

# UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA  
CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS  
DE INFORMACIÓN Y SISTEMAS  
LICENCIATURA EN SISTEMAS  
COMPUTACIONALES

Desarrollo de Mundos Virtuales en el Área de Más  
Sobre Ciencia del Museo Virtual 3D el Rehilete

**T E S I N A**  
PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN SISTEMAS COMPUTACIONALES

PRESENTA:

**MIGUEL ÁNGEL CRUZ RODRÍGUEZ**

ASESOR:

**DR. OMAR LÓPEZ ORTEGA**

COASESORES:

**M. En C. GONZALO ALBERTO TORRES SAMPERIO**  
**DR. GUSTAVO NÚÑEZ ESQUER**





Carretera  
Pachuca-Tulancingo Km. 4.5  
C.P. 42073  
Tel. (01 771) 717 20 00  
Ext. 6301 6302  
Fax (01 771) 717 21 09

# UNIVERSIDAD AUTÓNOMA DEL ESTADO DE ITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

P.D.LS.C. MIGUEL ÁNGEL CRUZ RODRÍGUEZ

PRESENTE

Por este conducto le comunico que el Jurado que le fue asignado a su trabajo de titulación denominado "**DESARROLLO DE MUNDOS VIRTUALES EN EL ÁREA DE MÁS SOBRE CIENCIA DEL MUSEO VIRTUAL 3D EL REHILETE**" y que después de revisarlo en reunión de sinodales han decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación se anotan las firmas de conformidad de los integrantes del Jurado:

PRESIDENTE:	M. EN C. G. ALBERTO TORRES SAMPERIO
PRIMER VOCAL:	Dr. OMAR LÓPEZ ORTEGA
SEGUNDO VOCAL:	Dr. GUSTAVO NUÑEZ-ESQUER
TERCER VOCAL:	M. EN C. FELIX A. CASTRO ESPINOZA
SECRETARIO:	M. EN C. ARTURO CURIEL ANAYA
PRIMER SUPLENTE:	DR. VIRGILIO LÓPEZ MORALES
SEGUNDO SUPLENTE:	LIC. EN COMP. LUIS ISLAS HERNÁNDEZ

Sin otro particular, reitero a usted la seguridad de mi atenta consideración.

ATENTAMENTE.  
"AMOR, ORDEN Y PROGRESO"  
Pachuca, Hgo., a 11 de noviembre de 2003

Lic. en Comp. Luis Islas Hernández.  
Coordinador de La Licenciatura  
En Sistemas Computacionales



*Quiero expresar mi más profundo  
agradecimiento...*

*A Dios y a mi madre por que ambos me dieron la oportunidad de vivir y me han permitido estar en el  
fugar que me encuentro.*

*A Carne, tíos, hermanos y a todas aquellas personas por el apoyo brindado en la realización de este  
trabajo.*

*A mis amigos, que con su apoyo y amistad logramos lo que algún día planeamos.*

*A mi asesor y coasesores por su apoyo brindado.*

*A todos ellos por ayudarme a cumplir uno de mis sueños.....*

*Por siempre gracias.*

*Miguel Ángel.*

# ÍNDICE

	Página.
Prólogo.	I
Problemática.	II
Objetivo General.	III
Objetivos Específicos.	IV
Justificación.	V
Limitaciones.	VI

## Capítulo 1: Introducción.

1.1.- Realidad Virtual.	1
1.2.- Lenguajes y librerías para generar gráficos 3D.	1
1.2.1.- Java 3D.	2
1.2.2.-OpenGL	2
1.2.3.-Cult3d.	3
1.2.4.-X3D.	3
1.3.- Paseos Virtuales.	3
1.4.- Museos tradicionales.	4
1.5.- Museos virtuales.	4
1.6.- Museo el Rehilete.	5
1.7.- Requerimientos de hardware para visualizar la página.	5
1.8.- Requerimientos de software para visualizar la página.	6
1.9.- Software requerido para el desarrollo.	6
1.10.- Lenguajes requeridos para el desarrollo.	7

## Capítulo 2: Lenguajes y Herramientas de Desarrollo.

2.1.-VRML	8
2.2.-3D Studio Max.	11
2.3. - Internet Space Builder.	11
2.4.-Macromedia Flash.	12
2.5.-JavaScript.	12
2.6.-HTML	12
2.7.-VrmlPad.	12

## Capítulo 3: Descripción del Método Empleado.

3.1.- Tipologías de métodos.	14
3.2.- Etapas del método.	15
3.2.1- Recopilación de información.	15
3.2.2.-Análisis.	15
3.2.3.-Diseño.	16
3.2.4.- Desarrollo de las exhibiciones.	16
3.2.5.-Integración.	17
3.2.6.- Pruebas de funcionamiento y tiempos de descarga.	17
3.2.7.-Validación.	17

## Capítulo 4: Desarrollo e Integración de Mundos Virtuales con Comportamiento Complejo.

4.1-Recta numérica.	19
4.1.1.- Desarrollo de la exhibición.	19
4.1.1.1.-Modelado 3D.	19
4.1.1.2.- Comportamiento complejo.	25
4.1.2.-Pruebas.	28
4.1.3.- Tiempo de descarga.	31
4.1.4.- Exhibición "Recta Numérica" integrada.	31
4.2.- De Plantas, Aire, Agua y el Sol.	32
4.2.1.- Desarrollo de la exhibición.	32
4.2.1.1.-Modelado 3D.	32
4.2.1.2.- Comportamiento complejo.	36
4.2.2.- Pruebas.	37
4.2.3.- Tiempo de descarga.	38
4.2.4.- Exhibición "De Plantas, Aire, Agua y el Sol" integrada.	39
4.3 Torre de Hanoi.	39
4.3.1.- Desarrollo de la exhibición.	39
4.3.1.1-Modelado 3D.	40
4.3.1.2.- Comportamiento complejo.	41
4.3.2.- Pruebas.	44
4.3.3.- Tiempos de descarga.	46
4.3.4.- Exhibición "Torre de Hanoi" integrada.	46

## **Capítulo 5: Desarrollo e Integración de Mundos Virtuales con Animaciones.**

5.1 Gira, gira.	48
5.1.1. Desarrollo de la exhibición.	48
5.1.1.1.- Modelado 3D.	48
5.1.1.2.- Adición de animación.	50
5.1.2.-Pruebas.	51
5.1.3.- Tiempo de descarga.	52
5.1.4.- Exhibición "Gira, gira" integrada.	52

## **Capítulo 6: Desarrollo e Integración de Mundos Virtuales (Visualizaciones).**

6.1.- Meteoritos.	54
6.1.1. Desarrollo de la exhibición.	54
6.1.1.1.- Modelado 3D.	54
6.1.1.2.- Texturizar meteoritos.	56
6.1.2.- Tiempo de descarga.	57
6.1.3.- Exhibición "Meteoritos" integrada.	58
6.2.- Calidoscopio.	58
6.2.1. Desarrollo de la exhibición.	58
6.2.1.1.- Modelado 3D.	59
6.2.1.2.- Simular espejo.	59
6.2.2.- Tiempos de descarga.	61
6.2.3.- Exhibición "Caleidoscopio" integrada.	61

## **Capítulo 7: Conclusiones y Perspectivas.**

7.1.- Conclusiones.	62
7.2.- Perspectivas.	63
Referencias	64
Glosario	67

## ÍNDICE DE FIGURAS

	Página
<b>Capítulo 2:</b>	
Fig. 2.1 programa que utiliza prototipos.	10
<b>Capítulo 4:</b>	
Fig. 4.1 Código para activar el PlaneSensor.	20
Fig. 4.2. Icono de 3D Studio Max para crear una caja.	20
Fig. 4.3 Visualización de una caja en 3D Studio Max.	21
Fig. 4.4 Vista de la caja para formar la estructura de la Recta Numérica.	21
Fig. 4.5 Icono para asignar color (Barra de herramientas derecha).	22
Fig. 4.6 Paleta para asignar color al objeto.	22
Fig. 4.7 Modelado de la estructura de la Recta Numérica.	23
Fig. 4.8 Menú para crear un texto.	23
Fig. 4.9 Menú para localizar el menú para extruir.	24
Fig. 4.10 Modelo 3D de la Recta Numérica.	24
Fig. 4.11 Variables.	25
Fig. 4.12 Segmento del programa que realiza los cálculos de suma.	25
Fig. 4.13 Segmento del programa que realiza los cálculos de resta.	26
Fig. 4.14 Rutas en VRML con Java Script.	26
Fig. 4.15 Código para mantener los objetos estáticos frente a la cámara.	27
Fig. 4.16 Recta Numérica.	28
Fig. 4.17 Control de movimiento del objeto "CONEJO".	28
Fig. 4.18 Nueva posición del objeto "Conejo" después de realizar una suma.	29
Fig. 4.19 Cambio de valores del String, que muestra la operación que realizo el usuario.	29
Fig. 4.20 Nueva posición del objeto "Conejo" después de realizar una suma.	29
Fig. 4.21 Panel de control de la Recta Numérica."	29
Fig. 4.22 Lado "A" de la Recta Numérica.	30
Fig. 4.23 Control de movimiento del "Conejo".	30
Fig. 4.24 Lado "B" de la Recta numérica.	30
Fig. 4.25 Control de movimiento del "Grillo".	31
Fig. 4.26 Exhibición "Recta Numérica" integrada.	32
Fig. 4.27 Caja para iniciar a modelar.	33

	Página
Fig. 4.28 Colocación de la segunda caja para hacer el corte.	33
Fig. 4.29 Menú para localizar los "Objetos de composición".	34
Fig. 4.30 Menú de "Objetos de composición" donde se encuentran las operaciones booleanas.	34
Fig. 4.31 Forma del objeto después de realizarse una operación booleana.	35
Fig. 4.32 Localización comando "Extruir".	35
Fig. 4.33 Tablero terminano.	36
Fig. 4.34 Exhibición terminada(Exhibición De Plantas, Aire, Agua y el Sol).	36
Fig. 4.35 Función para capturar posición de un objeto.	36
Fig. 4.36 Segmento de código, que determina cuando los discos se encuentran en la posición correcta.	37
Fig. 4.37. Exhibición de la Fotosíntesis.	38
Fig. 4.38. Exhibición de la Fotosíntesis tras interacción con el usuario.	38
Fig. 4.39 Exhibición "De Plantas, Aire, Agua y el Sol" integrada.	39
Fig. 4.40 Modelo 3D de disco para la Torre de Hanoi.	40
Fig. 4.41 Modelo 3D de la cubierta.	40
Fig. 4.42 Exhibición "Torre de Hanoi" terminada.	41
Fig. 4.43 Script para activar nodos sensores.	41
Fig. 4.44 Script para activar postes como nodos sensores.	42
Fig. 4.45 Función para aumentar un disco.	42
Fig. 4.46 Función para disminuir un disco.	42
Fig. 4.47 Función para reiniciar el juego.	43
Fig. 4.48 Función comenzar-jugar.	43
Fig. 4.49 Función para el poste 1.	44
Fig. 4.50 Sensores para configurar el número de discos.	45
Fig. 4.51 Sensor para comenzar a jugar.	45
Fig. 4.52 Exhibición al momento de ser configurada.	45
Fig. 4.53 Exhibición lista para comenzar a jugar.	45
Fig. 4.54 Sensor Reiniciar el juego.	46
Fig. 4.55 Exhibición "Torre de Hanoi" integrada.	47

## Capítulo 5:

Fig. 5.1 Inicio del modelado de "Gira, gira".	48
Fig. 5.2 Base de la exhibición "Gira, gira"	49
Fig. 5.3 Base de la exhibición con piso.	49
Fig. 5.4 Escalón de la exhibición "Gira, gira"	49
Fig. 5.5 Colocando postes y cuerdas.	50
Fig. 5.6 Exhibición "Gira, gira" terminada.	50
Fig. 5.7 Icono para animar.	51



Fig. 5.8 Fase inicial de la animación.	51
Fig. 5.9 Rutas para animar la exhibición "Gira, gira".	51
Fig. 5.10 Exhibición antes de iniciar la animación.	52
Fig. 5.11 Diferentes puntos de la animación.	52
Fig. 5.12 Exhibición "Gira, gira" integrada.	53

## Capítulo 6:

Fig. 6.1 Inicio del modelado de "Meteoritos".	55
Fig. 6.2 Mueble de la exhibición "Meteoritos".	55
Fig. 6.3 Modelado exportado a VRML.	55
Fig. 6.4 Fotografía de la exhibición.	56
Fig. 6.5 Modelado de "Meteoritos" en Internet Space Builder.	56
Fig. 6.6 Geoesferas deformadas con textura.	57
Fig. 6.7 Exhibición "Meteoritos" terminada.	57
Fig. 6.8 Exhibición "Meteoritos" integrada.	58
Fig. 6.9 Estructura de la base de la exhibición "Caleidoscopio".	59
Fig. 6.10 Modelado terminado de la exhibición "Caleidoscopio".	59
Fig. 6.11 Esfera dentro de los 3 espejos del "Caleidoscopio".	60
Fig. 6.12 Imagen producida por los 3 espejos en 3D Studio Max.	60
Fig. 6.13 Diferentes ángulos de la exhibición "Caleidoscopio".	60
Fig. 6.14 Exhibición "Caleidoscopio" integrada.	61

## ÍNDICE DE TABLAS

	Página
<b>Capítulo 4:</b>	
Tabla 4.1 Tiempos de descarga para la Recta Numérica.	31
Tabla 4.2 Tiempos de descarga para la Exhibición De Plantas, Aire, Agua y el Sol.	38
Tabla 4.3 Tiempos de descarga para Torre de Hanoi.	46
<b>Capítulo 5:</b>	
Tabla 5.1 Tiempos de descarga para Gira, gira.	52
<b>Capítulo 6:</b>	
Tabla 6.1 Tiempos de descarga para Meteoritos.	57
Tabla 6.2 Tiempos de descarga para Caleidoscopio.	61

---

## **Prólogo.**

El proyecto denominado Museo Virtual 3D El Rehilete, fue realizado mediante un convenio firmado por la Universidad Autónoma del Estado de Hidalgo (UAEH), a través del Centro de Investigación en Tecnologías de Información y Sistemas (CITIS), el Museo "El Rehilete" y el Gobierno del Estado de Hidalgo.

En el proyecto de virtualización del Museo Virtual 3D El Rehilete, participaron pasantes de la Licenciatura en Sistemas Computacionales, el trabajo se distribuyó de acuerdo a las áreas del Museo el Rehilete: Arte, Ciencia, Más sobre Ciencia, Nuestro Mundo, Tecnología, Servicios y Museos Históricos (Ex Convento de San Francisco de Asís situado en Tepeapulco, Hgo.).

El Museo el Rehilete es una Institución educativa y cultural creada por el Gobierno del Estado de Hidalgo que otorga un servicio educativo no escolarizado, interactivo, innovador y además divertido con el fin de que el público que asista a él se lleve un conocimiento científico, cultural, tecnológico y artístico.

Se virtualizó el Museo El Rehilete ubicado en Pachuca, Hidalgo, mediante el uso de técnicas de modelado interactivo de Realidad Virtual para ser consultado mediante Internet e Internet2. El Proyecto fue desarrollado dentro de las instalaciones del CITIS.

---

## **Problemática.**

Cuando se entra a Internet y se busca museos virtuales, aparecerán un sin número de páginas ligadas. Pocas veces se encuentra lo que se espera, páginas Web con imágenes es lo más que se llega a encontrar. Se necesita ser un buscador experto, o estar demasiado tiempo en Internet, para poder encontrar algo diferente e interactivo.

Existen en su mayoría museos virtuales hechos con texto e imágenes, hay otros que tienen además un recorrido interactivo, igualmente con imágenes, lo que resulta más atractivo para el usuario. Pero algo que realmente impacta y que tal vez supera las expectativas del usuario, es cuando se entra a una página Web que contiene un museo virtual tridimensional.

Se propone el desarrollo de exhibiciones tridimensionales de carácter interactivo que forman parte del área denominada "Más Sobre Ciencia", del Museo "El Rehilete".

Estas exhibiciones deben contar con un alto grado de interacción, además de tener carácter lúdico por lo que además de la especificación de VRML se utiliza un lenguaje ajeno a este, como lo es JavaScript para la asignación de comportamientos complejos.

---

## **Objetivo General.**

Desarrollar Mundos virtuales 3D para el proyecto "Museo Virtual 3D El Rehilete", desarrollados en 3 modalidades (Visualizaciones, Animaciones, y Comportamientos complejos), mediante el Lenguaje de Modelado de Realidad Virtual (VRML), además de que cada una de las exhibiciones desarrolladas (Caleidoscopio, De Plantas, Aire, Agua y el Sol, Gira, Gira, Meteoritos, Recta Numérica, Torre de Hanoi), estarán integradas en páginas Web, con información asociada.

---

## Objetivos específicos.

- Virtualizar, mediante el Lenguaje de Modelado de Realidad Virtual, la exhibición llamada *Recta Numérica* que se encuentra en el área de Más sobre Ciencia. Asignar comportamiento complejo por medio de Java Script, para que el usuario pueda manipular los diferentes objetos que se encuentran en la Recta Numérica.
- Virtualizar la exhibición llamada *De Plantas, Aire, Agua y el Sol*. Asignar comportamiento complejo, siendo este comportamiento el que los usuarios puedan mover los discos que se encuentran en la exhibición, para que cuando todos los discos se encuentren en la posición correcta, aparezca una imagen de plantas, el cual será el indicativo de que el juego ha terminado satisfactoriamente.
- Virtualizar la exhibición llamada *Gira, Gira*. Implementar animación de giro que existe en un impulso angular.
- Virtualizar la exhibición llamada *Torre de Hanoi*. Asignar comportamiento complejo para que los usuarios puedan mover los discos de un poste a otro, para que pueda realizarse el juego de las Torres de Hanoi.
- Virtualizar la exhibición llamada *Meteoritos*. Texturizar los meteoritos por medio de fotografías de los meteoritos del museo.
- Virtualizar la exhibición llamada *Caleidoscopio*. Texturizar las tres partes que conforman el caleidoscopio, y colocar un objeto dentro del caleidoscopio, de modo que simule los efectos de un espejo.

---

## Justificación.

Existen en su mayoría museos virtuales hechos con texto e imágenes, hay otros que tienen además un recorrido interactivo, igualmente con imágenes, lo que resulta más atractivo para el usuario. Pero algo que realmente impacta y que tal vez supera las expectativas del usuario, es cuando se entra a una página Web que contiene un museo virtual tridimensional.

El desarrollo del "Museo Virtual 3D El Rehilete" representó la oportunidad de desarrollar un museo virtual interactivo de la ciencia, lo cual implica la integración de las tecnologías de información para un sitio Web, además el virtualizar un museo resulta en sí, un proyecto de suma importancia por el impacto que tendrá dentro de la sociedad en general, los cibernautas y gente del medio informático.

Cabe mencionar que no existe un museo virtual de la ciencia con las características de interacción y que haga uso de la realidad virtual, por lo que se proporciona una nueva tecnología de información, y con esto el "Museo Virtual 3D El Rehilete" pretende cumplir y superar las expectativas de los usuarios de Internet, al ofrecer un ambiente virtual tridimensional.

---

## Limitaciones.

En el proyecto "Museo Virtual 3D el Rehilete" se contemplaron 103 exhibiciones, de las cuales 22 son del área de Más Sobre Ciencia. El equipo de desarrollo asignado está compuesto de 4 integrantes. A cada uno de ellos se les asignó una serie de exhibiciones y, específicamente, el trabajo reportado en esta tesina contempla la virtualización de las siguientes exhibiciones:

- Caleidoscopio
- De Plantas, Aire, Agua y el Sol
- Gira, Gira
- Meteoritos
- Recta Numérica
- Torre de Hanoi

La interactividad se verá limitada en exhibiciones como *Meteoritos* y *Caleidoscopio*, ya que éstas sólo podrán ser observadas en diferentes vistas. En otras se hará uso de dispositivos de entrada (Mouse), para activar la animación, al igual que en las exhibiciones con comportamientos complejos.

En la exhibición llamada Caleidoscopio se usan 3 espejos para reflejar un objeto. El lenguaje VRML carece de esta posibilidad, por lo que se hará la simulación colocando un objeto y texturas.

El desarrollo de un museo virtual 3D representa mucho más que una página Web con gráficos tridimensionales interactivos; su creación representa una nueva visión de lo que es un museo virtual. Por sus características resulta el primer museo virtual desarrollado para la Web.

Sin duda este proyecto amplía el horizonte de los desarrolladores de software que intentan estar a la vanguardia en el desarrollo de páginas Web.



# Capítulo 1

## Introducción.

### Resumen

En este capítulo se tratarán temas acerca de la realidad virtual, los museos tradicionales, del museo El Rehilete, museos virtuales. También se hace referencia a los diferentes lenguajes y librerías para generar gráficos 3D. Se muestra un listado del *software* requerido para la virtualización de las exposiciones, así mismo se presentan los lenguajes de programación empleados.

### 1.1.- Realidad virtual.

La realidad virtual es una representación de las cosas a través de medios electrónicos, que nos da la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea.

La realidad virtual puede ser de dos tipos: inmersiva y no inmersiva:

Los métodos inmersivos de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano.

La realidad virtual no inmersiva utiliza medios como el que actualmente nos ofrece Internet en el cual podemos interactuar a tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora[9] [ 10] [3 0] [31 ].

### 1.2.- Lenguajes y librerías para generar gráficos 3D.

Para el desarrollo de gráficos 3D existen diversos lenguajes y librerías. Algunos de estos lenguajes son aplicaciones orientados a la Web, y otros por sus características para funcionar en forma local.

A continuación se hace mención sobre algunos lenguajes y librerías para el desarrollo de gráficos 3D.

### 1.2.1.- Java 3D.

El API Java 3D es un interfaz para escribir programas que muestran e interactúan con gráficos tridimensionales. Java 3D es una extensión estándar del JDK 2 de Java. El API Java 3D proporciona una colección de constructores de alto-nivel para crear y manipular geometrías 3D y estructuras para dibujar estas geometrías. Java 3D proporciona las funciones para creación de imágenes, visualizaciones, animaciones y programas de aplicaciones gráficas 3D interactivas.

El API de Java proporciona un sistema de los interfaces orientados a objetos que apoyan un modelo de programación simple, de alto nivel. Esto permite a los desarrolladores construir, renderizar, y controlar el comportamiento de los objetos 3D y de los ambientes visuales [21] [22] [23].

Existen mundos virtuales creadas en Java 3D en donde se puede navegar de tres diversas maneras:

- Ya sea mediante el arrastre por ratón a la dirección deseada dentro del applet.
- Utilizando los botones de la navegación colocados en la página.
- El mismo efecto se puede alcanzar con el teclado al presionar las teclas de cursor después de hacer clic en la imagen.

### 1.2.2.- OpenGL.

OpenGL es un conjunto de librerías, que consta instrucciones que son utilizadas por lenguajes de programación como VisualC++, c, ANSÍ, Fortran, entre otros.

OpenGL, significa Graphic Library, y no es por lo tanto ningún lenguaje de programación, sino tan sólo un conjunto de librerías que son utilizadas a través de lenguajes de programación para conseguir un interfaz software entre las aplicaciones y el hardware gráfico. La librería está formada por unas 150 instrucciones diferentes que se utilizan para especificar los objetos y las operaciones necesarias para desarrollar aplicaciones interactivas tridimensionales. Los modelos se deben construir partiendo de un pequeño conjunto de "primitivas geométricas" como puntos, líneas y polígonos [26] [27].

Características.

- Es un estándar en el desarrollo de videojuegos y de gráficos en general.
- Es independiente de plataforma.
- Realiza efectos en tiempo real.
- Es el API de gráficos más rápido que existe.

### 1.2.3.-Cult3d.

Cult3d es un motor de render multiplataforma enfocado a escenas interactivas para páginas Web. Cult3d proporciona render<sup>1</sup> fotorealista sin necesidad de una tarjeta aceleradora de gráficos.

Características:

- Tiene soporte con ActiveX.
- Soporte para archivos PDF de Adobe Acrobat.
- Se pueden exportar objetos desde 3D Studio Max y Maya.
- La herramienta Cult3d designer permite crear la interactividad con los objetos.
- Cult3d viewer es un visualizador para Netscape, Explorer, PowerPoint, y para archivos PDF.

### 1.2.4.-X3D.

X3D es el futuro estándar para la especificación e intercambio de gráficos 3D y multimedia en aplicaciones Web.

Este nuevo lenguaje pretende ser usado en diferentes dispositivos de hardware y diferentes áreas de aplicación (ingeniería, presentación multimedia, entretenimiento, etc.) [37].

Características:

- Gráficos 2D y 3D.
- Animación.
- Audio y video.
- Interacción con el usuario mediante.
- Soporta y mejora VRML97.
- No hay especificación.

## 1.3.- Paseos virtuales.

Un recorrido virtual es una exploración interactiva de un espacio simulado, con el fin de dar conocimiento de un sitio real sin necesidad de estar presente en éste. Este producto es una excelente opción para las empresas que desean hacer guías turísticas o didácticas, o desean promover un sitio real.

---

<sup>1</sup> Render: En tratamiento de gráficos por computadoras, conversión del esquema de un dibujo en una imagen tridimensional totalmente formada, por medio de un modelo matemático.

El diseño de Recorridos Virtuales interactivos desde Internet nos permite tener una visión en 3 dimensiones de cualquier espacio (Calles, Edificios, Interiores, etc.). Se pueden visualizar espacios existentes o bien ser vistos antes de ser construidos, a partir de los planos.

El usuario puede explorar libremente el espacio virtual con la utilización del ratón. Además, se pueden incorporar herramientas que nos realice recorridos automáticos de la misma manera que lo haría un guía[28][29].

## **1 A-Museos tradicionales.**

Los museos que comúnmente se conocen contienen exposiciones con el fin de mostrar lo que ha acontecido en la historia del hombre, así como mostrar pinturas y esculturas.

Los museos tradicionales son instalaciones que albergan cantidades enormes de creaciones artísticas, en donde se pueden observar y admirar cada una de ellas.

### **1.5.-Museos virtuales.**

Cuando se entra a Internet y se busca museos virtuales, aparecerán un sin número de páginas ligadas. Pocas veces se encuentra lo que se espera, páginas Web con imágenes es lo más que se llega a encontrar. Se necesita ser un buscador experto, o estar demasiado tiempo en Internet, para poder encontrar algo diferente e interactivo.

Existen en su mayoría museos virtuales hechos con texto e imágenes, hay otros que tienen además un recorrido interactivo, igualmente con imágenes, lo que resulta más atractivo para el usuario. Pero algo que realmente impacta y que tal vez supera las expectativas del usuario, es cuando se entra a una página Web que contiene un museo virtual 3D.

El "Museo Virtual 3D el Rehilete" pretende cumplir y superar las expectativas de los usuarios de Internet, al ofrecer un ambiente virtual tridimensional, que tiene además un valor agregado, el cual es la interactividad que tendrá el usuario con algunas exhibiciones y la información asociada.

Este museo virtual 3D contará con reproducciones exactas de sus exhibiciones, así como de su inmueble que los alberga, podrá navegar dentro de las 6 áreas que se encuentran en el museo (Arte, Tecnología, Ciencia, Más Sobre Ciencia, Nuestro Mundo y Servicios).

## 1.6.- Museo "El Rehilete".

En la década de los 90's en México se abrieron nuevos museos, pero esta vez su enfoque es diferente a los museos normales. Estos museos fueron especialmente diseñados para los niños, en donde su función está orientada a fomentar el aprendizaje, la comunicación y la convivencia, a través de actividades interactivas.

En Pachuca, Hidalgo existe un museo con este tipo de características. El Museo el Rehilete es considerado una institución educativa y cultural, la cual abrió sus puertas en el año de 1997. Su nombre es en homenaje al lugar en que se encuentra, y a una característica de su clima, Pachuca "la Bella airosa", ya que este es sitio con mucho viento, elemento necesario para que el rehilete gire.

Uno de los objetivos de esta institución es el de brindar a la niñez y juventud hidalguenses la oportunidad de adquirir una visión global de la capacidad creativa del humano, así como el acceso equitativo a otras manifestaciones culturales, estimulando la sensibilidad y percepción mediante diversas actividades[25].

En la presente tesina se muestra el desarrollo de virtualización de algunas de las exhibiciones que se encuentran en el Museo el Rehilete (Caleidoscopio, De Plantas, Aire, Agua y el Sol, Gira, Gira, Meteoritos, Recta Numérica, Torre de Hanoi).

Para visualizar el sistema de forma eficiente es necesario que el usuario tenga en su computadora algunos requerimientos específicos, tanto en el *hardware* como en el *software*, lo cual se menciona enseguida.

## 1.7.- Requerimientos de *hardware* para visualizar la página.

- VRML es una tecnología que utiliza gráficos 3D, por lo tanto será necesario una tarjeta de video, o aceleradora de gráficos, para que la computadora tenga un desempeño óptimo.
- Así mismo al ser una tecnología con gráficos 3D, la computadora realizará un sin número de cálculos para una nueva posición de los objetos en el espacio virtual 3D, por lo que se recomienda, un procesador a 900 MHz, o mayor.
- Se recomienda una memoria RAM de 128 MB o mayor.

## 1.8.- Requerimientos de software para visualizar la página.

- El proyecto será publicado en Internet, por lo cual, principalmente se necesita e 1 acceso a Internet.
- Un navegador VRML, permitirá visualizar o interpretar los objetos en 3 dimensiones, este puede ser el de cosmo player, cortona, u otros, pero se recomienda el de cosmo player por ser uno de los más fáciles de usar.
- Un navegador para poder acceder a Internet (Internet Explorer o Netscape Navigator).
- Se desarrollarán aplicaciones desarrolladas en *Macromedia Flash*, por lo que será necesario el player de Flash.

Para el desarrollo de este trabajo fue necesario el uso diferentes tipos de lenguajes y herramientas, los cuales se mencionan a continuación:

## 1.9.- Software requerido para el desarrollo.

Para el desarrollo del sistema se hizo uso de herramientas para generar animaciones y botones, para crear los objetos 3D, para asignarles texturas, para generar imágenes, ya para la programación de eventos en VRML, las herramientas que se utilizaron son las siguientes:

- **Macromedia Flash:** Para la integración de la página se creó un menú, que contendrá cada una de las exhibiciones que conforman el área de Más Sobre Ciencia, éste esta a la izquierda de la pantalla; para Internet I se utiliza un menú que será programado con Java Script, y para Internet II se hará el diseño y programación con Macromedia Flash.
- **Internet Space Builder:** Es una herramienta desarrollada por Parallel Graphics, la cual ayuda al programador que desarrolla mundos 3D con VRML, a texturizar con mucha mayor facilidad los objetos que anteriormente han modelado.
- **Photoshop:** En el Museo El Rehilete cada exhibición tiene una cédula en la cual contiene información acerca de que hacer, sucesos relevantes e información acerca del fenómeno. Y para que esta información sea leída por el usuario fue necesario digitalizar las cédulas en forma de imagen en Photoshop.
- **3D Studio Max:** En VRML, el modelado de las exhibiciones requiere que el programador tenga un amplio sentido de orientación en espacios tridimensionales;

el diseñar en modo gráfico resulta mucho más fácil que estar en un editor de textos, y permite al diseñador ocuparse por otras cosas más importantes, y esto es posible con 3D Studio Max

- **VrmlPad:** Esta herramienta también fue desarrollada por Parallel Graphics, y su finalidad es el facilitar la programación en VRML. Muestra un árbol de nodos que se crean al programar los objetos, los llamados que se hacen de los recursos (las rutas de las texturas que se asignan, así como llamados a otros ficheros ".wrl"), y principalmente muestra rutas que existen entre los diferentes nodos, lo cual facilita enormemente la programación.

## 1.10.- Lenguajes requeridos para el desarrollo.

Para el desarrollo del sistema se hizo uso de lenguajes para generar las páginas Web, para representar los objetos 3D, y un lenguaje para generar comportamiento complejo a algunos objetos, los lenguajes que se utilizaron son las siguientes:

- **HTML:** Este es el lenguaje estándar para la -creación de páginas Web. Se programó en este lenguaje por la sencillez de su código, además de que se tomó en cuenta de que con este lenguaje las páginas son mucho más pequeñas en tamaño y no requieren de esto
- **VRML:** En la década de los 90's se liberó una de las tecnología más innovadoras; la idea de crear un ambiente con gráficos tridimensionales, fue por fin concretada al salir el lenguaje VRML [32]. Con este tipo de tecnologías se pueden representar objetos extraídos de la realidad, o bien ser creados por imaginación del propio desarrollados Cult3d y VRML son las opciones más viables al diseñar objetos tridimensionales, pero por razones de desempeño VRML es mucho mejor; Cult3d es un poco más lento en el render de los objetos, y su programación mucho más compleja, aunque hay que resaltar que tiene mucho mejor renderización el estándar Cult3d que VRML.
- **JavaScript:** Este lenguaje puede adherirse a un programa de un lenguaje que no es fundamentalmente para hacer cálculos, como por ejemplo: HTML o VRML.

## Capítulo 2

# Lenguajes y Herramientas de Desarrollo.

### Resumen

En este capítulo se explicará más a detalle sobre el software (Macromedia Flash, Internet Space Builder, Photoshop, 3D Studio Max.), requerido para el desarrollo del "Museo Virtual 3D El Rehilete". Así mismo se describirán los lenguajes HTML, VRML y Java Script, utilizados en la creación de los mundos virtuales 3D.

### 2.1.-VRML.

En 1989 se inició lo que fue el proceso de diseño de un lenguaje capaz de permitir la interacción con gráficos tridimensionales. Los encargados de ese proyecto fueron Rikk Carey y Paúl Strauss, de la empresa Silicon Graphics.

La primera fase del proyecto se concentraba en diseñar y construir la semántica y los mecanismos para la plataforma de trabajo. En 1992 se presentó el *Iris Inventor 3D Toolkit*, que fue el primer producto de dichos esfuerzos.

En 1994, Mark Pesce y Brian Dehlendorf crearon el VRML *mailing list* o lista de discusión "WWW-VRML" (<http://vag.vrml.org/www-vrml>) donde se hizo un llamado abierto al todo el público para dar propuestas para una especificación formal de 3D orientado a la Web. Dada la magnitud del trabajo se decidió avanzar por etapas y adoptar estándares existentes. En este mismo año Mark Pesce y Tony Parisi crearon un prototipo de visor de 3D para el WWW.

Después de varias propuestas se escogió la sintaxis de *OpenInventor* de Silicon Graphics como base de un formato de descripción de objetos geométricos texturizados, agregando la posibilidad de combinar objetos guardados remotamente en la red (mediante hipérligas como en HTML). De esta manera nació VRML 1.0 que aunque sólo era una solución parcial, era una muestra de lo que VRML podría llegar a ser.

Durante 1995 la especificación de VRML 1.0 sufrió un gran número de clarificaciones y reparaciones, pero funcionalmente quedó igual. En Agosto de 1995 hubo mucha controversia dentro del grupo de discusión WWW-VRML en cuanto a la creación de VRML 1.1 o de VRML 2.0. Algunos pensaban que VRML necesitaba sólo de unas cuantas adiciones de contenido, mientras que otros sentían la necesidad de una completa revisión del estándar. El segundo paso comenzó en Siggraph 95 y culminó en Siggraph 96. El nuevo estándar permitió el movimiento de la geometría estática definida en VRML 1.0. Se hizo un



llamado a presentar propuestas públicamente y se estableció una página de Web para votar. Hubo propuestas de más de 50 compañías como Silicon Graphics, Sony, Netscape, Apple, IBM, Microsoft, entre otras. Ganó la propuesta Moving Worlds de Silicon Graphics [12][14][15][32].

Mediante el lenguaje VRML se puede diseñar los mundos virtuales extraídos de la realidad. Este lenguaje utiliza una sintaxis muy fácil de aprender. Tan fácil como que con primitivas como *Box* se puede crear una caja con diferentes dimensiones, o *Sphere* para crear esfera.

Su sintaxis sencilla hace que el desarrollador se le facilite las cosas en cuanto programación, y deja que su mente se concentre más en las coordenadas precisas para que el mundo resulte lo más perfecto posible.

Crear animaciones con los objetos es otra de las propiedades que tiene el lenguaje VRML.

En VRML

- Se puede
  - Crear comportamiento para hacer que las formas se muevan, roten, cambien de tamaño, parpadeen, etc.
  - Proveer interacción usuario/mundo
- Se necesita
  - disparar y responder eventos
  - controlar el tiempo [34]

Dentro del lenguaje también existen los Nodos Censores los cuales permiten una interacción en entre el usuario y el mundo 3D, dado que estos sensores capturan los eventos del ambiente [33]:

- ProximitySensor, TimeSensor, VisibilitySensor
- Colusión es un nodo de agrupamiento y sensor de colisiones del usuario con las formas pertenecientes al grupo.
- LOD y Billboard son nodos de agrupamiento y sensores de visión con comportamiento predefinido.

Otros sensores que forman parte del lenguaje son del pointing device:

- TouchSensor.

- Anchor es un nodo de agrupamiento y sensor de acción, sensor con comportamiento predefinido.
- En particular, del arrastre del pointing device SphereSensor, CylinderSensor, PlaneSensor

Los nodos sensores son nodos hijos en la jerarquía y por lo tanto pueden estar en el campo children de un nodo de agrupamiento.

Los sensores que tienen pointing device influyen sobre toda la geometría que descende de su nodo padre [13].

También dentro del proyecto se hará un uso frecuente del nodo TouchSensor el cual genera eventos cuando el usuario toca con el Mouse un nodo del mundo 3D.

Los prototipos son nodos de alto nivel los cuales al declararlos encapsulan la información del nodo para que después, pueda ser vuelto a usar como un nuevo nodo.

La sentencia para declarar a este tipo de nodo es *FPROTO* en la que además se deben especificar los campos y eventos como argumentos.

En la siguiente figura se muestra un programa que nos muestra el manejo de prototipos.

```
PROTO BouncingBall [  
field SFFloat bounceHeight L 0  
field SFTIME bounceTime 1.0  
/{  
  DEFBall Transform {  
    children [  
      Shape { . . }  
    ]  
  }  
  1  
  |  
  DEF Clock TimeSensor {  
    cycleInterval IS bounceTime  
    startTime LO  
    stopTime 0.0  
    loop TRUE  
  }  
  DEF Bouncer Script f  
    field SFFloat bounceHeight IS bounceHeight  
  }  
  }  
ROUTE Clock.fr action_changed TO Bouncer. Set_fraction  
ROUTE Bouncer. value_jchanged TO Ball set_translation  
}
```

Fig. 2.1 programa que utiliza prototipos.

## 2.2.- 3D Studio Max.

3D Studio Max es una aplicación basada en el entorno Windows (9x/NT) que permite crear tanto modelados como animaciones en tres dimensiones (3D) a partir de una serie de vistas o visores (planta y alzados). La utilización de 3D Studio Max permite al usuario la visualización y representación de los modelos, así como su exportación y almacenamiento en otros formatos distintos del que utiliza el propio programa [1] [2][3][13].

3D Studio Max tiene un entorno gráfico que cuenta con 4 tipos de vistas, las cuales pueden ser modificadas por el usuario. Así mismo en esta aplicación su entorno se muestra fácil y claro, esto se debe a que con la ayuda de la iconografía el crear los modelados es relativamente fácil.

3D Studio Max fue una de las herramientas de mayor importancia para el desarrollo del proyecto de virtualización, debido a que en ella, se modelan las exhibiciones que se concentran en el museo, para posteriormente exportar el archivo al formato del Lenguaje de Modelado de Realidad Virtual (VRML).

## 2.3.- Internet Space Builder.

Internet Space Builder (ISB) es una herramienta de autoría de fácil utilización conveniente para los diseñadores de todos los niveles. Con ISB, crear los modelos 3D es más fácil [17].

Este es un programa especializado en representaciones 3D para Web, utilizando como formato de salida VRML, o Lenguaje de Modelado de Realidad Virtual, generando archivos para Web compactos, que permiten transmitirlos rápidamente. La extensión para estos archivos es WRL, debido a que se denomina mundo o "*World*" a los escenarios generados [18].

Con esta herramienta los modelados exportados de 3D Studio Max, se les asignan texturas cuando se es requerido. Ya que los colores que asigna el 3D Studio Max pueden ser suficientes para que el objeto sea idéntico al de la realidad.

Este programa es un editor de modelos 3D para su inclusión en páginas Web, de muy fácil manejo. Incorpora más de 1000 ejemplos de texturas, objetos tridimensionales, escenas, paisajes e imágenes. Importa objetos de 3DStudio y Autocad, y exporta en los formatos VRML 2.0, BMP, GIF, JPEG y PNG [19].

## 2.4.- Macromedia Flash.

Macromedia Flash es la herramienta número uno a nivel mundial para el desarrollo de contenidos multimedia interactivos. Lo que comenzó como un simple programa de animación vectorial orientado a la Web, a través de la evolución constante de sus nuevas versiones, se ha transformado en la herramienta más utilizada para generar interactividad tanto en soportes *on-line como off-line*. Hoy, Flash es la herramienta "de facto" de todo artista o desarrollador Web [24].

## 2.5.- JavaScript.

El lenguaje de Java Script no se parece en mucho con el Java, ya que Java es un lenguaje que soporta programación orientada a Objetos, y Java Script es un lenguaje sólo basado en objetos [20]. Este es un lenguaje el cual fue utilizado para poder hacer a las representaciones 3D cierto comportamiento complejo.

Java Script es un lenguaje totalmente diferente a VRML, pero estos pueden interactuar entre sí, con el fin de que los elementos 3D se puedan manipular y tengan un efecto de realismo mucho mayor [8] [11].

## 2.6.- HTML.

HTML es un lenguaje estándar para el diseño y creación de páginas Web, el cual es compatible con Windows y Linux.

"HTML (*HyperText Markup Languagé*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* (gráficos, sonido.) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape)" [7] [16].

## 2.7 VrmIPad.

VrmIPad es un completo, editor del lenguaje de programación de mundos virtuales VRML, especialmente indicado para profesionales de la programación en Internet [35].

Este programa tiene como principal ventaja que genera un árbol compuesto por cada uno de los nodos que insertamos en el mundo que estamos generando, lo que nos facilita la navegación por el código y su mantenimiento. Además tiene una herramienta para la localización de errores en el código y para autocompletar el código que estamos escribiendo [36].

## Capítulo 3

# Descripción del Método Empleado.

### Resumen

En este capítulo se mencionan los diferentes tipos de metodologías, además se describe el método empleado para el desarrollo de los mundos virtuales, y los diferentes tipos de visualización.

### 3.1.- Tipologías de métodos.

Los métodos para el desarrollo de sistemas indican como construir técnicamente un sistema. Los métodos abarcan una amplia gama de fases en las que se incluyen el análisis de los requisitos, diseño, construcción de programas, pruebas y mantenimiento.

Existen muchos métodos para el desarrollo de sistemas dentro de los cuales podemos mencionar [4] [5] [6]:

- El modelo lineal secuencial.
- El modelo de construcción de prototipos.
- El modelo DRA.
- El modelo incremental.
- El modelo de ensamblaje de componentes.
- IDEF.

No existe un método aceptado ni planteado internacionalmente para el desarrollo de sitios Web que contengan mundos virtuales tridimensionales.

El método utilizado para el desarrollo del sistema cumple con todas las etapas del método de desarrollo de software clásico; el método empleado comprende las siguientes etapas:

1. Recopilación de información.
2. Análisis
3. Diseño.
4. Desarrollo de exhibiciones.
5. Integración
6. Pruebas de funcionamiento y tiempos de descarga.
7. Validación.

Dentro de la etapa de "Desarrollo de exhibiciones" se encuentran las siguientes fases:

- Modelado 3D.
- Adición de animaciones.
- Asignación de comportamiento complejo
- Asignación de texturas.
- Optimización de código.

## **3.2.- Etapas del método.**

A continuación se describen las etapas del método usado en el desarrollo del sistema.

### **3.2.1- Recopilación de la información.**

La información de las exhibiciones fue proporcionada por el Museo el Rehilete, de forma impresa. Dicha información contiene: nombre de la exhibición, objetivo, descripción, principio y/o leyes, aplicaciones, propuesta interactiva, contenidos temáticos, bibliografía, y la cédula de la exhibición. Con esta información se creó el desarrollo de la documentación en línea.

Además de la información impresa, para modelar las exhibiciones, fue necesario también acudir al Museo el Rehilete a tomar medidas exactas de cada una de las exhibiciones, sus colores y texturas correspondientes.

### **3.2.2.-Análisis.**

En esta etapa se analiza la información obtenida previamente de cada una de las exhibiciones, y se identifica que nivel de representación se utilizará para cada exhibición.

Para el desarrollo de las exhibiciones se utilizarán los siguientes niveles de representación:

#### **Visualizaciones:**

Mundos tridimensionales con objetos con textura, sin interacción con el usuario.

#### **Animaciones:**

Mundos tridimensionales con objetos donde la interacción se limita a activar y observar comportamientos repetitivos determinados, para ilustrar visualmente un fenómeno, actividad o experimento.

#### **Comportamientos Complejos:**

Mundos tridimensionales con objetos que tienen movimientos complejos, que requieren simulación de fenómenos o experimentos.

### **3.2.3.- Diseño.**

En este proyecto se respetan los lineamientos de diseño establecidos en el Manual de Imagen Gráfica del Museo el Rehilete, a fin de utilizar correctamente aquellos que son de aplicación obligatoria en la reproducción y divulgación de la imagen.

En el desarrollo, se toman en cuenta aspectos relacionados con el logotipo, tipografías, reproducción y usos de la imagen del museo.

El diseño de la página cuenta con un número de 4 marcos. Un marco superior el cual contiene un menú para poder acceder a las diferentes áreas, dos marcos laterales, uno de ellos que contiene un menú de las exhibiciones presentadas en el área seleccionada, el otro marco lateral contiene la información y ayuda, y por último el marco principal en donde esta el área de interacción 3D.

Cada uno de los archivos de las exhibiciones con la extensión ".wrl", son embebidos en archivos ".htm".

Para el diseño de los mundos virtuales se hizo uso de movimientos simples propios de VRML como son el *TouchSensor*, *PlaneSensor*, *CylinderSensor* y *SphereSensor*, animaciones y uso de programas JavaScript

Por ejemplo el *PlaneSensor* es utilizado para la exhibición De Plantas, Aire, Agua y el Sol, para arrastrar las fichas en el tablero.

### **3.2.4.- Desarrollo de las exhibiciones.**

#### **Modelado 3D.**

En esta subetapa, se toma la información recabada sobre las medidas de las exhibiciones y se crean los objetos en 3D Studio Max.

#### **Adición de animaciones.**

La creación de las animaciones se hace también desde 3D Studio Max.

#### **Asignación de comportamiento complejo.**

Para la asignación de comportamiento complejo se utiliza el lenguaje JavaScript, éste a su vez se encuentra embebido dentro del lenguaje VRML.



### **Asignación de texturas.**

La asignación de color se realiza desde 3D Studio Max, y para la asignación de texturas se utiliza Internet Space Builder.

### **Optimización de código.**

En esta etapa las exhibiciones están en un proceso de reducción de código. Para las aplicaciones 3D por Internet la limitación más grande es la velocidad de la conexión a Internet, por tanto uno de los objetivos que se persiguen al desarrollar un mundo virtual es mantener bajo los tiempos de descarga por parte de los usuarios. Las herramientas que se utilizaron para la reducción de código fueron Internet Space Builder y VrmlPad. Internet Space Builder hace la reducción de código de forma automática, y para VrmlPad es necesario salvar el archivo en modo comprimido.

### **3.2.5.- Integración.**

Las exhibiciones pueden estar separadas en varios archivos ".wrl", para unir todos los componentes, así como luces y cámaras, se crea un archivo más, creando ligas a cada uno de los componentes.

Para integrar las exhibiciones se usa un archivo integrador en donde se ligan los archivos de las exhibiciones. Con este archivo se crea el área de Más Sobre Ciencia con todas sus exhibiciones, además de estar colocados con el inmueble.

### **3.2.6." Pruebas de funcionamiento y tiempos de descarga.**

En esta etapa se hacen pruebas de funcionamiento por exhibición. Con el sistema completo en el servidor, las pruebas también consisten en ingresar a la página Web desde las estaciones de trabajo, y verificar que el mundo 3D funcione correctamente y que la información asociada sea la correcta.

Los tiempos de descarga se toman con la herramienta VrmlPad, la cual da los tiempos con módems de 14.4, 28.8, y 56.6 kbps

### **3.2.7.- Validación.**

Para validar que las exhibiciones funcionen correctamente, personal del museo el Rehilete revisa las páginas desde las instalaciones del museo, en donde posteriormente se notifica

que la exhibición está completa y sin errores, o en su caso se comentan los errores para después ser depurados.

## Capítulo 4

# Desarrollo e Integración de los Mundos Virtuales con Comportamiento Complejo.

### Resumen

En este capítulo se aborda la creación e integración de las exhibiciones con comportamiento complejo, su desarrollo en 3D Studio Max, tomando medidas reales, la asignación de color, y también la forma en la que se les asigna texturas, y el desarrollo de los programas JavaScript para generar la interactividad con comportamiento complejo.

## 4.1-Recta Numérica.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "Recta Numérica". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

### 4.1.1.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se le dio comportamiento complejo.

#### 4.1.1.1- Modelado 3D.

En esta etapa del desarrollo de exhibiciones consiste en modelar los objetos reales en un ambiente virtual 3D. Para ello se procede a determinar las dimensiones exactas del objeto real. Una vez hecho esto, se utilizó la herramienta 3D Studio Max en donde se modelaron todos los objetos tridimensionales.

La complejidad de los objetos determina si se usa 3D Studio Max o se programa directamente en VRML. Por ejemplo, si se crea un objeto tridimensional que sea una caja o un cubo, en VRML su código sería tal como se muestra en la figura (4.1).

```

.#VRML V2.0 utf8

Transform
{
  translation 0 0 0
  children
  [
    Shape
    {
      appearance Appearance
      {
        material Material {diffuseColor 0.3373 0.3373 1 }
      }
      geometry Box {size 1 1 1 }
    }
  ]
}

```

Fig. 4.1 Código VRML para crear una caja.

A esto se le llama primitiva estándar, las cuales pueden ser una caja (Box), esfera (Sphere), cilindro (Cylinder), cono (Cone), y texto (Text).

Por otro lado en 3D Studio Max hay un menú de primitivas estándar (caja, cono, esfera, geoesfera, cilindro, tubo, toroide, pirámide, plano y tetera), primitivas extendidas (poliedro, nudo de toroide, chaflán caja, chafláncilindro, bidón, cápsula, etc.) y figuras (líneas, textos, círculos, etc.).

Con esta herramienta la creación de mundos virtuales se lleva a cabo con rapidez. Un ejemplo se muestra en la figura (4.2) en donde se selecciona la creación de una caja



Fig. 4.2 Icono de 3D Studio Max para crear una caja.

El objeto modelado de la caja aparecerá en 4 perspectivas como se muestra en la figura 4.3.

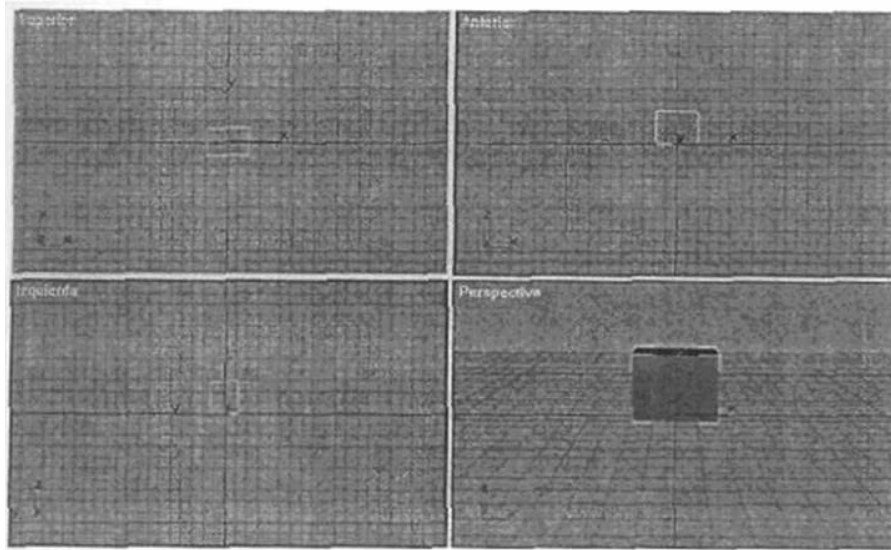


Fig. 4.3 Visualización de una caja en 3D Studio Max.

El uso de 3D Studio Max facilita la creación de objetos, y mucho más aún cuando se trata de objetos mucho más complejos, en los que se tengan que hacer operaciones booleanas, o que se tenga que modificar su figura.

La Recta Numérica se modela con 3D Studio Max, creando por principio la estructura, con cajas de diferentes dimensiones.

La primera caja tiene dimensiones de 300 cm. de largo 98 cm. de altura y 5 cm. de ancho (ver figura 4.4)

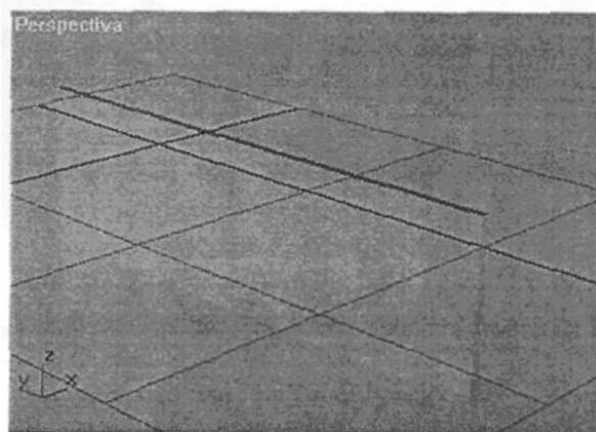


Fig. 4.4 Vista de la caja para formar la estructura de la Recta Numérica.

Para asignar el color se da un clic al cuadro de color que se encuentra en la barra de herramientas que aparece en el lado derecho de la pantalla.



Fig. 4.5 Icono para asignar color (Barra de herramientas derecha).

Aparece esta paleta de colores, en donde se asigna el color (antes se selecciona el objeto al que se le asignará el color deseado).

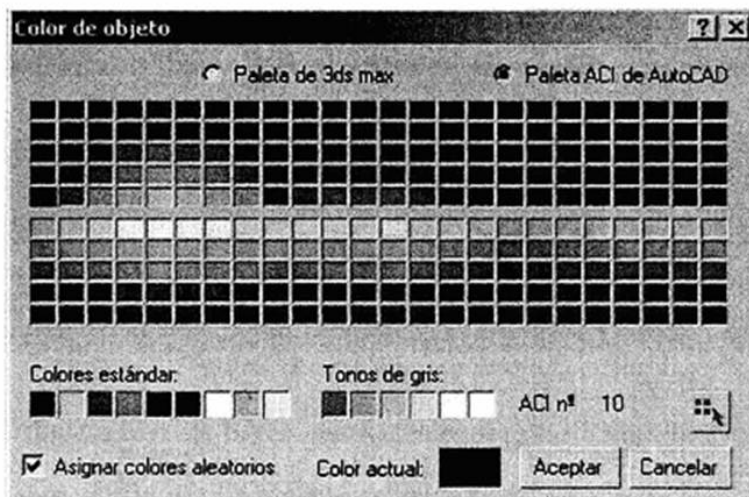


Fig. 4.6 Paleta para asignar color al objeto.

Se crea otra caja con dimensiones de 300 cm. de largo 25 cm. de altura y 5 cm. de ancho. Y se colocan las bases de la Recta Numérica.

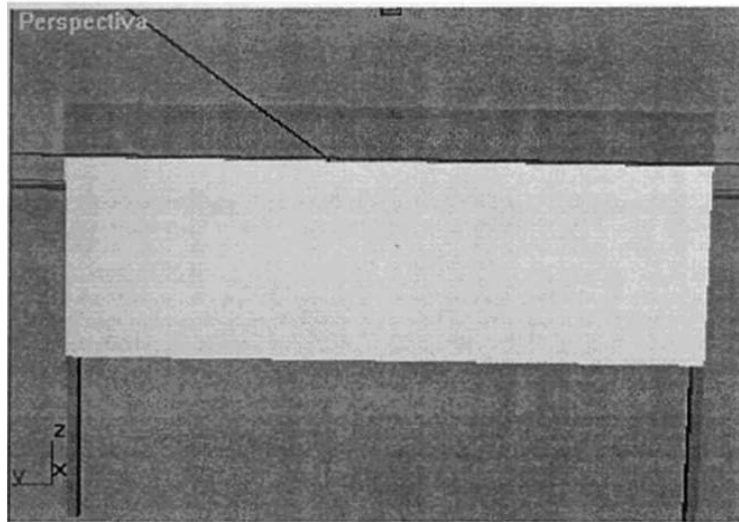


Fig. 4.7 Modelado de la estructura de la Recta Numérica.

En la caja de color azul contiene el letrero "Recta Numérica", creándolo desde el menú "Formas" en la opción de texto



Figura 4.8 menú para crear un texto.

El texto en 3D Studio Max se puede apreciar, pero no en VRML. Para que el texto sea visualizado también en VRML se tiene que extruir. Para hacer el proceso de extrusión, se selecciona el texto, y la pestaña en donde se encuentran las listas de modificadores en el menú de la derecha, y con el comando extruir, el texto tendrá volumen.

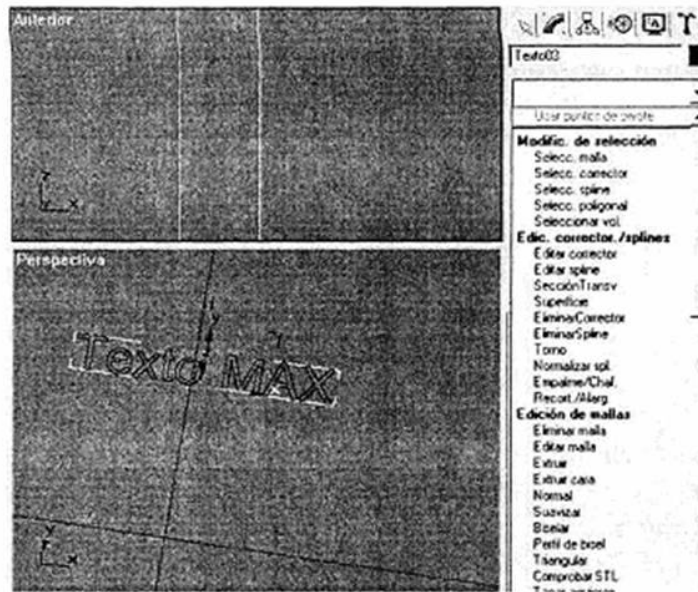


Figura 4.9 Menú para localizar el menú para extruir.

Se colocan de la misma manera los números que tiene en las dos caras la Recta Numérica. El modelado completo se muestra en la figura 4.10.

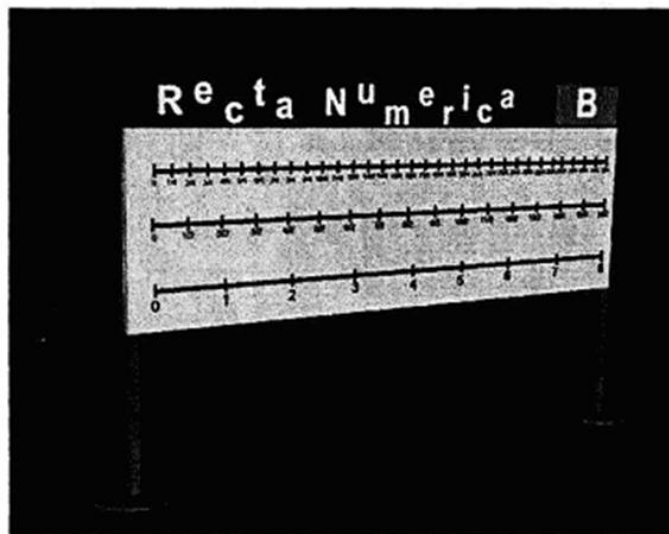


Figura 4.10 Modelo 3D de la Recta Numérica.

El modelado terminado se exporta de 3D Studio Max al lenguaje VRML, en donde se crea el comportamiento complejo que tendrá la exhibición.

En los manuales que se ofrecen en Internet para la creación de mundos virtuales con el lenguaje VRML, hay ejemplos prácticos para la creación de las primitivas estándar y también para la interacción de los nodos, ya sea para activar una cámara a través de un nodo sensor dentro del mundo tridimensional, animar los objetos con posiciones



predeterminadas, cambios de color de los objetos, activar sonidos, activar videos, etc.

#### 4.1.1.2- Comportamiento complejo

En el programa se declara una variable del tipo vectorial, de 3 dimensiones. Los valores que se obtendrán en "X" serán los que resulten de la activación del censor, y los de "Z", y "Y" serán declarados al inicio del programa y quedaran constantes.

Las variables de campo que se utilizarán en el programa son 2: una de tipo entero la cual será para que el contador aumente o disminuya según el censor activado, y la segunda, la cual tiene la función de determinar el valor de "X" del objeto dentro de la Recta Numérica. Ésta debe de comenzar en un punto determinado y disminuir o aumentar la distancia que sea necesaria y exacta para que el objeto represente gráficamente la operación aritmética. Por ultimo se declaran las variables de salida, las cuales son indispensables para que el objeto tenga los movimientos que deseamos.

```
eventIn SFTIME touchTime eventIn
SFTIME touchTimeO field SFInt32
touchCount 0 field SFInt32 aux 0
field SFInt32 eventCount 0 field
SFFloat contador 2.405 field
MFString tempString"" eventOut
MFString outputString eventOut
SFVec3f value changed
```

Fig. 4.11 Variables.

Existe una función por cada censor, en donde se aumenta y se disminuye, tanto el contador como la variable que indicará la posición en "X" del objeto.

En la figura 4.12 se muestra la parte del programa en donde se hacen las operaciones aritméticas de suma.

```
1 function touchTime 1 (valué, time) {
2   print ('touchTime eventIn received');
3   if(touchCount < 32 )
4     {
5     touchCount++;
6     contador = contador -0.8484;
7     value_changedc[0] = 4;
8     value_changedc [ 1 ] = -1.5 ;
9     value_changedc[2] = contador;
10    aux = touchCount - 1;
11    tempString [0] = aux.toString () + 74' + '+' + '1/4' + '=' + touchCount. toString ()
12    +74';}}
```

Fig. 4.12 Segmento del programa que realiza los cálculos de suma.

En esta función la variable "touchCount" incrementa su valor en uno; la variable "contador" decreuenta su valor 0.8484 unidades; a la variable de salida de 3 vectores se les asigna el mismo valor en "X", y "Y", y en "Z" se le asigna el valor de la variable "contador".

En la figura 4.13 se muestra la función del programa en donde se hacen las operaciones aritméticas de resta; en esta función la variable "touchCount" decrementará en uno, y "contador" incrementará su valor para dar la nueva posición al objeto dentro de la Recta Numérica.

```
function touchTime2 (value, time) {
  print ('touchTimeO eventIn received¹);
  if (touchCount > 0)
  {
    touchCount--;
    contador = contador + 0.848;
    value_changedc [0] = 4;
    value_changedc [1] = -1.5 ;
    value_changedc[2] = contador;
    aux = touchCount + 1;
    tempString [0] = aux.toString () + V4¹ + '-' + '1/4' + '-' + touchCount.toString ()
    + '1/4');
  }
}
```

Fig. 4.13 Segmento del programa que realiza los cálculos de resta.

Para el programa *Java Script* indique los nuevos valores, tanto de posición del objeto como del valor del String al VRML, se necesitan de rutas que indiquen de donde proviene el parámetro y hacia que nodo se enviará, en la figura 4.14 se muestran las rutas que se utilizaron para uno de los objetos en la Recta Numérica:

```
ROUTE SENSORO.touchTime TO SCRIPT.touchTime
#####
#####CODIGO PARA PONER EL TEXTO #####
#####
ROUTE SENSORI.touchTime TO SCRIPT.touchTimeO
ROUTE SCRIPT.outputString TO TEXT.string

#####
#####CODIGO PARA PONER MOVER EL OBJETO
#####
#####
ROUTE SCRIPT.value_changed TO conejo.setjranslation
```

Fig. 4.14 Rutas en VRML con Java Script

Estos segmentos de código son la parte esencial para que el programa funcione: la declaración de variables de entrada, de campo y de salida, las funciones que realizarán los cálculos a través de las variables declaradas, y las rutas que envían los valores de los sensores y las coordenadas de los objetos, así como el valor del String que estaremos utilizando.

Por cada objeto que se desea dar movimiento será necesario un programa *JavaScript*,

La Recta Numérica que se exhibe en el Museo el Rehilete, tiene 2 lados por los que se puede jugar con los objetos, el lado "A", y el lado "B". En el lado "A" se muestran solo números enteros, la sumas y la resta, en el lado "B" se desarrollan sumas y restas pero también se trabaja con quebrados, y se muestra la equivalencia.

Los controles que permiten al usuario interactuar con la exhibición deberán de ser 2 por cada objeto, uno para aumentar y otro para disminuir el contador del *JavaScript*.

Los controles se mantendrán siempre en pantalla, para lo cual se necesita un código, en el que se utiliza el *ProximitySensor*, el código se muestra en la figura 4.15.

```
DEF PANELPROX Proximity Sensor {
  size 100 100 100
}
DEF PANELTRANS Transform {
  children [Transform {children [
  Transform {
    translation-4-2.8 0.11 #rotation
    0-1 0 1.570796 children [Shape
    {appearance Appearance {
      material Material {
        i
        diffuseColor 0 0 0
      }}
    geometry DEF TEXT Text {
      fontStyle FontStyle {
        justify "MIDDLE" family
        "SANS"
        size 0.3}}
    ]}]
    translation00-10
  ]}]
  ROUTE PANELPROX.position_changed TO PANELTRANS. Set_translation
  ROUTE PANELPROX.orientation_changed TO PANELTRANS.set_rotation
}
```

Fig. 4.15 Código para mantener los objetos estáticos frente a la cámara.

La recta numérica compuesta por todos estos componentes (modelado 3D, sensores, objetos con movimiento, panel de control, el programa *JavaScript*), queda finalmente de esta manera, como se muestra en la figura 4.16.

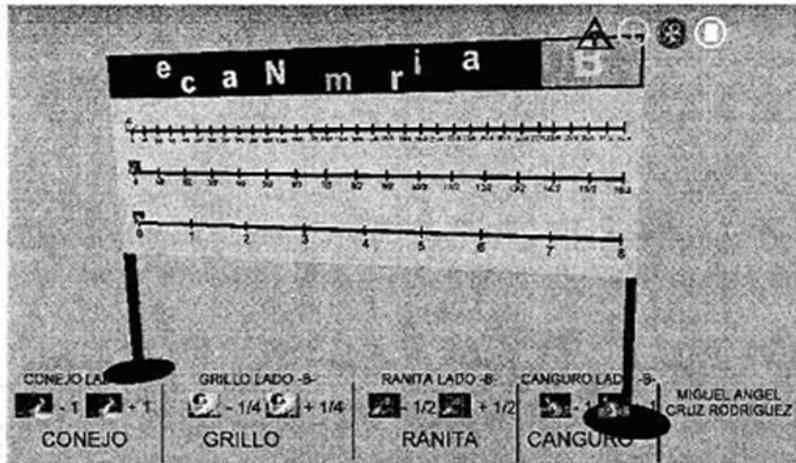


Fig. 4.16 Recta Numérica

El desarrollo de esta exhibición requirió del modelado tridimensional en 3D Studio Max, de la programación de un Script, y de la interactividad de ambos, lo cual permite que el usuario tenga una mayor y mejor idea de que es lo se pretende lograr con esta exhibición.

#### 4.1.2.- Pruebas.

En la figura 4.17 se muestra los controles para uno de los objetos, éste hace sumas y restas de enteros, y la figura que representa un conejo. En la parte superior indica la figura que se moverá y el lado al que pertenece, en la parte inferior el nombre del objeto, el cual cambiará al hacer clic sobre uno de los sensores, por los operandos y el operador que fue pulsado, ya sea "-1" o "+1".

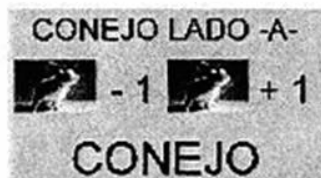


Fig. 4.17 Control de movimiento del objeto "CONEJO".

Los objetos se encuentran en la posición de inicio, en este caso en cero.

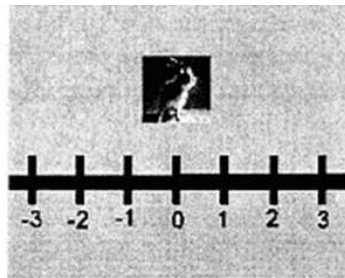


Fig. 4.18 Posición inicial del objeto "Conejo".

En la figura 4.19 se muestra el cambio de valor de la variable *String*. De inicio tiene el nombre del objeto que se representa (Fig. 4.17), y al dar clic en la imagen que representa la suma. El valor de *String* cambia y muestra el operador y los operandos.

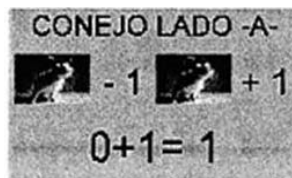


Fig. 4.19 Cambio de valores del String, que muestra la operación que realizó el usuario.

Al mismo tiempo que el String cambia sus valores, la posición del objeto cambia. En este caso, el objeto se encontraba en cero, y al sumarle 1, el objeto es movido en donde se encuentra el número 1 dentro de la Recta Numérica, tal como se ilustra en la figura 4.20.

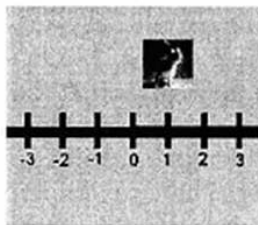


Fig. 4.20 Nueva posición del objeto "Conejo" después de realizar una suma.

Se crean controles para 4 objetos, 1 para el lado "A", y 3 para el lado "B". Los controles quedarán como se muestra en la figura 4.21.

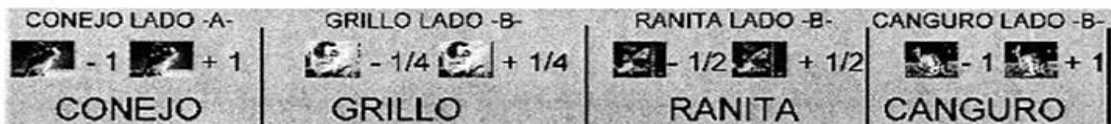


Fig. 4.21 Panel de control de la Recta Numérica

La figura 4.22 muestra como queda la exhibición una vez que el modelo hecho en 3D Studio ha sido exportado a código VRML. En el lado "A" (Ver figura 4.22) se puede observar una imagen de un "Conejo" (Ver figura 4.23), el cual inicia con la

posición cero dentro de la recta numérica. Los controles se encuentran en la parte inferior izquierda, e indican en su parte superior al lado de la recta al que pertenecen, en este caso el "Conejo" es el único que pertenece al lado "A". Debajo de el, 2 imágenes del conejo, los cuales son botones (censores) que indicarán si quiere sumar o restar una unidad a la posición del objeto.

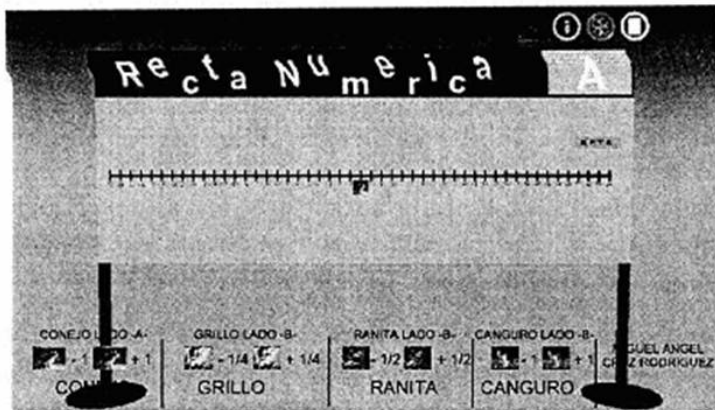


Fig. 4.22 Lado "A" de la Recta Numérica.

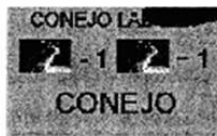


Fig. 4.23 Control de movimiento del "Conejo".

En la parte "B" (Ver figura 4.24) hay 3 imágenes, la primera de ellas un "Grillo" (Ver figura 4.25), la recta que le corresponde es la que se encuentra en la parte superior, en la recta numérica que hace referencia a la suma y resta de quebrados, en este caso de  $\frac{1}{4}$ . Los controles de este objeto se encuentran en segundo lugar, a la derecha de los primeros 2 botones pertenecientes al objeto "Conejo". En este caso las sumas y restas se harán a razón de  $\frac{1}{4}$  de unidad. Los resultados de la operación así como de los operandos se muestran debajo de los botones.

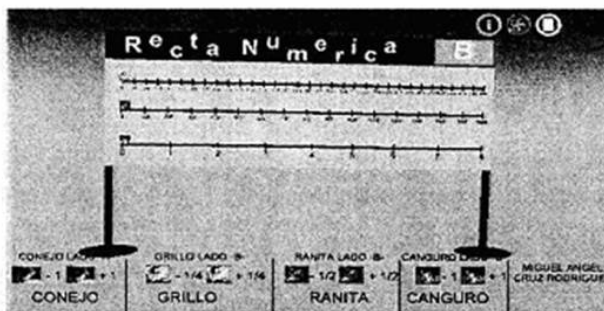


Fig. 4.24 Lado "B" de la Recta numérica.



Fig. 4.25 Control de movimiento del "Grillo".

A los controles los divide una línea, la división indica que los controles pertenecen a un objeto en específico, como se muestra en la figura 4.21.

Los otros dos objetos con los que se interactúa se puede hacer sumas y restas con fracciones de  $1/2$  ; y operaciones con fracciones con enteros. El lado "B" de la exhibición se puede apreciar las equivalencias de los números.

### 4.1.3.- Tiempo de descarga.

Un aspecto importante dentro de la aplicación es el tamaño de los archivos, ya que al ser una aplicación orientada a Internet, el tiempo requerido para el despliegue de las exposiciones no debe ser excesivo.

En la tabla 4.1 se muestra el tamaño de cada una de las exposiciones desarrolladas en la presente tesina y el tiempo estimado de descarga sobre distintas velocidades de conexión. Estos datos se obtuvieron mediante la herramienta VtmlPad la cual ofrece esta funcionalidad.

Exposición	Tamaño en Kb	Tiempos de descarga en segundos		
		Modem 14.4 Kb	Modem 28.8 Kb	MODEM 56.6-Kb
Recta Numérica	143	88	44	22

Tabla 4.1 Tiempos de descarga para la Recta Numérica.

### 4.1.4.- Exhibición "Recta Numérica" integrada.

En la figura 4.26 se muestra finalmente la exhibición "Recta Numérica", con todos los elementos integrados.

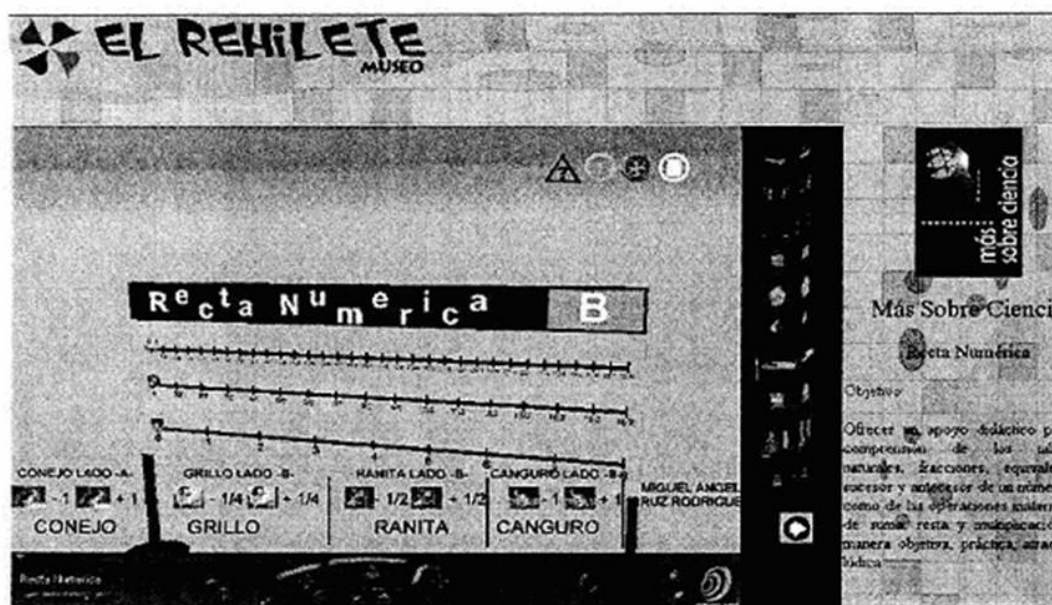


Fig. 4.26 Recta Numérica integrada.

## 4.2.- De Plantas, Aire, Agua y el Sol.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "De Plantas, Aire, Agua y el Sol". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

### 4.2.1.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se le dio comportamiento complejo.

#### 4.2.1.1.-Modelado 3D.

El modelado de esta exhibición requirió emplear más operaciones booleanas aplicadas a un mayor número de primitivas, tal como se muestra en la siguiente descripción. Se inició el modelado con una caja como se muestra en la figura 4.27.



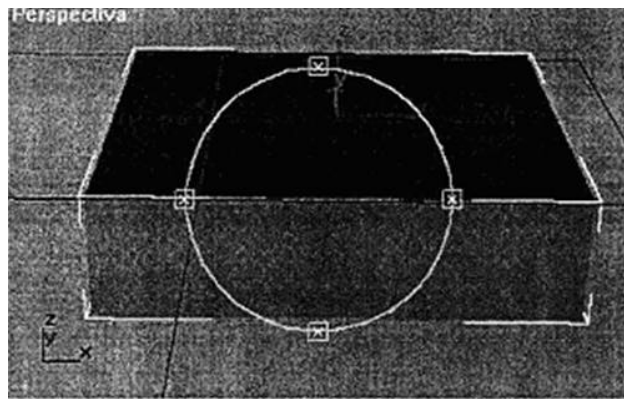


Fig. 4.27 Caja para iniciar a modelar

Esta tiene un corte a los 15 centímetros, en estos 15 centímetros tiene asiento la parte que contiene la imagen que se ilumina al terminar correctamente el diagrama. Para hacer el corte se necesita otra caja con dimensiones más grandes. Se rota y se coloca en la parte que se requiere ser eliminada de la primera caja, como se ilustra en la figura 4.28.

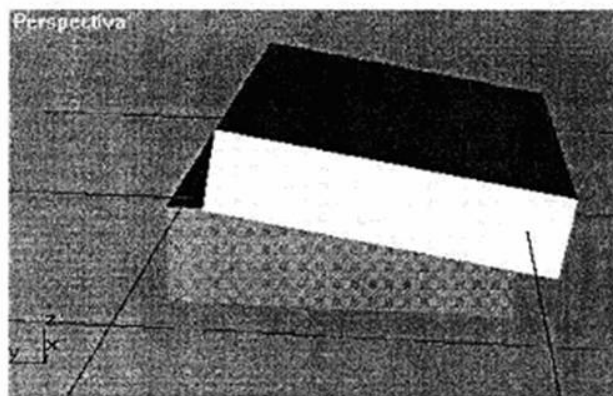


Fig. 4.28 Colocación de la segunda caja para hacer el corte.

Para iniciar el corte se selecciona primero la caja a la que se le va a hacer el corte. Después en el menú desplegable de la derecha seleccionamos "Objetos de composición", tal como se muestra en la Fig. 4.29.



Fig. 4.29 Menú para localizar los "Objetos de composición"

Del nuevo menú que aparece se selecciona "Booleano", en la figura 4.30 se muestra el menú en donde se encuentra al opción para hacer operaciones booleanas.



Fig. 4.30 Menú de Objetos de composición en donde se encuentran las operaciones booleanas.

Se selecciona la segunda caja la cual hará el corte. En la figura 4.31 se muestra como queda la caja después de la operación booleana, esta caja cortada queda de la forma que tiene el tablero de la exhibición en el mundo real.

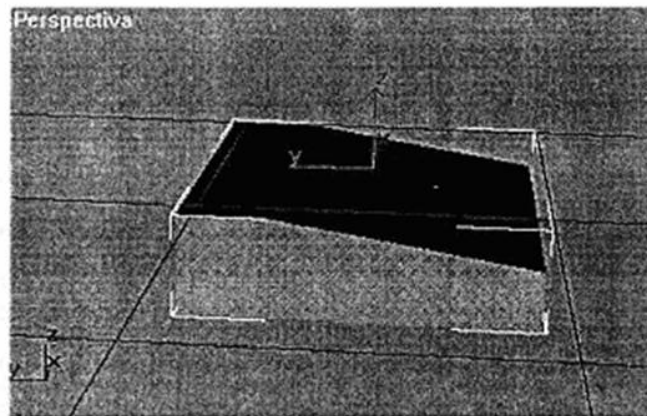


Fig. 4.31 Forma del objeto después de sufrir una operación booleana.

El siguiente paso es el modelar las piezas (fichas), con las cuales se forma el diagrama. Se crea primero una elipse del menú "Formas", y después se realiza un proceso de extrusión, ya que de los objetos que proporciona 3D Studio Max, ninguno da la forma elíptica que tienen las piezas.

Para extruir un objeto se selecciona la segunda pestaña que es la de "Modificar" y en menú desplegable, aparece la operación "Extruir".

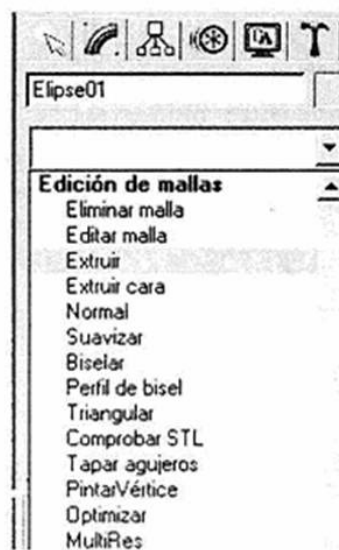


Fig. 4.32 Localización comando "Extruir"

Con las piezas y el tablero terminado, se clonan las piezas y se forman las hendiduras en el tablero, haciendo operaciones booleanas, con las piezas clonadas. El tablero tomará la forma que se muestra en la figura 4.33.

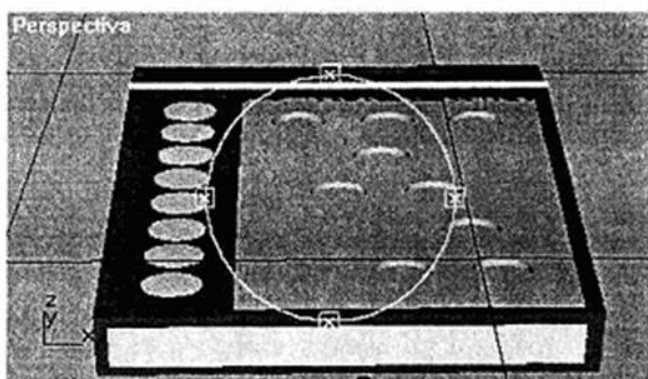


Fig. 4.33 Tablero terminado.

Se crean las bases sobre las que está el tablero y la parte en donde está colocado la imagen. Y finalmente el modelado queda como se muestra en la Fig. 4.34.

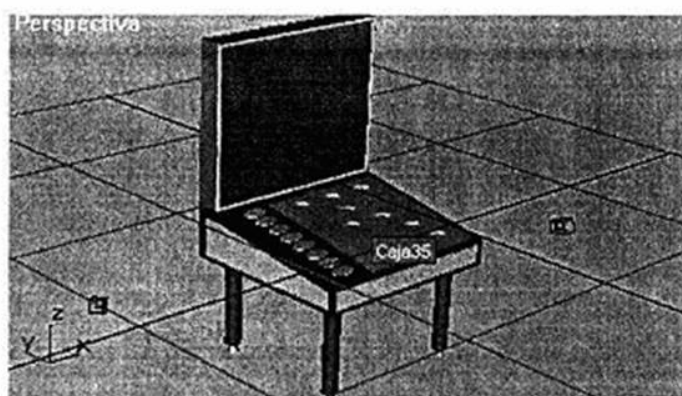


Fig. 4.34 Exhibición terminada (Fotosíntesis).

#### 4.2.1.2.- Comportamiento complejo

En la figura 4.35 se muestra la parte del código del programa en *JavaScript*, que lee la posición del objeto en sus 3 coordenadas que son asignadas a una variable de 3 vectores.

```
function Sd() ( valúe, ts)
{
    posG.x = valúe.x;
    posG.y = valúe.y; posG.z =
    valúe, z; tempString [0] =
    'DISCO 7 calculo ();
```

Fig. 4.35 Función para capturar posición de un objeto

A cada uno de los discos se les asigna una variable de 3 vectores y una función como la que se muestra en la figura 4.35, a fin de obtener la posición que tienen dentro del tablero. En la función llamada "calculo", se crea una condición, en donde en caso de que se encuentren los discos en la posición que les corresponde, se asigna un cambio de posición a un objeto mediante la variable de salida "value\_changed", y un cambio de valor a la variable string de salida outputString, con la cual se indicará que los discos han sido colocados correctamente (ver figura 4.36).

```
function calculo()
{
    outputString [0] =tempString[0];
    if(posA.x < -3.3 && posA.x > -3.9 && posA.y < -6.9 && posA.y > -7.5 && posB.x > -0.7 && posB.x
< 0 && posB.y < -4.2 && posB.y > -4.7 && posC.x < 1 && posC.x > 0.4 && posC.y < -7 && posC.y > -7.5
&& posD.x < -1.1 && posD.x > -1.7 && posD.y < -7 && posD.y > -7.6 && posE.x < -2.4 && posE.x > -3 &&
posE.y < -4 && posE.y > -4.55 && posF.x < -1.1 && posF.x > -1.7 && posF.y < -1 && posF.y > -1.6 &&
posG.x < 0.6 && posG.x > 0 && posG.y < -2.5 && posG.y > -3.1 && posH.x > 0.4 && posH.x < 1.1 &&
posH.y < -1 && posH.y > -1.6)
    { outputString[0] = temp String [0]
      value_changed [0] = -2.702;
      value_changed [ 1 ] = 4.52 ;
      value_changed [2] = -8.9 ;
    }
  else
  {
    outputString [0] ="
    value__changed[0] = -2.702;
    value__changed[1] = 4.52;
    value__changed[2] = -10.028 ;
  }
}
```

Fig. 4.36 Segmento de código, que determina cuando los discos se encuentran en la posición correcta.

### 4.2.2.- Pruebas.

Al inicio, la exhibición cuenta con los 8 discos en la parte izquierda del tablero, los cuales deben de moverse en la parte que le corresponda, y un disco estático dentro del tablero, el cual nos sirve como clave para ordenar a los demás. Para poder mover los discos el usuario da clic al disco, y sin soltarlo los arrastra hasta el orificio que crea que será el correcto para poder formar un cuadro sinóptico que nos permite organizar el proceso de la fotosíntesis.

En la figura 4.37 se muestra la exhibición en VRML al inicio del juego.

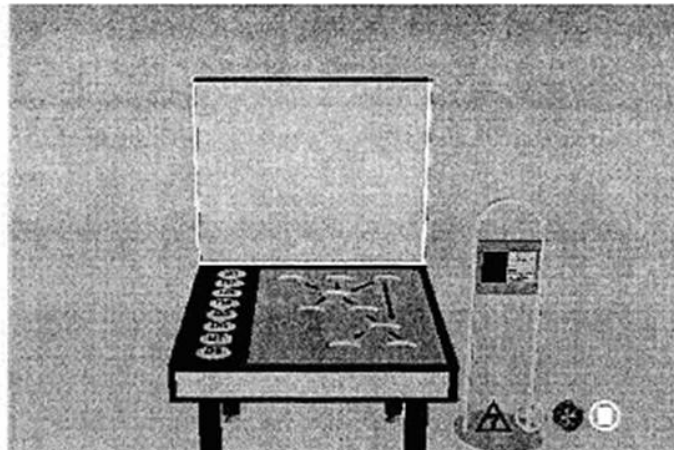


Fig. 4.37. Exhibición de la Fotosíntesis

Cuando se colocan los discos en forma correcta el *Script* acciona 2 nodos, uno de ellos despliega una leyenda que indica que las piezas están colocadas en forma correcta, y el otro nodo aparece una imagen de un paisaje, tal como se muestra en la figura 4.38.



Fig. 4.38. Exhibición de la Fotosíntesis tras interacción con el usuario

### 4.2.3.- Tiempo de descarga.

Estos tiempos de descarga corresponden a velocidades de conexión de líneas que van de los 14.4 kb a los 56 kb, en la Tabla 4.2 se muestran los tiempos de descarga.

Exposición	Tamaño en Kb	Tiempos de descarga en segundos		
		Modem 14.4 Kb	Modem 28.8 Kb	Modem 56.6 Kb
De plantas, agua, aire y el sol	317	198	99	49

Tabla 4.2 Tiempos de descarga para "De plantas, Agua, Aire y el Sol".

#### 4.2.4.- Exhibición "De plantas, Agua, Aire y el Sol" integrada.

En la figura 4.39 se muestra finalmente la exhibición "De plantas, Agua Aire y el Sol," con todos los elementos integrados.

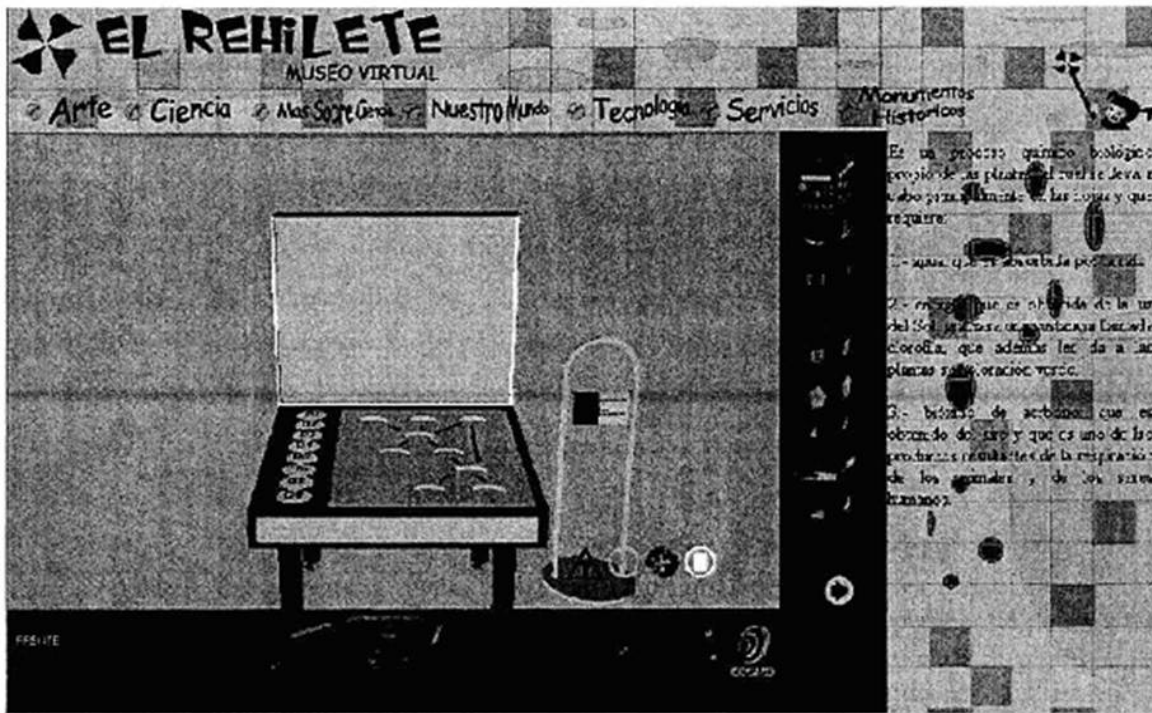


Fig. 4.39 Exhibición "De plantas, Agua, Aire y el Sol" integrada.

### 4.3 Torre de Hanoi.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "Torre de Hanoi". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

#### 4.3.1.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se le dio comportamiento complejo.

### 4.3.1.1.- Modelado 3D.

Para el modelado de la Torre de Hanoi, al hacer los discos se crea un cilindro con las dimensiones de los discos, y posteriormente se realiza la operación booleana. En la figura 4.40 se muestra el modelo 3D de uno de los discos, creado con 3D Studio Max.

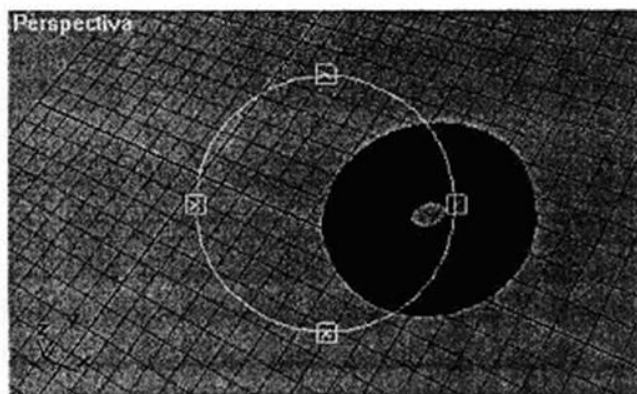


Fig. 4.40 Modelo 3D de disco para la Torre de Hanoi.

Para modelar la cubierta de la mesa, se crearon 2 prismas con diferentes dimensiones, uno de ellas con las dimensiones de 1 a m esa, y otro con 5 centímetros menos. El prisma de menor dimensión se clona, y con este se corta el prisma de mayor tamaño. El objetivo de cortar el prisma de mayor tamaño es el de evitar una sobre posición de objetos. La mesa quedará como se muestra en la figura 4.41.

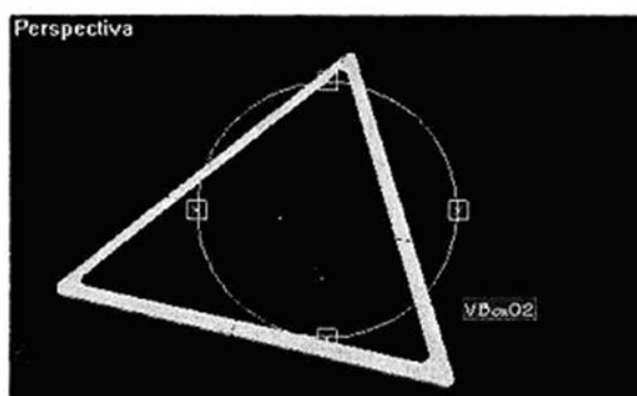


Fig. 4.41. Modelo 3D de la cubierta



Posteriormente se crean los postes y las patas de la mesa, ambos con cilindros, de modo que la exhibición terminada se muestra en la figura 4. 42.

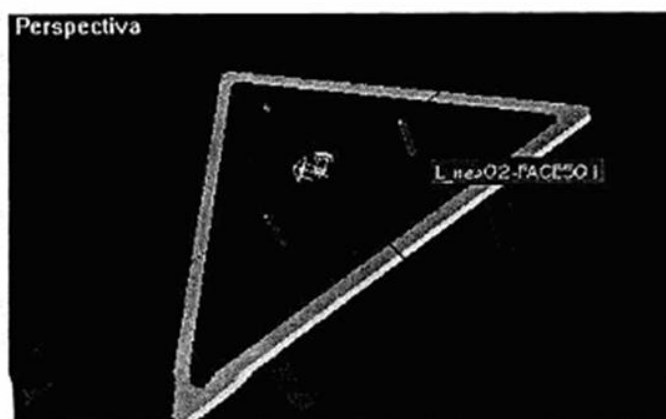


Fig. 4.42 Exhibición "Torre de Hanoi" terminada.

#### 4.3.1.2.- Comportamiento complejo.

En mundo virtual el usuario elige el número de discos con los que jugará. La exhibición final llama a 3 programas JavaScript para lograr la interactividad. Uno de ellos se emplea para activar y desactivar los nodos sensores (Texto "AGREGAR UN DISCO", y "QUITAR UN DISCO"). La figura 4.43 muestra el código correspondiente al *Script* antes mencionado.

```
DEF    ACTIVOS Script {
      eventIn  SFTime ESTADO
      eventIn  SFTime ESTADO_T
      eventOut SFBool ESTADO2
      url      "javascript:
      function ESTADO_T (){ ESTADO2=true;}
      function ESTADO () {ESTADO2=false;}
```

Fig. 4.43 Script para activar nodos sensores

Otro programa dota a los postes de la exhibición de propiedades para que se empleen como sensores. En la figura 4.44 se muestra el Script que se utilizó para activar y desactivar los postes. Cuando se esta configurando el juego, los postes no tiene la propiedad de un nodo sensor, sino que es hasta que se inicia el juego cuando estos se activan.

```
DEF  ACTIVOSpostes Script {
    eventIn  SFTIME ESTADO
    eventIn  SFTIME ESTADO_T
    eventOut SFBool  ESTADO2
    url      "javascript:
function ESTADO_T () { ESTADO2=false;}
function ESTADO () {ESTADO2=true;}
    "}
```

Fig. 4.44.- Script para activar postes como nodos sensores

El tercer programa JavaScript tiene el control de lo que despliega el contador del número de discos con los que se va a jugar, el contador del número de movimientos, calcula las nuevas coordenadas de los discos, crea los discos en tiempo de ejecución, etc.

Dentro del script tiene 2 funciones para configurar el número de discos, una detecta cuando el sensor para aumentar un disco es activado, cuando es activada el número de platos aumenta a razón de 1, con la condición de que no debe de ser mayor de 17, así mismo el número de discos es asignado a un string de salida. La función se muestra en la figura 4.45.

```
function disco_mas (){
    if(numero_platos< 17) {
        numero_platos+=1;
        numero_discos_texto [0]=''+numero_platos; } }
```

Fig. 4.45 Función para aumentar un disco.

Cuando el sensor para quitar un disco es activado, el número de platos disminuye a razón de 1, con la condición de que no debe de ser menor de 3, así mismo el número de discos es asignado a un string de salida. La función se muestra en la figura 4.46.

```
function disco_menos (){
    if(numero_platos>3) {
        numerojplatos--;
        numero_discos_texto [0]=''+numero_platos;
    }
}
```

Fig. 4.46 Función para disminuir un disco.

La función para reiniciar el juego actualiza a todas las variables como estaban en su etapa inicial, tal como se muestra en la figura 4.47.

ehilete

```

function reiniciar_juego(){
ok.removeChildren=nodosnuevos_;
for(i=0;i<numero_platos;i++) {
    nodosnuevos = new MFNodeO; nodosnuevos_[i]= nodosnuevos[0];
}
ok.addChildren=nodosnuevos_;
for (k=0 ;k<numero_platos ;k++) {
vector_numeros_platos_postel [k]=0;vector_numeros_platos_postel [k]=0;
vector_numeros_j>latos_postel [k]=0
}

movimientos=0;    movimientos_texto[0]=' '+movimientos
diferencia_platos= 2 numero_platos =3 radio_maximo= 0
distancia= .06 for(k=0;k<3;k++)
color[k]=0; numero_discos_texto[0]=' '+numero_platos;
poste_seleccionado 1 =false;poste_seleccionado2=false; poste_seleccionado3=false;
contador_postel=0;contador_poste2=0;contador_poste3=0;movimientos=0;
distanciároste 1=.06
distancia_poste2=.06
distancia_poste3=.06
i=0;    k=0;
}

```

Fig. 4.47 Función para reiniciar el juego.

La función "comenzar \_ jugar", que se muestra en la figura 4.48 crea a los discos, les da sus diferentes dimensiones de radio, les asigna color, y los coloca en el poste 1 con sus respectivas coordenadas en "Y".

```

function comenzar _ jugar(){
radio_maximo=. 005 *numero_platos;
for(i=0;i<numero_platos;i++) {
    for(k=0;k<3;k++) color[k]=(k+i)*.
    15*Math.random();
    plato-DEF platon'+i+' Transform { '
    plato+='children [ DEF sensor_plato'+i+' TouchSensor {enabled FALSE }
Shape { '
    plato+- appearance Appearance { '
    plato+='material Material { '
    plato+- diffuseColor '+color[0]+' '-fcolor[1]+' '+color[2];
    plato+='} }
    plato+=' geometry Cylinder {radius 10 height 2.125}'
    platos1}
    plato+=' }•
    nodosnuevos = new MFNode();
    nodosnuevos^Browser.createVrmlFromString(plato);
    nodosnuevos[0].translation=newSFVec3f(1.086,distancia,0.009517)
    nodosnuevos[0].scale=new SFVec3f(radio_maximo, .01, radio_maximo);
    distanciároste 1 =distancia;
    distancia+=".05;    radio_maximo=diferencia_platos/2000;
    diferencia_platos++; nodosnuevos_[i]= nodosnuevos [0];
    vector_numeros__platos_poste 1 [i]=i; contador_postel=i }
    distanciároste 1 =distancia;
    ok.addChildren=nodosnuevos_;
    contador _ poste 1 ++;}

```

Fig. 4.48 Función comenzar \_ jugar.

Existen 3 funciones necesarias para que los discos se muevan de un poste a otro. Cada una de estas funciones se activa cuando el usuario ha hecho un clic sobre el poste que le corresponde. Las funciones actuarán de emisor o receptor de disco, dependiendo de que poste fue primero seleccionado; si el poste 1 es seleccionado primero, significa que de él saldrá un disco, el cual será llevado al poste que sea seleccionado en segundo término. La función para el poste 1 se muestra en la figura 4.49.

```
function poste 1(){
  if (poste_seleccionado 1 =false&&poste_seleccionado2=false&&poste_seleccionado3=false) {
  for(k=0;k<i;k++)
  Browser.deleteRoute(ACTIVO,'nueva_cordenada_plato',nodosnuevos_[k],'translation');
  poste_seleccionado 1=true; poste_seleccionado2=false;
  poste_seleccionado3=false;
  Browser.addRoute(ACTIVO/nueva_cordenada_^lato',nodosnuevos_[vector_numeros_platosjpostel[contador
_po stel-l]],'translation'); contador_poste 1 —; distanciaroste 1 -=. 05
  }
  if(poste_seleccionado
  !=false){
  movimientos-H-;
  mo vimientos_texto [O]— '+mo vimientos;
  if(poste_seleccionado2=true)
  vector_numeros_platosjposte 1
[contador_poste 1 ]=vector_numeros_platos_poste2 [contador_poste2];
  if (poste_seleccionado3=true)
  vector_numeros_platos_poste 1
[contador_poste 1 ]=vector_numeros_platos_poste3 [contador_poste3];
  poste_seleccionado 1 =false; poste_seleccionado2=false;
  poste_seleccionado3=false; nueva_cordenada_plato.x= 1.086
nueva_cordenada_plato.y=distancia_poste 1
nueva_cordenada_plato.z=0.009517 distanciaroste 1+=. 05
contador_poste i ++;
  }
}
```

Fig. 4.49 Función para el poste 1.

### 4.3.2 Pruebas.

Para poder jugar con la exhibición en la parte inferior de la pantalla 2 sensores, uno de ellos, el que se encuentra a la izquierda es para agregar un disco como su nombre, de inicio el número de discos será de 3, podrá aumentar hasta 17 discos, y a la derecha esta el sensor con la leyenda "QUITAR UN DISCO", con el podrá disminuir el número de discos, el cual no podrá ser de menos de 3. Los sensores se muestran en la figura 4.50.



Fig. 4.50 Censores para configurar el número de discos.

En la parte superior izquierda se encuentra el texto "Comenzar a Jugar"(ver figura 4.51), el cual es un censor, que al dar clic sobre de él, el programa colocará el número de discos que el usuario haya elegido, y dará la propiedad de nodo censor a los postes.

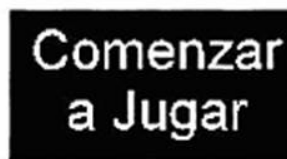


Fig. 4.51 Censor para comenzar a jugar.

En la figura 4.52 se muestra la exhibición cuando se está configurando, y en la figura 4.53 cuando ha sido configurada y se ha pulsado "Comenzar a jugar" y aparecen los discos que ha elegido



Fig. 4.52 Exhibición al momento de ser configurada.

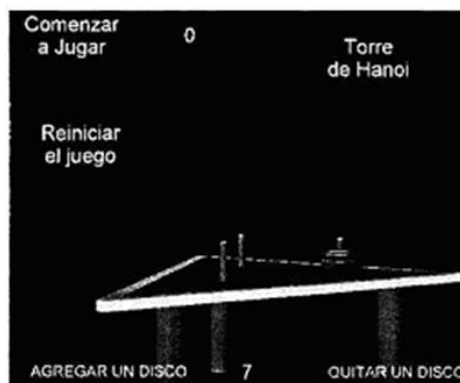


Fig. 4.53 Exhibición lista para comenzar a jugar

Después de haber configurado y pulsar "Comenzar a jugar", el usuario puede jugar, para ello, primero debe de pulsar el poste en donde se encuentra el disco que quiere mover, y después pulsar al poste a donde se desea mover. El disco que se moverá, será siempre el que se encuentra en la parte de arriba del poste.

Por ultimo tiene la opción de reiniciar el juego, ya sea porque ha logrado pasar los discos conforme a las reglas de juego, o para configurar nuevamente el número de discos. El censor "Reiniciar el juego" se encuentra debajo del censor "Comenzar a jugar", como se ilustra en la figura 4.54.

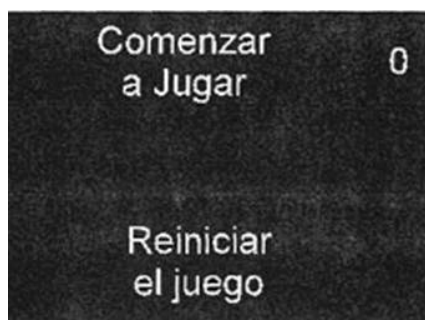


Fig. 4.54 Censor Reiniciar el juego.

#### 4.3.3.- Tiempo de descarga.

Estos tiempos de descarga corresponden a velocidades de conexión de líneas que van de los 14.4 kb a los 56 kb, en la Tabla 4.2 se muestran los tiempos de descarga.

Exposición	Tamaño en Kb	Tiempos de descarga en segundos		
		Modem 14 4 Kb	Modem 28 8 Kb	Modem 56 6 Kb
Torre de Hanoi	110	68	34	17

Tabla 4.3 Tiempos de descarga para Torre de Hanoi.

#### 4.3.4.- Exhibición "Torre de Hanoi" integrada.

En la figura 4.55 se muestra finalmente la exhibición "Torre de Hanoi", con todos los elementos integrados

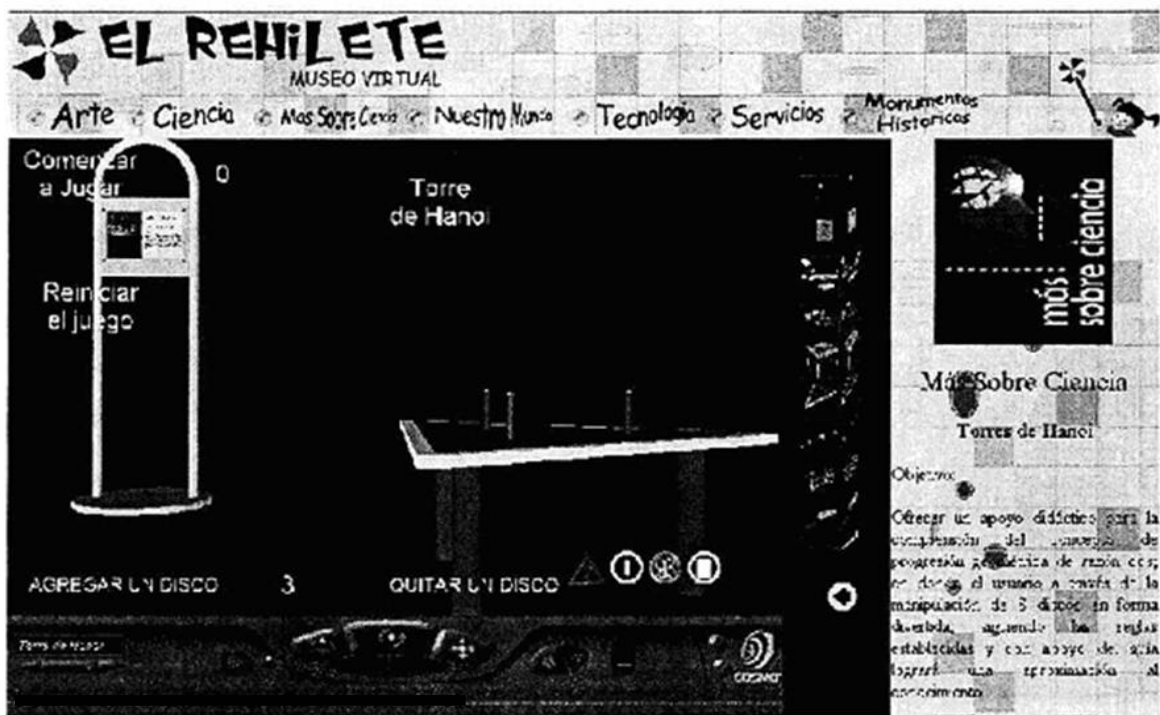


Fig. 4.55 Exhibición "Torre de Hanoi" integrada.

## Capítulo 5

# Desarrollo e Integración de los Mundos Virtuales con Animaciones.

### Resumen

En este capítulo se aborda la creación de las exhibiciones con animación, su desarrollo en 3D Studio Max, tomando medidas reales, se modelan los objetos, se les asigna color, y se adicionan las animaciones correspondientes.

### 5.1.-Gira, gira.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "Gira, gira". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

#### 5.1.1.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se le creo la animación.

##### 5.1.1.1.-Modelado 3D.

El modelado de la exhibición "Gira, gira", se inicio con la creación de la base, posteriormente se modelaron los tubos que sostienen el cable de protección y lo cables mismos, finalmente se crea la animación.

Para la base de la exhibición se crea primero un cilindro, el cual tiene un radio de 96 cm. y una altura de 40 cm. como se ve en la figura (5.1).

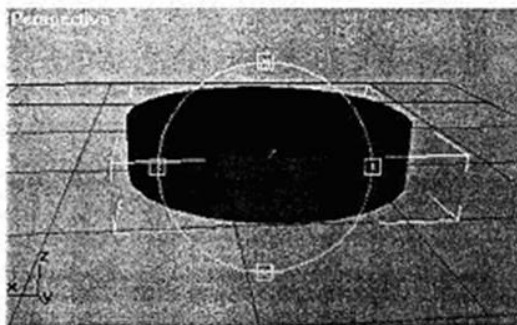


Fig. 5.1 Inicio del modelado de "Gira, gira".



A un objeto se le puede asignar un color, una textura, o un video, pero en la base hay 4 tipos de colores. La solución a este problema fue el hacer operaciones booleanas de resta, clonar el cilindro con el fin de tener 4 cilindros de diferentes colores y se procedió a hacer los cortes. En la figura 5.2 se muestra como queda la unión de los 4 cilindros con sus respectivos colores.

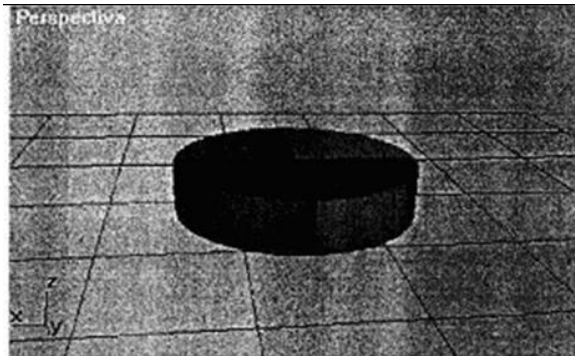


Fig. 5.2 Base de la exhibición "Gira, gira"

Se cubren los segmentos que forman la base, creando un círculo con el mismo radio, y después haciendo un proceso de extrusión de 0.5 cm.

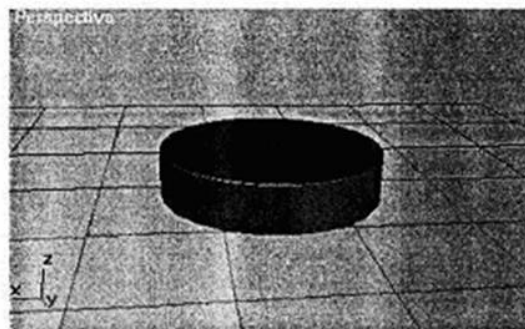


Fig. 5.3 Base de la exhibición con piso.

Para el escalón se creó un cilindro con 115 cm de radio, y se le hizo cortes para que el escalón se redujera en un solo segmento de la exhibición, como se muestra en la figura 5.4.

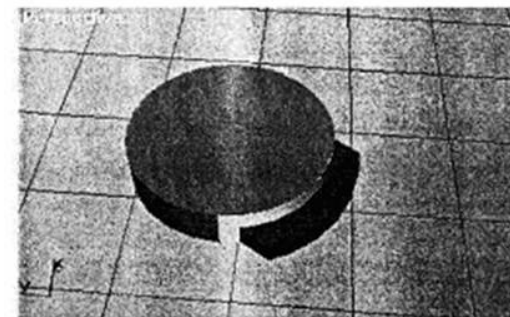


Fig. 5.4 Escalón de la exhibición "Gira, gira"

Los siguientes modelados fueron los postes y los cables. Los postes se crearon con cilindros. Para crear los cables se hizo un proceso de "Solevado"<sup>2</sup>, para ello se crearon arcos que dan la forma de una cuerda, se creó un círculo con el ancho de la cuerda.

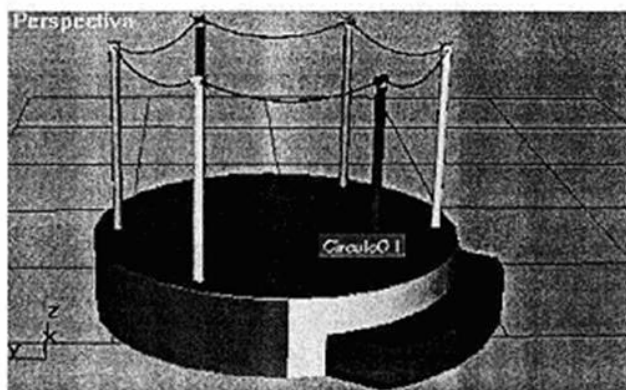


Fig. 5.5 Colocación de postes y cuerdas.

La exhibición terminada en 3D Studio Max se muestra en la figura 5.6.

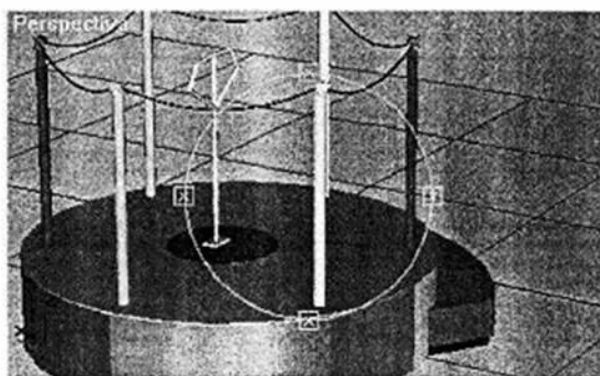


Fig. 5.6 Exhibición "Gira, gira" terminada.

#### 5.1.1.2. Adición de animación.

Los brazos, el torso, los pies, la cabeza, todo lo que conforma al avatar está creado en forma independiente. En 3D Studio Max se puede agrupar todos ellos para hacerlos girar, pero también se pueden desagrupar y trabajar con ellos en forma independiente. Para crear una animación en 3D Studio Max, está el icono de animar en la parte inferior de la pantalla. El icono se muestra en la figura 5.7.

---

<sup>2</sup>Solevado es la creación de objetos 3D a partir de objetos 2D como sección y recorrido.

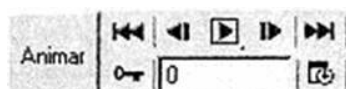


Fig. 5.7 Icono para animar

Cuando se crea una animación, debemos de cerciorarnos de que las llaves estén en cero (ver figura 5.7). La posición inicial del avatar se ilustra en la figura 5.8.



Fig. 5.8 Fase inicial de la animación.

Para iniciar a dar movimiento al avatar, se da los movimientos de los brazos y del torso. Con el movimiento de los brazos y torso terminado, se agrupa al avatar con el tubo y se gira sobre el eje "Y" para dar el giro que se hace en la exhibición. Posteriormente el modelo creado en 3D Studio Max se exporta a VRML.

La animación exportada tiene que ser reprogramada en VrmIPad. Se coloca un TouchSensor para que este active al TimeSensor y así activar la animación. Se cambia 'loop' a "FALSE", y el tiempo de la animación se aumenta a 12 segundos "cycleInterval 12".

El último paso para dar por terminada la animación es colocar las rutas, iniciando con el TouchSensor. Parte de las rutas se muestran en la figura 5.9,

```
ROUTE Sensor.touchTime TO mmm-TIMER.startTime
ROUTE mmm-TIMER.fraction_changed TO mmm-POS-INTERP.setJraction
ROUTE mmm-POS-INTERP.valúe changed TO mmm.translation
```

Fig. 5.9 Rutas para animar la exhibición "Gira, gira".

### 5.1.2.- Pruebas,

En esta etapa las exhibiciones son probadas para ver que funcionen correctamente.

Para accionar la animación para la exhibición "Gira, gira", se pulsa sobre el avatar, el cual funciona como censor, en la figura 5.10 se muestra la exhibición con el avatar en VRML antes de dar clic para iniciar la animación

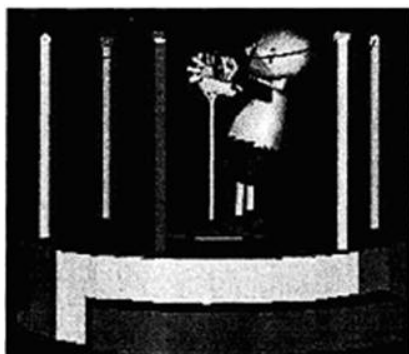


Fig. 5.10 Exhibición antes de iniciar la animación.

En la figura 5.11 se muestran los diferentes puntos de la animación.



Fig. 5.11 Diferentes puntos de la animación

### 5.13.- Tiempo de descarga.

Estos tiempos de descarga corresponden a velocidades de conexión de líneas que van de los 14.4 kb a los 56 kb, en la Tabla 5.1 se muestra que los tiempos de descarga .

Exposición	Tamaño en Kb	Tiempos de descarga en segundos		
		Modem 14.4 Kb	Modem 28.8 Kb	Modem 56.6 Kb
Gira, gira	49	30	15	7

Tabla 5.1 Tiempos de descarga para Gira, gira.

### 5.1.4.- Exhibición "Gira, gira" integrada.

En la figura 5.12 se muestra finalmente la exhibición "Gira,gira", con todos los elementos integrados.

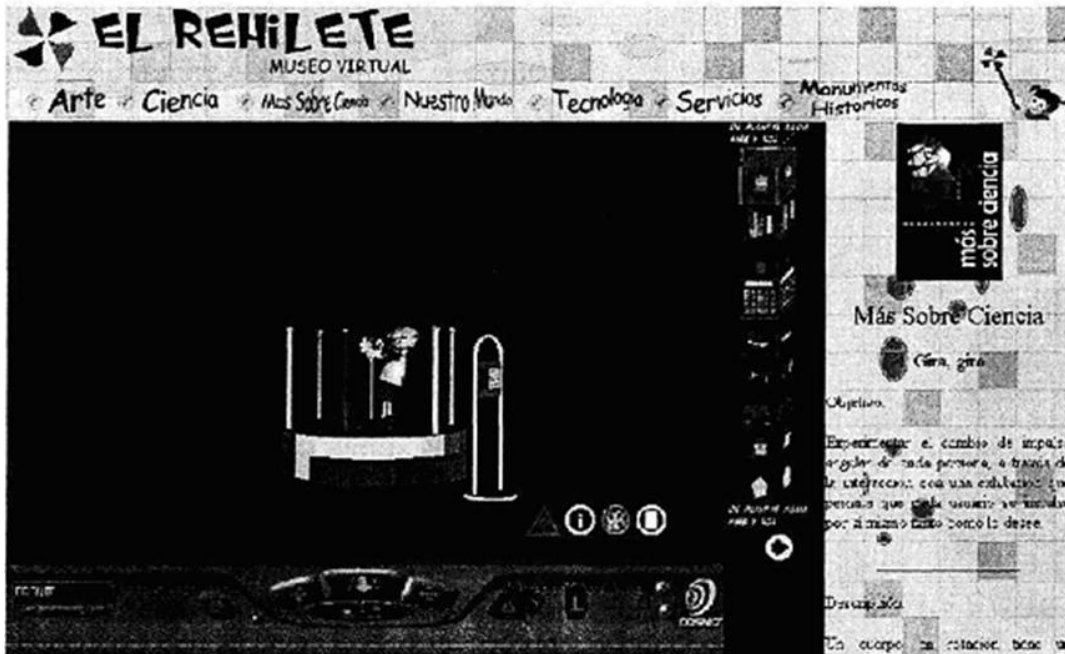


Fig. 5.12 Exhibición "Gira, gira" integrada.

## Capítulo 6

# Desarrollo e integración de Mundos Virtuales (Visualizaciones).

### Resumen

En este capítulo se aborda la creación de las exhibiciones nivel de representación de visualización, su desarrollo en 3D Studio Max, tomando medidas reales, la asignación de color, y también la forma en la que se les asigno texturas, y la simulación de los espejos de la exhibición "Caleidoscopio".

### 6.1.- Meteoritos.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "Meteoritos". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

#### 6.11.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se texturizaron los meteoritos.

##### 6.1.1.1.-Modelado 3D.

El modelado de la exhibición "Meteoritos", se inicia con la creación de una caja que es donde se colocan los meteoritos.

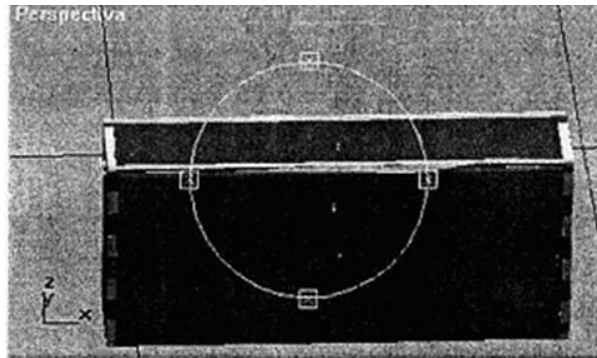


Fig. 6.1 Inicio del modelado de "Meteoritos".

En la orillas de la caja tiene bordes de color café los cuales son creadas con cajas, como se muestra en la figura 6.1.

La caja se sostiene por 2 bases cilíndricas con tapas y fondos de madera, estas bases están unidas por 2 tubos colocadas horizontalmente, como se muestra en la figura 6.2.

En la figura 6.2 se muestra el mueble terminado sobre el que se exhiben los meteoritos.

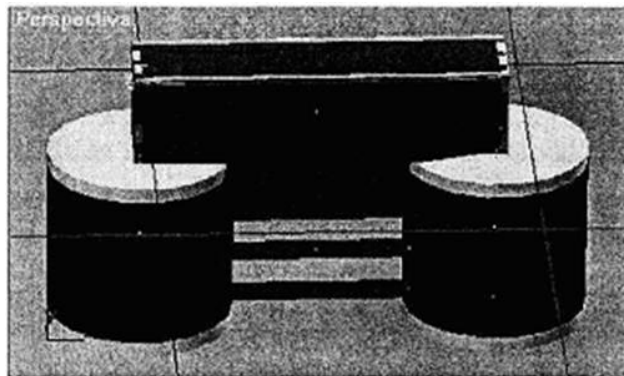
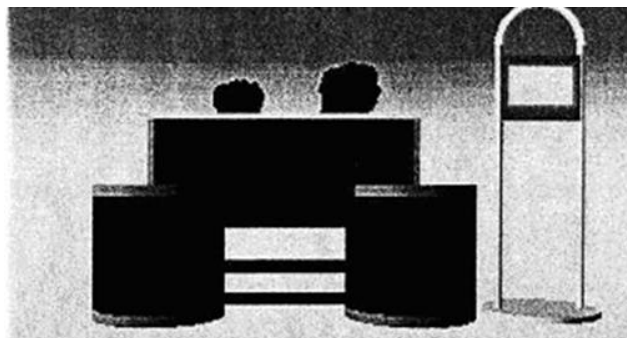


Fig. 6.2 Mueble de la exhibición "Meteoritos".

Los meteoritos son creados con *Geoesferas*, para después ser deformadas. En la figura 6.3 se muestra el modelado terminado y exportado a VRML.



En la figura 6.4 se muestra una fotografía extraída de la realidad, y se puede apreciar la similitud entre la realidad y la realidad virtual.

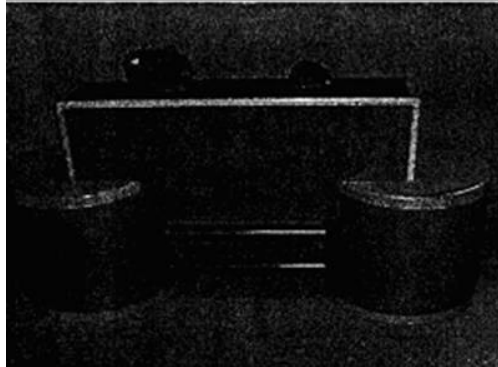


Fig. 6.4 Fotografía de la exhibición.

### 6.1.1.2.- Texturizar meteoritos.

Para texturizar los meteoritos, se tomó una fotografía digital a los meteoritos, de ellas se recortó solo la parte de los meteoritos con un editor de imágenes.

Para que la imagen esté colocada en forma correcta se utilizó el Internet Space Builder, para ello la exhibición debe estar en formato ".wrl". El modelo 3D en este paquete se muestra en la figura 6.5.

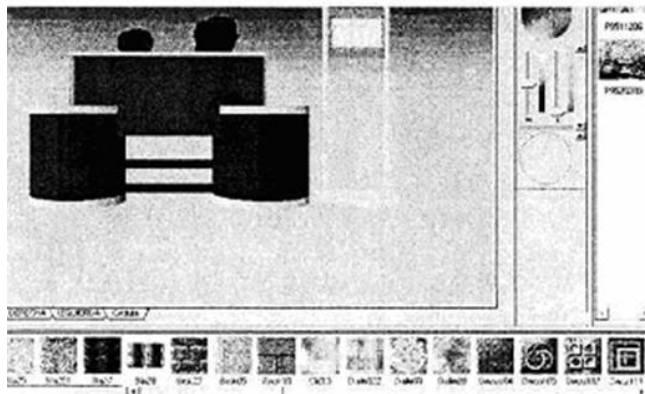


Fig. 6.5. Modelado de "Meteoritos" en Internet Space Builder.

Se toma la textura y se asigna a las 2 Geoesferas. La textura en las geoesferas se muestran en la figura 6.6.



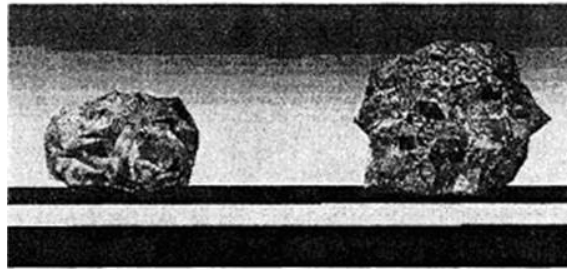


Fig. 6.6 Geoesferas deformadas con textura.

Para las tapas se toma una de las texturas que Internet Space Builder trae prediseñadas. La exhibición terminada se muestra en la figura 6.7.

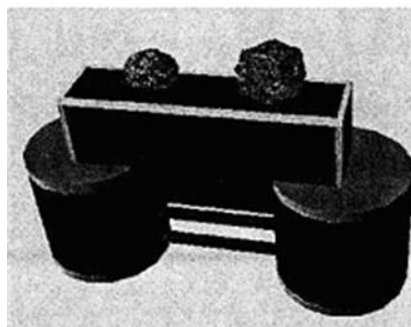


Fig. 6.7 Exhibición "Meteoritos" terminada.

### 6.1.2.- Tiempo de descarga.

Estos tiempos de descarga corresponden a velocidades de conexión de líneas que van de los 14.4 kb a los 56 kb, en la Tabla 6.1 se muestra que los tiempos de descarga.

Exposición	Tamaño en Kb	Tiempos de descarga en segundos		
		Modem14.4Kb	<b>Modem28.8Kb</b>	<b>Modem56.6Kb</b>
Meteoritos	265	162	82	41

Tabla 6.1 Tiempos de descarga para Meteoritos.

### 6.1.3.- Exhibición "Meteoritos" integrada.

En la figura 6.8 se muestra finalmente la exhibición "Meteoritos", con todos los elementos integrados.

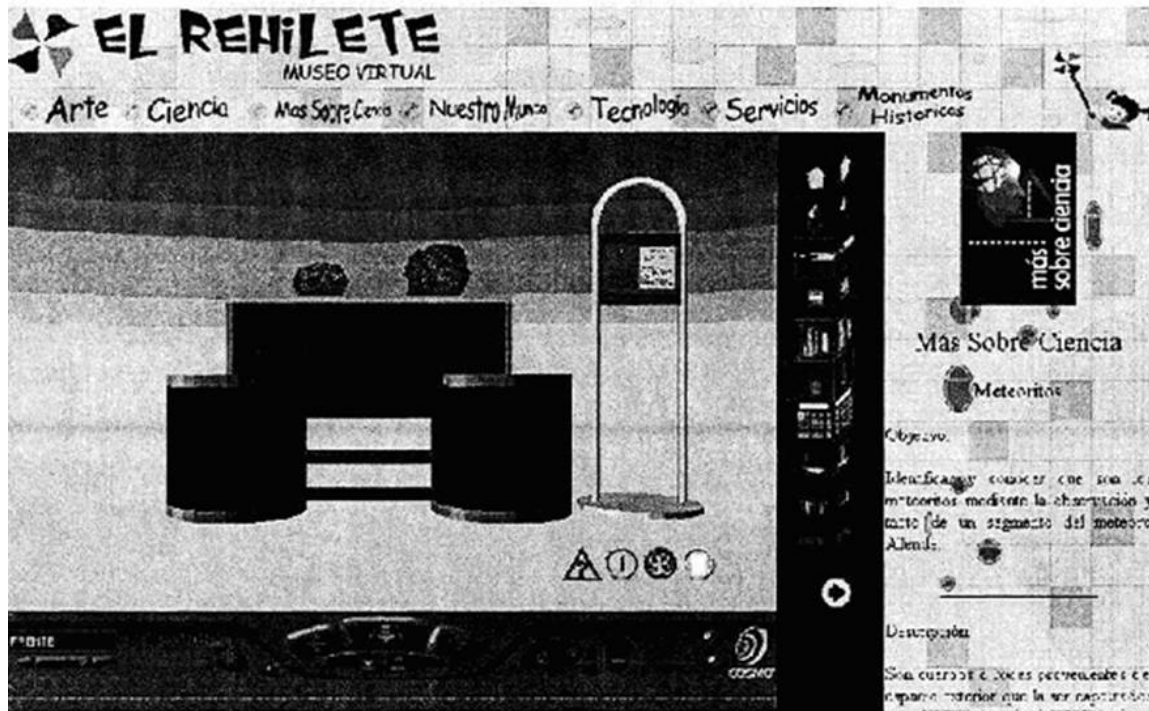


Fig. 6.8 Exhibición "Meteoritos" integrada.

## 6.2.- Caleidoscopio.

En esta sección se aplica el método de desarrollo de mundos virtuales para la exhibición llamada "Meteoritos". Se verá el desarrollo de la exhibición, pruebas de funcionamiento, tiempos de descarga y la integración de la misma con todos sus componentes.

### 6.2.1.- Desarrollo de la exhibición.

El desarrollo de la exhibición consta de la creación del modelado 3D de los objetos que conforman la exhibición, y la forma en como se simulo las reflexiones del espejo.

### 6.2.1.1.- Modelado 3D

Para modelar la exhibición llamada "Caleidoscopio" se creó la estructura sobre la cual se sostienen los espejos. La estructura en la realidad esta hecha de madera rectangular, en 3D Studio Max fue creada con cajas (Ver figura 6.9).

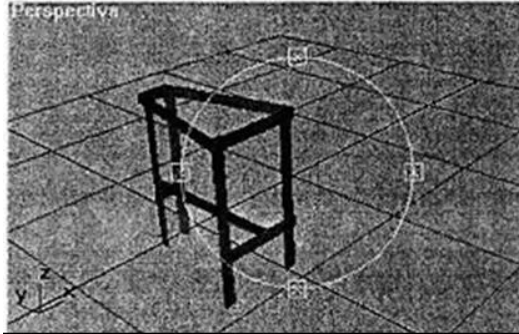


Fig. 6.9 Estructura de la base de la exhibición "Caleidoscopio".

Posteriormente se modeló los 3 espejos y las 3 cubiertas. Para finalmente tener el modelado terminado como se muestra en la figura 6.10.

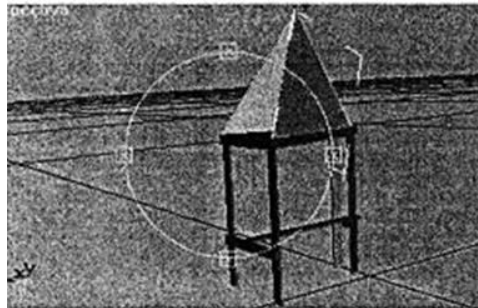


Fig. 6.10 Modelado terminado de la exhibición "Caleidoscopio".

En el Museo el Rehilete, en la exhibición se introduce un objeto entre los 3 espejos. En el lenguaje VRML no tiene la función de poder crear espejos y sus características.

### 6.2.1.2.- Simular espejo.

Para simular el efecto que tiene el caleidoscopio, se asignó en 3D Studio Max la propiedad de *RayTrace* a los espejos. Después se colocó una esfera dentro de los espejos (Ver figura 6.11), y se represento la escena en una imagen, para posteriormente guardar la imagen en un archivo jpg

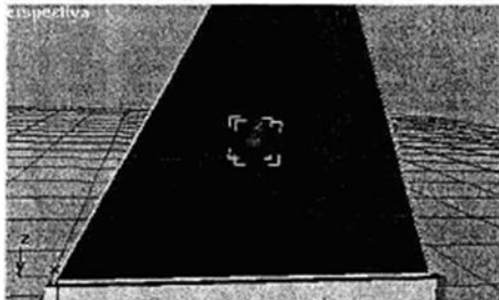


Fig. 6.11 Esfera dentro de los 3 espejos de el "Caleidoscopio".

La imagen que se produjo en 3D Studio Max, de los 3 espejos con la propiedad "RayTrace" se muestra en la figura 6.12.

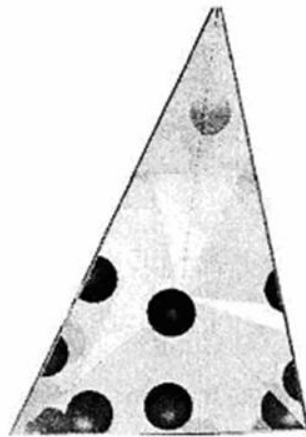


Fig. 6.12. Imagen producida por los 3 espejos en 3D Studio Max.

La imagen fue colocada en los espejos en VRML, con ayuda de Internet Space Builder. La esfera que se utilizó para crear la imagen con las reflexiones en 3D Studio Max, fue exportada y colocada en el archivo del modelado. La exhibición terminada se muestra en la figura 6.13.

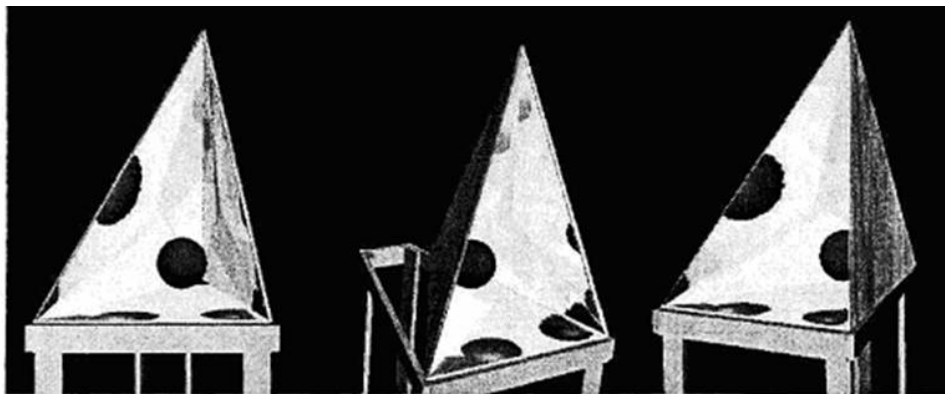


Fig. 6.13 Diferentes ángulos de la exhibición "Caleidoscopio".

## 6.2 Tiempo de descarga

Estos tiempos de descargas corresponden a velocidades de conexión de líneas que van desde los 14.4 Kb, a los 56 Kb, en la tabla 6.2 se muestran los tiempos de descarga.

Exposición	Tamaño en Kb	Tiempo de descarga en segundos		
		Modem 14.4 Kb	Modem 28.8Kb	Modem 56.6 Kb
Caleidoscopio	45	28	14	7

Tabla 6.2 Tiempos de descargas para caleidoscopio

## 6.2.3 Exhibición “Caleidoscopio” integrada

En la figura 6.14 se muestra finalmente la exhibición “Caleidoscopio”, con todos los elementos integrados

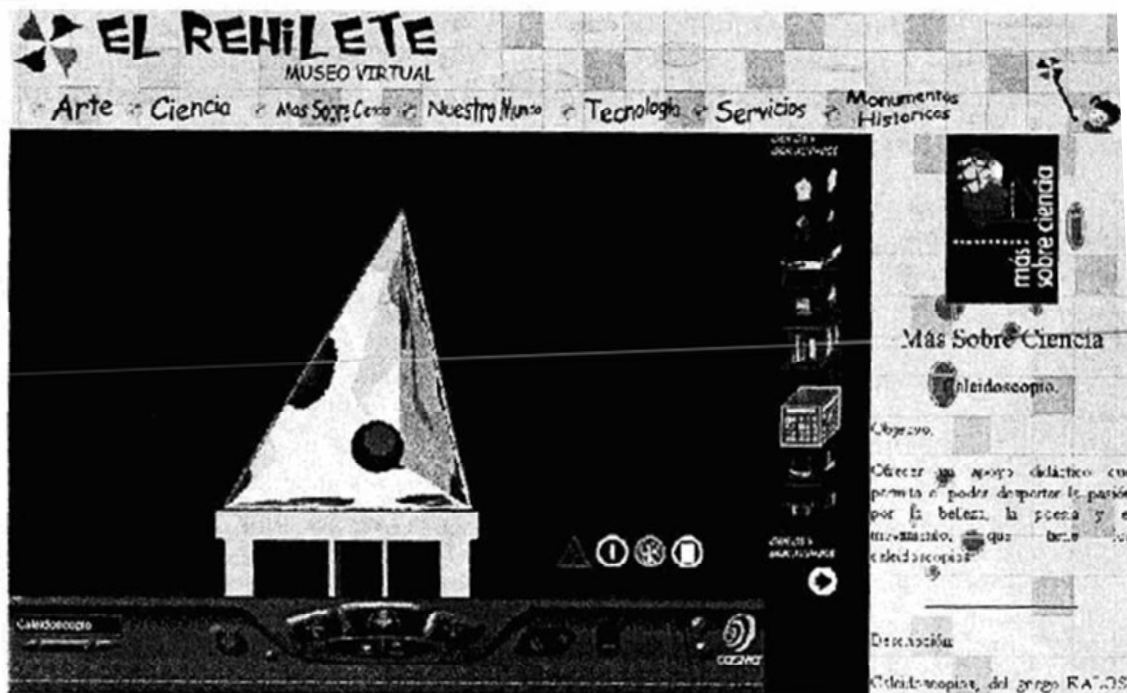


Fig. 6.14 Exhibición “Caleidoscopio” integrada.

## Capítulo 7

# Conclusiones y Perspectivas.

### 7.1.-Conclusiones.

En la presente tesina se abordó el desarrollo de las exhibiciones "Recta Numérica", "De Plantas, Aire, Agua y el Sol", "Torre de Hanoi", "Gira, gira", "Meteoritos", y "Caleidoscopio", exhibiciones pertenecientes al área de Más Sobre Ciencia, del Museo el Rehilete, que se virtualizaron para el proyecto del "Museo Virtual 3D El Rehilete".

El museo virtual 3D el Rehilete pretende cumplir y superar las expectativas de los usuarios de Internet, al ofrecer un ambiente virtual tridimensional, que tiene además un valor agregado, el cual es la interactividad que tendrá el usuario con algunas exhibiciones. Dicha interactividad proporcionada por medio de comportamiento complejo permite a los usuarios desarrollar actividades que se llevan a cabo en el Museo el Rehilete real.

Se utilizaron herramientas de desarrollo como 3D Studio Max, Internet Space Builder, Macromedia Flash y Photoshop, con lenguajes estándares para el diseño Web como HTML, VRML lenguaje que permite la interactividad con gráficos tridimensionales y que puede transmitirse a través de la "www".

El modelado tridimensional en su mayoría fue creado en 3D Studio Max, esta herramienta facilitó el modelado de las exhibiciones, creando objetos en forma gráfica, permitiendo hacer cortes a los objetos, y la manipulación de la posición de los vértices, con ello creando objetos complejos idénticos a los de la realidad.

En el transcurso del modelado se tuvo problemas básicamente por el tamaño del archivo; que es ocasionado por asignar un mayor número de segmentos a los objetos; los objetos modelados se ven mucho mejor definidos, pero con mayor tamaño del archivo, y mucho mayor número de vértices.

Un archivo de tamaño grande hace que aumente el tiempo de descarga, y el número de vértices hace que el procesador trabaje más, y el render sea más lento.

En este tipo de trabajos resulta mucho mejor el sacrificar los detalles para tener un archivo mucho más ligero, con lo que se cumplirá el objetivo de su diseño, el cual es publicarlo en Internet.

El proceso para asignar texturas a los objetos fue realizado con Internet Space Builder, desde 3D Studio Max; en su mayoría sólo fue necesario asignar color con los colores estándar que tiene 3D Studio Max.

## 7.2.- Perspectivas.

Las exhibiciones presentadas pueden ser mejoradas en un trabajo futuro en los siguientes puntos:

- "Recta Numérica": agregar una función al Script para detectar cuando existiesen equivalencias en el "lado A", así mismo dar aviso del suceso.
- "Torre de Hanoi": agregar una función la cual indique cuando los discos han sido colocados en forma correcta.
- "Caleidoscopio": esta podría ser una de las exhibiciones que tardaría en mejorar, ya que los lenguajes hasta el momento no integran algún nodo para poder crear refracciones, para crear espejos.

Para el desarrollo de gráficos 3D existen lenguajes para crear mundos como los desarrollados en este proyecto con VRML, algunos de ellos son Java3D, o las librerías de OpenGL, o Cult3D, entre otras más.

Los lenguajes y librerías viables para el desarrollo de mundos virtuales como el del "Museo virtual 3D el Rehilete", son Java3D,y Cult3D, por ser formatos de representación tridimensional orientados a Internet. En cambio OpenGL es mucho más rápido para desplegar gráficos 3D, pero sus aplicaciones no son orientados a Internet. X3D aun no hay especificación.

## Referencias:

[1] 3D Studio Max R3.  
Kinetix(1999).

[2] 3D Studio Max , Fundamentals.  
Michael Todd & Steve Burke. New  
Riders Publishing (1999).

[3] Designing 3D Graphics.  
Josh white.  
Wiley Computer Publishing (1996).

[4] Ingeniería de software un enfoque práctico.  
Roger S. Pressman. 4ta. Edición.  
MacGrawHill(1998).

[5] Ingeniería de Software Explicada .  
Mark Norris & Peter Rigby Noriega  
Editores (1994).

[6] Ingeniería de Software Ian  
Sommerville Addison-Wesley  
Iberoamericana.

[7] HTML, Referencia visual.  
2da. Edición.  
Dean Scharf.  
Prentice-Hall Hispanoamericana, S. A. (1996).

[8] JavaScript, soluciones instantáneas.  
Rick Darnell.  
Prentice-Hall Hispanoamericana, S.A.(1997).

[9] Lo real y lo Virtual.  
Tomas Maldonado.  
Editorial Gedisa (1994).

[10] Realidad Virtual, aplicaciones prácticas en los negocios y la industria.  
Dimitris N. Chorafas & Heinrich Steinmann. Prentice-Hall  
Hispanoamericana S. A. (1996).



- [11] Programación en JavaScript.  
Osear González Moreno.  
Ediciones Anaya Multimedia, S. A. (1998).
- [12] VRML modelado y diseño.  
Mark Pesce.  
Editorial Prentice may (1997).
- [13] 3DStudioMax.  
[http://www.unav.es/cti/manuales/3DStudioMax/#\\_Toc496441709](http://www.unav.es/cti/manuales/3DStudioMax/#_Toc496441709)
- [14] Evolución del VRML.  
<http://www.ucm.es/info/Psyap/Prieto/alum9798/virtual/virnet/webmaster.htm>
- [15] Historia de la Realidad Virtual.  
<http://www.terra.es/personal/trvOOOO1ZhistoRv.htm>
- [16] HTML.  
<http://www.etsit.upm.es/~alvaro/manual/manual.html#1>
- [17] Internet Space Builder.  
<http://www.parallelgraphics.com>
- [18] Internet Space Builder.  
<http://www.ensemble3d.com/isb.html>
- [19] Internet Space Builder.  
<http://software.latinguia.com/^tual/programa.phtml?id=2207>
- [20] Java Script.  
[http://www.redestb.es/javaaula/leccion1/javaintr.htm#JScript\\_Java](http://www.redestb.es/javaaula/leccion1/javaintr.htm#JScript_Java)
- [21] Java 3D.  
<http://www.programacion.com/java/tutorial/3d/>
- [22] Java 3D.  
<http://www.labvis.unam.mx/elio/J3D/tutorialJ3D.html>
- [23] Java 3D. <http://java.sun.com/products/java-media/3D/>
- [24] Macromedia Flash  
<http://www.imagecampus.com.ar/flashexpert.html>
- [25] Museo el Rehilete.  
<http://www.rehilete.org.mx>

[26] OpenGL.  
<http://pgrafica.webideas4all.comyContenido.html>

[27] OpenGL.  
<http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/January1998/article15.htm>

[28] Paseos virtuales.  
[http://www.vec.ca/Spanish/4/vancouver\\_360\\_tours.CFM](http://www.vec.ca/Spanish/4/vancouver_360_tours.CFM)

[29] Paseos virtuales.  
<http://www.infonetconsultores.com>

[30] Realidad virtual.  
<http://www.activamente.com.mx/vrml/>

[31] Realidad virtual.  
<http://www.rvltada.com/>

[32] VRML.  
<http://www.activamente.com.mx/vrml/historia.html>

[33] VRML  
<http://dmi.uib.es/~abasolo/cursovrml/diapos/063.html>

[34] VRML  
<http://dmi.uib.es/~abasolo/cursovrml/diapos/080.html>

[35] VrmIPad.  
<http://software.latinguia.com/Intemtual/>

[36] VrmIPad.  
<http://www.desarrolloweb.com>

[37] X3D.  
<http://dmi.uib.es/~abasolo/cursox3d/diapos/III-013.html>

## **Glosario.**

### **Aplicación:**

Comúnmente, este término se refiere a los programas, o software, que utilizamos en nuestra computadora. Por ejemplo, Microsoft Word es una aplicación de procesamiento de palabras.

### **Browser:**

Es un programa que sirve buscar y visualizar información de la WWW. Los browsers de VRML son programas que interpretan el contenido de un programa en este lenguaje y muestran en pantalla el escenario virtual descrito por ese programa.

### **Extruir:**

Modificar un objeto haciendo que éste incremente su tamaño en determinado eje (x, y o z).

### **Geoesferas:**

Las geoesferas generan una superficie más regular que las esferas normales y además se representan con un perfil algo más uniforme que éstas para un mismo número de caras. A diferencia de una esfera normal, la geoesfera no tiene polos, lo que puede ser una ventaja cuando se aplican ciertos modificadores como los de deformación de forma libre (FFD).

### **HTML:**

Es un lenguaje informático utilizado para crear documentos hipertexto. El HTML utiliza una lista de rútilos, que describe la estructura general de varios tipos de documentos enlazados entre si en el World Wide Web.

### **Internet:**

Es la interconexión de redes informáticas que permite a los ordenadores o computadoras conectadas comunicarse directamente. El término suele referirse a una interconexión en particular, de carácter planetario y abierto al público, que conecta redes informáticas de organismos oficiales, educativos y empresariales.

### **Operaciones Booleanas:**

Operaciones que se pueden aplicar a los objetos 3D (3D Studio Max), unión, intersección, sustracción.

**Plug-in:**

Es un programa que se conecta a un programa anfitrión para extender la funcionalidad de este último. El programa huésped tiene ciertas capacidades que el anfitrión no tiene. Cosmo Player es un Plug-In, que permite a browser de WEB, como Explorer, presentar al usuario escenas VRML.

**Realidad Virtual:**

La Realidad Virtual es una representación de las cosas a través de medios electrónicos, que nos da la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea. Los usuarios se mueven por una Realidad Virtual como si estuviesen en un mundo real. La Realidad Virtual puede ser de 2 tipos: inmersiva y no inmersiva.

**Realidad Virtual inmersiva:**

Los métodos inmersivos de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano.

**Realidad Virtual no inmersiva:**

La Realidad Virtual no inmersiva utiliza medios como el que actualmente nos ofrece Internet en el cual podemos interactuar a tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora.

**Recorrido Virtual:**

Un recorrido virtual es una exploración interactiva de un espacio simulado, con el fin de dar conocimiento de un sitio real sin necesidad de estar presente en éste. Este producto es una excelente opción para las empresas que desean hacer guías turísticas o didácticas, o desean promover un sitio real.

**3D:**

Tres Dimensiones (x, y, z). Se refiere a las dimensiones que un objeto tiene: largo, ancho y alto.