



CAPITULO 6

DESARROLLO DE UNA NTERFAZ BASADA EN REALIDAD VIRTUAL CON REPRESENTACION TRIDIMENSIONAL

“FACULTAD DIGITAL INTELIGENTE”

CAPITULO 6

6.1	DISPOSITIVOS.....	- 185 -
6.1.1	<i>Evolución de Web3d.....</i>	- 186 -
6.1.2	<i>Tecnologías Web3d.....</i>	- 187 -
6.1.3	<i>Tecnologías Basadas en Imágenes 2D Interactivas.....</i>	- 189 -
6.1.4	<i>Ventajas y Desventajas de las Nuevas Tecnologías.....</i>	- 189 -
6.1.5	<i>Simulación con X3D.....</i>	- 190 -
6.1.6	<i>Componentes, Niveles y Perfiles.....</i>	- 191 -
6.1.7	<i>Codificación.....</i>	- 194 -
6.2	APLICACIONES.....	- 196 -
6.3	CONSTRUCCION Y PUBLICACION DEL ESCENARIO VIRTUAL.....	- 198 -
6.3.1	<i>Construcción.....</i>	- 200 -

En este capitulo ahondaremos mas en el tema de Escenarios Virtuales publicados en la Web, analizaremos un poco de sus inicios y sus aplicaciones practicas hoy en día.

Con un ejemplo de desarrollo analizaremos sin demasiados detalles como construir nuestros propios mundos virtuales, lo que nos ayudara a entender mejor los aspectos a tener en cuenta antes de aventurarnos en la creación de una de estas Interfaces.

6.1 DISPOSITIVOS

El término Web3D hace referencia a cualquier lenguaje de programación, protocolo, formato de archivo o tecnología que pueda ser usado para la creación y presentación de universos tridimensionales interactivos a través de Internet [4,5]. Dentro de estos lenguajes para programar universos virtuales se incluyen como estándares abiertos: VRML (Virtual Reality Modeling Language), Java3D y X3D (Extensible 3D). También existen y se están desarrollando un gran número de soluciones a nivel propietario para satisfacer las necesidades concretas de los clientes, generalmente, encaminadas al comercio y entretenimiento electrónico: Cult 3D, Pulse 3D, ViewPoint, etc. [LIB090]

Desde 1997 el estándar gráfico VRML, no ha conocido ninguna variación. Este estándar utilizado en Internet en incontables sitios permitió en su momento la interactividad del usuario con mundos 3D de una forma simple a través de sus opciones de navegación o una interacción más complicada por medio de programación. VRML no ha quedado impulsado por ninguna firma comercial excepto un curioso club de usuarios produciéndose un estancamiento del mismo. Ante esta situación de escasa evaluación del estándar, diferentes firmas han realizado sus propios desarrollos de software gráfico para Internet denominado genéricamente como gráficos WEB3D. Estas compañías han intentado rellenar el vacío dejado por el estándar o en su defecto se han inclinado por utilizar el API de JAVA 3D e intentando mejorar las páginas destinadas a el comercio o venta electrónica y de entretenimiento. A la espera de nuevos estándares propietarios o el más reciente estándar abierto el X3D, todas estas nuevas tecnologías ofrecen soluciones muy atractivas para la obtención de gráficos interactivos para Internet.

Una de las principales diferencias entre VRML/X3D y Java3D, a nivel conceptual, es que Java3D se define como un lenguaje de programación de escenarios 3D a bajo nivel. La creación de objetos y elementos tridimensionales en Java3D requiere no sólo la formación de los elementos 3D, sino también la definición de todos los aspectos relacionados con la visualización y control de las

capacidades del escenario. Para la creación del escenario más simple, el código en Java3D es notoriamente superior al necesario en VRML/X3D, pero por otro lado el control de los distintos elementos presentes en el sistema es superior y más natural en Java3D.

Otro aspecto destacable es la pérdida de velocidad y prestaciones en el caso de Java3D frente a otros visores de VRML/X3D desarrollados en C/C++ y empleando directamente Direct 3D u OpenGL. También es posible emplear Java3D como visor de archivos VRML/X3D. Para ello sólo es necesario utilizar alguno de los cargadores de VRML/X3D desarrollados para Java3D. Las ventajas principales son la capacidad de ejecución en distintas plataformas y evitar la necesidad de instalar un plug-in específico para el navegador.

6.1.1 Evolución de Web3d

Desde sus inicios la WEB3D es relativamente joven a comparación de otras tecnologías muy usadas en la actualidad, sin embargo nadie descarta que estas interfaces son el futuro que nos espera al momento de interactuar con las maquinas debido a que son mas intuitivas que las metáforas actuales.

Aunque la idea de publicar mundos virtuales en la Web va de la mano con el nacimiento del Internet y la Realidad Virtual, podríamos considerar su nacimiento en el año de 1994 con el prototipo realizado por Mark Pesce, Tony Parisi y Peter Ensnared y la creación en el mismo año del lenguaje VRML. A continuación mostramos un cronograma que muestra la evolución de WEB3D desde sus orígenes hasta el año 2004 donde se estandariza la tercera versión de X3D. La *figura 6.1.*, ilustra la historia de los lenguajes de modelado de la realidad virtual.

Aunque existen varios otros estándares propietarios hoy en día el X3D es considerado el lenguaje de desarrollo oficial de una WEB3D.

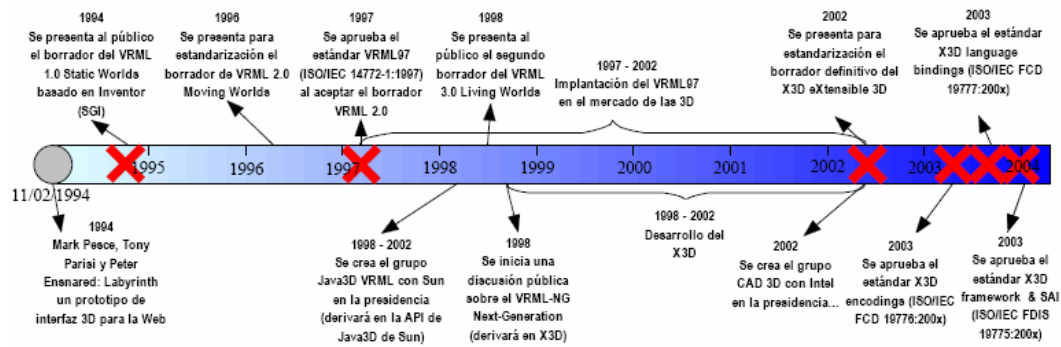


Figura 6.1. Cronograma del WEB3D

6.1.2 Tecnologías Web3d

Se pueden diferenciar dos grandes corrientes: gráficos WEB3D de aplicación general y los orientados a sistemas CAD CAM. En la **tabla 6.1.**, se destacan los lenguajes de modelado de realidad virtual. Estas últimas herramientas están orientadas a la ingeniería simultánea permitiendo ver modelos, editarlos, obtener medidas físicas, incluir anotaciones desde diferentes ubicaciones. [LIB091]

Tecnología	Arquitectura	Plataforma	Compresión	¿Plug-in?
<i>Aplicación General</i>				
3D Anywhere	Java	Win Unix MacOS		No
Anfy 3D	Java	Win Unix MacOS		No
Auran Jet	C**/Open GL	Win/MacOS		Si
Blaxxun 3D	X3D	Win/MacOS	Si	Si
Cortona	VRML modificado	Win/MacOS	Si	Si
Cult 3D	Propia	Win SunOS MacOS Linux	Si	Si
Enbaya	Propia	Win	Si	Si
EON Reality	Propia	Win		Si
Flatland	Propia	Win		Si
Hotmedia	Java	Win	Si	Si
Hyperscom	Propia	Win/MacOS/Linux		Si
Janet3D	Java3D renderer	Win/MacOS/Linux		No
Java3D	Java API	Win/SunOS		No
Kaon	Propia	Win/MacOS	Si	Si
Pulse 3D	Propia	Win/MacOS	Si	Si
Shout3D	Java	Win Unix MacOS		No
Sumea	Java	Win Unix MacOS		No

Superscape/Viscape	Propia	Win		Si
Viewpoint	Propia + XML	Win MacOS	Si	Si
Virtools	Propia	Win	Si	Si
Virtue3D	Propia	Win Mac	Si	Si
VRML97	ISO Standar	Win MacOS	Si	Si
Web Dimension	VRML	Win MacOS	Si	Si
WildTangent	Propia	Win		Si
X3D	VRML/XML	Todas	Si	Si
<i>Orientados a CAD / CAM</i>				
Adaptive media Vuent	Propia	Windows	Si	Si
Concepstation	Propia	Windows	Si	Si
Evis	Propia	Windows	Si	Si
Inmersive Design	Propia	Windows	Si	Si
Spinfire	Propia	Windows	Si	Si
Volo view/Streamline	Propia	Windows	Si	Si

Tabla 6.1. Lenguajes de Modelado de Realidad Virtual

Existen otros desarrollos no tan impactantes a primera vista como los anteriores. Normalmente han sido fruto de un proyecto universitario de interés. Alguno de ellos utiliza su propio lenguaje y ninguno de ellos está apoyado por empresas de software. En la **tabla 6.2.**, se muestran las tecnologías más conocidas.

Tecnología	Descripción
Alicex	Requiere plugin. Lenguaje de scripts
Interzat	Requiere plugin
b3b	Herramientas para incluir animaciones de 3D Studio Max y Maya dentro del web. Requiere plugin
Mendelbox	Requiere plugin. Animación de avatares
Spazz3D	Salida VRML o formato propio. Requiere plugin
Shell interactive	Basado en Macromedia Shockwave
Atomic3D	Compresión de audio y animaciones para anchos de banda pequeños en Internet. Requiere plugin
Visviva	Lenguaje de scripts y otras herramientas. Requiere plugin
3Donthe.net	Basados en cult3D
LigthSpace3D	Java3D. Solo Windows
Tabla 6.2. Proyectos de Lenguajes Modelado de Realidad Virtual	

6.1.3 Tecnologías Basadas en Imágenes 2D Interactivas

A veces resulta suficiente trabajar con imágenes 2D panorámicas. Utilizando técnicas fotográficas es posible obtener fotos que cubren 360° de tal forma que el usuario se encuentra en el centro de una esfera y puede navegar dentro de ella. Pueden incluir además hipervínculos. Las tecnologías más utilizadas son ilustradas en la *tabla 6.3*.

Tecnología	Notas	Plug in
IPIX	360x360 imagen panorámica. Transmisión progresiva	Si
livepicture	Gran oferta de herramientas para crear, examinar, realizar zoom, etc. con imágenes de diferentes resoluciones, desplazamiento por las diferentes imágenes progresivas	Si
webglide	Compresión de vídeo	Si
imove	Vídeo esférico	Si
be here	Vídeo comprimido 360° necesita las últimas versiones de Realplayer 7.0, Windows Media placer 6.4 o Quicktime 5.0	Si
Microsoft	Fotos panorámicas	Si
Hotmedia IBM	Permite la integración de todo desde sonido, imagines, panoramas IPIX, videos y otras combinaciones interactivas. Muy lento	Si
<i>Tabla 6.3.</i> Tecnologías Basadas en Imágenes 2D Interactivas		

6.1.4 Ventajas y Desventajas de las Nuevas Tecnologías

Las nuevas tecnologías ofrecen las siguientes ventajas:

- Son relativamente fáciles de crear y de programar
- Normalmente requieren un plugin de pequeño tamaño para su visualización aunque alguna tecnología no requiere ninguno.
- Válidos para diferentes plataformas (PC, Mac, Unix etc.)
- Calidad de imagen sorprendente
- Efectos de sombreado (reflexiones, anti-aliasing, sombras, etc.)
- Facilidad par su integración en páginas Web.
- Basados en estándares.

- Normalmente utilizan sistemas de compresión para que la descarga por el usuario resulte breve.
- Importantes empresas detrás de estos desarrollos lo que garantiza un desarrollo continuado de estas Tecnologías.

En contra de estos desarrollos cabe destacar:

- El software es propietario por lo que el desarrollador depende de las intenciones de la firma comercial.
- Normalmente se requiere plugin.
- La implementación de estos desarrollos se encuentra orientada hacia los principales clientes de los mismos normalmente comercio y entretenimiento electrónico.

6.1.5 Simulación con X3D

X3D (extensible 3D) es la próxima-generación del estándar abierto para la Web. Es el resultado de varios años de desarrollo por parte del Grupo de Trabajo X3D del Consorcio Web 3D y el reciente Grupo de Trabajo Browser. Este último ha trabajado de cerca con el Grupo de Trabajo X3D para crear una nueva Especificación X3D que reúna las necesidades de los Browser de las compañías y las necesidades de toda la comunidad. Esos requerimientos son:

- Compatibilidad con el existente contenido VRML, browsers, y herramientas.
- Mecanismo de extensión que permita introducir nuevas características, vista rápida de avances, y adopción formal de esas extensiones dentro de la especificación.
- Un perfil del "núcleo" pequeño y simple para la más amplia adopción posible del soporte X3D, importando y exportando.
- Perfil completo VRML (full) para soportar contenidos existentes.
- Soporte para otras codificaciones incluida XML para una firme integración con las tecnologías y herramientas WEB.

- Arquitectura y proceso de avance para la especificación y tecnología.

Estos requisitos fueron logrados introduciendo una arquitectura basada en componentes para apoyar extensiones, incompatibilidades, y errores, y codificaciones como problemas separados. Un componente representa una agrupación de rasgos relacionados, tales como una colección de nodos relacionados, una extensión al modelo de evento, o un nuevo soporte de script.

En lugar de una especificación enorme que requiera adopción completa para complacer, una arquitectura basada en componentes que apoya la creación de perfiles diferentes que pueden ser soportados individualmente. Estos perfiles son colecciones de componentes, y dos ejemplos de perfiles son el pequeño núcleo para soportar una simple animación no-interactiva, y el perfil base VRML-compatible para soportar mundos totalmente-interactivos. Pueden extenderse componentes individualmente o pueden modificarse a través de agregar niveles, o pueden agregarse nuevos componentes para introducir nuevas características, como streaming. A través de este mecanismo, los avances de la especificación pueden moverse rápidamente porque el desarrollo en un área no retarda la especificación en conjunto. [www030]

6.1.6 Componentes, Niveles y Perfiles

X3D para ser de verdad un estándar industrial amplio, los diseñadores comprendieron que las diferentes compañías no necesitan o querían soportar cada rasgo que X3D puede ofrecer. Por ejemplo, si una compañía quiere hacer una pequeña, eficiente animación 3D, podría no estar interesada en dar características de soporte a la geología. Debido a esto, se encapsulan grupos de características en lo que se llama componentes. Un componente puede ser específico para un área particular de funcionalidad, es decir un Geo el componente por manejar datos geográficos, o un componente geometría que contiene un grupo de nodos de geometría, o un componente del scripting que introduce el concepto de soporte de script.

El bloque básico de construcción de una escena X3D es el nodo, como ya lo era en VRML. Sin embargo, frente a la arquitectura monolítica de VRML, los nodos en X3D se agrupan en componentes, cada uno de ellos relativo a una característica del lenguaje. Para cada componente, se establecen también un conjunto de niveles, de forma que cada nuevo nivel añade nuevos nodos y/o características de los mismos al anterior. La *tabla 6.4.*, muestra el ejemplo de Geometry2D

Nivel	Prerrequisitos	Nodos	Soportado
1	Core 1 Grouping 1 Shape 1 Rendering 1		
		Polyline2D, Polypoint2D, Rectangle2D, TriangleSet2D	Todos los campos soportados
2	Core 1 Grouping 1 Shape 1 Rendering 1		
		Geometry2D nodes, Arc2D, ArcClose2D, Circle2D, Disk2D	Todos los campos soportados

Tabla 6.4. Ejemplo de componente: Geometry2D

Un *perfil* es una agrupación de componentes que cubren varias áreas diferentes de funcionalidad por ejemplo, un Completo perfil que maneja todos los nodos VRML97 y las áreas funcionales. Un perfil puede contener la funcionalidad de varios perfiles incluso, el perfil Completo incluye la funcionalidad del pequeño perfil Núcleo.

Una vez que un grupo de perfiles se considera importante para la inclusión a través de muchas aplicaciones, una nueva versión de X3D puede crearse que incluirá por defecto un juego de los perfiles. Una nueva Versión implica más funcionalidad que la Versión del número anterior.

Las compañías pueden crear browsers X3D, herramientas, importadores, y exportadores que soporten Versiones diferentes y Perfiles. Por ejemplo, un Player

pequeño podría ser compatible con X3D-1. Un browser compatible VRML97 totalmente sería compatible con X3D-2. X3D-3 podrían incluir áreas extras de funcionalidad incluyendo NURBS y streaming, etc.

Descripción de los Perfiles: Cada uno de los perfiles pretende satisfacer las necesidades de un grupo particular de aplicaciones, como se muestra en la **figura 6.2.**, para las cuales se escoge un conjunto concreto de nodos.

- **Core:** El conjunto mas reducido
- **Interchange:** Para intercambio de geometría, apariencia y animaciones entre diferentes aplicaciones. Solo fuentes de luz direccionales. No incluye sonido.
- **Interactive:** Interacción básica entre el usuario y la escena, incluye los componentes *Pointing device sensor, Key device sensor y Event utilities.* Incluye diferentes tipos de luz, pero no sonido.
- **MPEG-4:** Para la utilización de X3D con este otro estándar.
- **Immersive:** Mayor funcionalidad en gráficos e interacción. Geometría 2D y 3D, texto, sonido y script.
- **Full:** Toda la funcionalidad de X3D. Incluye componentes como *Geospatial, Humanoid Animation, (H-anim), Non Uniform Racional B-Spline (NURBS) y Distributed Interactive Simulation (DIS).*

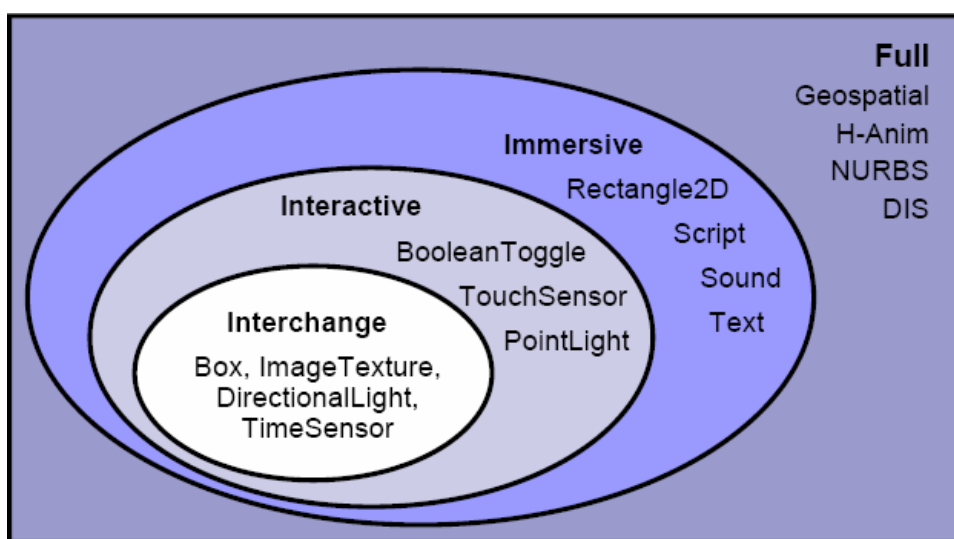


Figura 6.2. Perfiles

6.1.7 Codificación

XML se adoptó como una sintaxis para X3D para resolver varios problemas reales:

6.1.7.1 Rehospeabilidad.

La sintaxis VRML 97 es extraña a todos menos a la comunidad de VRML. Es similar a la sintaxis gráfica de escena de Open Inventor en la cual esta basada, y a algunas anotaciones de objeto. No obstante la sintaxis dominante de uso mundial es XML. El marcado ha demostrado ser la mejor solución a los problemas de los ciclos de vida largos de archivos de datos y rehosting.

6.1.7.2 Página de integración.

Las páginas XML basadas en integración van directamente al problema de mantener un sistema más simple para que más personas puedan desarrollarlas para las páginas web de tejido, en contenido e implementación.

6.1.7.3 Integración con la próxima-generación web.

Los miembros del Consorcio Web (W3C) están poniendo mucho esfuerzo en el desarrollo de XML.

El Grupo de Trabajo X3D se pasó varios meses mirando varias alternativas para representar los nodos VMRL97 en XML. Esta deliberación produjo un solo DTD que tuvo que ser acordado en el grupo. [www032]

A continuación se va a observar una comparación del código clásico como VRML y la codificación XML aplicada en el X3D. En la *figura 6.3.*, se muestra un ejemplo de código VRML clásico, mientras que en la *figura 6.4.*, se muestra un ejemplo de código XML

Ejemplo de classic VRML encoding:

```
#X3D V3.0 utf8
PROFILE "Full"
META "filename" "esfera.x3dv"

# Una esfera de color púrpura
Shape {
  appearance Appearance {
    material Material { diffuseColor .8 .0 .8 }
  }
  geometry Sphere { radius 2.3 }
}
```

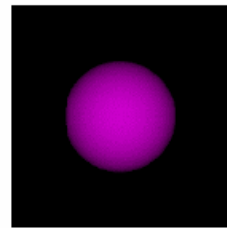


Figura 6.3. Codificación Clásica VRML.

Podemos Visualizar la diferencia marcada del uso de etiquetas con la codificación XML lo que permite que sea más extensible que utilizando la codificación clásica.

Ejemplo de XML encoding:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
  "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Full">
  <head>
    <meta name="filename" content="esfera.x3d"/>
  </head>
  <Scene>
    <!-- Una esfera de color púrpura -->
    <Shape>
      <Appearance>
        <Material diffuseColor=".8 .0 .8"/>
      </Appearance>
      <Sphere radius="2.3"/>
    </Shape>
  </Scene>
</X3D>
```



Figura 6.4. Codificación XML para el X3D.

Una vez que se ha creado un juego de etiquetas de XML entonces y dependiendo de como el browser usa XML, se puede incorporarlas a través de uno de los esquemas siguientes: [www033]

Stylesheet (hojas de estilos): El archivo XML es usado directamente. Internamente, tendrá una referencia a un archivo del lenguaje de stylesheet como XSL. El archivo de XSL tiene las instrucciones para desplegar las etiquetas XML.

Data islands (Islas de datos): Los valores de XML están dentro de otro archivo (ej., HTML) para nombrar las etiquetas como tablas HTML, divs, etc. Esto significa el XML será incluido por valor o referencia en el documento, y los valores dentro de las etiquetas son usados por etiquetas de HTML. Esta es una solución específica Microsoft.

Soporte por plugin: Esta es la forma donde una etiqueta objeto es usada en línea para indicar donde el código de soporte puede encontrarse, los parámetros para pasar, etc.

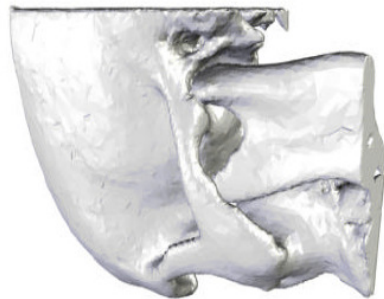
Soporte directo del objeto: El browser Web mantiene soporte nativo a las etiquetas aumentando su modelo de objeto para manejar a estos específicamente. Obviamente, a menos que el browser sea extensible por autores, las etiquetas y su aplicación serán codificadas en el browser.

6.2 APLICACIONES

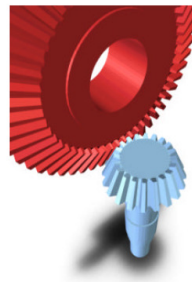
La WEB3D es aplicable en muchos campos, sin embargo en la actualidad predominan las interfaces 2D en la Web debido al paradigma creado por la metáfora del escritorio, por lo que tomara algún tiempo cambiar la forma en la que el usuario se comunica con la maquina pese a que las aplicaciones 3D ya son una realidad y son muy usadas en Juegos, Simulaciones, etc. Incluso una muestra de las bondades de la tercera dimensión se esta usando en interfaces de Escritorio como lo demuestra el ultimo sistema operativo publicado por Microsoft, Windows Vista que nos hace pensar que en un tiempo no muy lejano estaremos utilizando interfaces totalmente intuitivas en 3 dimensiones y las nuevas generaciones que nazcan con este nuevo paradigma lo explotaran al máximo en todos los campos.

A continuación veremos algunos ejemplos prácticos que se están usando en la actualidad de interfaces de Realidad Virtual en la Web:

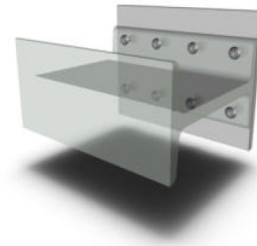
- **Proyecto WEBD:** El proyecto WEBD es un ejemplo de la potencia de este tipo de tecnologías para la enseñanza. Este proyecto está promovido por el Politécnico de Turín y la Universidad Politécnica de Madrid tiene encomendado la gestión del sitio WEB y la elaboración de los modelos tridimensionales interactivos necesarios para impartir nociones de ingeniería de diseño y biomecánica básica. A continuación se ofrecen algunos ejemplos de los entornos 3D realizados para el proyecto. Todos ellos son interactivos y pueden ser modificados por el usuario. (www.webd.etsii.upm.es)



Reconstrucciones de Huesos



Modelos Animados para el Diseño Mecánico



- **Second Life:** Es un mundo virtual en 3 dimensiones donde puedes interactuar con lo que te rodea tal como si lo hicieses en la realidad como comprar propiedades, este sitio esta enteramente construido y manejado por sus residentes. Su apertura al público se la realizo en el año 2003 y su crecimiento ha sido tal que hoy en día esta habitado por un total de 6'438.817 personas alrededor de todo el mundo. Vea la **figura 6.5.** (secondlife.com)
- **ManagerZone:** Es un sitio creado con la finalidad de convertir al usuario en el dueño y entrenador de su propio equipo deportivo, permitiéndole administrar los activos de su club, comprar jugadores del mercado mundial, contratar entrenadores para sus jugadores, crear semilleros, colocar tácticas de juego, etc. Con esto el usuario participa de varios torneos y competencias con otros

managers de todo el mundo simulando la realidad de un equipo deportivo. Lo interesante de este sitio es que con complejos algoritmos de cálculo se genera una simulación de un partido basándose en los atributos de cada equipo, que el manager estableció horas antes de un encuentro.



Figura 6.5. Second Life

Esta simulación es visible al usuario en tiempo real utilizando la capacidad de las tres dimensiones para darle el realismo necesario al juego que ayudara al manager evaluar las virtudes y defectos de su equipo para que logre hacer los correctivos necesarios para un mejor rendimiento de su equipo en el futuro. Actualmente dispone de dos deportes populares como son el Fútbol y el Hockey sobre hielo. (www.managerzone.com). Vea la *figura 6.6*.

6.3 CONSTRUCCION Y PUBLICACION DEL ESCENARIO VIRTUAL

En este apartado vamos a mostrar a grosso modo los pasos que debemos considerar para construir un mundo virtual. Debemos tener en cuenta que en el mercado existen diversas herramientas que nos pueden ayudar a construirlos, lo que quiere decir que las que se presentan aquí no necesariamente son el mejor camino para

realizar una WEB3D, sin embargo las mostradas aquí son las mas conocidas y utilizadas por la mayoría de los diseñadores por los resultados finales que muestran los proyectos.



Figura 6.6. ManagerZone

Antes de empezar a desarrollar cualquier proyecto debemos identificar la envergadura y los objetivos del mismo, debido a que quizás solo necesitemos crear una habitación por lo que no deberíamos complicarnos demasiado y utilizar una sola herramienta que fácilmente nos puede mostrar un resultado aceptable o quizás queramos ahondar mas en el lenguaje que hayamos elegido en conocer mas a fondo, entonces buscaremos un editor del mercado o hacerlo simplemente desde su Bloc de Notas.

En el caso de este ejemplo nos interesa diseñar un edificio completo con el que podamos interactuar con los objetos contenidos en este, además de que nos interesa que el proyecto sea tan mantenible que si la estructura del edificio cambiase podamos modificarlo desde sus cimientos. Antes de empezar si deseamos utilizar la metodología de este ejemplo deberemos abordar nuestros conocimientos en los siguientes puntos:

- Autocad 2D y 3D.
- Diseño y Modelado en 3D Studio Máx.
- Iluminación y Texturas con Materiales.
- Flux Studio.
- XML y X3D

6.3.1 Construcción

Una vez elegido el mundo que queremos crear deberemos empezar por conseguir los planos estructurales del edificio que queremos representar o quizás convertirse en su propio arquitecto e ingeniero y diseñarlos por si solos. De cualquier manera deberemos plasmarlo en las dos dimensiones para ir levantando pared por pared.

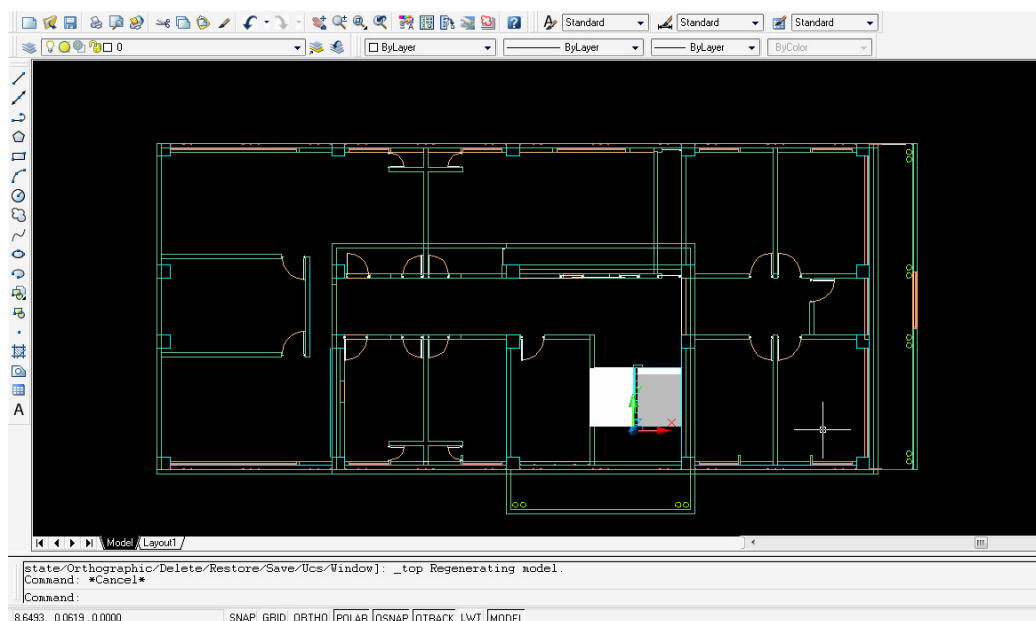


Figura 6.7. Plano 2D en Autocad.

En este caso se ha decidido utilizar una de las mejores herramientas arquitectónicas que existen en el Mercado como lo es Autodesk AutoCad. Como podemos observar en la *figura 6.7.*, crearemos una explanada que representara el piso y si es el caso la azotea o techo, luego una visión en 2D de los diferentes sectores de cada uno de los pisos que representaran luego nuestras oficinas,

laboratorios, aulas, etc. y los organizaremos en capas para poder modificarlos sin dañar los demás objetos de nuestro mapa.

Luego de terminado nuestro plano a escala levantaremos las columnas y paredes una por una extruyéndolas a lo alto y lo ancho, del modo ilustrado en la **figura 6.8.**, y lo mismo haremos para crear objetos como puertas, ventanas, gradas y todo lo que conformase nuestra estructura inicial.

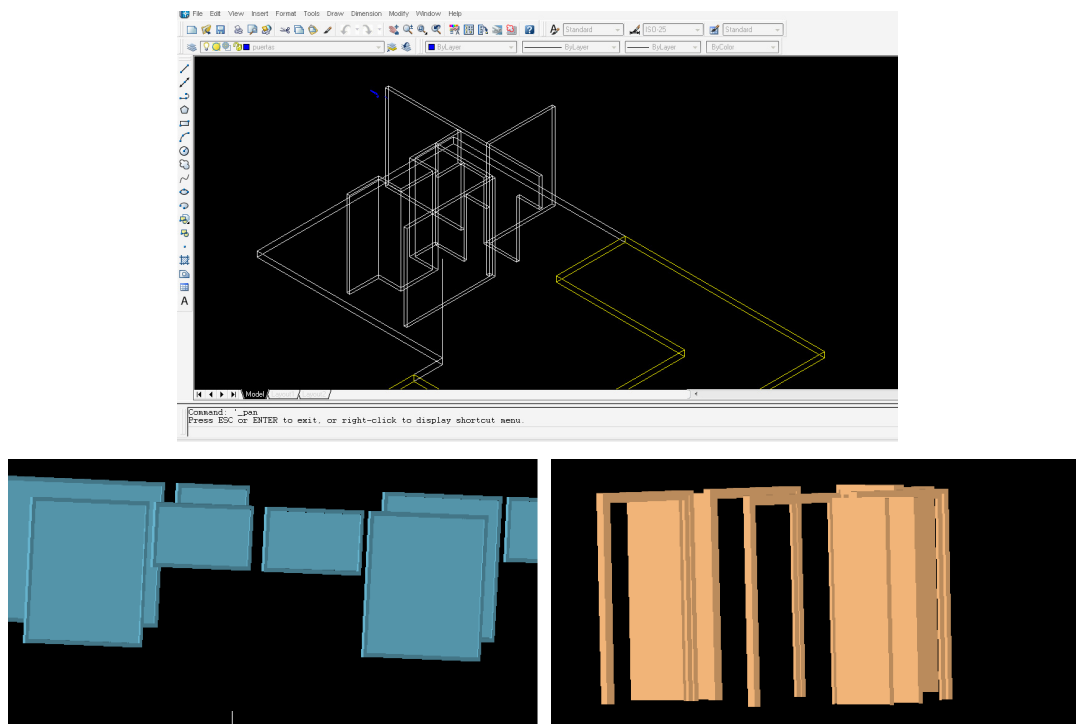


Figura 6.8. Levantamiento de paredes y creación de Objetos.

Una vez terminado todo este trabajo podremos apreciar el esqueleto de nuestro edificio (**figura 6.9.**) listo para ser exportado y empezar con la creación del realismo aplicándole texturas, iluminación y agregándole todos los objetos que compondrán nuestro pequeño mundo virtual.

Utilizaremos la herramienta 3D Studio Máx para modelar las habitaciones y crear la inmobiliaria que colocaremos luego dentro de nuestro edificio tal como se muestra en la **figura 6.10.**

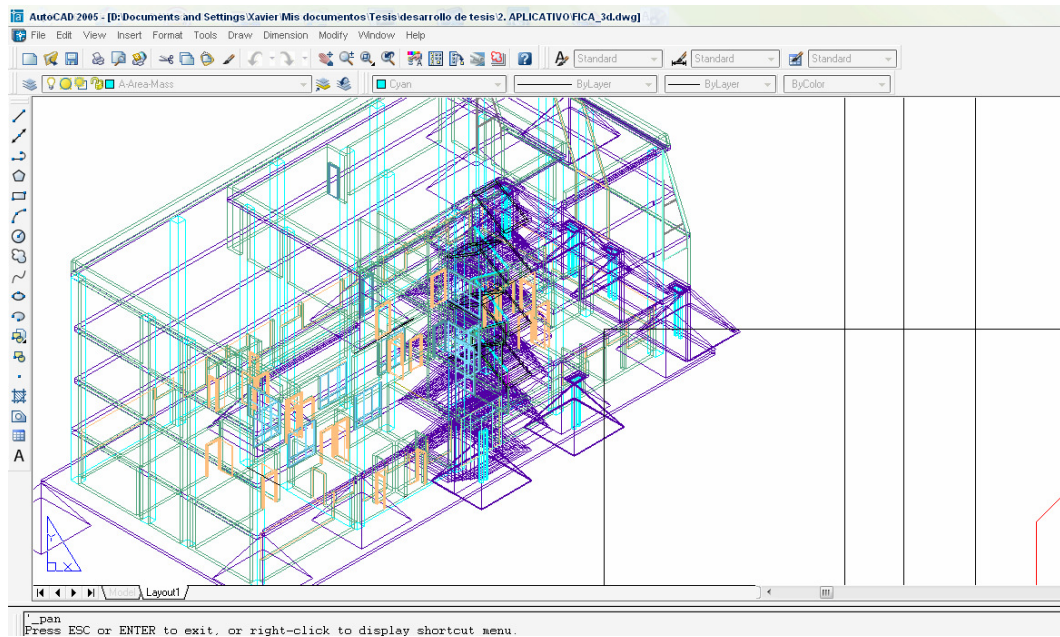


Figura 6.9. Estructura en tres dimensiones.

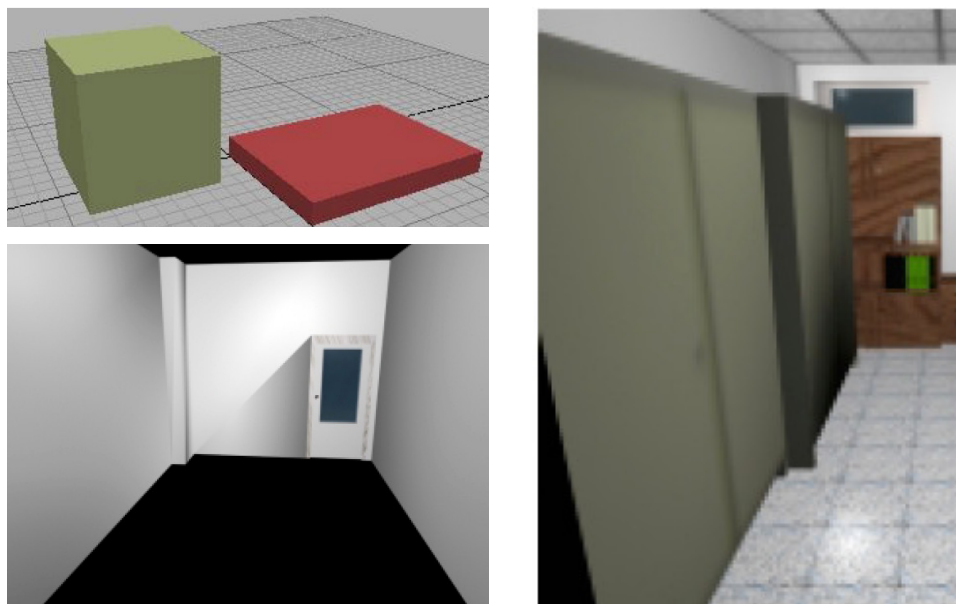


Figura 6.10. Modelado de Objetos en 3DS.

A partir de que hayamos terminado el arduo trabajo de completar y adornar nuestro estructura para acercarnos lo mas posible a realidad, habilidad que la iremos ganando con la practica y valiosos consejos de otros diseñadores con mas experiencia que nos ahorraran muchos disgustos procederemos a exportar nuestro proyecto a la especificación de Máx. que es el 3ds o una mas extensible como obj

para poder utilizarla en cualquier otra aplicación que nos ayude a generar nuestro mundo virtual.

Para este caso se eligió Flux Studio debido a que es el visor mas utilizado hoy en día en la Web para visualizar proyectos X3D, además de ser una herramienta muy completa desde donde añadiremos más realismo a nuestro proyecto y configuraremos la visita de nuestro usuario, como se puede observar en la **figura 6.11**.

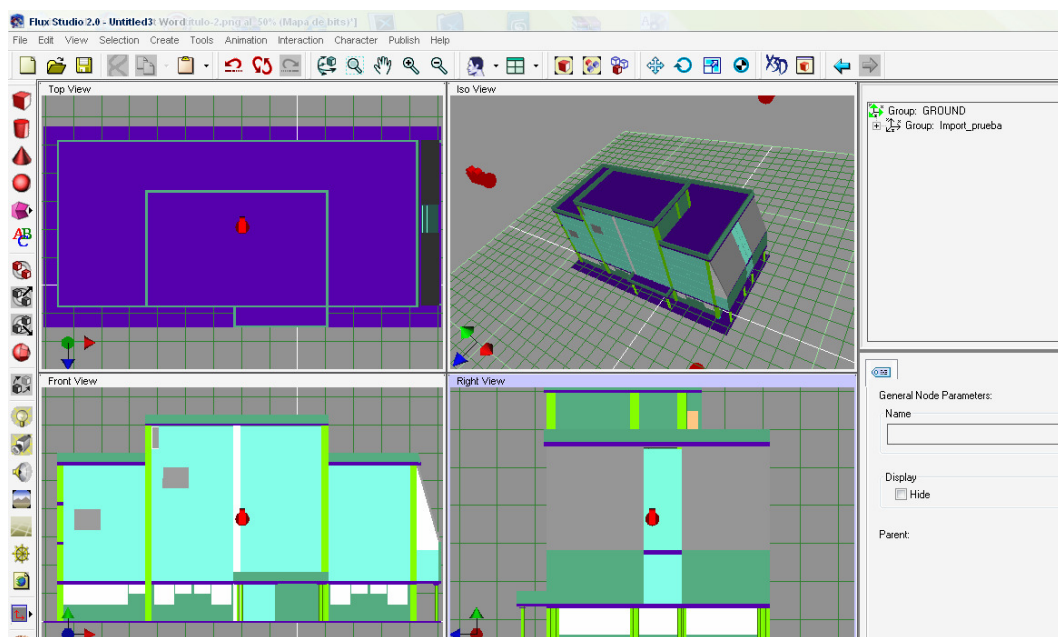


Figura 6.11. Generación del mundo virtual con Flux Studio

Luego de que hayamos considerado que nuestro mundo virtual esta completo y añadimos contenido dentro de este para que sea mas interactiva este tipo de interfaz procederemos a exportar nuestro proyecto procederemos a generar nuestro código X3D con la capacidad que esta herramienta nos brinda para exportar en este formato.

Y con esto quedara listo para poder visualizarlo en el navegador instalado un plug-in que nos permita visualizar archivos X3D, en nuestro caso se ha elegido el de la herramienta que utilizamos “Flux Placer”.