Modelo o Personaje.

Fig. 4. Personaje con aureola de acción

3.2.2 Diseño de Mecánicas:

Describe el comportamiento de todos los entes que conforman el juego, así como las reglas que lo rigen. Para Evolution fueron divididas de la siguiente manera: Reglas del Jugador: Fueron destinadas al control de la naturaleza, el jugador era el encargado de la supervivencia de la civilización, por lo cual si se abusa de esta característica se puede causar la muerte de todos los habitantes y por ende la pérdida del juego; Comportamientos de los Personajes, que consiste en que cada habitante actúa de manera libre e independiente interactuando con su entorno y tratando de sobrevivir a las condiciones que se le presenten disponiendo de los recursos según sus necesidades.

Cuando comienza el juego, cada uno de los habitantes se encuentra en posición caminando (un estado), su estado se modificará de acuerdo a la acción que realice el jugador. Por ejemplo si es necesario que la tribu crezca en población, el habitante llevará a otra habitante a la cueva para que nazca un nuevo habitante en la tribu, así el nuevo estado será “llevar-habitante”, y en él se tendrá una serie de posibilidades nuevas como realizar una acción “caminar-derecha”.

3.2.3 Diseño de Programación:

En esta categoría fue necesario aplicar la metodología de desarrollo Scrum que permitió cumplir con los resultados esperados. Evolution aplicó esta metodología debido a que cumple con las siguientes fases de desarrollo: Etapa de Desarrollo, que contempló la unión de todo el diseño por medio de programación. Evolution fue desarrollado en XNA Game Studio 2.0 que es un framework para la programación de Juegos; Análisis y Pruebas, dónde se corrigieron los errores de las diferentes versiones del juego, y al mismo tiempo, se dio énfasis al refinamiento

de la jugabilidad; siendo esta la característica fundamental para brindar calidad en este tipo de aplicaciones.

3.3 Fase de Post-Producción:

Finalmente en la fase de postproducción ya se dispone de una versión final del juego, la cual fue instalada en la consola Xbox 360 para su ejecución.

4. EVALUACIÓN DE RESULTADOS

Este proyecto consistió en la integración de varias técnicas de video-juegos de tercera generación. Debido a que este tipo de proyecto es fundamentalmente gráfico los resultados obtenidos pueden ser tangibles únicamente cuando el usuario tiene contacto visual y acceso a la aplicación, sin embargo se detalla a continuación los resultados obtenidos en base a las diferentes tareas realizadas en los 4 sprints definidos al utilizar la metodología Scrum [6]:



Fig. 5. Jugabilidad.

4.1 Sprint 1

En el Sprint 1 se definieron los parámetros iniciales del juego como story boards y prototipos de diseño del juego. El objetivo de esta etapa fue probar todas las herramientas utilizadas hasta familiarizarse con ellas. Los resultados al término del Sprint 1 fueron: *i)* Aplicación del concepto de la Jugabilidad en “Evolution”; *ii)* implementación de motor gráfico dentro de XNA; *iii)* carga de escenarios y personajes; *iv)* ambientación del juego; *v)* aplicación de conceptos de Skybox; *vi)* diseño del agua y terreno; *vii)* animación de personajes, y *viii)* aplicación de la técnica del rigging.

La Figura 5, muestra la jugabilidad en tiempo real, donde es posible identificar varios componentes importantes como: personajes, escenarios, la interfaz de usuario en la cual se encuentran los controles principales y las estadísticas del juego.

El uso de motores gráficos de tercera generación requiere de mecanismos y de procesos complejos, pero sus ventajas son notoriamente visibles como se muestra en la Figura 6, en donde fue posible recrear el escenario principal del juego mediante el uso de técnicas implementadas en juegos actuales mejorando la calidad y apariencia de la aplicación.



Fig. 6. Gráficos y técnicas avanzadas.

4.2. Sprint 2

En el Sprint 2 se obtuvo como resultado la correcta implementación del motor gráfico, el manejo y control de la iluminación, la aplicación de efectos visuales y finalmente el comportamiento de los elementos visuales dentro de la aplicación. Los resultados obtenidos fueron: *i*) Desarrollo y creación de menú del juego; *ii*) integración de música y efectos de sonido, *iii*) manejo de partículas, luces y cámaras; *iv*) integración de Shaders, y *v*) manejo de detección de colisiones entre caracteres.

Por otro lado la Figura 7 muestra la evolución del proyecto en cuanto a gráficos se refiere. Ya que la tecnología cambia constantemente con el transcurso del tiempo, se ha creado un Tour virtual en la isla, implementando tecnología de tercera generación utilizada en consolas actuales de videojuegos.



Fig. 7. Tour virtual.

4.3. Sprint 3

En este sprint se obtuvo como resultado la implementación de los diseños realizados en el sprint 1 en el motor gráfico, con lo cual se aplicaron conceptos como Inteligencia Artificial en personajes, integración de niveles y carga de menús en el video-juego. Los resultados fueron: *i)* Implantación de modelos matemáticos; *ii)* Inteligencia Artificial de personajes; *iii)* comportamientos esperados; *iv)* integración de niveles, y *v)* manejo de menús. La Figura 8, muestra el menú principal del juego, en el cual es posible escoger el nivel y la dificultad. Dentro de esta imagen es posible notar aspectos gráficos y de funcionalidad que fueron implementados de acuerdo al Story Board creado en la fase inicial o Sprint1.



Fig. 8. Menú Principal.

4.4. Sprint 4

Finalmente se tiene la estabilización y pruebas del juego y la exportación a la consola Xbox 360.

5. TRABAJOS RELACIONADOS

La industria mundial de video-juegos se ha expandido en todo el mundo, por lo que existen varias investigaciones relacionadas. Por ello, en este análisis, sólo se ha incluido algunos trabajos similares. Así, en el trabajo propuesto en [7] se presenta el diseño y desarrollo de un simulador de autos de carrera en 3D desarrollado en Microsoft Visual C++ 6.0 utilizando el SDK y DirectX. En este mismo contexto el trabajo presentado en [8] se plantea un método de diseño de un sistema inmersivo didáctico, que utilizando técnicas de realidad virtual recrea las evoluciones de un vehículo submarino. Finalmente, en el trabajo presentado por [9] se presenta una plataforma de software libre para el desarrollo de aplicaciones geográficas 3D multicapa. Comparados con nuestro trabajo, en ninguno se especifica el proceso de diseño e implementación de un video juego educativo con herramientas innovadoras.

6. CONCLUSIONES Y TRABAJO FUTURO

En esta investigación se ha desarrollado un videojuego educativo denominado Evolution. Su desarrollo requirió de un análisis de los aspectos fundamentales sobre la teoría de juegos, algoritmos, fórmulas matemáticas, herramientas como Framework XNA y los patrones de comportamiento relacionados con la inteligencia artificial a ser aplicada para la construcción del video-juego. Como parte del desarrollo de Evolution se contemplaron todos los pasos relacionados con el diseño, la creación de bocetos gráficos y guiones para posteriormente realizar el modelado, diseño de personajes y escenarios que comprenden el juego, los cuales ayudaron a complementar el enfoque o visión del juego. Se implementó la teoría de juegos como un referente para definir estrategias y objetivos del video-juego que conjuntamente con fórmulas matemáticas y físicas permitieron formar el núcleo principal del video-juego, dándole una mayor productividad a la aplicación. Dentro del proyecto, fueron implementados conceptos como skybox, ambientación, biped, rigging, transformaciones como rotación, escalado, traslación, entre otras que ayudaron a dinamizar el comportamiento del video-juego de tal manera que luzca interesante y realista. De igual manera la inteligencia artificial jugó un papel muy importante dentro del desarrollo del videojuego ya que le otorgó un mayor realismo y a su vez permitió evitar un comportamiento monótono en cuanto a la interacción de los personajes. Finalmente, la leyenda creada para el video-juego conduce al jugador al aprendizaje de formas de cuidar y preservar el medio en el que se desenvuelve, generando una historia animada y a la vez educativa con los personajes y escenarios necesarios para representarla.

Como trabajo futuro se contempla que de este proyecto se deriven aplicaciones y proyectos con enfoques a la ciencia, turismo o entretenimiento.

Referencias

- [1.] G. Piñeiros, “Videojuegos ahora generan más dinero que las taquillas de la industria del cine en E.U”, Disponible en <http://www.portafolio.com>, Informe Técnico. Mayo 2008
- [2.] F. Truta, “Torque Game Engine Advanced Adds Dual Graphics API”, Softpedia, March 2009.
- [3.] B. Nitschke, Professional XNA Game Programming: For Xbox 360 and Windows.
- [4.] DirectX SDK, home page: <http://msdn.microsoft.com/en-us/directx/aa937788.aspx>
- [5.] H. Zwei, The Genesis of Real-Time Strategy, Publisher: Sega Enterprises Ltd. 1989.
- [6.] J. Palacio, Flexibilidad con Scrum, Principios de Diseño e Implementación de campos de Scrum. Octubre 2008. Disponible en: <http://www.proyectosagiles.org/que-es-scrum>.
- [7.] C. Espinosa, Simulador de Autos en 3D, Centro Universitario de Ciencias Exactas e Ingenierías, División de Electrónica y Computación, Junio 2004
- [8.] Otero, J. Flores, P. Saco, J. E. Arias, Diseño e implementación de una experiencia audiovisual colectiva inmersiva con fines educacionales. Departamento de Electrónica y Computación. Universidad de Santiago de Compostela, Marzo 2006
- [9.] M. Castrillón, et al. Capaware: Plataforma de desarrollo de software libre para aplicaciones geográficas 3D multicapa. III jornadas de SIG libre, Universidad de Girona, 2006.