



**UNIVERSIDAD  
DE  
LEÓN**



**ESCUELA DE INGENIERÍAS  
INDUSTRIAL, INFORMÁTICA Y  
AERONÁUTICA**

**TRABAJO FIN DE CARRERA**

**ESTUDIO PARA LA REALIZACIÓN DE UN PASEO  
VIRTUAL EN EL EDIFICIO TECNOLÓGICO  
FASE-1**

**AUTOR: *Javier Lozano Mata***  
**DIRECTOR DEL PROYECTO: *Fernando Jorge Fraile Fernández***  
**FECHA: *Junio de 2007***

## **Agradecimientos**

*A mi coordinador de proyecto D. Fernando Jorge Fraile Fernández, al profesorado y alumnos de la universidad de León por su cooperación.*

*A mis familiares, compañeros y amigos que han hecho esto posible y que sin su inestimable ayuda la tarea habría sido mucho más ardua.*

El presente Trabajo de Fin de Carrera ha sido realizado por **D. Javier Lozano Mata**, alumno de la Escuela de Ingenierías Industrial, Informática y Aeronáutica de la Universidad de León, con el objetivo de obtener el título de Ingeniero Técnico Industrial en la especialidad de Estructuras e Instalaciones Industriales.

La tutoría de este trabajo ha sido llevada a cabo por **D. Fernando Jorge Fraile Fernández**, perteneciente al Departamento de Física, Química y Expresión Gráfica.

Vº Bº  
Tutor del trabajo

Vº Bº  
Oficina Técnica

Fdo: Fernando Jorge Fraile Fernández

Fdo: Rafael Rodríguez Álvarez

El Autor

Fdo: Javier Lozano Mata

León, Junio de 2007

# -ÍNDICE-

<b>PRÓLOGO .....</b>	<b>7</b>
<b>TEMA 1: CIMIENTOS DEL DISEÑO ASISTIDO POR ORDENADOR.....</b>	<b>9</b>
1. Definición y finalidad del sistema CAD .....	9
2. Diversidad de las diferentes técnicas de diseño.....	11
3. Tratamiento de imágenes .....	13
3.1. Imágenes Bidimensionales (2D).....	13
3.2. Imágenes Tridimensionales (3D).....	13
4. Evolución del proceso infográfico.....	14
5. Progreso evolutivo y fundamentos del Autocad 2006 R2 .....	16
6. Progreso evolutivo y fundamentos del 3D Studio MAX 6 SP1 .....	18
7. ¿Por qué se ha optado por estos soportes de software? .....	20
7.1. Aplicaciones, ventajas, recursos y resultados de que es objeto .....	20
8. Aspectos generales sobre el Hardware.....	23
8.1. Conceptos a tener en cuenta .....	23
8.2. Importancia de los diferentes componentes.....	27
8.3. Definición, tipos de APIs gráficas usadas.....	29
8.3.1. API OpenGL.....	30
8.3.2. API Direct3D (aplicación bajo directX) .....	33
9. Tratamiento de lo representado .....	35
<b>TEMA 2: LA IMAGEN. ....</b>	<b>38</b>
1. Introducción .....	38
2. Formatos gráficos .....	38
2.1. Modo de mapa de Bits .....	38
2.2. Gráficos Vectoriales .....	40
3. Vídeo Digital .....	42
4. Compresión de secuencias de video.....	43
<b>TEMA 3: TRABAJANDO CON AUTOCAD.....</b>	<b>46</b>

1. Soporte .....	46
2. Puntos a tener en cuenta .....	46
3. Modelado del recibidor de la planta baja .....	48
3.1. Huecos de ventanas y puertas .....	51
3.2. Cornisas y techos .....	53
3.3. Base inferior .....	56
3.4. Escaleras y cubrescaleras .....	57
3.5. Columna central .....	58
4. Modelado de la planta superior .....	60
4.1. Cornisa interior de la planta superior .....	62
4.2. Soportes superiores .....	63
4.3. Base del piso superior .....	64
4.4. Cornisa exterior y muro .....	65
4.5. Cubiertas .....	65
4.6. Ventanas superiores .....	66
4.7. Rodapiés .....	67
4.8. Modelado de elementos comunes .....	67
4.9. Exportar archivos .....	79
<b>TEMA 4: PROCESO DE TRABAJO CON 3D STUDIO MAX. ....</b>	<b>81</b>
1. Introducción .....	81
2. Proceso de importado de archivos .....	81
3. Edición y manejo de materiales para la escena .....	84
3.1. Consideraciones previas .....	85
3.2. Tipos de renderizadores .....	87
3.2.1. Render Vray 1.46.08 .....	89
3.3. Aplicación de los distintos materiales .....	94
3.3.1. Desde el inicio .....	94
3.3.2. Desde la biblioteca .....	97
3.3.3. Otros .....	98
3.4. Sólido Multimaterial .....	102
4. Creación de calcos en la escena .....	104
5. Imagen de entorno .....	106
6. Iluminación general .....	108
6.1. Iluminación ambiental .....	108
6.2. Iluminación solar .....	108

7. Cámaras.....	110
7.1. Generalidades.....	110
7.2. Creación de cámaras.....	111
7.3. Creación de los recorridos de cámara.....	112
7.4. Asignar el recorrido a la cámara.....	114
8. Tiempo y velocidad de animación.....	115
9. Preliminares.....	116
10. Representar la escena.....	116
<b>TEMA 5: CREACIÓN DE IMÁGENES EN 360 GRADOS .....</b>	<b>120</b>
1. Generalidades.....	120
2. ¿Qué se necesita para crear imágenes en 360°?.....	120
3. Creación de imágenes.....	121
4. Aplicación panorámica.....	123
4.1. Creación del panorama cilíndrico.....	123
4.2. Creación de la aplicación flash.....	124
<b>TEMA 6: FLASH .....</b>	<b>127</b>
1. Introducción.....	127
2. Qué es Flash.....	128
3. Que es un contenido dinámico y una película flash.....	129
4. Qué es un Script.....	129
5. Proceso.....	130
<b>CONCLUSIONES.....</b>	<b>138</b>
<b>ANEXO.....</b>	<b>140</b>
1. Anexo 01: Planta Baja ITI Fase – 1.....	141
2. Anexo 02: Planta Primera ITI Fase – 1.....	142
3. Anexo 03: Planta Segunda ITI Fase – 1.....	143
<b>BIBLIOGRAFÍA.....</b>	<b>144</b>

## PRÓLOGO

Las ventajas de diseñar en 3D son más que evidentes. Debido a que en realidad se construye virtualmente el objeto que se diseña, en lugar de dibujar diferentes vistas que representen la proyección del objeto sobre planos de proyección, el diseño resulta más preciso y adecuado. A partir de este modelo se pueden producir imágenes sombreadas y realistas (*rendering*) para mostrar el aspecto definitivo del diseño, además de producir dibujos a escala en 2D de multitud de vistas, para su posterior fabricación.

Lo que se quiere dar a conocer en este proyecto no es otra cosa que la representación y modelización de la manera más fehaciente posible de la entrada interior de la fase 1 de la escuela de Ingenierías situado en el Campus de Vegazana en León para su posterior conocimiento en base a una visita virtual, ya que de esta manera, el observador podrá conocer la estructura y la apariencia interactivamente.

Un buen ejemplo para su creación son las herramientas de realidad virtual en primera persona, que hacen uso de motores gráficos 3D y permiten un nivel de desarrollo bastante bueno; Crystal3D es una muestra de ello. No obstante y como contraposición a esto, diremos que los resultados esperados en el apartado gráfico no son muy realistas, por lo que he optado por utilizar panorámicas 3d estáticas y videos para simular el motor gráfico.

La abundancia de utilidades de diseño actuales para elegir es muy variada, pero en este caso se ha optado principalmente por dos aplicaciones ampliamente reconocidas, como son Autocad en su versión 2006 y 3D Studio Max 6 ambos programas desarrollados en sus versiones más actuales por la compañía Autodesk.

Como todos sabemos el desarrollo y modelado de una escena por muy simple que sea conlleva tiempo que dependerá del resultado que queramos obtener. El objetivo principal aquí, tiene que ver con la adecuación de la realidad al plano digital, es decir, que con el mero hecho de observar una escena, podamos casi "interactuar" con el ambiente; a fin de lo expresado anteriormente, se hará un estudio del trabajo seguido con estos dos utensilios de la manera más sencilla posible y sin que su transcurso narre de forma objetiva el compendio de que constan estos

programas, ya que lo que de verdad se quiere dar a conocer, es el método para obtener la representación de la Escuela de Ingenierías.

Dejando de lado la parte del modelado, se ha tendido en mayor medida a centrar el trabajo en la parte la representación virtual animada, ya que los materiales por sí solos no crean una imagen final completa, si no, que a poder ser, debemos ajustarlos para que funcionen como un todo conjunto en la escena.

La realidad es demasiado compleja e intervienen tantos factores que nuestra percepción de la composición tiende a “filtrar” las cosas, y sin duda, la idea aquí expuesta se basa en el “convencimiento” que evoque una respuesta emocional por parte de los espectadores a los que está destinado el trabajo.

La representación del interior del edificio de Industriales, da un ejemplo de lo que por medio del uso del espacio tridimensional se puede llegar a concebir, para dejar un camino abierto de forma rápida a nuevos puntos de vista en cuanto a su diseño y proceso de construcción.

# TEMA 1: CIMIENTOS DEL DISEÑO ASISTIDO POR ORDENADOR

## 1. DEFINICIÓN Y FINALIDAD DEL SISTEMA CAD

El diseño asistido por computador o CAD (*Computer Aided Design*), se basa en la aplicación de diferentes utilidades que hacen que el trabajo de los profesionales que desarrollan estos campos, sea más productiva y fácil. Además es la herramienta principal para la creación de entidades geométricas enmarcadas dentro de procesos de administración del ciclo de vida de productos (*Product Lifecycle Management*), y que involucra software y algunas veces hardware especiales.

“El diseño asistido por computador o CAD (*Computer Aided Design*), es el uso de un amplio rango de herramientas computacionales que asisten a ingenieros, arquitectos y a otros profesionales del diseño en sus respectivas actividades”<sup>1</sup>.

Los usos de estas herramientas varían desde aplicaciones basadas en vectores y sistemas de dibujo en 2 dimensiones (2D) hasta modeladores en 3 dimensiones (3D) a través del uso de modeladores de sólidos y superficies paramétricas. Se trata básicamente de una base de datos de entidades geométricas (puntos, líneas, arcos, etc.) con la que se puede operar a través de una interfaz gráfica. Permite diseñar en dos o tres dimensiones mediante geometría alámbrica, esto es, puntos, líneas, arcos, splines, superficies y sólidos para obtener un modelo numérico de un objeto o conjunto de ellos.

Los fabricantes del sector CAD siempre han sido punteros en aprovechar la tecnología informática más avanzada. Técnicas como el diseño vectorial, la organización de los proyectos en capas, la medición automatizada, el trabajo directo con objetos y procedimientos, la ampliación de los programas con extensiones especializadas o el diseño con modelos 3D, tienen su origen en aplicaciones de CAD, aunque en la actualidad se pueden encontrar en otros tipos de programas.

La evolución y desarrollo de las aplicaciones CAD han estado íntimamente relacionados con los avances del sector informático. El nacimiento del CAD, lo

---

<sup>1</sup> IntelliCAD Technology Consortium.

podemos situar al final periodo de los ordenadores de primera generación, pero tiene su pleno desarrollo a partir de la aparición de los ordenadores de cuarta generación en que aparecen los circuitos de alta escala de integración LSI (Large Scale Integration) y ya están desarrollados plenamente los lenguajes de alto nivel.

1. la memoria virtual utilizando sistemas de memoria jerárquicamente estructurados,
2. la multiprogramación y
3. la segmentación con el propósito de permitir la ejecución simultánea de muchas partes del programa.

Como ejemplo pondremos que a principios de los 60 Iván sutherland inventa en el laboratorio Lincoln (MIT) el primer sistema grafico CAD llamado "Sketchpad". Por el alto precio de estos ordenadores solo algunas compañías de aviación o automóviles desarrollaron en los 60 estos tipos de software.

A destacar, el gran interés estratégico que desde el principio ha tenido el CAD para las empresas, por el impacto enorme en la productividad. Las grandes empresas desde el principio han apostado por el CAD y ello supone importantes inversiones, que lógicamente potencian y convierten el CAD en un producto estratégico con un gran mercado y amplias miras hacia el futuro.



ejemplo, color, luminosidad, difusión, especularidad, características de reflexión, transparencia u opacidad, o el índice de refracción), agregar texturas, mapas de relieve y otras características.

El proceso de modelado puede incluir algunas actividades relacionadas con la preparación del modelo 3D para su posterior animación. A los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectara las porciones correspondientes del modelo.

Con todo ya dispuesto, hablaremos del texturizado que dará un aspecto realista a la escena:

- **La utilización de materiales fotorrealísticos sobre el modelo otorgarán un aspecto mucho más depurado y creíble.**

Una vez tenido el modelo básico del que haremos uso, daremos “vida” a la escena, haciendo una **composición**:

- **Esta etapa involucra la distribución de objetos, luces, cámaras y otras entidades en una escena que será utilizada para producir una imagen estática o una animación.** La iluminación es un aspecto importante de la composición de la escena. Como en la realidad, la iluminación es un factor importante que contribuye al resultado estético y a la calidad visual del trabajo terminado. Por eso, puede ser un arte difícil de dominar. Los efectos de iluminación pueden contribuir en gran medida al humor y la respuesta emocional generada por la escena, algo que es vital sobre todo en las imágenes estáticas.

Por último como conclusión del proceso, hablaremos del **renderizado de la escena**:

- **Se llama rénder al proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado a tomar una foto o en el caso de la animación, a filmar una escena de la vida real.** Generalmente se buscan imágenes de calidad fotorrealista, y para este fin se han desarrollado muchos métodos especiales. Las técnicas van

desde las más sencillas, como el rénder de malla alámbrica hasta el trazado de rayos fotográficos.

Otras técnicas de postproducción incluyen los sistemas de partículas que pueden simular lluvia, humo o fuego, el muestreo volumétrico para simular niebla, polvo y otros efectos atmosféricos, y las cáusticas para simular el efecto de la luz al atravesar superficies refractantes.

El amplio dominio de las diferentes técnicas en el campo del diseño gráfico pueden dar lugar a impresionantes obras de arte, en este caso, digital.

## **3. TRATAMIENTO DE IMÁGENES.**

### **3.1. IMÁGENES BIDIMENSIONALES (2D).**

Los gráficos 2D son aquellos que no tienen profundidad y por tanto solo se puede dibujar en un espacio R2.

En los inicios del diseño CAD, este era el medio más extendido ante la complejidad de realizar trabajos en tres dimensiones. Hoy en día las aplicaciones 2d para la informática gráfica son muy útiles para proyectos en los que se usen planos y vistas en planta, cortes, y otros.

A diferencia de las técnicas de modelado tridimensional, el diseño plano, no requiere de una renderización en tiempo real ya que sus requerimientos se cumplen con creces en casi cualquier máquina informática de hoy en día

### **3.2. IMÁGENES TRIDIMENSIONALES (3D).**

Se llaman gráficos 3D a todos los objetos que se pueden dibujar en un espacio R3: puntos, segmentos, curvas, superficies y varios cuerpos formados por caras poligonales, así como textos y macros.

En una palabra, son gráficos que tienen profundidad y además aportan mucha sensación de realismo.

En la actualidad, la mayoría de juegos 3d y aplicaciones de compleja interfaz han sido modelados mediante polígonos formando una malla y a continuación han sido renderizados.

## 4. EVOLUCIÓN DEL PROCESO INFOGRÁFICO.

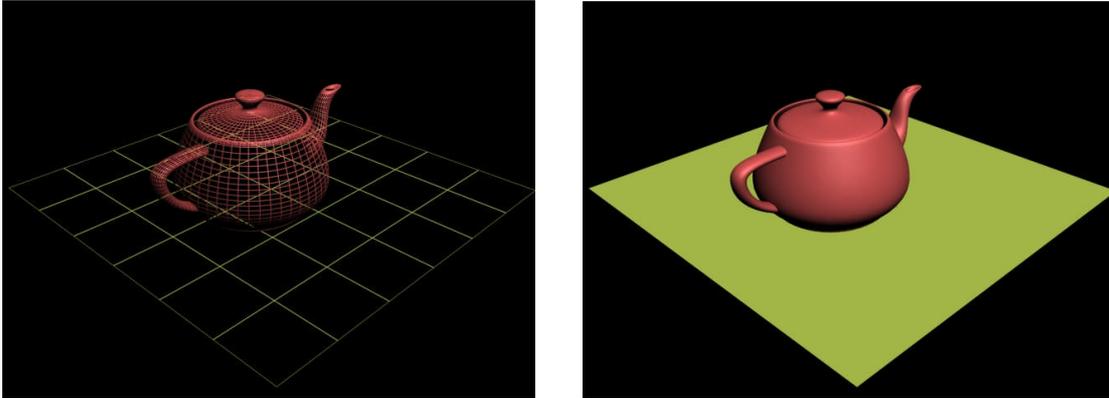
Desde hace algunos años, la infografía trata de explicar a través de gráficos e imágenes formas complejas con una inquietud en fase creciente. En el término infografía, un análisis etimológico plantea serios problemas. La partícula “info”, no proviene de informática sino de información. En realidad no es tan importante la capacidad del ordenador (base de la informática), es decir, la variable informática, ya que antes ya se realizaban imágenes de estas características a mano. Por tanto, la infografía periodística no viene de la informática, sino de la información que suministra. Este es el verdadero núcleo de la actividad.

El término también se ha popularizado para referirse a todas aquellas imágenes generadas por ordenador. Más específicamente suele hacer referencia a la creación de imágenes que tratan de imitar el mundo tridimensional mediante el cálculo del comportamiento de la luz, los volúmenes, la atmósfera, las sombras, las texturas, la cámara, el movimiento, etc.

Estas técnicas basadas en complejos cálculos matemáticos, pueden tratar de conseguir imágenes reales o no, en cuyo caso se habla de fotorrealismo.

Hablaremos de renderización cuando en infografía este proceso se desarrolla con el fin de imitar un espacio 3D formado por estructuras poligonales, comportamiento de luces, texturas, materiales, animación, simulando ambientes y estructuras físicas verosímiles, etc. Una de las partes más importantes de los programas dedicados a la infografía son los motores de render los cuales son capaces de realizar técnicas complejas como radiosity, raytrace (trazador de rayos), canal alpha, reflexión, refracción, iluminación global, etc.

Cuando se trabaja en un programa de diseño 3D por computadora, no es posible visualizar en tiempo real el acabado final deseado de una escena 3D compleja ya que esto requiere una potencia de cálculo demasiado elevada. Por lo que se opta por crear el entorno 3D con una forma de visualización más simple y técnica y luego generar el lento proceso de **renderización** para conseguir los resultados finales deseados.



Hablaremos algo sobre su evolución ya que las imágenes han servido desde siempre como un medio de comunicación.

La imagen visual nació mucho antes que la textual. Los hombres prehistóricos ya realizaban dibujos explicativos para enseñar a sus hijos los animales y las técnicas de caza, cuando todavía no contaban con un alfabeto escrito. Esto refleja la ansiedad del ser humano por explicar la realidad a través de los dibujos, mucho antes incluso que el propio hablar.

A medida que pasan los años, esta técnica se divulga cada vez más, dando lugar a otro tipo de métodos, como por ejemplo la cartografía, un ejemplo de representación gráfica próxima a la infografía, usado ya desde la Edad Media.

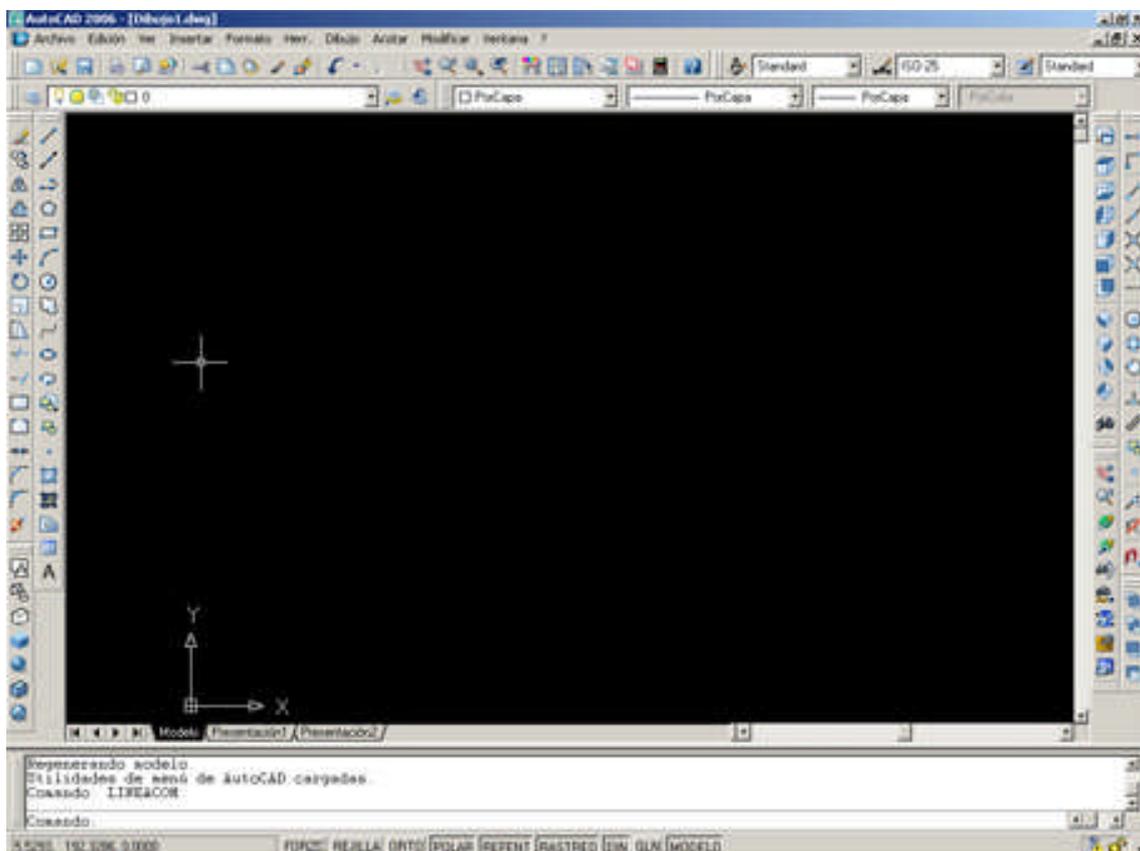
Más adelante el término infografía, estará estrechamente ligado con el periodismo, como un medio de información complementario que desarrollaban individuos provenientes de Bellas Artes y Diseño Gráfico.

En la actualidad, este procedimiento es vital y de uso cotidiano, y da las pertinentes bases para adquirir plena información de todo lo que de un modo escrito el ser humano no es capaz a concebir.

## 5. PROGRESO EVOLUTIVO Y FUNDAMENTOS DEL AUTOCAD 2006 R2.

**AutoCAD** como ya sabemos, es un programa de diseño asistido por ordenador para dibujo en 2D y 3D. Actualmente es desarrollado y comercializado por la empresa Autodesk, y la versión aquí descrita, Acad 2006 data del año 2005.

Al igual que otros programas de este campo, AutoCAD gestiona una base de datos de entidades geométricas (puntos, líneas, arcos, etc) con la que se puede operar a través de una pantalla gráfica en la que se muestran éstas, el llamado editor de dibujo. La interacción del usuario se realiza a través de comandos, de edición o dibujo, desde la línea de órdenes, a la que el programa está fundamentalmente orientado. Las versiones modernas del programa permiten la introducción de éstas mediante una interfaz gráfica de usuario, que automatiza el proceso.



Como todos los programas de CAD, procesa imágenes de tipo vectorial, aunque admite incorporar archivos de tipo fotográfico o mapa de bits, donde se dibujan figuras básicas o primitivas (líneas, arcos, rectángulos, textos, etc.), y mediante herramientas de edición se crean gráficos más complejos. El programa permite organizar los objetos por medio de *capas* o estratos, ordenando el dibujo en partes independientes con diferente color y grafismo. El dibujo de objetos seriados se gestiona mediante el uso de *bloques*, posibilitando la definición y modificación única de múltiples objetos repetidos.

Parte del programa AutoCAD está orientado a la producción de planos, empleando para ello los recursos tradicionales de grafismo en el dibujo, como color, grosor de líneas y texturas tramadas. AutoCad, a partir de la versión 14, utiliza el concepto de *espacio modelo* y *espacio papel* para separar las fases de diseño y dibujo en 2D y 3D, de las específicas para obtener planos trazados en papel a su correspondiente escala. La extensión del archivo de AutoCAD es **.dwg**, aunque permite exportar en otros formatos (el más conocido es el **.dxf**). Maneja también los formatos IGES y STEP para manejar compatibilidad con otros softwares de dibujo.

El formato **.dxf** permitía compartir dibujos con otras plataformas de dibujo CAD, reservándose AutoCAD el formato **.dwg** para sí mismo. El formato **.dxf** puede editarse con un procesador de texto básico, por lo que se puede decir que es abierto. En cambio, el **.dwg** sólo podía ser editado con AutoCAD, si bien desde hace poco tiempo se ha liberado este formato (DWG), con lo que muchos programas CAD distintos del AutoCAD lo incorporan, y permiten abrir y guardar en esta extensión.

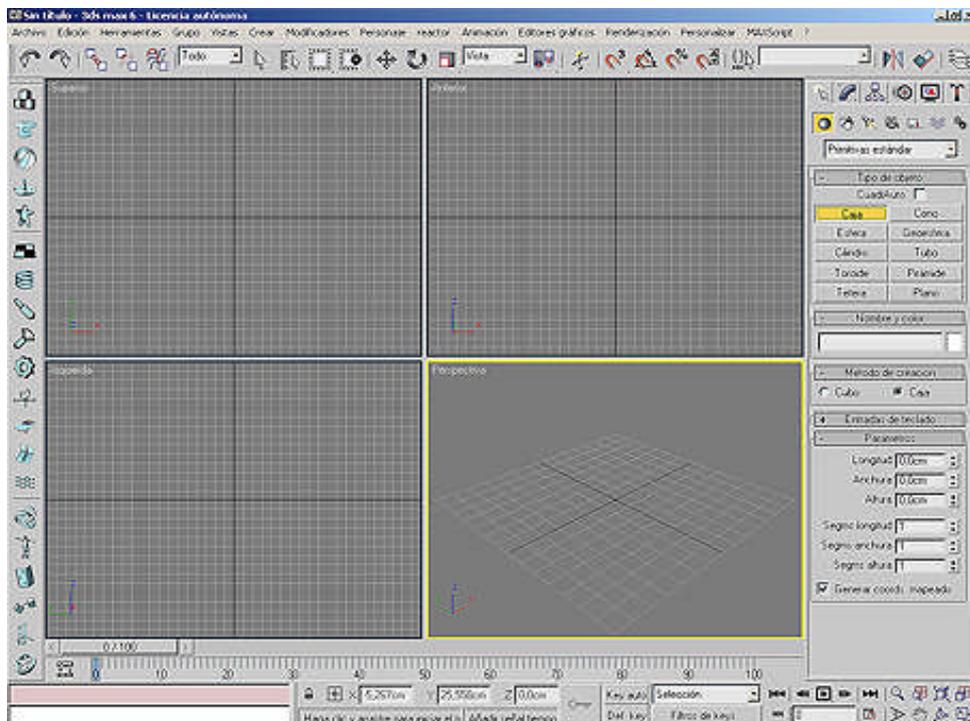
El formato **.dwg** ha sufrido cambios al evolucionar en el tiempo, lo que impide que formatos más nuevos **.dwg** no puedan ser abiertos por versiones antiguas de AutoCAD u otros CADs que admitan ese formato. La última versión de AutoCAD hasta la fecha es el AutoCAD 2008, y tanto él como sus productos derivados (como Architectural DeskTop ADT o Mechanical DeskTop MDT) usan un nuevo formato no contemplado o trasladado al OpenDWG, que sólo puede usar el formato hasta la versión 2000.

Las aplicaciones del programa son múltiples como ya hemos citado anteriormente, desde proyectos y presentaciones de ingeniería, hasta diseño de planos o maquetas de arquitectura.

## 6. Progreso evolutivo y fundamentos del 3D Studio MAX 6 SP1

**3D Studio Max** o simplemente *Max*, es un programa de creación de gráficos y animación 3D desarrollado por Autodesk actualmente. Fue desarrollado como sucesor para sistemas operativos Win32 del 3D Studio creado para DOS. Kinetix fue más tarde fusionada con la última adquisición de Autodesk, Discreet Logic. La versión más reciente de este programa es 3ds max 9 en inglés que data de noviembre de 2006, pero en nuestro caso, haremos uso de la versión 6 debido a limitaciones tanto en hardware como en software.

Esta herramienta es uno de los programas de animación 3D y texturizado foto realista más utilizados. Dispone de una sólida capacidad de edición, una omnipresente arquitectura de plugins 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 arquitectura.



Dentro de este programa podremos distinguir varios aspectos relevantes:

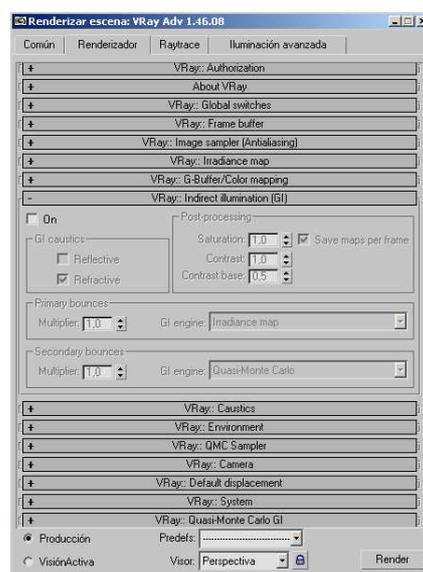
- Modelado de objetos partiendo de primitivas.

- Texturizado del modelo a representar.
- Iluminación y entorno del modelo y puesta final en escena ya sea como imagen o vídeo renderizado.

Haremos especial mención del tercer apartado, ya que se ha utilizado un plugin complementario de renderización que hace las funciones del sistema que trae por defecto el programa, pero ahorrando una cantidad considerable de tiempo y cómo no, aumentando la calidad; la aplicación en sí es el renderizador Vray en su versión 1.46.08.



Además de renderizador, Vray proporciona excelentes resultados de iluminación debido a los abundantes controles de simulación de luz y cáusticas de que hace gala, no obstante ya iremos viendo más adelante este completísimo plugin.



## 7. ¿POR QUÉ SE HA OPTADO POR ESTOS SOPORTES DE SOFTWARE?:

Mientras que las aplicaciones bidimensionales, o “2D”, sólo manipulan elementos geométricos planos, las aplicaciones tridimensionales, o “3D”, también pueden manipular elementos tridimensionales. Es decir, manipulan tres coordenadas frente a las dos coordenadas bidimensionales. En consecuencia, una aplicación 3D podrá manejar, al menos, los mismos elementos planos que una aplicación 2D, pero con la ventaja de poder situarlos en un espacio tridimensional. De esta forma diferentes elementos planos pueden estar contenidos en planos distintos. Además, en las aplicaciones 3D de mayores prestaciones, también se pueden manipular directamente formas geométricas tridimensionales (como prismas, cilindros, esferas, etc.) sin necesidad de descomponerlas en elementos planos que formen una figura que sea equivalente, bajo ciertas condiciones, al modelo; tal como se hace en geometría descriptiva.

Sin más, todo esta explicación viene simplificada de una manera bastante intuitiva, en las aplicaciones que en este proyecto se dan a conocer, o bien generalizando, herramientas de diseño CAD.

### 7.1. VENTAJAS, RECURSOS, APLICACIONES Y RESULTADOS DE QUE ES OBJETO:

Debe notarse que, manipular elementos geométricos tridimensionales, no sólo implica tener capacidad para almacenar la información necesaria para describir la forma y posición de dichos elementos en un espacio tridimensional. También implica disponer de un gestor de visualización que permita generar de forma automática todas las proyecciones que el usuario pueda requerir (resolviendo todos los problemas de ocultación, iluminación, etc.), de forma que el conjunto de formas geométricas se comporte como una *escena tridimensional* que se ve a través de la pantalla plana del ordenador.

Por tanto, con las aplicaciones 3D se puede trabajar directamente con el modelo geométrico tridimensional. El usuario manipula una escena de formas tridimensionales, y no tiene que realizar el proceso de proyección para visualizar la escena. Mientras, con aplicaciones 2D es el usuario quien debe generar todo tipo de

imágenes planas aplicando el proceso de “modelado+proyección”. En otras palabras, mientras que con aplicaciones 2D se puede hacer *delineación asistida*, en aplicaciones 3D se pueden construir *modelos geométricos virtuales*.

Por ejemplo, el uso de los prototipos virtuales creados a partir de estas aplicaciones, permite el posterior análisis con otras aplicaciones. Por tanto, sustituyendo la delineación por el modelado se reduce el tiempo de ejecución y con el consiguiente ahorro de los costes.

En este trabajo vamos a justificar los aspectos críticos que nos parecen más interesantes para evaluar este tipo de aplicaciones. Los aspectos considerados son, fundamentalmente, los siguientes:

- . Compatibilidad
- . Amigabilidad de la interacción (GUI), acceso a la información
- . Visualización
- . Dominio de los modelos
- . Entorno de esbozo (intención de diseño)
- . Asociatividad 2D <>3D
- . Funciones definidas por el usuario
- . Gestión de datos del producto

Por otra parte, se usan las bibliotecas de elementos en casi la totalidad de las aplicaciones CAD existentes. También es amplio el uso de las tablas de familia, consistentes en la creación de elementos relacionados por parámetros dimensionales, de modo que, definiendo uno de ellos y variando ciertas cotas, se obtiene el resto de los elementos.

Por otra parte si puntualizamos aspectos triviales, podremos sacar varias conclusiones:

- El entorno del que se habla, está presente en la actualidad en casi todas las empresas del sector, adoptando un uso casi universal del software.
- Tras el estudio comparativo realizado, quedan patentes las ventajas del modelado frente a la delineación, siendo una de las principales ventajas la creación de prototipos virtuales que reducen el tiempo de puesta del producto en el mercado, permitiendo a la empresa, con mucho, incrementar su competitividad.

- Por otra parte habrá que hacer especial hincapié, en el bajo coste debido a la mayor velocidad en el diseño.
- Los complementos y/o recursos de que dispone este tipo de entorno hacen que encontremos soluciones para casi todos los problemas y objetivos marcados.
- La compatibilidad entre ficheros con este tipo de extensiones .dwg/dwf .max es total en las versiones que se estudian aquí.
- La precisión de que hacen gala estos sistemas es más que evidente y sólo dependiente del sistema de impresión que se utilice para su correcta visualización en formato ya sea papel u otro.
- Asociatividad de elementos diferentes mediante su clasificación por capas y diagramas de flujo para una mayor claridad en lo que se refiere a la presentación de datos y modelos.
- Es también un entorno programable en múltiples lenguajes, entre otros se pueden destacar:
  - AutoLISP - Una adaptación de LISP para AutoCad.
  - Visual LISP - Nueva versión de AutoLISP para las últimas versiones de AutoCAD, con más funciones y un IDE visual integrado.
  - VBA - Programación con el Visual Basic para aplicaciones integrado.
  - ObjectARX - Permite desarrollar librerías en C/C++ para ser utilizadas por AutoCAD.
  - MAXScript – Interfaz de programación para 3D Studio.
- Ambas aplicaciones engloban herramientas para captación de datos de forma directa, ya sean medidas, intersecciones, cálculos de volúmenes, áreas.
- La interfaz que incorporan, es muy intuitiva, dando al usuario la posibilidad de interactuar con el programa de manera autodidacta o bien apoyándose de tutoriales que van desde lo más simple hasta lo más complejo.
- Posibilidad de exportación de animaciones y texturas, creación de vídeos y postproducción en entornos MAX

Dentro de las aplicaciones de CAD, los sistemas CAD 3D amplían las prestaciones de los 2D, incorporando, además de aspectos propios del 2D, herramientas de modelado con las consiguientes herramientas de generación

automática de planos, herramientas de captura de la intención de diseño del ingeniero, gestores de datos del producto asociados, herramientas potentes de visualización, herramientas para la reutilización de modelos geométricos, herramientas de captura de datos fundamentales del proyecto en la fase más temprana del diseño como es la fase conceptual, etc.

Pongamos unos ejemplos de estos sistemas haciendo especial consideración en los campos que desarrollan:

- **Arquitectura y construcción:** Las herramientas CAD que engloban este campo crean modelizados 3D foto realistas y paseos virtuales con herramientas de modelizado a priori sencillas acerca de ingeniería civil y obras arquitectónicas.
- **Industria y fabricación:** Que comprende de manera intuitiva y fácil el diseño de sistemas mecánicos, eléctricos e incluso con aplicaciones adicionales la simulación de los mismos.
- **Infraestructuras y GIS:** Creación de mapas mediante elevaciones digitales y gestión de datos espaciales.
- **Sistemas de gestión:** Principalmente destinados a trabajar en segundo plano con todos los tipos de aplicaciones CAD, mediante diagramas y otras ayudas que simplifiquen el trabajo.

## **8. ASPECTOS GENERALES SOBRE EL HARDWARE:**

### **8.1. CONCEPTOS A TENER EN CUENTA:**

En este apartado daremos a conocer algunos de los conceptos básicos a tener en cuenta para poder entender de una manera más profunda lo contenido en puntos posteriores.

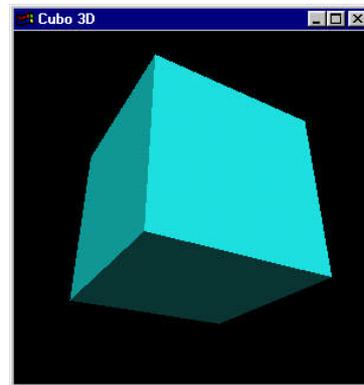
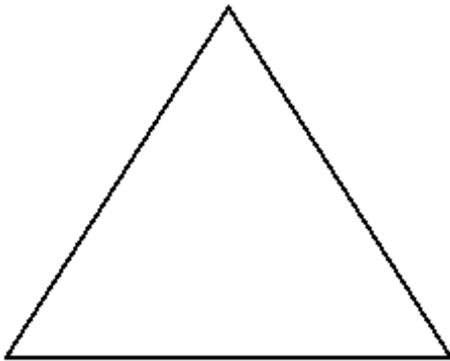
**Píxel:** Esta palabra proviene de la abreviación del término inglés PEL (*Picture Element*, elemento del dibujo) El píxel es la unidad más pequeña que un adaptador puede generar. Los colores que pueden tener los píxeles dependen de la tarjeta de vídeo que se posea.

Si se tiene una resolución "grande" de 1024x768 por ejemplo, los píxeles serán de un tamaño diminuto, si por lo contrario se adopta una resolución más baja, cada píxel adoptará un tamaño más grande cuanto más baja sea ésta.

**Polígono:** Un polígono es una figura cerrada con espacio dentro y fuera de él, construido al menos por tres líneas.

Los triángulos de tres caras forman la base elemental de creación de objetos en 3d.

Por lo general todos los gráficos y objetos diseñados se componen de polígonos que son calculados y procesados para su correspondiente visualización en conjunto.



En la mayoría de los casos, se necesitan cientos o miles de polígonos para un objeto sencillo en 3D, con la consiguiente cantidad de datos para generar y manipular, tarea ya de las unidades de proceso.

**Gráfico vectorial:** Los gráficos vectoriales o *modelado geométrico* son los que se representan en los gráficos por ordenador por medio de "trazos", es decir, por primitivas geométricas como puntos, líneas, curvas o polígonos.

**Bitmap:** Un mapa de bits, o bitmap, es la representación binaria en la cual un bit o conjunto de bits corresponde a alguna parte de un objeto como una imagen o fuente.

Por ejemplo, en sistemas monocromáticos, un bit en el mapa de bits representa un píxel en pantalla. Para la escala de grises o color, varios bits en el mapa de bits representan un píxel o grupo de píxeles. El término también puede hacer referencia al área de memoria que contiene el mapa de bits.

**Unidad de proceso CPU:** O *Central Procesor Unit* (Unidad Central de Proceso), es un componente en un computador digital que interpreta las instrucciones y procesa los datos contenidos en los programas de ordenador. Los CPU proporcionan la característica fundamental del computador digital, la programabilidad, y son uno de los componentes necesarios encontrados en los computadores de cualquier tiempo, junto con el almacenamiento primario y las facilidades de entrada/salida. Es conocido como microprocesador el CPU que es manufacturado con circuitos integrados. Con esta somera explicación, basta para tener una ligera idea de la importancia que tiene en el proceso de desarrollo de diseño gráfico.

**Unidad de proceso GPU:** O *Graphics Processing Unit* (Unidad de Procesado de Gráficos) Una GPU es una CPU dedicada exclusivamente al procesamiento de gráficos, para aligerar la carga de trabajo del procesador central en aplicaciones como los videojuegos. De esta forma, mientras gran parte de lo relacionado con los gráficos se procesa en la GPU, la CPU puede dedicarse a otro tipo de cálculos (como la inteligencia artificial o los cálculos mecánicos en el caso de los videojuegos).

Una GPU implementa ciertas operaciones gráficas llamadas primitivas optimizadas para el procesamiento gráfico. Una de las primitivas más comunes para el procesamiento gráfico en 2D es la operación BitBLT, que se encuentra en una unidad llamada "blitter", adicionalmente existen primitivas para dibujar rectángulos, triángulos, círculos y arcos. Los GPU modernos tienen muchas más primitivas que soportan también gráficos de 3D.

**Sprite:** Un sprite es un objeto gráfico que puede ser movido por encima de una imagen de fondo (Un cursor).

**Tasa de Frames por segundo:** Definido por el acrónimo *fps* (*frames per second*, Fotogramas por segundo), es la velocidad a la que se muestran los fotogramas en una animación tridimensional. Habrá que tener en cuenta los formatos profesionales con los que trabajan los sistemas de vídeo, que son: PAL/SECAM = 25 FPS (sistema europeo); NTSC = 29.97 FPS.

**Texel:** Es un píxel de una textura mapeada que ha sido aplicado a un polígono. Podemos decir que un texel es el equivalente a un píxel pero en 3D. Se suele utilizar para crear objetos volumétricos como nubes o simulaciones aerodinámicas.

**API:** Del inglés *Application Programming Interface* - Interfaz de Programación de Aplicaciones, es un conjunto de especificaciones de comunicación entre componentes software. Se trata del conjunto de llamadas al sistema que ofrecen acceso a los servicios del sistema desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las APIs asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API. Los gráficos 3D se han convertido en algo muy popular, particularmente en juegos de computadora, al punto que se han creado APIs especializadas para facilitar los procesos en todas las etapas de la generación de gráficos por computadora. Estas APIs han demostrado ser vitales para los desarrolladores de hardware para gráficos por computadora, ya que proveen un camino al programador para acceder al hardware de manera abstracta, aprovechando las ventajas de tal o cual placa de video.

**Renderización:** La palabra renderización es una adaptación al castellano del vocablo inglés *rendering* y que define un proceso de cálculo complejo desarrollado por un ordenador destinado a generar una imagen 3D o secuencia de imágenes 3D. La traducción más fidedigna es "interpretación", aunque se suele usar el término inglés. También se emplean coloquialmente los términos "renderizar" y "renderizado".

La renderización se aplica a los gráficos por ordenador, más comúnmente a la infografía

En infografía este proceso se desarrolla con el fin de imitar un espacio 3D formado por estructuras poligonales, comportamiento de luces, texturas, materiales, animación, simulando ambientes y estructuras físicas verosímiles, etc. Una de las partes más importantes de los programas dedicados a la infografía son los motores de

render los cuales son capaces de realizar técnicas complejas como radiosidad, trazado de rayos, reflexión, refracción, iluminación global, etc.

Cuando se trabaja en un programa de diseño 3D por computadora, no es posible visualizar en tiempo real el acabado final deseado de una escena 3D compleja ya que esto requiere una potencia de cálculo demasiado elevada. Por lo que se opta por crear el entorno 3D con una forma de visualización más simple y técnica y luego generar el lento proceso de renderización para conseguir los resultados finales deseados.

**Rasterización:** Se denomina rasterizar a la operación de convertir una imagen vectorial en otra de mapa de bits.

**Buffer:** Un buffer de datos es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras que está esperando ser procesada. El Z-Buffer es el usado para el renderizado de imágenes 3D.

**Pipeline gráfico:** La segmentación es un método por el cual se consigue aumentar el rendimiento de algunos sistemas electrónicos digitales. Es aplicado, sobre todo, en microprocesadores y en núcleos gráficos que atienden unos procesos de trabajo concretos.

**Blitter:** Blitter (acrónimo de Bit-Block Transfer) es un chip coprocesador dedicado a transferencias de datos en memoria, normalmente independiente de la CPU usando métodos bit blit (operación gráfica de computadores en la cual dos patrones de mapas de bit son combinados en uno)

## **8.2. IMPORTANCIA DE LOS DIFERENTES COMPONENTES:**

La característica fundamental de los sistemas CAD en lo que se refiere al hardware es que son sistemas con gran capacidad de cálculo y, sobre todo, con subsistemas gráficos de altas prestaciones.

Otra característica diferenciadora es el uso de dispositivos específicos de entrada y salida, por las necesidades que la funcionalidad de estos sistemas requiere del tipo de trazadores o plotters, dispositivos digitalizadores (scanners) y un largo etcétera.

Se puede dar el caso de que el trabajo desempeñado sea tan extenso, que el hecho de usar solamente una máquina sea a veces escaso y se recurra a sistemas de renderizado y animado e red.

Metiéndonos más de lleno en especificaciones, podremos decir que los pilares de todo sistema CAD, se basan en la elección de un buen monitor, una memoria abundante y adecuada, procesador y una GPU (Graphic Processor Unit) que cumpla con todas las expectativas.

Como todos sabemos la visualización apropiada del espacio de trabajo es muy importante, y por ello será necesario disponer de pantallas que nos permitan trabajar de manera fluida y sin mal interpretaciones a la hora de manejar nuestro diseño. Anteriormente el soporte recaía en formatos de 17", en la actualidad lo normal oscila entre 19" o incluso 21", pudiéndose incluso montar dispositivos con doble pantalla, para alternar diferentes espacios de trabajo.



Debido a la infinidad de procesos matemáticos que conllevan este tipo de trabajos, no menos importante es la memoria, ya sea memoria RAM o la propia de la GPU. Hoy en día la variedad en cuanto a este tipo de campo se refiere es muy extensa. Podremos encontrar memorias RAM genéricas, que lo único que aporten sea cantidad bruta, o por otra parte rapidez además de lo anteriormente especificado; en base a lo que el CAD se refiere, se buscarán ambos propósitos, rapidez y cantidad. En el aspecto de las GPU, la memoria será una simple antesala para almacenar texturas y otros elementos a grandes rasgos.

La unidad central de proceso (CPU), realizará los cálculos oportunos en procesos tipo render y animación, con lo cual es muy conveniente, que su velocidad de núcleo sea lo más elevado posible dentro de nuestras posibilidades.

Por último, haremos especial hincapié en el dispositivo gráfico del equipo, la tarjeta gráfica por la cuales hoy compiten dos compañías netamente diferenciadas Nvidia y ATI. Si bien es cierto que para los procesos de diseño en malla tridimensional, se exprime casi exclusivamente el procesador (CPU), para los desarrollos de render y trabajos derivados con implementación de luces y otros factores, es de vital importancia un conjunto de chips específicos para acelerar estas aplicaciones. Lo que se pretende dar a conocer en este punto, es la diferencia entre el uso de GPUs profesionales y domésticas:

¿Rinden más?

¿Son más eficientes?

¿Son compatibles con juegos?

Analizándolo de esta manera, alguien que haga uso de ellas para jugar lo que le importa en mayor medida es que la imagen se vea bien y se procese rápido, pues bien, a un diseñador 3D lo que menos le importa es que su procesador grafico "procese" una imagen "completa" muchas veces por segundo, si no que le importa mas que los vértices, y los millones de polígonos que usa sean procesados con rapidez y con eficiencia.

Por lo tanto a la primera pregunta, diremos que su rendimiento será altamente superior si las utilizamos con aplicaciones específicas, 3D Studio, Maya... etc.; evidentemente su eficiencia es más que notoria en entornos para los que han sido diseñadas. Por otra parte el problema puede venir del lado de la compatibilidad con aplicaciones de tipo lúdico, ya que las instrucciones que manejan este tipo de chips no satisfacen en muchas ocasiones sus requerimientos.

### **8.3. DEFINICIÓN, TIPOS DE APIS GRÁFICAS USADAS:**

Como habíamos visto anteriormente en otro apartado, una API (Application Programming Interface) o Interfaz de Programación de Aplicaciones, es un conjunto de funciones que realizan tareas específicas facilitando la vida al programador. Al hablar de API grafica nos referimos a un set de funciones para inicializar por ejemplo modos

gráficos, realizar copiado de datos de la memoria del computador a la tarjeta de video (blitting), etc. Existen API's específicas para cada tipo de tarea que queramos realizar y a continuación pasaremos a describir algunas de las más importantes para este sector.

- API OpenGL
- API Direct3D

### 8.3.1. API OpenGL:

OpenGL es un API de renderización de 2D y 3D desarrollado por Silicon Graphics que se ha convertido en el estándar de facto de la industria. Básicamente es una herramienta para tomar una imagen en 3D, y producir una imagen 2D adecuada para dibujarla en pantalla. A su vez, es una especificación, es decir, un documento que describe un conjunto de funciones y su comportamiento exacto. A partir de ella, los fabricantes de hardware crean implementaciones (bibliotecas de funciones creadas para enlazar con las funciones de la especificación OpenGL, utilizando aceleración hardware cuando sea posible). Dichos fabricantes tienen que superar pruebas específicas que les permitan calificar su implementación como una implementación de OpenGL.

Existen implementaciones eficientes de OpenGL suministradas por fabricantes para Mac OS, Microsoft Windows, Linux, varias plataformas Unix, Wii y PlayStation 3. Existen también varias implementaciones software que permiten que OpenGL esté disponible para diversas plataformas sin soporte de fabricante.

OpenGL tiene dos propósitos principales:

- Ocultar la complejidad de la interfaz con las diferentes aceleradoras 3D, presentando al programador una API única y uniforme.
- Ocultar las diferentes capacidades de las diversas plataformas hardware, requiriendo que todas las implementaciones soporten el conjunto completo de características de OpenGL (utilizando emulación software si fuese necesario).

A continuación nombraremos algunas características que esta API implementa:

- **Primitivas geométricas:** Permiten construir descripciones matemáticas de objetos. Las actuales primitivas son: puntos, líneas, polígonos, imágenes y bitmaps.
- **Codificación del Color:** Modos RGBA (Rojo-Verde-Azul-Alfa) o de color indexado.
- **Visualización y Modelado:** Permite disponer objetos en una escena tridimensional, mover nuestra cámara por el espacio y seleccionar la posición ventajosa deseada para visualizar la escena de composición.
- **Mapeado de texturas:** Ayuda a traer realismo a nuestros modelos por medio del dibujo de superficies realistas en las caras de nuestros modelos poligonales
- **La iluminación de materiales:** Es una parte indispensable de cualquier gráfico 3D. OpenGL provee de comandos para calcular el color de cualquier punto dadas las propiedades del material y las fuentes de luz en la habitación.
- **El doble/triple buffering:** Ayuda a eliminar el parpadeo de las animaciones. Cada fotograma consecutivo en una animación se construye en un buffer separado de memoria y mostrado solo cuando está completo.
- **El Anti-aliasing:** Reduce los bordes escalonados en las líneas dibujadas sobre una pantalla de ordenador. Los bordes escalonados aparecen a menudo cuando las líneas se dibujan con baja resolución. Es una técnica común en gráficos de ordenador que modifica el color y la intensidad de los píxeles cercanos a la línea para reducir el zig-zag artificial.
- **El sombreado Gouraud:** Técnica usada para aplicar sombreados suaves a un objeto 3D y producir una sutil diferencia de color por sus superficies.
- **El Z-buffering:** Mantiene registros de la coordenada Z de un objeto 3D. El Z-buffer se usa para registrar la proximidad de un objeto al observador, y es también crucial para el eliminado de superficies ocultas que aligeran la tarea.
- **Efectos atmosféricos:** Como la niebla, el humo y las neblinas hacen que las imágenes producidas por ordenador sean más realistas. Sin efectos atmosféricos las imágenes aparecen a veces irrealmente nítidas y bien definidas. *Niebla* es un término que en realidad describe un algoritmo que simula neblinas, brumas, humo o polución o simplemente el efecto del aire, añadiendo profundidad a nuestras imágenes.

- **El Alpha blending:** Usa el valor Alfa (valor de material difuso) del código RGBA, y permite combinar el color del fragmento que se procesa con el del pixel que ya está en el buffer.
- **Los planos de plantilla:** Permiten restringir el trazado a ciertas regiones de la pantalla.
- **Las listas de Display:** Permiten almacenar comandos de dibujo en una lista para un trazado posterior.
- **Los Evaluadores Polinómicos:** Sirven para soportar B-splines racionales no uniformes, es decir, para ayudar a dibujar curvas suaves a través de unos cuantos puntos de referencia, ahorrándose la necesidad de acumular grandes cantidades de puntos intermedios.
- **Características de Feedback, Selección y Elección:** Ayudan a crear aplicaciones que permiten al usuario seleccionar una región de la pantalla o elegir un objeto dibujado en la misma. El modo de feedback permite al desarrollador obtener los resultados de los cálculos de trazado.
- **Primitivas de Raster:** Bitmaps y rectángulos de píxeles
- **Operaciones con Píxeles**
- **Transformaciones:** Rotación, escalado, perspectivas en 3D...etc.

Muchas aceleradoras 3D actuales proporcionan una funcionalidad superior a la básica aquí expuesta, pero las nuevas características generalmente son mejoras de esta pipeline básica más que cambios revolucionarios de esta.

#### **Una descripción somera del proceso en la pipeline gráfica podría ser:**

1. Evaluación, si procede, de las funciones polinomiales que definen ciertas entradas, como las superficies NURBS (*Non Uniforms Rational B-Splines*), aproximando curvas y la geometría de la superficie.
2. Operaciones por vértices, transformándolos, iluminándolos según su material y recortando partes no visibles de la escena para producir un volumen de visión.
3. Rasterización, o conversión de la información previa en píxeles. Los polígonos son representados con el color adecuado mediante algoritmos de interpolación.
4. Operaciones por fragmentos o segmentos, como actualizaciones según valores venideros o ya almacenados de profundidad y de combinaciones de colores, entre otros.

5. Por último, los fragmentos son volcados en el Frame buffer (designa a un método de acceso a dispositivos gráficos).

Por último decir haciendo referencia a este tipo de API, que existe una versión de código abierto que permite la estructura, la sintaxis y la semántica del OpenGL, pero obteniendo además ventajas en el campo de disponibilidad:

- Mesa 3D esta disponible y puede ser compilado en virtualmente cualquier plataforma moderna.
- Por no ser una implementación oficial de OpenGL por razones de licencia, los autores de Mesa 3D han trabajado para mantener el API en línea con los más recientes estándares y pruebas conformadas de OpenGL.
- Mientras Mesa 3D soporta varios tipos de aceleradores gráficos por hardware, también puede ser compilado como un renderizador solamente de software. Por lo que es código abierto, es posible usarlo para estudiar el comportamiento interno de un render compatible con OpenGL.

### **8.3.2. API Direct3D (aplicación bajo directX)**

Microsoft DirectX es una aplicación de programas (API) para PC's basadas en Windows que permite a los programadores escribir códigos que da acceso a características especiales de ciertos equipos sin necesidad de tener que escribir códigos específicos para cada equipo. Esto significa que DirectX permite que el mismo código, por ejemplo, en el caso de los videojuegos, sean usados en diferentes tipos de ordenadores.

En general se compone de:

- DirectPlay: Funciones de red para Internet, network, módem, etc.
- DirectGraphics: Calculo de gráficos 3D y 2D.
- DirectMusic: Reproducción de sonido, música, mp3, etc.
- DirectInput: Manejo del teclado, ratón, joystick, volantes, etc.
- DirectShow: Permite reproducir vídeos en general, dvd, mpg, etc.

En una palabra: DirectX es un conjunto de librerías para programadores de Windows que deseen realizar cualquier tipo de juego para PC.

**Direct3D** es parte de DirectX una API propiedad de Microsoft como ya hemos dicho, disponible tanto en los sistemas Windows de 32 y 64 bits, como para sus consolas Xbox y Xbox 360 para la programación de gráficos en 3D .

Esta API facilita el manejo y trazado de entidades gráficas elementales en cualquier aplicación que despliegues gráficos en 3D, como líneas, polígonos y texturas, así como efectuar de forma transparente transformaciones geométricas sobre las entidades desplegadas. Provee también una interfaz transparente con el hardware de aceleración gráfica.

Direct3D se usa en aplicaciones donde el rendimiento es casi tan importante como los juegos, aprovechando el hardware de aceleración gráfica si está disponible en la tarjeta gráfica. Además permite que las aplicaciones se ejecuten en pantalla completa o en ventana.

Es utilizada por la mayoría de los juegos en 3D y algunas aplicaciones disponibles en la plataforma Windows.

Direct3D es una API mayoritariamente para 3D. Contiene muchos comandos para la renderización en 3 dimensiones, pero también contiene algunos comandos para la renderización gráfica en 2D. Microsoft actualiza continuamente esta API para implementar la última tecnología disponible en la tarjeta gráfica 3D, además ofrece emulación completa por software de vértices (teoría de grafos) pero no emulación por software de píxel.

Sopesando todo lo relacionado con estos dos puntos y valorando lo que actualmente se depura en estos entornos, podremos distinguir una serie de pros y contras:

**Rendimiento:** La productividad del software creado por medio de una u otra API sigue siendo un parámetro muy debatido. Dado que DirectX implementa los drivers a nivel del núcleo en el sistema operativo y de nivel usuario con un paquete provisto por Microsoft, existe la necesidad de hacer un cambio entre un modo y otro en las llamadas a la API. Como consecuencia, la CPU utiliza valiosos microsegundos en completar la operación, quedando inutilizada para otra tarea. En cambio, OpenGL, a través de una técnica conocida como *marshalling* (proceso por el cual debe pasar toda información para que ésta sea utilizable en entornos heterogéneos.), evita ese

inconveniente y obtiene una valiosa ventaja en el rendimiento final. A partir de Windows Vista, DirectX podrá sortear esta deficiencia mediante este mismo procedimiento, ya que los drivers estarán funcionando más a nivel de usuario.

**Extensibilidad:** OpenGL incluye una propiedad que le posibilita extender sus capacidades a través de un driver, añadiendo nuevas funcionalidades. Pero esto puede conducir a generar una gran cantidad de variantes de la API, provocando cierta confusión entre los programadores. Por el lado de DirectX, Microsoft se encarga de mejorarla pero con menos regularidad. No obstante, desde hace un par de años que las actualizaciones aparecen cada vez más seguidas.

**Usuarios:** La ventaja en el ámbito profesional está del lado de OpenGL. De hecho, muchas de las placas profesionales sólo soportan esta API, lo que es un decir. Las películas animadas por ordenador son desarrolladas en base a OpenGL, dado su carácter de plataforma de propósitos generales. La situación cambia cuando lo miramos desde el punto de vista del mercado de los videojuegos. DirectX no requiere de la implementación completa que demanda OpenGL para funcionar (incluyo funciones no soportadas por el hardware), por lo que se torna más sencillo de utilizar. Para hacerlo simple, podemos decir que Direct3D es un acceso directo de bajo nivel a los registros disponibles. Además, posee un mecanismo que determina si una funcionalidad está o no presente en el dispositivo, evitando la implementación obligatoria y dejando a elección del diseñador del videojuego su incorporación en el software. Por este detalle importante es que vemos muchos más juegos de ordenador programados en DirectX que en OpenGL.

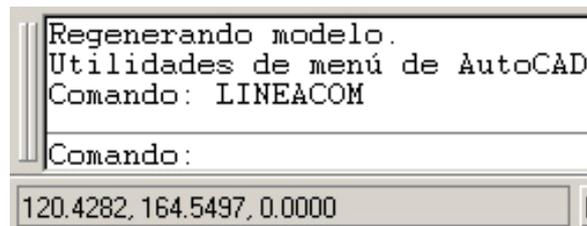
## 9. TRATAMIENTO DE LO REPRESENTADO

*AutoCAD* usa tres líneas perpendiculares entre sí, que intersecan en un punto llamado *origen*, para identificar la ubicación de los puntos en el espacio. Estas tres líneas reciben los nombres *eje X*, *eje Y* y *eje Z*. Con ellas, la situación de cualquier punto en el espacio 3D puede especificarse por la distancia de dicho punto al origen en la dirección X, la distancia en la dirección Y y la distancia en la dirección Z. Estas tres distancias, a las que llaman *coordenadas*, se suelen escribir juntas, separadas por comas.

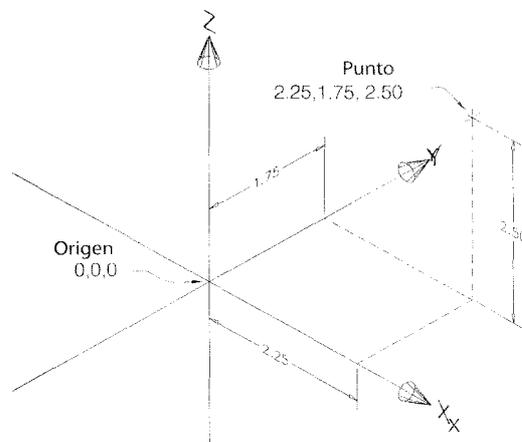
Por ejemplo, un punto que tiene como coordenadas 2.25, 1.75, 2.50 se encuentra a 2.25 unidades del origen en la dirección de X, a 1.75 unidades en la dirección de Y y a 2.50 unidades en la dirección de Z. Las coordenadas del origen son

0,0,0. Las unidades que se usan para las distancias pueden ser centímetros, pulgadas, milímetros, millas, años luz, micras o cualquier otra unidad que se escoja.

*AutoCAD* muestra las coordenadas del cursor de pantalla en la barra de estado del editor de dibujo de la pantalla. A partir de la Versión 14, los tres valores de las coordenadas se muestran (la coordenada Z será, por lo general, 0). Las versiones anteriores de *AutoCAD* sólo muestran los valores de X e Y.



Los ejes X, Y y Z se extienden con una longitud infinita en las direcciones correspondientes, siempre a partir del origen, teniendo cada eje un sentido positivo y uno negativo. Un signo menos precediendo al valor *de* la coordenada, significa que la distancia indicada se *encuentra* en la dirección negativa *del* eje correspondiente, a partir del origen.



*AutoCAD* hace referencia a esta configuración para la designación de la ubicación *de* los puntos como el *Sistema de coordenadas Universal (SCU)* además dispone de un sistema de coordenadas secundario, llamado *Sistema de coordenadas Personal (SCP)* que puede moverse dentro del SCU para facilitar el dibujo de elementos en 3D.

Evidentemente la representación de cualquier punto se rige de manera única e independiente por este tipo de sistema; por ello diremos que a mayor número de puntos mayor cantidad de cálculos y complejidad a la hora de actualizar el proceso en tiempo real.

## TEMA 2: LA IMAGEN

### 1. INTRODUCCIÓN

Para que el resultado de la visualización final que se pretende conseguir sea satisfactorio, es conveniente conocer algunos aspectos sobre las imágenes. Estos aspectos son; los formatos de archivos de imagen y vídeo, sus diferentes calidades y compresiones, y saber cuales son más convenientes según el resultado que queramos conseguir ya que hoy en día la calidad de una imagen depende en gran medida del algoritmo de compresión con que se ejecute.

### 2. FORMATOS GRÁFICOS

#### 2.1. MODO DE MAPA DE BITS

También llamadas imágenes de trama, usan una cuadrícula formada por pequeños cuadros (pixels) para representar los gráficos. Cada píxel tiene una posición determinada y un valor de color asignado. Al modificar estos gráficos manipulamos grupos de píxeles en lugar de objetos y formas. Así trabajan los programas de retoque fotográfico. El manejo por parte del programa de este conjunto de puntos o píxeles que aparecen en pantalla no requiere de ningún tratamiento matemático, aunque por otro lado los ficheros pueden resultar de gran tamaño. Es conveniente por tanto el uso de algoritmos de compresión para éste tipo de imágenes.

Ya que estas imágenes pueden representar graduaciones sutiles de sombras y colores, las imágenes de mapa de bits constituyen el medio electrónico más común para las imágenes de tono continuo (fotografías). Las imágenes de mapa de bits dependen de la resolución, es decir, representan un numero fijo de píxeles. Como consecuencia, aparecerán dentadas y perderán detalle si se las escala en pantalla o se imprimen a mayor resolución que a la que se crearon.

Algunos de los formatos de archivo para imágenes digitales de mapa de bits más utilizados son:

**Bitmap (BMP):** es el formato estándar de Windows. Tiene una profundidad de imagen de 1 a 24 bits. En imágenes de 4 y 8 bits, podemos escoger la compresión

Run-Length-Encoding (RLE); este esquema de compresión no produce pérdidas, es decir, no elimina ningún detalle de la imagen, y ahorra espacio en disco.

**Tiff (TIF):** es el formato más adecuado para trabajar imágenes profesionales destinadas a ser impresas, ya que soporta CMYK (color de 32 bits). El formato TIF soporta compresión LZW, un método de compresión que no ocasiona pérdidas, ya que no elimina detalles de la imagen. Dada la gran calidad de éste formato será el que usaré para la obtención de las imágenes del Edificio Tecnológico.

**Jpeg (JPG):** formato de gran compresión para imágenes de calidad fotográfica (color verdadero) que se utiliza principalmente en Internet. La compresión es con pérdidas (reduce la calidad de la imagen), pero la reducción del tamaño del archivo es notable. Al abrir la imagen se descomprime automáticamente.

**Gif (GIF):** formato limitado a paletas de 256 colores (8 bits/pixel) utilizado especialmente en la Web. Tiene buen nivel de compresión y ofrece la posibilidad de hacer transparente uno de sus colores. También es posible la animación de gifs mediante la inserción de múltiples cuadros en un mismo archivo.

**PSD:** El formato PSD es el que habitualmente se usa en adobe photoshop para guardar un documento el cual tiene las capas sin acoplar o cuando hemos guardado una selección, ya que este formato nos permite retomar el trabajo tal y como lo dejamos. Es un formato que no tiene pérdidas, ya que no realiza compresión alguna.

**PNG:** El formato PNG se desarrolló como alternativa al formato GIF y, como GIF, se utiliza para mostrar imágenes en World Wide Web y otros servicios electrónicos.

PNG conserva toda la información del color y de los canales alfa de una imagen, y utiliza un esquema de compresión sin pérdidas para reducir el tamaño del archivo. Al contrario que los demás formatos gráficos, el formato PNG es de libre distribución.

Si bien es un buen formato para utilizar para la Web, el problema es que los navegadores más antiguos no soportan este formato.

En ocasiones, adobe photoshop no nos permite utilizar otro formato gráfico más que PSD. Cuando nos ocurra esto, debemos comprobar si tenemos alguna selección guardada o está la imagen sin acoplar.

Otros formatos quizá menos utilizados son TGA, PCX, WMA...etc.

## **2.2. GRÁFICOS VECTORIALES**

La principal diferencia con las imágenes de mapa de bits está en el tratamiento de la información. Si un segmento se definía en un gráfico de mapa de bits como un conjunto de pixels, cada uno con sus coordenadas, en un gráfico vectorial se define como un vector por las coordenadas de sus extremos, calculando el programa el resto de puntos cuando se le pide que lo dibuje. Una ampliación en el monitor de la imagen de pixels nos mostrará una línea gruesa con las irregularidades propias de la resolución, mientras que con el método vectorial, cada ampliación necesita recalcular la línea que siempre tendrá el grosor y la perfección de la primera pantalla.

Este tipo de gráficos es ideal cuando queremos mostrar texto, logotipos, planos, etc., que requieren líneas nítidas que puedan escalarse a varios tamaños. Programas como el AutoCAD o el Corel Draw utilizan éste tipo de archivos.

### **Ventajas**

En muchas ocasiones las imágenes vectoriales requieren menor espacio en disco que un bitmap, aunque depende mucho de la imagen y de la calidad que se desee. La poca cantidad de información necesaria en un gráfico vectorial se traslada en un archivo de menor tamaño comparado con el de una imagen rasterizada de gran formato. Las imágenes formadas por colores planos o degradados sencillos son más factibles de ser vectorizadas. A menor información para crear la imagen, menor será el tamaño del archivo. El tamaño de representación de un gráfico vectorial no depende de la dimensión del objeto, por ejemplo, si se tienen dos imágenes con dimensiones de presentación distintas —una creada para tamaño carta y otra para tamaño de pliego— pero con la misma información vectorial —el mismo dibujo vectorial— ocuparán el mismo espacio en disco.

A los gráficos vectoriales se les puede aplicar zoom de forma ilimitada sin que sus bordes se afecten (solamente desde un programa que acepte su lenguaje

natural en vectores); en el caso de las imágenes rasterizadas, llega un momento en el que el zoom revela que la imagen está compuesta por píxeles.

Al aplicar zoom sobre una figura compuesta vectorialmente se puede notar que el contorno de las líneas rectas y las curvas no se engruesan proporcionalmente. Normalmente, el grosor no aumenta o aumenta menos de lo que debería, para poder mantener una mancha visualmente proporcional a la mostrada en la imagen de tamaño real. En este mismo tipo de gráficos se pueden presentar las curvas irregulares por medio de figuras geométricas simples, las cuales permitirían controlar con mayor cuidado el aumento proporcional de su grosor, para así mantener su apariencia suave original.

Los parámetros de los objetos configurados por medio de vectores pueden ser guardados y modificados en el futuro. Esto quiere decir que se pueden mover, escalar, rotar, rellenar y transformar; sin dañar la calidad ni la apariencia del dibujo. Más adelante durante el proceso de producción de la imagen, se especifican las dimensiones en un formato independiente, lo que da como resultado la mejor posible rasterización usando programas específicos para esta tarea (un ejemplo claro de este procedimiento pueden ser las imágenes que se exhiben como muestra en esta página: aunque su origen es vectorial, deben ser representadas en formatos de mapas de bits para poder ser vistas dentro de la página electrónica; esto quiere decir que si se les aplica zoom tarde o temprano se evidenciarán los píxeles, ya que son imágenes rasterizadas. Para poder hacer el ejercicio de aplicar zoom y no perder la definición de la figura se debe contar con un programa que no traduzca la definición vectorial). En resumen, se dibuja en figuras geométricas simples, se dan los parámetros específicos (color, grosor, forma) y cuando el dibujo ya está listo para ser publicado se genera una imagen rasterizada que tenga las dimensiones y el formato específico de uso que potencie al máximo la visualización correcta de la imagen.

Algunos formatos permiten animación. Esta se realiza de forma sencilla mediante operaciones básicas como traslación o rotación y no requiere un gran acopio de datos, ya que lo que se hace es reubicar los nodos base de los vectores en nuevos puntos dentro de los ejes x, y y z en el caso de 3D.

Desde el punto de vista del diseño en 3D, los efectos de sombra de los *renders* son mucho más realistas cuando se generan a partir de gráficos vectoriales, ya que las sombras pueden ser construidas con relación a los rayos de luz que las

generan. Esto permite que a partir de fotografías se puedan hacer renders e imágenes mucho más realistas.

### **Desventajas**

Los gráficos vectoriales no son aptos para mostrar fotografías o imágenes complejas, aunque algunos formatos admiten una composición mixta (vector + imagen bitmap).

Los datos que conforman que describen el gráfico vectorial deben ser procesados, es decir, el computador debe sumar todos los datos para formar la imagen final. Si hay demasiados datos se puede ralentizar la presentación de la imagen, incluso en imágenes pequeñas.

Se debe contar con un software que lo previsualice, tales como Acrobat Reader, Corel Draw, Adobe Ilustrador, Freehand, 3d Max, Maya, entre otros. Por ejemplo, los PostScript y los PDF mantienen la descripción básica en lenguajes de descripción de páginas con los gráficos vectoriales del modelo original.

Por más que se construya una imagen con gráficos vectoriales su visualización tanto en pantalla, como en la mayoría de sistemas de impresión, en última instancia tiene que hacer una traducción a sistema rasterizado.

## **3. VÍDEO DIGITAL**

El video es el formato multimedia que más memoria requiere, ya que incluye gran cantidad de información gráfica (de 15 a 30 imágenes por segundo), además de información sonora. Para hacerse una idea de la memoria necesaria, baste decir que 1 segundo de vídeo ocupa, a un tamaño de pantalla de 640x480 pixels, color de 24 bits y 30 imágenes por segundo aproximadamente 27 Mbytes.

Estos archivos de video se pueden grabar en un reproductor de vídeo si se dispone de una tarjeta capturadora. Según el tipo de dispositivo de grabación, tenemos varias velocidades de imagen (cuadros por segundo).

**Vídeo NTSC:** 30 frames por segundo.

**Vídeo PAL:** 25 frames por segundo.

**Película:** 24 frames por segundo.

Dentro de este campo, podremos encontrarnos con diferentes calidades al igual que en lo que a imágenes se refiere. Vamos a poner algunos ejemplos de los formatos más utilizados en este campo.

**AVI y AVI 2.0:** El formato AVI (Audio Video Interleave) es el formato standard de video digital que se caracteriza por mover entre 15 y 30 frames por segundo. Su funcionamiento es muy simple pues almacena la información por capas, guardando una capa de video seguida por una de audio. Cuando capturamos video hacia nuestra computadora llega en formato AVI. Puede generar archivos muy grandes y de difícil manejo.

**MPEG:** El formato MPEG (Moving Picture Experts Group) es un standard para compresión de video y de audio. Al ser creado se establecieron 4 tipos de MPEGs, MPEG-1, MPEG-2, MPEG-3 y MPEG-4. Cada uno de estos según su calidad. De aquí nace el popular formato MP3 para audio y también se habla de que el MPEG-4 que es el de mayor compresión le da vida al DivX explicado a continuación.

**Apple Quicktime (Mov):** También tiene una interesante opción nativa de los sistemas Mac. Sus archivos .mov requieren de un player especial que es el Quicktime player para visualizarlos. Este player tiene una versión sencilla gratuita y una versión profesional que entre otros permite realizar videos en dicho formato y editar algunas cualidades de los mismos.

## 4. COMPRESIÓN DE SECUENCIAS DE VÍDEO

Una vez obtenida la secuencia, habremos de darnos cuenta de que el resultado final es costoso de manejar en cuanto a tamaño se refiere; Para evitar este inconveniente, podremos elegir varias opciones:

**Crear el vídeo a un tamaño notablemente inferior al de pantalla completa.**

**Disminuir el número de cuadros por segundo, hasta un límite de 15, ya que por debajo de esa cifra, debido al fenómeno de persistencia retiniana, el ojo humano percibiría el movimiento a saltos, sin continuidad.**

**Guardar los archivos utilizando algún sistema de compresión.**

La compresión de vídeo es casi tarea imposible, conseguir que todos los ordenadores, indistintamente de la velocidad que tengan, consigan reproducir vídeo con calidad no es una tarea fácil. La compresión es el proceso mediante el cual una colección de técnicas y algoritmos reemplazan la información relativa a la posición y contenido de los pixels por una compactada descripción matemática. El mejor compresor de vídeo (codec) es aquel que logra reducir en mayor proporción el tamaño de la información de vídeo sin perjudicar la calidad final del mismo. Existe una amplia gama de codecs de compresión de vídeo, citemos algunos de ellos:

**Cinepak:** Proporciona video de buena calidad a pantalla completa; alta velocidad de muestreo y profundidad de muestra; buenas características de reproducción.

**Intel Indeo:** Permite ver videos AVI que hayan sido comprimidos bajo la nueva norma Indeo 5. Aparte de integrarse al Windows, también funciona como un plug-in para el Netscape. Su tasa de compresión es bastante buena.

**Mpeg:** MPEG (Moving Pictures Experts Group) es el acrónimo del Grupo de Expertos en Imágenes en Movimiento creado por la ISO (Organización Internacional de Estándares, International Standards Organization) para redactar los estándares para la compresión y transmisión del audio y del vídeo. El primer resultado fue el MPEG-1, que al ser una aplicación bastante limitada fue evolucionado al MPEG-2, pudiendo abarcar un ámbito de trabajo más grande, y es el que actualmente se utiliza en muchos de los medios de información.

**DivX:** DivX es un codec de vídeo basado en el estándar MPEG-4. Este codec permite comprimir un vídeo usando algoritmos de compresión con pérdida, es decir, siempre la copia comprimida es distinta byte a byte que el original, y normalmente de menor calidad (hay excepciones, por ejemplo si comprimimos un vídeo de mala calidad usando filtros de imagen puede conseguirse una copia de mejor calidad). No obstante, la pérdida de calidad en el proceso de compresión suele ser mínima si se

siguen las técnicas adecuadas, por lo que nos permite comprimir una película DVD en uno o dos CD con buena calidad.

## **TEMA 3: TRABAJANDO CON AUTOCAD**

### **1. SOPORTE**

Independientemente de que éste proyecto está destinado al trabajo en tres dimensiones, normalmente el proceso de modelado tridimensional de un edificio parte de los dibujos en dos dimensiones de las plantas, alzados y secciones. Es necesario por tanto, tener unos conocimientos básicos del manejo del AutoCAD en dos dimensiones, ya que a pesar de contar con los planos digitalizados, siempre tendremos que realizar alguna pequeña modificación en ellos. Estas modificaciones consistirán por ejemplo en eliminar entidades del dibujo (líneas, arcos, sombreados, cotas, etc) que no nos van a hacer falta para construir el modelo 3D.

La total realización del proyecto ha sido en base a planos en soporte digital y en formato .dwg de Autocad, de únicamente las plantas del edificio Tecnológico Fase-1. Para los alzados y el levantamiento de muros se ha recurrido a las mediciones pertinentes de todos los elementos modelizados en el mismo.

Una vez obtenido todo lo anterior se ha optado por hacer una colección de fotos y vídeos mediante cámara digital para obtener las texturas que posteriormente serán aplicadas.

### **2. PUNTOS A TENER EN CUENTA**

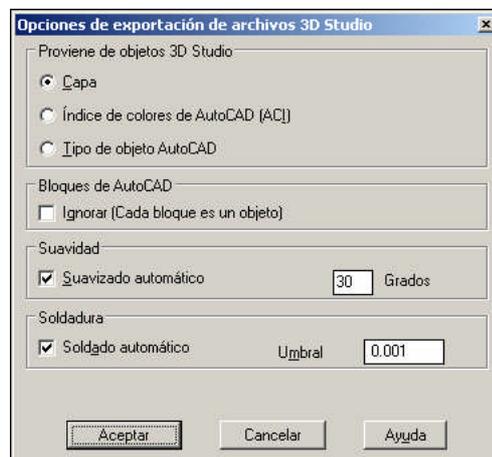
La Fase-1 del edificio Tecnológico correspondiente al módulo de la escuela de ingenierías Industrial, Informática y Aeronáutica comprende 2 pisos totalmente diferenciados.

El proceso seguido ha sido llevar a cabo el modelado de objetos en archivos diferenciados, para en pasos posteriores ir uniéndolos en base a referencias externas con el Autocad.



Como bien sabemos el Autocad y el 3D Studio Max son dos programas que se complementan perfectamente, y de lo que se trata fundamentalmente en este proyecto es de hacer uso intensivo de los dos de manera óptima, de manera que todo lo creado en el Autocad va a pasar a ser un objeto del 3D Studio para ser tratado de manera íntegra.

El uso de las capas es de vital importancia, ya que a la hora de texturizar la escena, Autocad diferencia entre tres opciones diferentes:



De esta manera, Autocad tendrá en cuenta que:

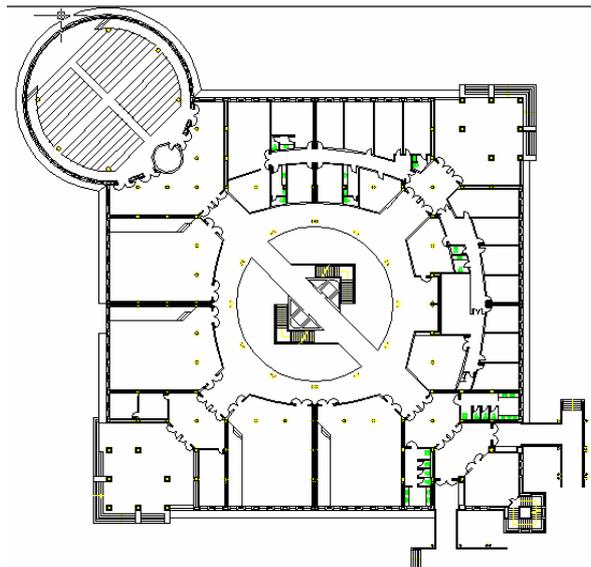
- Los objetos contenidos en una capa de Autocad pasen a formar un solo elemento en 3D Studio
- Los objetos de un mismo color en Autocad, se conviertan en un solo objeto de 3D Studio.

- Que todos los objetos iguales de Autocad, se conviertan en una sola entidad en el 3D Studio.

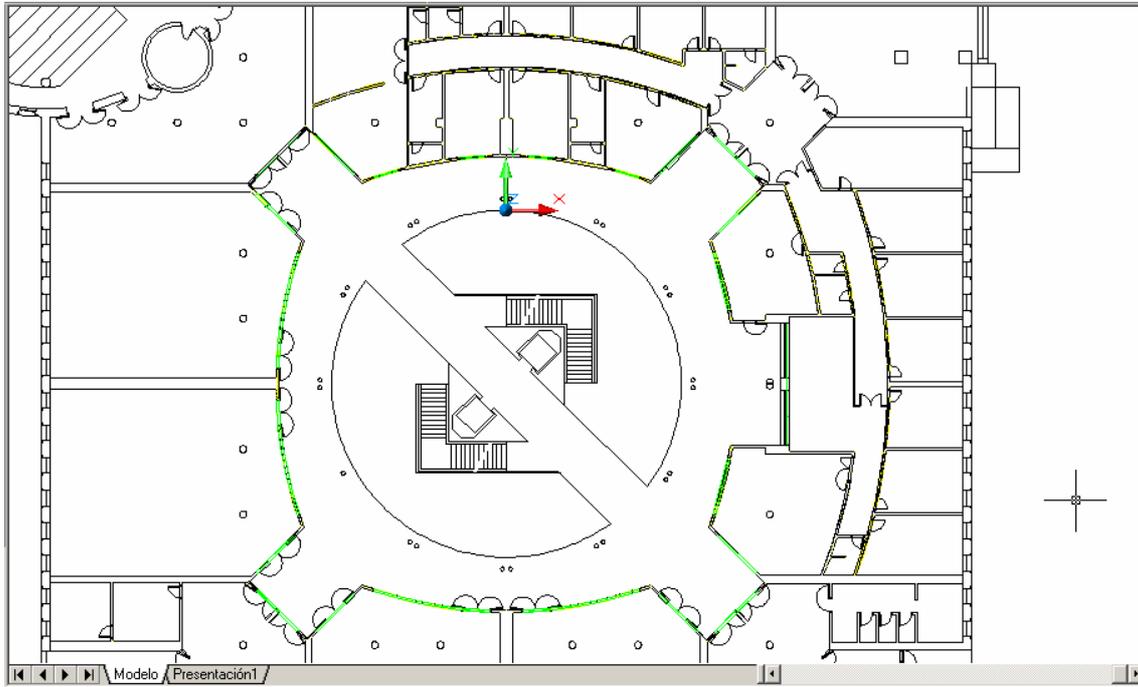
Con ello tendremos en cuenta el método a seguir diferenciando los elementos que creamos diferentes, para colocarlos en capas diferenciadas y de esta forma trabajar sin demasiadas complicaciones en el texturizado fundamentalmente.

### 3. MODELADO DEL RECIBIDOR DE LA PLANTA BAJA

Lo primero que vamos a hacer es situar en el Autocad los planos en dos dimensiones de la planta baja del Edificio Tecnológico Fase-1 que nos han sido proporcionados.



Lo que realmente nos interesa es la parte del recibidor tanto de esta planta como de la superior, así pues con el comando **Pol** dibujo un contorno de la parte que nos interese para su levantamiento en 3 dimensiones.



Las regiones que nos interesa trabajar serán extruídas para darle forma 3d; Para ello Utilizo el comando Extrusión, selecciono la polilínea del contorno que antes he dibujado y le aplico la cantidad de altura de extrusión. No es necesario que le indique eje de extrusión, ya que por defecto toma la altura en el eje Z, que es precisamente donde quiero darla tal como tengo colocado el sistema de referencia.

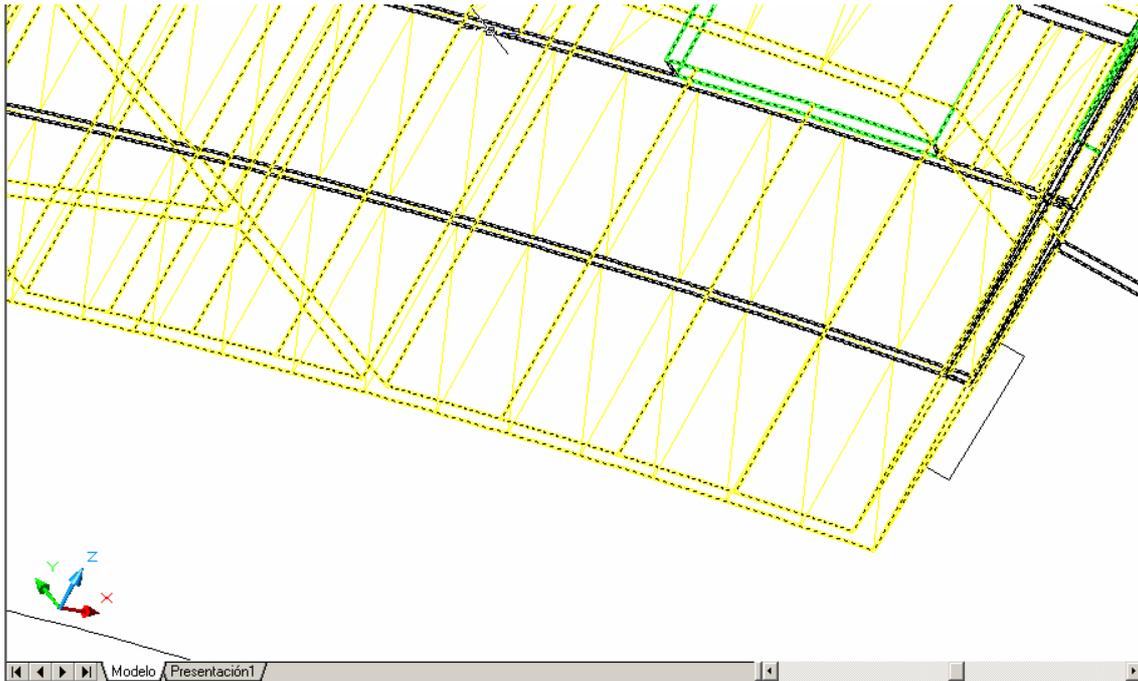
La utilización de polilíneas es bastante más útil que crear contorno con simples líneas, ya que por un lado, las polilíneas permiten crear secuencia completas de líneas sin tener que soltar el cursor y creando después una especie de región totalmente extruible.

En cambio las líneas para poder extruir un contorno, hemos de crear una diferencia entre dos regiones cerradas previamente o bien habría que convertirlas a polilínea mediante el comando Editpol.



Se ha creado una elevación de todo el contorno de los interiores para dar una visión más general del interior del edificio.

De esta forma tenemos la elevación de los muros interiores que nos interesa, pero como hemos dicho anteriormente, hemos de tener cuidado a la hora de poner las texturas más tarde. Los muros que hemos levantado tienen diferentes texturas de mármol, por lo que deberemos separar las caras que nos interese, para ello utilizaremos la opción **Modificar/ editar sólidos/ estampar**, y conseguiremos diferenciar regiones para posteriormente aplicar diferentes texturas a cada una.

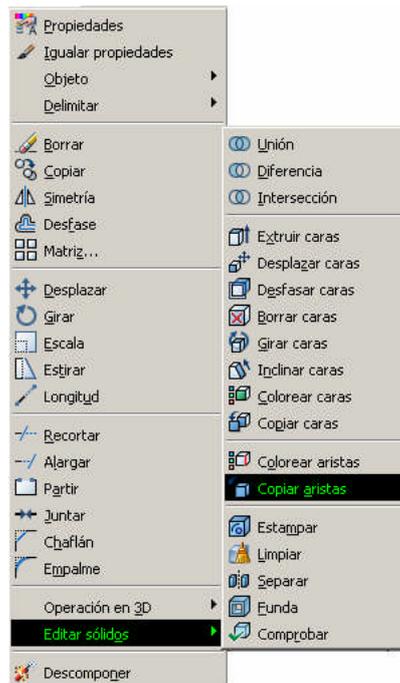


Otra herramienta que hemos seguido a la hora de estampar las líneas en cada cara ha sido la opción **Copiar aristas** para que de esta forma, nos aseguremos que las líneas son coplanares en cada punto a la parte que queramos que haga de separación.

### 3.1. HUECOS DE VENTANAS Y PUERTAS

Para ello hemos hecho las correspondientes mediciones tanto en paredes planas como en paredes de forma elíptica y se ha creado lo siguiente en cada caso:

- Para paredes planas, se ha creado un prisma con la opción **Dibujo/ sólidos/ prisma rectangular** y se ha creado una diferencia entre el sólido 3d del contorno extruído y el prisma.
- Para paredes que de alguna forma tienden a formar una elipse o semicircunferencia, se ha optado por utilizar la opción **Modificar/ editar sólidos/ copiar aristas** y de esta forma crear una región del tamaño de la ventana para extrusionarla posteriormente.



Para alinear los prismas para crear el hueco de la ventanas y las puertas, se ha optado por el comando **Alinear**, que nos pedirá que especifiquemos puntos de origen y destino para situar el prisma.

Una consideración importante a la hora de manejar modelos que presenten geometría curva, es el configurar de manera adecuada la variable del sistema **Facetres** que por defecto está establecida con un valor de 0.5 pero que si queremos que el visualizado sea óptimo (con la carga adicional de trabajo que ello supone) deberemos aumentarlo a su valor máximo de 10.

Esta variable Realiza un ajuste del suavizado de objetos sombreados y modelizados, y de objetos cuyas líneas ocultas se hayan eliminado; es vital para la exportación posterior al 3D Studio.

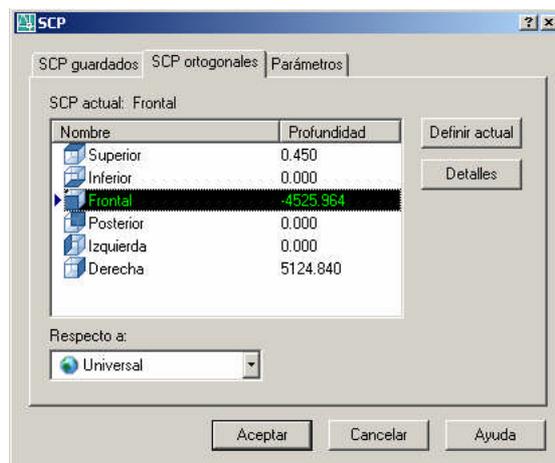
### 3.2. CORNISAS Y TECHOS

El proceso de creación de la cornisa que se une al techo de separación de las dos plantas se basa en la creación del perfil previo de la misma una vez hechas las medidas correspondientes.

Para ello me he ayudado de polilíneas y curvas con la secuencia de comandos **Pol** y **Arco** hasta obtener el perfil que más adelante extruiremos a lo largo de una circunferencia.

Una parte importante en la creación de estos elementos es el manejo casi constante del sistema SCP. Para extruir el contorno puedo hacerlo de dos formas:

- Utilizando la ventana del **SCP** en la opción de menú herramientas/SCP ortogonal/Predefinido. Seleccione una orientación para que el eje de coordenadas sea paralelo a la sección de la cornisa.



- Utilizando el comando **SCP** que da diferentes opciones para establecer el sistema de coordenadas.

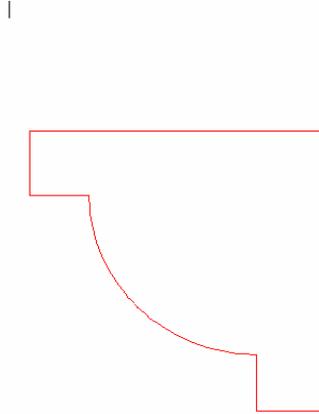
```

Comando:
Comando: scp
Nombre de scp actual: *FRONTAL*
Indique una opción [Nuevo/DEsplazar/Ortogonal/PRev/Rest/Guardar/Supr/Aplicar/?/Univ] <Univ>:

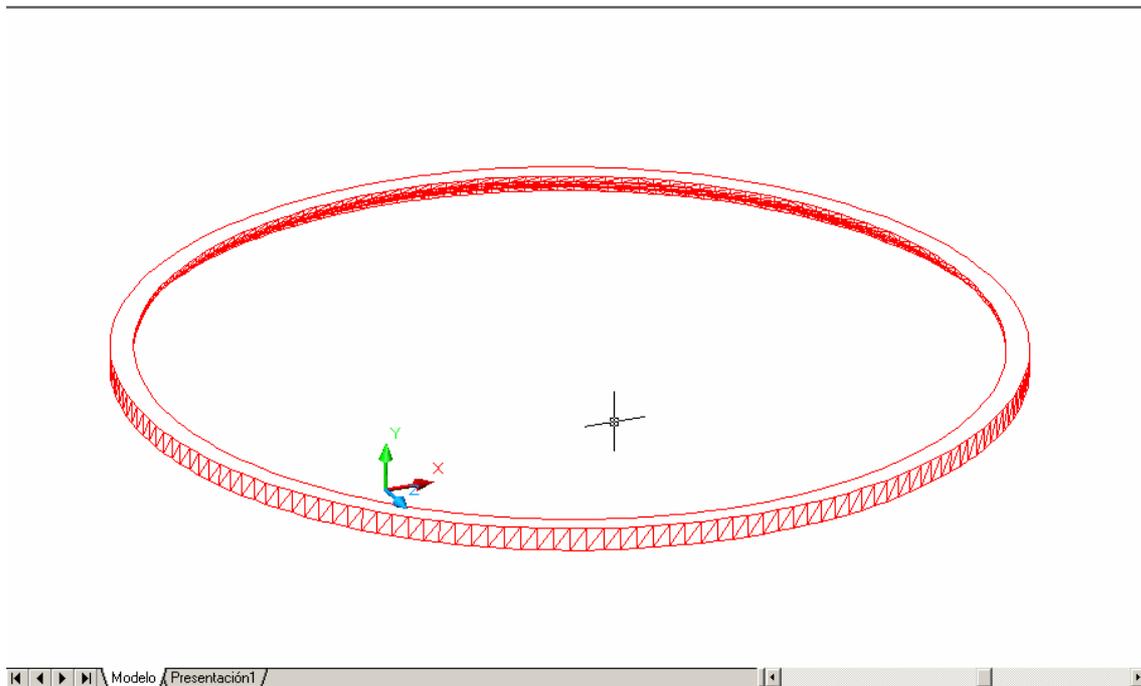
```

Aquí elegiremos la opción **Nuevo**, 3 puntos refiriendo un punto para abscisas otro para ordenadas y punto en plano XY.

El contorno sería de la forma:

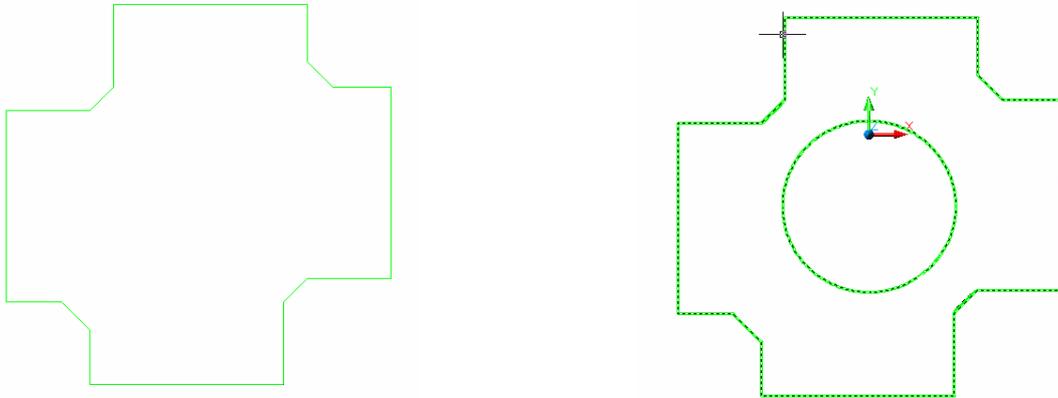


Obtenida la sección extruiremos a lo largo de una polilínea curva (circunferencia) sin cambiar para nada el sistema de coordenadas y eligiendo como eje de extrusión la curva, teniendo como resultado final la cornisa.



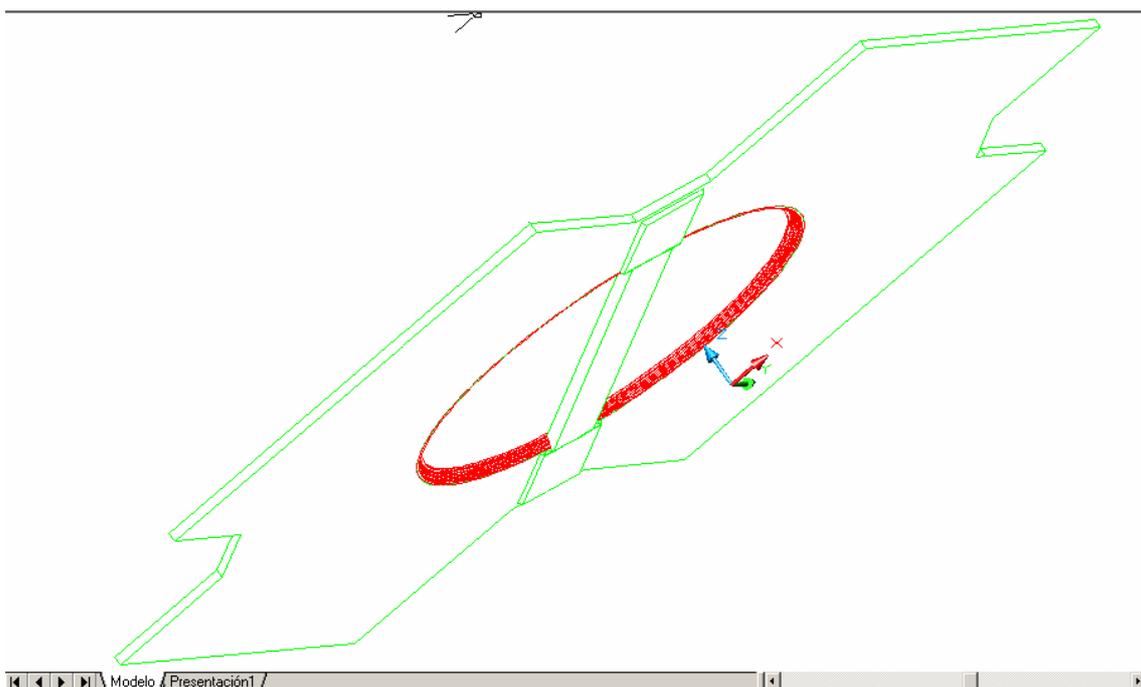
Con la cornisa creada, ya tenemos delimitada la zona para levantar el techo que nos servirá de separación entre las dos plantas.

Para ello creamos el contorno que nos interesa a base de una polilínea y diferencia de una circunferencia, que después de extruir ambos dejará a la vista la cornisa anteriormente creada.



Fijándome ahora en la parte inferior del techo, observo, que no es uniforme sino que presenta salientes que se inclinan con cierto ángulo para dar sensación de relleno y más estética.

Este efecto lo conseguimos creando prismas rectangulares e inclinando sus caras alrededor de  $40^\circ$  (de forma aproximada, ya que no se ha dispuesto de ninguna ayuda para su medición).



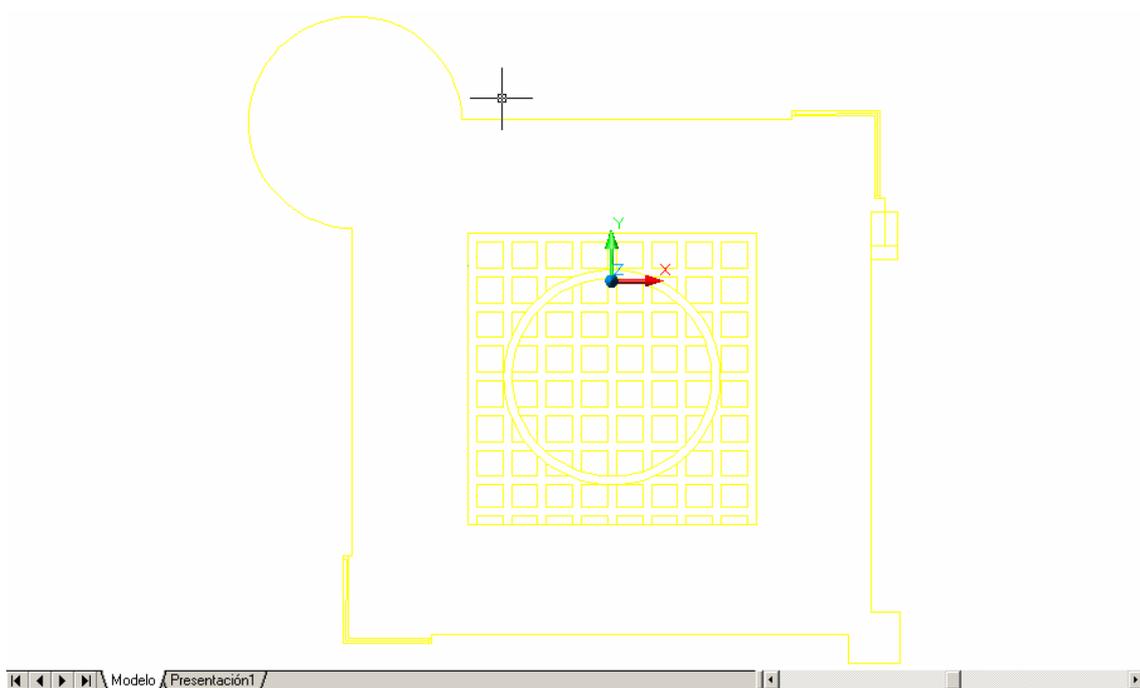
Las partes de los salientes que presenta la estructura son simétricas, para ello hacemos uso del comando **Simetría** estableciendo como línea de la misma, un eje perpendicular al pasillo central.

### 3.3. BASE INFERIOR

Para la base utilizo el mismo procedimiento de creación del contorno con polilíneas. Obtenido éste, se ejecuta el comando extrusión con una distancia como hasta ahora, previamente medida. Como de costumbre, para el uso de este comando en tal caso, no es necesario darle un eje de extrusión, ya que como norma general, el SCP universal sitúa como eje perpendicular el Z que es la referencia para extruir.

Hay que tener en cuenta que la planta baja de este edificio presenta un mosaico de texturas en su base, con diferentes gamas de mármoles formando contornos geométricos.

Si queremos que a la hora de texturizar la base nos resulte menos engorroso el proceso, deberemos separar las caras que llevan diferentes materiales; para ello desde Autocad estamparemos las líneas y arcos pertinentes para lograr caras texturizables de una manera relativamente fácil.



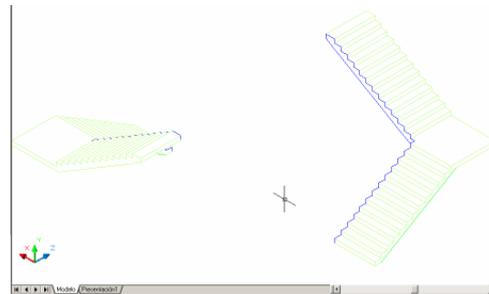
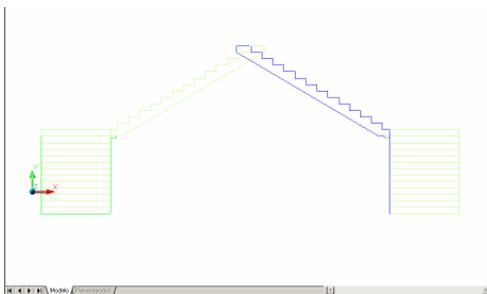
El mosaico central se ha obtenido mediante la creación previa de líneas formando la “parrilla” y circunferencias. Para el resultado visto anteriormente he optado por la opción **Recorta** y así evitar que las líneas intersequen unas con otras.

**Modificar/ editar sólidos/ estampar** hace que seleccionando cada línea anterior se creen nuevas caras en el conjunto.

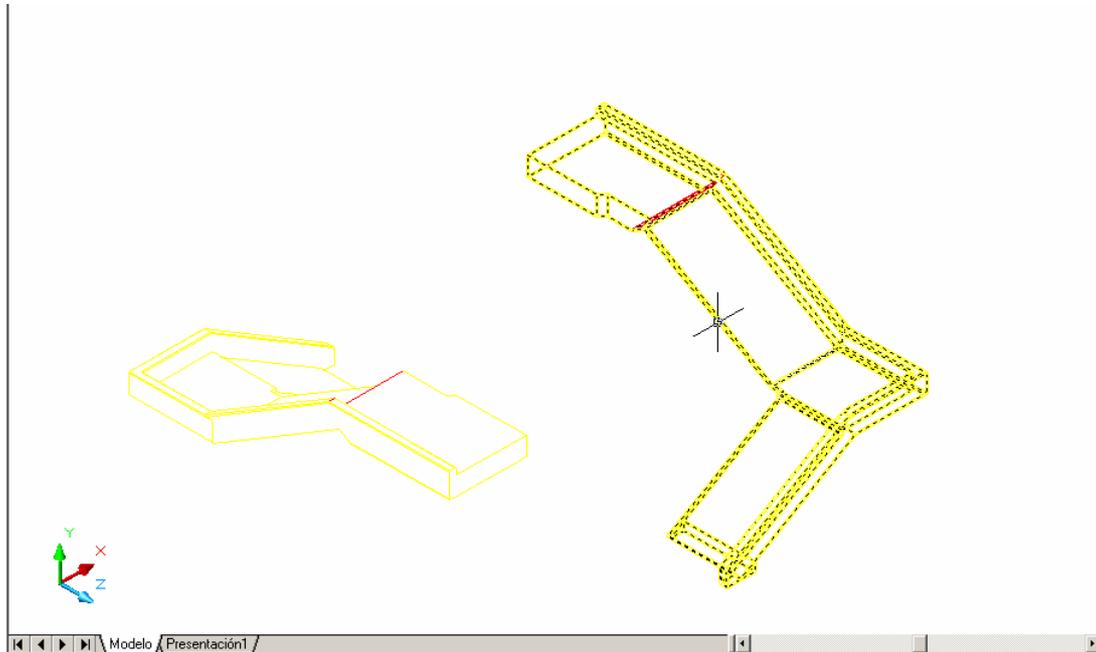
### 3.4. ESCALERAS Y CUBRE ESCALERAS

El modelado de las escaleras podríamos considerarlo de tres partes:

1. Se crea el perfil de las escaleras que queremos dimensionar posteriormente. Para ello, debemos tener en cuenta que son escaleras dobles, es decir, dan la vuelta a una columna central, con lo cual haremos uso del comando **SCP** para alinear el dibujo a cada plano que más tarde queramos extruir.
2. Una vez hecho lo anterior, el siguiente paso es darle profundidad al conjunto, para ello recurrimos otra vez al comando **Extrusión**, diferenciando como hemos dicho antes, cada plano por separado.
3. Hacemos simetría del conjunto realizado, de tal manera que las escaleras queden perfectamente colocadas.



Con las escaleras dispuestas, procedemos a la creación del cubre escaleras, de igual forma he creado el perfil del mismo y le he dado profundidad. No vamos a olvidar que en el proceso seguido hasta ahora ha sido medido manualmente cada elemento para así conseguir una exactitud muy aproximada.



He tenido alguna dificultad en la creación de estos dos elementos pues de alguna forma se situaban mal en el conjunto; para solventar este contratiempo, me he ayudado de los comandos **Gira3d** y **Desplaza** de tal manera que la posición final sea tal cual se puede ver en los planos de la planta proporcionados.

El elemento cubre escaleras, tiene achaflanadas varias de sus aristas para evitar los cantos vivos y dar mejor estética al conjunto; si ejecutamos el comando **Chaflán** conseguiremos el suavizado de esquinas.

### 3.5. COLUMNA CENTRAL

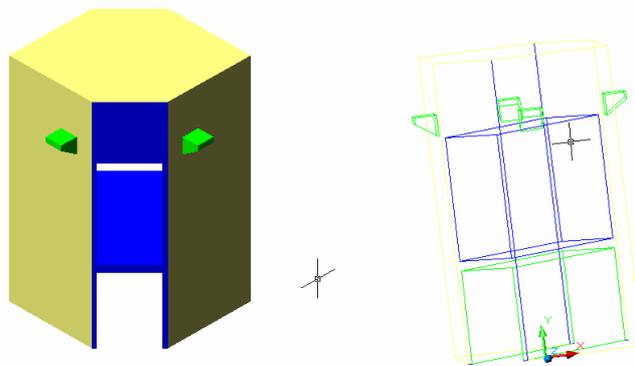
El edificio de la escuela de ingenierías tiene en su centro una columna que actúa como “pilar maestro” y que sustenta la mayor parte del peso de la estructura superior.

Para su diseño se ha optado por hacer su contorno mediante la ya citada polilínea y hacer una extrusión como de costumbre.

Se ha de tener cuidado pues, alrededor de esta estructura se van a desarrollar gran parte de los elementos, así que convendrá que su dimensionamiento sea lo más preciso posible.

A la hora de su elaboración, habrá que darse cuenta de lo siguiente:

- Huevo inferior para el paso a su través.
- Huevo superior para el paso en el pasillo central.
- Huevo adicional para el conjunto de tuberías que guardan circuitería y otros dispositivos.



Los cuatro soportes en forma de trapecios piramidales, se han situado **Alineando** los mismos al centro de cada cara lateral y posteriormente **Desplazándolos** debidamente hasta alcanzar la altura deseada.

Su creación ha sido en base a la sección de que consta, con su posterior extrusión. Creamos una simetría de cada par y tenemos los 2 pares perfectamente alineados.



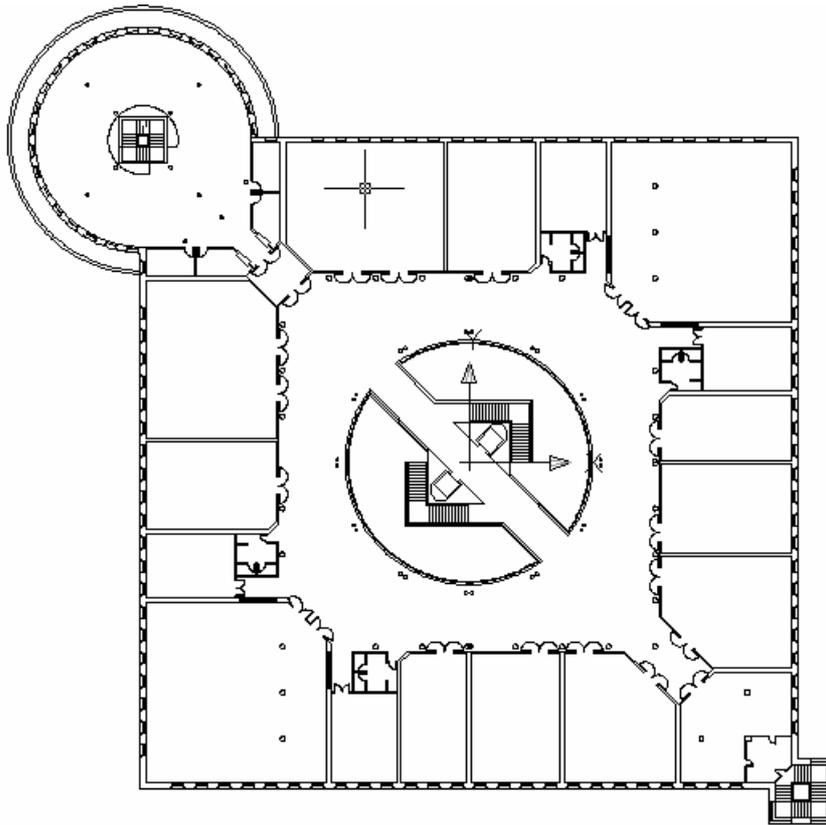
## 4. MODELADO DE LA PLANTA SUPERIOR

Para su modelado, lo que haré será colocar el plano en planta del que me he provisto e iré haciendo lo mismo que en un principio con la parte de abajo.

Con el comando polilínea, haré el contorno necesario para levantar las paredes de esta parte.

Esta parte la vamos a realizar en un archivo diferente para no sobrecargar el conjunto al trabajar con el otro archivo.

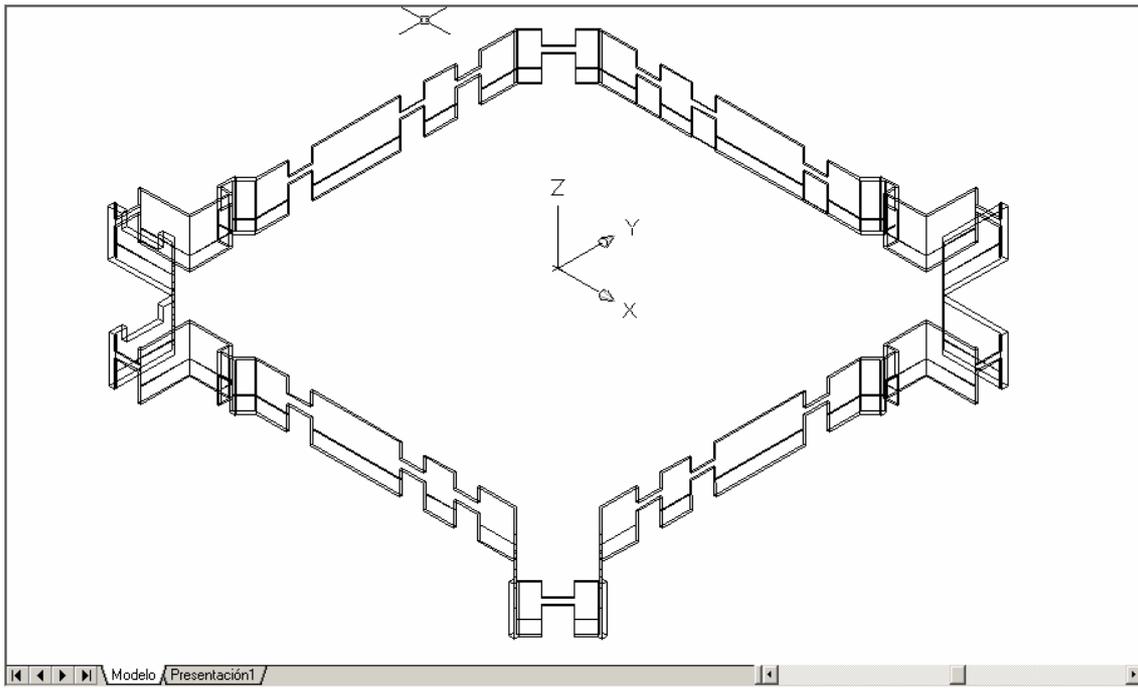
Cada capa contendrá objetos del mismo tipo.



Ahora mediante el comando Extruir, elevo ambas polilíneas la distancia que hay entre el suelo del forjado de la planta baja y el suelo del de la la planta. Esa distancia se obtiene haciendo la diferencia que hemos tomado con las mediciones manuales. Al extrusionar, obtengo dos sólidos "macizos", por tanto tengo que "vaciar" el sólido creado por la polilínea exterior, para que aparezcan los muros. Para ello utilizo el comando Diferencia, seleccionando primero el sólido que deseo sea del que se reste (el creado por la polilínea exterior), pulsando "Intro", y seleccionando después el sólido a restar (el creado por la polilínea interior). Seleccionar los dos sólidos en orden inverso no crearía el resultado deseado.

Por otra parte y siempre con vista al editado en el 3D Studio Max, la paredes de la estructura presentan diferentes materiales. Por medio de la creación de líneas a la altura correspondiente, y su posterior **Estampado** diferenciaremos las caras que queremos representar.

De esta forma obtenemos el sólido deseado.



Tengo que hacer ahora los huecos en los muros donde irán alojadas las ventanas. Para ello dibujo una polilínea sobre el plano en planta que siga el contorno del hueco de las ventanas. Extrusiono la polilínea la altura que tengan las ventanas midiendo, y mediante el comando **Desplaza**, coloco el sólido creado a la altura que tiene respecto al suelo. Con la ayuda de otro comando como es **Alinear** consigo que los huecos se alojen perfectamente.

Realizo copias pertinentes de los sólidos ayudándome con **Gira3d** para colocarlos en sus respectivos sitios, y finalmente hago la diferencia de los sólidos, restando los prismas obtenidos al muro levantado.

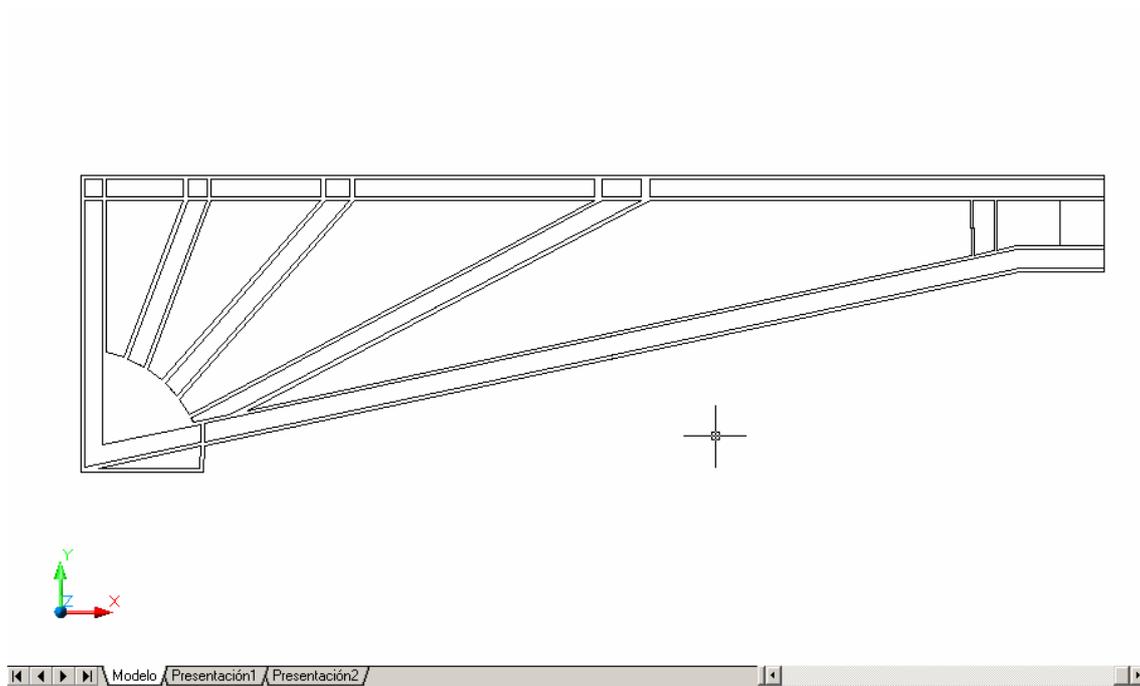
#### 4.1. CORNISA INTERIOR DE LA PLANTA SUPERIOR

Lo que haremos con este elemento al ser idéntico al de la planta inferior será copiar y pegar asignándole un punto base y otro de desplazamiento.

De esta forma incluiremos el objeto en la planta superior con un mínimo esfuerzo.

## 4.2. SOPORTES SUPERIORES

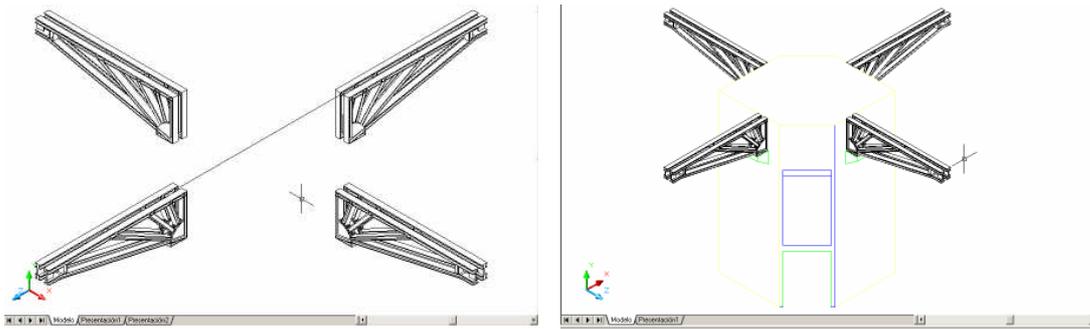
Se han creado a base de la unión de líneas y regiones. Primero se ha creado un contorno y posteriormente se han ido diferenciando las regiones interiores para poder quedarme con la región única que será extraída.



Para la elaboración de los cantos de las vigas metálicas he optado por usar el comando **Editar sólidos/ Estampar**, de esta forma consigo caras diferentes en un principio en el modelo 3d obtenido.

Estas caras resultantes, serán desfasadas en base a una dirección ya dada. Para ello ejecutamos **Editar sólidos/ desfasar caras** y escribimos una distancia de desfase para ello.

El número de soportes es de 4, que lo haremos por medio de una simetría a pares por medio de una línea de apoyo una vez que tengamos el primero. El aspecto que presentará dicho conjunto será el siguiente.

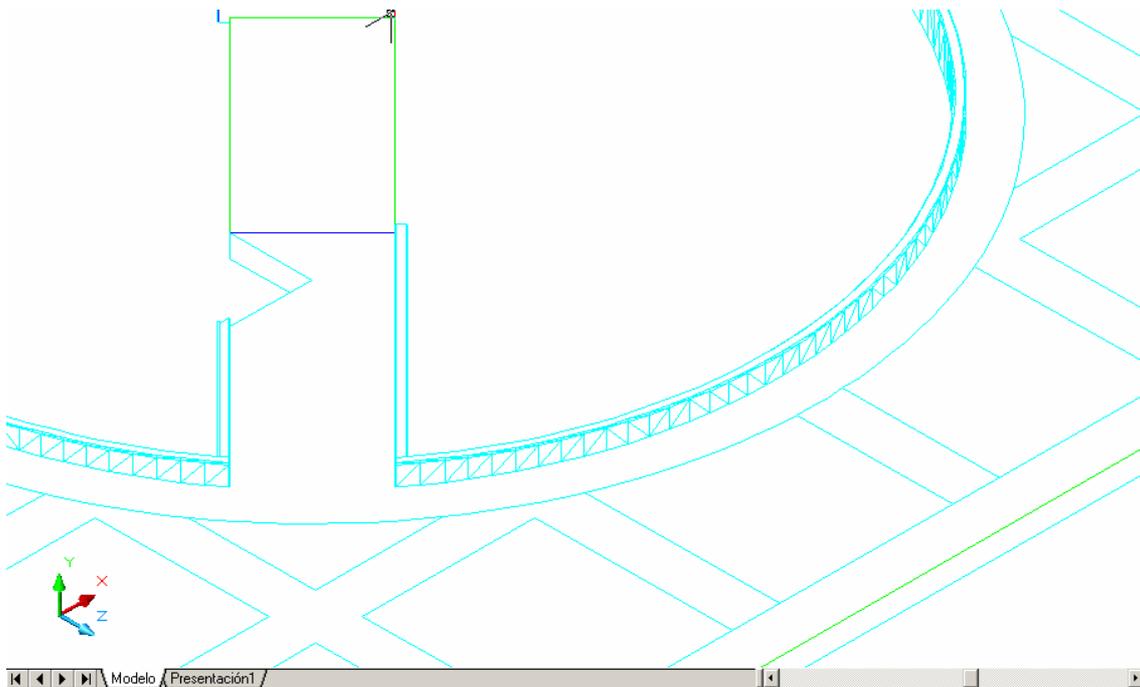


### 4.3 BASE DEL PISO SUPERIOR

Para su realización se ha optado por hacer los contornos con polilíneas desde el plano en planta que tenemos. Una vez hecho esto y en base a nuestras mediciones extruímos normalmente con el grosor deseado.

La parte del círculo central se ha creado como una diferencia con un cilindro desde el mismo centro del que parten los planos bidimensionales.

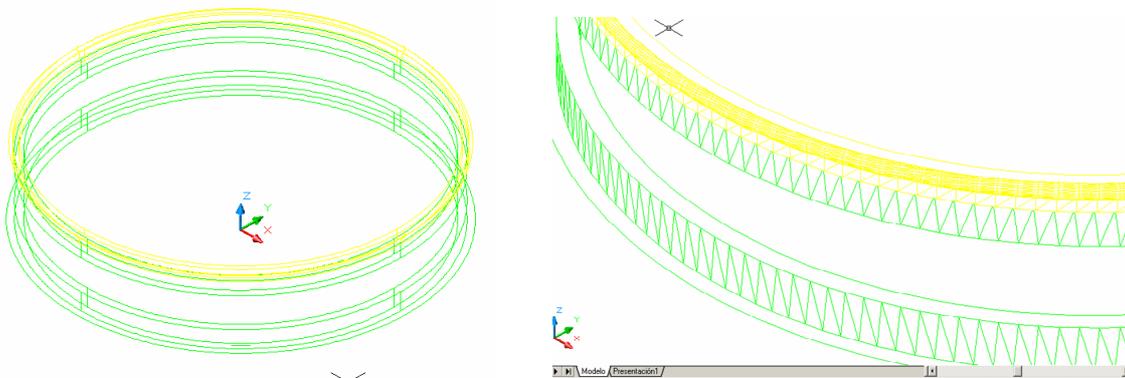
Se sabe a su vez que parte de la estructura circular presenta un saliente en forma de aro. Pues bien, como hemos hecho hasta ahora, por medio del comando **Editar sólidos/ estampar** crearemos una nueva cara para posteriormente desfasarla y conseguir el saliente deseado.



Una vez tengamos la estructura, procederemos al estampado de las regiones a las que posteriormente aplicaremos diferentes materiales.

#### 4.4. CORNISA EXTERIOR Y MURO

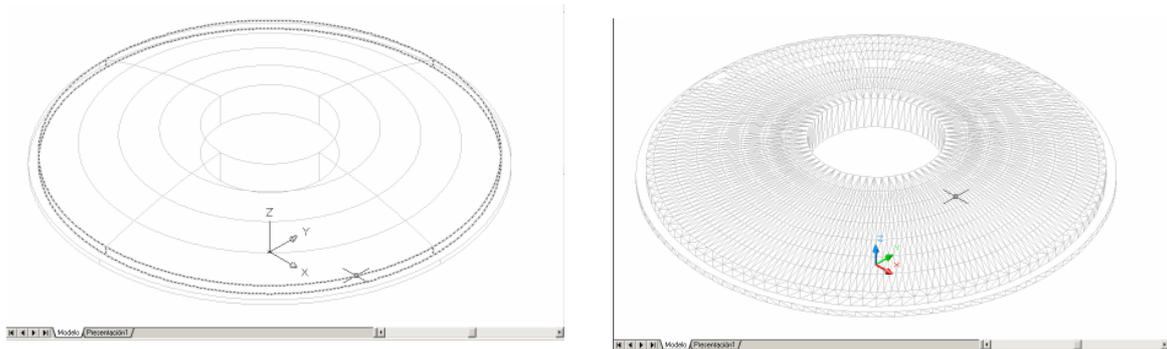
Para hacer las cornisas de los muros circulares, dibujo la polilínea del perfil de la cornisa, que es la misma de la forma circular de la planta baja. La coloqué en su lugar y creé un sólido de revolución mediante el comando Revolución, indicando como eje de revolución una línea vertical en el centro del círculo y seleccionando giro de 360°.



#### 4.5. CUBIERTAS

Las cubiertas de las dos zonas circulares son diferentes al resto, ya que tienen forma semicircular y llevan un material distinto, por tanto las incluyo en una nueva capa. Para modelar la cubierta de la zona circular más exterior al edificio, creé una región circular que hará de base (Región) y sobre ella otra base cilíndrica (círculo extruído), tal como aparece en el plano de sección del edificio. La cubierta es una bóveda semiesférica. Para dibujarla, creé un perfil con la forma de media sección de la bóveda, al que le apliqué una Revolución de 360° alrededor de un eje vertical en el centro de la bóveda.

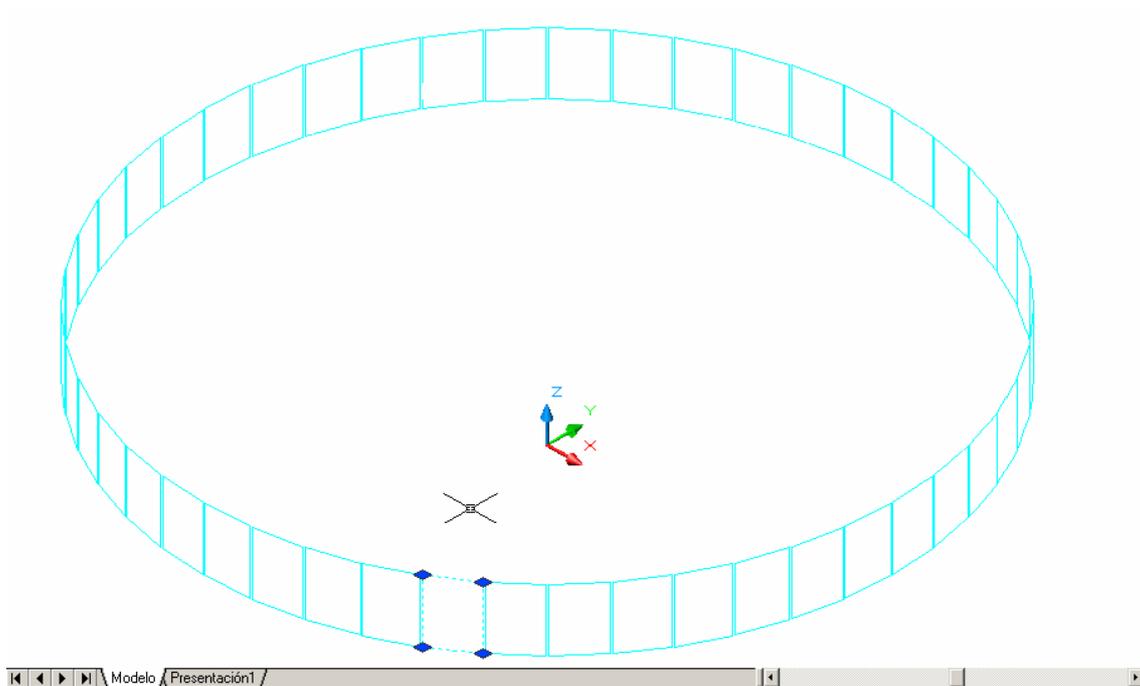
Practicaremos una diferencia entre el sólido de la semiesfera y un cilindro que dimensionaremos para obtener el agujero en el centro.



#### 4.6. VENTANAS SUPERIORES

Las ventanas de la forma circular central están dispuestas alrededor del círculo de la planta. Para dibujarlas utilizo como apoyo el plano 2D donde aparece la carpintería. Primero divido la circunferencia exterior de la carpintería en tantas partes iguales como tramos de ventana tiene, y lo hago mediante el comando **Divide**. Luego, dibujo uno solo de los tramos de ventana y lo dibujo entre dos divisiones de la circunferencia. La ventana la modelo mediante el comando **Cara3d** para así ahorrarnos carga en el dibujo, ya que el aspecto exterior no es primordial en este caso.

Posteriormente creo el resto de ventanas, posicionando el **SCP** en el centro del círculo y creando una **Matriz** polar de 48 elementos cubriendo un ángulo total de  $360^\circ$ .



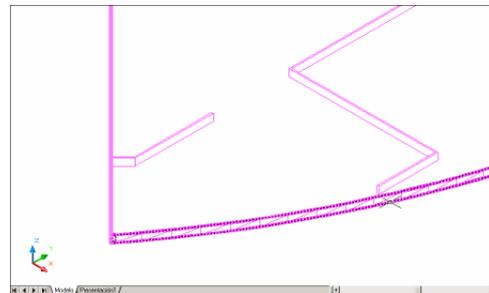
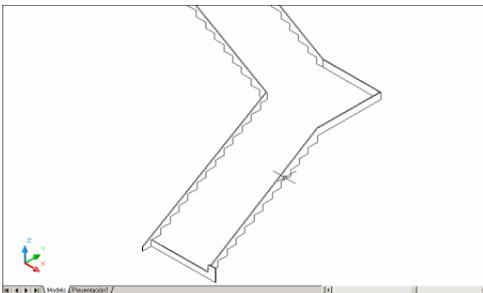
#### 4.7. RODAPIÉS

Para su creación hemos dispuesto de una rebanada que formará su sección y posteriormente con la ayuda de una polilínea que defina el contorno sobre el cual va a ser extruído formaremos el recorrido de todo el rodapié. De esta forma definiendo como eje de extrusión la polilínea, hemos colocado este elemento decorativo.

Para el caso del rodapié en el piso superior, el cual es de forma circular en dos semicircunferencias, se ha usado un arco de polilínea, copiando una arista de la estructura a la que va ir pegada el rodapié y convirtiéndola en arco de polilínea, que igualmente servirá de eje de extrusión.

Los que van adheridos a las escaleras, basta con hacer el perfil a una altura determinada por medio de la siempre útil polilínea y extruir sin más.

A continuación veremos la forma final que adoptan.



#### 4.8. MODELADO DE ELEMENTOS COMUNES

A continuación vamos a proceder a realizar una tabla con los elementos comunes que se han modelado en este proyecto.

Se ha tratado de conseguir el mayor detalle posible ya que la representación trata del interior del recibidor del Edificio Tecnológico Fase-1, por lo tanto he decidido que cada uno sea un archivo diferente para no tener tanta saturación de líneas en el espacio modelo.

<b>ELEMENTO</b>	<b>PISO</b>	<b>NÚMERO DE CAPAS</b>
Puerta acceso clases	Ambos	8
Puerta WC	Ambos	4
Puerta de entrada	Inferior	7
Puerta acceso vestíbulo	Inferior	5
Puerta Secretaría	Inferior	6
Puerta reuniones alum.	Superior	7
Ventana simple	Superior	3
Ventana Curva Cuádruple	Inferior	3
Ventana Cuádruple	Inferior	3
Ventana Curva Doble	Inferior	3
Ventana Triple	Inferior	3
Ventana Triple	Superior	3
Ventana Curva Triple	Inferior	3
Ventana Múltiple Conserj.	Inferior	4
Radiador 70 cm.	inferior	1
Radiador 90 cm.	Superior	1
Banco	Ambos	6
Papelera	Ambos	5
Barandilla	Ambos	1

Vamos a describir algunos de ellos, pero primero vamos a recalcar cuatro puntos importantes:

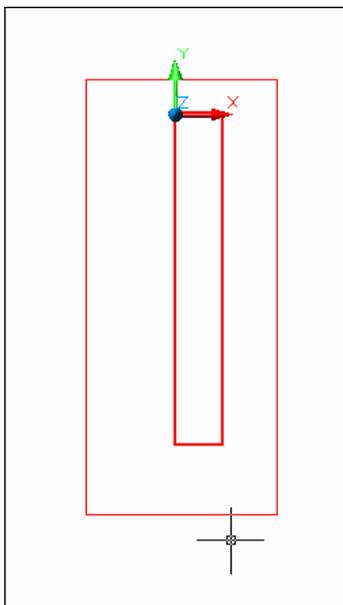
- Cada objeto es un archivo diferente
- No se incluyen elementos en la capa **0** que trae por defecto Autocad, para no crear confusión posterior en el 3D Studio.
- Es de vital importancia que cada archivo tenga la variable del sistema **Facetres** al máximo, es decir con el valor 10, ya que de lo contrario, las caras de los elementos curvos se exportarían en un proceso posterior de forma facetada, lo cual es estéticamente perjudicial.
- La unión posterior se realizará mediante referencias externas, usando **Refx**, ya que los elementos son de origen dispar y variado.

### Puerta de acceso a clases

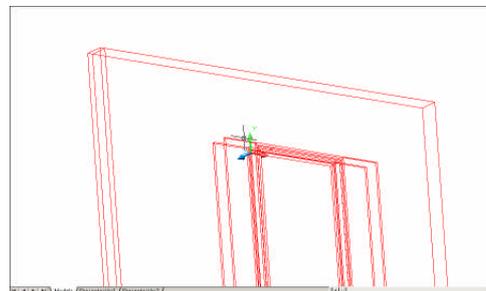
Para la elaboración de la misma, se ha creado un rectángulo de las medidas de la misma, el cual formará el marco al que se le harán modificaciones para obtener un resultado lo más próximo a la realidad.

Una vez obtenido el rectángulo extruímos la región y la duplicamos a fin de tener un sólido con el que poder hacer diferencia para alojar la ventana.

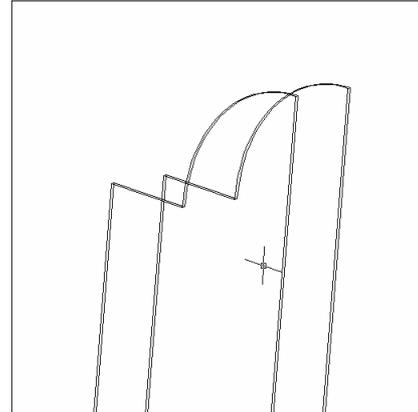
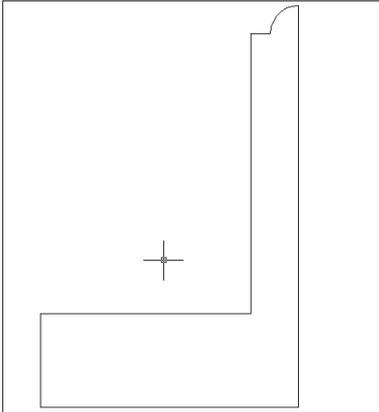
Con la ayuda de la secuencia **Modificar/ Editar sólidos/ Desfasar caras** encogeremos esta copia hasta obtener el tamaño del cristal, para posteriormente aplicar una diferencia.



Pasamos ahora a la creación del estampado de lo que pasará a ser la madera. Los cantos que bordean al cristal están hundidos cierta cantidad; para no tener que crear más regiones y hacer su posterior diferencia, he optado por usar el comando **Estampar** y una vez hecho, **desfaso** las caras, con lo que consigo el resultado deseado.

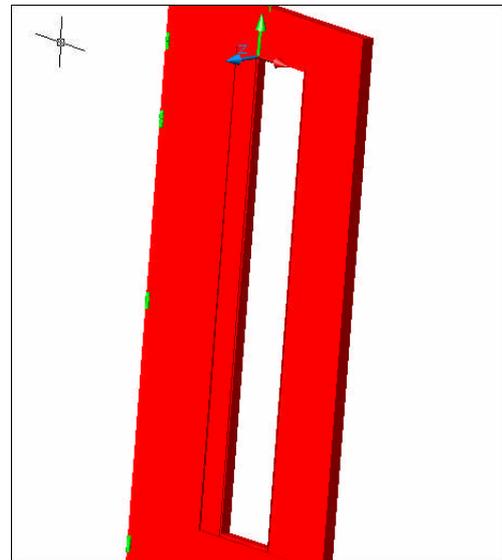
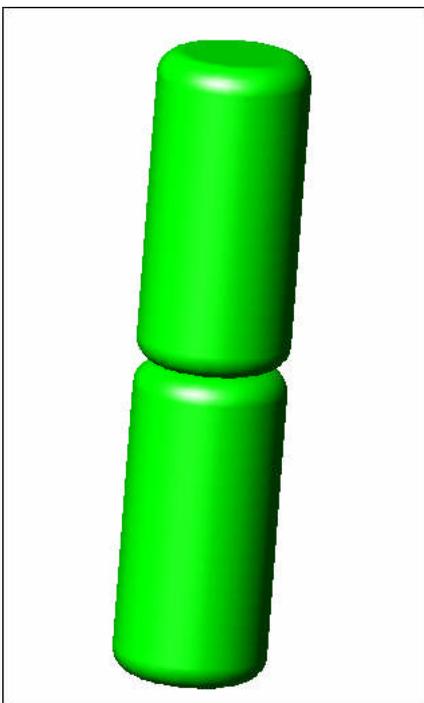


Para los embellecedores color oro, se ha calcado el proceso anterior, es decir, creamos una región que tenga su forma por medio de polilíneas y arcos de polilínea y se le da profundidad como hasta ahora se ha hecho con el comando **Extrusión**.



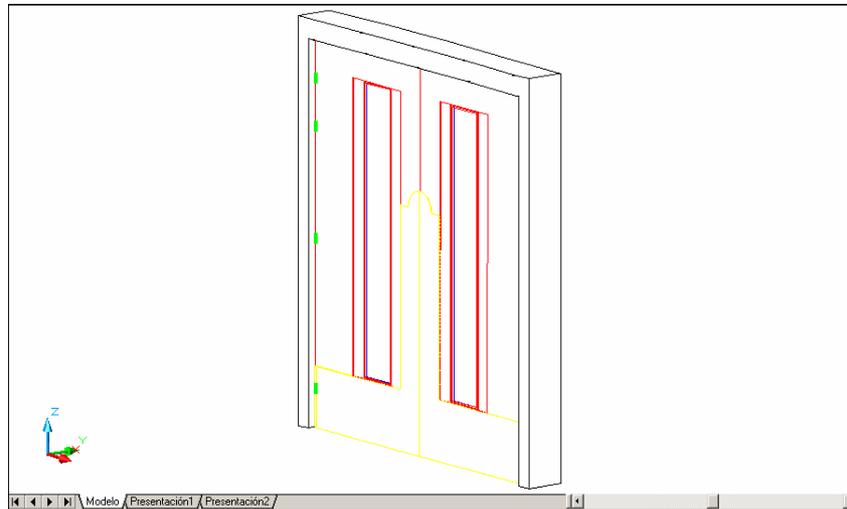
Para la creación del cristal se hará un rectángulo de profundidad el grosor del mismo y con la ayuda de referencia a objetos se colocará en el medio del hueco creado para este fin.

Como no, las bisagras, que con el uso de la primitiva cilindro achaflanando los cantos y usando simetría conseguimos la forma que han de tener. Posteriormente se agrupará en bloques para que cada puerta lleve incorporados el número exacto de ellas de manera fácil.

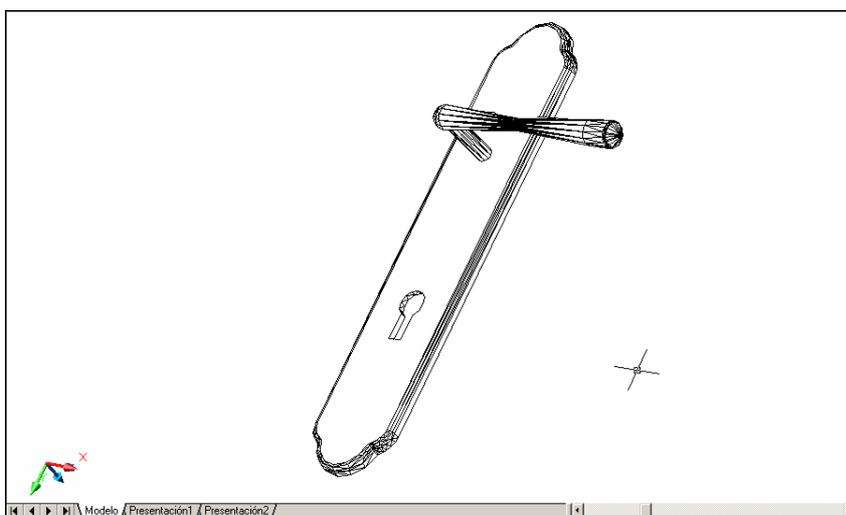


Sólo quedaría incorporar el marco para que el conjunto encaje perfectamente en el hueco realizado en los muros interiores. Pues bien, volvemos otra vez a la

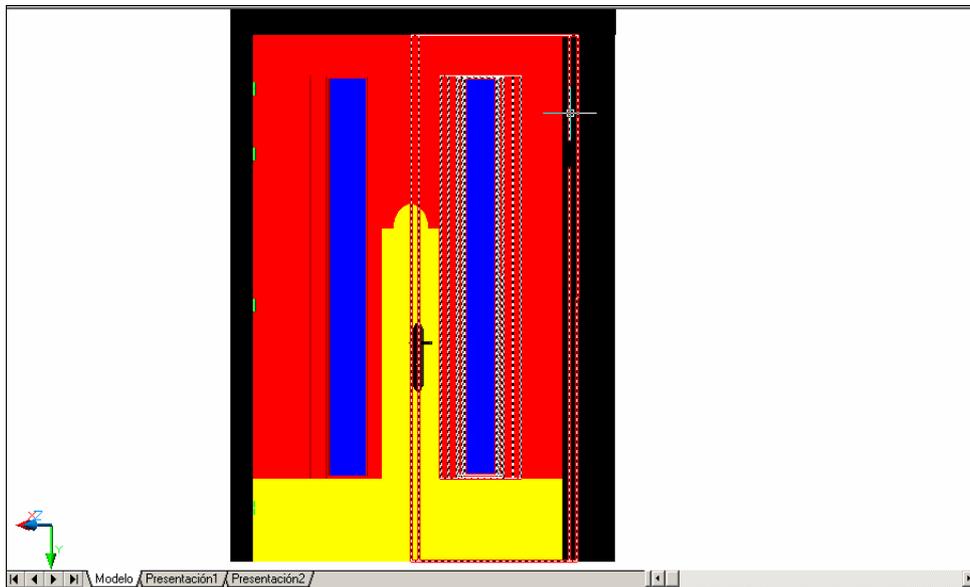
polilínea y creamos un contorno para dar forma al marco, extruímos asegurándonos que tenemos salientes por ambos lados para que la puerta no sea coplanar con el marco y duplicamos la puerta obteniendo lo siguiente.



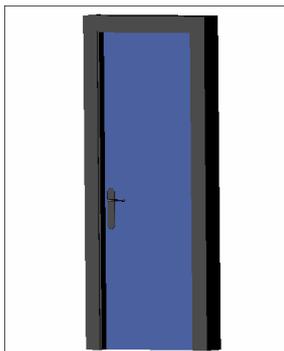
Por último la manilla de la puerta que se incorporará al conjunto posteriormente. Su creación ha sido por medio de tangencias. Usando polilíneas y arcos de polilíneas, obtengo el contorno de la manilla, y con el comando **Revolución** hacemos un sólido girando  $360^\circ$  el contorno creado alrededor de un eje (línea). El embellecedor de la cerradura se ha hecho como resultado de la extrusión de la región que le da forma. Una vez obtenido todo, se procede a suavizar los cantos mediante empalmes, lo cual hará el conjunto más atractivo a la vista.



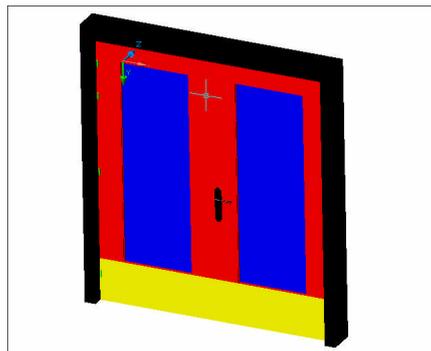
Vamos a mostrar el resultado final de la puerta que servirá como elemento común en ambos pisos.



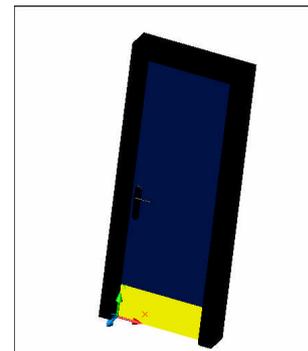
De la manera que hemos procedido con la puerta de acceso a clases, hacemos las restantes puertas; no he descrito el proceso, puesto que es muy similar al anterior pero en cada caso diferente; tan sólo veamos los ejemplos correspondientes.



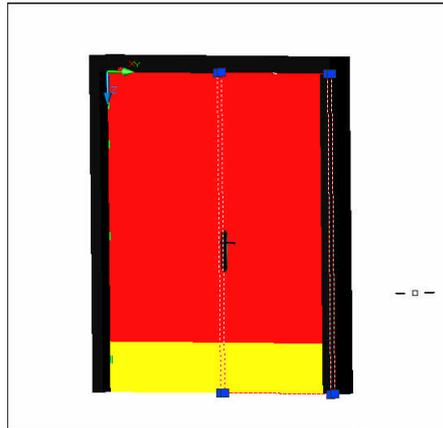
***Puerta WC***



***Puerta de entrada***



***Acceso vestíbulo***



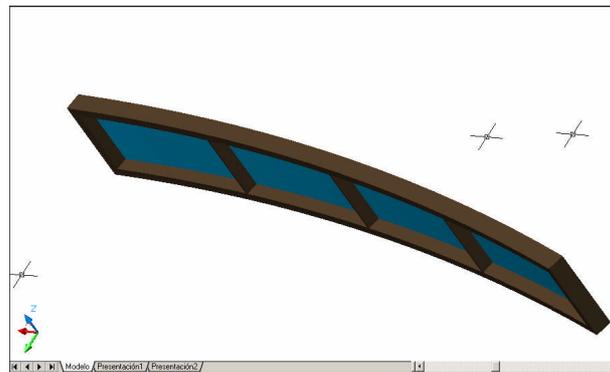
***Puerta secretaría***

### **Ventana curva cuádruple**

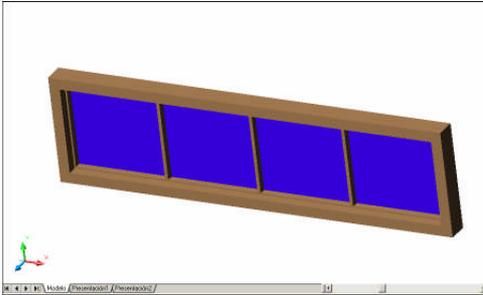
Corresponde a un elemento del piso inferior. Lo que hemos hecho para su creación ha sido ayudarnos con la creación de una primitiva que tenga su forma. Para ello ha sido necesario copiar una cara del hueco perteneciente a esta ventana desde la capa “interiores”, pasarla a un archivo diferente y hacer el sólido desfasando la cara. A partir de ahí obtenemos un bloque macizo y curvo de la forma del hueco, lo que servirá como marco.

Como toda ventana, ha de tener un cristal, en esta caso cuatro. La división la efectuaremos con el comando **Divide**, en una arista del bloque que previamente habremos copiado.

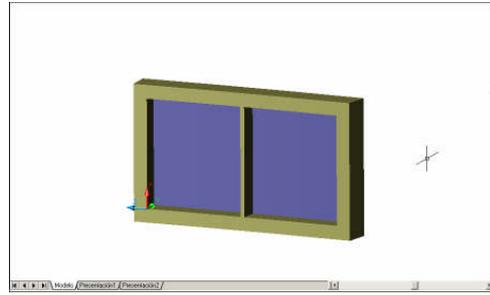
Cada división la unimos mediante líneas y creamos rectángulos de la profundidad del hueco de la ventana que nos servirán como separadores. El proceso es bastante laborioso, pero conforme avanzamos obtenemos el resultado dispuesto incluidos los cristales ya.



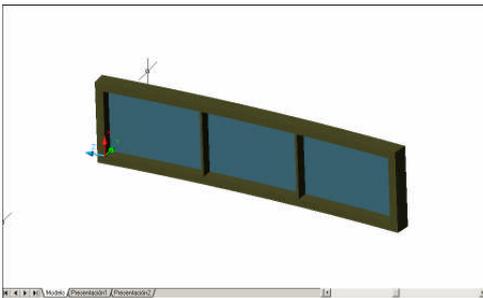
El proceso de creación para las demás ventanas sigue una pauta común, por ello vamos a exponer los ejemplos restantes sin describir el proceso que sería una réplica de lo hasta ahora expuesto.



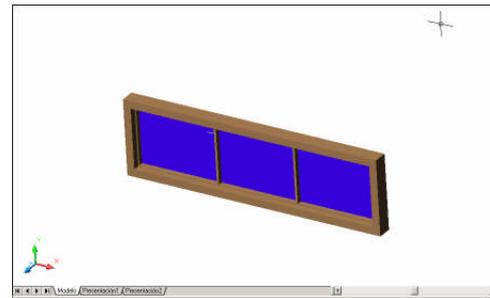
***Ventana recta cuádruple***



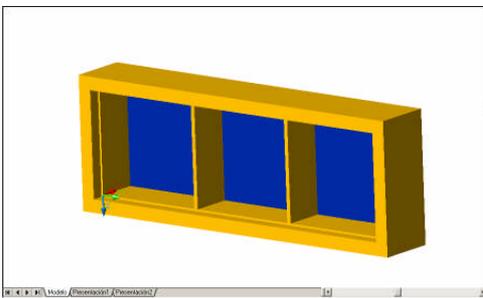
***Ventana curva doble***



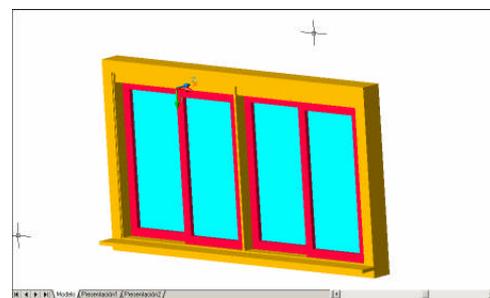
***Ventana curva triple***



***Ventana recta triple***



***Ventana triple superior***

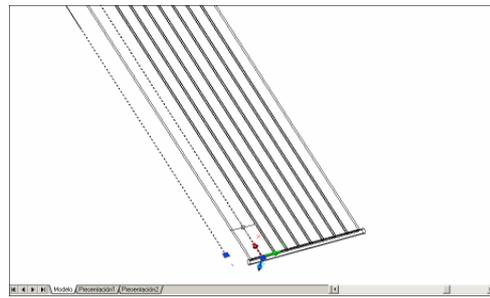
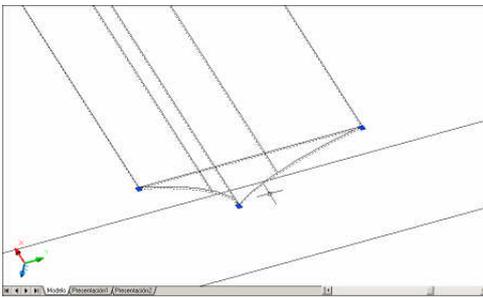


***Ventana múltiple conserj.***

## **Radiadores**

Su elaboración se basa en las primitivas prisma rectangular y cilindro; de tal forma que el prisma rectangular formará el cuerpo del mismo y los cilindros los extremos.

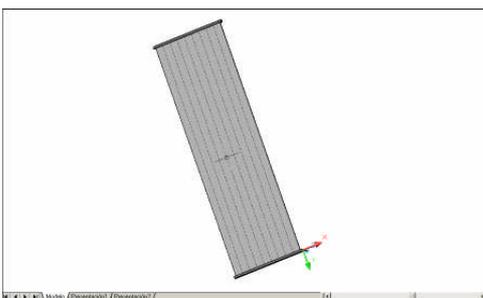
Los elementos radiantes de los que se forma, han sido diseñados con tangencias de líneas y arcos de polilíneas, creando una región y extruyendo el elemento a lo largo del cuerpo del radiador.



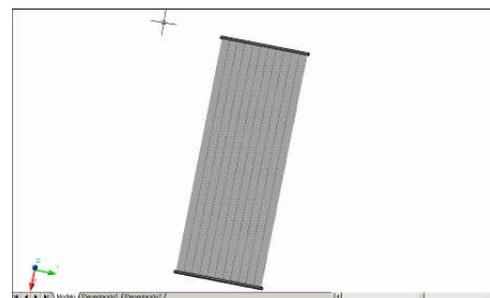
Una vez tengamos los elementos que nos sirvan de separación, los colocaremos de forma exacta a lo ancho del cuerpo del radiador, para ello nos ayudaremos de una línea que se habrá dividido convenientemente conforme al número de elementos que queramos crear.

Llegado a este punto, haremos una simetría por la otra cara y crearemos la diferencia en el cuerpo del radiador.

Para los soportes haremos copias de la primitiva cilindro a ambos lados de la parte superior y uno solo en la parte inferior. El aspecto que presenta el conjunto después de hacer la unión de los elementos es la siguiente.



**Radiador 70 cm.**

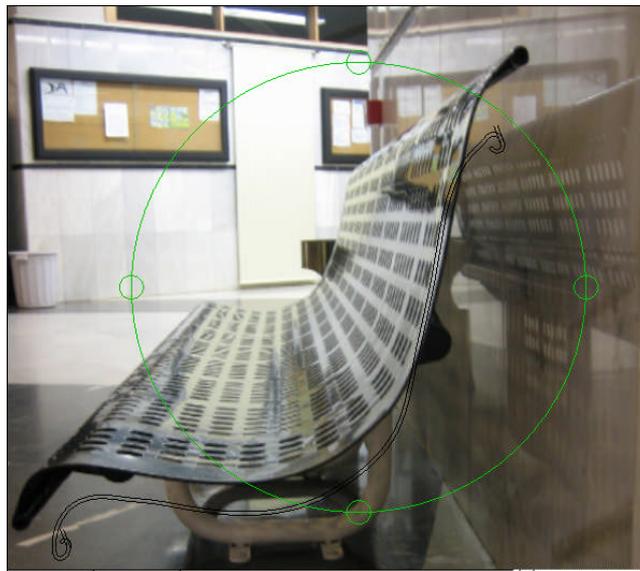


**Radiador 90 cm.**

## **Banco**

Para elaborar este elemento he tenido que ayudarme de numerosas fotos para poder hacer una toma válida del perfil del mismo, pues es un elemento de difícil medición al presentar un contorno totalmente curvo.

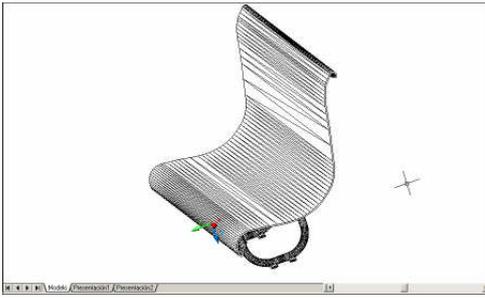
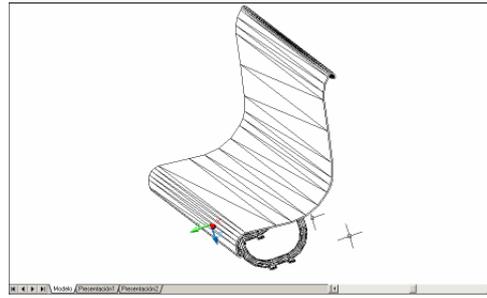
Autocad dispone de la opción insertar imagen (**Insertar/ Imágenes ráster**) dentro del espacio modelo, y eso ha sido lo que he realizado para dimensionar el perfil del objeto.



Lo que conseguimos con esto, es una vez hechas las mediciones aproximadas, ayudarnos de la imagen para realizar los perfiles del banco. En la imagen se puede ver el perfil en negro (desplazado levemente para su mejor visualización) que se ha diseñado siguiendo los trazos de la imagen. Con ello conseguimos una región perfectamente extraíble que nos dará la parte más notable del banco.

A partir de aquí el proceso es simple, los apoyos se crearán con segmentos y arcos de polilíneas unidas que servirán de eje para extruir un círculo perpendicular a ellos. De igual manera procedemos con los soportes traseros.

Es muy importante como ya hemos dicho al principio de este capítulo que la variable **Facetres** esté en su valor máximo de lo contrario perderemos calidad en el suavizado de las curvas.

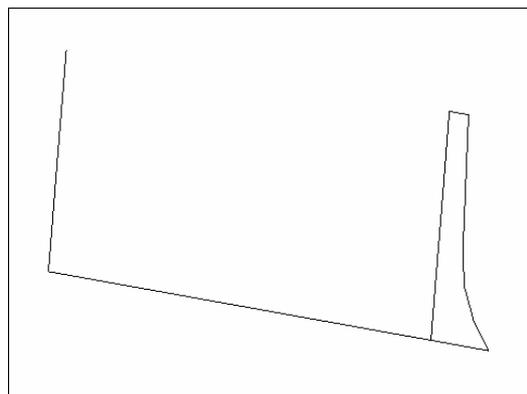
**Modelo Facetres = 10****Modelo Facetres = 0.5**

### Papelera:

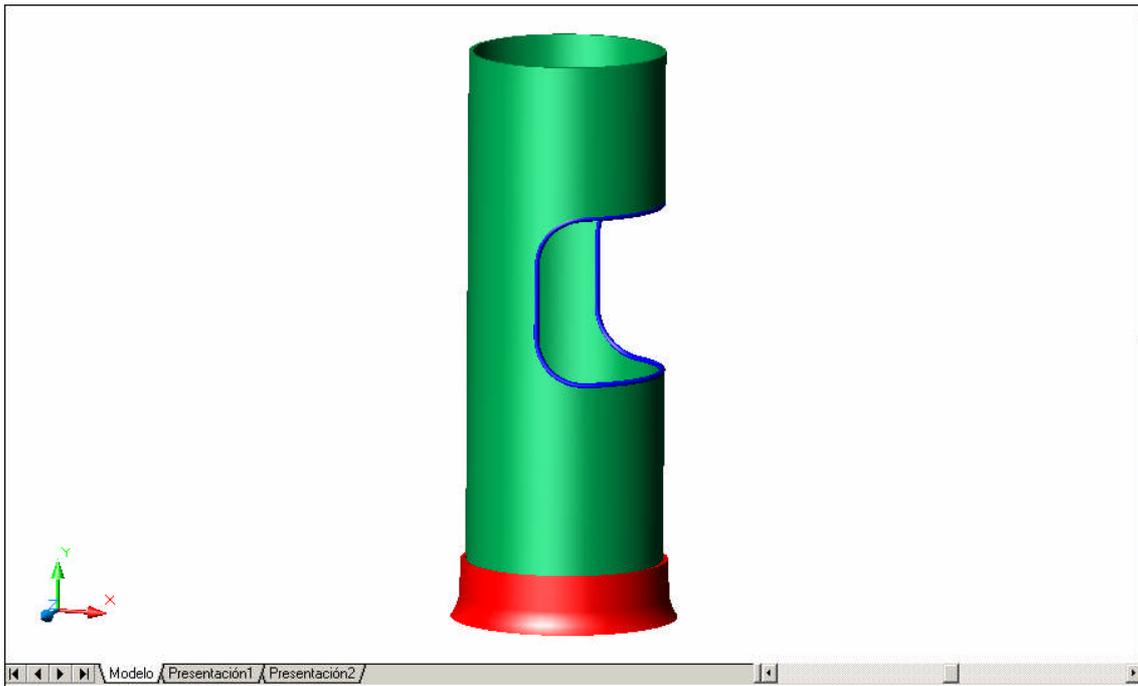
Consta del cuerpo de metal, el embellecedor de goma mate negra y la base de plástico.

He creído oportuno usar la primitiva Cilindro con la altura de la papelera, para después practicarle un vaciado con el comando **Funda**, excluyendo la tapa de arriba, de esta forma obtenemos el cuerpo de la misma.

Para la base formamos una región y un eje a cierta distancia para su posterior revolución.



Practico un corte en el cuerpo de la papelera con la forma deseada mediante diferencia de un sólido 3D previo y añado el embellecedor que creo mediante una polilínea que recorra una de las aristas del hueco. Para ello diseño una rebanada que tendrá el perfil del embellecedor y como eje la polilínea anterior, obteniendo el objeto final.



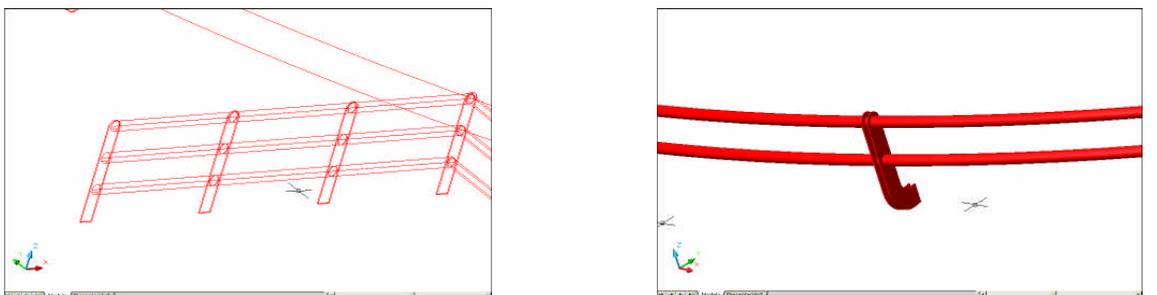
### **Barandillas:**

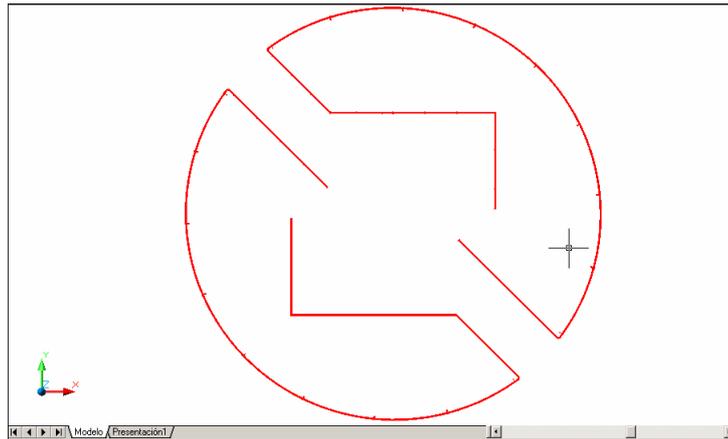
Partimos de segmentos de polilínea a lo largo del recorrido de lo que va a ser toda la barandilla desde las escaleras inferiores hasta el piso de arriba.

Es importante situar el SCP en su plano XY perpendicular al recorrido de lo que va a ser la barandilla. Una vez hecho este paso situamos tantas circunferencias como elementos extruídos queramos tener.

Los elementos de apoyo serán diferentes en el caso del tramo superior curvo. El ángulo de cada semicircunferencia del piso superior son aproximadamente  $163^\circ$ , son lo que practico una matriz polar para los apoyos obteniendo el conjunto.

Una vez obtenido todo, mediante el comando **Unión** hago que los elementos pasen a ser un solo sólido.





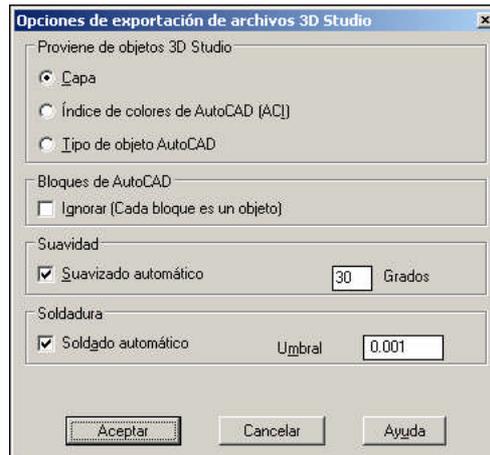
#### 4.9. EXPORTAR ARCHIVOS

Una vez modelado todo el conjunto, procedemos a exportar los archivos resultantes.

Autocad tiene compatibilidad total con el 3D Studio Max como ya hemos dicho, y como tal nos da la posibilidad de exportarlo de dos formas diferentes:

- Se podrán exportar como archivos con formato de 3D Studio, que tienen por extensión 3ds, para posteriormente importarlos en 3D Studio Max.
- Guardar los archivos como hasta ahora hemos venido haciendo, con extensión dwg de AutoCAD para posteriormente en 3D Studio Max importar los archivos como en el caso anterior.

Desde mi punto de vista, sería recomendable importar los archivos al 3D Studio con un formato previo de \*.3ds, si bien cualquiera de los dos tipos de archivo es perfectamente manejable, 3D Studio Max importa los archivos 3ds necesitando menos polígonos que con los archivos dwg, y dado el alto volumen de caras del dibujo me he decantado por importarlos con formato 3ds. De cualquier forma, el formato no tiene demasiada importancia siempre que la importación se haga correctamente, es decir, que no queden en el camino caras que tengamos que dibujar otra vez en 3D Studio Max. Los archivos de AutoCAD, los exporto uno a uno al formato 3ds mediante la opción Exportar del menú Archivo.



Ya tendremos los archivos en formato \*.3ds, el hall del primer piso y el correspondiente al segundo piso; seguidamente comenzaremos la edición en el 3D Studio Max.

## TEMA 4: PROCESO DE TRABAJO CON 3D STUDIO MAX

### 1. INTRODUCCIÓN

El siguiente paso después del trabajo con Autocad se basa en el texturizado de la escena. Con esto quiero dar a conocer que nuestro cometido en este entorno no es más que una vez obtenido el modelado sólido del conjunto, darle un cambio radical de imagen a nuestro proyecto a fin de plasmar de la forma más fehaciente la realidad en nuestra escena por medio de materiales, luces y otros elementos o efectos.

### 2. PROCESO DE IMPORTADO DE ARCHIVOS

Es lo primero que habremos de hacer para que el trabajo previamente hecho se pueda visualizar en nuestro nuevo entorno.

Para ello una vez arrancado el programa 3D Studio Max 6, ejecutamos la siguiente ruta:

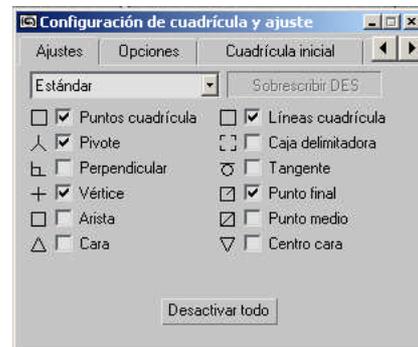
**Archivo/ importar** y se nos abrirá un cuadro de diálogo para que designemos el archivo a tratar.

Seguidamente, tenemos dos opciones:

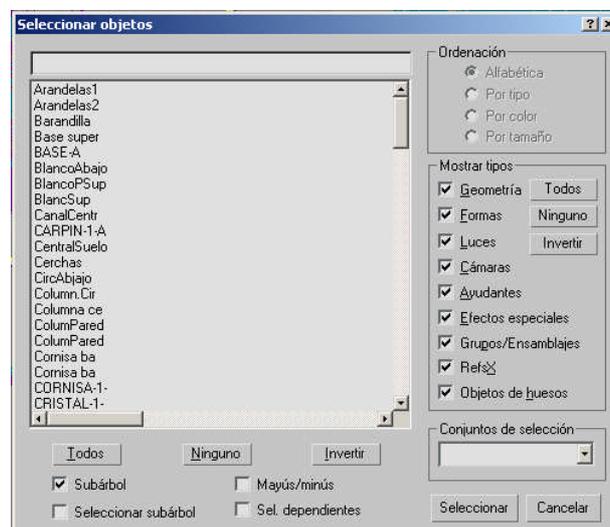


Como la escena que tenemos es del todo nueva, en este caso nos daría igual cualquiera de las dos opciones, yo he elegido **fusionar objetos con la escena actual**. El manejo de cualquier objeto se hace de forma muy similar que con el Autocad, de tal forma que nosotros para una mejor ayuda dispondremos en todo momento de la

referencia a objetos, de una rejilla en espacio 3D y de varios visores, todo ello totalmente configurable.

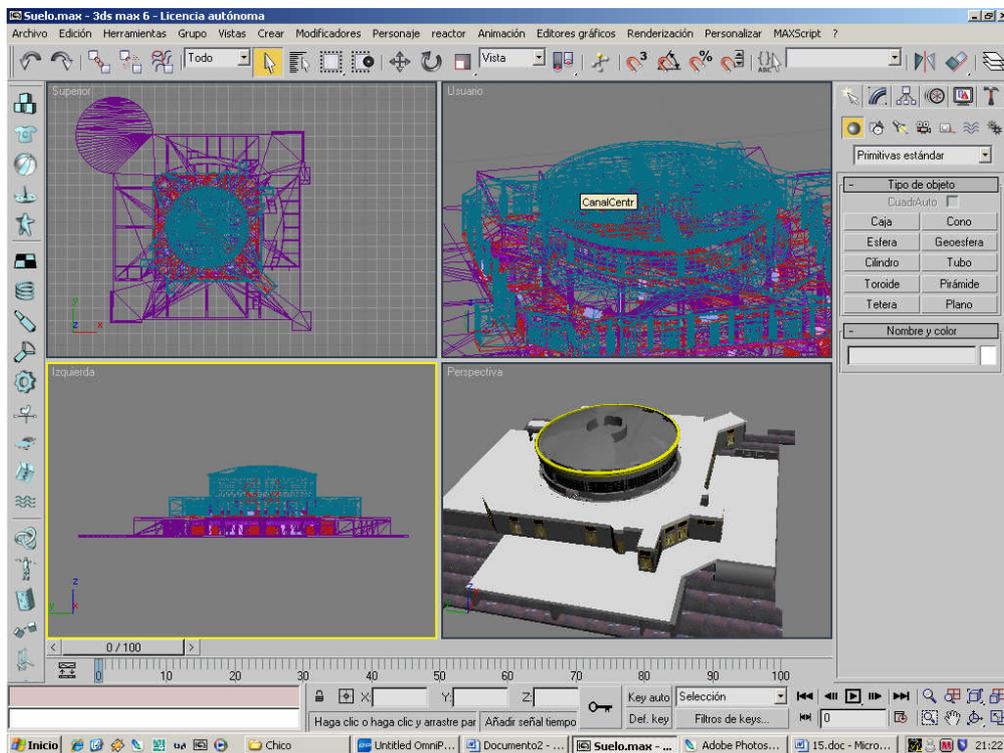


Con los objetos en escena, podremos tener absoluto control de todas las capas que la forman, de forma que haciendo **Edición/ seleccionar por/ nombre** veremos una lista de las capas que conforman el archivo. Cabe resaltar, que el nombre que en su momento se dio en el Autocad, se mantiene en este programa, excepto aquellos nombres que puedan provenir de referencias externas y resulte largo, a lo que 3D Studio responde abreviándolo.



Si se desea, se pueden agrupar los objetos utilizando la opción Agrupar del menú Grupo, para poder seleccionar más cómodamente. Una vez hecho esto guardamos el archivo, damos un nombre y el programa lo guardará con extensión \*.max, que es la propia del 3D Studio Max.

La forma que va tomando el proceso en el 3D Studio se muestra a continuación.



Vemos como hemos dicho antes que el entorno es similar al de Autocad. Pero manejando archivos potencialmente grandes es recomendable manejar estructuras alámbricas y de esa manera no suavizadas.

Tenemos varias opciones para la configuración de todo esto; haciendo click con el botón derecho sobre el nombre de la ventana de cualquiera de las cuatro visualizaciones tendremos acceso a diferentes configuraciones.



Siempre hay un visor activo, marcado con un borde resaltado. El visor activo es donde los comandos y otras acciones tienen efecto. Sólo un visor puede estar activo a la vez. Otros visores se definen sólo para observación; a no ser que se desactiven, siguen simultáneamente las acciones realizadas en el visor activo.

Si el proceso se antoja cargado conviene dejar una sola de las ventanas activas, para que no se procesen todos los datos en los 4 visores a la vez. Para ello nos movemos con el cuadro de comandos que tenemos en la parte inferior derecha, y concretamente activamos/desactivamos la casilla  para dejar un solo visor, y trabajar de manera más fluida.

Otra posibilidad es la de **Ocultar** objetos en la escena, para ello bastaría con pulsar sobre el objeto en cuestión con el botón secundario y activar esta opción. De la misma manera tenemos **Congelar** objetos, parecida a la anterior, pero en este caso los sólidos no desaparecen de la escena, sino que simplemente se quedan como "no editables".

Acorde con el manejo de capas tenemos una última opción para ir editando aquellas que sean estrictamente necesarias. Se trata con el comando **Ocultar no seleccionadas** accediendo de la misma manera que hemos procedido antes.

Se debe saber también que 3D Studio Max, es un programa que hace uso intensivo de la aceleración por hardware, y por consiguiente convendría conocer el equipo que manejamos en base a una buena configuración del programa, así evitamos contratiempos en la visualización y otros aspectos de índole parecida.

### 3. EDICIÓN Y MANEJO DE MATERIALES PARA LA ESCENA

Este proceso consiste en aplicar materiales a los objetos que hemos modelado, y como tal "adheriremos" fotografías de distintos materiales como bloques, piedras o cristales, sobre las paredes de estos objetos, y darles luego un correcto tamaño, repetición, relieve, transparencia, reflexión, etc., que haga que aumente su realismo.

Hay que procurar que los materiales o mapas de textura que aplicamos sean lo más parecido posible a la realidad, ya que de estos materiales depende en gran medida el realismo de la representación final. Una mala elección de materiales puede estropear todo el trabajo realizado hasta ahora, ya que es lo que primero se va a ver en el resultado final.

Si el edificio fuese sólo un proyecto, seleccionaríamos los materiales que se indicasen en el proyecto, o bien podríamos realizar varias representaciones con distintos materiales para seleccionar el más conveniente estéticamente. Al estar trabajando sobre algo que ya es vigente, procuraremos que las texturas aplicadas sean una copia exacta en la medida de lo posible de la realidad.

Los materiales a aplicar los podemos elegir directamente de los múltiples materiales que aparecen en la biblioteca incluida en el 3D Studio, que por defecto viene con el nombre "3dsmax.mat" o variar la creación con otro tipo de materiales que puedan ser importados o añadidos desde otros medios, esto es:

- Al igual que disponemos en Autocad, 3D Studio es una herramienta de uso muy extendido, por lo cual existen multitud de bibliotecas con materiales y aplicaciones para su edición ya sean gratuitos o de índole comercial.
- La web es una buena posibilidad para intercambiar archivos de materiales.
- Podremos optar también por el soporte fotográfico para incluir materiales.
- Al igual que lo anteriormente expuesto, programas de dibujo o de retoque fotográfico, nos permiten crear nuevas texturas.

En este proyecto se han usado varias bibliotecas, y como hemos dicho al principio de este trabajo, la mayor parte de las texturas aplicadas son resultado de fotografiar la escena real.

### 3.1. CONSIDERACIONES PREVIAS

Para que los materiales tengan un aspecto lo más parecido a la realidad hay que tener en cuenta qué es lo que se quiere representar:

- **Materiales naturales:** El color ambiental es el complementario del color de la iluminación de la escena (si hay luz amarilla debe ser un azul muy oscuro). El

color difuso es el del elemento natural y el especular es el mismo color difuso pero algo más brillante. El brillo debe ser bajo.

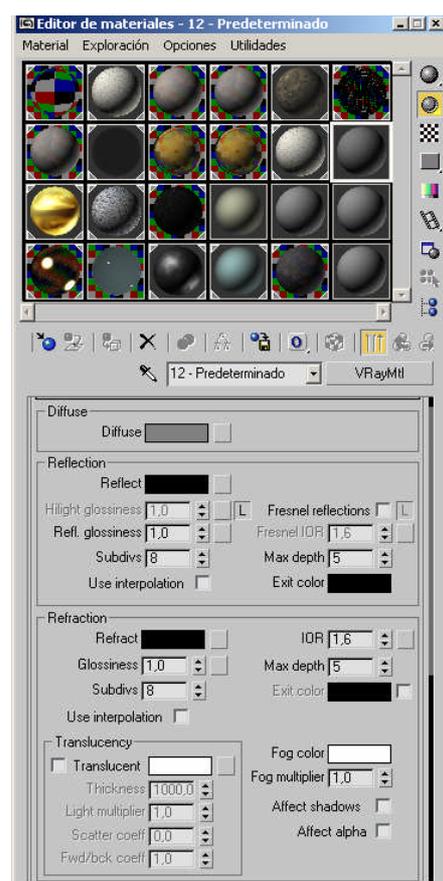
- **Materiales artificiales:** El color ambiental depende de la luz de la escena, como antes, el color difuso es el del objeto y el color especular debe ser blanco. El brillo es más alto que en los materiales naturales.
- **Materiales de propiedades metálicas:** Al seleccionar el tipo de sombreado metálico basta con elegir el color para la zona difusa.

Antes de nada diremos que dependiendo del renderizador que seleccionemos, serán convenientes aplicar unos materiales u otros.

Dicho esto, vamos a ver que aspecto diferenciado tendrá la sección de materiales con respecto a unos u otros sistemas.



**Editor de materiales  
predeterminado**



**Editor de Material Vray**

A simple vista parecen igual, pero la imagen final puede diferir ampliamente; en la imagen de la izquierda, hemos optado por un **material estándar** mientras que en la imagen de la derecha nos hemos decantado por un **material Vray**.

Es importante saber que no todos los materiales pueden ser renderizados de la misma manera, es decir:

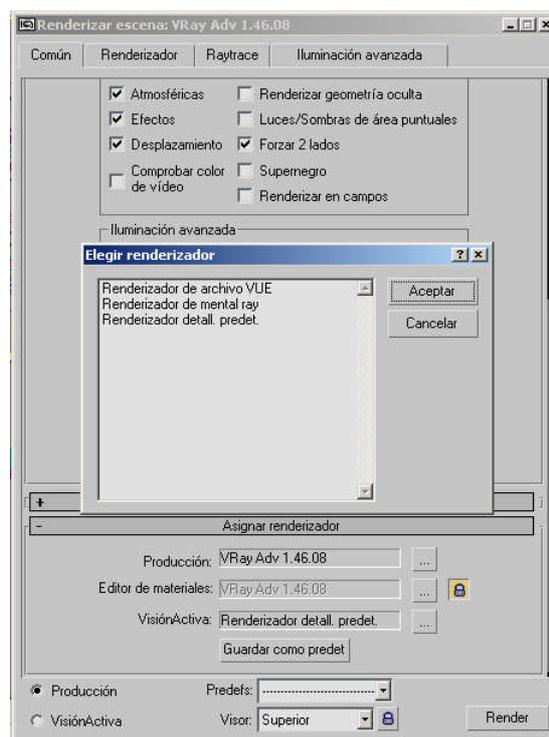
- Los materiales Vray, se representarán de forma satisfactoria con el renderizado Vray. El renderizador predeterminado falla con estos materiales.
- Los materiales estándar se pueden visualizar correctamente con cualquier motor de render, por ello son los que presenta el 3D Studio por defecto.

He optado por materiales Vray mayoritariamente, para un mayor atractivo visual.

### 3.2. TIPOS DE RENDERIZADORES

La herramienta 3D Studio Max, utiliza varios tipos de motor render, que pueden ser seleccionados para su uso solamente uno a la vez.

Para poder ver esta opción, dentro del programa ejecutaremos **Renderización/Render...** o simplemente presionaremos sobre la tecla de acceso rápido **F10**.



En la imagen anterior se muestran los renderizadores asignados a Producción y Visión activa (y al editor de materiales) y permite cambiar la asignación. El programa 3D Studio Max incluye tres renderizadores:

1. El Renderizador detallado predeterminado, que es el asignado por defecto tanto a Producción como a Visión activa.
2. El renderizador de Mental Ray.
3. Y el renderizador de archivo VUE.

Estos dos últimos sólo están disponibles para Producción. El renderizador de archivo Vue envía los datos a un fichero en formato ASCII, con la extensión *.vue*, que puede abrirse para inspeccionar, corregir a mano y enviar a otros motores de render, con el Bloc de Notas o un procesador de textos corriente. Si hay otros plug-ins cargados pueden escogerse otros renderizadores que mostrarán sus controles específicos en la siguiente sección.

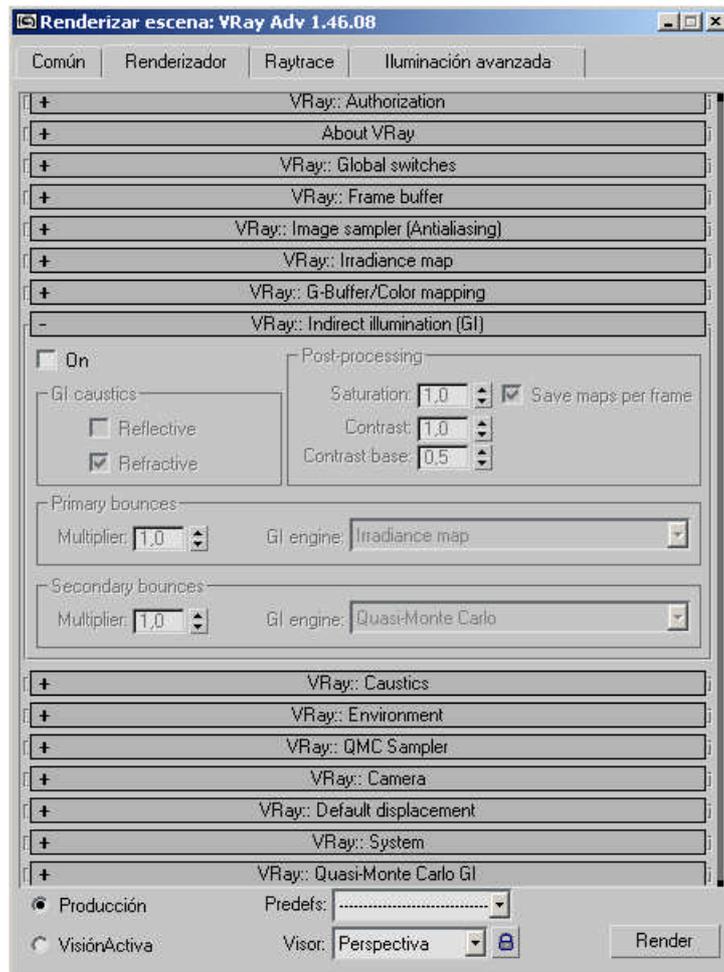
En nuestro caso hemos implementado un sistema de renderizado actual, que no tiene nada que ver con los que el programa incorpora por defecto, el **Vray** en su versión 1.46.08.

Con ello queremos dar a conocer de alguna forma el soporte del que hace gala el programa con el que estamos trabajando, dejando así entrever sus múltiples posibilidades.

### **3.2.1. RENDER VRAY 1.46.08**

Este motor es uno de los más usados hoy en día por las principales compañías que se dedican a este mundo de forma profesional. Resulta una aplicación rápida y en cuanto a los resultados muy atractivos en apariencia.

Los controles de la aplicación de texturas y materiales son muy similares al que lleva incorporado el renderizador predeterminado del 3D Studio, pero donde realmente podremos encontrar configuraciones para un enfoque más realista es en la ventana de opciones del renderizador, que pasaremos a ver algunas de ellas a continuación.



### **Panel de Image Sampler:**

Tendremos los controles descritos a continuación.

#### **Fixed rate:**

Test simple de imagen que nos da un número fijo de muestras para cada píxel, funciona mejor en escenas con texturas muy detalladas.

Subdivs: Nos determina el número de muestras por píxel.

#### **Adaptive QMC:**

Nos da muy buena calidad y rebaja los tiempos de fix rate.

Min subdivs: Controla el número mínimo de muestras por píxel.

Max subdivs: Controla el número máximo de muestras por píxel.

**Adaptive subdivisión:**

Realiza menos de una muestra por pixel, lo que nos permitirá igualar la calidad de alisado en comparación con adaptive QMC pero tardando menos tiempo en realizarlo.

Min rate: Controla el número mínimo de muestras por pixel, tomando como valor cero una muestra por pixel.

Min rate: Controla el número máximo de muestras por pixel.

Threshold: Nos define el umbral usado para realizar el muestreo, cuanto menor sea el valor de este mayor será la calidad del alisado.

Rand: Nos permite un mejor alisado acercando las muestras que rodean a cada pixel definiéndolo en mayor medida.

**Antialiasing filter:**

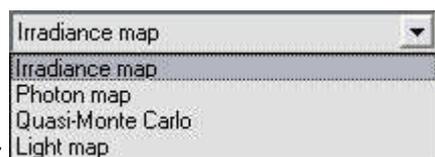
Nos permite controlar el alisado de los objetos en la escena. Entre todos los filtros disponibles el Catmull-rom nos servirá para la mayor parte de las escenas, realizando mejor las líneas.

**Panel de Indirect Illumination (GI):****Gi caustics:**

Reflective-refractive, activa o desactiva las reflexiones y refracciones caústicas en la escena.

**Primary bounces:**

Multiplier: Nos determina el valor para el motor de iluminación que seleccionemos (GI engine) en el primer rebote de la luz.



GI engine:

**Secondary bounces:**

**Multiplier:** Nos determina el valor para el motor de iluminación que seleccionemos (GI engine) en el segundo rebote de la luz.



**GI engine:**

### **Post-processing:**

**Saturation:** Regula la cantidad de saturación en la escena.

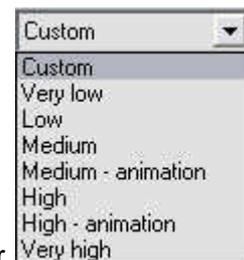
**Contrast:** Regula la cantidad de contraste en la escena.

**Contrast base:** Aumentando su valor reduciremos la exposición del blanco.

### **Panel de Irradiance Map:**

#### **Built-in presets:**

Nos permite seleccionar 7 valores preestablecidos y uno personalizable en **current**



**preset**, lo que nos servirá para poder regular la calidad del render.

#### **Basic parameters:**

**Min rate:** Cantidad mínima de muestras por pixel. El valor cero significaría una muestra por cada pixel, mucha calidad pero tiempos de render largos. Los valores positivos casi nunca suelen ser usados, ya que sería igual que utilizar computación directa, lo que prolongaría en exceso el tiempo de cálculo del render. Los valores negativos son los apropiados, pero cuanto menores sean estos mayor pérdida tendremos en calidad y menor será el tiempo de cálculo del render. Estos datos también son aplicables a **Max rate**.

**Max rate:** Cantidad máxima de muestras por pixel.

*(Cuanto mas próximos estén los valores a cero mas calidad obtendremos en el render)*

Hsph subdivs: Cantidad de muestras que van a ser computadas por la iluminación global.

Determina el número de rayos que serán lanzados en un punto dado.

HSphere	subdivs:	1	=	1	ray
HSphere	subdivs:	2	=	4	rays
HSphere	subdivs:	3	=	9	rays
HSphere	subdivs:	4	=	16	rays
HSphere	subdivs:	5	=	25	rays
HSphere	subdivs:	6	=	36	rays
HSphere	subdivs:	7	=	49	rays
HSphere	subdivs:	8	=	64	rays
HSphere	subdivs:	9	=	(9x9=81)	81 rays, etc....

A mayor cantidad de muestras mas calidad general para el render.

Habrá que tener en cuenta que las hsph subdivs e interp samples deben de estar equilibrados.

Incrementado el hpsh reduciremos un poco las interp ya que no serán necesarias tantas muestras para el calculo, así evitaremos artefactos (manchas) en el render.

Ocurriría a la inversa si aumentamos las interpolaciones, en este caso no serian necesarias tantas hpsh subdiv.

Un ejemplo que podamos usar por norma general y que nos de suficiente calidad seria hpsh subdivs: 50 e interp. samples: 20.

Interp samples: Cantidad de muestras que son almacenadas por el mapa de irradiación.

Clr thresh: Añade mas muestras en zonas de cambio de iluminación mas acusadas según la diferencia de color de dos muestras cercanas o el ángulo que formen entre ellas.

Cuanto menor sea el valor de este mas calidad nos dará en el render. Un valor intermedio que nos de una calidad aceptable puede ser **0,3**.

*Nrm thresh*: Añade mas muestras en zonas de cambio de iluminación menos acusadas según la diferencia de color de dos muestras cercanas o el ángulo que formen entre ellas.

Cuanto menor sea el valor de este mas calidad nos dará en el render. Un valor intermedio que nos de una calidad aceptable puede ser **0,3**.

*Dist thresh*: Es muy útil para zonas de contacto entre objetos, aumentando o disminuyendo la cantidad de muestras entre esas partes mas próximas y que necesitan ser mas definidas.

El ejemplo mas claro son las esquinas. Este valor funciona al contrario que *Nrm* y *Clr thresh*, cuanto mayor sea mejor sera la definición de este y mas muestras tomara en esas zonas.

*Show calc. phase*: Nos va mostrando las fases previas del calculo del mapa de irradiancia con respecto a los valores dados en *min-max rate*.

*Show direct light*: Visiona un previo de las muestras que serán tomadas para el calculo del mapa en las luces directas.

*Show samples*: Visiona las muestras que serán tomadas para el cálculo del mapa de irradiación.

### **Panel de Vray Environment:**

#### **Gi enviroment (skylight):**

*Override Max's*: Activa la iluminación global del entorno, pudiendo ser aplicado un color especifico o una imagen. Si utilizamos esta casilla anularemos el efecto del entorno del Max sobre la iluminación. El multiplier nos permitirá regular la intensidad de esta iluminación.(Por defecto 1).

#### **Reflection/refraction etc enviroment:**

Similar a *gi enviroment* pero aplicable solo a las reflexiones y refracciones de los materiales.

Se han descrito ya los que quizá sean los controles más importantes a la hora de configurar una escena, de todos modos, no vamos a entrar en más detalles, ya que no se pretende plasmar lo que de alguna forma sería el manual de este programa.

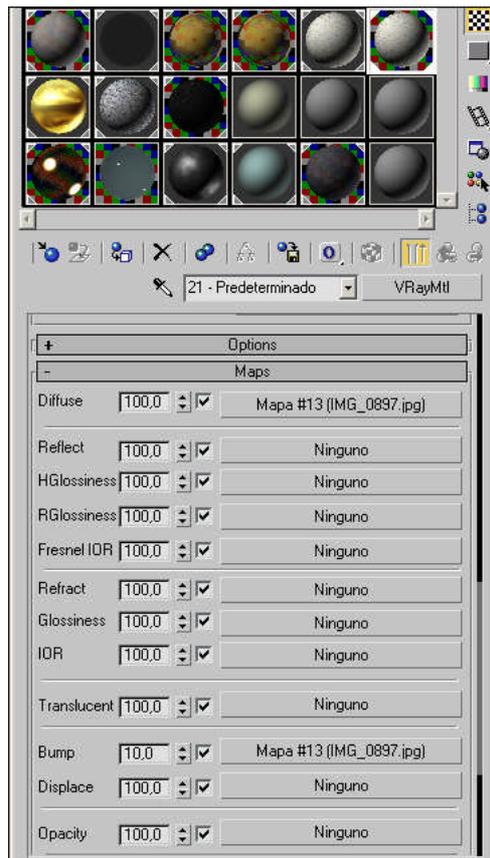
### 3.3. APLICACIÓN DE LOS DISTINTOS MATERIALES

#### 3.3.1. DESDE EL INICIO

Diremos en primera instancia, que tenemos varias posibilidades, en cuanto la elección de materiales Vray o Standard.

Voy a explicar el proceso que he seguido para aplicar todos los materiales de éste tipo que hay en la escena. Lo hago con un material solo, ya que los demás se aplican de la misma forma.

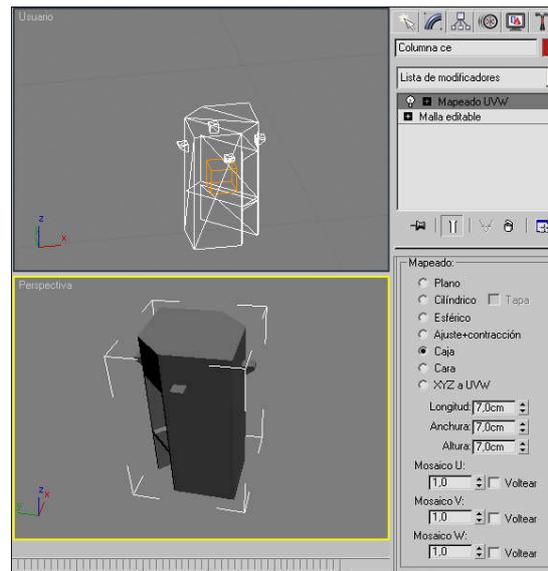
1. En primer lugar aplico el material de pintura plástica *beige* con resaltes a la columna central. Para ello escojo un material Vray. Antes, para mayor claridad en el dibujo, oculto todos los objetos menos los que llevarán el ladrillo utilizando las opciones de la persiana **Ocultar** en el panel **Presentación**.
2. Selecciono el sólido y abro el menú **Editor de materiales**. Selecciono una casilla de material, extendiendo la persiana **Maps** y en **Difuse** hago clic sobre el botón **Ninguno**. En la lista de nuevos materiales que aparece selecciono **Imagen bitmap**. Ahora en la persiana **Parámetros de imagen bitmap** hago clic sobre el botón **imagen bitmap** para elegir un archivo de imagen almacenado en una carpeta de nuestro pc.
3. Una vez elegido, en el botón **Imagen bitmap** aparece el nombre del archivo y en los recuadros de muestra del material aparece una vista previa de cómo queda el material. Ahora hago clic sobre el botón **Mostrar mapa en el visor** si quiero ver representado el material en los visores cuando están en modo sombreado, sin necesidad de hacer un render.
4. Le damos un valor de reflejo de 5 y en la ranura de Bump, aplicamos el mismo mapa que para la textura, y con un valor de 10, para simular relieve.



5. Asignamos el material a la selección mediante  para que el sólido adopte la textura en el visor.

La aplicación de un material que contenga textura bitmap, hay que especificarla mediante un **mapa UVW**, con ello conseguimos que el mosaico de textura se escale de forma a nuestro cometido.

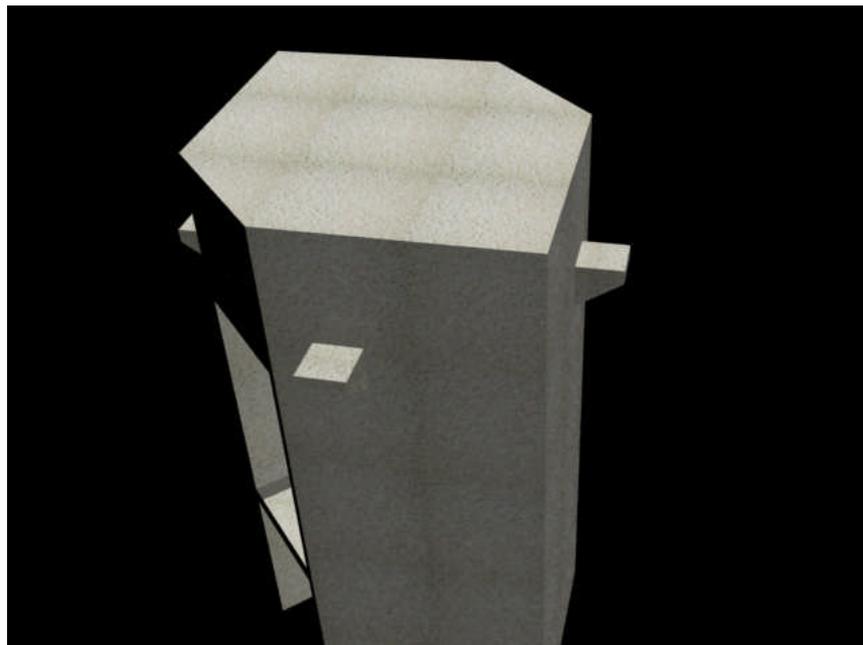
- Con la selección en vigor, ejecutamos en el panel **modificar**, la orden **Mapa UVW**.



6. Para los sólidos con aspecto de prisma, usaremos un mapeado en forma de caja, ya que la textura se distribuirá de una forma más o menos uniforme conforme a la geometría.

7. Los parámetros U, V, W nos ayudarán a ajustar el bitmap, para un resultado óptimo.

He tenido algún tipo de problema con el visor en escena, y es que, no se mostraban algunas veces los mapas en los visores, por lo que vamos a mostrar una captura render del sólido tratado.



De esta forma sigo aplicando materiales y coordenadas de mapeado a todos los objetos de la escena que lleven materiales con un mapa de bits externo a la biblioteca de MAX, y aplicándoselo a la ranura de Difuse (Vray Mat.) o Difusa (Mat. Estándar).

Es importante atender a la geometría del sólido al que le aplico el mapa, ya que no quedaría bien, implementar un mapeado esférico a una geometría plana, y a la inversa.

Por ello a partir de ahora diremos que:

- Mapeado "Caja" para superficies planas y de geometría 3D a base de regiones extraídas en Autocad anteriormente.
- Entorno esférico para cubiertas y demás tipos.

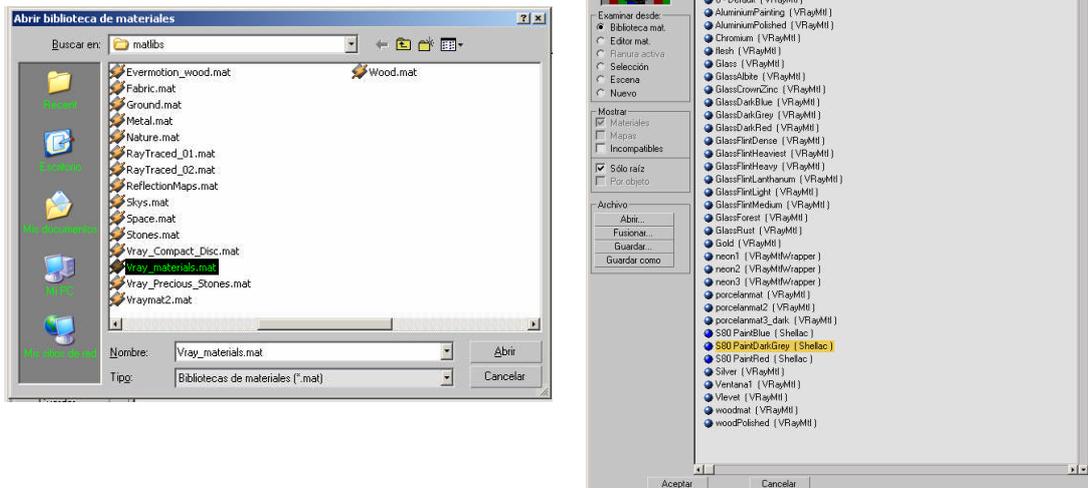
### 3.3.2. DESDE LA BIBLIOTECA

Para los soportes superiores en forma de semicerchas adosadas a la columna central voy a seleccionar un material de la biblioteca de materiales.

Al tratarse de un material Vray, lo elegiremos de una biblioteca con la que nos habremos hecho anteriormente.

El proceso seguido es el siguiente:

1. En la ventana del Editor de Materiales pulso el botón  Asignar material.
2. Selecciono **Examinar** desde biblioteca de materiales y elijo un material brillante de color negro similar al que llevan estos elementos, como por ejemplo el material "S80 PaintDarkGrey (Shellac)". Éste está ubicado en la biblioteca Vray\_Materials.mat. Hago doble clic sobre él y ya lo tengo en el recuadro de muestra del editor de materiales.



3. Al igual que hemos hecho anteriormente, asignamos un mapeado U, V, W y ya tendremos el material deseado.

### 3.3.3. OTROS

He tenido que hacer otro apartado para casos excepcionales en los que por circunstancias de la realidad a veces se antoja complicado modelar los objetos con el Autocad como hemos hecho hasta ahora. Por ello tenemos opciones:

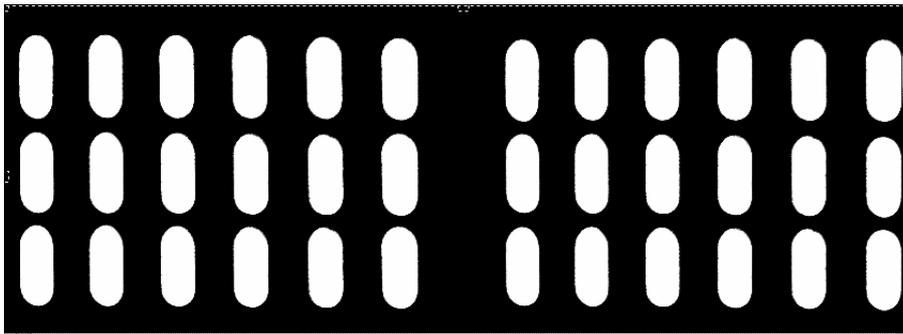
- Ya que queremos representar objetos de geometría complicada, podremos dimensionarlos, con el tiempo y las consecuencias de inexactitud que ello conllevaría...
- O bien, aplicaremos efectos en la geometría con el 3D Studio a base de texturas, que es lo que hemos hecho en nuestro caso.

Vamos a poner como ejemplo a este caso, el texturizado de un elemento común como es el banco de ambos pisos.

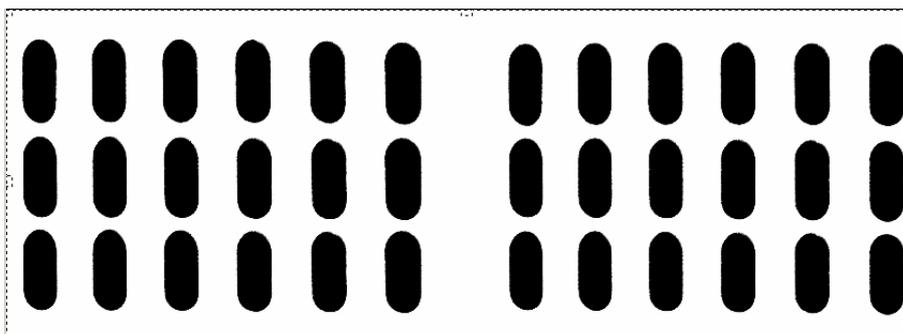
Este archivo junto con la papelerera que conforman los diferentes complementos se les han aplicado texturas en un archivo aparte.

1. Una vez importado el banco en \*.3ds procedemos a aplicar texturas.

2. Abrimos el editor de materiales y creamos un material Vray que simule los agujeros del respaldo y cuerpo del banco.
3. Para ello hemos tomado una foto del banco y tras tratarla en el photoshop, llegamos a un resultado plano como el visto a continuación, que nos servirá para asignarla en la ranura de difuse.
4. Aplicamos un mapa U, V, W como hasta ahora.

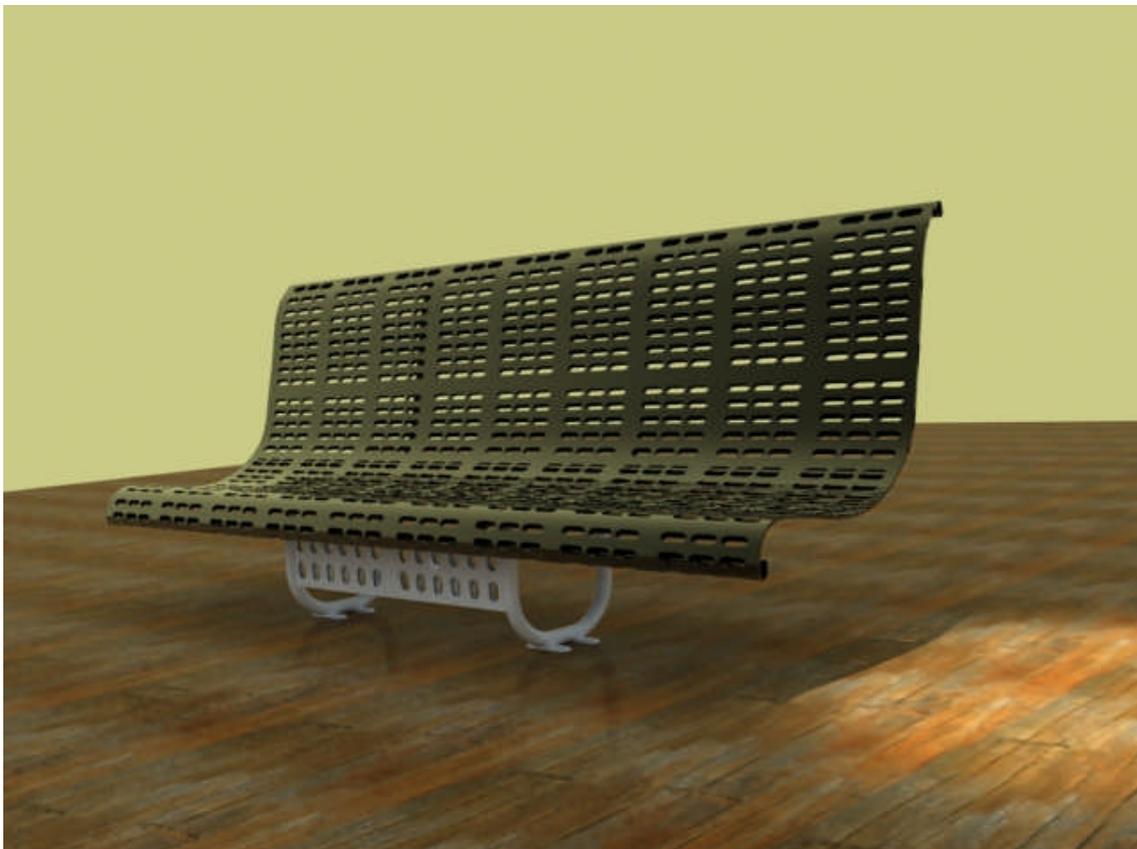
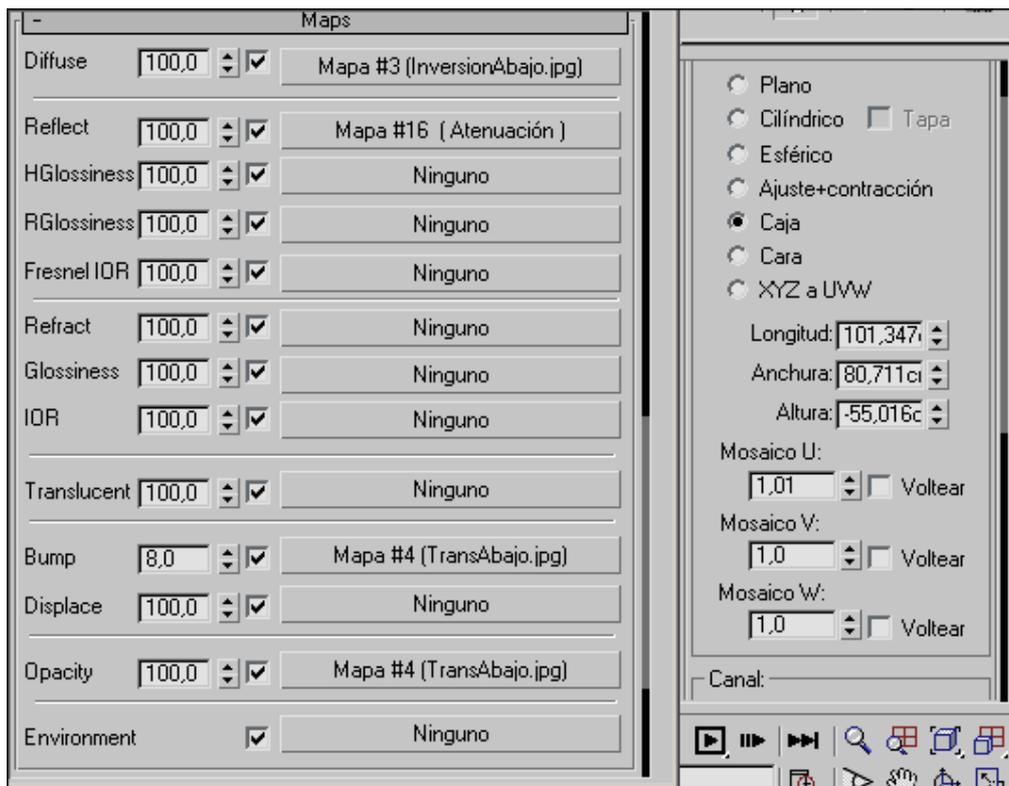


5. Ya tenemos la textura que formará en respaldo del banco, pero si la aplicamos así sin más, notaremos que los agujeros que han de ser transparentes, quedan blancos.
6. Para solventar este contratiempo, existe una ranura, con el nombre **Opacity**. Ésta trabaja de forma que la imagen que se asigne ha de ser en blanco (se mantiene la geometría) y negro (hace que la geometría desaparezca) para un mejor resultado. Es decir que si lo que nosotros queremos es que los agujeros sean transparentes aplicaremos la misma imagen, sólo que invertida; es decir lo que antes era negro ahora pasa a blanco y a la inversa.

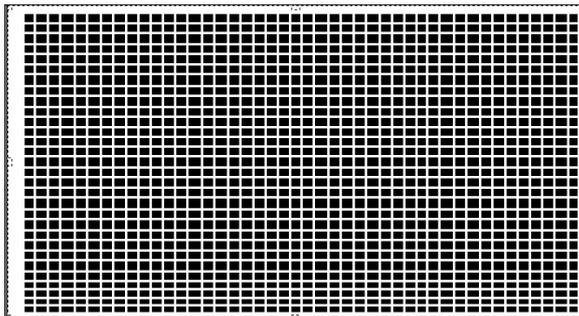
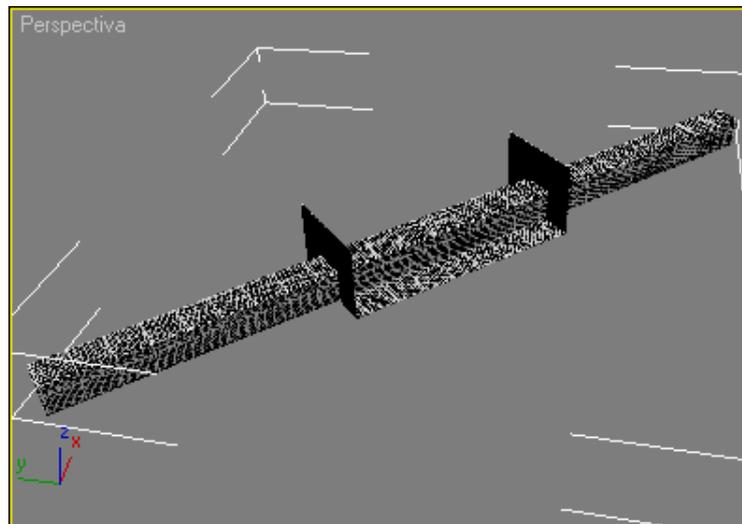


7. Ahora ya tendremos constancia de que los agujeros son transparentes. Sólo quedaría hacer que el banco presente efecto de tridimensionalidad en los agujeros, para ello en la ranura **Bump** coloco el mismo mapa que para la ranura **opacity**.
8. Para dar un efecto de suavizado y relación con el entorno, aplicamos una atenuación (mapa 3D desde el árbol de materiales) en el canal **Reflect**.
9. Para los demás elementos del banco procedemos de igual forma, con la excepción de que ya no tendremos q aplicar transparencias, lo haremos como hasta ahora.

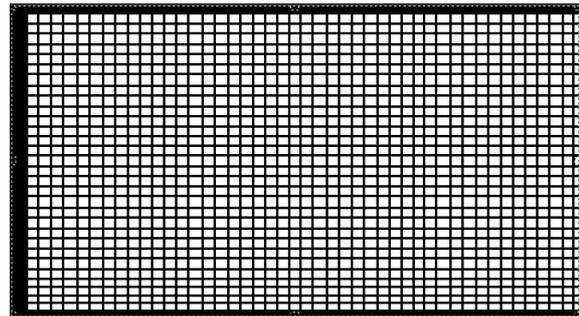
Vamos a ver el aspecto que presentaría la ranura de materiales/mapeado UVW de este elemento y su representación final aplicando un entorno diferente para verlo con mayor claridad.



Para el caso del canal que pasa a través de la columna central, se ha hecho lo mismo, sólo que aplicando otros mapas en las mismas ranuras.



**Mapa Opacity**



**Mapa diffuse**

De esta forma podemos comprobar que las posibilidades con las texturas son infinitas.

Cabe resaltar que este proceso de transparencias ralentiza de manera importante el proceso de renderizado, pero los resultados son óptimos.

### 3.4. SÓLIDO MULTIMATERIAL

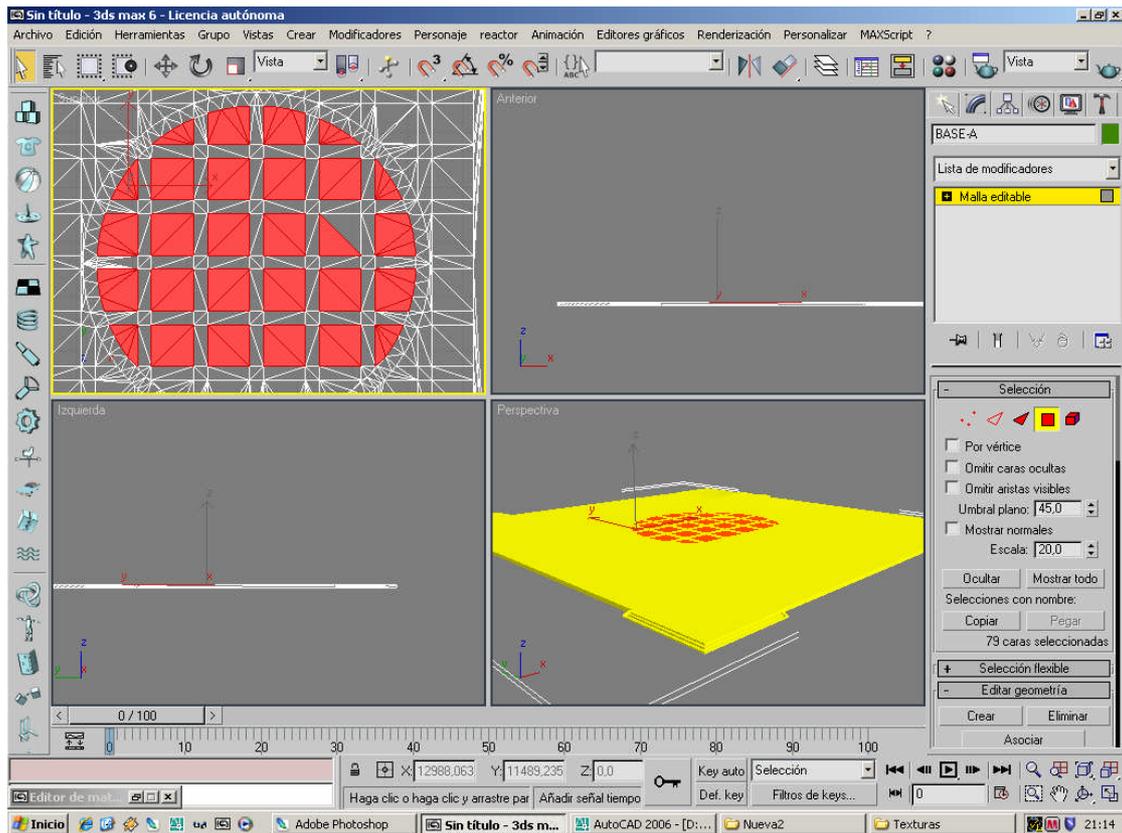
Durante el proceso de aplicación de materiales nos damos cuenta de que parte de un objeto, como por ejemplo alguna de sus caras, tienen diferente material al resto, para ello tenemos que disociar esas caras o polígonos en nuestro caso.

Se hace seleccionando el objeto y haciendo clic en el panel **Modificar**. Seleccionamos el modificador **Editar Malla**. Tenemos cinco niveles de selección:

- Vértice  : Activa el nivel de subobjetos Vértice, que permite seleccionar un vértice bajo el cursor; la selección por región selecciona vértices situados dentro de la región.
- Arista  : Activa el nivel de subobjetos Arista, que permite seleccionar una arista de cara o polígono bajo el cursor; la selección por región selecciona múltiples aristas situadas dentro de la región. Cuando el nivel de subobjetos Arista está activado, los vértices ocultos se muestran como líneas discontinuas, permitiendo una selección más precisa.
- Cara  : Activa el nivel de subobjetos Cara, que permite seleccionar una cara triangular bajo el cursor; la selección por región selecciona múltiples caras triangulares situadas dentro de una región.
- Polígono  : Activa el nivel de subobjetos Polígono, que permite seleccionar todas las caras coplanares (definidas por el valor del contador Umbral plano) que hay bajo el cursor. Normalmente, el polígono es el área que puede verse entre las aristas alámbricas visibles. La selección por región selecciona varios polígonos situados dentro de una región.
- Elemento  : Activa el nivel de subobjetos. Elemento, que permite seleccionar todas las caras contiguas de un objeto. La selección por región permite seleccionar múltiples elementos.

Elegiremos **Polígono**. Ahora en cualquiera de los visores seleccionamos las caras que queremos disociar (se pondrán de color rojo) y hacemos clic en el botón **Disociar de la persiana Editar polígono** dentro del panel de los modificadores. Damos un nombre a ese nuevo objeto y ya tenemos dividido el objeto original en varios objetos. Ahora podemos aplicar diferentes texturas.

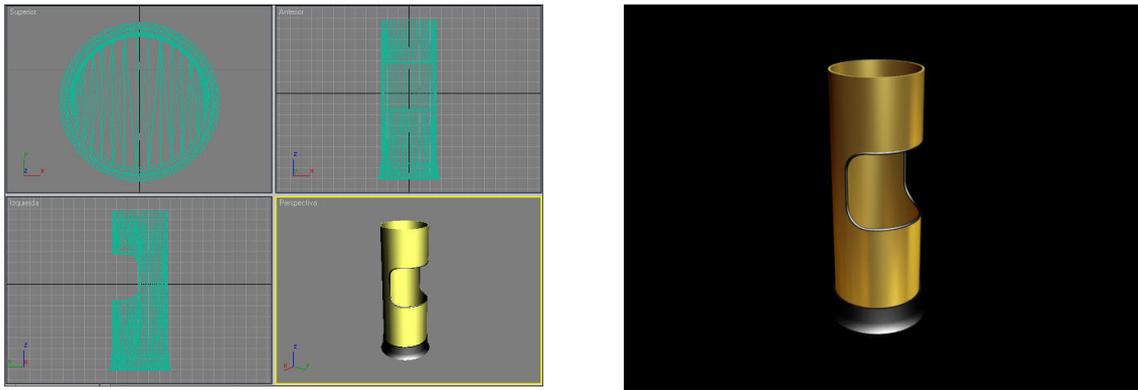
Esta disociación se la he aplicado por ejemplo a los muros interiores inferior y superior de los pisos y a las bases de los mismos, ya que presentan diferentes gamas de mármoles y en mosaicos claramente diferenciados.



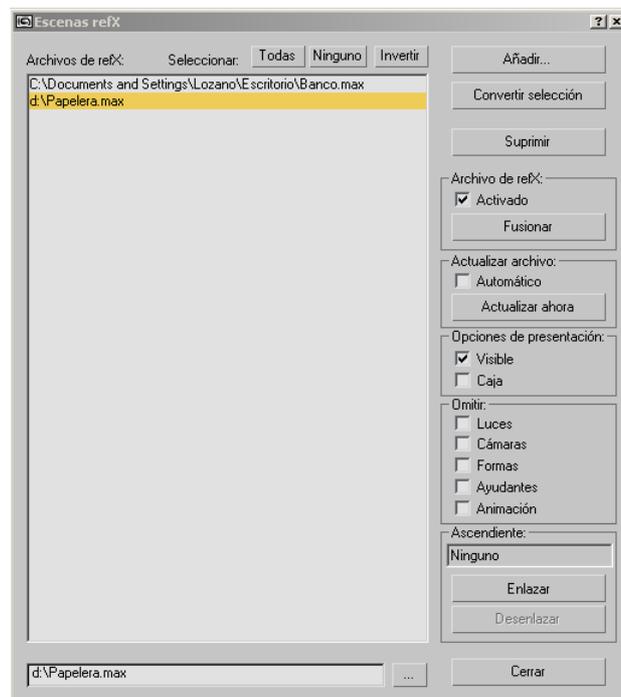
La disociación de las caras que hemos hecho es laborioso, debido al gran número de caras que se han tenido que editar; para una mayor fluidez hemos realizado este proceso con todas las demás capas que no se necesiten en este apartado ocultas.

## 4. CREACIÓN DE CALCOS EN LA ESCENA

Anteriormente se expuso, que objetos tales como el banco o la papelera se crearon en archivos aparte. A su vez se les incorporó las texturas de forma independiente y se guardaron como archivo de 3D Studio en formato \*.3ds formando lo que se llama una escena.



A la hora de agregar dichos archivos al conjunto, importamos la escena mediante la secuencia **Archivo/ Escena refX...** y allí seleccionamos el o los archivos que queremos agregar.



Una vez obtenidos los archivos, podremos hacer con ellos lo que queramos dentro de la escena.

Si observamos tanto la papelera como el banco son dos archivos que poseen varios elementos en su constitución, por lo cual para un manejo más cómodo, agruparemos los distintos objetos de cada archivo en uno solo mediante **Grupo/ Agrupar** y le daremos un nombre, de esta forma podremos seleccionar cada archivo como una única pieza en el conjunto.

Ahora bien, posiblemente, los objetos sean de un tamaño poco acorde con lo previsto, así pues mediante el comando **Escalar**  le damos el tamaño adecuado.

El último paso es realizar el calcado de los objetos. Calcar un objeto es hacer una copia que mantiene las mismas propiedades que su primitiva, de tal manera que si cambiamos una propiedad, se cambian en todos los objetos calcados. Con el objeto seleccionado, y presionando la tecla **Shift** arrastramos y soltamos, se nos pedirá que especifiquemos lo siguiente:



Una vez hechas todos los objetos, solamente quedaría colocarlos en la posición deseada dentro de la escena.

## 5. IMAGEN DE ENTORNO

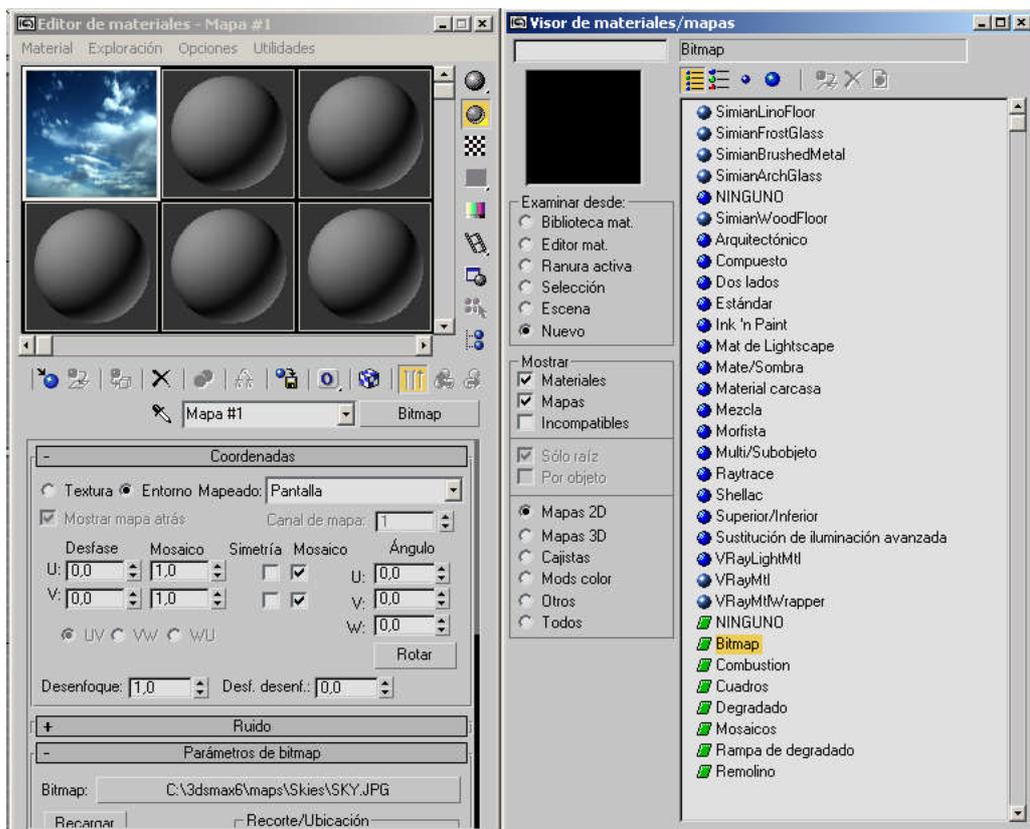
Ahora hay que colocar algo como fondo de la escena, que sirva para simular el cielo. Tengo varias posibilidades:

- Desde una imagen de un cielo despejado, con nubes o cualquier otro tipo.
- Hasta un fondo que consista en un color sólido, o degradado.

El color sólido se selecciona en el recuadro Color de la ventana Entorno perteneciente al menú Representación.

Para aplicar una imagen, lo hago desde el Editor de materiales. Una vez en él:

1. Hago clic en el botón Asignar material , y en la lista de Nuevo selecciono Imagen bitmap. En el botón Imagen bitmap selecciono una imagen de un cielo.
2. Seguidamente, en la persiana Coordenadas selecciono la opción **Entorno**, y en la lista desplegable selecciono:
  - **Entorno esférico** si lo que voy a representar es un video
  - **Pantalla** si voy a renderizar una imagen fija.



3. Ahora, en la opción **Entorno** del menú **Renderización**, hago clic en el botón **Mapa de entorno**. Simplemente lo que haré será arrastrar el mapa desde el editor de materiales hasta la sección de **Entorno/ Parámetros comunes/ Fondo** y allí busco la ranura de mapa de entorno.

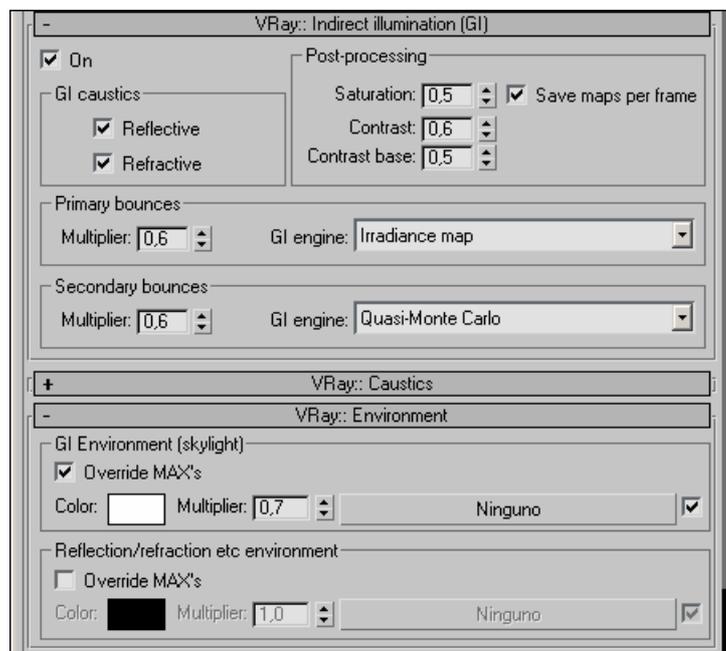
Un inconveniente si elijo una foto de un cielo con nubes es que al hacer el video puede salir el empalme entre extremos del mapa; esto lo comprobaremos más adelante, aunque puede que no suponga mucho problema ya que las escenas son interiores.

## 6. ILUMINACIÓN GENERAL

### 6.1. ILUMINACIÓN AMBIENTAL

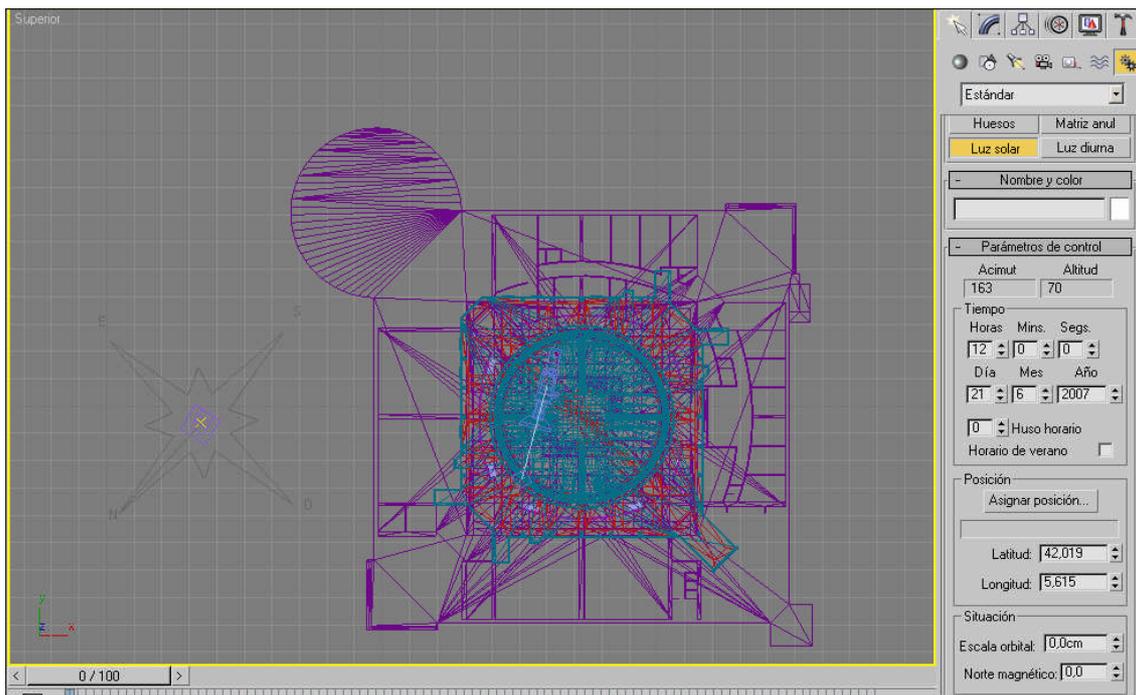
La iluminación es una fase de desarrollo complicada y que requiere de grandes dosis de tiempo. Elegir bien el tipo de luz a utilizar, su intensidad, posición del foco emisor, etc., es fundamental para que la representación final sea lo más realista posible.

Como fuente de luz principal se ha escogido la configuración típica del sistema GI de Vray, es decir además de la iluminación indirecta (GI, Indirect Illumination) el entorno (Environment).



### 6.2. ILUMINACIÓN SOLAR

Para simular una luz solar en algunas escenas aisladas, 3D Studio Max utiliza una luz tipo Direccional libre. Se puede escoger esta luz u otras de otro tipo en la categoría **Luces** del panel crear. Sin embargo hay una opción específica que se llama luz solar y que permite configurar, además de los parámetros propios de cada tipo de luz, la ubicación geográfica real de la escena mediante coordenadas de latitud y longitud que establecerán automáticamente el acimut y la altitud, y el momento del día mediante fecha y hora.



Para ello, dentro del panel Crear, selecciono la categoría Sistemas, y en tipo de objeto selecciono Luz solar.

En la persiana Parámetros de control hago clic en el botón Asignar posición y selecciono en el mapa la ubicación aproximada de la ciudad de León:

- País: **España (ES)**
- Latitud Estación = **42.58N** (decimal)
- Longitud Estación = **5.65W** (decimal)
- Altura Estación = **926** (metros)
- Diferencia horaria con respecto a horario GMT = **+1** (horas)

También indico el día y la hora para que la luz se sitúe con el ángulo correcto. Ahora en el visor superior hago clic con el ratón y arrastro para crear la Rosa de los vientos que no es más que un objeto que me indica los puntos cardinales. En realidad esto me crea dos objetos, la Rosa de los vientos, que se puede girar para que señale el norte correctamente, y una luz tipo Direccional libre cuyos parámetros de

intensidad, color, etc., se pueden variar seleccionando esta luz y picando en el panel modificar.

De esta forma, en la persiana Parámetros generales, en el recuadro Color, selecciono un color de tono blancuzco para intentar un mayor contraste, y en Multiplicador ajusto la intensidad de la luz probando con varios valores, de los cuales me he decantado por 1. En la persiana Parámetros de sombra, indico si quiero Proyectar sombras y qué tipo de mapa de sombras voy a usar. Tanto las sombras raytrace, como las designadas por Vray son mucho más realistas y precisas pero ralentizan enormemente el renderizado. Elegir un tipo u otro de sombras estará en función de la complejidad de la escena, esto es, el gran número de capas y geometrías y de la potencia del ordenador de que dispongamos. Las sombras que vamos a utilizar en todo momento son las Vray, ya que estamos creando todo el proceso con este motor.

## 7. CÁMARAS

### 7.1. GENERALIDADES

Las cámaras presentan una escena desde un determinado punto de vista. Los objetos de cámara simulan una imagen fija, de película o de vídeo real.

Mediante un visor de cámara podremos ajustar la cámara como si estuviéramos mirando por su objetivo. Los visores de cámara son útiles para editar geometría y configurar una escena con el fin de renderizarla. La utilización de varias cámaras puede proporcionar distintas vistas de la misma escena. Hay dos tipos de objetos de cámara:

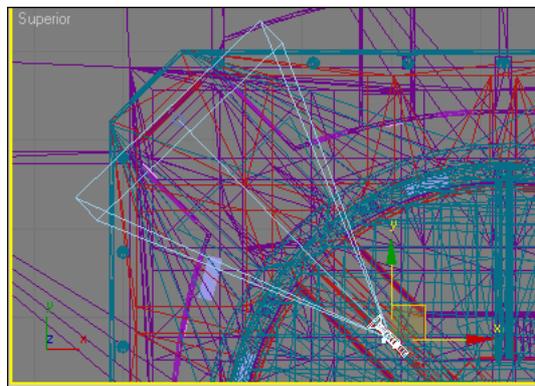
- Cámaras con objetivo : Ven el área que rodea al objeto de objetivo. Cuando se crea una cámara con objetivo, se ve un icono con dos partes que representa la cámara y el objetivo (una caja blanca). La cámara y el objetivo pueden animarse de forma independiente, por lo que resultan más fáciles de utilizar cuando la cámara no se desplaza a lo largo de un recorrido.
- Las cámaras libres : Ven el área en la dirección a la que apuntan. Cuando se crea una cámara libre, aparece un icono sencillo que representa la cámara y

su campo de visión. El icono de cámara aparece igual que el icono de cámara con objetivo, pero no dispone de un icono de objetivo para animar. Las cámaras libres son más sencillas de utilizar cuando se anima su posición por un recorrido.

## 7.2. CREACIÓN DE CÁMARAS

Para crear una cámara:

1. **Cámaras** del panel **Crear**, y como **Tipo de objeto** selecciono **Objetivo**. Ahora, en uno de los visores (preferiblemente el Superior), hago clic y arrastro el ratón al lugar y en la dirección a la que quiero apunte la cámara.



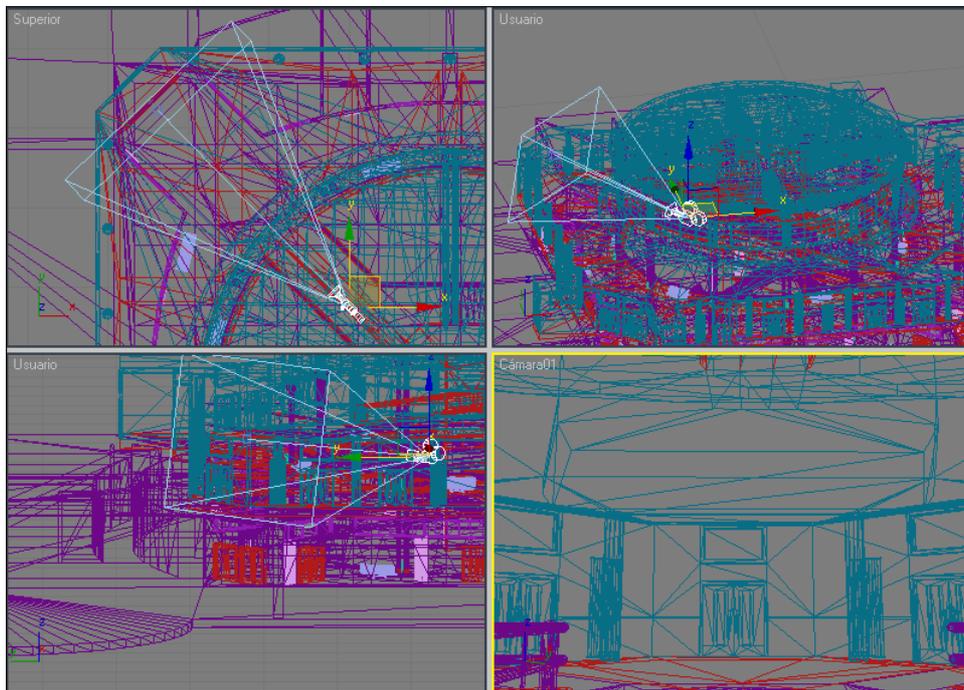
Se puede desplazar y rotar la cámara como un objeto más de la escena, si **queremos** colocarla en otro lugar.

Para ver la vista de la cámara en un visor, plico con el botón derecho sobre el nombre del visor. En el menú que aparece selecciono **Vistas** y elijo el nombre de la cámara.

2. Si selecciono la cámara y abro el panel **Modificar** puedo variar los parámetros de la cámara, como por ejemplo escoger alguno de los **Objetivos disponibles**. Lo mejor es probar con diferentes objetivos hasta conseguir una buena vista, pero teniendo en cuenta que una distancia focal baja produce el efecto de un gran angular de una cámara real, por tanto, los objetos aparecerían algo distorsionados. También puedo ajustar la cámara usando los botones de exploración que aparecen en la parte inferior derecha de la pantalla:



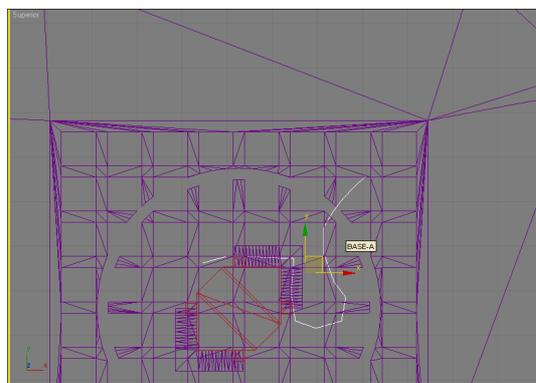
Vamos a ver un ejemplo desde una vista de cámara en la escena de la parte del hall superior del Edificio Tecnológico Fase-1.



### 7.3. CREACIÓN DE LOS RECORRIDOS DE CÁMARA

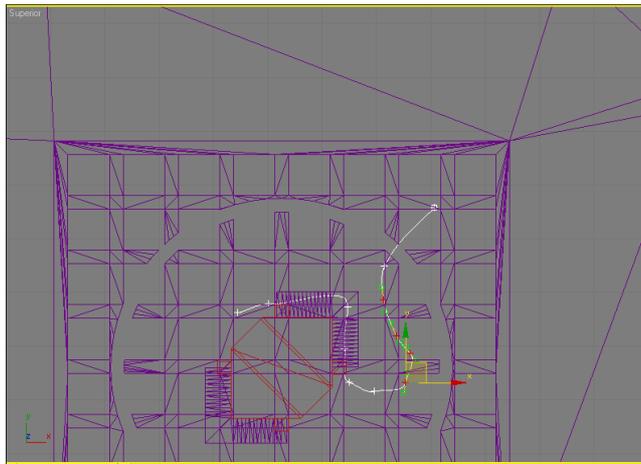
Debemos de tener claro desde un principio el recorrido que queremos para la cámara. Tenemos varios recorridos, por lo cual, como ejemplo vamos a tomar el primero de ellos, que va desde la entrada hacia el descansillo del primer piso.

1. Dibujo los segmentos de línea del recorrido mediante **Crear/ Forma/ Línea**. La dibujo en el visor superior para tener mejor visión.

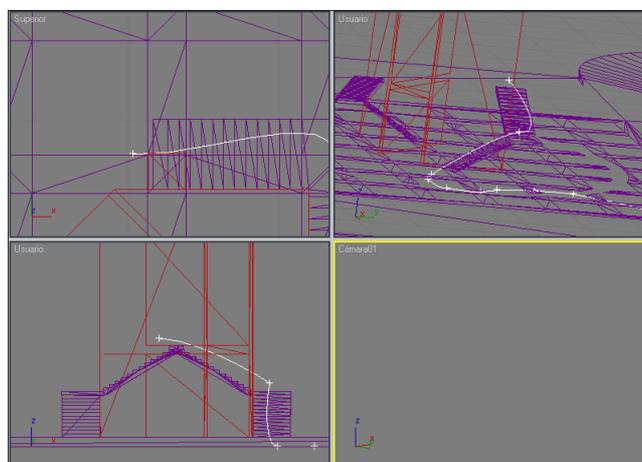


2. En el panel Modificar elijo Subobjeto vértice como Nivel de selección.

Selecciono todos los vértices, hago clic con el botón derecho del ratón sobre uno de los vértices y en el menú que aparece selecciono Bézier. Con esto consigo que los ángulos que tenía la línea se suavicen. En cada vértice nos dará la opción de estirar mediante tangencias, por medio de anclajes, para ajustar el nivel de curva del recorrido y así poder completar la guía.



3. Como el recorrido se pretende que sea subiendo las escaleras en algún momento del mismo, lo que tendremos que hacer será modificar la altura de algunos puntos que nos interese. Así pues, el visor superior no es el adecuado para ello, así que jugando con los otros visores a la vez (anterior y posterior) elevaremos el recorrido.



4. Si lo deseamos también podremos asignar el recorrido al objetivo, de tal

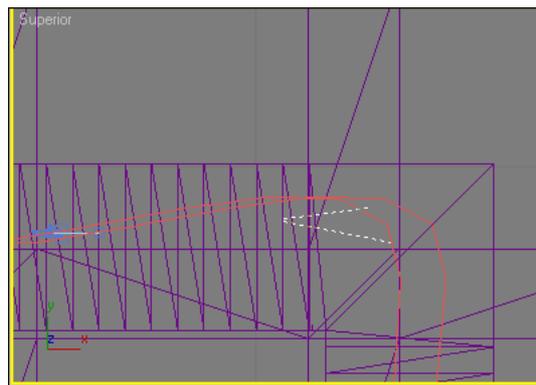
forma, que el sentido del mismo no coaccione los distintos puntos de mira que queramos dar.

#### 7.4. ASIGNAR EL RECORRIDO A LA CÁMARA

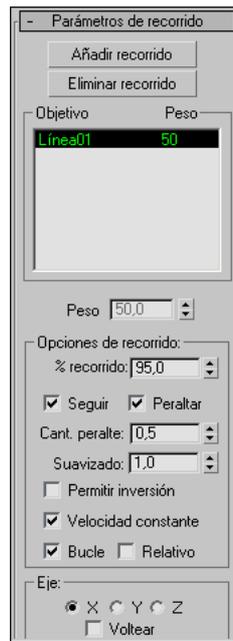
Ahora tendremos que enlazar la cámara con su recorrido. Para ello tendremos dos partes diferenciadas:

- Asignación del recorrido de la cámara.
- Asignación del recorrido del objetivo.

Para la primera opción sólo tendremos que ir a la siguiente ruta, **Animación/ Restricciones/ Restricción de recorrido**. Una vez hecho este paso, nos saldrá una línea discontinua blanca, indicándonos que designemos el recorrido de la misma, véase a continuación.



Procederemos del mismo modo con el objetivo de la cámara, teniendo en cuenta que no tiene por qué ser el mismo recorrido que antes.



## 8. TIEMPO Y VELOCIDAD DE ANIMACIÓN

Antes de asignar la cámara a su recorrido hay que configurar el tiempo que durará la animación de ésta cámara.

Haciendo clic en el botón de **Configuración de tiempo** de la parte inferior derecha de la pantalla, aparece la ventana con los controles del tiempo de la animación que nos permite ver tanto el tiempo que va a durar la secuencia, como la velocidad y el modo de representarlo.

En este caso me he decantado por el modo NTSC de 30 cuadros por segundo.

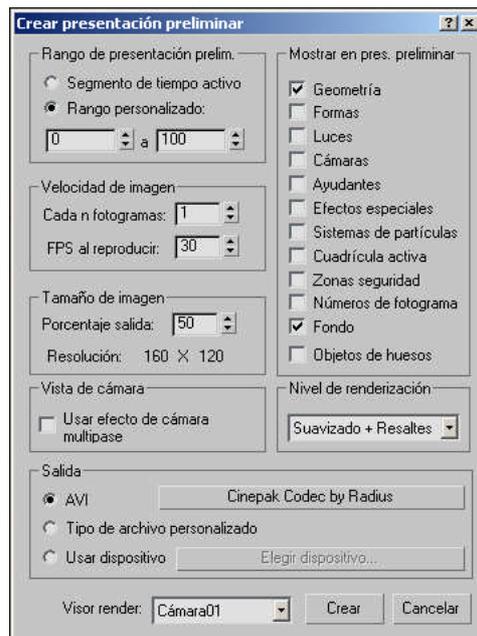


Con una duración de 10 o 15 segundos por vídeo, a una velocidad de 30 frames por segundo, obtendremos en número de frames final, que será de 450 aproximadamente.

## 9. PRELIMINARES

Antes de hacer el video definitivo, es mejor crear una presentación previa en forma de boceto, donde poder ver posibles fallos que hay que cambiar.

Para hacer esto selecciono la opción **Animación/ Crear presentación preliminar**.



En la sección Rango indico el Segmento de tiempo activo. Como Salida elijo un fichero AVI, y selecciono como Nivel de representación la opción Suavizado + resaltes.

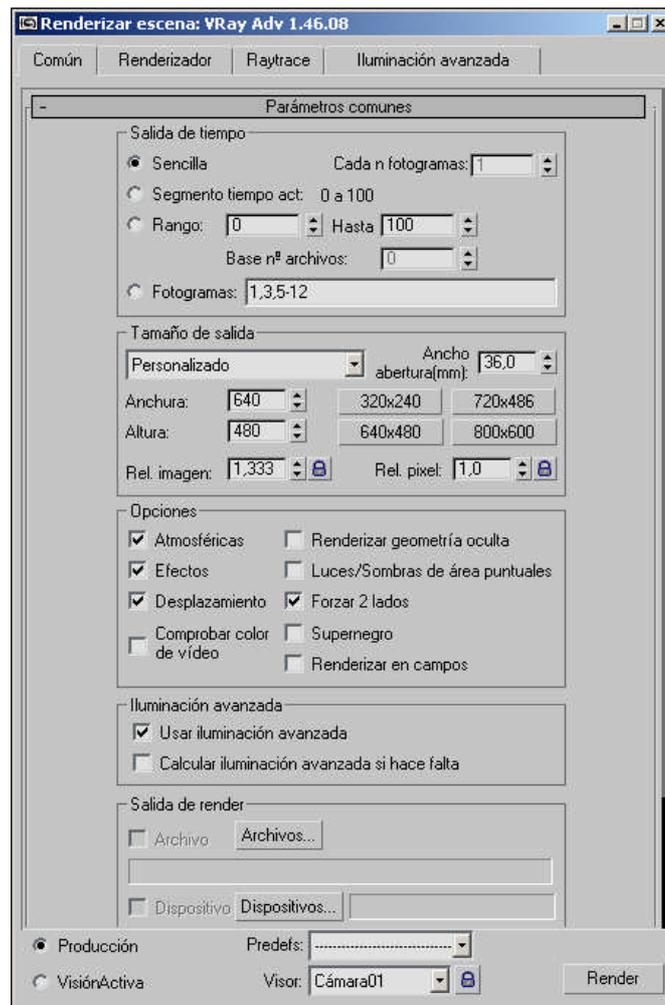
Todo esto se puede variar como se quiera, pero hay que fijarse bien en que aparezca el visor de la cámara del recorrido en Visor representación, en nuestro caso con el nombre Camara 01, y le damos al botón crear.

## 10. REPRESENTAR LA ESCENA

Ahora, queda sacar las imágenes finales del modelado tridimensional, tanto imágenes fijas como vídeos animados. Este es el proceso más importante y el que más tiempo lleva, ya que todo el trabajo previo realizado se dirige hacia este paso final, en el que se calcula el aspecto de las superficies, las sombras...etc, para dar a la

imagen su apariencia definitiva.

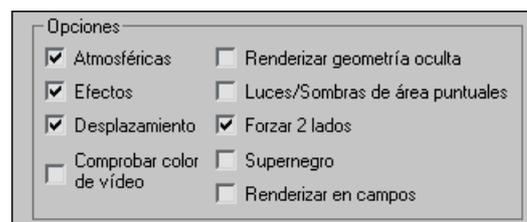
Si el resultado de la presentación preliminar fue satisfactorio, puedo crear la presentación definitiva. Para ello, activo la vista de la cámara del recorrido en uno de los visores. Ahora, uso las herramientas de representación de la barra de herramientas o bien utilizo el atajo **F10**.



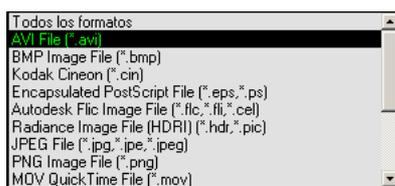
En la sección **Salida de tiempo**, selecciono:

- **Sencilla** si lo que pretendo es sacar una imagen fija, es decir, un solo cuadro.
- Si lo que quiero sacar es un vídeo animado, puedo seleccionar **Segmento de tiempo activo**, es decir, todo el rango de la animación. Si sólo quiero parte de los cuadros de la animación selecciono **Rango** e indico el cuadro inicial y el final de la animación.

En cuanto al **Tamaño de salida**, me he decantado por los 640x480 pixels para las imágenes fijas y por los 320x240 pixels para los vídeos. Este tamaño de 320x240 es bueno, ya que no es excesivamente grande y por tanto no habrá dificultad para reproducir el vídeo final en la mayoría de ordenadores. Además el tiempo de representación es mucho menor que eligiendo un tamaño mayor. Para las imágenes fijas es mejor sacar un tamaño mayor al 320x240, para así poder observar mejor los detalles de la imagen. En la sección Opciones de la ventana Representar escena, una opción importante es la de Forzar 2 lados. Es conveniente seleccionarla si los objetos de la escena provienen de importar un archivo, como es este caso, ya que si no puede ser que no se represente alguna cara de algún objeto. Activar la casilla Forzar 2 lados ralentiza algo la representación, pero no de una manera considerable, y así evito que algunas caras desaparezcan, sobre todo aquellas que están conformadas por regiones exclusivamente.

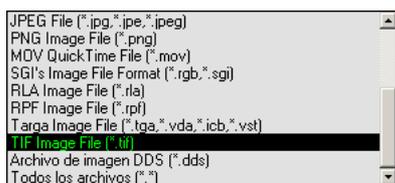


Para guardar los archivos que resultan del proceso introduzco un nombre y un formato para el archivo de salida de la representación. Para ello, dentro de **Archivos**, doy un nombre al archivo y selecciono AVI File



si lo que quiero es un vídeo.

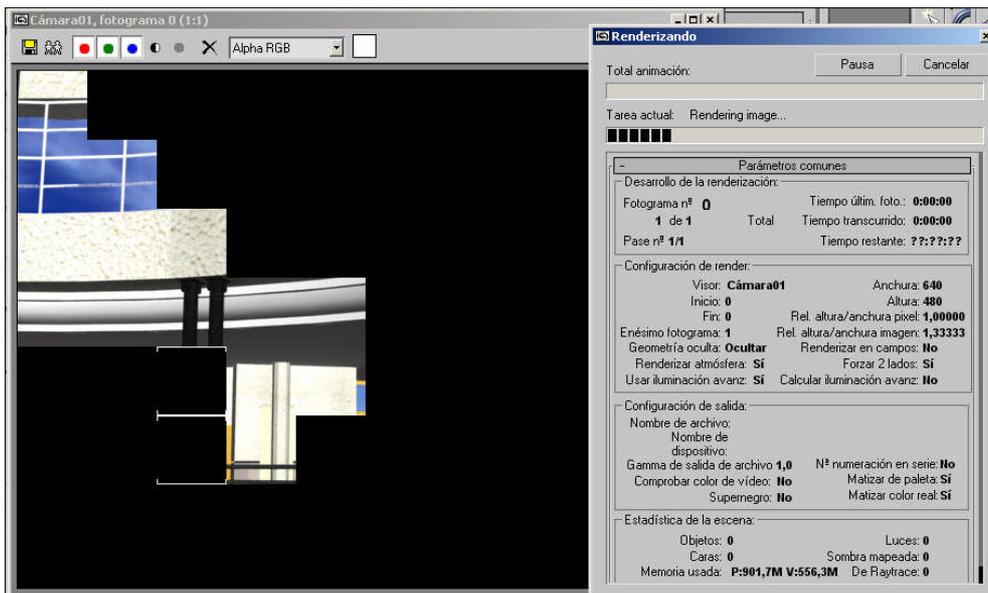
Si lo que quiero representar es una imagen fija, TIF Image o BMP.



El hecho de seleccionar el formato Tiff para las imágenes, es la sencilla razón de que su método de compresión no afecta de manera ostensible a la calidad de las imágenes, por lo cual, la visualización será más exacta a lo representado.

Al seleccionar el destino de vídeo, tenemos la opción de comprimir en cualquiera de los formatos expuestos. Aquí he elegido el DivX como destino final.

Ya sólo queda pulsar en el botón Representar de la ventana Representar escena. Nos aparecen dos ventanas, la de Búfer de cuadro virtual, donde aparecerá la imagen con el formato seleccionado, y la de Representando, donde podemos ver la barra de progreso de la representación, el cuadro que está representando, el tiempo estimado de representación, además de los parámetros antes seleccionados.



Cuando la renderización finalice, ya tendremos el archivo guardado en la ruta que le hayamos asignado anteriormente, en caso de que se trate de un vídeo. Si es una imagen lo podremos hacer desde el botón guardar (icono superior izquierdo en la imagen anterior).

Así podré obtener todas las imágenes y/o animaciones que quiera.

# TEMA 5: CREACIÓN DE IMÁGENES EN 360 GRADOS

## 1. GENERALIDADES

Las imágenes de 360° brindan una posibilidad, sólo mejorable por el streaming de audio y vídeo, excelente de agregar interactividad a una página web en la que se necesitan mostrar ambientes del mundo real, como habitaciones, show rooms, paisajes, etc. De manera que el visitante pueda hacerse una idea más realista que con una foto estática del ambiente que se desea mostrar por medio del ensamblaje de fotografías individuales generalmente.

## 2. ¿QUÉ SE NECESITA PARA CREAR IMÁGENES EN 360°?

Como en la mayoría de las aplicaciones informáticas, hay dos aspectos fundamentales para crear este tipo de imágenes: el hardware y el software.

Respecto del hardware necesario para tomar las fotografías, existen dos tipos. A saber, las cámaras digitales (la mejor elección) y el complemento cámara fotográfica de película / escáner.

Decimos que las cámaras digitales son la mejor opción debido a dos características fundamentales:

- Las cámaras digitales ya tienen como salida las imágenes en formato digital, por lo cual no se pierde calidad en el proceso de digitalización con un escáner.
- Las cámaras digitales modernas traen herramientas que facilitan la toma de imágenes 360°.

Respecto del software, se requieren las siguientes aplicaciones:

- Una aplicación que tome los distintos fotogramas y los ensamble de manera inteligente, de manera que conecte exactamente los bordes de los fotogramas consecutivos y el final del último con el principio del primero.
- Una aplicación que cree la imagen 360° mediante algoritmos que causan el efecto de perspectiva.
- Una aplicación que genere una forma de insertar la imagen 360° generada en una página web. Generalmente, las imágenes 360° se insertan a través de applets Java o de películas Quick Time.

Como la finalidad de todo el proceso, es llevar el conjunto en la medida de lo posible a un entorno Flash, me he decantado por una aplicación que permita exportar los archivos resultantes en formato \*.swf o lo que es lo mismo, el formato que utilizan las aplicaciones Flash; actualmente hay muchas aplicaciones que editen fotografías panorámicas, pero no todas manejan el soporte Flash, he optado por el programa Pano2QTVR, para la aplicación final y el PTGui para la creación de panoramas cilíndricos en formato imagen.

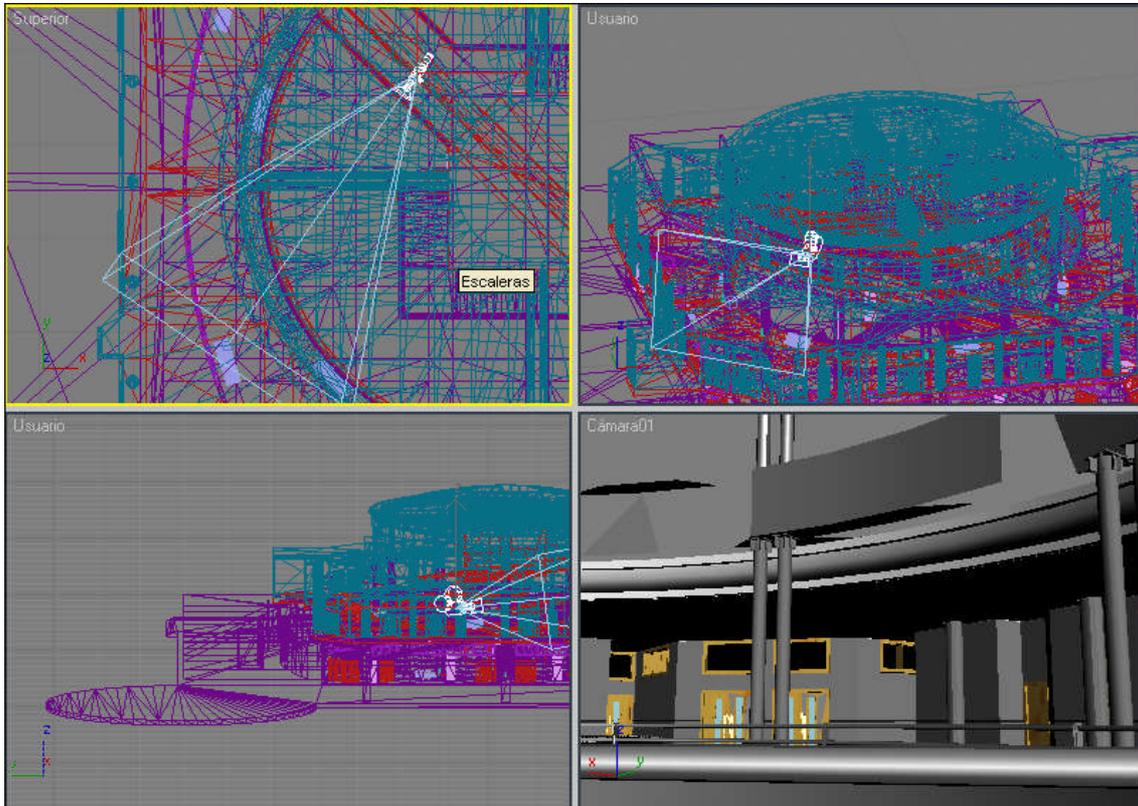
### **3. CREACIÓN DE IMÁGENES**

Como bien hemos dicho al principio de este capítulo, la creación de entornos en 360° se basa en imágenes sueltas o bien en una tira panorámica ya creada.

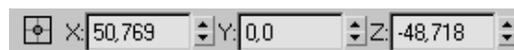
En nuestro caso, crearemos varios puntos de vista en el interior que estamos estudiando.

La forma de proceder es aparentemente simple:

1. Dentro de la aplicación del 3D Studio Max, elegiremos un punto de referencia para ser tratado con la vista panorámica. Esto se hará en base a la colocación de una cámara con objetivo apuntando hacia donde queramos.



2. Una vez hayamos fijado la situación de la cámara, haremos varias tomas desde ángulos diferentes cubriendo la totalidad de la circunferencia, los 360°. Me he decantado por tomar imágenes en intervalos de 20°, pues dependiendo de la abertura del foco (en este caso de 35°), es conveniente que haya superposición entre ellas. El proceso, se basa simplemente en seleccionar la cámara con el objetivo, y definiendo la **transformación en modo absoluto** , rotamos 20° cada vez que queramos sacar una imagen.



3. Así hemos obtenido 18 fotografías que posteriormente uniremos mediante la aplicación elegida.

Estas imágenes formarán una tira posteriormente que harán un efecto envolvente y permitirán visualizar todo el rango de visión desde el punto donde han

sido tomadas, de una manera estática.

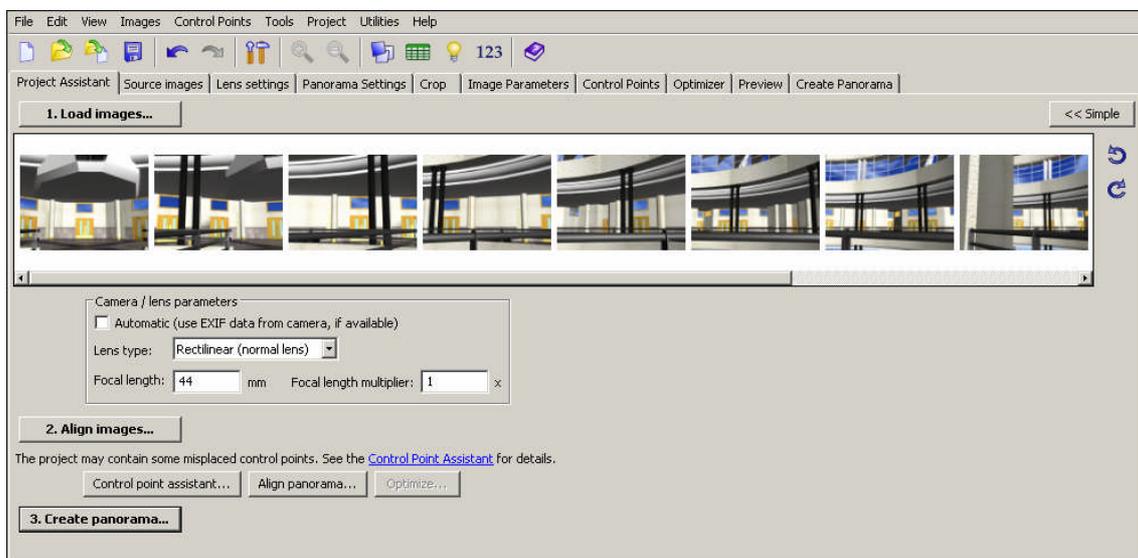
## 4. APLICACIÓN PANORÁMICA

### 4.1. CREACIÓN DEL PANORAMA CILÍNDRICO

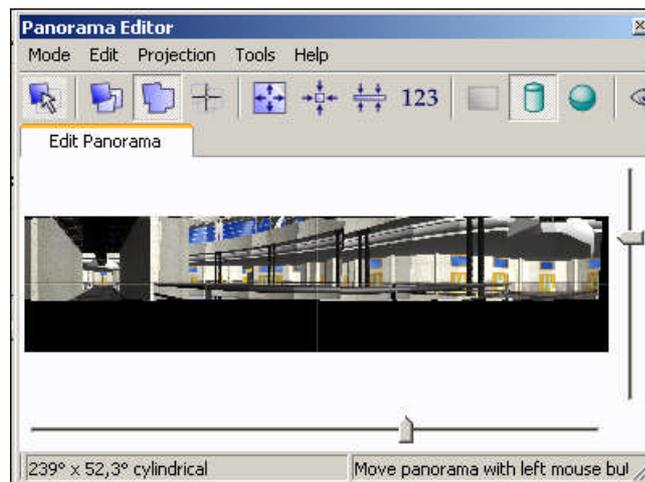
La manera más fácil de proceder, es haciendo primeramente una tira de las imágenes obtenidas.

Abriremos una aplicación que simula este efecto y que lleva por nombre PTGui.

Este programa, da múltiples posibilidades en cuanto al soldado de imágenes. Nosotros nos guiaremos por un asistente que nos de la opción de guiarnos fácilmente.



1. Cargamos las imágenes de la tira que queremos presentar.
2. Ordenamos en base a la escena que tengamos
3. Especificamos el tipo de lente y la abertura del foco, para ello echamos un vistazo en el 3D Studio MAX, en el panel modificar, mientras tenemos seleccionada la cámara.
4. Una vez asegurados estos pasos, ejecutamos **Align images...** para que nos cree el panorama en una tira.



5. Y finalizado esto creamos el panorama (**Create panorama...**), ajustando los valores de tamaño deseados, de tal manera que el resultado final tendría este aspecto.

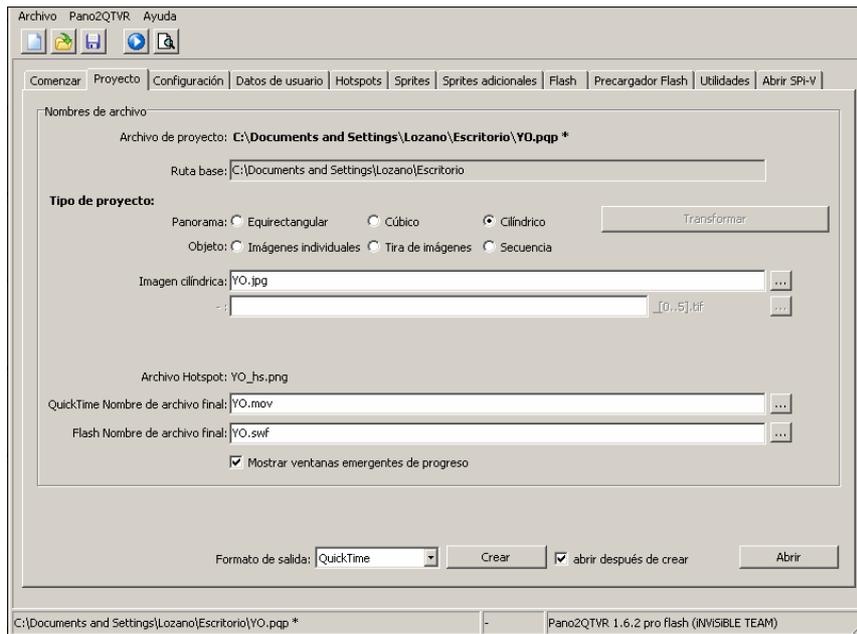


Guardamos la imagen, porque posteriormente va a ser importada al programa que creará el aspecto final.

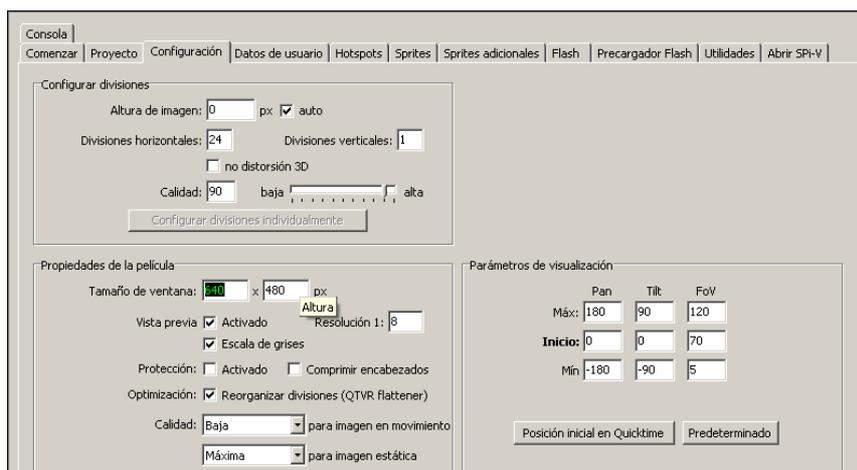
#### 4.2. CREACIÓN DE LA APLICACIÓN FLASH

Con la imagen creada anteriormente, vamos a dar forma a nuestro entorno esférico. Para ello vamos a ver los siguientes pasos.

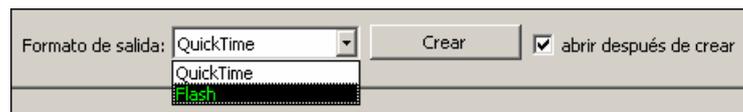
1. Localizamos el archivo de imagen panorámico anterior y lo arrastramos a la aplicación Pano2QTVR.
2. Se nos abrirá de forma automática con el archivo cargado.



3. Puesto que la imagen anterior, era de carácter cilíndrico, el la sección tipos de proyecto, le asignamos este carácter.
4. Hecho esto, simplemente con darle al botón crear, tendríamos nuestra aplicación en 360°, pero configuraremos un poco el tamaño y otros aspectos en la pestaña configuración. Como las fotos anteriores son de 640x480, convendría que el tamaño de la ventana tuviera estas mismas dimensiones, para ello le indicamos las dimensiones en el apartado correspondiente.

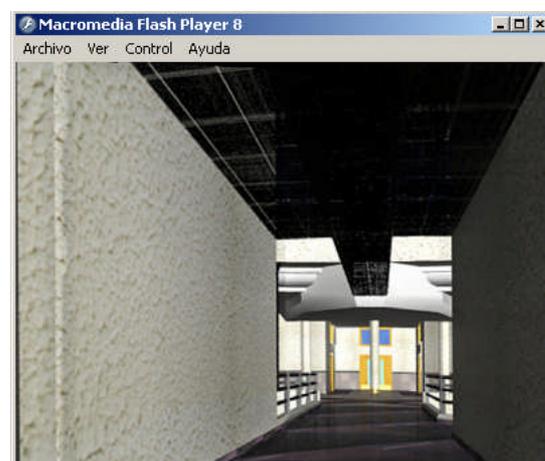


5. Ajustado el tamaño, ya podremos crear la salida final. Tenemos 2 formatos, QuickTime y Flash. Como dijimos anteriormente, el formato \*.swf es el que nos interesa para el entorno final, con lo que será este el que elijamos.



Finalmente tenemos la aplicación deseada en formato \*.swf, que nos permitirá incluirla en un entorno Flash para visualizar un entorno de 360° de una forma totalmente estática.

El aspecto final que presenta este tipo de aplicaciones en formato Flash (\*.fla) se muestra a continuación.



De la misma forma podremos obtener, el formato que por defecto usa Quick Time (.mov).



## TEMA 6: FLASH

### 1. INTRODUCCIÓN

La creciente popularidad de *Flash*, ha permitido que cada día, miles de nuevos diseñadores de páginas *Web*, adopten el uso de películas *Flash* en sus páginas personales o profesionales. En este último caso, los *Webmasters* son los que más se han decantado por la tecnología de *Flash*, ya que les permite crear contenidos impactantes que antes apenas podían trasladar a un simple boceto en el papel.

En la actualidad, esta popularidad y el crecimiento imparable de la adopción del *player* de *Flash* en casi todos los navegadores actuales, ha permitido, que otras muchas aplicaciones adopten la extensión *\*swf* como una más, en la ya larga lista de extensiones que es capaz de reconocer e interpretar un navegador como Explorer o *Nestcape*. Véase ficheros *Gif*, *Gif animados*, *JPG*, *PNG*, *Bmp*, *Applets de Java* y un largo etcétera. Pero no es esta la razón, la que le ha llevado a diversas aplicaciones gráficas y navegadores, a adoptar este estándar, sino la popularidad propia de *Flash*, es decir, la de la propia aplicación, la cual no deja de ser noticia cada día que pasa.

Para bien o para mal, *Flash* ha cambiado la cara de la web. La omnipresencia del plugin ha determinado que *Flash* se convierta en un entorno viable para el desarrollo de aplicaciones de contenido multimedia. En ese sentido, las experiencias creadas con *Flash* se benefician de la tecnología, más allá de los gráficos y animaciones vectoriales.

Brindarle al usuario la habilidad de interactuar con aplicaciones en el navegador es una de las razones primordiales por las cuales *Flash* se ha convertido en la herramienta preferida por diseñadores de interacción alrededor del mundo. Asimismo, la capacidad de presentar fácilmente diferentes segmentos de información sin necesidad de cambiar o “actualizar” la página, juega un rol importante en que cada vez más aplicaciones que trabajan con datos dinámicos sean desarrolladas en *Flash*.

Por otro lado, el soporte de video streaming y su creciente calidad en las

últimas versiones del reproductor de Flash, lo han convertido en el plugin de video mejor distribuido del mercado, siendo rápidamente adoptado frente a alternativas propietarias, que además son difíciles de ajustar a escenarios particulares.

En este contexto, *Macromedia*, desarrolladora de *Flash* y otros productos tan interesantes como *Dreamweaver 4* y *Fireworks 4*, por citar algunos, también ha adoptado la idea de extender el formato de *Flash* a otras de sus aplicaciones de éxito como puedan ser *Freehand* o *Illustrator*.

## 2. QUÉ ES FLASH

Flash es un software originalmente diseñado para crear animaciones que se pueden usar en páginas web. Usa vectores gráficos que son ideales para la web, porque son ligeros y no consumen muchos recursos.

En las nuevas versiones de Flash, Macromedia (principal compañía puntera) ha hecho Flash más y más controlable; esto es usando nuevas herramientas de programación, lo que lo tiene posicionado ya como un competidor para el HTML, en la creación de web sites interactivos y aplicaciones como e-commerce store. Flash ofrece las siguientes ventajas en lugar de HTML por las siguientes razones:

- Las películas Flash cargan más rápido lo que ahorra tiempo de descarga; y esto es porque Flash es basado en vectores a diferencia de HTML.
- Flash inteligentemente almacena en caches sus películas, para que no necesiten ser cargadas de nuevo.
- Flash da a los usuarios (Persona viendo/usando la película Flash) una 'rich-client' experiencia.

Las páginas Flash pueden ser hechas para cargar más rápido, pero la mayor parte del tiempo, en la realidad no lo hacen y esto es por la forma en que son diseñadas, de manera ostentosa y pesada.

El HTML también almacena las páginas tan pronto las imágenes son descargadas, y estas se almacenan en el caché de los navegadores, para luego ser

usadas en lugar de bajarlas otra vez del servidor.

Con la nueva tecnología como ASP.net y Java Server Faces, HTML puede ser similar a 'rich-client' application. Incluso sin estas nuevas herramientas, y cuando es apropiadamente diseñado; el HTML para la mayoría de los sitios dinámicos pueden proveer una buena experiencia para el usuario.

### 3. QUÉ ES UN CONTENIDO DINÁMICO Y UNA PELÍCULA FLASH

Un contenido dinámico de *Flash*, no es más que una película que permite mostrar eventos multimedia y a la vez permitir interactuar con lo que se está viendo en ese momento. Esto se consigue añadiendo elementos en la película como botones interactivos o clips de película. Esto significa que *Flash* permite aplicar acciones a dichos elementos. Una acción es la suma de un número indeterminado de sentencias o eventos.

*Flash* permite aplicar una gran variedad de sentencias a dichos elementos. Estas sentencias, más próximas a un lenguaje de programación por medio de scripts, nos permitirán crear acciones que requieran de respuestas por nuestra parte.

Por otro lado una película Flash es el contenido completo de un trabajo que ha hecho con este programa. Mas adelante, esta película, podrá ser reproducida en el player de *Flash* o en un reproductor habitual que permita este formato.

Dicha película podrá ser distribuida a través de Internet, un CD o un simple *Disquete*. *Flash* permite además, distribuir una película en modo proyector, esto significa, que podremos distribuir ese trabajo con el player de *Flash*.

### 4. QUÉ ES UN SCRIPT

El **guión** o **archivo de procesamiento por lotes** (en inglés "script") es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión. El uso habitual de los guiones es realizar

diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de éste tipo de programas.

Se diferencian de los programas de aplicación principalmente en la complejidad.

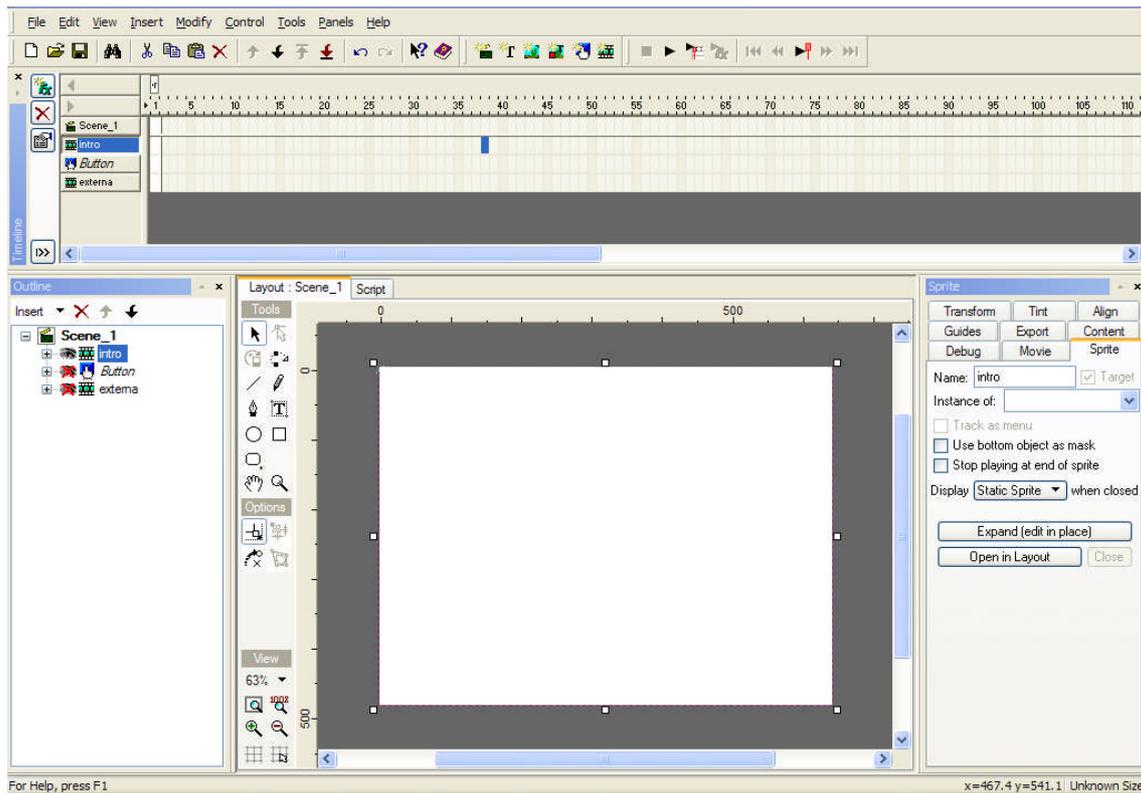
La Interfaz de Programación de Aplicaciones de Adobe Flash está basada en Javascript-C, es decir, los comandos de C++ no se interpreta directamente desde C/C++, sino con Javascript, es decir, da mayor flexibilidad al desarrollador en vista a la ampliación o personalización de la aplicación.

## **5. PROCESO**

Para la realización de este proyecto, nos hemos apoyado en programas tipo, como puedan ser SwisHMAX y Macromedia Flash 8.

El objeto de su manejo, no es más que crear una aplicación en la que se engloben los pasos seguidos hasta ahora para crear una aplicación sencilla que muestre de alguna manera las posibilidades de estos programas.

Empezaremos observando la interfaz del SwisHMAX; su funcionamiento es a base de capas, de tal manera que cada cual controla el momento en que ha de mostrarse mediante una barra de tiempos que se muestra en la parte superior.



La Línea de tiempo organiza y controla el contenido de una película a través del tiempo, en capas y fotogramas. Los componentes principales de la Línea de tiempo son las capas, los fotogramas y la cabeza lectora.

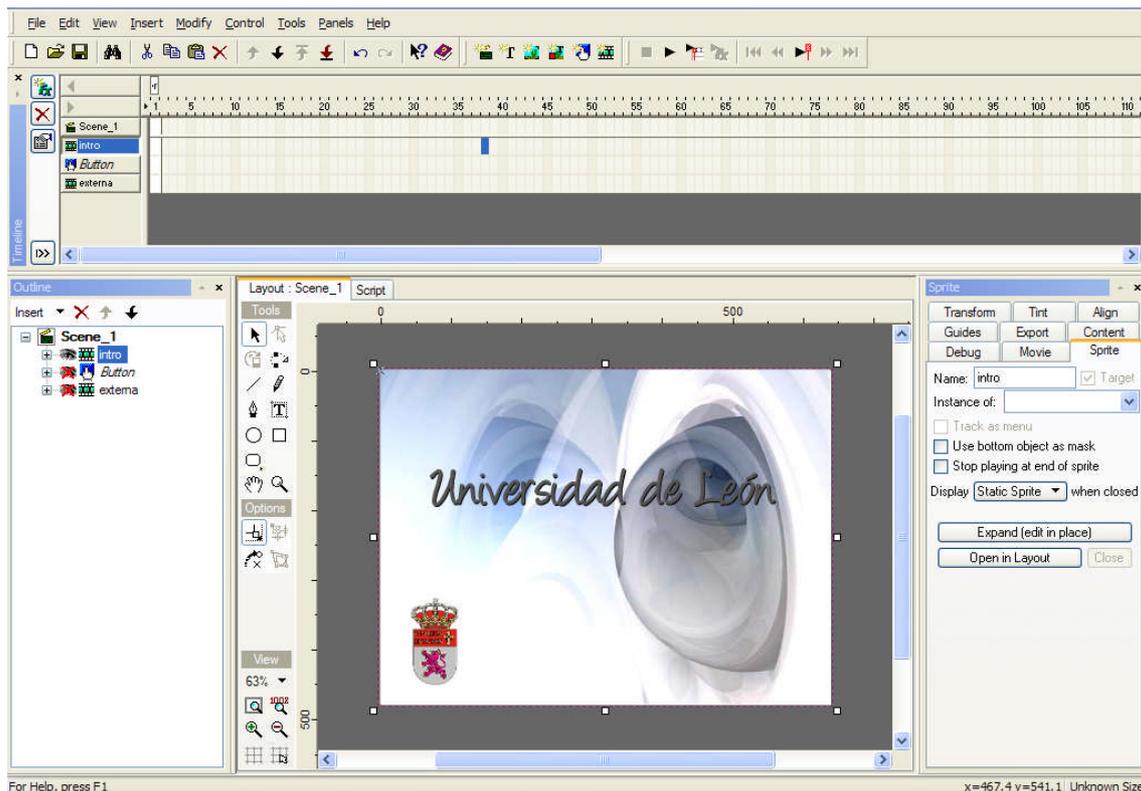
Las capas de una película aparecen en una columna situada a la izquierda de la Línea de tiempo. Los fotogramas contenidos en capa cada aparecen en una fila a la derecha del nombre de la capa. El encabezado de la Línea de tiempo situado en la parte superior de la Línea de tiempo indica los números de fotogramas. La cabeza lectora indica el fotograma actual que se muestra en el Escenario.

La información de estado de la Línea de tiempo situada en la parte inferior de la Línea de tiempo indica el número de fotograma actual, la velocidad de fotogramas actual y el tiempo transcurrido hasta el fotograma actual.

Por otra parte, nosotros podremos cambiar la manera según la que se muestran los fotogramas y mostrar miniaturas del contenido de los fotogramas en la Línea de tiempo. La Línea de tiempo muestra dónde hay animación en una película, incluyendo la animación fotograma a fotograma, la animación interpolada y los trazados de movimiento.

En primer lugar vamos a crear la primera capa, que no será más que una especie de introducción.

Para ello seleccionamos el tamaño de diapositiva que queremos tomar y le asignamos la capa correspondiente, en este caso la 1. Al ser una diapositiva, contendrá una imagen que mostrar, con lo cual arrastramos aquella que nos interese o la creamos.



Flash nos permite además, insertar efectos de transición y otros, por medio de scripts predeterminados que ya vienen incorporados en el programa.

Si lo que queremos es que haya un salto diferente de una imagen a otra, por medio de otra pantalla, asignaremos en este caso una transición, que se habrá llamado **Blur**.

Todos los efectos en Flash pueden tener la duración que uno desee, con tal de controlarlo en la barra de tiempos.

La cabeza lectora se mueve por la Línea de tiempo para indicar el fotograma que se muestra en cada momento en el Escenario. El encabezado de la

Línea de tiempo muestra los números de fotograma de la animación. Para que aparezca en el Escenario un determinado fotograma, puede mover la cabeza lectora hasta ese fotograma en la Línea de tiempo.

Para ir a un fotograma, haré clic en la posición del fotograma en el encabezado de la Línea de tiempo o bien arrastro la cabeza lectora hasta la posición deseada.

Para esta diapositiva concretamente, hemos aplicado 3 efectos diferentes.

1. Blur: Crea el efecto de transición deseado de una diapositiva a otra. En nuestro caso se le ha asignado 26 frames.
2. Zoom: Permite que el texto vaya apareciendo gradualmente en la pantalla.
3. Fade In: Otro modo de aparecer en pantalla. Con una duración de 10 frames.

La base de cualquier diapositiva en nuestro caso, es ir aplicándole capas y configurándolas mediante la línea de tiempos.

Para la segunda diapositiva, aplicaremos el mismo procedimiento, de tal manera que se muestre el título de nuestro proyecto para pasar a otra imagen mediante otra transición.

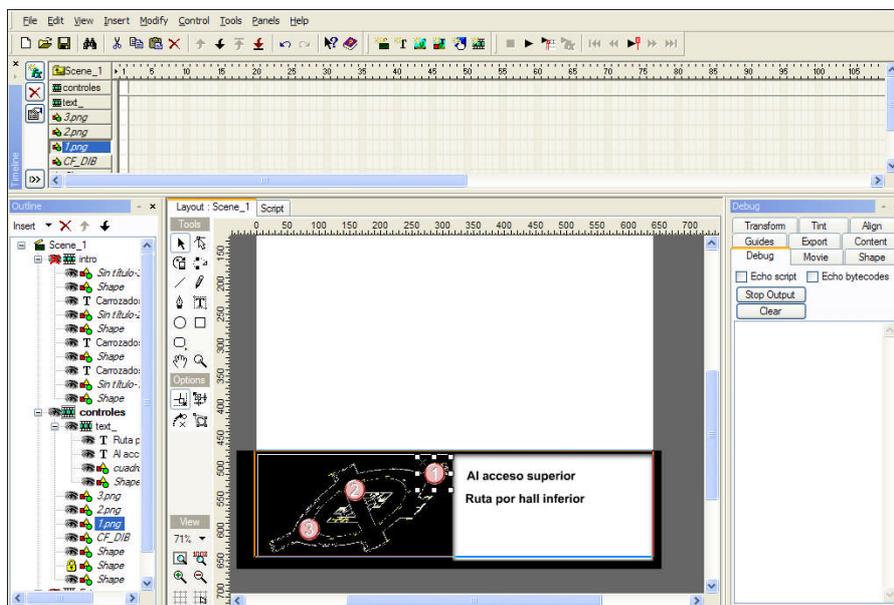


Para la tercera diapositiva, se ha optado por crear una combinación de espacios en los que se alojarán diferentes elementos.

Al ser una especie de paseo virtual por el centro, se ha optado por mostrar:

- En primera instancia, una panorámica en 360° del punto en el que nos encontramos.
- Seguidamente, accederemos a otros sitios mediante la posibilidad de hacer clic en la opción deseada.
- Una vez hecha la elección, tendremos un vídeo que mostrará el paseo hasta donde el visor tiene la oportunidad de ir.

Para conseguir todo esto en Flash hemos creado anteriormente los archivos de vídeo y de realidad virtual (panorámicas 360°), para que podamos enlazarlas en formato Flash.

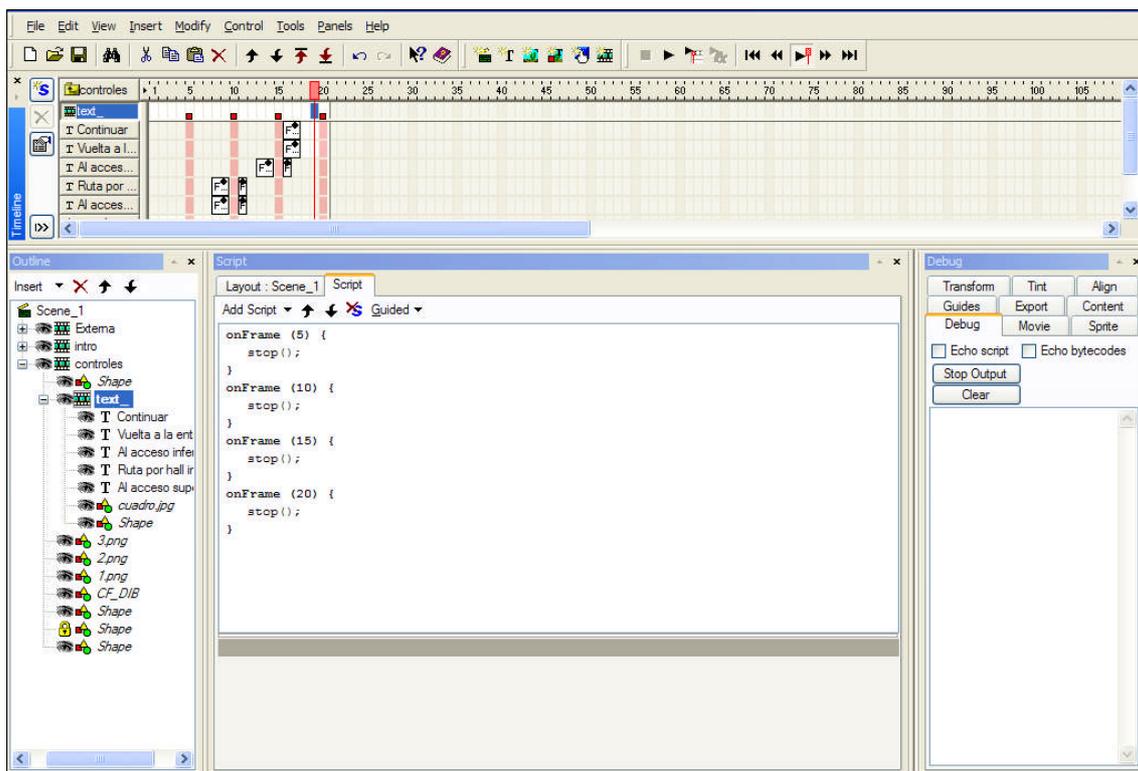


En la parte inferior izquierda, se ha situado una imagen en planta de la parte de abajo del Edificio Tecnológico Fase – 1.

En él, se han colocado tres puntos que serán los spots hacia donde nosotros podremos ir haciendo la visita por el recorrido.

Cada punto marcado como 1, 2 o 3, tiene asignada una función, es decir, se ha puesto un script a cada número para que cuando nosotros situemos el ratón encima, aparezca la una opción en el recuadro blanco de la derecha. Al igual que esto, cuando se haga clic con el puntero, sobre una de las opciones, nos llevará a otra diapositiva en la que se verá el vídeo de muestra del recorrido.

Todo este proceso, lo realizaremos mediante Scripts aplicado a esos puntos, asociándoles el texto correspondiente.



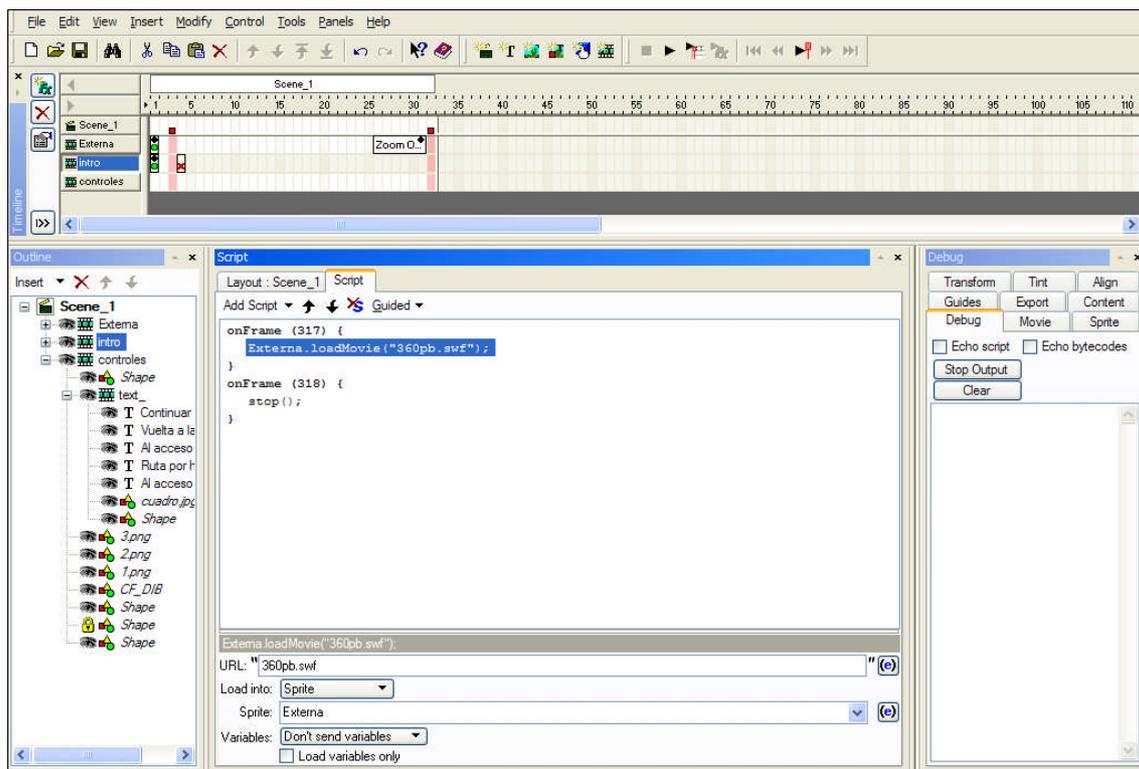
Es evidente, que como anteriormente dijimos, se han asignado una duración de frames a cada evento de tal manera, que la duración de muestra sea la adecuada.

La parte inferior derecha, consta de un espacio asignado al texto que mostrará los recorridos por las plantas. Como ejemplo se ha puesto el perteneciente a la inferior desde la entrada, que da dos opciones:

- Al acceso superior, en el que se mostraría un vídeo hasta el descansillo de la escalera.
- Ruta por hall inferior.

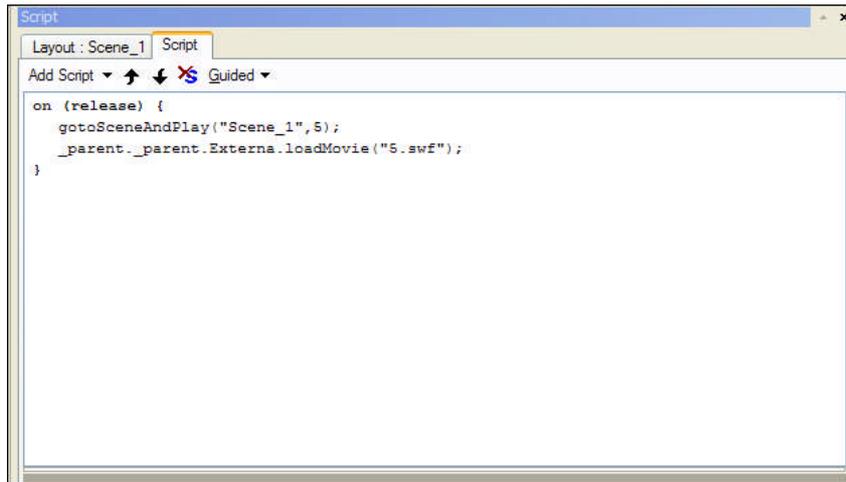
Podremos poner todo el contenido que queramos a estos espacios, y los efectos que creamos oportunos.

Para el hueco que queda en a parte central de la diapositiva, se ha optado por poner en primera instancia una visión general del punto en el que estamos por medio de una panorámica 3D, que se agregará como objeto de realidad virtual en nuestro caso, llamándole por medio del script siguiente.



Esto hará el efecto de cargar la vista en 360° en un primer momento, y después podremos elegir el sitio al que queremos desplazarnos, con sólo el hecho de presionar las opciones designadas en el cuadro blanco de la parte inferior derecha citado anteriormente.

El vínculo del vídeo se hará por medio de otro script, al que llamaremos de una forma externa para que se muestre a continuación.



Una vez creados los pasos hasta aquí descritos, no habrá más que aplicar los mimos, de una manera general para el resto de contenidos.

Para alargar la duración de un fotograma clave, arrastraremos con la tecla Alt. presionada el fotograma clave hasta el fotograma final de la nueva duración de la secuencia.

Para copiar un fotograma clave o secuencia de fotogramas mediante la operación de arrastre, haremos clic con el ratón, y presionando Alt, y arrastramos el fotograma clave a la nueva posición.



Para copiar y pegar un fotograma o una secuencia de fotogramas, selecciono el fotograma o la secuencia y elijo **Edición/ Copiar fotogramas** para posteriormente pegar los fotogramas.

Una vez obtenido el proceso, guardaremos el archivo para reproducirlo en cualquier momento.

## CONCLUSIONES

La finalidad para la cual se ha tratado este trabajo, no es más que mostrar las opciones posibles de la representación tridimensional para un conocimiento en forma de paseo virtual, de proyectos de ingeniería y arquitectura por medio de diferentes aplicaciones y caminos variados.

Las posibilidades de representación tridimensional de edificios usando AutoCAD y 3D Studio Max son infinitas. Se puede dar a conocer un realismo muy aproximado a la actualidad simulando todo tipo de objetos con iluminación avanzada, materiales, etcétera. Además una vez terminado el trabajo, hacer cualquier modificación resulta fácil y rápido si se dispone del equipo necesario para una experiencia fluida.

En primer lugar se tenía en mente proyectar de una manera en la que se pudiera interactuar con el medio de la forma más cercana posible, esto es, mediante un motor 3D, representando la escena en primera persona. Pero debido a los contratiempos de programación, el amplio uso que éstos hacen de ello y las carencias de su apartado gráfico, se optó por otros caminos más simples pero no menos útiles.

La tecnología de realidad virtual o VR como actualmente se la conoce, por medio de panorámicas prediseñadas, nos brinda la posibilidad de tener conocimiento del entorno con el menor esfuerzo posible y así obtener una concepción in situ del lugar observado.

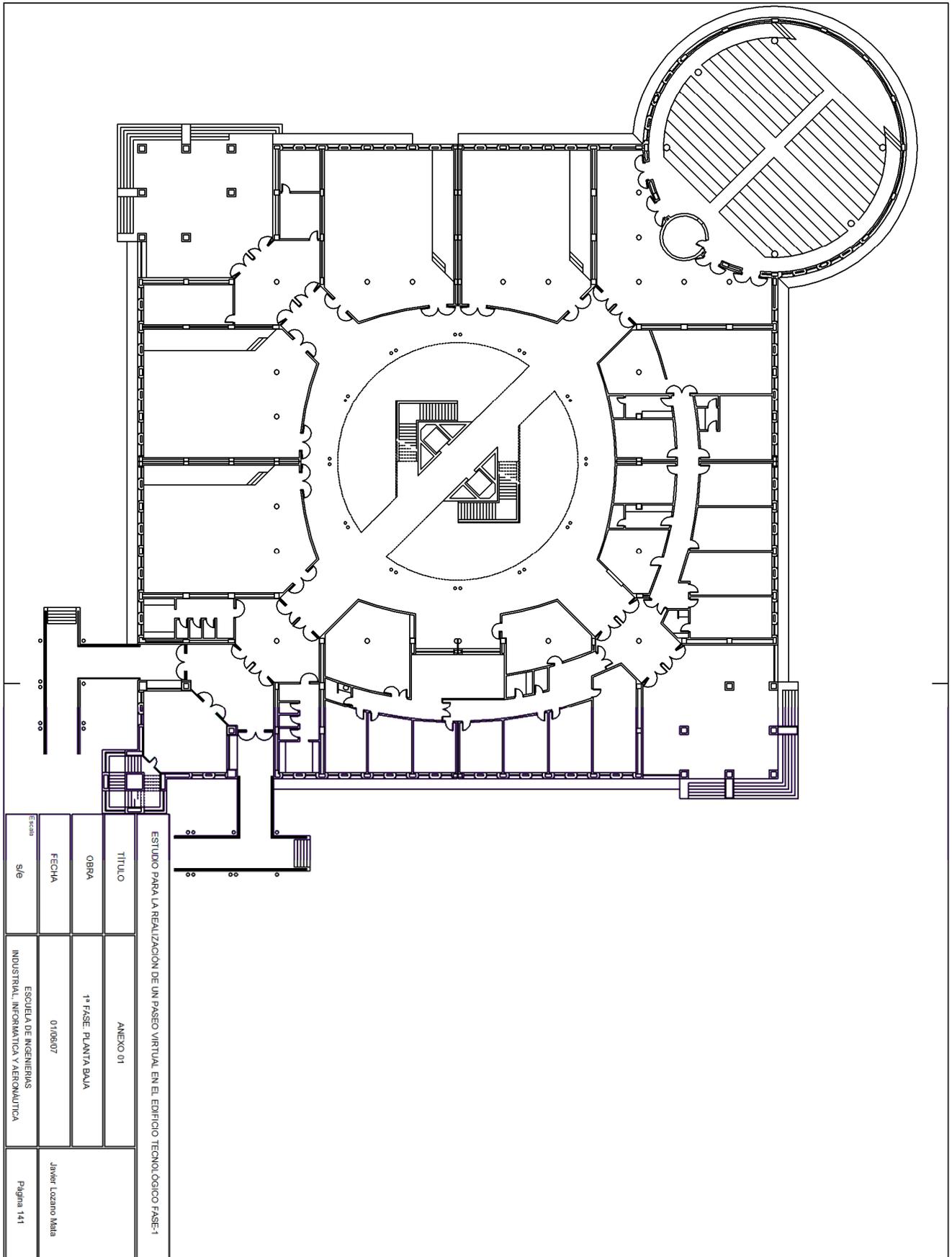
Como colofón a todo esto, habremos de decir, que son innumerables las formas de compatibilidad que hay entre todas las aplicaciones tratadas.

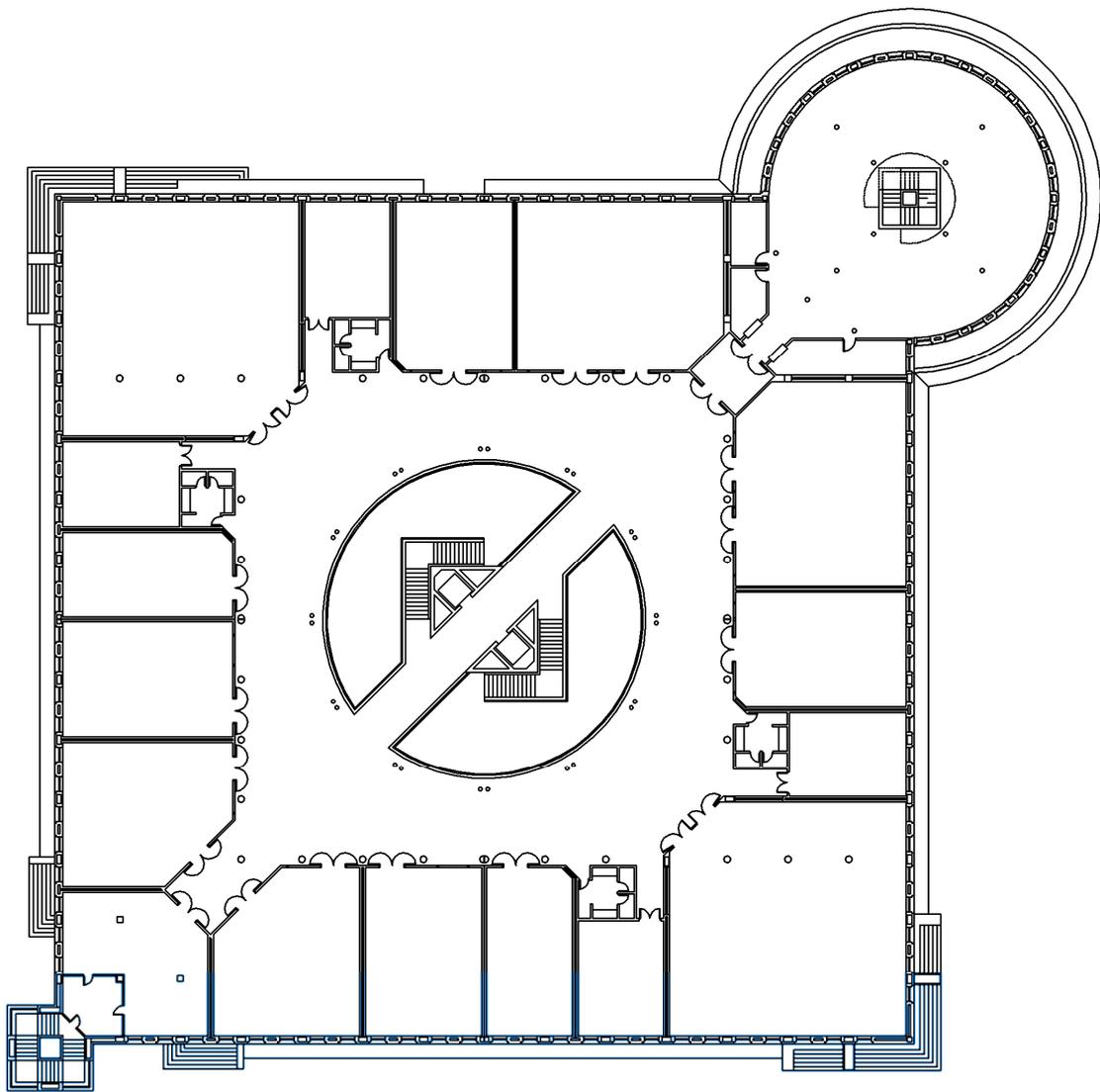
Han sido de gran ayuda los programas que se han dado a conocer en este proyecto, ya que hoy en día resultan útiles en cualquier campo de la ingeniería. Espero que el proyecto realizado deje entrever las posibilidades que tenemos a nuestro alcance, y de esta manera contribuir en cierto modo a la visualización de una pequeña parte del Edificio Tecnológico Fase -1.

Para finalizar, animo a todos aquellos que muestren interés por el campo del diseño asistido por ordenador a que avancen en este terreno porque además de ser

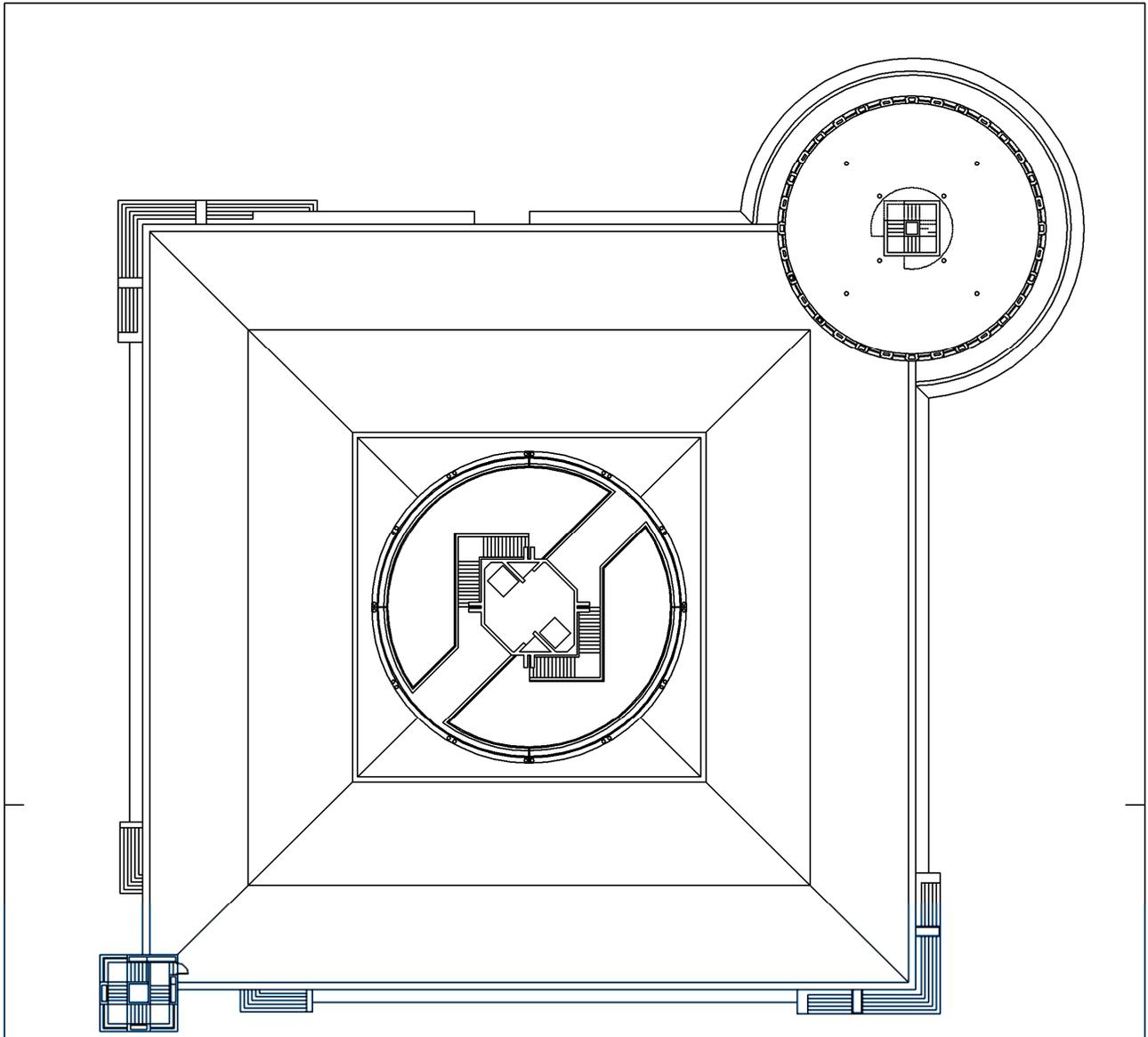
necesario hoy en día, no dejará de ser una herramienta de creación para un futuro inmediato.

## **ANEXO**





ESTUDIO PARA LA REALIZACIÓN DE UN PASEO VIRTUAL EN EL EDIFICIO TECNOLÓGICO FASE-1		
TÍTULO	ANEXO 02	
OBRA	1ª FASE. PLANTA PRIMERA	
FECHA	01/06/07	
Escala	INDUSTRIAL, INFORMÁTICA Y AERONÁUTICA	Javier Lozano Mata
s/e		Página 142



ESTUDIO PARA LA REALIZACIÓN DE UN PASEO VIRTUAL EN EL EDIFICIO TECNOLÓGICO FASE-1		
TÍTULO	ANEXO 03	
OBRA	1ª FASE. PLANTA SEGUNDA	
FECHA	01/06/07	Javier Lozano Mata
Escal	ESUELA DE INGENIERIAS INDUSTRIAL, INFORMATICA Y AERONAUTICA	Página 143
S/e		

---

## BIBLIOGRAFÍA

- BOARDMAN, TED. (2004). "3DS MAX 6". Ed. ANAYA MULTIMEDIA. Madrid.
- FEIJÓ MUÑOZ, JESÚS. (1992). "Proyectos Arquitectónicos Asistidos por Ordenador". UNIVERSIDAD DE VALLADOLID.
- HERNÁNDEZ, CLAUDIO. (2001). "Flash 5". Ed. ANAYA MULTIMEDIA. Madrid.
- TICKOO, SHAM (2000). "AutoCAD 2000 Básico". Ed. PARANINFO. Madrid.
- TICKOO, SHAM (2000). "AutoCAD 2000 Avanzado". Ed. PARANINFO. Madrid.
- WILSON, JOHN E. (2002). "Modelado 3D con AutoCAD". Ed. ANAYA MULTIMEDIA. Madrid.