



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA

RECONSTRUCCIÓN DE SITIOS
ARQUEOLÓGICOS: TEOTIHUACAN

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A N :
AVENDAÑO MARTÍNEZ JOSÉ CARLOS
SALGADO RODRÍGUEZ JOSÉ FRANCISCO

A S E S O R D E T E S I S :
ING. SANTIAGO IGOR VALIENTE GÓMEZ

CIUDAD UNIVERSITARIA, MAYO 2004



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA
DIVISION DE INGENIERIA ELECTRICA**

PROPUESTA DE SEMINARIO (X) Y/O TESIS (X)

DEPARTAMENTO DE ADSCRIPCION DEL PROFESOR: COMPUTACIÓN
CLAVE (34)

CARRERA INGENIERO ELECTRICO ELECTRÓNICO (GEN. 94 Y POSTERIORES)
CLAVE () NO. DE ALUMNOS _____

CARRERA INGENIERO MECANICO ELECTRICISTA (GEN. 93 Y POSTERIORES)
MODULO _____ CLAVE () NO. DE ALUMNOS _____
MODULO _____ CLAVE () NO. DE ALUMNOS _____

CARRERA INGENIERO EN COMPUTACIÓN
CLAVE (110) NO. DE ALUMNOS 2

CARRERA INGENIERO EN TELECOMUNICACIONES
CLAVE () NO. DE ALUMNOS _____

TEMA PROPUESTO: RECONSTRUCCIÓN DE SITIOS ARQUEOLÓGICOS: TEOTIHUACAN

OBJETIVO: DESARROLLAR UN PASEO VIRTUAL DE LA ZONA ARQUEOLÓGICA DE TEOTIHUACAN
REUTILIZANDO COMPONENTES DE SOFTWARE

INDICE DESGLOSADO:

EN COLABORACIÓN CON LOS ORGANISMOS:

NOMBRE Y GRADO ACADEMICO DEL PROFESOR: SANTIAGO IGOR VALIENTE GOMEZ
INGENIERO EN COMPUTACIÓN
NOMBRE DE LA DEPENDENCIA DONDE LABORA: DEPARTAMENTO DE COMPUTACIÓN
FACULTAD DE INGENEIRIA, U.N.A.M.

DOMICILIO OFICINA
CALLE: IRAPUATO 23 **COLONIA:** CONDESA
DELEGACIÓN O MUNICIPIO: CUAUHTEMOC **C.P.:** 06140

DOMICILIO PARTICULAR
CALLE: IRAPUATO 23 **COLONIA:** CONDESA
DELEGACIÓN O MUNICIPIO: CUAUHTEMOC **C.P.:** 06140

TELEFONO(S): 55 15 30 51

HORARIO DE ASESORIA:	DOMICILIO	OFICINA	EXT.
MARTES Y JUEVES			7:00 A 9:00
	DIA(S)	HORARIO	
CIUDAD UNIVERSITARIA A	30	DE	AGOSTO DE 2002

FIRMA DEL PROFESOR
R.F.C. VAGS-701005RV3

APROBADO POR EL COMITÉ DICTAMINADOR DE SEMINARIOS
EL _____ DE _____ DE _____

JEFE DEL DEPARTAMENTO **COORDINADOR DE LA CARRERA**

SE SOLICITA ANEXAR INFORMACIÓN COMPLMENTARIA
RECUERDE UD. QUE LOS TEMAS TIENEN 2 AÑOS DE VIGENCIA A PARTIR DE LA FECHA DE APROBACIÓN
ESTA INFORMACIÓN DEBE ENTREGARSE EN ORIGINAL Y 2 COPIAS

INFORMACIÓN ADICIONAL QUE DEBE INCLUIRSE CON CARÁCTER OBLIGATORIO EN LA PROPUESTA DE SEMINARIO / TESIS

▪ **DEFINICIÓN DEL PROBLEMA**

Describir claramente el problema a resolver en cuanto a contexto, alcance, conexión con otros problemas, justificación, relevancia y objetivo preciso.

▪ **METODO**

Discutir los métodos, caminos o procedimientos mediante los cuales puede resolverse el problema e indicarse cual o cuales de ellos adoptaran y por qué. Una vez seleccionando el método a seguir, éste debe describirse detalladamente.

▪ **INVENTARIO DE MATERIAS / TEMAS DE LA CARRERA**

Inventario se materias/temas de la carrera que se utilizarán para el desarrollo del seminario/tesis.

▪ **RESULTADOS ESPERADOS**

S E M I N A R I O / T E S I S

RECONSTRUCCION DE SITIOS ARQUEOLOGICOS: TEOTIHUACAN

DEFINICIÓN DEL PROBLEMA

El problema que se plantea es el aprovechamiento de las herramientas tecnológicas de cómputo para ser aplicadas en aspectos didácticos y educativos; para poder contribuir en áreas como la histórica, arqueológica y a la solución de otros problemas como son de aspecto turístico, reconstrucción arqueológica y de tipo histórico.

Lo relevante del proyecto es la aplicación de la ingeniería en computación en otras áreas del conocimiento para lograr un impacto cultural en la sociedad.

El objetivo es aplicar los conocimientos adquiridos durante el transcurso de la carrera y poder con juntarlos en una aplicación específica.

Utilizaremos planos arqueológicos para la obtención de las medidas reales de cada uno de los elementos, así como la ubicación correcta de los edificios que conforman la zona arqueológica de Teotihuacan, con esta información realizaremos un modelo virtual de la zona arqueológica, utilizando herramientas de graficación, tales como el 3D Studio para la creación de objetos tridimensionales, Rhinoceros para el modelado gráfico, y una "game engine" o "máquina de juegos" para realizar el paseo virtual por la zona arqueológica; junto con los conocimientos adquiridos durante la carrera podremos obtener un paseo virtual en dicha zona arqueológica.

MÉTODO

Método de desarrollo clásico o en cascada, que consta de las fases de análisis, diseño abstracto, diseño concreto, implementación y pruebas. En dichas etapas o fases se deben obtener especificaciones a partir de las cuales se orienta el trabajo de la(s) etapa(s) posterior(es) hasta lograr el producto deseado.

HERRAMIENTAS

Para la solución de nuestro problema utilizaremos las siguientes herramientas:
Para el modelado de nuestro sistema utilizaremos UML. Para la parte del modelado gráfico utilizaremos RHINOCEROS.

Y finalmente para poder realizar el paseo virtual necesitamos de una herramienta que nos permita navegar en un ambiente tridimensional, dicha herramienta será el game engine de Quake o el Fly 3D, esta elección será hecha dependiendo cual de los dos paquetes nos brinde mayores beneficios para la solución práctica y eficiente de nuestro problema.

INVENTARIO DE MATERIAS

Graficación por Computadora
Ingeniería de Programación
Análisis Grafico

Bases de Datos
Estructuras de Datos
Temas Selectos de Historia, Literatura y Sociedad

RESULTADOS ESPERADOS

Aplicar los conocimientos adquiridos en la integración de las tecnologías de graficación por computadora en una aplicación didáctica cuyo resultado final será un paseo virtual interactivo de Teotihuacan.

INDICE DESGLOSADO

- 1 Antecedentes
- 2 Descripción y problemática
- 3 Marco Teórico
- 4 Marco Tecnológico
- 5 Análisis
- 6 Diseño
- 7 Desarrollo
- 8 Pruebas
- 9 Conclusiones
- 10 Glosario
- 11 Bibliografía



AGRADECIMIENTOS



A DIOS

Porque siempre iluminó el camino que debía seguir en el andar de mi vida, así como me otorgo las fuerzas necesarias y enriqueció mi espíritu para luchar y no rendirme ante las adversidades.

A LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Por tener el privilegio y el honor de pertenecer a la máxima casa de estudios, alma mater de nuestro país, y portar con orgullo el nombre de nuestra universidad.

A LA FACULTAD DE INGENIERÍA

Por mi formación profesional y forjarme como un gran ingeniero e inculcar la responsabilidad de afrontar los problemas que aquejan a nuestra sociedad.

A MIS PROFESORES Y COMPAÑEROS

Por convivir de manera respetuosa, y compartir en las aulas sus ideas, conocimientos y experiencias, y fuera de ellas saber que cuento con su apoyo.

A JOSÉ FRANCISCO SALGADO RODRÍGUEZ

Por ser mi compañero de carrera pero más que eso, por ser un amigo y por confiar en mis capacidades para lograr nuestra gran meta como profesionistas y compartir esta experiencia en la cual aprendimos uno del otro de nuestras virtudes y nuestras deficiencias.

AL ING. SANTIAGO IGOR VALIENTE GÓMEZ

Por ser parte crucial de esta etapa de nuestra formación profesional, para poder culminar de manera exitosa.

AL DR. ENRIQUE RUÍZ VELASCO SÁNCHEZ

Por sus consejos, siendo estos de gran utilidad durante el transcurso de mi preparación como profesionista.

A MI ESPOSA

CLAUDIA IVETT SOLANO SALAS

Porque en ti tengo el complemento que necesito, para que cada día me sienta bien y por confiar en mí y hacerme notar de que capacidades y deficiencias cuento para poder enfrentarme a los retos que se me presentan. Con mucho amor, Gracias.

A MI HIJO

DIEGO AVENDAÑO SOLANO

Porque te encuentras en mi corazón y mi mente, por eso eres todo mi sentir y mi pensar, ya que eres la principal razón que motiva mis deseos de superación.

RECONSTRUCCIÓN DE SITIOS ARQUEOLÓGICOS: TEOTIHUACAN

AGRADECIMIENTOS

A MIS PADRES

**JOSÉ ANTONIO AVENDAÑO CROCKER
JULIA MARTÍNEZ VILLAGRÁN**

Porque han inculcado en mi una buena educación y se que en todo momento contaré con su amor y comprensión incondicionalmente.

A MIS HERMANOS

**JULIO CESAR AVENDAÑO MARTÍNEZ
GILBERTO AVENDAÑO MARTÍNEZ**

Porque siempre podrán contar conmigo cuando lo necesiten, y porque siempre estemos juntos.

A MIS SUEGROS

**FIDEL SOLANO ZÚÑIGA
MARÍA DEL LOURDES SALAS SANTAMARÍA**

Por su comprensión y confianza.

A MI CUÑADA Y A MI CONCUÑO

**MARÍA DEL LOURDES SOLANO SALAS
RODOLFO MARTÍNEZ ARTEAGA**

Porque en cualquier momento puedo contar con ustedes.

A MIS ABUELITAS

MINA Y ROSITA

Por su amor y cariño.

A MIS TIOS

**C.P. CARLOS GARCÍA VILLAGRÁN
LUIS REY GARCÍA VILLAGRÁN**

Por su ejemplo de lucha, dedicación y superación.

A MIS TIOS, TIAS, PRIMOS Y PRIMAS

Por sus muestras de afecto y solidaridad.

A MIS AMIGOS

**ADRIÁN, ALEJANDRO, CARLOS, CESAR, CHRISTIAN,
EDGAR, GUSTAVO, JACOBO, JESÚS ALEJANDRO, JORGE,
JOSÉ ANTONIO, JUAN PABLO, RAFAEL EINAR, RUBEN**

Por su amistad y los gratos momentos que pasamos juntos.

JOSÉ CARLOS AVENDAÑO MARTÍNEZ.

RECONSTRUCCIÓN DE SITIOS ARQUEOLÓGICOS: TEOTIHUACAN

A mis mejores amigos:

Julieta Rodríguez Quezada y J. Francisco Salgado Rico.

Por haberme guiado con amor y rectitud. Por que día a día han estado a mi lado para apoyarme. Por enseñarme que el trabajo y la fuerza de voluntad vencen cualquier obstáculo y transforman nuestro mundo.

Luis Ángel Salgado Rodríguez, por tu cariño y comprensión. Por que somos el mejor equipo.

A mis abuelitas **Ma. de Jesús Quezada y María Rico.**

En memoria de mi abuelito **José Carmen Salgado Rodríguez:**

Por su infinito amor y bondad. Por ser el vivo ejemplo de que la fe mueve montañas.

Los llevo en mi corazón.

A mi tía **Guadalupe Salgado Rico:**

Por su invaluable apoyo y confianza

A la **Universidad Nacional** y a la **Facultad de Ingeniería:**

Espíritu de un pueblo, ejemplos de grandeza y generadoras de ilusiones que transforman realidades.

Por siempre agradecido.

A mis maestros:

Quienes guiaron mi camino con su vocación y conocimiento, y me enseñaron que la grandeza del hombre radica en la grandeza de su espíritu.

A mis familiares, compañeros y amigos. A mi compañero **José Carlos** y a todos quienes hicieron posible el desarrollo de ésta tesis.

JOSÉ FRANCISCO SALGADO RODRÍGUEZ.
"Cuauhtlehuani!"



ÍNDICE



ÍNDICE GENERAL

CAPÍTULO I	INTRODUCCIÓN.....	1
CAPÍTULO II	ANTECEDENTES.....	7
CAPÍTULO III	MARCO TEÓRICO.....	21
CAPÍTULO IV	MARCO TECNOLÓGICO.....	35
CAPÍTULO V	INGENIERÍA DE SOFTWARE.....	63
CAPÍTULO VI	ANÁLISIS Y DISEÑO.....	79
CAPÍTULO VII	DESARROLLO.....	129
CAPÍTULO VIII	PRUEBAS Y MANTENIMIENTO.....	169
CAPÍTULO IX	CONCLUSIONES.....	175
CAPÍTULO X	GLOSARIO.....	181
CAPÍTULO XI	BIBLIOGRAFÍA.....	189

ÍNDICE DESGLOSADO

CAPÍTULO I	INTRODUCCIÓN.....	1
CAPÍTULO II	ANTECEDENTES.....	7
II.1	REALIDAD VIRTUAL.....	9
II.1.1	¿QUÉ ES REALIDAD VIRTUAL?.....	10
II.1.2	CRONOLOGÍA DE LA REALIDAD VIRTUAL.....	13
II.2	HISTORIA DE TEOTIHUACAN.....	17
CAPÍTULO III	MARCO TEÓRICO.....	21
III.1	CONCEPTOS DE REALIDAD VIRTUAL.....	23
III.1.1	CARACTERÍSTICAS BÁSICAS DE RV.....	23
III.1.2	COMPONENTES DE UN SISTEMA DE RV.....	25
III.1.3	RAMAS DE LA RV.....	26
III.1.4	NIVELES DE RV.....	29
III.1.5	MUNDOS VIRTUALES.....	32
III.1.6	METAS DE LA RV.....	32
III.2	CONCEPTOS DE MULTIMEDIA.....	33
III.3	DIFERENCIA ENTRE MULTIMEDIA Y REALIDAD VIRTUAL.....	34

CAPÍTULO IV	MARCO TECNOLÓGICO.....	35
IV.1	SOFTWARE.....	37
IV.1.1	TIPOS DE SOFTWARE.....	37
IV.1.2	COMPUTACIÓN GRÁFICA.....	38
IV.1.3	CONSTRUCCIÓN TRIDIMENSIONAL.....	40
IV.1.4	TRANSFORMACIONES GEOMÉTRICAS.....	42
IV.1.5	REPRESENTACIÓN TRIDIMENSIONAL.....	44
IV.1.6	TEXTURAS E ILUMINACIÓN.....	46
IV.1.7	CARACTERÍSTICAS DEL SOFTWARE DE RV.....	49
IV.1.8	ESTRUCTURA DE UN SOFTWARE DE RV.....	51
IV.2	HARDWARE.....	53
IV.2.1	DISPOSITIVOS INPUT / OUTPUT.....	54
IV.2.2	VISUALIZACIÓN INTERACTIVA.....	54
IV.2.3	DESPLAZAMIENTO TRIDIMENSIONAL.....	56
IV.2.4	RASTREO DE MOVIMIENTOS.....	57
IV.2.5	MANIPULACIÓN DE OBJETOS.....	57
IV.2.6	SONIDO VIRTUAL.....	58
IV.3	APLICACIONES.....	59
IV.3.1	ARQUITECTURA.....	59
IV.3.2	EDUCACIÓN.....	60
IV.3.3	ENTRETENIMIENTO.....	60
IV.3.4	MEDICINA.....	61
IV.3.5	INDUSTRIA.....	61
IV.3.6	CAVE.....	61
IV.3.7	OTRAS APLICACIONES.....	62
CAPÍTULO V	INGENIERÍA DE SOFTWARE.....	63
V.1	INGENIERIA DE SOFTWARE.....	65
V.1.1	¿QUÉ ES LA INGENIERÍA DE SOFTWARE?.....	65
V.1.2	PARADIGMAS.....	67
V.1.3	EL PROCESO.....	67
V.1.4	VISIÓN GENÉRICA DE LA INGENIERÍA DE SW.....	69
V.2	CICLO DE VIDA.....	71

V.2.1	FASES DEL CICLO DE VIDA.....	71
	V.2.1.1 ANÁLISIS.....	71
	V.2.1.2 DISEÑO.....	72
	V.2.1.3 CODIFICACIÓN.....	73
	V.2.1.4 PRUEBAS.....	73
	V.2.1.5 UTILIZACIÓN.....	74
	V.2.1.6 MANTENIMIENTO.....	74
	V.2.1.7 SUSTITUCIÓN.....	75
V.2.2	CICLO DE VIDA CLASICO (O EN CASCADA).....	75
CAPÍTULO VI	ANÁLISIS Y DISEÑO.....	79
VI.1	ANÁLISIS DE REQUISITOS.....	81
	VI.1.1 ANÁLISIS DE REQUISITOS DEL SISTEMA.....	81
	VI.1.2 ANÁLISIS DE REQUISITOS DEL SOFTWARE.....	83
VI.2	PLANEACIÓN.....	86
VI.3	DISEÑO DE DATOS.....	88
	VI.3.1 TABLA DE MEDIDAS.....	92
	VI.3.1.1 OBTENCIÓN DE MEDIDAS.....	92
	VI.3.1.2 OBTENCIÓN DE LA HOJA DE CÁLCULO.....	93
	VI.3.2 TABLA DE REFERENCIAS.....	100
	VI.3.3 TABLA DE TEXTURAS.....	104
	VI.3.4 DISEÑO PARA LA DETECCIÓN DE COLISIONES.....	105
	VI.3.5 DISEÑO DE LA ILUMINACIÓN.....	109
	VI.3.6 DISEÑO DEL AUDIO.....	117
	VI.3.7 DISEÑO DEL AMBIENTE.....	118
	VI.3.8 DISEÑO DE LAS TRANSICIONES.....	120
VI.4	DISEÑO ARQUITECTÓNICO.....	121
VI.5	DISEÑO PROCEDURAL.....	122
VI.6	DISEÑO DE LA INTERFASE.....	126

CAPÍTULO VII	DESARROLLO.....	129
VII.1	DESARROLLO DE LA ARQUITECTURA.....	131
VII.1.1	CREACIÓN DE OBJETOS.....	133
VII.1.1.1	CREACIÓN DE LA GEOMETRÍA.....	135
VII.1.1.2	PREPARAR LA GEOMETRÍA PARA LA APLICACIÓN DE TEXTURAS	152
VII.1.2	APLICACIÓN DE LAS TEXTURAS.....	154
VII.1.2.1	DISEÑO Y APLICACIÓN DE MATERIALES.....	154
VII.1.2.1.1	Crear el archivo de materiales.....	157
VII.1.2.1.2	Asignación del id a la geometría.....	157
VII.1.2.1.3	Asignar el archivo de materiales.....	157
VII.1.2.2	MODIFICADOR UVW MAP.....	158
VII.1.2.2.1	Asignar el modificador UVW Map.....	159
VII.1.2.2.2	Aplicar el comando Collapse (Colapsar).....	159
VII.1.3	EXPORTACIÓN.....	160
VII.1.4	CARACTERÍSTICAS DE LAS PLATAFORMAS.....	161
VII.2	DESARROLLO DE LA NAVEGACIÓN.....	162
CAPÍTULO VIII	PRUEBAS Y MANTENIMIENTO.....	169
VIII.1	PRUEBAS.....	171
VIII.1.1	PRUEBAS PARA EL MODELADO.....	171
VIII.1.2	PRUEBAS PARA LA NAVEGACIÓN.....	172
VIII.2	MANTENIMIENTO.....	173
CAPÍTULO IX	CONCLUSIONES.....	175
CAPÍTULO X	GLOSARIO.....	181
CAPÍTULO XI	BIBLIOGRAFÍA.....	189
RECONSTRUCCIÓN DE SITIOS ARQUEOLÓGICOS: TEOTIHUACAN		VI



CAPÍTULO I

INTRODUCCIÓN



“El pueblo que conoce su pasado, se identifica con sus raíces, ama y defiende lo suyo, entenderá mejor su presente, y conociendo éste, construirá su futuro con la grandeza existente en su pasado”.

**J.C.A.M.
J.F.S.R.**

“Valor y confianza ante el porvenir hallan los pueblos en la grandeza de su pasado. Mexicano, contéplate en el espejo de esa grandeza. Comprueba aquí, extranjero, la unidad del destino humano. Pasan las civilizaciones; pero en los hombres quedará siempre la gloria de que otros hombres hayan luchado para erigirlos”.

Jaime Torres Bodet.

A lo largo de las últimas dos décadas la Realidad Virtual ha cobrado un auge cada vez mayor, gracias a las capacidades potenciales que esta tecnología brinda a la investigación, la ciencia, la ingeniería, la medicina, la educación, la arquitectura, las artes y el entretenimiento, por tan sólo mencionar algunas.

Formar a las generaciones actuales y futuras para la vida en una sociedad inmersa en la tecnología, es un reto que debe ser asumido con responsabilidad. Este reto se manifiesta en la necesidad de una educación sólida, con valores éticos y morales, con un sentido nacionalista, y a la vez flexible y dinámica, para adaptarse a las condiciones cambiantes del desarrollo tecnológico.

En este contexto, los enfoques educativos tradicionales, basados en el uso de libros de texto en el aula han llegado a ser cuestionados. En vez de memorizar hechos, con la realidad virtual se le está dando más énfasis a las habilidades necesarias para construir y aplicar el conocimiento.

Los estudiantes tienen que aprender a localizar, interpretar y combinar creativamente la información y aislar, definir y resolver problemas. La tecnología de la RV¹ ha sido ampliamente señalada como un desarrollo tecnológico importante que puede apoyar al proceso de enseñanza - aprendizaje de varias formas.

Algunas de sus excepcionales capacidades son la posibilidad de permitir a los estudiantes la visualización de conceptos abstractos, observar eventos a escalas atómicas o planetarias, visitar ambientes e interactuar con eventos que la distancia, el tiempo o los factores de seguridad los hacen completamente inalcanzables en condiciones reales.

¹ RV. Abreviatura de Realidad Virtual.

Las actividades educativas que pueden ser soportadas por estas capacidades de la RV conducen a la opinión actual de que los estudiantes podrán alcanzar un mejor dominio, retención y generalización de los nuevos conocimientos en la medida en que se involucren activamente en la construcción de ese conocimiento en situaciones de aprendizaje activo (aprender haciendo).

En los países desarrollados se han establecido varios programas para introducir una gran cantidad de estudiantes y educadores en esta tecnología. Asimismo, un número creciente de instituciones académicas han desarrollado programas para la investigación de las aristas fundamentales de esta tecnología, y hasta algunas escuelas de nivel primario y medio evalúan las ventajas de su introducción.

Pero las condiciones de introducción de esta tecnología no son un patrimonio exclusivo de los países desarrollados. En México con la apertura del observatorio y auditorio de realidad virtual Ixtli (palabra náhuatl que se refiere a la visión, único en Latinoamérica) ubicado en la Dirección General de Servicios de Cómputo Académico de la Universidad Nacional Autónoma de México, se han abierto las puertas para la investigación y el desarrollo de la RV.

Los usuarios podrán sumergirse en mundos simulados como si fueran realidad, desde el interior de las moléculas esenciales para la vida hasta la reconstrucción de los sitios arqueológicos, por citar solamente algunos. Es decir, se podrá incursionar en todas las áreas del conocimiento.

Es necesario que las futuras generaciones de profesionales mexicanos se eduquen combinando las técnicas actuales de educación con la RV, motivo por el cual este trabajo resulta de mucha valía, ya que de manera didáctica e interactiva muestra parte de la historia y arqueología de Teotihuacan.

La relevancia de nuestro trabajo radica en que el Paseo Virtual puede ser extensivo a cualquier área del conocimiento, reforzando así los conocimientos adquiridos por el alumno, lo que conlleva a un mejor aprendizaje.

En muchas ocasiones la distancia y el costo que implica un viaje hacia cualquier zona arqueológica, impiden a los estudiantes conocer dicha zona y la historia de su cultura. El PV² solucionaría este problema, y ayudaría a reforzar el aprendizaje de los alumnos, realizando primero una visita virtual y posteriormente visitando la zona arqueológica ya con pleno conocimiento del lugar que visitarán.

Este trabajo de tesis demuestra que se puede desarrollar de manera sencilla una aplicación de RV, en este caso un PV en la zona arqueológica de Teotihuacan, cuyas principales características son la facilidad de uso, es decir, amigable y la interacción ilimitada con el mundo virtual.

² PV. Abreviatura de Paseo Virtual.

Con el uso de nuestro PV los estudiantes podrán alcanzar un mejor dominio y retención de los nuevos conocimientos adquiridos, y lograremos que se interesen por lo que están aprendiendo, en este caso la historia prehispánica de México.

Finalmente agradecemos, a las instituciones y personas que ayudaron a la elaboración de nuestra tesis y cuya ayuda fue indispensable para el desarrollo de la misma:

Dra. Linda Manzanilla Naím, Investigadora del Instituto de Investigaciones Antropológicas UNAM.

Dr. Jaime Litbak King, Director del Periódico de Humanidades UNAM.

Dr. Manuel González Casanova, Jefe del Departamento de Multimedia, Divulgación de la Ciencia UNAM.

Dra. Geneviève Lucet L., Directora de Cómputo para la Investigación, Dirección General de Servicios de Cómputo Académico, UNAM.

Instituto Nacional de Antropología e Historia (INAH).

Centro de Estudios Teotihuacanos INAH.

Museo Nacional de Antropología e Historia INAH.

Museo de Sitio de Teotihuacan INAH.

Museo de la Pintura Teotihuacana INAH.

Museo Regional de Querétaro.



CAPÍTULO II

ANTECEDENTES



II.1 REALIDAD VIRTUAL

“Realidad Virtual, cada una de las palabras que componen este término resultan ser conflictivas a la hora de buscarles una definición única y exacta. No cabe duda que la realidad resulta ser altamente relativa para las diferentes entidades que la viven, y que el término virtual (que tiene existencia aparente y no real), puede abarcar demasiado”.

Iván Santelices Malfanti.

El avance, en los últimos tiempos, de la informática ha propiciado la creación y masificación de un nuevo término, “Realidad Virtual”. Iniciada en los programas de entrenamientos militares, simuladores de vuelo, centros de investigación y académicos, ha pasado a los programas más variados en el ámbito profesional y doméstico. Hoy se empieza a aplicar en ingeniería, medicina, arquitectura, educación, juegos, entre otros campos. Cuál será el futuro de la RV, es aún una interrogante con bastantes aristas, dado lo impredecible que es acotar una disciplina que depende de los avances que se puedan obtener tanto en software como hardware, así como de nuevas aplicaciones aún inexploradas.

La RV, evidentemente se encuentra en su estado inicial. Las posibilidades, en todos los campos, son halagadoras, como también lo son sus posibles consecuencias, según los usos para los que pueda destinarse. Por primera vez, vamos a tener la posibilidad de actuar simultáneamente sobre los sentidos, creando un mundo en el que, a diferencia del real, es una fantasía en la que sus participantes poseen una libertad absoluta, y cuyas acciones cualesquiera que sean no afectan a terceros.

El concepto agrupa dos ideas aparentemente opuestas: “realidad” y “virtualidad”. Es “real” aquello que “tiene existencia verdadera y efectiva”, menciona el Diccionario de la Real Academia de Lengua Española. El término “virtual”, viene definido como aquello “que tiene existencia aparente pero no real”. Lo virtual se utiliza “frecuentemente en oposición a lo efectivo o real”. Nótese el carácter paradójico que adquiere el concepto “RV”, al quedar su significado como “realidad no real”.

La imagen clásica invocada por el término RV es la de una persona interactuando con una computadora a través de un casco y un guante sensitivo. Es, en su definición más simple, es una manera de amplificar o sustituir uno o más sentidos. La “realidad” percibida por dicho sentido puede mantener o no alguna correlación con la vida o el mundo real.

Cuando la percepción de los sentidos es amplificada se pueden sobrepasar algunas de las limitaciones del cuerpo humano. Se puede ver más rápido o más claro, escuchar sonidos con frecuencias que se encuentran fuera del intervalo audible, así como observar objetos de dimensiones microscópicas o astronómicas, que se pueden encontrar a miles o millones de kilómetros de distancia.

Desde este punto de vista, la RV puede mejorar nuestras experiencias acerca del mundo real. Para algunos incapacitados, la RV simplemente significa ganar la facultad de percibir objetos y fenómenos del mundo real, lo que para la mayoría de las personas es un hecho común. Mientras la RV puede ayudar a interactuar con el mundo real, también puede alimentar la percepción sensorial con datos generados por una simulación en computadora, transportando al participante desde su sitio hasta lugares históricos, míticos, viajes interplanetarios, o dentro del cuerpo humano, por citar solo algunos casos.

II.1.1 ¿QUÉ ES REALIDAD VIRTUAL?

Definir Realidad Virtual es difícil. Existen posiblemente tantas definiciones como investigadores haya, pues su reciente y rápida evolución no ha permitido establecer una definición clara. Por otra parte, el término RV ha sido aplicado a cualquier desarrollo tecnológico que vaya desde juegos de computadora hasta películas tridimensionales, por lo que mucha gente no conoce lo que es verdaderamente la RV.

De este modo, no resulta extraño que la RV resulte ser relativa para diferentes personas y en diferentes situaciones. Más que definirla veremos un conjunto de definiciones, donde cada una entrega una idea de lo que se debería entender por RV:

“Es una simulación de un ambiente tridimensional interactivo generado por computadora, en el que el usuario se siente introducido en un mundo artificial, donde es capaz de ver y manipular los contenidos de ese ambiente así como relacionarse con representaciones pasadas, presentes o ficticias, y que lo percibe como real basado en estímulos a los órganos sensoriales”.

“Sistema que permite a uno o más usuarios ver, moverse y reaccionar en un mundo simulado por computadora. Los distintos dispositivos de interfaz permiten al usuario ver, tocar y hasta manipular objetos virtuales. Los mundos virtuales y todo lo que contienen se representan con modelos matemáticos y programas de computadora”.

“Es la reproducción de forma artificial de un ambiente tridimensional como en el que cotidianamente nos desenvolvemos, pero con una diferencia sustancial: ese ambiente artificial no existe en un sentido físico. Es sólo la reproducción por medio de una computadora y con el uso de distintos dispositivos llamados efectores, que engañan nuestra percepción y nos sumergen en ese ambiente artificial”.

Los elementos clave de esta definición son los siguientes: la RV es una simulación generada por computadora; es tridimensional y finalmente es interactiva.

Algunas de las características que tiene la RV son las siguientes: es en tiempo real, interactiva e inmersiva. Además trata de utilizar la mayor cantidad de sentidos para crear la sensación de inmersión.

- **Tiempo Real.** Significa que la computadora hace todos los cálculos necesarios y presenta una imagen nueva en un periodo de tiempo muy pequeño, esto hace que los requerimientos computacionales sean grandes.
- **Interactiva.** Nos da a entender que el mundo virtual y sus objetos o personajes reaccionan a las acciones del usuario y de ellos mismos y este a su vez reacciona a ellos.
- **Inmersión.** Es el proceso por medio del cual el usuario cree que esta formando parte de la experiencia virtual, es lo que crea la sensación de "estar ahí".

El objetivo de una interfaz de RV es conseguir "la inmersión completa de los canales sensoriales humanos en una experiencia vital generada por computadora".

La tecnología permite crear un entorno y participar en un guión a nuestra elección. Uno se puede ver inmerso de la forma que desee. También se puede asignar a personas u objetos virtuales atributos lógicos como el peso, el aspecto físico, la gravedad y la movilidad.

La realimentación electrónica basada en estas cualidades refuerza la experiencia llegando a convencer de que algo está ocurriendo realmente. Los entornos o escenarios de la RV pueden ser predefinidos o enfocados de tal manera que el usuario obtenga una destreza específica o una percepción clara, como si estuviese realmente en ellos.

Las simulaciones de RV difieren de otras simulaciones de computadora en la medida en que requieren dispositivos de interfaz especiales. Estos dispositivos transmiten al usuario las imágenes, el sonido y las sensaciones de los mundos simulados. En lugar de utilizar un teclado o un ratón para comunicarse con la computadora, estos dispositivos especiales permiten al participante moverse, actuar y comunicarse con la computadora de forma parecida a como lo hace en su vida cotidiana. Este estilo natural de comunicación y la capacidad de mirar a su alrededor dan al usuario la sensación de estar inmerso en el mundo simulado.

De esta forma, se puede estar en entornos virtuales que simulen o que sean réplica exacta de lugares existentes o simplemente en lugares imaginarios, permitiendo la interacción con los elementos allí ubicados. En un entorno virtual se puede aprender a manipular equipos sofisticados, a conducir, realizar operaciones, conocer una construcción antes de que se realice y muchas cosas más. La RV nos lleva a lugares cotidianos desde una nueva perspectiva, a sitios reales de difícil o de imposible acceso, e incluso a mundos imaginarios de leyes arbitrarias. Podríamos recorrer Marte o hacer un viaje entre las arterias humanas o simplemente estar entre la estructura molecular de algún elemento químico.

En otras palabras, RV, también llamada ambiente virtual, es una simulación tridimensional en computadora que proporciona información sensorial (visión, sonido y/u otros), con el propósito de hacer que el participante sienta que está en un "cierto lugar".

Se puede experimentar un ambiente virtual usando una computadora personal típica y unos pocos dispositivos de hardware especializados: una tarjeta gráfica 3D, una tarjeta de sonido, un display montado en un casco, un guante sensitivo, etc. También se necesita el soporte de software diseñado especialmente para manipular los datos del ambiente virtual. Desde el punto de vista técnico, la RV es un modo de visualizar, manipular e interactuar con computadoras y datos extremadamente complejos.

Con el propósito de alcanzar una sensación de realidad creíble, las computadoras deben ser capaces de calcular y visualizar la información sensorial lo suficientemente rápido para engañar a los sentidos del participante. Hasta no hace mucho tiempo, tanto el hardware como el software adecuado para esta tarea eran tan caros que sólo los gobiernos, unas pocas universidades y las grandes corporaciones podían destinar fondos para invertir en esta tecnología. Sin embargo, en la actualidad el precio de las computadoras se ha visto reducido, y los periféricos específicos se han visto simplificados hasta el punto de que muchas personas tienen la posibilidad de explorar el mundo de la RV en sus sistemas personales.

Se reconocen dos variantes de interacción con los ambientes virtuales: el modo inmersivo, cuando el participante “se sumerge” completamente en la simulación haciendo uso de los dispositivos de hardware especializados, y el modo no inmersivo (llamado también RV de computadora de escritorio), cuando el participante explora los ambientes haciendo uso de los dispositivos de hardware comunes hoy en día: monitor, mouse, tarjeta de sonido, bocinas, etc.

Para una inmersión total se requieren tres condiciones indispensables asociadas al equipamiento:

1. Display de campo visual total.
2. Seguimiento de la posición, movimientos y acciones de los participantes
3. Actualización inmediata del display con la retroalimentación de los movimientos y las acciones, de los participantes y objetos.

La RV de computadora de escritorio no requiere las cuatro condiciones inmersivas mencionadas anteriormente. Esta modalidad es muy popular porque puede ser explotada a un costo adicional muy bajo respecto al costo de una computadora personal con capacidades gráficas y multimedia. De manera que mientras se abarata el equipo para inmersiones totales y por consiguiente su disponibilidad, se pueden ir introduciendo alternativamente los sistemas de RV.

Existen también diversos términos para referirse al mismo concepto, tal es el caso de “Realidad Sintética”, “Mundos Virtuales o Ficticios”, palabras diferentes, mismos significados. Recientemente se ha ido acordando en los círculos científicos el término “Ambientes Virtuales” (Virtual Environments) y particularmente “Presencia”, que se define como la experiencia subjetiva de estar en un lugar o ambiente, cuando se está físicamente situado en otro.

Muchos confunden la RV con su pariente la multimedia, lo que no es sorprendente debido a que ambas tecnologías alcanzaron su madurez casi al mismo tiempo; ambas involucran imágenes y sonido, y ambas han merecido cantidades comparables de atención en los medios masivos de comunicación. Sin embargo, las diferencias entre ambas tecnologías son importantes. Además también a la RV se le confunde con otras tecnologías que utilizan gráficas por computadora. Generalmente se le confunde con la Animación, la Visualización y la Graficación.

II.1.2 CRONOLOGÍA DE LA REALIDAD VIRTUAL

Por regla general se reconoce que el término “RV” fue acuñado originalmente por el norteamericano Jaron Lanier en un artículo titulado “The Ultimate Display”. Este ex director de la empresa VPL Research, utilizó el término en 1989, para distinguir entre simuladores y la experiencia de inmersión de RV. Sin embargo, otros autores difieren de esta posición, planteando que ya en 1987, David Zelzer utilizaba dicha expresión.

Aunque los dispositivos tradicionales de entrada y salida de datos de la computadora (teclado, mouse, monitor, etc.) ofrecen grandes facilidades para el manejo de información, están limitados cuando las aplicaciones van ganando en complejidad y sofisticación, como es el caso de la RV que requiere de otros dispositivos para poder aprovechar todo su potencial.

- 1958** Philco Corporation desarrolló un dispositivo visual en un casco controlado por los movimientos de la cabeza del usuario.
Algunos años después, Iván Sutherland y otros investigadores crean un accesorio denominado “Head Mounted Display” (HMD), montando tubos de rayos catódicos en un armazón de alambres, mediante el cual el usuario podía examinar, moviendo la cabeza, un ambiente gráfico.
Luego Morton Heiling inventa y opera el Sensorama, una instalación que estimula una diversidad de sentidos (vista, olfato, sonido y fuerza motriz).
- 1965** Surge el concepto de RV, cuando Ivan Sutherland (hoy miembro de Sun Microsystems Laboratories) publicó un artículo titulado "The Ultimate Display", en el cual describía el concepto básico de la RV. El trabajo inicial del Dr. Sutherland fue básico para investigaciones subsecuentes en este campo.
- 1966** Sutherland creó el primer casco visor de RV al montar tubos de rayos catódicos en un armazón de alambre. Este instrumento fue llamado "Espada de Damocles", ya que requería de un sistema de apoyo que pendía del techo.

- 1968** Ivan Sutherland y David Evans crean el primer generador de escenarios con imágenes tridimensionales, datos almacenados y aceleradores. En este año se funda también la sociedad Evans & Sutherland.
- 1969** Myron Krueger crea ambientes interactivos que permiten la participación del cuerpo entero, en eventos apoyados por computadoras. Paralelamente, Frederick Brooks, de la Universidad de Carolina del Norte, trabajaba en un proyecto para que los usuarios interactuaran con objetos gráficos mediante un manipulador mecánico.
- 1971** Redifon en el Reino Unido fabrica simuladores de vuelo con pantallas gráficas. Henri Gouraud presenta su tesis de doctorado "Despliegue por computadora de Superficies Curvas".
- 1972** General Electric, bajo comisión de la Armada norteamericana, desarrolla un simulador computarizado de vuelo. Los simuladores de vuelo serán un importante renglón de desarrollo para la RV.
- 1976** Tom de Fanti fue el inventor del guante de datos, aunque su diseño fue mejorado posteriormente por Tom Zimmerman, dando origen al DataGlove, diseñado originalmente para poder tocar una guitarra virtual o imaginaria.
P. J. Kilpatrick publica su tesis de doctorado "El uso de la Cinemática en un Sistema Interactivo Gráfico".
- 1977** Dan Sandin y Richard Sayre inventan un guante sensitivo a la flexión.
- 1979** Eric Howlett (LEEP Systems, Inc.) diseñan la Perspectiva Óptica Mejorada de Extensión Larga (Large Expanse Enhanced Perspective Optics, LEEP).
A fines de los 70, en el Media Lab del Instituto Tecnológico de Massachusetts (MIT), se obtiene el "Mapa Filmado" de Aspen, EE.UU, una simulación de video a través de la ciudad, donde el usuario puede recorrer sus calles y edificios.
- 1980** A principios de los 80's la RV es reconocida como una tecnología viable.
Jaron Lanier es uno de los primeros generadores de aparatos de interfaz sensorial, acuñó la expresión "Realidad Artificial", también colabora en el desarrollo de aparatos de interfase VR, como guantes y visores.
- 1981** Tom Furness desarrolló la "Cabina Virtual", y G.J. Grimes, asignado a Bell Telephone Laboratories, patentó un guante para introducir datos, basándose en sensores ópticos, de modo que la refracción interna puede ser correlacionada con la flexión y extensión de un dedo.

- 1982** Ocurre uno de los acontecimientos históricos en el desarrollo de los simuladores de vuelo, cuando Thomas Furness presentó el simulador más avanzado que existe, contenido en su totalidad en un casco y creado para la Fuerza Aérea de la Armada de los E.E.U.U.
- 1984** William Gibson publica su novela de ciencia ficción, Neuromancer en el que se utiliza por primera vez el término "Ciberespacio" refiriéndose a un mundo alternativo al de las computadoras; con lo que algunos aficionados empiezan a utilizarlo para referirse a la RV. Aunque fue Jaron Lanier quien comenzó a creer y a divulgar las posibilidades de las nuevas tecnologías y se hizo famoso al presentar una versión mejorada del guante de datos para la NASA.
Mike Mc Greevy y Jim Humphries desarrollaron el sistema VIVED (Representación de un Ambiente Virtual, Virtual Visual Environment Display) para los futuros astronautas en la NASA.
Michael McGreevy y sus colegas de la NASA desarrollan Lentes de Datos, con los cuales el usuario puede mirar el interior de un mundo gráfico.
- 1985** Jaron Lanier funda la institución VPL Research. Los investigadores del laboratorio Ames de la NASA construyen el primer sistema práctico de visores estereoscopios. Mike Mc Greevy y Jim Humphries construyen un HMD con un LCD monocromo del tamaño de una televisión de bolsillo.
- 1986** En el centro de investigaciones de Schlumberger, en Palo Alto, California, Michael Deering y Howard Davidson trabajaron en estrecha relación con Sun Microsystems para desarrollar el primer visor de color basado en una estación de trabajo de Sun. Existen ya laboratorios como el de la NASA, Universidad de Tokio, Boeing, Sun Microsystems, Intel, IBM y Fujitsu dedicados al desarrollo de la tecnología de RV.
- 1987** La NASA utilizando productos comerciales, perfecciona la primera realidad sintetizada por computadora mediante la combinación de imágenes estereo, sonido 3D, guantes, etc.
Tom Zimmerman desarrolla un guante interactivo.
- 1988** M. Deering y H. Davidson se incorporan a la planta de científicos de Sun, el Dr. H. Deering diseñó características de RV dentro del sistema de gráficos GT de la empresa, mientras que el Dr. H. Davidson trabajaba en la producción de visores de bajo costo.
- 1989** VPL, y después Autodesk, hacen demostraciones de sus completos sistemas RV. El de VPL es muy caro, mientras que el de Autodesk no lo es tanto.
Jaron Lanier, creó el término "RV".
Eric Howlett construyó el Sistema I de HMD de vídeo LEEP.

VPL Research, Inc. comenzó a vender los lentes con audífonos que usaban despliegues ópticos LCD y LEEP.

Autodesk, Inc. Hizo una demostración de su PC basada en un sistema CAD de RV, Ciberespacio, en SIGGRAPH.

Robert Stone y Jim Hennequin inventaron el guante Teletact I.

- 1990** A los inicios de los 90, los sistemas de RV salen de los ambientes de laboratorios en busca de aplicaciones comerciales, apareciendo cabinas de entrenamiento para pilotos de guerra, simuladores de vuelo, recorridos virtuales, etc.
Surge la primera compañía comercial de SW RV, Sense8, fundada por Pat Gelband. Ofrece las primeras herramientas de SW para RV, portables a los sistemas SUN.
J. R. Hennequin y R. Stone, asignados por ARRL, patentaron un guante de retroalimentación tangible.
- 1991** Richard Holmes, asignado por Industrias W, patentó un guante de retroalimentación tangible.
- 1992** SUN hace la primera demostración de su Portal Visual, el ambiente RV de mayor resolución hasta el momento.
Zimmerman, asignado por VPL Research, patentó un guante con sensores ópticos.
Division hace una demostración de un sistema de RV multiusuario.
Thomas De Fanti et al. Hizo una demostración del sistema CAVE en SIGGRAPH.
- 1993** SGI anunció un motor de RV.
- 1994** La Sociedad de RV fue fundada.
IBM y Virtuality anunciaron el sistema V-Space.
Virtuality anunció su sistema serie 2000.
Division hizo una demostración de un sistema integrado de RV multiplataforma en IITSEC, Orlando.

En la actualidad, estamos aún en presencia del crecimiento y consolidación de las técnicas y recursos de la RV, el cual ha sido posible gracias al esfuerzo e interés combinado de científicos, y al dinero de las empresas que ven en ella una nueva y prometedora tecnología.

II.2. HISTORIA DE TEOTIHUACAN

Desde los inicios de Teotihuacan entre 1-150 d.C., cuando ya podemos hablar de una presencia propiamente Teotihuacana, se planeó la ciudad con dos ejes o avenidas que tenían como centro la Pirámide del Sol. Hacia el 200 d.C el centro se trasladó un poco mas al sur, a la Ciudadela, en donde el centro recayó en este conjunto, y el templo de Quetzalcoatl pasó a ser el edificio principal orientado, al igual que la Pirámide del Sol, hacia el poniente. De esta manera, la ciudad adquiriría una fisonomía especial, pues estas calzadas se orientaban hacia los cuatro rumbos del universo teniendo a la Ciudadela como centro.

La parte ceremonial ocupó un espacio amplio y central de la ciudad, que se extendía desde la Ciudadela hasta la Plaza de la Luna, es decir, que llegó a tener hasta 2 kilómetros de largo, aproximadamente, por medio kilómetro de ancho. Alrededor de este espacio se levantaron áreas habitacionales tanto de la élite como de los artesanos y de otros miembros de la comunidad.

El hecho de que acudieran a ella los señores de las provincias para ser coronados ha hecho pensar que la palabra Teotihuacan significa “lugar donde se hacen los señores”; pero como allí nació el Quinto Sol y fue creada nuevamente la Luna, bien puede significar “Lugar donde se hacen los dioses” o simplemente, “Lugar de los dioses”.

En los inicios de la ciudad, la arquitectura teotihuacana estuvo caracterizada por conjuntos de tres templos. Poco después hizo su aparición el sistema de talud y tablero, es decir, un muro inclinado (talud) y sobre él un elemento vertical con una moldura envolvente y en medio de ésta un remetimiento (tablero).

En la construcción de un edificio ceremonial se hacia el núcleo de adobes con lodo y piedra, que luego se revestía con los elementos mencionados. Para ello se careaba la piedra, y sobre lo que llamamos 'obra negra', se colocaba estuco. Sobre una capa final muy fina de estuco se pintaban los motivos decorativos con pigmentos de origen mineral.

La edificación de templos y palacios contó con un buen número de especialistas, desde arquitectos que dirigían hasta albañiles y peones que aportaban la mano de obra. Escultores y pintores daban el toque final al monumento, y asesorados seguramente por sacerdotes elegían los motivos que decorarían el edificio, según la deidad a que estuviera dedicado.

El uso de madera fue muy abundante tanto para la construcción (vigas, techos, dinteles, muros) como para la elaboración de los hornos donde se quemaba la piedra para la obtención del estuco. Miles de hombres debieron participar en las grandes obras públicas, como calzadas, templos y palacios, bajo el mando de los especialistas.

Este patrón urbano estuvo vigente en sociedades posteriores a la teotihuacana. De hecho, la ciudad azteca de Tenochtitlán, construida muchos siglos más tarde, se inspiró en esta traza.

Cuando los aztecas llegaron al Valle de México, Teotihuacan se encontraba destruida y abandonada; sin embargo, los edificios, cubiertos de tierra y plantas, además de restos de objetos esparcidos por el área, indicaban que ahí se había desplegado una gran ciudad. Estos nuevos grupos desconocían quién había construido Teotihuacan; ante esto, le otorgaron un carácter mítico atribuyéndola a los dioses. La importancia de Teotihuacan fue tal que ahí dieron origen al mito del Quinto Sol, según el cual los dioses crearon al hombre. Cuando los aztecas fundaron la Tenochtitlán no hicieron sino imitar la obra de los dioses que, según creían, habían construido Teotihuacan.

La Pirámide de la Luna tiene 150m de oriente a poniente, 130 de norte a sur; es una estructura que alcanza casi los 40 m de alto, y cuya fachada principal se orienta al sur. Destaca el llamado Edificio Adosado, que consta de varios cuerpos con talud y tablero con una gran escalinata que conduce a la parte alta. La Pirámide de la Luna, al igual que la del Sol, está formada por cuatro enormes cuerpos en talud. A ambos lados del Edificio Adosado y junto a la Pirámide de la Luna bajan dos canales recubiertos de estuco que servían para encauzar el agua.

La construcción de la Pirámide de la Luna y la plaza debió iniciarse en el mismo periodo que la Pirámide del Sol, es decir, entre 1 y 150 d.C. Varias etapas constructivas se superpusieron, y tanto los edificios de la plaza como la misma pirámide tuvieron presencia hasta el momento del abandono de la ciudad alrededor del 700 de nuestra era.

Avanzando hacia el sur por el Micaotli se encuentra a la izquierda la gran Pirámide del Sol, que mide en su base 260m por lado; como todo edificio principal, la pirámide se orienta al poniente y se compone de cuatro cuerpos superpuestos que debieron tener en su parte alta un adoratorio del que hoy no quedan restos. Si contamos los cuerpos veremos que son cinco. El cuarto, guarda una forma muy peculiar, lo que ocurre es que al ser excavado el edificio entre 1905-1907 por don Leopoldo Batres, éste construyó ese cuerpo extraño. Es decir, que la imagen que hoy tenemos de la pirámide no corresponde a su forma original.

Por el lado poniente se adosó a la fachada de la pirámide un basamento que mide 38m de ancho por 17 de largo: consta de tres cuerpos con talud y tablero y su orientación es distinta pues marca con mayor exactitud la puesta del sol el día de su paso por el cenit de la ciudad.

La Pirámide del Sol se encuentra construida sobre una cueva subterránea en forma de trébol de cuatro hojas, unida a la orilla de la calzada de los muertos por un pasaje subterráneo cuya entrada está al pie del basamento adosado. Parece que esta estancia subterránea se hizo para representar cuatro cuevas o bocas de la madre tierra, que simbolizan el origen del hombre. Por último, diremos algo acerca de la dedicación de este monumento. Todos los datos; cueva, restos óseos infantiles, etcétera, indican que estuvo dedicada a una deidad acuática, quizá Tlaloc.

Mas al sur y adosado al lado oriente del Micaotli, se encuentra otro centro ceremonial que sin fundamento ha sido llamado la Ciudadela. Consta de un cuadrángulo de 400m por lado, limitado por anchas plataformas en sus cuatro lados, siendo mas baja la que se encuentra hacia la calzada, por la cual se penetra al recinto mediante una escalinata colocada entre dos de los cuatro basamentos piramidales que coronan esta plataforma. Otros cuatro basamentos están colocados sobre cada una de las plataformas laterales y tres en la del fondo.

La pirámide de Quetzalcoatl cuenta con siete cuerpos escalonados, compuestos de taludes de poca altura que sostienen tableros enmarcados que contienen el cuerpo ondulante de una serpiente en alto relieve tallada en piedra, como todo el monumento, cubierta de plumas de quetzal, gran jeroglífico fonético del nombre de Quetzalcoatl en náhuatl. Coatl significa serpiente, pero también gemelo, y las plumas de quetzal dan el adjetivo de preciosos o divino, y así Quetzalcoatl significa “el gemelo divino” referido al astro que lo representa en el cielo: el planeta Venus, que es un gemelo por ser la estrella de la mañana y estrella de la tarde.

Las serpientes de cada tablero muestran, en escultura dos cabezas, una de serpiente y otra del dios Tlaloc, que viene a ser la serpiente de nubes o Mixcoatl deidad de la lluvia. Este monumento estuvo completamente policromado y muestra en el frente una escalinata que tiene en sus alfardas otras cabezas de serpiente brotando de un rosetón.

La sociedad teotihuacana estaba dividida en varios estratos sociales. Esto lo corroboramos en las representaciones cerámicas de personajes bien ataviados, quizá sacerdotes o personajes de alto rango, y gente del pueblo sin mayores adornos. La pintura mural también muestra esta división. El hallazgo de barrios de artesanos como en La Ventilla y otros conjuntos más ricos en acabados arquitectónicos, señalan claramente esta división social; las prácticas mortuorias, igualmente nos remiten a esto.

La forma de gobierno en Teotihuacan no se ha establecido con claridad; se piensa que tenía un gobierno colectivo, aunque la práctica mesoamericana es la del gobierno de un solo individuo o, en ocasiones, presencias duales. Lo que sí queda claro es que debió existir un gobierno central que repercutió en lugares como Copán, en Honduras, Kaminaljuyú, en Guatemala, y en otras regiones de Mesoamérica.

Gracias a la representación pictórica, pero también a los análisis de los especialistas, se ha podido saber que la dieta teotihuacana tenía como base plantas como el maíz, calabaza, frijol, chile, tomate, y verdolaga. También aprovechaban frutos como el tejocote, capulín, tuna, y quizá zapote blanco y ciruelo. El aguacate y el cacao pudieron ser traídos de otras regiones. Algunos animales se consumían, como era el caso de distintas especies de aves y otras como el armadillo, venado, conejo, etcétera. También hay representaciones de peces y serpientes, que bien pudieron formar parte de la dieta teotihuacana.

El clásico en Teotihuacan se dio antes que en otras regiones mesoamericanas. En la zona maya por ejemplo, tuvo lugar un poco más tarde, hacia el 250 d.C. No así en Oaxaca, en sitios como Monte Albán, que florecieron al mismo tiempo, de tal manera que podemos ver barrios zapotecas asentados en Teotihuacan, además de una influencia en cerámicas similares. Al parecer, también hubo presencia teotihuacana en la Costa del Golfo y en el Occidente de México. Hacia el 700 d.C. Teotihuacan desaparece; sin embargo muchas ciudades mayas lograron sobrevivir hasta el 950 de nuestra era.

Después del año 750 existe una población apenas de 5 mil habitantes, terminando así la importancia que Teotihuacan tuvo a lo largo de ocho siglos. La arqueología ha mostrado que para este momento hay evidencias de que la ciudad fue incendiada y saqueada, con el consiguiente abandono. ¿A qué se debió esto?. Algunos investigadores piensan que hubo un levantamiento armado contra las clases dirigentes, otros piensan que fueron grupos provenientes del norte quienes destruyeron la ciudad.

Creemos que lo que pasó en Teotihuacan es lo mismo que ocurrió en muchas sociedades mesoamericanas: llegan a su máxima expansión conquistando pueblos a los que someten militarmente y aplican el tributo correspondiente. Luego, en un momento de debilidad de la metrópoli esos grupos se levantan y acaban con quien los tiene sometidos. Esto ocurrió con Azcapotzalco y lo mismo pasó con los Aztecas. No nos extrañaría que en Teotihuacan se presentara una situación similar.

PERIODO	AÑOS	POBLACION/MILES	EXTENSIÓN/km ²
<i>Patlachique</i>	100 a.C - 1 d.C.	5	8
<i>Tzacualli</i>	1 - 150 d.C.	30	17
<i>Miccaotli</i>	150 - 250	45	22.5
<i>Tlamimilopa</i>	250 - 450	65	22
<i>Xolalpan</i>	450 - 650	85	20.5
<i>Metepc</i>	650 - 750	70	20

ILUSTRACIÓN II.1
Periodos de la Cultura Teotihuacana



CAPÍTULO III

MARCO TEÓRICO



III.1 CONCEPTOS DE REALIDAD VIRTUAL

III.1.1 CARACTERÍSTICAS BÁSICAS DE RV

Para distinguir un sistema de RV como tal, es necesario que cumpla con ciertas condiciones, y se toman como características básicas de un sistema de RV las siguientes:

- PUNTO DE VISTA

Hay dos cosas que definen el punto de vista. La primera es la ubicación espacial, es decir, sus coordenadas en el espacio; y el segundo, la orientación, es decir, la dirección hacia la cual se está viendo. El punto de vista se define, en la forma en que percibimos una escena; al cambiar sus coordenadas espaciales y su dirección, cambia la percepción de los objetos ubicados en la escena.

- NAVEGACIÓN

La navegación se puede definir como la capacidad de controlar el punto de vista. El hecho de poder hacer este cambio de punto de vista se le conoce como navegación. Esto implica la capacidad de la computadora para representar la escena y sus objetos de acuerdo a los cambios de punto de vista que se vayan presentando.

- MANIPULACIÓN

Se puede establecer una comunicación mucho más eficiente cuando podemos manipular los objetos a nuestro alrededor. Para la manipulación de objetos en ambientes de RV existe también una amplia variedad de dispositivos. La manipulación se puede dividir en tres categorías:

- Selección. Es la capacidad de elegir cualquier objeto del ambiente de RV.
- Interacción. Es la capacidad de actuar sobre un objeto, y ejecutar acciones sobre él.
- Comandos. Proporciona una gama de acciones para interactuar con cualquier objeto.

- INMERSIÓN

Esta palabra implica bloquear toda distracción y enfocarse solo en la información u operación sobre la cual se trabaja. Posee la habilidad para enfocar la atención del usuario, estimulando su sistema natural de aprendizaje.

La inmersión es más bien lograr un estado mental adecuado en el cual se logra focalizar casi completamente la atención del usuario en el ambiente de RV.

- INTERACCIÓN

La interacción es tener control del sistema creado. De no existir esta interacción el sistema sería una película o recorrido fijado a priori. Para lograr la interacción existen diversas técnicas e interfaces hombre-máquina, que van desde teclado y mouse hasta guantes o trajes sensoriales.

La interactividad con el mundo virtual supone que el usuario pueda mover objetos y modificarlos, y que tales acciones produzcan cambios en ese mundo artificial.

Existen dos aspectos únicos de interacción en un mundo virtual. El primero de ellos es la navegación, que es la habilidad del usuario para moverse independientemente alrededor del mundo. Las restricciones para la navegación son especificadas por el programador, lo que permite varios grados de libertad; es decir, si se puede volar o no, caminar, nadar, etc.

Otro punto importante de la navegación es el posicionamiento del punto de vista del usuario. El usuario se puede mirar a sí mismo, u observar desde varios puntos de vista.

El otro aspecto de la interacción es la dinámica del ambiente, que no es más que las reglas de cómo los componentes del mundo virtual interactúan con el usuario para intercambiar información.

- PERCEPCIÓN

Es el factor más importante. Algunos sistemas de RV se dirigirán principalmente a los sentidos (visual, auditivo, táctil) por medio de elementos externos (Cascos de Visualización, Guantes de Datos, Cabinas, etc.) de este modo, el usuario puede creer que realmente está viviendo situaciones artificiales que el sistema computacional genera, alcanzando una sensación de “inmersión” en un ambiente digital.

Además, en un sistema de RV las imágenes mostradas al usuario no se encuentran almacenadas en ningún sitio, son generadas dependiendo de la posición actual. Esto para tener total libertad de movimiento del usuario, lo que hace imposible tener guardadas todas las imágenes correspondientes a todos los posibles puntos de vista. Los sistemas de RV poseen una base de datos con todos los elementos que componen el mundo virtual, a partir de la cual, generan la información que se mostrará al usuario. También debe existir una dimensión de profundidad (tridimensionalidad). Para conseguir este efecto, los objetos del mundo virtual deben tener una tercera dimensión que indique su profundidad en la pantalla.

- SIMULACIÓN

Capacidad de representar un sistema, ésta debe replicar aspectos suficientes de la realidad para convencer al usuario que constituyen una situación paralela. Es decir, una simulación del modelo o del mundo a experimentar, donde regirá una serie de reglas, no todas necesariamente iguales a las de la vida real.

- TRIDIMENSIONALIDAD

Esta es una característica básica para cualquier sistema de RV, tiene que ver directamente con la manipulación de los sentidos del usuario, principalmente la visión, para dar forma al espacio virtual; los componentes del mundo virtual se muestran al usuario en las tres dimensiones del mundo real, en el sentido del espacio que ocupan.

III.1.2 COMPONENTES DE UN SISTEMA DE RV

Todo sistema de RV puede componerse en las siguientes 4 partes:

- EFECTORES

Son interfaces que dan acceso a un ambiente virtual. Estas incluyen salidas para dispositivos que estimulan los sentidos de presencia en el mundo virtual, semejantes a los HMD³ o receptores estéreo, y entradas de dispositivos que registran cuando se está viendo, o la dirección que se está indicando, como los head-trackers y guantes controladores, respectivamente.

- SIMULADOR DE REALIDAD

Es el corazón de un sistema RV, el simulador de realidad incluye el sistema de computadora y el hardware externo, tales como gráficas y un dispositivo de sonido sintetizado que suplen a los efectores con la información sensorial necesaria.

- GEOMETRÍA

Es la información que describe los atributos físicos de los objetos (forma, color, etc.)

- SONIDO

Después de la visión el sonido es lo más importante en el canal sensorial para las experiencias virtuales. Este tiene la ventaja de ser un canal de comunicación que puede ser procesado paralelamente con la información visual, sin ser interrumpida.

El sonido es no obstante otra poderosa herramienta de expresión para un constructor de mundo. Este tiene la habilidad de traer muchos detalles dentro del ambiente virtual.

Sin embargo, en otras áreas; los sonidos pueden contribuir con experiencias visuales. Estudios han demostrado que el simple perfeccionamiento de la calidad del sonido puede influir en la percepción de la calidad de la imagen visual.

- Sonidos de Ambiente. El diseñador del mundo virtual tiene flexibilidad casi ilimitada en lo que tiene que ver con el sonido. Cualquier tipo de sonido puede ser grabado o capturado: Voces, instrumentos musicales y efectos especiales.
- Reconocimiento de Voz. Sistemas de RV pueden efectivamente usar técnicas de reconocimiento de voz basada en una conversación independiente y comandos discretos. Cada palabra puede controlar una acción. La parte fuerte de estos es que dan libertad en las manos para el control de la simulación. La desventaja es que este no es muy efectivo en la navegación o dirigiendo movimientos.

³ **HMD.** Head Mounted Display. Interfase de los sistemas de RV que genera la inmersión del usuario. Consiste en un casco con pantallas integradas.

III.1.3 RAMAS DE LA RV

La RV ha derivado en algunas ramas de estudio, entre las que encontramos la Realidad Inmersiva o Intrusiva, la Realidad Proyectada (Tanque de Peces o No Intrusiva), la Realidad Ampliada o Superpuesta y la Telepresencia.

Se puede considerar otra rama llamada Desktop Virtual Reality (RV de Escritorio) o Window on World (Ventana al Mundo). Esta se puede considerar dentro de la RV Inmersiva, se refiere al hecho de que la pantalla de la computadora actúa como una ventana a través de la cual observamos el mundo virtual. El punto de vista que nos proporciona esa ventana puede ser cambiado, por lo tanto nos permite navegar.

Tomando en cuenta las características básicas, los tipos de RV existentes, son:

- INMERSIVA O INTRUSIVA

Lo que se pretende en esta rama es lograr la inmersión de los usuarios mediante el uso de efectores que bloqueen totalmente la percepción que estos tengan del mundo real para concentrarlos en la percepción del ambiente virtual.

En general es el tipo de RV más cara, dependiendo del número de sentidos a los cuales se pretenda estimular, pero también es la que encuentra más usos, sobre todo en los aspectos de diseño y paseos virtuales.

Ejemplos de este tipo de sistemas son los modernos simuladores de vuelo de la Fuerza Area de los Estados Unidos, simuladores de combate para soldados de infantería del Departamento de Defensa de ese país y algunos sistemas de la NASA ha ido desarrollando desde mediados de la década de los 80.

- DE ESCRITORIO O SISTEMAS DESKTOP

En lugar de utilizar cascos para mostrar la información visual se utiliza un monitor grande de computadora o un sistema de proyección.

Engloban aquellas aplicaciones que muestran una imagen 2D o 3D en una pantalla de computadora en lugar de proyectarla a un HMD. Puesto que representan mundos de tres dimensiones los exploradores pueden viajar en cualquier dirección dentro de estos mundos, los ejemplos característicos de estos ambientes son los simuladores de vuelo para computadora y la mayoría de los juegos de alto nivel de realismo. En resumen, los sistemas desktop RV muestran mundos tridimensionales a través de pantallas de 2D. Algunos comprenderán interfaces sofisticadas, como guantes, controles, cabinas personalizadas, pero todos tendrán en común la característica antes mencionada (3D en 2D).

- PROYECTADA (DE TANQUE DE PECES O NO INTRUSIVA)

En este tipo de RV una imagen en movimiento del usuario es proyectada junto con otras imágenes en una extensa pantalla donde el usuario puede verse a sí mismo, como si estuviese en la escena, en la cual es proyectada la imagen misma del participante pero sumada su imagen de video (chroma-keyed) con otra imagen utilizada como fondo o ambiente, entonces el participante visualiza en la pantalla el mundo virtual completo. Mediante un software que realiza detección de contornos es posible realizar manipulaciones dentro de la escena, las cuales son visualizadas en la pantalla.

Los usuarios pueden pintar diseños de colores en el aire, o hacer cualquier movimiento que el sistema reacciona en tiempo real. Un ejemplo actual de este tipo de RV son los escenarios virtuales que se utilizan en ciertos programas de televisión.

La Realidad Proyectada se basa en la idea de que no es necesario el uso de efectores intrusivos para lograr la inmersión del usuario. Estos sistemas consisten, por lo regular, de una habitación sobre la cual se proyectan en sus paredes imágenes del ambiente virtual (por ello también se les conoce como ambientes de Tanque de Peces). Para ello se tienen situadas cámaras que van monitoreando los gestos y acciones del usuario y una computadora se encarga de interpretarlos y realizar acciones sobre el ambiente, basadas en la interpretación de dichos gestos.

Como ejemplo, está el proyecto llamado 'The CAVE' - La Cueva - del Electronic Visualization Laboratory (EVL) de la Universidad de Illinois en Chicago.

- AMPLIADA O SUPERPUESTA

Esta se logra cuando una persona utiliza el mundo real como línea de referencia, y simultáneamente hace uso de visores de cristal transparentes u otros medios inmersivos para aumentar la realidad, superponiendo esquemas, diagramas, textos, referencias, etcétera.

Estos sistemas consisten de ambientes en los que mezclamos la percepción que tenemos del mundo real junto con objetos virtuales. Se utilizan para complementar la información que recibimos de nuestro mundo con el análisis que la computadora nos proporciona en tiempo real.

Como ejemplo la Boeing está explorando la posibilidad utilizar este sistema en la ingeniería de los aeroplanos, de tal manera que sus técnicos e ingenieros no tengan que irse a ver un manual para resolver un problema, pues el sistema de realidad aumentada les mostraría los diagramas esquemáticos o las listas de las partes del aeroplano, sin que el operario tenga que moverse de su silla. Estos ambientes se están desarrollando principalmente para asistentes médicos, en donde el médico ve la información sobre el paciente o el órgano que esté revisando u operando, sobrepuesta en el punto de vista exacto que tiene el médico.

- TELEPRESENCIA

Término creado por Marvin Minsky que significa presencia remota, es un medio que proporciona a la persona la sensación de estar físicamente en otro lugar por medio de una escena creada por computadora. Esta rama de la RV es la que más interés presenta actualmente a la gente dedica a la robótica, la manufactura o actividades de cierto riesgo a la integridad personal de los usuarios.

La idea es utilizar ambientes de RV donde los estímulos proporcionados por los efectores no son generados por una computadora, sino que se generan y transmiten desde un dispositivo robótico a distancia. Así pues, en el dispositivo robótico se puede montar una cámara de video estereoscópica que proporciona al usuario una perspectiva visual parecida a la que él tendría al estar en el lugar del robot.

Se pueden poner algunos exoesqueletos que simulen las resistencias que el robot encuentre al moverse o manipular objetos. Se puede tener un brazo robótico que sea operado por el usuario mediante un guante o joystick de la misma forma que si el brazo robot fuera el suyo propio.

- CABINA DE SIMULACIÓN

El ejemplo más común de este tipo de simulador es la cabina para el entrenamiento de aviadores. Generalmente la cabina recrea el interior del dispositivo o máquina que se desea simular (un carro, un avión, un tanque etcétera), las ventanas de la misma se reemplazan por pantallas de computadoras de alta resolución, además existen bocinas estereofónicas que brindan el sonido ambiental y puede estar colocada fija o sobre ejes móviles. El programa está diseñado para responder en tiempo real a los estímulos que el usuario le envía por medio de los controles dentro de las cabinas.

- VENTANAS ACOPLADAS VISUALMENTE

Esta es la clase de sistema de inmersión que se asocia más a menudo con RV. Este sistema se basa en colocar las muestras directamente en frente del usuario, y conectando los movimientos de la cabeza con la imagen mostrada.

Para lograr mayor acople la inmersión se logra con un casco HMD estereofónico, que posee sensores de posición y orientación que informan a la máquina la posición del usuario en todo momento, además de indicarle hacia donde está mirando. Un ejemplo de este tipo de RV son los juegos que hicieron popular esta técnica de computación en los últimos años

III.1.4 NIVELES DE RV

“Para tener una escala de niveles de RV, primero hay que saber a cuáles y a cuántos sentidos estamos estimulando. En una enumeración tradicional tendríamos a los 5 sentidos:

- | | |
|---------|----------|
| 1 Vista | 4 Olfato |
| 2 Oído | 5 Gusto |
| 3 Tacto | |

En una revisión de estos sentidos, en el Diaspar Virtual Reality Network, se enumeraron los sentidos como sigue:

- | | |
|-----------|----------------|
| 1 Vista | 5 Olfato |
| 2 Oído | 6 Gusto |
| 3 Tacto | 7 Feromonal |
| 4 Balance | 8 Inmunológico |

Esta lista la hicieron basados en pruebas científicas que hacen creer que existen órganos y redes neuronales dedicadas a los puntos de Balance, Feromonal e Inmunológico.

Basados en esta lista de sentidos a estimular, el Diaspar Virtual Reality Network creó la siguiente lista para medir niveles de la RV:

- | | |
|---------------------------------|----------------------------------|
| - V0 Realidad | - V5 Cuatro sentidos estimulados |
| - V1 Ocho sentidos estimulados | - V6 Tres sentidos estimulados |
| - V2 Siete sentidos estimulados | - V7 Dos sentidos estimulados |
| - V3 Seis sentidos estimulados | - V8 Un sentido estimulado |
| - V4 Cinco sentidos estimulados | - V9 Ningún sentido estimulado |

El uso de una computadora en modo texto significa tener un sistema de RV de nivel V9, puesto que no se estimula a ningún sentido, pues el texto aunque se capta por la vista es sólo la codificación simbólica de la información.

Al nivel tecnológico actual, podemos calificar a la mayoría de los sistemas existentes dentro de un nivel V6, es decir, se está estimulando a por los menos tres sentidos que comúnmente se consideran a la vista, al oído y tacto.

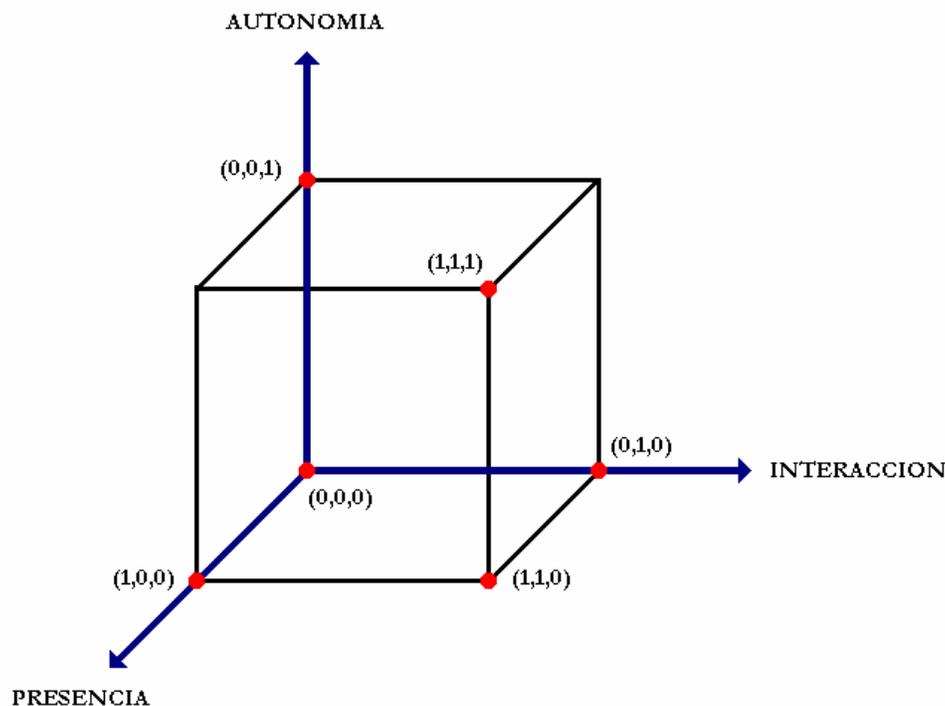
Existe otra taxonomía para la RV propuesta por Zeltzer, asumiendo que cualquier sistema de RV tiene tres componentes:

1. Un conjunto de modelos / objetos o procesos
2. Medios para modificar los estados de estos modelos o procesos
3. Un rango de modalidades sensoriales que permiten al participante experimentar el ambiente virtual

Se representan cada una de estas componentes dentro de un cubo unitario con vectores paralelos a sus ejes cartesianos y que se identifican con autonomía, interacción y presencia.

- **Autonomía.** Es la medida cualitativa de la cantidad de elementos que tienen un comportamiento propio y su capacidad de reaccionar a estímulos; si no hay reacción, la mediada es cero y en caso de una autonomía total, la media es igual a uno.
- **Interacción.** Es el grado de acceso a los parámetros o variables de un objeto. Cero representa un control de las variables que no depende en tiempo real, mientras que uno se asigna a variables que se pueden manipular durante la ejecución en tiempo real.
- **Presencia.** Es la medida de la fidelidad de la entrada / salida sensorial que proporciona el sistema. Esto es altamente dependiente en las tareas del sistema.

La RV está situada en las coordenadas (1,1,1) - autonomía, interacción, presencia -; los antiguos sistemas de graficación están en el (0,0,0), pues que son sistemas que corrían no en tiempo real, sin iteración y en procesamiento por lotes. El punto (0,1,0) - interacción - se puede alcanzar en la actualidad sin dificultad mientras que el (0,1,1) - interacción, presencia - serían los sistemas de RV más sofisticados de hoy en día.”⁴



ILUSTRACION III.1
Representación gráfica de la taxonomía de Zeltzer

⁴ VALIENTE Gómez, Santiago Igor., Carreón Granados, Juan José. (Asesor), “Realidad Virtual: Panorámica y Aplicaciones”, Facultad de Ingeniería, UNAM, México, 1997, Pag. 3.13 – 3.16.

ESTADOS DE LA SIMULACIÓN MULTISENSORIAL

- **Pasivo.** Son entornos inmersivos no interactivos. Es un entorno digital en el cual podemos ver y oír y quizás sentir lo que sucede. El entorno puede moverse lo que da sensación de movimiento (tránsito forzado) pero no es posible controlar el movimiento. En sentido estricto se trata de una pseudo-RV. Corresponde a las llamadas películas dinámicas (o "ride films"), habituales en los parques de atracciones y en los salones recreativos.
- **Exploratorio.** Son sistemas que permiten desplazarse por un entorno virtual para explorarlo lo que supone un salto cualitativo en cuanto a funcionalidad. Es el estadio habitual de los paseos arquitectónicos y de las obras de arte virtuales.
- **Interactivo.** Un sistema virtual interactivo permite experimentar y explorar el entorno y, además, modificarlo. Un verdadero sistema de RV debe ser interactivo.

INDUCCIÓN ELECTRÓNICA DE LOS SENTIDOS

Relación entre los sentidos, la percepción y las interfaces en un sist. de simulación multisensorial		
SENTIDO	PERCEPCIÓN	INTERFACES
Vista	luz	Pantallas, sistemas de proyección y ópticas generadoras de imagen 3D, cascos visualización 3D, gafas de obturación rápida
Oído	onda sonora	Tarjeta de sonido, audio 3D, altavoces, auriculares
Tacto	percepción táctil y propioceptiva (auto percepción)	Dispositivos táctiles (guantes y trajes); sistemas de retorno de fuerzas
Olfato	química aire	Sistemas odoríferos (experimentales - poco desarrollados -)
Gusto	química solución	No hay investigación en este campo
Vestibular	equilibrio	Plataformas móviles alfombras continuas sistemas de rastreo de posición / orientación

III.1.5 MUNDOS VIRTUALES

Se puede decir que existen tres tipos básicos de mundo virtuales que pueden existir por separados como también mezclados entre sí.

MUNDO MUERTO

Es aquel en el que no hay objetos en movimiento ni partes interactivas, por lo cual sólo se permite su exploración. Suele ser el que vemos en las animaciones tradicionales, en las cuales las imágenes están precalculadas y producen una experiencia pasiva.

MUNDO REAL

Es aquel en el cual los elementos tienen sus atributos reales, de tal manera que si miramos un reloj, marca la hora. Si pulsamos las teclas de una calculadora, si visualizan las operaciones que esta realiza y así sucesivamente.

MUNDO FANTÁSTICO

Es el que nos permite realizar tareas irreales, como volar o atravesar paredes. Es el típico entorno que visualizamos en los videojuegos, pero también proporcionan situaciones interesantes para aplicaciones serias, como puede ser observar un edificio volando a su alrededor o introducirnos dentro de un volcán.

III.1.6 METAS DE LA RV

Se pueden distinguir varias metas que se proponen lograr a través de la RV y, de entre ellas, Tom Hayward dice: "Crear mejores formas de comunicarse". El mismo Tom Hayward dice: "Y el propósito de la RV es para mejorar la efectividad de todas las formas de comunicación: con las computadoras, con nosotros mismos, con otros seres".

Otra meta que identifican otras personas es la de utilizar a la RV como el vehículo adecuado para lograr la "amplificación de la inteligencia" (intelligence amplification o IA) en el ser humano. Esta meta se pretende lograr mediante la combinación de habilidades del ser humano y de la computadora en las cuales sean respectivamente mejores que el otro.

Así, de esta forma, al combinar habilidades, se puede lograr una simbiosis cuyo resultado serán seres humanos con habilidades mentales mucha más desarrolladas. La mayoría de los investigadores en el campo de la RV parecen guiarse bajo esta filosofía.

La forma en que la RV pretende alcanzar su meta es a través se simular la forma en cómo percibimos el mundo, la forma en que cómo lo manipulamos y la forma en la que somos parte del mundo.

III.2 CONCEPTOS DE MULTIMEDIA

Etimológicamente, la palabra multimedia significa “múltiples intermediarios”, y utilizada en el contexto de las tecnologías de la información, hace referencia a que existen “múltiples intermediarios entre la fuente y el destino de la información, es decir, que se utilizan diversos medios para almacenar, transmitir, mostrar o percibir la información”. Según esta definición tan general, una televisión o un periódico serían dispositivos multimedia, pero nosotros vamos a restringir este concepto al de multimedia digital: “es la integración en un sistema informático de texto, gráficos, imágenes, vídeo, animaciones, sonido y cualquier otro medio digital”.

Es difícil definir en pocas palabras el término multimedia. Se puede decir que, en una computadora personal, es la capacidad de mostrar gráfico, vídeo, sonido, texto y animaciones como forma de trabajo, e integrarlo todo en un mismo entorno llamativo para el usuario, que interactuará o no sobre el para obtener un resultado visible, audible o ambas cosas.

Para poder combinar e integrar fácilmente todos estos elementos, es preciso almacenarlos bajo una misma y única forma y por lo tanto crear dispositivos adaptados de almacenamiento, transmisión y tratamiento, tales como CD-ROM, redes de transmisión de datos (especialmente, de fibra óptica) y métodos de compresión y descompresión.

De cualquier manera, aunque Multimedia es más un concepto que una tecnología, si necesitamos de ésta para crear y ejecutar Multimedia; aunque es importante recordar que la tecnología no es su base principal. En estos términos para la creación de Multimedia existen toda una gran gama de productos, abarca software de desarrollo de sistemas y utilitarios, adaptadores que integran funciones que permiten capturar, digitalizar, reproducir o visualizar elementos de aplicaciones modularmente, como el usuario lo requiera; la familia de sistemas y soluciones Multimedia incrementa los porcentajes de retención de información y favorecen la comprensión; las presentaciones se convierten en escenarios dinámicos de información, los servicios se presentan al usuario final de una manera atractiva y contundente, se proporciona acceso a la información a toda persona que lo requiera y no se necesita ser un experto en sistemas para obtenerla.

En un futuro no tan lejano, multimedia cambiará la manera de concebir la enseñanza y el aprendizaje, además de arribar con pasos de conquistador en el mundo de los negocios, el entretenimiento y el hogar. En la educación, los beneficios muestran sus resultados en procesos educativos rápidos y efectivos. Multimedia apoya la educación al facilitar la visualización de problemas o soluciones; incrementa la productividad al simplificar la comunicación, elimina los problemas de interpretación y estimula la creatividad e imaginación al involucrar a los sentidos. En resumen Multimedia es aplicable en cualquier área, siempre y cuando la aplicación sea bien desarrollada y se cumplan con las necesidades básicas para que una aplicación Multimedia tenga sentido; es decir, exista un problema o una necesidad a solucionar y se tenga una metodología adecuada que permita un desarrollo interdisciplinario.

III.3 DIFERENCIA ENTRE MULTIMEDIA Y RV

Se discute mucho la diferencia entre RV y Multimedia. Es pertinente aclarar que RV y Multimedia son dos cosas distintas. Ambos se valen del mismo equipo, pero con fines diferentes. Ambas tecnologías se basan en la integración de distintos medios de transmisión de información para proporcionar ésta por tantos canales como sea posible pero en la RV es importante que la simulación sea en tiempo real para no provocar desfases entre lo que perciben nuestros sentidos y lo que interpreta el cerebro.

En la RV se debe de lograr un grado de inmersión mayor de lo que se pretende con la Multimedia. En la RV se debe de tener un grado de interacción muy grande, que si bien algunos sistemas de Multimedia proporcionan esa interacción, la mayoría de las veces lo que se tiene es un secuencia predeterminada de pasos a seguir para llegar a algún lado en la información que se persigue. La RV debe de tener un mayor grado de libertad para navegar los ambientes de lo que los sistemas de Multimedia permiten.

También, no es lo mismo la representación virtual de un objeto a que ese objeto y su uso representen RV. Para decir que estamos usando RV necesitamos varias cosas, entre ellas, el uso de representaciones virtuales de objetos, una amplia libertad de navegación en el ambiente y manipulación de objetos en el ambiente y, por ultimo, la sensación de inmersión, o sea, sentir estamos ahí.

Más que los sistemas de Multimedia, son muchos de los video juegos los que se acercan más a la meta de la RV, la inmersión. Se puede decir que un video juego no pide mucho en términos de hardware y efectores y no está pensado para ser un ambiente de RV pero comparte muchas características que debe de poseer un ambiente de RV.

En resumen, la multimedia se refiere a una información preelaborada y preprogramada, que se presenta a través de una interfase. Por el contrario, la RV es absolutamente dinámica y cambiante. La multimedia es básicamente bidimensional, un conjunto de imágenes planas presentadas en secuencia en una pantalla. La RV es por su propia naturaleza tridimensional. En multimedia la única “interacción” a disposición del usuario es la capacidad de seleccionar una secuencia diferente para presentar la información grabada; el usuario no puede alterar lo que está allí, y no puede adicionarle nada. Por el contrario, la RV es intrínsecamente interactiva y maleable. Aún el más primitivo sistema de RV es mucho más interactivo que la más avanzada combinación de hardware y software multimedia.



CAPÍTULO IV

MARCO TECNOLÓGICO



IV.1 SOFTWARE

El software de RV es el programa encargado de darle vida al mundo, por medio de él, podemos dar movimiento a los objetos, hacer que el mundo aparezca ante nosotros desde una perspectiva diferente cuando miramos alrededor, simular situaciones tanto cotidianas como imposibles, y permitir que una persona interactúe con los elementos que le rodean.

Un modelo de RV es realmente una gran labor en el desarrollo del software, necesita la creación de centenares de líneas de código que pueden ser descifrados por una computadora. Secciones enteras de código que representan órdenes e instrucciones deben ser refinadas, probadas y protegidas contra la falsificación. Cuando se realizan cambios o modernizaciones por un gran número de personas, el código se vuelve confuso.

IV.1.1 TIPOS DE SOFTWARE

- SOFTWARE PARA EL MODELADO TRIDIMENSIONAL

Dado que un ambiente virtual es un medio 3D, todos los objetos en un mundo virtual tienen que ser descritos de manera que puedan ser vistos desde cualquier ángulo. Una simple imagen del objeto no es suficiente. La verdadera geometría de los objetos tiene que especificarse usando un software de modelado 3D, y luego exportarse hacia el ambiente virtual.

- SOFTWARE PARA GRÁFICOS BIDIMENSIONALES

Dado que existen límites en la complejidad geométrica de los objetos en un mundo virtual, frecuentemente es útil poder “pintar” la superficie de los objetos con detalles adicionales. Este proceso recibe el nombre de “mapeo de textura” y requiere de software de gráficos bidimensionales, llamados a menudo “programas para la edición de imágenes”. Las imágenes creadas con este tipo de software (o fotografiadas, digitalizadas, y luego editadas con este software) pueden ser luego usadas para “forrar” la geometría producida en el software de modelado 3D, con interesantes efectos realistas.

- SOFTWARE PARA EDICIÓN DE SONIDO DIGITAL

El sonido es un aspecto muy importante de la RV, aunque frecuentemente se le presta poca atención. A pesar que el desarrollo alcanzado en materia de gráficos 3D es impresionante, aún no es lo suficientemente realista ni abarca todo el potencial del sistema sensorial humano. La tecnología de reproducción del sonido digital exhibe tanta o más resolución que la del propio oído humano. El software de edición de sonido digital permite cortar, insertar, pegar, mezclar y enlazar los sonidos del ambiente virtual.

- SOFTWARE PARA SIMULACIÓN

La RV exige de un software para proporcionar una experiencia inmersiva. El software tiene que ser capaz de procesar las señales provenientes de los localizadores y otros dispositivos de entrada para actualizar las visualizaciones por lo menos 20 veces cada segundo. Esto se puede complicar más por el hecho de que puede haber más de un participante en el ambiente virtual, más de un conjunto de dispositivos de entrada y más de una computadora conectada en red que esté ejecutando la simulación. El software tiene que crear y mantener actualizada una base de datos que toma cuenta de todos los objetos presentes en el mundo virtual. Como ejemplos de software de simulación de RV se pueden mencionar World Tool Kit, Division, Superscape, Cosmo y VRML.

IV.1.2 COMPUTACIÓN GRÁFICA

El aporte que ha realizado la computación gráfica a diversas actividades ha sido de una magnitud insospechada, es así que actualmente un trabajo de ingeniería o arquitectura no se concibe sin el aporte de la computación.

La información gráfica se maneja en el computador básicamente de dos maneras: como un mapa de puntos (bitmaps) o como una geometría vectorial. Las imágenes de puntos se utilizan en el procesamiento de fotografías digitalizadas o en dibujos sencillos y consisten en una trama de puntos de color (píxeles) que representan la imagen.

Se caracterizan por su facilidad de manipulación cambiando la tonalidad de los puntos, pero están limitadas en resolución y precisión, además que son planas (bidimensionales). El archivo digital se puede conservar en una diversidad de formatos, entre los que podemos mencionar: BMP, GIF, PCX, TIFF o JPG.

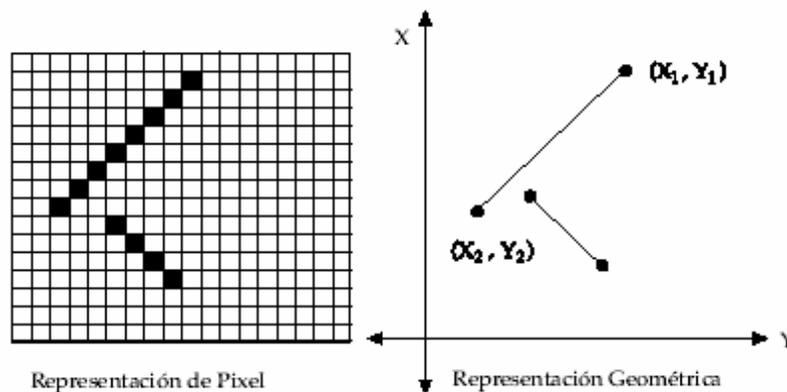


ILUSTRACIÓN IV.1
Bitmap vs Imagen vectorial

Por su parte, los archivos geométricos guardan la información precisa de las formas gráficas, a través de un sistema de coordenadas cartesianas (X, Y). De este modo, cada figura se identifica por sus puntos extremos, y pueden ser representadas a cualquier escala o calidad de imagen.

Naturalmente, el diseño de las imágenes vectoriales es más complejo, pero permite trabajar con dimensiones exactas e incluso con formas tridimensionales, basta con agregar un tercer eje de coordenadas (Z). Estos programas son denominados CAD⁵. Por ello en la actualidad es fácil encontrar diseños de construcciones, automóviles, computadores y otros, realizados con programas CAD.

La información gráfica consiste en una base de datos numérica almacenada en el computador, que define cualquier tipo de forma o volumen. Para poder visualizarlos, es necesario interpretarlos y generar una imagen en la pantalla. Aunque la geometría sea tridimensional, se genera una vista plana de la información. Ocasionalmente esta vista reúne varias caras de las formas (perspectiva) otorgando una representación de la volumetría. Esta información es coherente, en el sentido que la alteración de una parte de la geometría tridimensional, altera el conjunto, por tanto cada vista presentará la geometría modificada, asegurando la sensación de manejar un modelo completo.

Cuando los diseños están prácticamente completos, se aplican fórmulas de iluminación y presentación de superficies para producir imágenes más realistas. Estas propiedades son aplicadas con mayor o menor detalle de exactitud dependiendo de las exigencias requeridas. Otra potencialidad que se le ha adicionado a estos productos es la animación, utilizada para diversos diseños.

Otro aspecto importante dentro de la computación gráfica son los tipos de software, éstos pueden ser clasificados en dos grandes grupos: paquetes generales de programación y aplicaciones para propósitos especiales. En los primeros podemos reconocer las librerías de funciones gráficas que se utilizan con un lenguaje de programación de alto nivel, como el lenguaje C. En el segundo grupo se pueden clasificar los productos diseñados para personas que no son programadores, este es el caso de los CAD.

De este modo, la computación gráfica se encuentra en diversos campos, que ya son parte de nuestra vida cotidiana, como el cine, diseño de edificios, prendas de vestir, avisos publicitarios, edición de periódicos, muebles, vehículos y videojuegos, por citar solamente algunos.

⁵ CAD. Computer-Aided Design (Diseño Asistido por Computadora).

IV.1.3 CONSTRUCCIÓN TRIDIMENSIONAL

Los modelos tridimensionales corresponden a escenarios con clases diferentes de objetos, tales como árboles, edificios, muebles, etc. Por esto, es necesario utilizar geometrías que representen con precisión las características de los objetos en cuestión.

Los métodos como los fractales y los sistemas de partículas permiten, a su vez, representar elementos naturales como nubes, montañas y vegetales, entre otras formas.

Difícilmente habrá un solo método para describir cabalmente todos los objetos que existen en la naturaleza y que incluya todas sus características.

Básicamente se tienen dos modalidades para representar la geometría:

- a) Representaciones de frontera (B-rep: Boundary - representation), la cual describe los objetos 3D, como el conjunto de sus superficies externas.
- b) Representaciones de divisiones de espacio, en que se subdivide el ambiente en pequeñas unidades volumétricas (voxels: volume element), determinando la región espacial que ocupa cada objeto.

En los inicios de la computación gráfica las descripciones geométricas se estructuraban sólo en puntos o líneas con dirección (vectores), pero actualmente la mayoría de los programas de CAD y RV se basan en la representación por fronteras.

De este modo la descripción de una geometría tridimensional en la computadora se realiza, generalmente, debido a los polígonos externos de las formas, localizados en un espacio determinado por tres ejes cartesianos, que representan el espacio, con cierto tamaño del universo y medidas.

En estas descripciones gráficas las posiciones de cada punto se identifican por sus coordenadas. Con dos puntos extremos se puede indicar una línea tridimensional, y a su vez tres puntos definen un plano.

De este modo, un objeto tridimensional se define como un conjunto de superficies poligonales (denominadas caras), en el cual se identifican sus aristas y vértices. También se reconoce en cada polígono, una Normal, que indica la orientación de la cara, estableciendo uno de los lados como visible.

Los programas gráficos especifican una superficie poligonal mediante la serie de coordenadas de sus vértices y los parámetros de atributos asociados. Estos datos se colocan en tablas que se utilizan en el procesamiento y generación de imágenes, y en la manipulación de objetos. Estas tablas de datos de polígonos pueden ser organizadas en dos grupos, tablas geométricas y tablas de atributos.

Las primeras contienen las coordenadas de vértices y los parámetros para identificar la orientación espacial de las superficies. La segunda tabla contiene los parámetros que especifican el grado de transparencia del objeto, las características de reflexión y textura de las superficies, entre otros datos. Una organización muy utilizada para la tabla geométrica es crear 3 campos: vértices, aristas y polígonos.

Los objetos tridimensionales son usualmente poliedros de base rectangular (paralelepípedos regulares, como los cubos), también denominados “cajas”. Agrupando estas formas básicas se puede conformar una gran variedad de objetos. También se utilizan mucho volúmenes básicos como pirámides, cilindros o esferas, aunque las superficie curvas son aproximadas a través de secuencias de planos. Una mayor cantidad de polígonos permite una descripción más exacta de la curvatura, pero a la vez exige mayor procesamiento computacional.

Para las formas curvas de simetría radial (como las esferas, cilindros, conos, vasos o jarrones), se pueden generar las superficies poligonales basadas en el concepto de sólido de revolución. Consiste en generar la forma curva mediante la rotación de un perfil determinado, alrededor de un eje, denominado “eje de rotación”. Estableciendo también la cantidad de segmentos o tramos en los cuales se subdividirá la superficie curva en polígonos rectos.

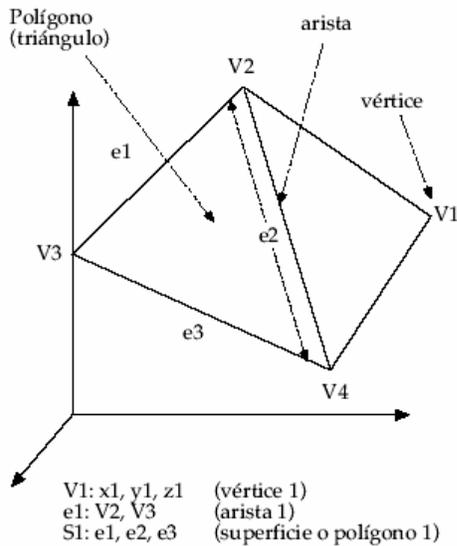


ILUSTRACIÓN IV.2
 Información geométrica de un objeto 3D

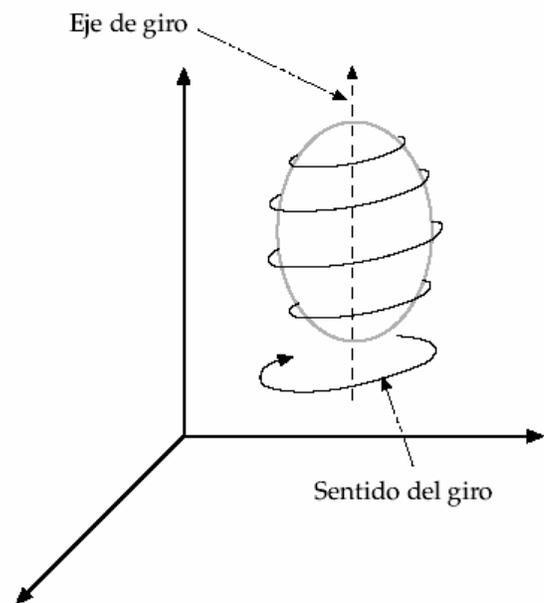


ILUSTRACIÓN IV.3
 Generación de un objeto 3D por revolución

IV1.4 TRANSFORMACIONES GEOMÉTRICAS

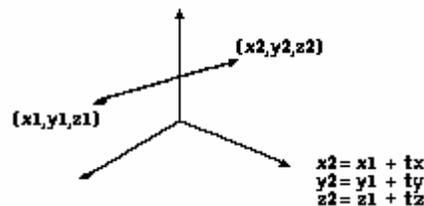
Las transformaciones geométricas son utilizadas para manipular los objetos dentro de un modelo 3D. Un objeto se puede trasladar y/o rotar en los tres ejes coordenados (x, y, z). Para el caso de las rotaciones se puede considerar como una composición de tres rotaciones, una para cada eje cartesiano, es así que cualquier secuencia de transformaciones se representa como una secuencia de transformaciones en cada eje. Las transformaciones más utilizadas son:

a) Traslación: es la traslación de un punto desde la posición $P1 = (x1, y1, z1)$ a la posición $P2 = (x2, y2, z2)$. La operación se representa mediante la matriz:

$$\begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix}$$

donde tx , ty , tz corresponden a la distancia de traslación para los ejes x, y, z respectivamente.

Dicha traslación se traduce en:



Esta forma de traslación es válida para todos los puntos del objeto en cuestión. Por lo tanto, un objeto se traslada en tres dimensiones al trasladar cada uno de los puntos de definición inicial del objeto. Para un objeto que se representa como un conjunto de superficies de polígonos, trasladamos cada vértice de cada superficie y volvemos a trazar las facetas del polígono en la nueva posición.

b) Rotación: para girar un objeto se debe designar un eje de rotación (en el cual se girará el objeto) y el valor de la rotación.

Por norma, los ángulos de rotación positiva producen giros en sentido opuesto a las manecillas del reloj con respecto al eje de una de las coordenadas. Si se gira con respecto al eje z la matriz será:

$$\begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos A & -\text{sen} A & 0 & 0 \\ \text{sen} A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix}$$

donde la rotación, con respecto al eje z, es en A grados. La matriz se puede adaptar para los giros en los otros ejes.

c) **Escalamiento:** escalar un objeto se refiere a cambiar el tamaño y volverlo a posicionar con respecto al origen de las coordenadas. Si todos los parámetros de transformación son iguales se cambian las dimensiones relativas del objeto. La matriz que representa el escalamiento es la siguiente:

$$\begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix}$$

donde S_x , S_y , S_z son los parámetros de escalamiento con respecto al origen. Existen otras transformaciones tridimensionales como la reflexión y el recorte.

Otras operaciones volumétricas bastante utilizadas en los programas tridimensionales, son las operaciones “booleanas”. Se refieren cuando se utiliza una geometría de construcción sólida (CSG: constructive solid geometry) para operaciones similares al álgebra de Boole, esto es: adición, sustracción o intersección. Aplicadas en volúmenes, la adición o unión ($A+B$) genera la fusión de los dos volúmenes en uno solo, descartando cualquier parte que se interseque entre ambos. Si no se intersectan, se crea un volumen compuesto de dos partes separadas.

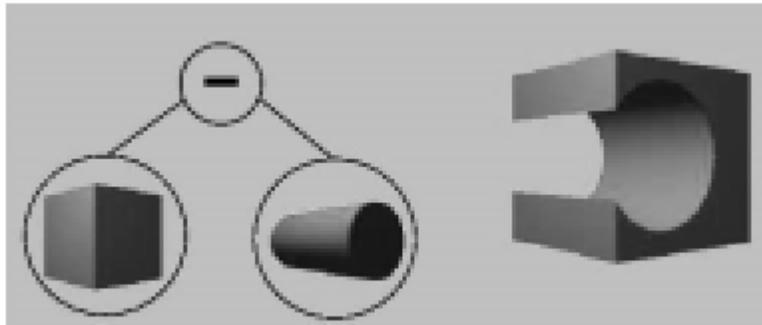


ILUSTRACIÓN IV.4
Operación booleana

La sustracción ($A - B$) genera un volumen en el que se ha eliminado la parte del otro en que se intersecta, el volumen sustraído simplemente desaparece (si no se intersectan, queda íntegro el original). En este caso es muy relevante cuál volumen es el original, generalmente reconocido como el primer operando. En la intersección ($A \cap B$), se genera el volumen común en el cual participan los dos originales, en los cuales se eliminan en sus partes restantes. En algunos programas también se agrega una operación de intersección que conserva los volúmenes restantes (generando tres formas). Estas modalidades son muy efectivas para construir fácilmente ciertas formas complejas, como por ejemplo, muros con ventanas o piezas mecánicas aunque la información computacional es estrictamente de superficies y no sólida, ocasionalmente se producen errores en la presentación gráfica.

IV.1.5 REPRESENTACIÓN TRIDIMENSIONAL

Existen varios métodos de representación tridimensional, pero inicialmente diremos que para obtener la imagen de una escena 3D, primero debemos establecer una posición (usualmente en coordenadas cartesianas) para el observador o cámara, que son los conceptos utilizados para denominar al punto de vista desde el cual se ve el modelo computacional. Frente a éste se establece un plano de visión, en el cual se proyecta la información geométrica de las formas 3D.

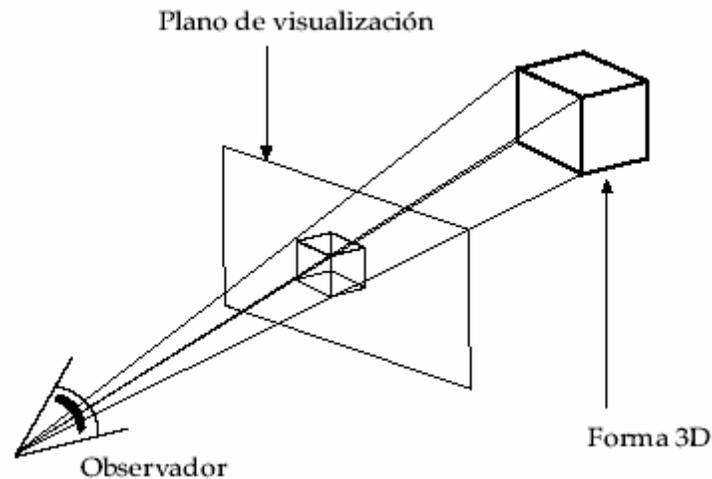


ILUSTRACIÓN IV.5
Proyección en el plano de visión

En principio, los objetos son presentados como formas transparentes (wireframe) y luego se aplican técnicas de iluminación y presentación de superficie para colorear y sombrear las áreas visibles. Así la computadora genera las imágenes a partir de la posición de la cámara con respecto a los objetos que ésta debiera estar visualizando. Dentro de las técnicas para generar la vista de un objeto tridimensional, tenemos:

a) **Proyección paralela:** este método consiste en proyectar puntos del objeto a través de líneas paralelas al eje de visión. Es así, como en diferentes posiciones de la cámara, se proyectan las superficies visibles del objeto sobre el plano de visualización para obtener diversas vistas (superiores, laterales, isométricas) que posean medidas proporcionales del objeto.

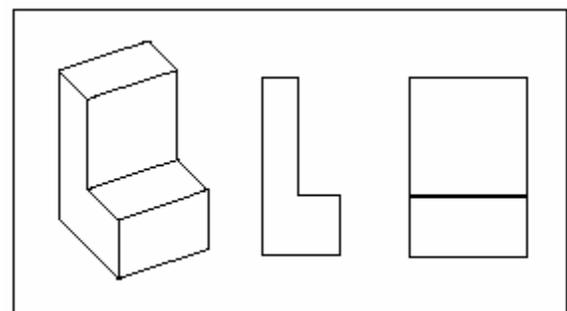


ILUSTRACIÓN IV.6
Proyecciones paralelas

b) **Proyección polar (perspectivas):** Esta técnica se basa en proyectar puntos del objeto al plano de visualización a lo largo de trayectorias convergentes al punto de vista. Esto implica que los objetos que están más lejos de este plano se despliegan más pequeños y representan cabalmente la profundidad del entorno, pero sin medidas precisas. En este tipo de proyección las imágenes de escenarios 3D obtienen mayor realismo visual. Además, las aristas de los objetos tienden a converger en una línea de fondo (horizonte).

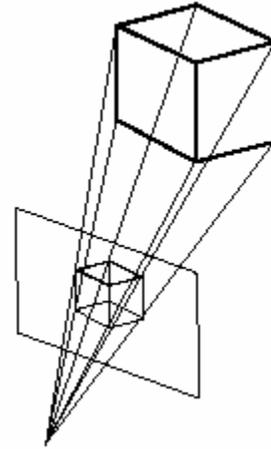


ILUSTRACIÓN IV.7
Proyección polar

c) **Eliminación de líneas ocultas:** Se utiliza como técnica para poder distinguir el frente y la parte de atrás de los objetos desplegados. Sin ocultamiento un objeto no podría ser comprendido en su posición con respecto a otros.

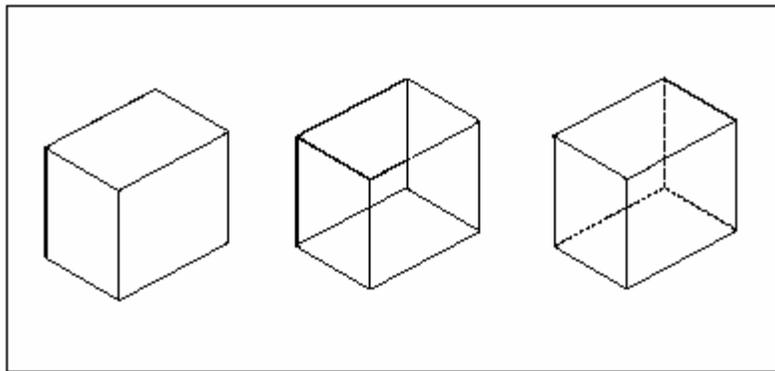


ILUSTRACIÓN IV.8
Eliminación de líneas ocultas

Otra manera de indicar el volumen es variar la intensidad de los objetos con respecto a su distancia de la posición de visualización, es decir, las líneas que están más cerca del punto de vista se despliegan con mayor intensidad, seleccionando los valores en un rango de distancia en que deben variar. Otra forma de indicar la profundidad es otorgar un efecto de atmósfera entre los objetos, es decir, los objetos más lejanos tienen una apariencia más tenue que los más cercanos, en algunos casos se agregan distorsiones de color para fortalecer la visión borrosa a la distancia.

IV.1.6 TEXTURAS E ILUMINACIÓN

Para lograr que el modelo tridimensional alcance una apariencia realista, es necesario incorporar a la geometría dos capacidades relevantes, superficies texturizadas y fórmulas de iluminación. Estas dos propiedades complementan la geometría y generan una mejor reproducción de la situación real que representa el modelo computacional.

Inicialmente, los objetos geométricos pueden ser representados con colores básicos. Los colores se asignan de acuerdo con patrones RGB (rojo-verde-azul) o HSL (tono-intensidad-brillo), en combinaciones proporcionales de cualquiera de estos tres valores. Como los colores están representados por intensidad luminosa, los valores mayores son más claros. Los tres indicadores al máximo producen el blanco, al mínimo generan, el negro como ausencia de color. Adicionalmente se incorpora un valor para la transparencia (alpha-channel).

Sin embargo, se ha reconocido que la variedad de colores, por muy amplia que sea, no alcanza a otorgar un gran realismo visual, debido a que en los ambientes reales, los colores varían de tono en las superficies. Una de las técnicas más eficaces para mejorar la apariencia realista de las superficies, ha sido la incorporación de texturas digitalizadas.

La utilización de texturas consiste en utilizar fotografías de materiales reales, que se aplican como imágenes planas en algunas superficies. La técnica de aplicación es similar a la proyección de mapas, por lo que se denomina “mapeo”. Si la forma es distinta a la imagen, ésta simplemente se recorta. Si el tamaño de la imagen es muy pequeño se puede repetir en la superficie, o se puede agrandar, aunque esto último muestra notoriamente los puntos de la imagen (pixelado).

Las texturas pueden tener un valor de rugosidad (bumpy), en que las sombras adquieren profundidad, lo que es muy adecuado para reproducir, por ejemplo, ladrillos o piedras. De hecho, es más eficaz en algunas formas pequeñas, no modelarlas tridimensionalmente sino simplemente texturizarlas con una imagen real, con el fin de ahorrar procesamiento geométrico en objetos que tienen un rol secundario en la escena. También algunas formas naturales difíciles de modelar apropiadamente (como los árboles), es preferible representarlas con una fotografía, en que se asigna un valor de transparencia al color de fondo, para generar un perfil recortado. Otra aplicación usual de las texturas es de paisaje general (background), para tener un entorno más realista, a veces ajustado a una forma cilíndrica o semiesférica, para representarlo apropiadamente al desplazarse por el escenario virtual.

El otro aspecto relevante en la apariencia visual del modelo, es la incorporación de métodos de iluminación. Cuando la luz enfrenta una superficie opaca, en parte se refleja y en parte se absorbe, esta última es llamada reflexión difusa. La mayoría de los programas otorgan una iluminación pareja en todo el ambiente, reproduciendo esta difusión indirecta. A veces esta iluminación posee un sentido determinado como el sol, tal que las superficies enfrentadas y perpendiculares a la fuente de luz están más iluminadas y las más inclinadas o posteriores, se oscurecen. De este modo se reconocen “sombras propias” en los objetos, y se distinguen las

distintas superficies, por tonalidades más o menos luminosas del mismo color, otorgando un sentido de volumen.

Otro tipo de iluminación consiste en colocar algunos focos de luz, posicionados dentro del modelo geométrico, especialmente para iluminar zonas posteriores o interiores, que la iluminación ambiental deja oscurecidas. Indicando la posición tridimensional se establece una luz “omnidireccional”, ilumina con similar intensidad en todas las direcciones. Algunas fuentes de luz pueden poseer un sentido o eje determinado basado en un punto particular (spot light). Estas luces focalizadas se aplican en un “cono”, reproduciendo la iluminación de una lámpara con pantalla, destacando algunas partes de la escena.

Esta capacidad exige un procesamiento adicional en el modelo, porque se debe calcular el trazado de la sombra de cada objeto en los demás, de acuerdo con la posición de la fuente de luz, de manera similar al “mapeo” de texturas. Algunos objetos pueden considerar transparencia, en que parte de la luz atraviese la forma, o pueden considerar reflejo, en que rebote la iluminación. El cálculo del valor de la tonalidad de la textura de cada superficie se realiza con ciertos algoritmos gráficos. Básicamente se relaciona la dirección e intensidad de la luz y/o distancia (si es un foco), con la “normal” de la superficie.

Según el modelo de Iluminación de Lambert, la iluminación cuenta con tres principales componentes, las cuales son: Ambiental, Difusa y Especular.

- COMPONENTE AMBIENTAL

Si suponemos que la **luz ambiental** afecta de la misma forma a todas las superficies desde todas direcciones, nuestra ecuación de iluminación ambiental será: **$I = I_a K_a$** .

I_a es la **intensidad de la luz ambiental**, que se supone constante para todos los objetos. La cantidad de luz ambiental reflejada por la superficie de un objeto está determinada por **K_a** , el coeficiente de **reflexión ambiental**, que varía entre 0 y 1.

El coeficiente de reflexión ambientales una propiedad del material.

Como sucede con otras propiedades, el coeficiente de reflexión ambiental es una conveniencia empírica y no corresponde directamente a ninguna propiedad física de los materiales reales.

- COMPONENTE DIFUSA

Si consideramos la iluminación de un objeto a partir de una fuente luminosa puntual, cuyos rayos emanan uniformemente en todas direcciones a partir de un solo punto. La brillantez del objeto varía de una parte a otra, dependiendo de la dirección y la distancia con respecto a la fuente luminosa. La ecuación de iluminación difusa es: **$I = I_p K_d \cos \theta$**

I_p es la intensidad de la fuente luminosa puntual; el **coeficiente de reflexión difusa k_d** del material es una constante entre 0 y 1 que varía de un material a otro. El ángulo θ debe estar entre 0° y 90° para que la luz tenga efecto directo en el punto que sombrea, Esto quiere decir que la luz emitida detrás del objeto, no iluminará las caras frontales, y viceversa.

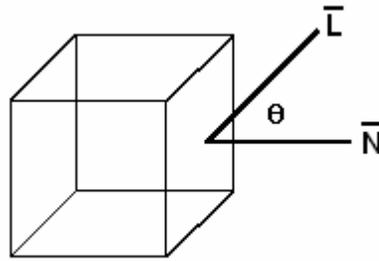


ILUSTRACIÓN IV.9
Componente Difusa

Suponiendo que los vectores \mathbf{N} y \mathbf{L} se normalizaron, podemos describir la ecuación usando el producto punto: $\mathbf{I} = I_p K_d (\mathbf{N} \cdot \mathbf{L})$

Si una fuente luminosa puntual se encuentra a distancia suficiente de los objetos que se somborean, formará en esencia el mismo ángulo con todas las superficies que comparten la misma normal. En este caso, la luz se conoce como **fuentes luminosa direccional** y \mathbf{L} es una constante para la fuente luminosa.

Los objetos iluminados de esta manera se ven con demasiado contraste entre las regiones de luz y sombra; por lo tanto, es común añadir un término de ambiente para obtener una ecuación de iluminación más realista: $\mathbf{I} = I_a K_a + I_p K_d (\mathbf{N} \cdot \mathbf{L})$

- COMPONENTE ESPECULAR

La reflexión especular se puede observar en cualquier superficie brillante. El punto de máximo brillo (highlight) es ocasionado por la reflexión especular, mientras que la luz reflejada en el resto del objeto es el resultado de la reflexión difusa.

El color del punto de máximo brillo, así como el de la luz reflejada de forma especular, es del color de la luz incidente y no el que corresponde al color del objeto.

Si movemos el punto de vista, también se moverá el punto de máximo brillo, esto sucede porque las superficies brillantes reflejan la luz en forma desigual en distintas direcciones.

La luz se refleja sólo en la dirección de reflexión \mathbf{R} , que es \mathbf{L} invertida con respecto de \mathbf{N} . De esta manera, el observador puede ver la luz reflejada especularmente en un espejo sólo si el ángulo \mathbf{a} es cero; \mathbf{a} es el ángulo entre \mathbf{R} y la dirección \mathbf{V} al punto de observación.

Phong Bui Tuong desarrolló un popular modelo de iluminación para reflectores imperfectos. El modelo supone que la máxima reflectancia especular ocurre cuando \mathbf{a} es cero y decrece rápidamente conforme aumenta \mathbf{a} . Esta caída rápida es aproximada por $\cos \mathbf{a}^n$, donde \mathbf{n} es el **exponente de reflexión especular** del material. Los valores típicos de \mathbf{n} varían entre uno y valores muy grandes, dependiendo del material de superficie que simule. Un valor de 1 proporciona una caída amplia, suave, mientras que los valores más elevados simulan un punto de máximo brillo definido y enfocado. Para un reflector perfecto, \mathbf{n} sería infinito, en el caso de que el $\cos \mathbf{a}$ sea negativo, será considerado como 0.

La ecuación que define la iluminación especular, utilizando el modelo Phong será:

$$I_s = I_a K_a + K_d I_p (\mathbf{N} \cdot \mathbf{L}) + k_s (|\mathbf{N}| \mathbf{H})^n; \quad \mathbf{H} = (\mathbf{L} + \mathbf{V}) / 2$$

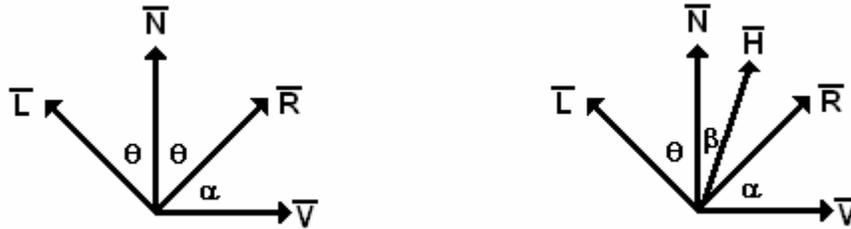


ILUSTRACIÓN IV.10
Componente Especular

El **vector intermedio H**, llamado así porque su dirección es intermedia entre las direcciones de la fuente luminosa y el observador, también se le conoce como la **dirección de reales máximos**. Si la superficie se orientara de manera que su normal estuviera en la misma dirección que **H**, el observador vería el punto máximo de brillo especular, ya que **R** y **V** también apuntarían en la misma dirección.

IV.1.7 CARACTERÍSTICAS DEL SOFTWARE DE RV

Los productos de software para el desarrollo de aplicaciones de RV se fundamentan en los conceptos de computación gráfica presentados previamente, pero se establece un conjunto de distinciones específicas y exigencias mayores que diferencian los programas de RV con los otros software gráficos e incluso con programas de modelación tridimensional o animación realista.

En el diseño tridimensional, el software de RV es muy similar a los programas CAD, pero agrega capacidades de navegación en tiempo real, interacción con el usuario, detección de colisiones, audio, programación de comportamientos, etc.

De este modo en los programas de RV encontraremos comúnmente:

1. **Importación de modelos:** Capacidad de importar formas 3D para incorporarlas en una determinada aplicación. Generalmente son provenientes de un programa CAD y se utiliza mayormente el formato DXF (drawing exchange format), aunque existen muchos otros.
2. **Librerías:** La mayoría de los programas de RV están provistos de librerías 3D, con formas básicas o primitivas tales como cajas, esferas, conos, pirámides, etc. que sirven para generar formas compuestas. También cuentan con librerías de objetos complejos, texturas, etc.

3. **Operaciones Geométricas:** Consiste en las capacidades de manipular los objetos creados en una posición definida, típicamente referida a las coordenadas cartesianas, y a partir de éstos se puede trasladar, rotar o re-escalar a otra posición.
4. **Nivel de Detalle:** Esta característica conocida en los textos como LOD (level-of-detail) permite la optimización de la visualización de una escena virtual, al cambiar una forma con un alto nivel de detalle, por otra más simple, dependiendo de la distancia del punto de vista.
5. **Animación:** Corresponde a la asignación de una traslación o rotación a un objeto en un periodo de tiempo, sincronizado con la navegación por el ambiente virtual.
6. **Articulado:** Se refiere a que los objetos puedan ser organizados en jerarquías; es decir, que partes componentes de un objeto posean propiedades de movimiento distintas a otras partes, pero supeditadas al total.
7. **Detección de colisiones:** Es una característica que permite identificar cuando un objeto intersecta a otros, de modo que pueda ser obstaculizado el movimiento del objeto, de manera similar a si éstos fueran sólidos, como en el mundo real.
8. **Propiedades físicas:** Adicionalmente se presenta una serie de atributos relacionados con características físicas, como masa o roce, reconocimiento de gravedad (movimiento vertical acelerado en proporción al tamaño o peso) e incluso de ascensión (salto sobre el objeto).
9. **Color y texturización:** Asignación de colores a las superficies y utilización de texturas digitalizadas. Incluyendo propiedades de transparencia (para generar objetos de textura, como árboles o nubes) y secuencias (como videos).
10. **Fuentes de luz:** Definición de iluminación ambiental y focos de luz con cierta posición, orientación, intensidad e incluso colores propios. Ocasionalmente con movimientos propios.
11. **Incorporación de audio:** Es la propiedad de asociarle a los objetos, un sonido que les corresponda en el mundo real. Una característica importante es controlar el volumen en relación a la distancia existente entre el objeto y el navegador.
12. **Lenguajes de programación:** Corresponde a la disposición de comandos de control que dicten comportamientos a los objetos y manejen datos de entradas y salidas.
13. **Manipulación de eventos:** Refleja la capacidad de activar un comportamiento al interactuar con un determinado objeto. Esto implica reconocer la posición y acción del usuario, interpretar una programación y modificar la geometría consecuentemente.
14. **Configuración de dispositivos múltiples:** Consiste en permitir la incorporación de distintos dispositivos de entrada y salida de datos.
15. **Conectividad en red:** Permite que el mundo virtual pueda ser utilizado en una red computacional, a través de diversos dispositivos o señales de entrada y salida, y además que permita la interacción de diversos usuarios en una misma aplicación.
16. **Mundos paralelos:** Se refiere a la generación de ambientes virtuales constituidos por submundos, en los cuales el navegante puede interactuar al momento que ingresa a cada uno de ellos. Esto con el fin de disminuir la complejidad que se tendría si fuera un solo mundo completo y por ende optimizar el procesamiento.

IV.1.8 ESTRUCTURA DE UN SOFTWARE DE RV

Un programa de RV presenta una estructura de mayor complejidad que lo normal de un software computacional, debido a que, por un lado, constituye un ambiente tridimensional que se extiende hasta capacidades multimediales, a la vez que dispone de una programación específica y el control de múltiples dispositivos externos, todo funcionando y modificándose en tiempo real.

Esto establece una amplitud sofisticada de características, que se han descrito previamente en dos procesos principales: el desarrollo de la aplicación, en el cual se utiliza el programa para diseñar el escenario 3D, incorporar el audio, programar los comportamientos y configurar dispositivos, y otra actividad es la navegación interactiva, en que se puede visualizar, recorrer y manipular el ambiente virtual de acuerdo con lo preparado en el desarrollo.

De ésta forma la estructura de un software de RV está compuesto por los siguientes aspectos:

1. **Entradas y Salidas.** Son los diferentes dispositivos de entrada y salida de datos que utilizará la aplicación virtual. Además del mouse y del monitor, usualmente los sistemas de RV utilizan joystick, rastreadores, visualizadores estereoscópicos, etc.
2. **Base de Datos.** El almacenamiento de la información de los objetos y del mundo virtual es realizado en la base de datos del mismo. Lo que se almacena en este archivo de descripción son los objetos, los programas que describen las acciones de estos objetos del usuario, métodos de iluminación, mecanismos de control y soporte del hardware. Los elementos almacenados en la base de datos son los siguientes:
 - a) Objetos
 - b) Posición y orientación
 - c) Jerarquía (de la composición del objeto)
 - d) Volumen de contorno (si el volumen está fuera de cámara, el objeto no se procesa)
 - e) Geometría de objetos
3. **Programación.** La interacción es la clave del éxito e interés de la RV. Esto requiere definir las acciones que los objetos realizarán por ellos mismos y cuando interactúe el usuario con éstos. Lo que se denomina generalmente, “guión del mundo virtual”. Algunos autores dividen estos guiones de comportamiento en 3 tipos básicos: guiones de movimiento, guiones de reacción ante un evento y guiones de conexión.
4. **Procesamientos.** Durante la utilización de la aplicación virtual se realiza un intenso procesamiento de datos en tiempo real, esto es, que se debe realizar en una mínima fracción de tiempo, mientras se navega libremente por el escenario 3D.

Básicamente es un procesamiento de la imagen visual, pero a su vez interviene un complejo cálculo geométrico, un procesamiento del audio, una interpretación de la programación y un control de los dispositivos de entrada y salida.

Primero se debe identificar la posición del punto de vista y de su orientación. Después se debe identificar la activación de algún evento, por colisión de un objeto. Luego de determinar la situación del modelo geométrico, se debe calcular la proyección en el plano de visión correspondiente, descartando las formas no cubiertas por el ángulo de visión, omitiendo las líneas ocultas, calculando las aplicaciones de texturas y los tonos de color de las superficies, de acuerdo con las fuentes de iluminación definidas y algoritmos de render. Posteriormente generar la imagen visual y activar las reacciones en dispositivos según la situación de los objetos con el usuario. También se deben procesar las frecuencias de audio que estén activadas, en función de las distancias actuales del punto de vista a las posiciones definidas.

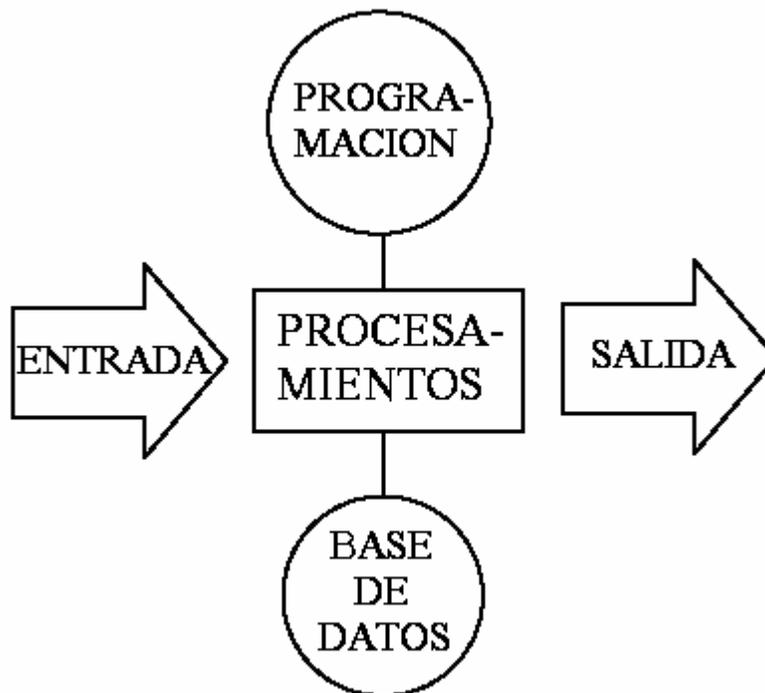


ILUSTRACIÓN IV.11
Estructura de un software RV

IV.2 HARDWARE

Para lograr la inmersión en un mundo de RV, es necesario el uso de dispositivos que simulen los estímulos sensoriales que recibimos a diario en nuestra "Realidad Real". Así, por medio de cascos o gafas estereoscópicas, podemos simular nuestra visión tridimensional, y veremos el mundo a nuestro alrededor; mediante el uso de guantes podemos coger y manipular los elementos del entorno, de una forma natural e intuitiva; y mediante el uso de otros dispositivos podemos oír y hasta sentir el entorno que nos rodea.

Los sistemas de RV se han popularizado por los cascos y guantes, que otorgan la sensación de estar “dentro” de los modelos computacionales, y permiten interactuar con éstos. Los dispositivos de RV se basan en diversas técnicas, pero en general, implican una mayor relación entre el usuario y los ambientes digitales.

El diseño digital ha permitido crear modelos con una apariencia visual muy realista. Pero el concepto de “RV” se aplica propiamente cuando se alcanza una vinculación integral con el ambiente electrónico, utilizando varios sentidos, además de la presentación gráfica. Obteniendo una sensación de “presencia” o “inmersión” en el espacio digital, para lo cual se requieren algunas técnicas y dispositivos especializados en la interacción tridimensional.



ILUSTRACIÓN IV.12
Inmersión virtual

IV.2.1 DISPOSITIVOS INPUT / OUTPUT

A través de los dispositivos de entrada y salida de datos, se facilita la comunicación hombre-máquina. Con los dispositivos de entrada (input) el usuario puede transmitir sus órdenes al sistema de RV, indicándole que desea desplazarse, cambiar el punto de vista o interactuar con algún objeto, y los dispositivos de salida (output), permiten que el usuario se sienta inmerso en el mundo virtual.

En un sistema de RV no inmersivo, los dispositivos de entrada necesarios pueden reducirse a un simple mouse, pero cuando se trata de un sistema inmersivo, las capacidades deben ser mayores. Hay que disponer de rastreadores de posición para averiguar la dirección en la que el usuario está mirando y su ubicación relativa en el mundo virtual. Simultáneamente, el sistema puede estar recibiendo información de un guante de datos y procesándola para mostrar una representación de la mano del usuario dentro del mundo virtual. Por otro lado, los dispositivos de salida permiten al usuario observar, oír y tocar el mundo creado.

Los dispositivos entrada de RV se pueden agrupar en rastreadores de posición (sensores ubicados en el cuerpo del usuario, que pueden ser magnéticos, ópticos, mecánicos o acústicos), y en elementos de control (Guantes, Trajes de Datos, Joysticks 3D, Mouse 3D, etc.). Así también, los dispositivos salida de RV se pueden dividir en generadores de imágenes (Casos Visores, Lentes estereoscópicos, etc.), de sonido u otras sensaciones y de manipulación táctil.

IV.2.2 VISUALIZACIÓN INTERACTIVA

En los modelos computacionales se pueden definir movimientos de objetos o puntos de vista, produciendo la animación de un recorrido por el ambiente digital, pero se reconoce propiamente una simulación virtual cuando se establece un control interactivo del desplazamiento. Es decir, que se pueda decidir en el momento la dirección del movimiento y recorrer el modelo libremente. Esto implica un cambio sustantivo para el usuario, que pasa de ser un “espectador” de la imagen o situación computacional, que está previamente determinada (como una película), a convertirse en un “participante” del mundo digital y modificar la experiencia. Sin embargo esto involucra también una fuerte exigencia a los equipos informáticos.

Los recorridos computacionales se generan con el mismo “truco” que se utiliza en el cine y la TV, una secuencia de imágenes que se despliegan rápidamente otorgando la ilusión de movimiento. Esta técnica aprovecha la persistencia del ojo humano, que necesita aproximadamente 1/30 de segundo para percibir completamente la imagen visual (debido a la saturación electroquímica de la retina y la memoria visual), y por lo tanto mantiene brevemente la información percibida. De este modo, los movimientos se reconocen por las diferencias de los objetos entre una imagen y la siguiente. Así, una secuencia de imágenes similares, presentadas rápidamente (30 veces por seg.), serán interpretadas por el ojo como un movimiento fluido.

Los sistemas informáticos facilitan la generación de animaciones en modelos tridimensionales, definiendo posiciones iniciales y finales, para que el computador produzca automáticamente las vistas intermedias requeridas.

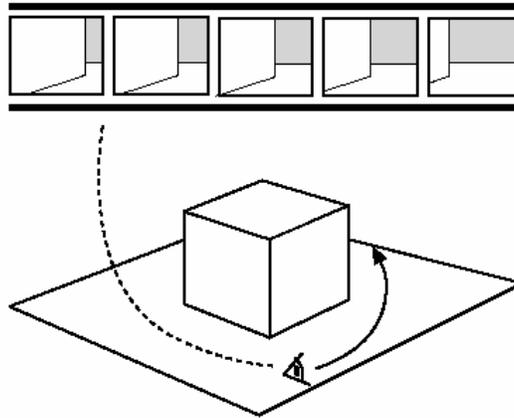


ILUSTRACIÓN IV.13
Producción de una animación 3D

La producción de la imagen computacional (“render”), involucra un cálculo geométrico de la perspectiva adecuada y de la representación, según los diversos métodos gráficos.

En un modelo de “RV” el cálculo y presentación de la imagen debe realizarse en el instante para tener un recorrido interactivo. Es decir, para que el usuario pueda desplazarse libremente, se debe estar produciendo instantáneamente la animación tridimensional. Para lo cual se dispone entonces de una mínima fracción de segundo para generar las imágenes y calcular nuevas posiciones del punto de vista. Permitiendo decidir en el momento la velocidad y dirección del recorrido computacional, tal como en un ambiente real. Esto es lo que se conoce como visualización interactiva o navegación en “tiempo real”, e implica una relación entre la capacidad del computador, la geometría del modelo y su presentación realista.

Por este motivo, en PC los modelos virtuales suelen parecer acartonados por la simplificación gráfica que se requiere para una navegación fluida. En equipos económicos actualmente se presenta la disyuntiva entre dos posibilidades, se desarrolla un modelo de alto realismo visual para imágenes y animaciones preparadas, o se tiene navegación interactiva en un modelo más simplificado.

Algunas técnicas computacionales, como QuickTimeVR de Apple, logran una representación interactiva con vistas panorámicas, a partir de fotografías de ambientes reales. Estas son montadas en un cilindro o esfera, que pueden girar como si se estuviese visualizando el entorno fotografiado. Combinando varias vistas panorámicas en una secuencia, se puede lograr una experiencia de navegación inmersiva. Sin embargo no se maneja el detalle geométrico y no se pueden programar comportamientos o configurar dispositivos, por lo cual constituye una técnica atractiva de presentación gráfica, pero sin mayores desarrollos virtuales.

IV.2.3 DESPLAZAMIENTO TRIDIMENSIONAL

Los dispositivos de salida de datos del mundo virtual (output) se complementan con sistemas de entrada de datos (input), que permiten desarrollar una actividad en el entorno digital. La acción más básica es el desplazamiento del punto de vista, estableciendo un recorrido por el modelo computacional, con movimientos en los seis grados de libertad (tres ejes y tres giros respectivos).

Para definir el desplazamiento 3D, lo más sencillo es utilizar el mouse, de que dispone la mayoría de las computadoras personales (también se puede utilizar el teclado). El mouse, aunque es un apuntador bidimensional, permite configurar movimientos adicionales con los botones. Lo más frecuente es establecer el movimiento principal del mouse para la traslación adelante-atrás y el giro derecha-izquierda (a partir del centro de la pantalla o un punto determinado), ya que éstos son los movimientos naturales del recorrido peatonal. Luego, oprimiendo algún botón (izquierdo, central o derecho) se configuran traslaciones en los ejes laterales, levantar o rotar la visión. Normalmente se omiten giros laterales porque produce un desequilibrio natural del cuerpo y genera mareos. La limitación del mouse es que debe desplazarse en un plano fijo, por lo que dificulta un movimiento libre del cuerpo y el uso de otros dispositivos.

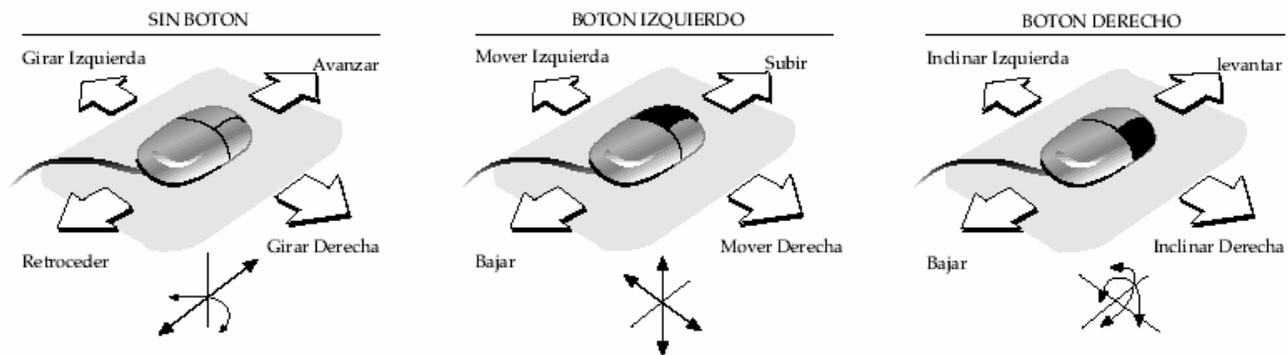


ILUSTRACIÓN IV.14
Configuración de movimientos 3D con el mouse

Los joystick se pueden configurar para el desplazamiento en el ambiente virtual de una manera similar, utilizando también los botones o disparadores incorporados. La ventaja es que el movimiento principal (adelante-atrás y derecha-izquierda), con la varilla es más intuitivo y sensible, además se puede sostener con la otra mano o adosar en un lugar fijo, liberando más la acción corporal.

Algunos dispositivos manuales, como el SpaceMouse, SpaceBall, o Joystick 3D poseen el control inmediato de los seis grados de libertad, a través de un elemento que “flota” en todos los sentidos, con gran sensibilidad, a la mínima presión. Exigen un acostumbramiento del usuario y también otorgan una libertad de movimiento que tiende a desorientar. Es recomendable restringir el desplazamiento al tipo de actividades por realizar para no distraerse excesivamente.

IV.2.4 RASTREO DE MOVIMIENTOS

Para lograr una sensación integral de inmersión en el mundo virtual es importante tener una relación directa entre el movimiento del cuerpo y la acción en el ambiente virtual, lo que se denomina “paralelismo cinético” (motion-parallax). Se utiliza para controlar la visión del entorno digital, el desplazamiento del punto de vista o incluso la localización y movimiento de distintos instrumentos o partes del cuerpo representados digitalmente. Esta capacidad es fundamental para tener un sentido de presencia en el ambiente virtual y vivir éste como una realidad paralela.

El reconocimiento de movimientos del usuario se logra con rastreadores (trackers) emplazados en algunas partes del cuerpo. Los rastreadores deben identificar en tiempo real una posición tridimensional y una orientación (giros en X,Y,Z).

Los más sencillos son giroscopios que se instalan en la parte posterior de los cascos para reconocer los giros de la cabeza y adecuar la imagen de las pantallas según la rotación natural de la vista, otorgando la sensación de estar rodeado por el modelo digital.

IV.2.5 MANIPULACIÓN DE OBJETOS

Uno de los dispositivos más publicitados en este sentido son los “guantes”, sin embargo han presentado un escaso desarrollo. Aunque la movilidad de la mano permite una gran versatilidad de acciones, la mayor parte ha sido posible representarla con controles específicos (como el mouse y los botones). Estos dispositivos utilizan básicamente dos técnicas, los “exoesqueletos” y los “guantes de datos”.

Los exoesqueletos, que se aplican también para dedos individuales, piernas o para el manejo de pequeños instrumentos manuales en medicina virtual, consisten en la instalación de una estructura mecánica paralela y sobrepuesta a la mano (adherida a distancia en distintos tramos). Con rotores en cada articulación, los sensores correspondientes y el cableado, conformando una compleja instalación normalmente fija. La ventaja de este sistema es su precisión, por lo cual se presta para aplicaciones delicadas, pero su defecto es la sofisticación y necesidad de regularla cuidadosamente con cada usuario. Algunos dispositivos utilizan guantes rígidos, con un mínimo de articulaciones y controles asociados, pero presentan un campo limitado de utilización.

El otro sistema son los “guantes de datos” (como el DataGlove), compuestos de lycra con cables de fibra de vidrio por cada dedo. Cada fibra posee un emisor de luz al inicio y un sensor al final, de modo que se determinan los giros por la intensidad de luz recibida. Posee bastante flexibilidad y portabilidad, pero la identificación de la posición debe realizarse con un rastreador adicional.

IV.2.6 SONIDO VIRTUAL

Una de las mayores expectativas de la RV es una representación múltiple de la percepción humana, esto es, involucrando todos los sentidos. Sin embargo, el desarrollo se ha concentrado mayormente en capacidades visuales. Aunque hay experimentaciones en lo táctil (como retroalimentación por fuerza con joystick con motores), así como investigaciones con dispositivos gustativos y olores, aún no constituyen tecnologías muy utilizadas. Sólo la reproducción del sonido se ha integrado adecuadamente, basándose en la tecnología multimedia de los computadores actuales.

En la mayoría de los sistemas virtuales se pueden incorporar sonidos, internos o externos, en tiempo real y establecer una posición como fuente para controlar el volumen según la distancia del usuario, lo que se conoce como “sonido volumétrico”. Sin embargo el reconocimiento de obstáculos físicos o reverberancia de materiales aún está en desarrollo. En todo caso la mayoría de los cascos contempla audífonos de distinto tamaño que se pueden conectar a la tarjeta de sonido del PC y, adecuadamente configurado el mundo virtual, tener una percepción sonora tridimensionalmente coherente en el ambiente computacional.



ILUSTRACIÓN IV.15
Rastreador Intersense (giroscopio)



ILUSTRACIÓN IV.16
DataGlove



ILUSTRACIÓN IV.17
Audífonos del Vfx-1 Forte

IV.3 APLICACIONES

La RV puede ser aplicada en cualquier campo de la actividad humana, sus posibles usos son bastante amplios ya que los mundos simulados no necesariamente tienen que adaptarse a las leyes físicas naturales. Aunque en sus orígenes, los primeros ejemplos han consistido en recorridos arquitectónicos o entrenamiento para el manejo de aeronaves y videojuegos, en la actualidad ha trascendido a muchos otros campos, tales como la medicina, la industria, el diseño, el arte, etc.

La RV busca modelar la Realidad misma; por ello, tiene cientos de aplicaciones, ya que permite crear prácticamente cualquier clase de experiencia, tanto real como imaginaria y se puede aplicar en cualquier área del conocimiento: Arquitectura, Arte y Diseño, Astronomía, Ciencias Físicas y Químicas, Deporte, Educación, Electrónica, Entrenamiento, Industria, Juegos, Mecánica, Medicina, Mercadeo, Milicia, Psicología, Simuladores, Turismo.

Los arquitectos están llevando a sus clientes a pasear por edificaciones que aún no existen en el mundo real, haciendo cambios interactivamente según sus sugerencias. Los educadores están creando tutoriales en aulas que existen sólo dentro de los mundos virtuales de sus computadoras, y los están usando para enseñar, Física, Matemática, Química, y Geografía.

Los médicos están planeando de antemano las operaciones quirúrgicas complejas y los estudiantes de medicina se están entrenando con “cadáveres virtuales” y situaciones de emergencia en “hospitales virtuales”.

Los artistas están encontrando también un nuevo medio expresivo en la tecnología de RV. Se crean no sólo esculturas y pinturas estáticas, sino experiencias dinámicas que combinan los mejores elementos de otras manifestaciones artísticas, incluyendo el cine y el teatro. Los ingenieros y diseñadores laboran conjuntamente de un modo nuevo y excitante, que resultaba inimaginable hasta hace sólo unos pocos años.

IV.3.1 ARQUITECTURA

Una de las aplicaciones “naturales” de la RV ha sido la simulación de proyectos arquitectónicos. El diseño de un edificio es eminentemente un trabajo de anticipación que se realiza con planos técnicos difíciles de comprender, de modo que reproducir una caminata por la obra, similar a la visita de un futuro ocupante, posee enormes ventajas expresivas.

Imagine que se puede estar adentro de un edificio que aun no ha sido construido, y puede recorrerlo, observar a través de sus ventanas y ver la forma como va a quedar la decoración. Las construcciones virtuales invitan a los participantes a entrar en ellas. Un PV les proporciona otra forma de comprender los planos experimentalmente. Antes de que un edificio sea construido, el arquitecto y el cliente pueden ver y modificar planos, andando de habitación en habitación, a través de un modelo tridimensional del espacio generado por computadora.

IV.3.2 EDUCACIÓN

La capacidad de la RV en la educación es muy grande, la mayor proyección de los medios virtuales en educación constituye su convergencia con las nuevas tecnologías de comunicación, pues el alumno está dentro de un ambiente diseñado que contiene todos los elementos necesarios para facilitar el aprendizaje. La RV se puede usar en áreas artísticas, para el desarrollo de la creatividad, o en áreas técnicas como física, química y matemáticas, donde ayuda a la comprensión de conceptos abstractos o aspectos difícilmente visibles en condiciones normales.

Se avistan distintas posibilidades educativas para la RV. Más allá de las predicciones de la ciencia-ficción, estas aplicaciones serán relevantes debido a que un conocimiento se retiene mucho más cuando se experimenta directamente. Además, la capacidad de representar escenarios distantes o elementos abstractos puede permitir una mejor comprensión de algunos hechos o fenómenos para los estudiantes.

La tecnología de la RV ha sido ampliamente señalada como un desarrollo tecnológico importante que puede apoyar al proceso de enseñanza – aprendizaje. Permite a los estudiantes la visualización de conceptos abstractos, observar eventos físicamente imposible de lograr, o visitar ambientes e interactuar con eventos que la distancia, el tiempo o los factores de seguridad los hacen completamente inalcanzables en condiciones reales.

IV.3.3 ENTRETENIMIENTO

Al permitir la RV crear cualquier clase de mundo o fantasía, el uso en entretenimiento es enorme. Se pueden crear juegos sobre cualquier tema, recorridos increíbles o interacciones con los usuarios que antes no se podían.

Distintas instalaciones de laboratorio han sido expuestas como juegos virtuales, cascos y guantes en situaciones de combate, sillas con soportes neumáticos para experimentar montañas rusas, grandes pantallas estereoscópicas para recorridos, etc. También se han mostrado, en distintas ferias, aparatos o teatros virtuales para breves experiencias de inmersión en ambientes abstractos o reales. Algunos de los proveedores importantes de video juegos han experimentado con diversos dispositivos virtuales, visores estereoscópicos, guantes de acción, volantes sincronizados, etc.

Una evolución algo más inesperada se ha dado en la difusión del entretenimiento y en el cine, para representar actores virtuales y la simulación de escenografías tridimensionales interactivas que han constituido un negocio creciente en el sector.

IV.3.4 MEDICINA

La sofisticada preparación de los médicos en órganos difícilmente visibles, el desarrollo de tratamientos a distancia y operaciones con mínimas alteraciones anatómicas, han sido posibilidades concretas para implementar tecnologías de RV.

La exploración e intervención médica a niveles celulares y genéticos es facilitada en un entorno virtual, ya que sus características de disminución y aumento de escalas permiten a los cirujanos trabajar como si sus áreas fueran expandidas. Plantas médicas y facilidades virtuales, equipos virtuales y pacientes virtuales proporcionan a los practicantes y estudiantes más oportunidades de experiencias sin compromiso de las que han sido posibles hasta ahora.

Los lugares del cuerpo previamente inaccesibles o que inhibían intervenciones quirúrgicas o mecánicas son áreas candidatas a intervenciones microquirúrgicas, que están ayudadas por exhibiciones virtuales ampliadas. Una realización virtual muy ampliada que mantenga un paralelismo con la realidad, proporciona a los cirujanos una opción añadida para ayudarles a visualizar mejor el lugar de la operación, las micromáquinas y sus componentes de trabajo.

IV.3.5 INDUSTRIA

Se ha trabajado en algunas alternativas en la capacitación industrial. La formación de operadores de herramientas complejas, evitando gastos de operación real y riesgos de seguridad. Se pueden crear sistemas de entrenamiento, que simulan las condiciones reales del ambiente de trabajo, y anulan cualquier clase de peligro. Además de aprender dentro de este entorno virtual, con la confianza de que si comete algún error, este no va a provocar un accidente serio. Esto es útil en la industria, en áreas como: plataformas petroleras, ensambladoras, minas, fundidoras.

IV.3.6 CAVE

“Esta habitación de pantallas interactivas a las que llamamos CAVE es como un cuartito de estar controlado por computadoras que siempre saben donde estás y ajustan el punto de vista en las pantallas, con lo que te da la impresión de que estás dentro de ese mundo virtual”⁶, dice Carolina Cruz Neira Ingeniera Española y creadora del CAVE.

El CAVE (Cueva) es un sistema espacial inmersivo que permite a diferentes grupos de personas entrar, navegar, e interactuar en el mismo mundo virtual tridimensional. En los CAVE C6, todas las caras del cubo, incluidos el suelo y el techo, son pantallas de videoproyección.

⁶ Muy Interesante, Editorial Televisa, Año XX, No. 3, México, Pág. 39.

Algunas personas lo han comparado con la Holosala de Star Trek, pero para eso todavía falta, por que en el CAVE no hay sensaciones de suave o áspero, pesado o ligero..., y tampoco puedes andar físicamente, debido a las dimensiones físicas del cuarto, las cuales se vuelven una limitante. La interacción virtual en un CAVE es muy sencilla; en el VRAC (Centro de RV) han comprobado que la gente de la calle se mueve muy natural y aprenden experimentando, igual que van de turistas a un lugar de vacaciones.

El proyecto Miller es uno de los primeros proyectos que patrocinó la fundación Miller, fue el virtual heritage (patrimonio virtual), para las clases de arquitectura histórica que se dan en el CAVE. Los edificios empiezan en la era de las pirámides egipcias y van hasta nuestra época. Una vez terminada la reconstrucción, dieron paso a la segunda fase del proyecto, que consistió en recrear lo ocurrido en el interior de los edificios históricos. El primer proyecto fue la ceremonia del cambio de estaciones en un templo hindú. Se incorporaron actores virtuales con motor de inteligencia artificial en tiempo real, que les permitió comportarse como sacerdotes, feligreses, músicos... Se devuelve la vida al patrimonio histórico, no sólo al edificio sino también todo lo que sucedía en su interior.

Al estar inmerso eres un observador, pero a la vez participas en la ceremonia porque todo sucede a escala normal, los actores virtuales se dirigen a ti, entras en un templo y te hacen espacio, etcétera. Lo llaman "aprendizaje a través de la experiencia".

IV.3.7 OTRAS APLICACIONES

En la experimentación real encontramos otro tipo de aplicaciones, instalaciones artísticas que exploran intensivamente la utilización de medios virtuales son frecuentes en los Museos. Como también usos en la misma informática, la enseñanza de los componentes internos de los computadores con modelos virtuales, o la representación tridimensional de complejas bases de datos. La representación de fenómenos naturales y sus posibilidades de control abren también expectativas en la forestación, ganadería y agricultura. También se ha utilizado la modelación de acontecimientos criminales que se dirimen en un juzgado. Como apoyo a evidencia en los casos legales y forenses, es una herramienta poderosa y capaz de mostrar la información.

En la administración empresarial se han desarrollado algunos ejemplos de representación interactiva de datos complejos para la toma de decisiones. También la simulación del comportamiento de incendios o el movimiento de la gente en caso de catástrofes ha permitido formular aplicaciones en la seguridad pública.

En una tecnología tan multifacética como la RV, indudablemente se abrirán muchas más posibilidades con el avance del tiempo y con la experimentación práctica de cada campo.



CAPÍTULO V

INGENIERÍA DE SOFTWARE



V.1 INGENIERÍA DE SOFTWARE

La Ingeniería de Software es el establecimiento y uso de principios de ingeniería para obtener software que sea confiable y que funcione eficientemente en máquinas reales.

La Ingeniería de Software es relativamente nueva ya que aparece a finales de los años sesenta y principios de los setenta, comenzando con las Técnicas de Programación Estructurada, incorporándolas a las fases del ciclo vital de software. La Programación Estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño. Además, comenzaron a usarse tecnologías orientadas a objetos. En un principio la programación era la tarea de oro de la Ingeniería de Software pero ahora la ingeniería y el diseño de requisitos son más importantes.

En los años noventa la gerencia de proyecto ganó interés y llegó a ser un componente importante en la Ingeniería de Software. En la década pasada, los estándares de la Ingeniería de Software y la madurez de proceso han caracterizado la industria del software como una disciplina madura.

En un nivel más técnico, la Ingeniería de Software comienza con una serie de tareas que hacen modelos y que resultan en una especificación completa de requisitos y una representación comprensiva de diseño del software que será construido. Se han desarrollado muchos métodos para hacer modelos de sistemas de información. Sin embargo, los métodos Orientados a Objeto (OO) van a llegar a ser el estándar.

El Desarrollo de Software Orientado a Objetos comenzó en los años ochenta como una etapa natural de los métodos estructurados. UML (Unified Modeling Language) ha emergido como una unificación de los diversos métodos orientados a objetos y se está convirtiendo en un estándar.

V.1.1 ¿QUÉ ES LA INGENIERÍA DE SOFTWARE?

El término de Ingeniería de Software se puede definir de una manera muy sencilla, algunas de estas definiciones se presentan a continuación:

- Es la ciencia que ayuda a elaborar sistemas con el fin de que sea económico, fiable y funcione eficientemente sobre las máquinas reales.
- Es el estudio de principios y métodos para el desarrollo y mantenimiento de sist. de SW.
- Es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación necesaria requerida para desarrollar, operar (funcionar) y mantenerlos.

Por otro lado, una definición del término Ingeniería de Software mas completo es el siguiente: La Ingeniería del software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

Hoy día es cada vez más frecuente la consideración de la Ingeniería del Software como una nueva área de la Ingeniería, y el Ingeniero del Software comienza a ser una profesión implantada en el mundo laboral internacional.

La ingeniería del software trata con áreas muy diversas de la Informática y de las Ciencias de la Computación, tales como construcción de compiladores, sistemas operativos o desarrollos de Intranet / Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, redes, etc.

La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos, estos facilitan al gestor a controlar el proceso de desarrollo de software y suministra a los que practican dicha ingeniería las bases para construir software de alta calidad y a continuación se describen cada uno de ellos:

- **Los métodos de la ingeniería de software.** Suministran el cómo construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño de procedimientos algorítmicos, codificación, prueba y mantenimiento.

Los métodos de la ingeniería de software introducen una notación especial orientada al lenguaje o gráfica y a un conjunto de criterios para la calidad del software.

- **Las herramientas de ingeniería de software.** Suministran un soporte automático o semiautomático para los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software llamado ingeniería de software asistido por computadora, Las herramientas determinan como aplicar los métodos para tener un desarrollo, ya que son los métodos necesarios para desarrollar el sistema.

Por mencionar alguna de estas herramientas existen las herramientas CASE. (Ingeniería de software asistida por computadora (Computer Aided Software Engineering).

- **Los procedimientos de la ingeniería de software.** Facilita un desarrollo racional y oportuno del software de computadoras.

Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas que se requieren y los controles que ayuden asegurar, la calidad y coordinar los cambios y las guías que facilitan a los gestores de software establecer el desarrollo.

V.1.2 PARADIGMAS

La ingeniería de software esta compuesta por una serie de pasos de abarcan los métodos, las herramientas y los procedimientos antes mencionados. Estos pasos se denominan frecuentemente paradigmas de la ingeniería de software. La elección de un paradigma para la ingeniería de software se lleva a cabo de acuerdo con la naturaleza del proyecto y de la aplicación, los métodos, herramientas a usar, los controles y entregas requeridos.

El siguiente diagrama presenta la gama de paradigmas de la ingeniería de software:

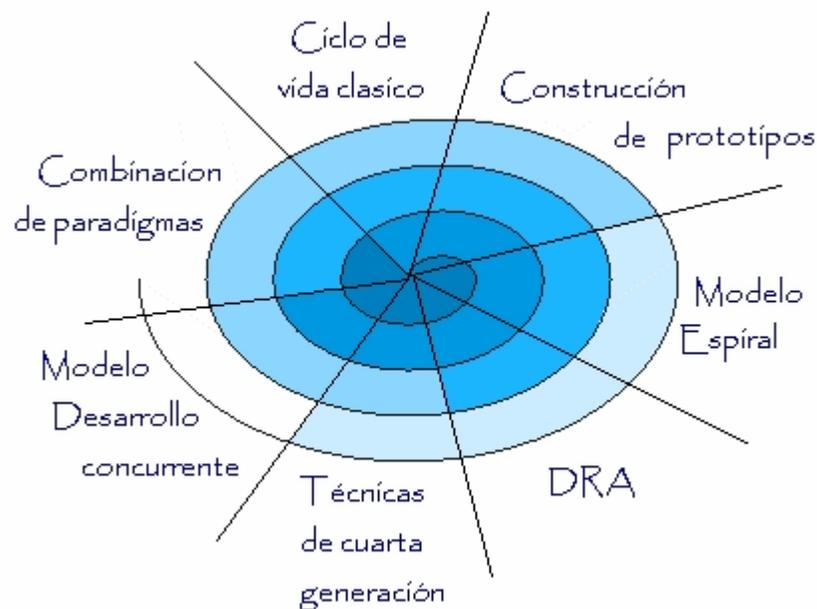


ILUSTRACIÓN V.1
Gama de paradigmas de la Ingeniería de Software

V.1.3 EL PROCESO

Las fases genéricas que caracterizan el proceso de software (definición, desarrollo y mantenimiento) son aplicables a todo software. El problema es seleccionar el modelo de proceso apropiado para la ingeniería de software que debe aplicarse en el proyecto.

El gestor del proyecto debe decidir que método de proceso es el mas adecuado para el proyecto, después debe definir un plan preliminar basado en un conjunto de actividades estructurarles validas para cualquier proyecto. Una vez establecido el plan preliminar, empieza la descomposición del proceso. Es decir, se debe crear un plan completo reflejando las tareas requeridas a las personas para cubrir las actividades estructurales.

MADURACIÓN DEL PROBLEMA Y EL PROCESO

La planificación del proyecto empieza con la maduración del problema y del proceso. Todas las funciones que se deben tratar dentro de un proceso de ingeniería por el equipo de software deben pasar por el conjunto de actividades estructurales que se han definido por una organización de software. Asuma que la organización ha adoptado el siguiente conjunto de actividades estructurales:

- Comunicación con el cliente: tareas requeridas para establecer una comunicación eficiente entre el desarrollo y el cliente.
- Planificación: Tareas requeridas para definir los recursos, la planificación temporal del proyecto y cualquier información relativa a él.
- Análisis de riesgo: Tareas requeridas para valorar los riesgos técnicos y de gestión.
- Ingeniería: Tareas requeridas para construir una o más representaciones de la aplicación.
- Construcción y entrega: Tareas requeridas para construir, probar, instalar y proporcionar asistencia al usuario. (ejemplo: documentación y entrenamiento).
- Evaluación del cliente: Tareas requeridas para obtener la información de la opinión del cliente basadas en la evaluación de las representaciones de software creadas durante la fase de ingeniería e implementadas durante la fase de ingeniería e implementadas durante la fase de instalación.

DESCOMPOSICIÓN DEL PROCESO

Una vez que se ha elegido el modelo de proceso, la estructura común de proceso (ECP) se adapta a él. En todos los casos, la ECP (comunicación con el cliente, planificación, análisis de riesgo, ingeniería, construcción, entrega y evaluación del cliente) puede adaptarse al paradigma. Funcionara para modelos lineales, para modelos interactivos e incrementales, para modelos de evolución e incluso para modelos concurrentes o de ensamblaje de componente. La ECP es invariable y sirve como base para todo el trabajo de software realizado por una organización.

La descomposición de trabajo empieza cuando el gestor del proyecto pregunta: "¿cómo vamos a realizar esta actividad?".

Si consideremos un proyecto completo que tenga un ámbito amplio y un mayor impacto comercial, podría requerir las siguientes tareas para la actividad de comunicación con el cliente:

- Revisar la petición del cliente.
- Planificar y programar una reunión formal con el cliente.
- Realizar una investigación para definir soluciones propuestas y enfoques existentes.
- Preparar un "documento de trabajo" y una agenda par la reunión formal.
- Realizar una reunión.
- Desarrollar conjuntamente especificaciones que reflejan la información, función y características de comportamiento de software.
- Revisar las especificaciones para comprobar su corrección, consistencia y ambigüedad.
- Ensamblar las mini especificaciones en un documento de alcance del proyecto.
- Revisar ese documento general con todo lo que pueda afectar.
- Modificar el documento de alcance del proyecto cuando se requiera.

V.1.4 VISION GENÉRICA DE LA INGENIERÍA DE SW

Independientemente del paradigma que se utilice, del área de aplicación, y del tamaño y la complejidad del proyecto, el proceso de desarrollo de software contiene siempre una serie de fases genéricas, existentes en todos los paradigmas. Estas fases son la definición, el desarrollo y el mantenimiento.

DEFINICIÓN (¿QUÉ?). La fase de definición se centra en el qué. Durante esta fase, se intenta identificar:

- qué información es la que tiene que ser procesada.
- qué función y rendimiento son los que se esperan.
- qué restricciones de diseño existen.
- qué interfaces deben utilizarse.
- qué lenguaje de programación, sistema operativo y soporte de hardware se utilizarán.
- qué criterios de validación se necesitan para conseguir que el sistema final sea correcto.

Aunque los pasos concretos dependen del modelo de ciclo de vida utilizado, en general se realizarán dos tareas específicas:

- Análisis de requisitos.
 - Establecer el ámbito del software.
 - Definición detallada de la función del software.
- Planificación.
 - Análisis de riesgos.
 - Definición de tareas.
 - Estimación de costos.
 - Asignación de recursos.

DESARROLLO (¿CÓMO?). La fase de definición se centra en el cómo.

- cómo ha de ser la arquitectura de la aplicación.
- cómo han de ser las estructuras de datos.
- cómo han de implementarse los detalles procedimentales de los módulos.
- cómo van a ser las interfaces.
- cómo ha de traducirse el diseño a un lenguaje de programación.
- cómo van a realizarse las pruebas.

Aunque, al igual que antes, los pasos concretos dependen del modelo de ciclo de vida utilizado, en general se realizarán tres tareas específicas:

- Diseño.
 - Arquitectura de la aplicación.
 - Estructura de los datos.
 - Estructura interna de los programas.
 - Diseño de las interfaces.
- Codificación.
- Pruebas.

MANTENIMIENTO (CAMBIO). La fase de mantenimiento se centra en los cambios que va a sufrir el software a lo largo de su vida útil. Como ya hemos dicho, estos cambio pueden deberse a la corrección de errores, a cambios en el entorno inmediato del software o a cambios en los requisitos del cliente, dirigidos normalmente a ampliar el sistema.

La fase de mantenimiento vuelve a aplicar los pasos de las fases de definición y de desarrollo, pero en el contexto de un software ya existente y en funcionamiento.

Aunque, al igual que antes, los pasos concretos dependen del modelo de ciclo de vida utilizado, en general se realizarán tres tareas específicas:

- Utilización.
- Mantenimiento.
 - Corrección de errores.
 - Cambios en el entorno.
 - Cambios en los requisitos.
- Sustitución.

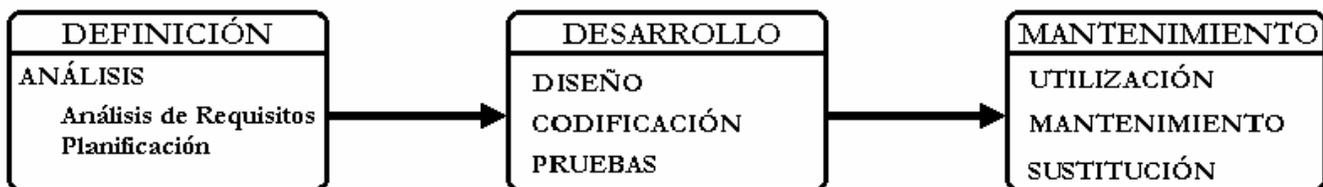


ILUSTRACIÓN V.2
Diagrama de etapas de la Ingeniería del Software

V.2 CICLO DE VIDA

Por ciclo de vida se entiende por la sucesión de etapas por las que pasa el software desde que un proyecto es concebido hasta que se deja de usar (desde el planteamiento del problema a resolver mediante el producto software, hasta la retirada de dicho producto una vez que ha finalizado su vida operativa.). Cada una de estas etapas lleva asociada una serie de tareas que deben realizarse, y una serie de documentos que serán la salida de cada una de las fases y servirán de entrada a la siguiente. Estas etapas son agrupadas en Fases (Unidad de trabajo de un proyecto software que culmina en la consecución de un punto clave del proyecto o hito).

Es probable que el modelo inicial del ciclo de vida del software fue propuesto por Royce en 1970, y desde entonces el modelo ha tenido varios perfeccionamientos y variaciones.

Existen diversos modelos de ciclo de vida, es decir, diversas formas de ver el proceso de desarrollo de software, y cada uno de ellos va asociado a un paradigma de la ingeniería de software, es decir, a una serie de métodos, herramientas y procedimientos que debemos usar a lo largo de un proyecto. La elección de un paradigma u otro se realiza de acuerdo con la naturaleza del proyecto y de la aplicación, los métodos a usar y los controles y entregas requeridos.

V.2.1 FASES DEL CICLO DE VIDA

Este modelo divide el ciclo de vida del producto en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados. Las fases son las siguientes: análisis, diseño, instrumentación, pruebas y mantenimiento.

V.2.1.1 ANÁLISIS

En esta fase se establecen los servicios, restricciones y objetivos del sistema. Una vez determinados, deben definirse de una manera comprensible, tanto para los usuarios como para los desarrolladores.

Consta de dos subfases: definición de requisitos y planeación.

- ANÁLISIS DE REQUISITOS

La definición de requisitos se refiere a la identificación de las funciones básicas de los componentes de programación en un sistema. Se pone atención en las funciones y restricciones bajo las cuales se deben desarrollar.

Incluye la comprensión del problema del cliente, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación, y planeación del proceso de desarrollo. De esta fase surge la Definición del Sistema y el Plan de Proyecto.

La primera etapa del ciclo de vida de un proyecto consiste en un análisis de las características y el comportamiento del sistema del cual va a formar parte. En el caso de que se quiera implementar un sistema nuevo se deberá analizar cuales son los requisitos y la funcionalidad del sistema. Por el contrario, para uno ya existente se deberá analizar el funcionamiento operativo del lugar en el que se va a implantar para posteriormente adecuarlo a las funciones que se van a automatizar.

En resumen consiste en analizar y comprender todos los elementos del entorno en el que el software se va a integrar. Hay que determinar perfectamente los requisitos globales del sistema, así como los elementos típicos con los que el software se puede relacionar en un sistema.

- PLANIFICACIÓN

Durante esta etapa se lleva a cabo el análisis de riesgos, se definen los recursos necesarios para desarrollar el software y se establecen las estimaciones de tiempo y costos. El propósito de esta etapa de planificación es proporcionar una indicación preliminar de la viabilidad del proyecto de acuerdo con el costo y con la agenda que se hayan establecido. Posteriormente, la gestión del proyecto durante el desarrollo del mismo realiza y revisa el plan de proyecto de software.

V.2.1.2 DISEÑO

Es el proceso que traduce los requisitos en un representación de las funciones de cada sistema del software que describe cómo va a estructurarse los datos, cuál va a ser la arquitectura de la aplicación, cuál va a ser la estructura de cada programa y cómo van a ser las interfaces; así como la funcionalidad e incluso la calidad del mismo a fin de poderlo transformar con facilidad en uno o más programas de computación para poder posteriormente comenzar la fase de codificación. Básicamente esta fase se centra en el diseño de las siguientes partes fundamentales del programa:

- De los Datos.
- De la Arquitectura.
- De las Interfaces.
- De la Programación.

El diseño se divide en estructural y detallado. El diseño estructural comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales y la interconexión entre componentes. El diseño detallado se refiere a cómo empacar los módulos de procesamiento, y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

Al igual que el análisis, el diseño debe documentarse y formar parte de la configuración del sistema. Teóricamente, con esta documentación, los programadores se pueden poner a programar de inmediato (en la fase de codificación), ya que el documento de diseño les deja especificado las estructuras de datos, las funciones, las ventanas, los formularios que son necesarios para programar.

V.2.1.3 CODIFICACIÓN

Durante esta etapa, el diseño del software se realiza como un conjunto de programas o unidades de programa escritos en algún lenguaje de programación ejecutable, el cual consiste en la traducción del diseño a un formato que sea legible para la máquina, es decir, en un código fuente, dando como resultado un programa ejecutable. Si el diseño se realizó correctamente, la codificación debería ser una tarea fácil. Si el diseño es lo suficientemente detallado, la codificación es relativamente sencilla y puede hacerse en forma automática usando generadores de código.

Podemos observar que las fases del ciclo de vida consisten básicamente en una traducción:

- En el análisis del sistema, los requisitos, la función y la estructura de este se traducen en un documento.
- En el análisis de requisitos se profundiza en el estudio del componente de software del sistema y esto se traduce a un documento.
- En el diseño, los requisitos del software se traducen a una serie de diagramas que representan la estructura del sistema, de sus datos, de sus programas y de sus interfaces.
- Y por último, en la codificación se traducen estos diagramas de diseño a un lenguaje fuente, que luego se traduce (compila) para obtener un programa ejecutable.

V.2.1.4 PRUEBAS

Una vez que ya se tiene el programa ejecutable comienza la fase de pruebas y es preciso probarlo para detectar errores de codificación, de diseño o de especificación. Las pruebas son necesarias para encontrar el mayor número posible de errores antes de entregar el programa al cliente.

Es necesario probar cada uno de los componentes por separado (cada uno de los módulos o programas) para comprobar el rendimiento funcional de cada una de estas unidades. La prueba se centra en la lógica interna del software y su objetivo es comprobar que no se hayan producido errores en algunas de las fases de traducción anteriores, especialmente en la codificación, para ello deben probarse todas las sentencias, no solo en condiciones normales y además, como parte de esta etapa, se deben probar todos los módulos que conforman el sistema.

A continuación se procede a integrar los componentes para probar toda la arquitectura del software, y probar su funcionamiento y las interfaces. En este punto hay que comprobar si se cumplen todos los requisitos de la especificación.

En términos generales, la fase de pruebas debe contemplar los siguientes puntos:

- Se deben probar todas las unidades lógicas del programa.
- Todas las entradas deben producir una salida.
- Una vez realizadas todas estas pruebas, se debe realizar la prueba de integración del programa dentro del sistema, como un sistema completo para asegurar que se cubren las necesidades del software. Es decir, su funcionamiento en el punto en el cual va a ser utilizado día a día.

Las pruebas del sistema comprenden dos tipos de actividades: pruebas de integración y de aceptación. El desarrollo de una estrategia para integrar los componentes de un sistema de programación requiere una planeación cuidadosa. Las pruebas de aceptación se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema satisface las necesidades establecidas.

V.2.1.5 UTILIZACIÓN

Esta fase suele ser (aunque no necesariamente) la más larga del ciclo de vida. Se instala el sistema y se pone en uso práctico.

Una vez superada la fase de pruebas, el software ha sido aceptado por el cliente y se entrega a este para su operación y comienza la vida útil del mismo e inicia la fase de mantenimiento. La fase de utilización se sustenta con las posteriores ejecuciones - el mantenimiento y la sustitución- y dura hasta que el software, ya reemplazado por otro, deje de utilizarse o de ser útil.

V.2.1.6 MANTENIMIENTO

Se considera mantenimiento a cualquier cambio que se realice en el programa una vez que este se ha instalado en su entorno de trabajo. La actividad de mantenimiento implica corregir errores que no se descubrieron en las primeras etapas del ciclo de vida, mejorar la aplicación de las unidades del sistema y aumentar los servicios de éste a medida que se perciben nuevas necesidades. El software sufrirá cambios a lo largo de su vida útil, estos pueden ser debido a distintas causas:

- La corrección de errores no detectados en la fase de prueba.
- La petición del usuario del cambio de alguna característica que no le satisface.

- Que durante la utilización, el cliente detecte errores en el software.
- Que se produzcan cambios en alguno de los componentes del sistema informático.
- Que el cliente requiera modificaciones funcionales (normalmente ampliaciones) no contempladas en el proyecto.

En cualquier caso el mantenimiento supone volver atrás en el ciclo de vida, a las etapas anteriores dependiendo de la magnitud del cambio. Todos los cambios se verán sujetos igualmente a las fases de análisis, diseño y codificación del ciclo de vida.

V.2.1.7 SUSTITUCIÓN

La vida del software no es ilimitada y cualquier aplicación, por buena que sea, acaba por ser sustituida por otra más amplia, más rápida o más fácil de usar. La sustitución de un software que está funcionando por otro que acaba de ser desarrollado es una tarea que hay que planificar cuidadosamente y que hay que llevar a cabo de forma organizada.

Es conveniente realizarla por fases, si esto es posible, no sustituyendo todas las aplicaciones de golpe, puesto que la sustitución conlleva normalmente un aumento de trabajo para los usuarios, que tienen que acostumbrarse a las nuevas aplicaciones, y también para los implementadores, que tienen que corregir los errores que aparecen.

La sustitución implica el desarrollo de programas para la interconexión de ambos sistemas, el viejo y el nuevo, y es necesario hacer un vaciado de la información que maneja el sistema viejo a la estructura y el formato requeridos por el nuevo. Además, es conveniente mantener los dos sistemas funcionando en paralelo durante algún tiempo para comprobar que el sistema nuevo funcione correctamente y para asegurarnos el funcionamiento normal del programa aún en el caso de que el sistema nuevo falle y tenga que volver a alguna de las fases de desarrollo.

V.2.2 CICLO DE VIDA CLASICO (O EN CASCADA)

El paradigma denominado “cascada o clásico” es de los más antiguos empleados en la ingeniería de software, basado en el ciclo convencional en ingeniería, el cual surgió como una copia de otras ingenierías y se desarrollo a partir del ciclo convencional, para dar solución a los problemas más comunes que aparecían al desarrollar sistemas de software complejos.

Es un ciclo de vida en sentido amplio y se trata de un enfoque secuencial y sistemático del desarrollo de software, que incluye todas las etapas de ingeniería de software las cuales describen el desarrollo de la vida del producto: desde el análisis del sistema de los requerimientos, pasando por el diseño y el desarrollo, hasta las pruebas, el uso (la vida útil del software) y el mantenimiento, hasta que llega el momento de sustituirlo.

Este ciclo de vida tiene algunas ventajas, las cuales se mencionan a continuación:

- Es sencillo y fácil de implantar y gestionar.
- En ingeniería se está muy acostumbrado a este paradigma.

Sin embargo el ciclo de vida “cascada o clásico” ha sido criticado debido a los problemas que se plantean al intentar aplicarlo en determinadas situaciones, por ejemplo:

- En realidad los proyectos no siguen un ciclo de vida estrictamente secuencial como propone el modelo. Siempre hay iteraciones, esto implica que siempre hay que volver a alguna de las fases anteriores, pero también es muy frecuente en que una fase, se detecten errores que obliguen a volver a la fase anterior.
- Es difícil que se puedan establecer inicialmente todos los requisitos del sistema. Sin embargo, el ciclo de vida clásico requiere la definición inicial de todos los requisitos y no es fácil acomodar en él las incertidumbres que suelen existir al comienzo de todos los proyectos. Es difícil que se puedan establecer inicialmente todos los requisitos del sistema. Los requisitos se van aclarando y refinando a lo largo de todo el proyecto, según se plantean dudas concretas en el diseño o la codificación.
- Hasta que se llega a la fase final del ciclo de vida: en el desarrollo de la codificación, no se dispone de una versión operativa del programa. Como la mayor parte de los errores se detectan cuando el cliente puede probar el programa y no antes, entonces estos errores no se detectan si no hasta el final del proyecto, cuando son más costosos de corregir y más presiones hay por que el programa se ponga definitivamente en marcha.

Es mucho mejor desarrollar software siguiendo el modelo de ciclo de vida en cascada que hacerlo sin ningún tipo de guías. Además, este modelo describe una serie de pasos genéricos que son aplicables a cualquier otro paradigma.

De acuerdo con las ventajas y desventajas que ofrece el ciclo de vida clásico, nos basaremos en un ciclo de vida alternativo llamado **“Top Down – Botton Up” (Arriba Abajo – Abajo Arriba)**, el cual tiene la gran ventaja, de permitir regresar hacia alguna etapa preeliminar, de modo que se logran depurar los errores que se presentan y no fueron observados en su momento (Abajo Arriba), así como proseguir con la secuencia normal del ciclo por sus distintas etapas (Arriba Abajo).

Podemos decir que el ciclo de vida Top Down – Botton Up, es el ciclo de vida clásico, pero con la gran ventaja que nos permite avanzar o retroceder libremente entre las distintas etapas, de forma secuencial o alterna, dependiendo de las necesidades del proyecto.

Este exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel de ingeniería de sistemas y avanza a través fases secuenciales sucesivas. Estas fases son las mismas que sigue el ciclo de vida en cascada, las cuales son las siguientes: Análisis, Diseño, Codificación, Pruebas, Utilización, Mantenimiento, Sustitución.

Es útil diferenciar las fases del ciclo de vida del software, pero, en la práctica, cabe destacar las siguientes características que se presentan para mejorar los procesos y los resultados:

- Las etapas de desarrollo se superponen y proporcionan información unas a otras.
- La fase final del ciclo de vida (Utilización, Mantenimiento y Sustitución) se han colocado en una posición particular a propósito, pues, durante estas fases, la información es retroalimentada a todas las fases previas del ciclo.

Utilizaremos el ciclo de vida Top Down – Botton Top, debido a que podemos interactuar entre las etapas que lo componen y avanzar o retroceder entre las etapas para la corrección de errores, de manera secuencial o alterna, dependiendo las necesidades del proyecto.

En resumen podemos decir que utilizaremos el ciclo de vida Top Down – Botton Up, debido a que esta basado o fundamentado en el ciclo de vida clásico o en cascada, aprovechando la interacción que se pueden realizar con este ciclo de vida entre las etapas que lo componen, de esta forma aprovechando las características que ofrece dicho ciclo de vida podemos depurar o corregir todos lo errores que vayan surgiendo durante el desarrollo de la aplicación, ya que como ya hemos visto, nos permite tanto avanzar en las etapas de manera secuencial o de manera alterna entre las etapas, así como la gran ventaja, de regresar entre dichas etapas, ya sea de manera secuencia o bien de manera alternada.

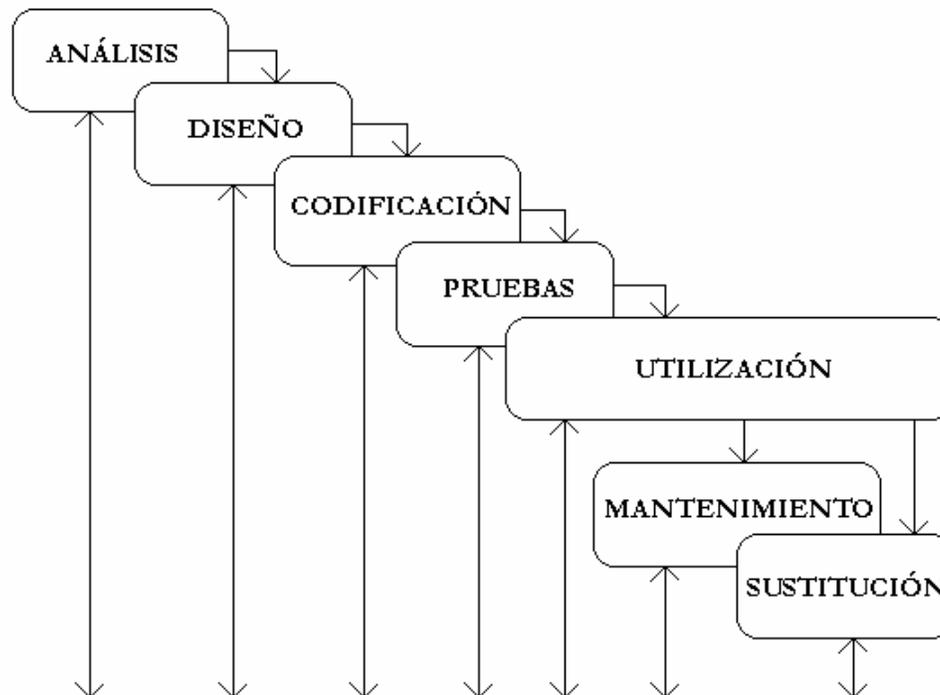


ILUSTRACIÓN V.3

Diagrama del paradigma del ciclo de vida Top Down – Botton Up

Finalmente, cabe destacar un aspecto importante durante el desarrollo de nuestra aplicación, sobre todo en lo que respecta a la aplicación del ciclo de vida, ya que la estructura del programa final la hemos considerado que se encuentra conformada por dos partes o subaplicaciones. La primera de ellas tiene por objetivo obtener el modelo arquitectónico completo de la zona arqueológica de Teotihuacan, esta parte es desarrollada sobre la plataforma 3D Studio MAX 5, la cual será la entrada o parte inicial de la segunda parte, que tiene por objetivo desarrollar el recorrido o paseo virtual por dicha zona arqueológica, esta subaplicación es desarrollada sobre la plataforma Shockwave de Director 8.5.

La aplicación del ciclo de vida fue para las dos partes que conforman el programa, es decir, que cada una de estas tiene sus correspondientes etapas aplicando el paradigma del ciclo de vida Top Down – Botton Up, hasta finalmente obtener el resultado deseado en cada una de las subaplicaciones y por consecuencia de la aplicación final.

Si consideramos por un lado que salida de la primera subaplicación sirve de entrada a la segunda, y la salida de la segunda es la salida de nuestra aplicación completa; podemos decir que tenemos dos secciones en serie.

Sin embargo ambas secciones se pueden traslapar parcialmente para trabajarse en forma paralela con un cierto desfaseamiento, pero siempre respetando que la entrada de la segunda aplicación corresponda a la salida de la primera.

El ciclo de vida aplicado en ambas secciones, contemplará únicamente las 4 primeras etapas, desde el análisis hasta las pruebas; y el resto de las etapas serán aplicadas al programa final. Esto puede observarse de mejor forma en el siguiente diagrama.

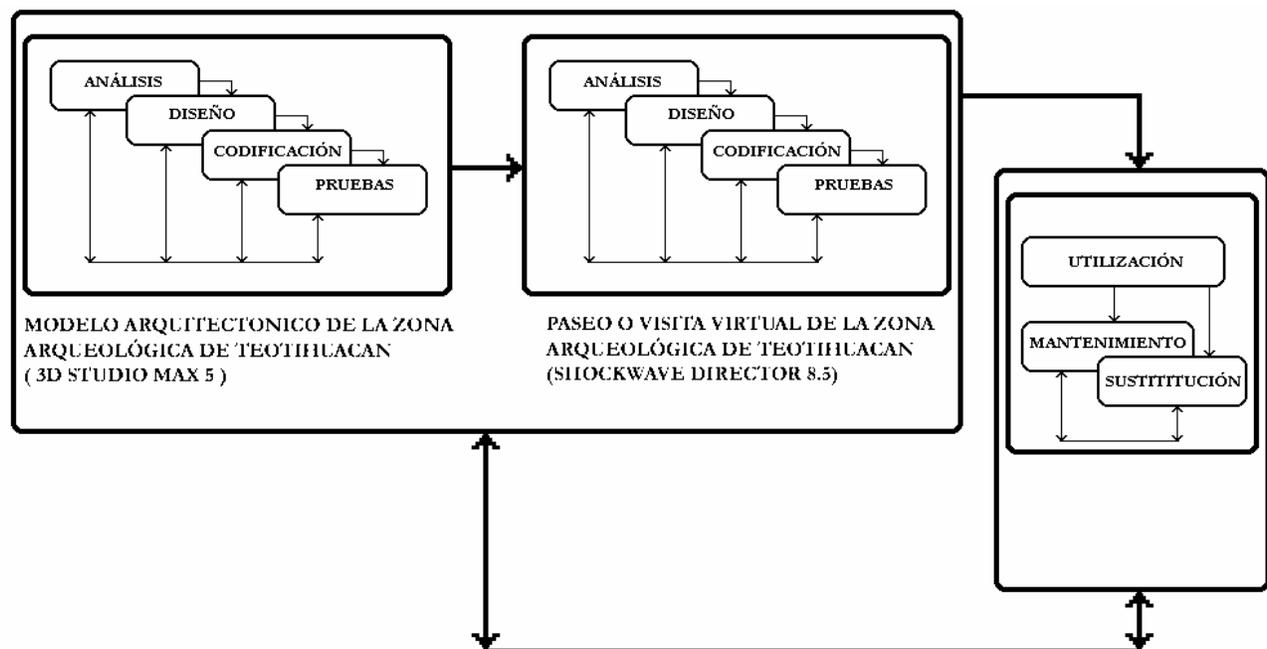


ILUSTRACIÓN V.4

Diagrama del ciclo de vida aplicado a las dos secciones



CAPÍTULO VI

ANÁLISIS Y DISEÑO



La etapa de Análisis comprende las fases; análisis de requisitos y planeación. La primera de ellas, se divide en Análisis de requisitos del sistema y Análisis de requisitos del Software.

A continuación comenzaremos a explicar cada una de las fases antes mencionadas:

VI.1 ANÁLISIS DE REQUISITOS

El análisis de requisitos es el primer paso técnico del proceso de ingeniería del software. Es aquí donde se refina la declaración general del ámbito del software en una especificación concreta que se convierte en la base de todas las actividades que siguen.

Entre las tareas que hay que realizar en esta fase, están el estudio de las características y la función del sistema, la definición de los requisitos del software, y del sistema del que el software forma parte, así como la planificación inicial del proyecto.

Todas estas tareas deben realizarse al comienzo del proyecto, pero el principal problema que se nos presentó es que en el inicio, fue difícil tener una idea clara, saber cuáles iban a ser los requisitos del sistema y del software, para llegar a comprender en su totalidad la función que el programa debería realizar. Por esto, utilizamos el modelo del ciclo de vida Top-Down Bottom-Up, ya que propone un enfoque cíclico de refinamiento.

VI.1.1 ANÁLISIS DE REQUISITOS DEL SISTEMA

Consiste en comprender cuál va a ser la función y el ámbito de la información de un nuevo proyecto. El objetivo es conseguir representar un sistema en su totalidad, incluyendo hardware y software, mostrando la relación entre los diversos componentes y sin entrar en la estructura interna de los mismos. También define a los elementos de un sistema informático dentro del contexto del sistema en que va a ser usado. Hay que identificar las funciones del sistema y asignarlas a alguno de sus componentes. Para ello, partimos de los objetivos y restricciones definidas, para realizar una representación de la función del sistema, de la información que maneja, de sus interfaces y del rendimiento y las restricciones del mismo.

La primera tarea a realizar es delimitar el sistema, indicando el ámbito de funcionamiento y el rendimiento deseado.

Una vez que se ha logrado delimitar la función, el ámbito de información, las restricciones, el rendimiento y las interfaces del sistema, se debe proceder a la asignación de funciones. Las funciones del sistema deben de ser asignadas a alguno de sus componentes (ya sean éstos software o hardware). Se debe estudiar varias opciones de asignación, teniendo en cuenta las ventajas e inconvenientes de cada una de ellas para decidirse por alguna.

Para explicar lo anterior realizaremos el siguiente desglose, de acuerdo a nuestro sistema:

- ESPECIFICACIÓN INFORMAL (FUNCIÓN)

El nombre del proyecto, **Reconstrucción de sitios arqueológicos: Teotihuacan**, consiste en el recorrido de la zona arqueológica. Permite el libre movimiento a cualquier punto que el usuario desee (similar a los video juegos). Contempla la animación de los ambientes con base en un tiempo virtual, similar al transcurso de un día completo. Dentro de las animaciones incluimos, la animación de la luz, de los sonidos, del entorno ambiental. La idea del recorrido es simular la época de esplendor teotihuacano, donde podemos apreciar la reconstrucción de la arquitectura actual.

La entrada de los datos se realiza por medio del teclado para el PV, y para poder interactuar con el resto del sistema así como cambiar de interfase haremos uso del ratón.

- OBJETIVO

Desarrollar un PV en la zona arqueológica de Teotihuacan, cuyas principales características son la facilidad de uso, la interacción ilimitada, la portabilidad de la aplicación y la generación de los ambientes virtuales en tiempo real.

- INTERFASES

Debido a que el PV está enfocado hacia usuarios que deseen conocer la historia de Teotihuacan, reforzamos el PV con diversas interfases del tipo multimedia, para que sea mas completo, ilustrativo y educativo.

- RENDIMIENTO

El rendimiento del sistema, dependerá en su totalidad del equipo de cómputo donde sea ejecutado. Mientras el equipo tenga mayor velocidad y capacidad de memoria RAM y de Video, mejor será el rendimiento del sistema. A continuación enlistamos las características mínimas y óptimas del hardware requerido para la ejecución del sistema:

- Sistema operativo Microsoft Windows, versiones 98, 2000, Me, XP
- Procesador a 750 Mhz (Recomendado: 2 Ghz)
- 256 MB de memoria RAM (Recomendado: 512 MB)
- Unidad CD ROM y Bocinas
- Tarjeta de Video de 32 MB (Recomendado: 64 MB)
- Monitor a Color, Teclado y Mouse

- RESTRICCIONES

El sistema funcionará correctamente si es ejecutado cumpliendo los siguientes puntos:

- Equipo Compatible (PC)
- Sistema Operativo Microsoft Windows

VI.1.2 ANÁLISIS DE REQUISITOS DEL SOFTWARE

Para la etapa de Análisis, el análisis de requisitos del software define con mayor precisión las funciones y rendimiento del software, las interfaces que ha de tener con otros componentes del sistema y las restricciones que debe cumplir.

Para la etapa de Diseño, el análisis de requisitos proporciona una representación de la información y la función del sistema que él se encargará de traducir en un diseño de datos y programas.

El objetivo debe centrarse en qué es lo que hay que hacer, pero no cómo hay que hacerlo, es decir, ¿qué datos debe manejar el sistema software?, ¿qué función debe realizar?, ¿qué interfaces debe tener?, y ¿qué restricciones tiene que cumplir?.

Con la información obtenida de la especificación del sistema se debe ir definiendo y refinando los flujos y la estructura de la información, la función del programa y su papel en el contexto del sistema, las características de las interfaces y las restricciones de diseño.

Con este conocimiento que se va adquiriendo del sistema software, se debe sintetizar una o varias soluciones al problema, comprobando que se ajustan al plan del proyecto.

El análisis de requisitos puede entenderse como una tarea de análisis síntesis: se debe analizar el problema y los objetivos y, a partir de esto, sintetizar soluciones. Como ayuda en este proceso de análisis síntesis, se utilizan modelos para entender mejor los flujos de información y la función de cada elemento del sistema software. Estos modelos servirán de base para el diseño de software.

- ESPECIFICACIÓN PRELIMINAR

Para el desarrollo del PV es necesario dividir éste en el modelado de los elementos geométricos y en la navegación a través en dichos elementos. Para desarrollar el modelado, podemos utilizar cualquier software que permita generar elementos tridimensionales, como es el caso de 3D Studio Max, Maya, Rhinoceros, Lightwave, etc. Para el caso de la navegación, utilizaremos cualquier software que tenga la capacidad de integrar la geometría tridimensional con la programación que nos permita movernos e interactuar dentro de dicha geometría, tal es el caso de algunos videojuegos, que nos permiten la utilización de sus correspondientes maquinas de juegos (game engines), por ejemplo, Quake, Unreal Tournament, Half Life, Brazil, etc., otro caso para la navegación es Performer para Silicon Graphics, o Shockwave 3D de Macromedia Director para PC o Macintosh.

En nuestro caso para el desarrollo del modelado, elegimos 3D Studio Max 5.0, debido a que contamos con varias fuentes de información a nuestro alcance, como son; tutoriales en la Web, foros de discusión, libros especializados, cursos, etc. Otra de las razones que nos llevaron a la elección de éste programa, es que debido a la preferencia que el mercado ha ido teniendo con el 3D Studio Max, éste se ha ido complementando y enriqueciendo con cada nueva versión, no sólo por su propio desarrollador, Discreet sino también por varias empresas especializadas en el desarrollo de programas externos (plug-ins) que complementan al mismo programa.

En nuestro caso, utilizamos el plug-in desarrollado conjuntamente por Discreet y Macromedia, el cual permite exportar los modelos generados en 3D Studio Max a Director, por medio de ShockWave 3D, almacenándolos en archivos *.W3D. Ésta fue la primera de tres razones que nos llevaron a elegir Director como plataforma par el desarrollo de la navegación del PV. La segunda razón es que cuenta con un lenguaje de programación orientado a objetos y de carácter tridimensional (Lingo 3D), que nos permite manejar los diferentes eventos que ocurren en el mundo virtual. La tercera y última razón es que nos permite personalizar cada una de las interfases requeridas y manejar en paralelo un entorno multimedia.

- ÁMBITO DE INFORMACIÓN

La tarea que realiza el software va a consistir siempre en procesar información: cualquier aplicación que consideremos responde a una estructura de entrada-procesamiento-salida (en nuestro caso cuando se presiona alguna tecla para la navegación hablamos de un evento de entrada, dicha entrada es procesada para saber que está ocurriendo dentro del mundo virtual, y finalmente es desplegada la información actualizada en el monitor, que sería nuestra salida), donde el software se encarga de procesar unos determinados datos de entrada para producir unos datos de salida. Estos pueden ser lógicos, numéricos, cadenas de caracteres, imágenes, etc.

Pero en un sistema software, la información no está representada sólo por datos sino también por sucesos o eventos. Los datos se refieren normalmente a la información que procesa el sistema y los eventos a cuándo debe procesarse esa información (en nuestro caso, podemos considerar algunos eventos como, la detección de colisiones, el cambio de horario y por consiguiente de iluminación y ambiente).

- DESCOMPOSICIÓN Y MODELADO DEL SISTEMA DE SOFTWARE

La descomposición se utiliza para abordar problemas que son demasiado grandes o demasiado complejos para resolverlos directamente. Mediante descomposición, el problema se divide en problemas más sencillos, que realizan una función más clara y que pueden entenderse más fácilmente.

La descomposición se puede aplicar a los ámbitos de la información, de la función o del comportamiento, con esto llegaremos a establecer una jerarquía de modelos, que irán desde el más general a los más específicos los cuales tendrán un mayor nivel de detalle.

En la Ingeniería del Software se utilizan modelos para la información que transforma el software (modelo de datos), para las funciones que transforman esa información (modelo de procesos) y también para definir el comportamiento del sistema (modelo de control).

Para realizar un modelo podemos utilizar una notación gráfica (y el modelo estará representado mediante una serie de diagramas) o bien una notación textual (y tendremos modelos en lenguaje natural o en un lenguaje de especificación más formal).

A continuación mostramos el diagrama general del sistema que representa las relaciones entre los elementos, y que sirve de base para las etapas posteriores.

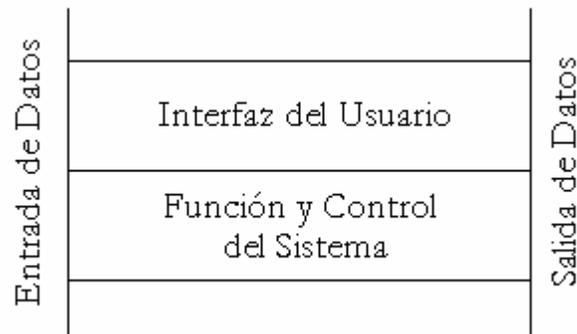


ILUSTRACIÓN VI.1

Diagrama de representación general de la arquitectura

Ahora mostramos la descomposición del sistema en los tres diferentes modelos, indicando los elementos que conforman a cada uno de ellos.

Entrada y Salida de Datos (Modelo de Datos)

- Entrada de Datos por medio del teclado y ratón
- Salida de Datos desplegados en el monitor

Función del Sistema (Modelo de Procesos)

- Detección de Colisiones
- Manejo de Cámara
- Generación del Tiempo virtual
 - o Animación de la Iluminación
 - o Animación del Sonido
 - o Generación del Ambiente

Control del Sistema e Interfaz del Usuario (Modelo de Control)

- Creación de la interfaz para el PV
- Creación de las interfases multimedia

VI.2 PLANEACIÓN

Se conoce como planeación a toda estimación de recursos, como pueden ser, tiempos, actividades y costos, necesarios para llevar a cabo un proyecto de desarrollo. Durante esta etapa se lleva a cabo el análisis de riesgos, se definen los recursos necesarios para desarrollar el software.

El propósito de esta etapa es proporcionar una indicación preliminar de la viabilidad del proyecto de acuerdo con la agenda que se hayan establecido

Esta etapa debe refinarse para crear un visión detallada del proyecto, donde el refinamiento empieza tomando en cuenta cada tarea principal y descomponiéndola en un conjunto de subtarear. Éstas forman la base para la planificación de actividades en el desarrollo del proyecto, incluso, hay que ejecutar un cierto número de pequeñas tareas antes de alcanzar el objetivo

Existen herramientas útiles para la planeación y el control de proyectos. En nuestro caso solamente contemplaremos la fase de planeación de actividades, realizando el cronograma correspondiente, que se presenta a continuación:

1. ANÁLISIS

1.1 Análisis de Requisitos

1.1.1 Análisis de Requisitos del Sistema

1.1.2 Análisis de Requisitos del Software

1.2 Planeación

2. DISEÑO

2.1 Diseño de Datos

2.1.1 Modelado

2.1.1.1 Tabla de medidas

2.1.1.2 Tabla de referencias

2.1.1.3 Tabla de texturas

2.1.2 Navegación

2.1.2.1 Tabla de Iluminación

2.1.2.2 Tabla de Animación Sonidos

2.1.2.3 Tabla de Modificación de ambiente

2.1.2.4 Tabla de Parámetros de cámara

2.1.2.5 Tabla de Diseño de colisiones

2.1.2.6 Tabla de modelos SHOCKWAVE3D

- 2.2 Diseño Arquitectónico
- 2.3 Diseño Procedimental
 - 2.3.1 Diagrama de Flujo
 - 2.3.2 Diagrama de Chapin
 - 2.3.3 Pseudocódigo
- 2.4 Diseño de la Interfase
 - 2.4.1 Diseño de las salas
 - 2.4.1.1 Ciudadela
 - 2.4.1.2 Calzada 1
 - 2.4.1.3 Calzada 2
 - 2.4.1.4 Plaza del Sol
 - 2.4.1.5 Mural del Puma
 - 2.4.1.6 Plaza de la Luna
 - 2.4.1.7 Conjunto de Palacios
 - 2.4.2 Menú principal
 - 2.4.2.1 Paseo Virtual
 - 2.4.2.2 Historia
 - 2.4.2.3 Fotos
 - 2.4.2.4 Juegos

3. DESARROLLO

- 3.1 De la Arquitectura
 - 3.1.1 Modelado
 - 3.1.2 Texturizado
- 3.2 De la Navegación
 - 3.2.1 Detección de Colisiones
 - 3.2.2 Manejo de Cámara
 - 3.2.3 Generación del Tiempo Virtual
 - 3.2.3.1 Animación de la Iluminación
 - 3.2.3.2 Animación del Sonido
 - 3.2.3.3 Generación del Ambiente

4. PRUEBAS

- 4.1 De la Arquitectura
- 4.2 De la Navegación

La etapa de Diseño es el proceso de aplicar distintas técnicas y principios con el propósito de definir un sistema con suficiente detalle como para permitir su implementación.

Dentro de la etapa de diseño, desarrollamos cuatro diferentes modelos de diseño, los cuales se enuncian a continuación:

- **Diseño de Datos.** Especifica las estructuras de datos necesarias para implementar el sistema.
- **Diseño Arquitectónico.** Define las relaciones entre los elementos estructurales (módulos) del programa.
- **Diseño Procedimental.** Transforma los elementos estructurales de la arquitectura del programa en una descripción procedimental.
- **Diseño de la Interfase.** Describe como se comunica el software consigo mismo, con los sistemas que operan con él y con los operadores que lo emplean.

A continuación desarrollamos cada uno de los modelos para la etapa de diseño, explicando cada uno de ellos a detalle y en el orden del listado anterior.

VI.3 DISEÑO DE DATOS

El modelo de diseño de datos, en nuestro caso particular fue dividido en dos partes principales, en primer lugar hablaremos del diseño de datos para el modelado geométrico, y en segundo término hablaremos del diseño de datos para la navegación.

Cada una de ellas está conformada de la siguiente manera:

- Modelado
 - Tabla de medidas
 - Tabla de referencias
 - Tabla de texturas

- Navegación
 - Tabla de Iluminación
 - Tabla de Animación Sonidos
 - Tabla de Modificación de ambiente
 - Tabla de Parámetros de cámara
 - Tabla de Diseño de colisiones
 - Tabla de modelos SHOCKWAVE3D

Para el caso del diseño de datos en la sección de modelado, es importante recalcar que para realizar los modelos, necesitamos en primera instancia las medidas de las edificaciones, y para obtener éstas necesitamos una fuente de información, en este caso las fotografías de la maqueta del Museo de Sitio de la Zona Arqueológica de Teotihuacan.

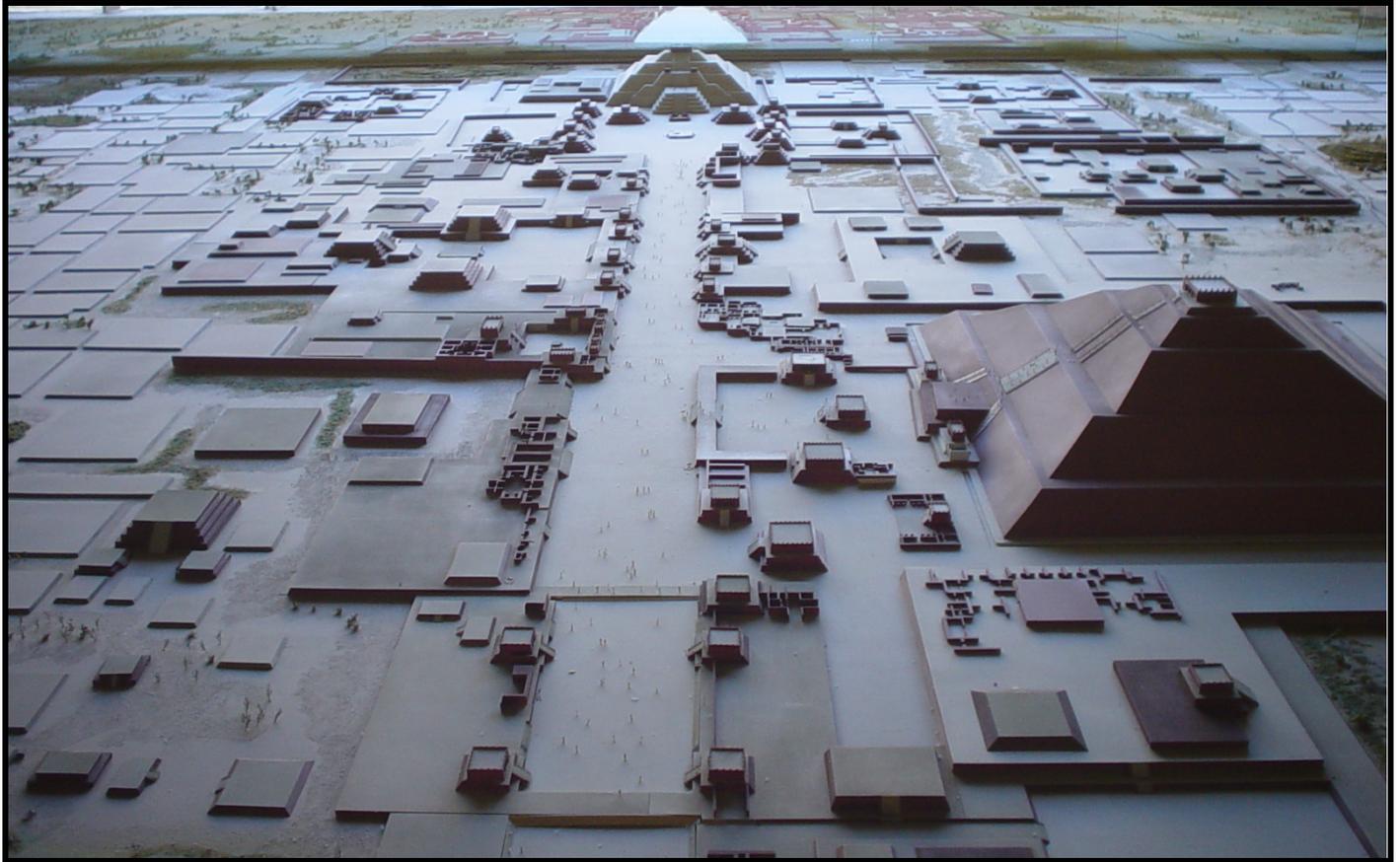


ILUSTRACIÓN VI.2

Maqueta del Museo de Sitio de Teotihuacan

Utilizando como referencia la maqueta del Museo de Sitio de Teotihuacan podemos obtener toda la información necesaria para realizar un modelo completo, con dimensiones, proporciones y ubicaciones, muy aproximadas a las reales.

La maqueta representa la zona principal de Teotihuacan, abarca la parte actual de la Calzada de los Muertos, que comprende desde la Ciudadela, pasando por la Plaza y Pirámide del Sol, hasta llegar a la Plaza y Pirámide de la Luna donde también se ubica el Conjunto de Palacios.

En la maqueta se pueden observar claramente los edificios principales, las edificaciones que se construyeron sobre la Calzada de los Muertos y las edificaciones que se asentaron alrededor de la misma.

Para obtener las medidas de los elementos de la zona arqueológica, la primera tarea consistió en tomar varias fotografías de la maqueta de tal manera que pudiéramos abarcar la mayor cantidad de información para obtener las dimensiones, proporciones y ubicaciones de los elementos a modelar.

Después de haber tomado las fotografías, el siguiente paso consiste en editar las imágenes que se utilizaran para la elaboración del modelado. Existen dos tipos de imágenes tomadas de la maqueta; imágenes para la obtención de medidas, e imágenes para la obtención de referencias.

Las características de los dos tipos de imágenes editadas, son las siguientes:

- **Imágenes editadas para obtener medidas.**

A partir de estas imágenes, se obtendrán las medidas para el modelado. Serán exportadas al programa AutoCad, donde se procesarán para obtener las medidas de los edificios en cuestión.

La característica principal de estas imágenes es que van a ser referencia a los edificios principales de la zona arqueológica, ya que estos edificios son los de mayor relevancia y atractivo en el paseo virtual, por lo cual requieren un análisis con mayor profundidad para buscar la exactitud en las medidas de dichos edificios, como son la Ciudadela la cual incluye el templo de Quetzalcoatl, la Plaza del Sol que incluye la Pirámide del mismo nombre y la Plaza de la Luna que incluyendo su pirámide.

- **Imágenes editadas para obtener referencias.**

A partir de estas imágenes, se obtendrán las referencias para el modelado.

La característica principal de estas imágenes es que van a ser referencia para ubicar los edificios secundarios de la zona arqueológica, ya que estos edificios son de menor relevancia en el paseo virtual, por lo cual no requieren un análisis con mayor profundidad, además porque dichas edificaciones abarcan zonas muy grandes.

El proceso de referenciación va a consistir en identificar que tipos de edificaciones se encuentran en cada una de las zonas de la Calzada de los Muertos y a partir de esta referencia se modelará cada una de las zonas, como son el primer nivel de la Calzada de los Muertos donde incluye el Puente del Río San Juan, el segundo nivel donde se ubica el Conjunto de los Edificios Superpuestos o el séptimo nivel donde se encuentra el Mural del Puma, por citar sólo algunos casos.

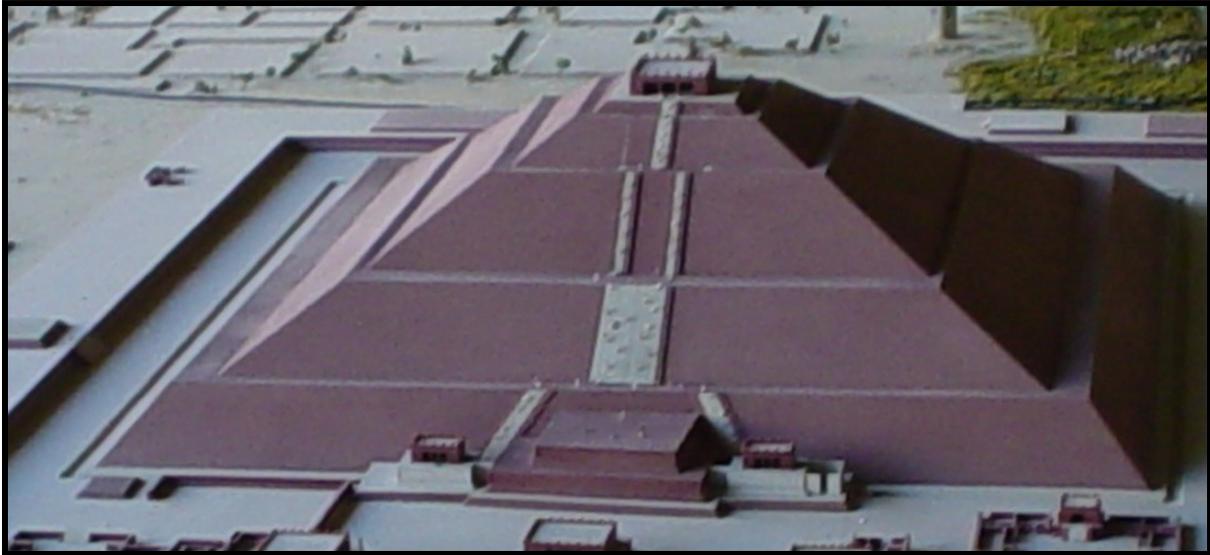


ILUSTRACIÓN VI.3
Fotografía editada para la obtención de medidas

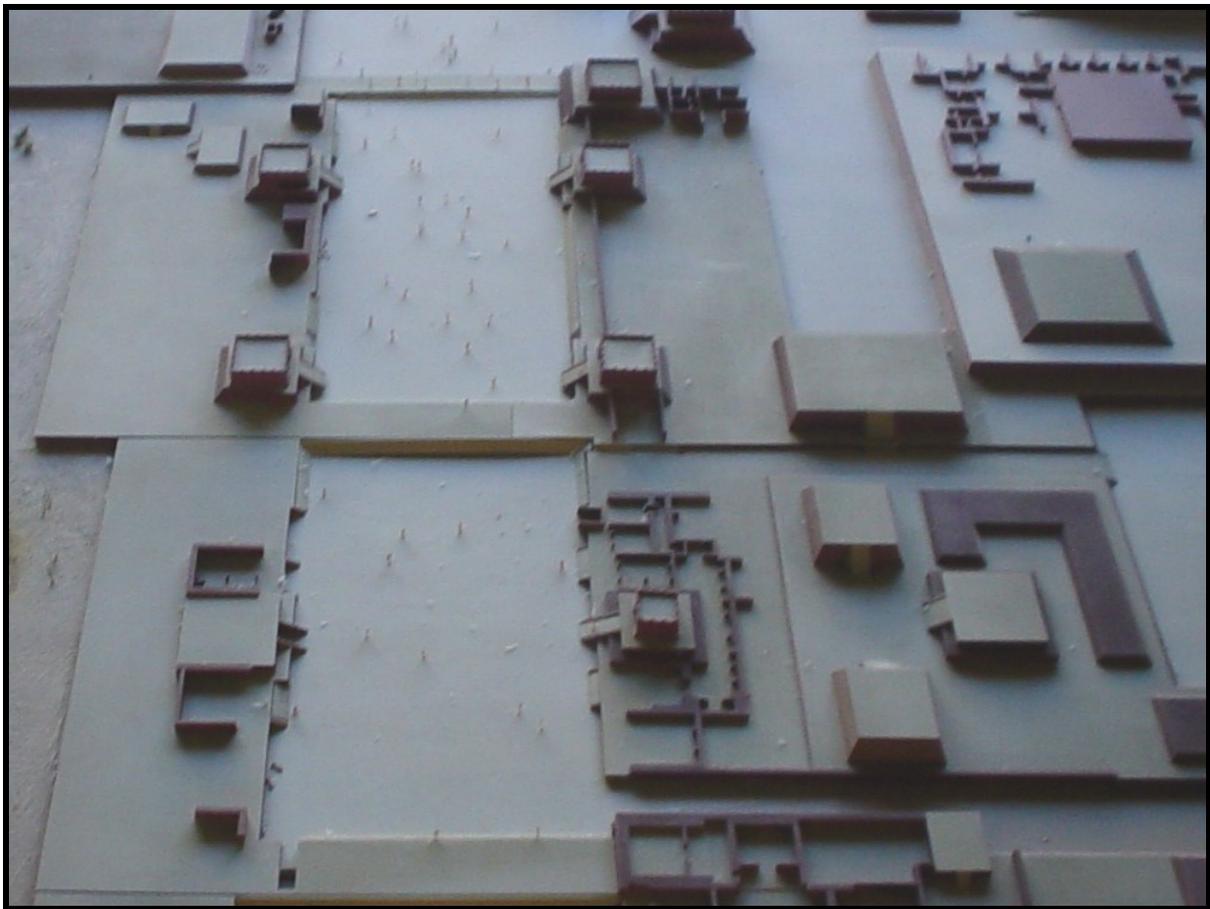


ILUSTRACIÓN VI.4
Fotografía editada para la obtención de referencias

VI.3.1 TABLA DE MEDIDAS

VI.3.1.1 OBTENCIÓN DE MEDIDAS

Las imágenes que utilizaremos para el proceso de medición, hacen referencia de manera precisa a cada uno de los edificios principales de la zona arqueológica. Una vez clasificadas las imágenes, se realizarán las mediciones correspondientes utilizando el software AutoCAD R14.

Las imágenes editadas previamente, van a ser importadas a este paquete, con la finalidad de medirlas, haciendo uso de la escala de AutoCAD.

Su utilización para obtener los resultados requeridos es muy sencilla ya que se manejan pocas herramientas y los resultados son casi inmediatos.

Dentro de AutoCAD, se utilizará la herramienta «Dimensión Linear» además de otras auxiliares. Se acotarán las dimensiones de las edificaciones que se desean medir, y dichas medidas que se vayan obteniendo aparecerán sobre las imágenes.

La escala de medidas que aparece en pantalla, es propia del editor de AutoCAD, aunque realmente no importa en que escala se obtengan dichas medidas, ya que únicamente nos interesa obtener las proporciones existentes entre cada uno de los elementos de las edificaciones para ser comparadas con las medidas reales. Posteriormente aplicaremos un factor de escalamiento para poder obtener las medidas reales de toda la edificación.

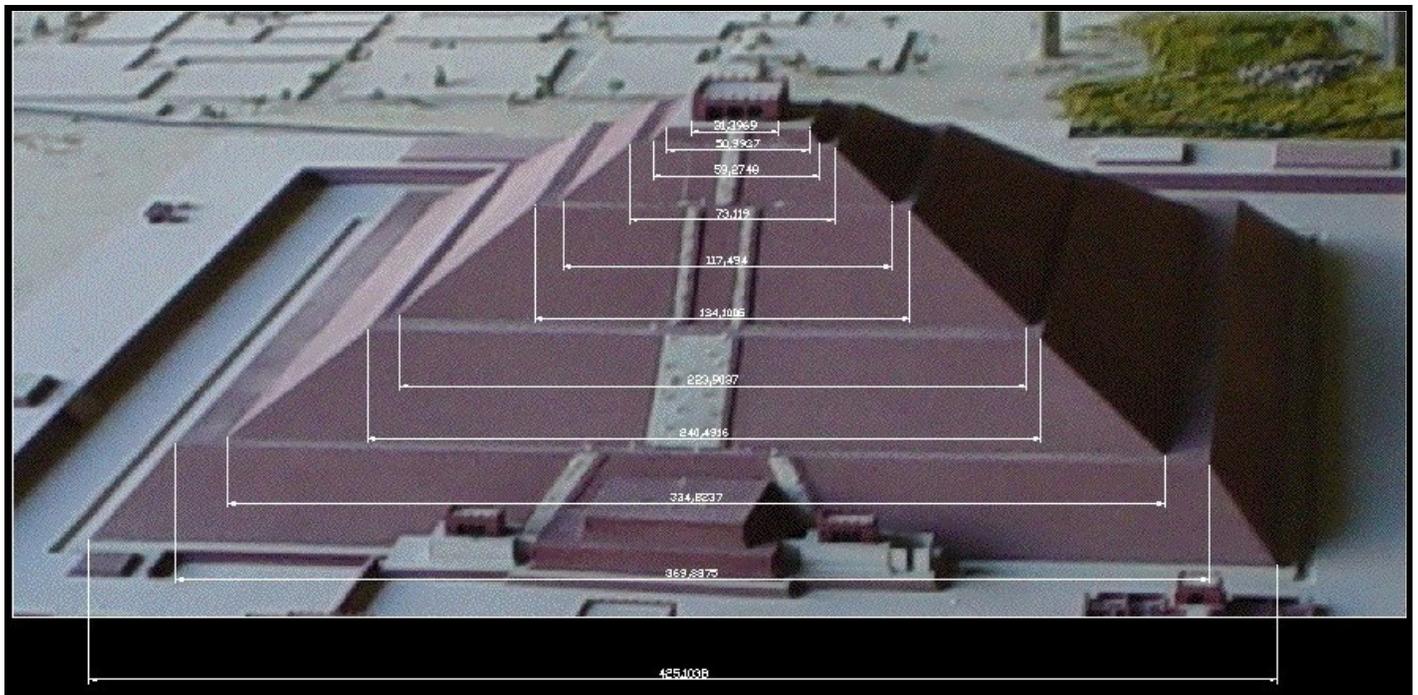


ILUSTRACIÓN VI.5
Obtención de medidas en AutoCad

VI.3.1.2 OBTENCIÓN DE LA HOJA DE CÁLCULO

Habiendo obtenido las medidas de los edificios principales, estas serán almacenadas en los archivos de AutoCAD, para posteriormente utilizar dicha información en una hoja de cálculo.

Lo anterior será realizado con la ayuda del paquete Microsoft Excel, en el cual diseñaremos de manera organizada una hoja de cálculo, que contendrá las medidas de las imágenes procesadas en AutoCAD, las medidas reales conocidas para las edificaciones principales, las operaciones y los cálculos necesarios para obtener las medidas reales que conforman la edificación.

Para obtener las medidas reales de un edificio principal, por ejemplo la Pirámide del Sol, partimos del conocimiento de una o más medidas del edificio Real, que al ser comparadas con la mismas medidas que arroja la fotografía de la maqueta al ser procesada en AutoCAD, podemos determinar el factor de escalamiento que afectará al resto de las medidas del archivo, para obtener estas últimas en dimensiones reales.

Cabe mencionar que la escala que utiliza AutoCAD está referenciada a la pantalla del editor, y por tal motivo se deben hacer ciertos ajustes de estas medidas que no son las que deseamos obtener, pero dicha información que contienen estos archivos son de gran utilidad para poder manipularlos de una manera eficaz, ya que las medidas son proporciones acotadas de cada una de las edificaciones, por lo que lo único que queda por realizar es un análisis de estas y obtener los cálculos correspondientes, para que de esta forma obtengamos las medidas reales.

Así que la idea en la que se basa este análisis es el entender que lo que se tiene son medidas proporcionales, por lo que únicamente hay que aplicar un factor de escalamiento, el cual sale de las medidas reales que tenemos de las edificaciones, y esta se compara con la medida que arroja el archivo con referencia a esa medida, y a partir de esta relación se obtiene dicho factor.

Con base en lo anterior, lo que procede a realizar es una hoja de calculo con ayuda del software Microsoft Exel 2000, donde diseñaremos de manera muy sencilla una hoja de calculo, la cual realiza las operaciones y cálculos necesarios para que a partir de la información de las imágenes medidas en AutoCAD, vaciemos la información que arrojan dichos archivos a la hoja de calculo y de acuerdo a lo especificado en ella, se determinarán las medidas reales de las edificaciones principales de la zona arqueológica, donde cabe resaltar que el diseño de dicha hoja no es complejo, solamente se obtienen los factores de escalamiento y a partir de ellos se realizan los cálculos de cada una de las medidas arrojadas para así obtener las reales.

La información existente en la hoja de cálculo, será de vital importancia en la etapa de desarrollo, ya que las medidas calculadas serán la fuente de información para desarrollar el modelado correspondiente. Las tablas generadas son utilizadas para almacenar las medidas de la Pirámide del Sol y su plaza, la Pirámide y laza de la Luna, Ciudadela y Templo de Quetzalcoatl y finalmente para la Calzada de los Muertos.

En el caso de la tabla referente a la Pirámide y Plaza del Sol, con base en las imágenes obtenidas de la maqueta generamos todos los datos, debido a que conocemos las medidas reales de la base de la pirámide, así como su altura. Dicha tabla de cálculo, no tuvo mayor complicación a la hora de realizar las operaciones y obtener los resultados.

MEDIDAS DE LA PIRAMIDE PRIMARIA DEL SOL			
FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
PRIMER NIVEL ANCHO	402	222	0,55
PRIMER NIVEL LARGO	387	225	0,58
ANCHO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	402	222	
SEGUNDO NIVEL	317	175	
TERCER NIVEL	213	118	
CUARTO NIVEL	112	62	
QUINTO NIVEL	56	31	
LARGO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	387	225	
SEGUNDO NIVEL	294	171	
TERCER NIVEL	190	110	
CUARTO NIVEL	98	57	
QUINTO NIVEL	47	27	
ANCHO SUPERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	349	193	
SEGUNDO NIVEL	227	125	
CUARTO NIVEL	69	38	
QUINTO NIVEL	47	26	
LARGO SUPERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	322	187	
SEGUNDO NIVEL	211	123	
TERCER NIVEL	112	65	
CUARTO NIVEL	58	34	
QUINTONIVEL	40	23	
FACTOR DE ESCALAMIENTO PARA LA INCLINACION (ANCHO)	ANCHO INFERIOR	ANCHO SUPERIOR	%ANCHO
PRIMER NIVEL	222	193	0,87
SEGUNDO NIVEL	175	125	0,72
TERCER NIVEL	118	71	0,60
CUARTO NIVEL	62	38	0,62
QUINTO NIVEL	31	26	0,84
FACTOR DE ESCALAMIENTO PARA LA INCLINACION (LARGO)	LARGO INFERIOR	LARGO SUPERIOR	%LARGO
PRIMER NIVEL	225	187	0,83
SEGUNDO NIVEL	171	123	0,72
TERCER NIVEL	110	65	0,59
CUARTO NIVEL	57	34	0,59
QUINTO NIVEL	27	23	0,85
FACTOR DE ESCALAMIENTO PARA LA INCLINACION	%ANCHO	%LARGO	%
PRIMER NIVEL	0,87	0,83	0,85
SEGUNDO NIVEL	0,72	0,72	0,72
TERCER NIVEL	0,60	0,59	0,60
CUARTO NIVEL	0,62	0,59	0,60
QUINTO NIVEL	0,84	0,85	0,85

FACTOR PARA OBTENER LA MEDIDA REAL (= MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ALTURA TOTAL DE LA PIRAMIDE	176	65	0,37

ALTURA	MEDIDA MAQUETA	MEDIDA REAL
PRIMER NIVEL	45	17
SEGUNDO NIVEL	49	18
TERCER NIVEL	51	19
CUARTO NIVEL	25	9
QUINTO NIVEL	6	2

TEMPLO	MEDIDA MAQUETA	MEDIDA REAL
ANCHO	29	16
LARGO	26	15
ALTURA	7	3

MEDIDAS DE LA PIRAMIDE SECUNDARIA DEL SOL

FACTOR PARA OBTENER LA MEDIDA REAL (= MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO PRIMER NIVEL PIRAMIDE PRIMARIA	402	222	0,55
ALTURA PRIMER NIVEL PIRAMIDE PRIMARIA	33	17	0,52

ANCHO	MEDIDA MAQUETA	MEDIDA REAL
ANCHO BASE RECTANGULAR	186	103
ANCHO BASE CUADRANGULAR	93	51
ANCHO PRIMER NIVEL PIRAMIDE SECUND.	79	44
ANCHO SEGUNDO NIVEL PIRAMIDE SECUND.	52	29
ANCHO BASE TEMPLO	28	15
ANCHO TEMPLO	18	10

ALTURA	MEDIDA MAQUETA	MEDIDA REAL
ALTURA BASE RECTANGULAR	3	2
ALTURA BASE CUADRANGULAR	3	2
ALTURA PRIMER NIVEL PIRAMIDE SECUND.	8	4
ALTURA SEGUNDO NIVEL PIRAMIDE SECUND.	9	5
ALTURA BASE TEMPLO	7	4
ALTURA TEMPLO	5	3

MEDIDAS DE LA PLAZA DEL SOL

FACTOR PARA OBTENER LA MEDIDA REAL (= MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA REAL	MEDIDA MAQUETA	%
ANCHO PRIMER NIVEL PIRAMIDE PRIMARIA	130	222	1,71
LARGO PRIMER NIVEL PIRAMIDE PRIMARIA	137	225	1,64

ANCHO	MEDIDA MAQUETA	MEDIDA REAL
ANCHO PLAZA	205	350
ANCHO HERRADURA	25	43
ANCHO SEPARACION HERRADURA/PIRAMIDE	15	26

LARGO	MEDIDA MAQUETA	MEDIDA REAL
LARGO PLAZA	260	427
LARGO HERRADURA	25	41
LARGO SEPARACION HERRADURA/PIRAMIDE	25	41

TABLA VI.1
Medidas para la Plaza y Pirámide del Sol

Para la tabla de la Pirámide y Plaza de la Luna, con base en las imágenes obtenidas de la maqueta podemos generar únicamente las medidas del ancho, ya que de las fotografías editadas no pudimos obtener una imagen lateral de la pirámide debido al acceso restringido del pasillo que cruza la maqueta. De esta forma las medidas del largo se obtuvieron con base en las medidas del ancho, además de conocer la medida real de la base de la Pirámide de la Luna.

MEDIDAS DE LA PIRAMIDE PRIMARIA DE LA LUNA			
FACTOR PARA OBTENER EL LARGO (=LARGO/ ANCHO)	MEDIDA REAL ANCHO	MEDIDA REAL LARGO	%
	150	140	0,93
FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
PRIMER NIVEL LARGO	347	150	0,43
ANCHO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	347	150	
SEGUNDO NIVEL	288	124	
TERCER NIVEL	202	87	
CUARTO NIVEL	117	51	
LARGO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	324	140	
SEGUNDO NIVEL	269	116	
TERCER NIVEL	189	81	
CUARTO NIVEL	109	47	
ANCHO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	303	131	
SEGUNDO NIVEL	221	96	
TERCER NIVEL	158	68	
CUARTO NIVEL	72	31	
FACTOR DE ESCALAMIENTO PARA LA INCLINACION (ANCHO)	ANCHO INFERIOR	ANCHO SUPERIOR	%ANCHO
PRIMER NIVEL	150	131	0,87
SEGUNDO NIVEL	124	96	0,77
TERCER NIVEL	87	68	0,78
CUARTO NIVEL	51	31	0,62
FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ALTURA TOTAL DE LA PIRAMIDE	108	45	0,42
ALTURA	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	31	13	
SEGUNDO NIVEL	36	15	
TERCER NIVEL	23	10	
CUARTO NIVEL	18	8	
TEMPLO	MEDIDA MAQUETA	MEDIDA REAL	
ANCHO	47	20	
LARGO	44	19	
ALTURA	13	5	

MEDIDAS DE LA PIRAMIDE SECUNDARIA DE LA LUNA			
---	--	--	--

FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO PRIMER NIVEL PIRAMIDE PRIMARIA	290	150	0,52
ALTURA PRIMER NIVEL PIRAMIDE PRIMARIA	102	45	0,44

ANCHO INFERIOR	MEDIDA MAQUETA	MEDIDA REAL
PRIMER NIVEL	215	111
SEGUNDO NIVEL	170	88
TERCER NIVEL	101	52
ANCHO SUPERIOR	MEDIDA MAQUETA	MEDIDA REAL
PRIMER NIVEL	182	94
SEGUNDO NIVEL	133	69
TERCER NIVEL	66	34

FACTOR DE ESCALAMIENTO PARA LA INCLINACION (ANCHO)	ANCHO INFERIOR	ANCHO SUPERIOR	%ANCHO
PRIMER NIVEL	111	94	0,85
SEGUNDO NIVEL	88	69	0,78
TERCER NIVEL	52	34	0,65

ALTURA	MEDIDA MAQUETA	MEDIDA REAL
PRIMER NIVEL	22	10
SEGUNDO NIVEL	18	8
TERCER NIVEL	23	10

MEDIDAS DE LA PIRAMIDE TERCEARIA DE LA LUNA			
--	--	--	--

FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO PRIMER NIVEL PIRAMIDE PRIMARIA	290	150	0,52
ALTURA PRIMER NIVEL PIRAMIDE PRIMARIA	102	45	0,44

ANCHO Y LARGO	MEDIDA MAQUETA	MEDIDA REAL
PRIMER NIVEL	107	55
SEGUNDO NIVEL	96	50
TERCER NIVEL	83	43
CUARTO NIVEL	72	37
QUINTO NIVEL	60	31
ALTURA	MEDIDA MAQUETA	MEDIDA MAQUETA
PRIMER NIVEL AL QUINTO NIVEL	8	4

MEDIDAS DE LA PLAZA DE LA LUNA			
---------------------------------------	--	--	--

FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO PRIMER NIVEL PIRAMIDE PRIMARIA	80	150	1,88
LARGO PRIMER NIVEL PIRAMIDE PRIMARIA	70	140	2,00

ANCHO	MEDIDA MAQUETA	MEDIDA REAL
ANCHO PLAZA	250	469
LARGO	MEDIDA MAQUETA	MEDIDA REAL
LARGO PLAZA	220	440

TABLA VI.2
Medidas para la Plaza y Pirámide de la Luna

Para el caso de la Pirámide de Quetzalcoatl, debido a que contamos con una imagen lateral, solo pudimos obtener las medidas del largo, pero esto fue solucionado ya que la ésta es cuadrangular.

Debido a que no teníamos la medida real para la base de la Pirámide, ésta tuvo que ser obtenida con base en el largo de la Ciudadela, realizando una comparación entre las medidas obtenidas de la foto para la plaza y la pirámide, y con esto generamos un factor de escalamiento.

MEDIDAS DE LA PIRAMIDE DE QUETZALCOATL			
FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO Y LARGO CIUDADELA	210	400	1.90
MEDIDA REAL DE LA PIRAMIDE	MEDIDA MAQUETA	MEDIDA REAL	
ANCHO Y LARGO PRIMER NIVEL PIRAMIDE (=MEDIDA REAL/ MEDIDA MAQUETA)	37	70	
PRIMER NIVEL ANCHO	70	257	0.27
ANCHO Y LARGO	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL	257	70	
SEGUNDO NIVEL	237	65	
TERCER NIVEL	212	58	
CUARTO NIVEL	187	51	
QUINTO NIVEL	158	43	
SEXTO NIVEL	133	36	
SEPTIMO NIVEL	105	29	
OCTAVO NIVEL	75	20	
(=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ALTURA TOTAL DE LA PIRAMIDE	110	25	0.23
ALTURA	MEDIDA MAQUETA	MEDIDA REAL	
PRIMER NIVEL AL SEPTIMO NIVEL	12	3	
OCTAVO NIVEL	6	1	
TEMPLO	MEDIDA MAQUETA	MEDIDA REAL	
ANCHO Y LARGO	61	17	
ALTURA	14	3	
MEDIDAS DE LA PLAZA DE LA CIUDADELA			
FACTOR PARA OBTENER LA MEDIDA REAL (=MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO Y LARGO PLAZA	210	400	1.90
ANCHO	MEDIDA MAQUETA	MEDIDA REAL	
ANCHO PLAZA	210	400	
ANCHO HERRADURA	23	44	
ANCHO HUECO PLAZA	123	234	
ANCHO HUECO CONJUNTOS SACERDOTALES	44	84	
ANCHO SEPARACION HERRADURA/PIRAMIDE	15	29	
LARGO	MEDIDA MAQUETA	MEDIDA REAL	
LARGO PLAZA	210	400	
LARGO HERRADURA	23	44	
LARGO HUECO PLAZA	33	63	
LARGO HUECO CONJUNTOS SACERDOTALES	105	200	
LARGO SEPARACION HERRADURA/PIRAMIDE	25	48	

TABLA VI.3
Medidas para la Ciudadela y Pirámide de Quetzalcoatl

Para construir la calzada de los muertos, se tomó en cuenta únicamente la sección que se conserva en nuestros días, es decir desde la ciudadela hasta la Plaza de la Luna, considerando una longitud de 1,700 metros para dicha calzada. Para obtener sus medidas, se tomó como referencia un plano de la zona arqueológica; para este caso las fotografías no son de utilidad en este caso debido a que la maqueta es muy grande y no se alcanza a abarcar toda la calzada.

Se tomaron en cuenta elementos como el puente del Río San Juan, los descansos de las escaleras que dividen cada uno de los diferentes niveles de la calzada y ocho niveles de la misma.

MEDIDAS DE LA CALZADA DE LOS MUERTOS			
FACTOR PARA OBTENER LA MEDIDA REAL (= MEDIDA REAL/ MEDIDA MAQUETA)	MEDIDA MAQUETA	MEDIDA REAL	%
ANCHO	6,2	40	6,45
ANCHO	10,4	65	6,25
LARGO	222,6	1700	7,64
ANCHO			
PRIMER NIVEL	10,4	65	
SEGUNDO NIVEL	6,2	40	
TERCER NIVEL	6,2	40	
CUARTO NIVEL	6,2	40	
QUINTO NIVEL	6,2	40	
SEXTO NIVEL	6,2	40	
SÉPTIMO NIVEL	6,2	40	
OCTAVO NIVEL	6,2	40	
LARGO			
PRIMER NIVEL	65,5	500	
SEGUNDO NIVEL	11,2	86	
TERCER NIVEL	16,1	123	
CUARTO NIVEL	10,6	81	
QUINTO NIVEL	16,5	126	
SEXTO NIVEL	11,3	86	
SÉPTIMO NIVEL	13,7	105	
OCTAVO NIVEL	77,7	593	
ESCALERAS INTERMEDIAS DE LOS NIVELES			
ANCHO	6,2	40	
LARGO	1,3	10	
PUNTE DEL RIO SAN JUAN			
ANCHO	4	26	
LARGO	2	15	

TABLA VI.4
Medidas de la Calzada de los Muertos

VI.3.2 TABLA DE REFERENCIAS

Como su nombre lo indica, esta tabla contendrá la información que nos arrojaron las fotografías de referencia. Su función será ubicar o “referenciar” a los edificios de menor importancia, que se ubican en los costados de la calzada de los muertos, sobre ésta misma, o formando parte de algún conjunto ceremonial de la zona.

La tabla de referencias deberá leerse como si estuviésemos caminando a lo largo de la calzada de los Muertos, es decir, de Sur a Norte, y dividida en dos partes; las secciones Este y Oeste.

Para su mejor comprensión fueron incluidos diferentes diagramas de referencia obtenidos de las fotografías de la maqueta. En dichos diagramas podemos ubicar espacialmente las pirámides que integran las diferentes secciones y conjuntos.

Finalmente mencionaremos que debido al gran número de pequeñas pirámides, éstas fueron clasificadas en 7 diferentes tipos, para facilitar el modelado de las diferentes secciones de la zona.

A continuación se presenta el diagrama de las zonas, la tabla de referencia, la simbología de la tabla y los diagramas correspondientes a cada una de las secciones de referencia.

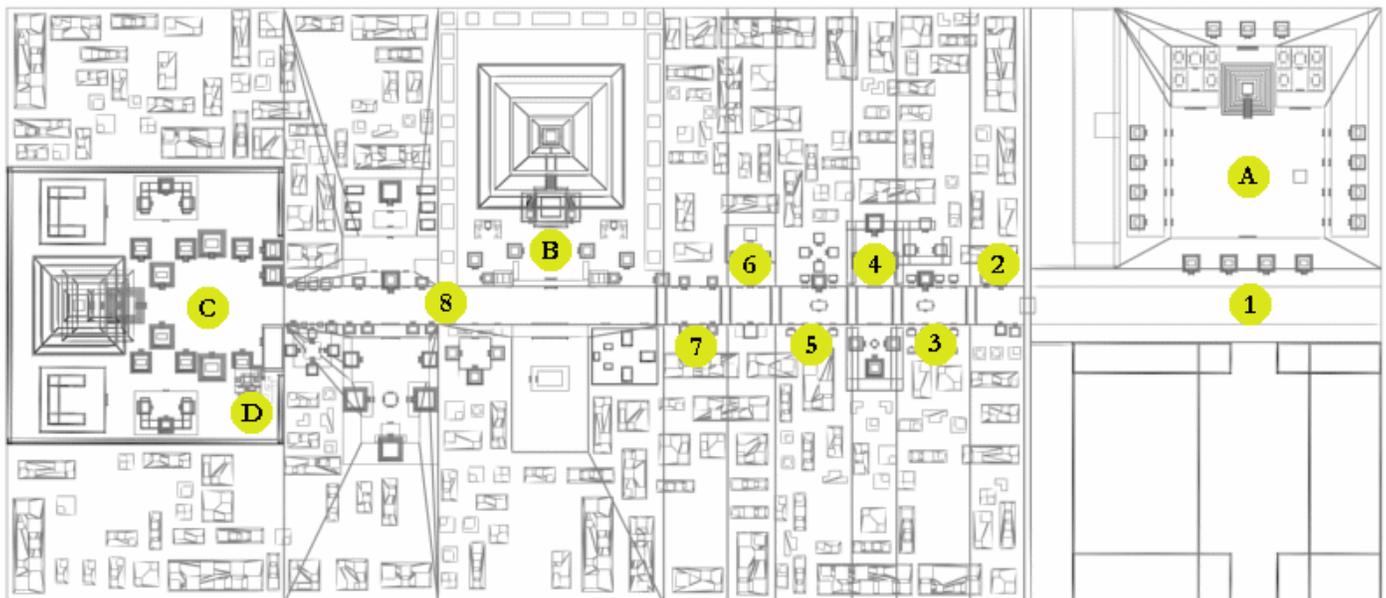


DIAGRAMA VI.1
Diagrama de zonas

TABLA DE REFERENCIA						
ZONA	OESTE	ESTE	NORTE	SUR	CENTRAL	DIAGRAMA
A	3C, 3C, 3C, 3C	2C, 2C, 2C	2C, 2C, 2C, 2C	2C, 2C, 2C, 2C	1C s/t c/e	A
B	-----	-----	2C+, 2C, 2C+, 1C	2C+, 2C, 2C+, 1C	1C	B
C	4R, 4R++, 4R, 4R+, 4R	4R, 4R++, 4R, 4R+, 4R	-----	1R++, 4R, 4R	1C s/t	C
1	Plataforma de acceso	Plataforma lateral	-----	-----	-----	E-1
2	3R+, 2C+	2C+, 3R+	-----	-----	-----	E-2
3	1R, 1R, 1R, 1R	1R, 1R, 4R, 1R	-----	-----	1R+ s/t	E-2
4	2C, 2C+, 2C	3C s/t, 4C s/t, 3C s/t	-----	-----	-----	E-2
5	1R, 1R, 1R, 1R	1C, 3R, 1C	-----	-----	1R+ s/t	E-3
6	2C+ s/t (altura a la mitad)	2C+	-----	-----	-----	E-3
7	2C, 2C	2C, 2C, 2C+	-----	-----	-----	E-3

TABLA VI.5
Tabla de referencia

SIMBOLOGÍA

A	CIUDADELA
B	PLAZA DEL SOL
C	PLAZA DE LA LUNA
D	CONJUNTO DE PALACIOS
1	PRIMER NIVEL DE LA CALZADA DE LOS MUERTOS
2	SEGUNDO NIVEL DE LA CALZADA DE LOS MUERTOS
3	TERCER NIVEL DE LA CALZADA DE LOS MUERTOS
4	CUARTO NIVEL DE LA CALZADA DE LOS MUERTOS
5	QUINTO NIVEL DE LA CALZADA DE LOS MUERTOS
6	SEXTO NIVEL DE LA CALZADA DE LOS MUERTOS
7	SÉPTIMO NIVEL DE LA CALZADA DE LOS MUERTOS
8	OCTAVO NIVEL DE LA CALZADA DE LOS MUERTOS
1C	PIRÁMIDE DE UN NIVEL CUADRANGULAR
1R	PIRÁMIDE DE UN NIVEL RECTANGULAR
2C	PIRÁMIDE DE DOS NIVELES CUADRANGULAR
3C	PIRÁMIDE DE TRES NIVELES CUADRANGULAR
3R	PIRÁMIDE DE TRES NIVELES RECTANGULAR
4C	PIRÁMIDE DE CUATRO NIVELES CUADRANGULAR
4R	PIRÁMIDE DE CUATRO NIVELES RECTANGULAR
s/t	SIN TEMPLO
+	AUMENTA EL TAMAÑO PROPORCIONALMENTE (ANCHO Y LARGO)

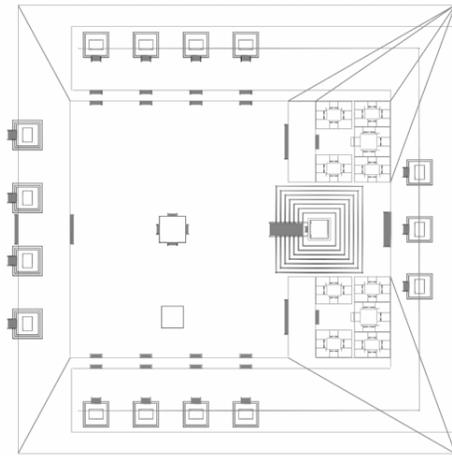


DIAGRAMA VI.2
Diagrama A

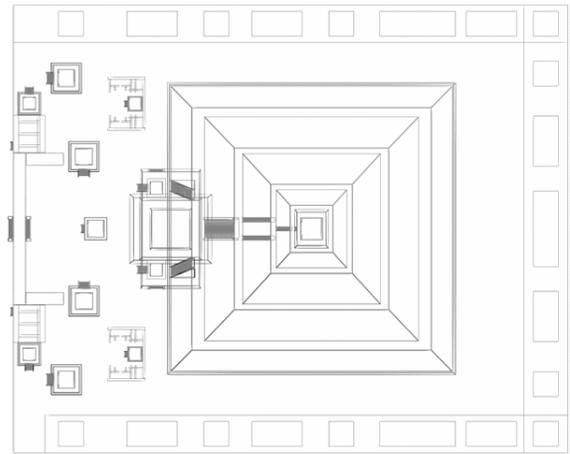


DIAGRAMA VI.3
Diagrama B

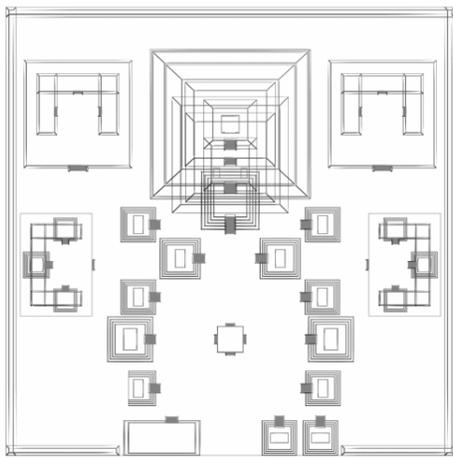


DIAGRAMA VI.4
Diagrama C

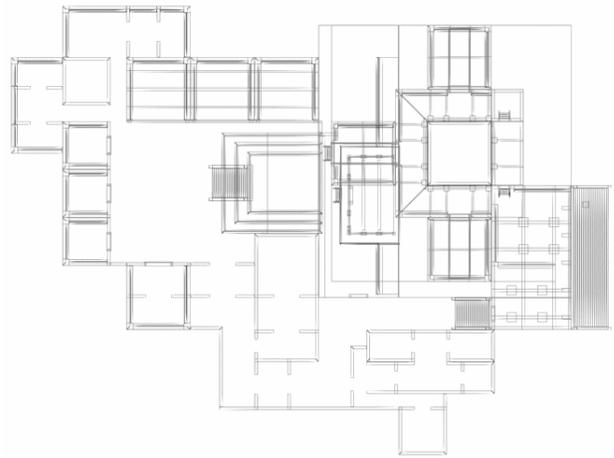


DIAGRAMA VI.5
Diagrama D

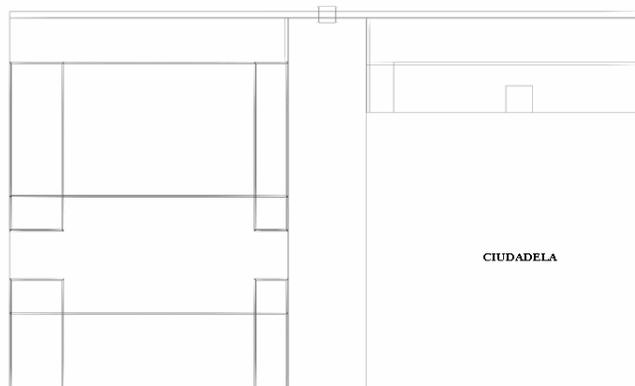


DIAGRAMA VI.6
Diagrama E-1

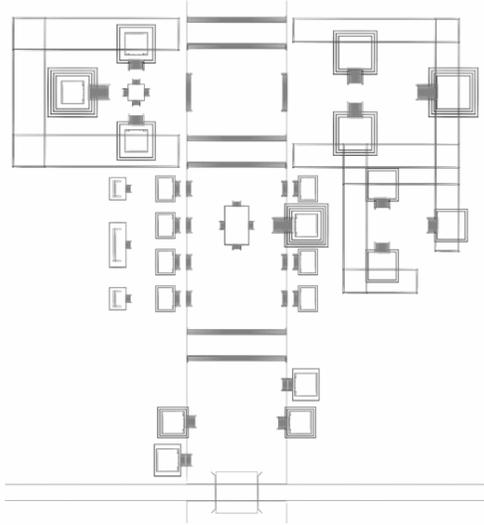


DIAGRAMA VI.7
Diagrama E-2

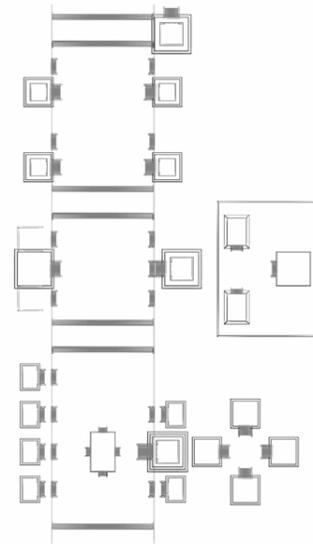


DIAGRAMA VI.8
Diagrama E-3

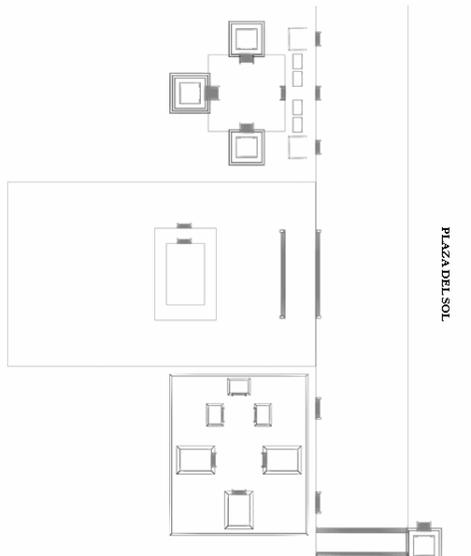


DIAGRAMA VI.7
Diagrama E-4



DIAGRAMA VI.8
Diagrama E-5

VI.3.3 TABLA DE TEXTURAS

Cada uno de los elementos modelados, debe ser texturizado completamente. Debido a que cada uno de los objetos que conforman el mundo virtual contiene más de una textura, es necesario diseñar una Tabla de Texturas. Dicha tabla contiene únicamente el nombre del objeto modelado y las texturas que en él intervienen. En la tabla no serán incluidas las referencias de las texturas hacia las caras del objeto en cuestión.

Cabe mencionar que las texturas incluidas en el PV fueron tomadas de murales y grifos del periodo Teotihuacano, pero sus ubicaciones no corresponden a las que tenían en aquella época. Éste problema surge debido a que en la actualidad se conservan muy pocos murales dentro de la zona arqueológica. Los únicos murales ubicados correctamente son: El mural del Puma, los glifos pintados en los tableros de las pirámides de la ciudadela, los jaguares ubicados en el patio del mismo nombre, las columnas y tablero superior del palacio de Quetzalpapalotl, los frescos pintados en los taludes de la antesala del mismo palacio, y algunos otros más.

La mayor parte de las imágenes utilizadas para texturizar los modelos, utilizan el formato BMP con una profundidad de 16 bits, y sin exceder los 256 píxeles de tamaño para el ancho y largo. El resto de las imágenes utilizan el formato PNG, este formato es muy útil, ya que al permitir transparencia podemos simular el modelado de objetos complejos, como se hizo para las almenas. Como podemos observar el tamaño de las imágenes y su profundidad, pueden ser consideradas como de una resolución media, debido a que en el PV se requiere la eficiencia y un alto desempeño, y esto se puede lograr simplificando la geometría y comprimiendo las imágenes.

A continuación presentamos un pequeño ejemplo que incluye la tabla de texturas para una pirámide de la ciudadela:



ILUSTRACIÓN VI.6
Modelo texturizado

Nombre del Objeto: ct3

Tabla de Texturas

ID	Nombre	Extensión
1	Estuco Rojo	BMP
2	Estuco Blanco	BMP
3	Estuco Plata	BMP
7	Estuco Amarillo	BMP
34	Ciudadela Glifo 01	BMP
75	AlmenaPng	PNG

TABLA VI.6
Tabla de Texturas

VI.3.4 DISEÑO PARA LA DETECCIÓN DE COLISIONES

La colisión o “choque” es necesaria para la realización de un buen paseo virtual, sin ésta traspasaríamos los objetos desapareciendo éstos de la vista u ocurriendo efectos de corte, tampoco podríamos subir, bajar, o trasladarnos de un lugar a otro de forma natural. Por esto la importancia de un buen diseño para la detección de colisiones.

El algoritmo utilizado pertenece a Paul Catanese⁷ el cual considera únicamente las colisiones frontales, cuyo movimiento es restringido a un solo plano. Modificamos el algoritmo de tal forma que se pudieran detectar colisiones frontales y traseras. Otra ventaja que obtenemos al modificar el algoritmo es que el movimiento a lo largo del eje de las cotas, no está restringido, es decir, podemos subir o bajar libremente a través de cualquier objeto que así lo permita.

Lo primero que tenemos que hacer es generar un cilindro, el cual nos servirá como objeto padre en la jerarquía de objetos y a quien se le asignarán las transformaciones geométricas de rotación y traslación; los demás nodos generados serán hijos del cilindro y por lo tanto las transformaciones geométricas que se le apliquen a este último afectarán igualmente a todos y cada uno de los nodos hijos. Como nodos hijos tenemos; seis cajas (cubos), dos planos de seguridad y una cámara. Nuestro método para la detección de colisiones funciona cuando el usuario navega hacia adelante y hacia atrás; aunque únicamente explicaremos cuando se detecta una colisión frontal, debido a que el método de detección trasera es el mismo.

El método simplificado (para la detección de colisiones frontales) estará formado por los siguientes objetos: el cilindro, tres cajas, el plano de seguridad frontal y la cámara. En la siguiente figura se muestra la jerarquía para los objetos generados en tiempo real para la detección de colisiones en el mundo virtual.



ILUSTRACIÓN VI.7
Jerarquía de los objetos del mundo virtual

La función de las cajas será indicar el punto de partida para la emisión de los vectores de colisión a quienes llamaremos “rayo derecho”, “rayo izquierdo” y “rayo superior”.

Los rayos, izquierdo y derecho servirán para detectar las colisiones con todos los objetos, tales como bardas, escaleras, columnas, etc.; mientras que el rayo superior servirá únicamente para hacer el ajuste de altura del cilindro y por ende de todos los nodos.

⁷ Catanese Paul, “Director 8.5. Fundamentos de la Programación 3D.”, Editorial Anaya Multimedia, 2002, Pág. 824-827.

La forma en la que el sistema detecta las colisiones, es la siguiente:

No importando la posición del cilindro, podemos calcular en cualquier instante de tiempo la posición de las tres cajas desde las cuales se “disparan” los vectores de colisión, la posición del plano de seguridad es actualizada también en tiempo real. La distancia del plano de seguridad respecto al cilindro es directamente proporcional a la velocidad de este, es decir, que mientras más rápido vaya el cilindro mayor será la distancia que guarde el plano de seguridad respecto al cilindro, para poder frenar totalmente al momento de colisionar.

Suponiendo que el cilindro tiene una velocidad diferente de cero, podemos decir que ocurre una colisión cuando un objeto se encuentra entre las cajas derecha o izquierda y el plano de seguridad; en ese momento el cilindro comienza a desacelerar para cuando el cilindro se encuentre justo en el objeto, su velocidad sea igual a cero y evitar que traspase el objeto colisionado.

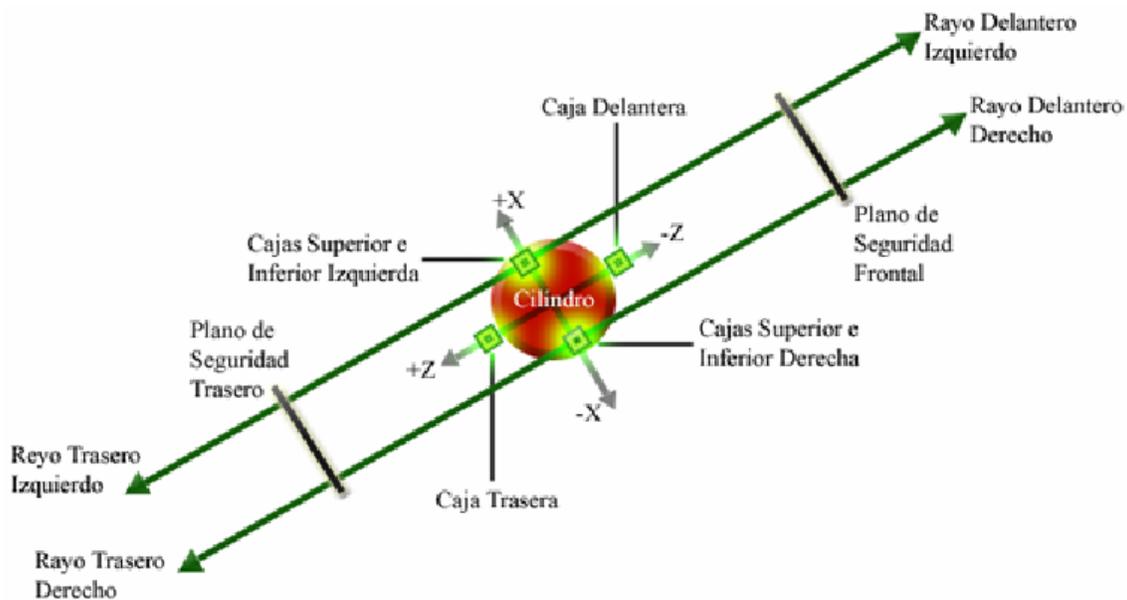


ILUSTRACIÓN VI.8

Vista Superior para la detección de colisiones

Los vectores de colisión fueron generados con el comando `ModelsUnderRay()`, a partir de la posición de las tres cajas (derecha, izquierda y frontal). A continuación explicamos la forma de utilizar la función, así como sus parámetros de entrada y de salida:

Sintaxis:

```
member(modelo 3D).modelsUnderRay(Vector de posición de inicio, Vector director {,
Numero de modelos a listar, Nivel de detalle})
```

Descripción:

Devuelve una lista con los modelos colisionados por los vectores derecho e izquierdo, a partir del vector de posición inicial (a partir de la posición de las cajas), en dirección del vector director.

El primero de los dos siguientes parámetros opcionales (Número de modelos a listar), lo fijaremos en “uno”, ya que sólo requerimos la información del último modelo colisionado. Al segundo parámetro opcional de la función (Nivel de detalle), podemos asignarle dos valores:

`#simple`.- Regresa una lista conteniendo únicamente el nombre de los modelos colisionados. Es la asignación por default.

`#detailed`.- Regresa una lista de los objetos colisionados, con sus propiedades que a continuación se enuncian:

- `#model`.- Regresa el nombre del modelo colisionado, que es el mismo nombre que asignamos al objeto en el momento de su creación dentro de 3D Studio.
- `#distance`.- Es la distancia desde el punto indicado por el vector de posición de inicio, hasta el punto de intersección con el modelo.
- `#isectNormal`.- Es el vector normal al objeto colisionado, en el punto de intersección con el vector de colisión.
- `#meshID`.- Es el ID (número de identificación) de la malla intersectada.
- `#faceID`.- Es el ID de la cara intersectada, que a su vez es parte de la malla intersectada.
- `#vertices`.- Son 3 puntos que representan de forma única el plano o cara intersectada.
- `#uvCoord`.- Propiedad que indica las coordenadas del mapeo UVW.

En el caso que no encuentre ninguna intersección, la función regresará una lista vacía. La opción utilizada en el nivel de detalle es `#detailed` y las propiedades que utilizamos dentro de esta opción son, `distance` e `isectNormal`, la primera para realizar el frenado y la segunda para calcular el ángulo de choque con el objeto y saber si se refracta o se detiene.

Para obtener el ángulo de la colisión, realizaremos el producto punto entre el vector director y el vector normal al objeto colisionado.

$$\theta = (\mathbf{A} \cdot \mathbf{B}) \times (180^\circ / \pi)$$

A: Vector Director

B: Vector normal del objeto colisionado

Como el resultado lo queremos en grados y Director maneja los ángulos en radianes, multiplicamos por el factor $180^\circ / \pi$.

Si el ángulo de refracción (θ), es mayor a 26° o menor a -26° (se considera este intervalo, debido a que así lo establece el algoritmo realizado por Paul Catanese) entonces se produce un desvío o refracción del cilindro hasta que la trayectoria del mismo es paralela al objeto colisionado, el cilindro no se detiene únicamente se produce un ligero frenado.

En caso contrario el cilindro se desacelerará por completo, si la trayectoria del cilindro no cambia. La siguiente figura muestra la colisión entre el muro y el cilindro debido a que el muro se encuentra entre las cajas derecha e izquierda y el plano de seguridad, además de mostrar el ángulo de refracción.

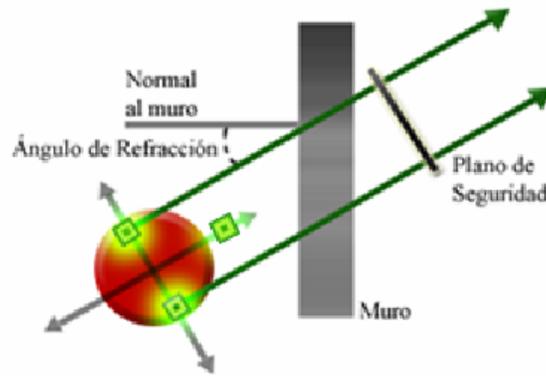
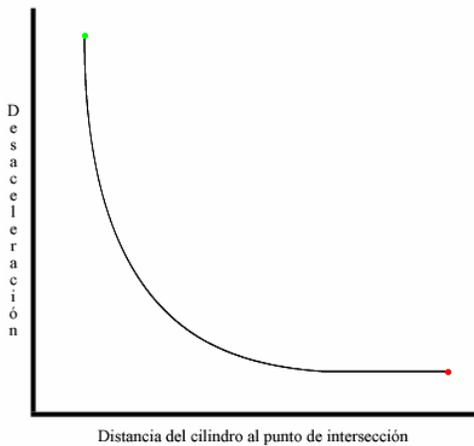


ILUSTRACIÓN VI.9
Colisión y Refracción del cilindro



Existen tres factores que determinan cuando se aplica la desaceleración del cilindro.

El primero de ellos es que exista una colisión, el segundo es que el ángulo de colisión sea menor a 52° o mayor a -52° .

El tercer factor es la distancia entre el objeto colisionado y el cilindro, mientras más cerca del objeto se encuentre mayor será la desaceleración que sufra.

Hasta este momento hemos trabajado la detección de colisiones en el plano XZ del modelo virtual, es decir, que no podemos subir o bajar en dirección del eje Y; para esto utilizamos la caja frontal de la cuál parte el rayo superior que calcula la distancia entre el centroide del cilindro y el plano

principal XZ del mundo virtual por medio de la siguiente formula:

$$\text{Ajuste vertical} = 23 - \text{Distancia de la caja frontal al piso}$$

El cilindro mide 40 unidades en altura, el ajuste es referenciado al centroide del cilindro, por lo que dividimos entre dos la altura sumándole a esta tres unidades, debido a que el plano principal se encuentra en $Y=3$.

Limitantes del diseño para la detección de colisiones

- Cuando alguna de las dimensiones de un objeto del mundo virtual es menor a la distancia entre los rayos derecho e izquierdo, el método de detección de colisiones no detecta dicho objeto, y por lo tanto lo traspasa.
- Cuando se detecta una esquina, se produce un movimiento oscilatorio debido a que tanto la caja izquierda como la caja derecha están detectando una colisión, aclarando que simultáneamente no se detectan ambas colisiones; lo que ocurre es que se detecta una sola colisión produciéndose una refracción, y debido a la cercanía de la otra cara del objeto, inmediatamente se detecta la colisión con dicha cara, ocurriendo nuevamente una refracción, y así sucesivamente.

VI.3.5 DISEÑO DE LA ILUMINACIÓN

Aunque al exportar nuestros modelos de 3D Studio Max hacia Shockwave 3D, se nos permita exportar la iluminación, no lo haremos, ya que son traducidas incorrectamente, y por lo tanto los resultados obtenidos no son los deseados. Lo que hicimos en primer lugar fue borrar la luz que por default 3D Studio Max coloca a los escenarios para su visualización en los puertos de vista y en el render.

Tenemos obligatoriamente que borrarla ya que dicho nodo genera un aumento en la iluminación y la alteración en el sombreado de las caras de los objetos.

A continuación mencionaremos los tipos de luz existentes en director, para saber que tipo de luz podemos utilizar, y evitar una sobrecarga en el proyecto además de lograr el mejor desempeño en tiempo real.

TIPOS DE LUZ EN DIRECTOR 8.5

El propósito de cada nodo Light se puede dividir en tres categorías: iluminación modelaje y enfoque. Sin luces en el mundo en 3D, no verá nada en sus escenas. El propósito de la iluminación es permitir a los usuarios ver el entorno, mientras que los objetivos del modelaje y del enfoque son tanto revelar la geometría como ocultarla. Debido a que el único propósito de la iluminación es revelar el entorno, cuanto más modelos se vean afectados mejor.

El segundo objetivo de las luces es utilizar los nodos Light para modelar y describir las cualidades geométricas de los modelos y las relaciones espaciales dentro de la escena. Además utilizar luces para modelar puede también establecer el momento del día de la escena. La geometría de los modelos que no se han modelado con luz dará la apariencia de ser lisa, incluso aunque la geometría sea compleja.

El término modelaje tiene dos significados en el desarrollo en 3D que deben ser aclarados: en el campo de la geometría, utilizamos el término modelaje para describir el proceso de creación de modelos con software de modelaje en 3D. En el campo de diseño de luces, modelaje se refiere a la manera en la que se pueden utilizar las luces para definir las zonas de su composición que tengan más relevancia.

El elemento de composición principal que alimenta el enfoque, es el contraste. Mediante la iluminación selectiva de parte de un determinado modelo o escena, puede dirigir la vista del espectador hacia elementos específicos que quiere que vea.

La vista tiende a ser dividida en áreas de contraste y esto se puede utilizar par definir las áreas más relevantes de la composición.

Existen 4 tipos diferentes de nodos Light que utiliza Director cada uno de ellos con características propias. Teniendo presente los tres objetivos de una buena iluminación, es momento de explicar los cuatro tipos de nodo Light: #ambient, #directional, #point y #spot.

- Nodo Light #ambient (luz ambiental)

El tipo de nodo Light #ambient representa la luz reflejada dentro de una escena. En el mundo físico, las superficies absorben y reflejan la luz que los golpea. La luz que no se absorbe se refleja y viaja a otras superficies para ser absorbida o reflejada hasta que, no queda nada de luz.

En la computadora, sólo podemos simular cualidades específicas de luz. Los cálculos de múltiples íter reflejos de luz en un determinado entorno es una pérdida de tiempo. Por tanto conseguiremos aproximarnos a la luz reflejada que contribuye a la iluminación de los modelos en el mundo real mediante el tipo Light #ambient.

Dicha fuente de luz se encuentra presente en todo el mundo virtual y afecta a cada uno de los modelos. Carece de posición y dirección, por lo que su único parámetro será el color que emite este tipo de nodo.

- Nodo Light #directional

El tipo de Light #directional se compara frecuentemente con la luz solar. La razón para esta comparación es la manera en que la luz solar llega a la tierra. Debido a que el sol es tan grande y está tan lejano, los rayos de la luz solar que llegan a un punto determinado de la tierra tienden a ser casi completamente paralelos. Los rayos de un Light #directional son totalmente paralelos y afectan a la escena de una manera uniforme.

Desde el punto de vista técnico, un Light #directional reacciona con cualquier superficie que se encuentra situada en la dirección de la luz, independientemente de la posición de estas superficies en la escena.

En resumen cada modelo de su escena, sin tener en cuenta la distancia a la que se encuentre, será iluminado de la misma manera por un determinado Light #directional.

La cantidad de luz que recibe cada objeto de la escena depende únicamente de su orientación, mientras el ángulo que forme el vector normal de una cara del objeto con el vector director de la luz sea cercano a 90° dicha cara recibirá muy poca iluminación, si es igual a 90° , no recibirá iluminación; en cambio si el ángulo es cercano o igual a 180° , recibirá mucha luz o el máximo posible, respectivamente.

La luz direccional cumple con la definición de las tres áreas claves en los modelos: luces, tonos medios y sombras.

Como en la luz direccional no nos importa la posición del nodo emisor de luz ni la distancia a la que se encuentran los objetos respecto al nodo #directional, de esta forma los únicos parámetros que se asignan son: el color en RGB, y la dirección de la luz, especificada por su vector de rotación.

- Nodo Light #point

Un nodo Light #point se puede comparar con una simple bombilla de luz; rayos de luz emanan de la bombilla en todas direcciones. Los nodos Light #point tienen tres propiedades definitorias: Color, Specular y Posición. Debido a que la luz emana en todas direcciones desde un Light #point, la propiedad fundamental que lo define es la posición. El nodo Light #point afectará a cualquier superficie de una escena que se encuentre situada en la dirección de la luz emitida.

- Nodo Light #spot

Principalmente se utilizan para centrar la atención en una zona específica de la escena o para resaltar las cualidades escultóricas de una superficie. Usando correctamente #spot se puede añadir dinamismo y dimensión a una escena. Usado en exceso puede provocar que la computadora se quede bloqueado. Las propiedades que definen al Light #spot son color, rotación, posición, Specular, ángulo de foco y caída de foco.

En el caso del Light #spot, tanto la posición como la rotación desempeñan un papel extremadamente importante. Concretamente un Light #spot tiene una posición en la escena, pero también tiene que ser girado para apuntar hacia los modelos que quiere iluminar.

El nodo Light #spot funciona como una lámpara de mano, nosotros tenemos que apuntar hacia los objetos que queremos iluminar, para esto es importante la posición donde se encuentre la lámpara y la rotación que le hayamos dado.

APLICACIÓN DE LA ILUMINACIÓN Y DE SOMBRAS

Para llevar a cabo con éxito el diseño de la iluminación es necesario utilizar combinaciones de nodos Light; en nuestro caso se utilizó un nodo tipo ambiental y un nodo tipo direccional; la luz direccional al funcionar de manera similar al sol nos permite iluminar con luz y sombra todos los modelos existentes en el mundo virtual, sin embargo las regiones en sombra respecto a las regiones iluminadas contrastan mucho; combinando el parámetro #color de la luz direccional con el mismo parámetro de la luz ambiental podemos generar una iluminación realista y poco pesada para el microprocesador.

La única limitante de la iluminación generada es que no produce sombras, dichas sombras tienen que ser generadas como imágenes en formato PNG, con un canal alfa de transparencia. De esta forma tendremos una buena iluminación con sombras, aunque por la animación de luces que generamos simulando el recorrido de la luz solar durante el día y la lunar durante la noche, las sombras tendrían que cambiar de posición y forma lo cual no ocurre.

Hay que recordar que para que las imágenes PNG se vean transparentes, tenemos que habilitar el render de Director para que detecte el canal alfa de las imágenes, de la siguiente manera:

```
getRendererServices().textureRenderFormat = #rgba8888
```

Dicha instrucción nos brinda 8 bits para el canal alfa, es decir 256 opciones de transparencia, donde el 0 es totalmente transparente y el 255 totalmente opaco.

Para obtener las sombras, en 3D Studio Max generamos un archivo que contiene la arquitectura de toda la zona arqueológica, y añadiremos 2 luces; una tipo ambiental y la segunda tipo Omnidireccional que proyecte sombras y del tipo “Area Shadows”; ubicamos las luces en el lado Sur Oeste de tal forma que la iluminación generada sea similar a la iluminación programada en Lingo 3D, posteriormente elegimos la vista superior y enfocamos el edificio del cual queremos obtener su sombra, renderizamos y guardamos la imagen en cualquier formato.

Una vez obtenida la imagen utilizaremos PhotoShop para separar la sombra de los demás objetos de la imagen, usamos la imagen de 3D Studio Max como fondo (background) y añadimos una capa (layer) transparente sobre el fondo para poder dibujar únicamente la sombra.

Dicho Layer lo grabamos como una imagen independiente en formato PNG, y es así como obtenemos la sombra de cualquier objeto dentro de nuestro mundo virtual.

Hasta este momento tenemos completa la iluminación y la proyección de sombras, el siguiente paso es animar la iluminación para simular el paso del día.

ANIMACIÓN DE LA ILUMINACIÓN

Para realizar la animación de la iluminación, definimos de manera arbitraria, la ruta que seguirá la luz a través de las 24 horas del día. La ruta será igual a la rotación del vector director del nodo #directional. La rotación se hace tomando como referencia los ejes X y Y, tal como se muestra en la figura:

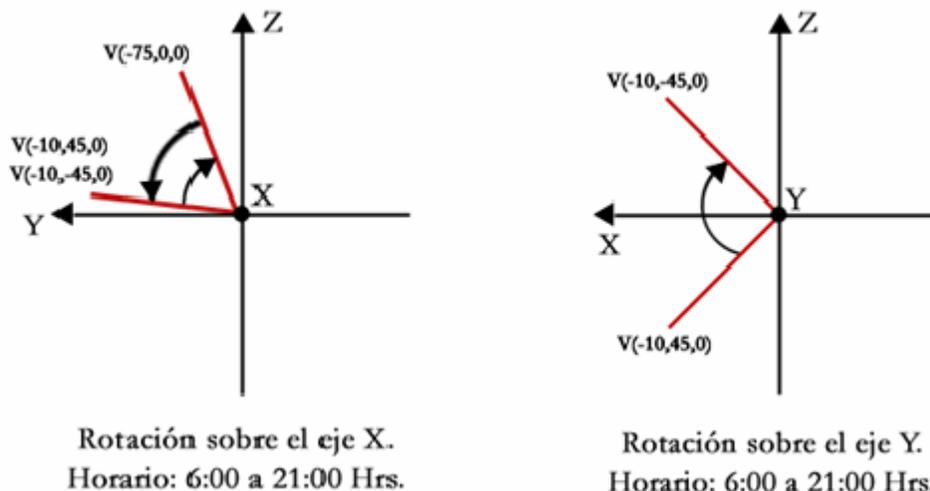


ILUSTRACIÓN VI.10
Graficas de Rotación para la luz direccional.

Además cambiamos el color de la luz ambiental y direccional para simular el momento del día (por ejemplo una tonalidad rojiza para simular el amanecer o una tonalidad azul para la noche).

Para obtener la dirección del nodo #directional cada minuto, en primer lugar necesitamos dividir el día en cuatro intervalos, 6:00-13:30, 13:30-21:00, 21:00-1:00, 1:00-6:00. Posteriormente calculamos los incrementos en cada una de las componentes del vector de rotación. Igualmente para obtener el color de la luz debemos calcular el incremento por minuto de cada una de las componentes (RGB) de los nodos, #directional y #ambient, para cada uno de los cuatro intervalos.

Los cálculos de los incrementos en los intervalos del día se presentan a continuación.

Incrementos para el intervalo de tiempo de las **6:00** a las **13:30** horas:

Cálculo del vector de rotación de la luz direccional (ΔR):

$$\Delta x = 130^\circ / (15 \times 60) = 0.1444 \text{ [}^\circ/\text{min]}$$

$$\Delta y = 90^\circ / (15 \times 60) = 0.1 \text{ [}^\circ/\text{min]}$$

$$\Delta z = 0 \text{ [}^\circ/\text{min]}$$

Cálculo del color de la luz direccional (ΔD):

$$\Delta r = (225-175) / (7.5 \times 60) = 0.11111 \text{ [Unidades/min]}$$

$$\Delta g = 0 \text{ [Unidades/min]}$$

$$\Delta b = (210-175) / (7.5 \times 60) = 0.07778 \text{ [Unidades /min]}$$

Cálculo del color de la luz ambiental (ΔA):

$$\Delta r = (240-210) / (7.5 \times 60) = 0.06667 \text{ [Unidades /min]}$$

$$\Delta g = (240-140) / (7.5 \times 60) = 0.22222 \text{ [Unidades /min]}$$

$$\Delta b = (240-140) / (7.5 \times 60) = 0.22222 \text{ [Unidades /min]}$$

Incrementos para el intervalo de tiempo de las **13:30** a las **21:00** horas:

Cálculo del vector de rotación de la luz direccional:

$$\Delta x = 130^\circ / (15 \times 60) = 0.1444 \text{ [}^\circ/\text{min]}$$

$$\Delta y = 90^\circ / (15 \times 60) = 0.1 \text{ [}^\circ/\text{min]}$$

$$\Delta z = 0 \text{ [}^\circ/\text{min]}$$

Cálculo del color de la luz direccional:

$$\Delta r = (175-60) / (7.5 \times 60) = 0.25556 \text{ [Unidades/min]}$$

$$\Delta g = (175-60) / (7.5 \times 60) = 0.25556 \text{ [Unidades/min]}$$

$$\Delta b = (210-90) / (7.5 \times 60) = 0.26667 \text{ [Unidades /min]}$$

Cálculo del color de la luz ambiental:

$$\Delta r = (240-180) / (7.5 \times 60) = 0.13333 \text{ [Unidades /min]}$$

$$\Delta g = (240-180) / (7.5 \times 60) = 0.13333 \text{ [Unidades /min]}$$

$$\Delta b = (255-240) / (7.5 \times 60) = 0.03333 \text{ [Unidades /min]}$$

Incrementos para el intervalo de tiempo de las **21:00** a la **1:30**:

Cálculo del vector de rotación de la luz direccional:

$$\Delta x = 130^\circ / (9 \times 60) = 0.24074 \text{ [}^\circ/\text{min]}$$

$$\Delta y = 90^\circ / (9 \times 60) = 0.16667 \text{ [}^\circ/\text{min]}$$

$$\Delta z = 0 \text{ [}^\circ/\text{min]}$$

Cálculo del color de la luz direccional:

$$\Delta r = 0 \text{ [Unidades/min]}$$

$$\Delta g = 0 \text{ [Unidades/min]}$$

$$\Delta b = 0 \text{ [Unidades/min]}$$

Cálculo del color de la luz ambiental:

$$\Delta r = 0 \text{ [Unidades/min]}$$

$$\Delta g = 0 \text{ [Unidades/min]}$$

$$\Delta b = 0 \text{ [Unidades/min]}$$

Incrementos para el intervalo de tiempo de la **1:30** a las **6:00** horas:

Cálculo del vector de rotación de la luz direccional:

$$\Delta x = 130^\circ / (9 \times 60) = 0.24074 \text{ [}^\circ/\text{min]}$$

$$\Delta y = 90^\circ / (9 \times 60) = 0.16667 \text{ [}^\circ/\text{min]}$$

$$\Delta z = 0 \text{ [}^\circ/\text{min]}$$

Cálculo del color de la luz direccional:

$$\Delta r = (225-60) / (4.5 \times 60) = 0.61111 \text{ [Unidades/min]}$$

$$\Delta g = (175-60) / (4.5 \times 60) = 0.42592 \text{ [Unidades/min]}$$

$$\Delta b = (175-90) / (4.5 \times 60) = 0.31481 \text{ [Unidades/min]}$$

Cálculo del color de la luz ambiental:

$$\Delta r = (210-180) / (4.5 \times 60) = 0.11111 \text{ [Unidades/min]}$$

$$\Delta g = (180-140) / (4.5 \times 60) = 0.14814 \text{ [Unidades/min]}$$

$$\Delta b = (255-140) / (4.5 \times 60) = 0.42592 \text{ [Unidades/min]}$$

A continuación se presenta una tabla que incluye los incrementos calculados hasta este momento, sin embargo la información recabada hasta este punto no es suficiente para calcular la animación de los dos nodos de luz.

Hora	Rotación Luz Direccional R (Rx, Ry, Rz)	Color RGB Luz Direccional D (Dx, Dy, Dz)	Color RGB Luz Ambiental A (Ax, Ay, Az)
6:00	R (-10,45,0)	D (225,175,175)	A (210,140,140)
13:30	R (-75,0,0)	D (175,175,210)	A (240,240,240)
21:00	R (-10,-45,0)	D (60,60,90)	A (180,180,255)
1:30	R (-75,0,0)	D (60,60,90)	A (180,180,255)

TABLA VI.7

Diseño de la iluminación

Para calcular la iluminación a cualquier hora del día es necesaria la siguiente información: los intervalos, los incrementos de rotación, los incrementos de color y los contadores necesarios para repetir el número de veces indicado por ellos mismos, a los incrementos de color o de grados. A continuación se enuncian las formulas que describen en su totalidad a los nodos, #ambient y #directional, a cualquier hora del día.

Fórmulas generales para obtener la iluminación a cualquier hora del día:

Rotación de la luz Direccional $R_n(R_{nx}, R_{ny}, R_{nz})$:

Vector $(R_x \pm (\Delta x * (\text{contador1})), R_y \pm (\Delta y * (\text{contador2})), 0)$

Color de la Luz Direccional (R,G,B):

$(D_x \pm (\Delta r * (\text{variable1})), D_y \pm (\Delta g * (\text{variable1})), D_z \pm (\Delta b * (\text{variable1})))$

Color de la Luz Ambiental (R,G,B):

$(D_x \pm (\Delta r * (\text{variable1})), D_y \pm (\Delta g * (\text{variable1})), D_z \pm (\Delta b * (\text{variable1})))$

Las únicas incógnitas en las formulas anteriores, son: **variable1** y **contador2**. Variable1 puede tomar dos valores diferentes:

Variable1= contador1 (6:00-13:30, 21:00-1:30)

Variable1= CTE – contador1 (13:30 -21:00, 1:30-6:00)

Los intervalos 6:00-21:00 y 21:00-6:00 fueron divididos en dos debido al comportamiento de la componente R_x del vector R (que va de -10° a -75° y posteriormente de -75° a -10°) teniendo como horas intermedias a las 13:30 y a la 1:30.

El valor de la **CTE** para el primer semi-intervalo (13:30-21:00), es 450 ya que este intervalo cuenta con 15 horas (6:00-21:00) es decir 900 minutos y justo la mitad de ese intervalo son los 450 minutos que se utilizan como constante, mientras que para el segundo semi-intervalo (1:30-6:00) es 270 ya que este intervalo cuenta con 9 horas (21:00-6:00) es decir 540 minutos y justo la mitad de ese intervalo son los 270 minutos que se utilizan como constante.

Las variables **contador1** y **contador2** significan el número de segundos a partir de la hora que sirve como inicio del intervalo.

Por ejemplo si queremos el valor de los contadores para el intervalo de **6:00 a 13:30**, las fórmulas serán las siguientes:

cont1= (hora*60)+minuto-360

cont2= (hora*60)+minuto-360

El número que se resta es 360 debido a que son los minutos a partir de las 0:00 horas hasta las 6:00 horas que es el inicio del intervalo en cuestión.

El factor (hora*60) sirve para obtener los minutos de las 0:00 horas a la hora especificada en la variable hora. Finalmente la variable minuto almacena el minuto de la hora virtual.

Fórmulas para obtener los contadores en cada uno de los intervalos:

Intervalo de las **13:30** horas a las **21:00** horas:

$$\text{cont1} = -((\text{hora} * 60) + \text{minuto} - 1260)$$

$$\text{cont2} = (\text{hora} * 60) + \text{minuto} - 360$$

Intervalo de las **21:00** horas a las **0:00** horas:

$$\text{cont1} = (\text{hora} * 60) + \text{minuto} - 1260$$

$$\text{cont2} = (\text{hora} * 60) + \text{minuto} - 1260$$

Intervalo de las **0:00** horas a las **1:30** horas:

$$\text{cont1} = (\text{hora} * 60) + \text{minuto} + 180$$

$$\text{cont2} = (\text{hora} * 60) + \text{minuto} + 180$$

Intervalo de las **1:30** horas a las **2:00** horas:

$$\text{cont1} = 300 - \text{minuto}$$

$$\text{cont2} = (\text{hora} * 60) + \text{minuto} + 180$$

Intervalo de las **2:00** horas a las **6:00** horas:

$$\text{cont1} = -((\text{hora} * 60) + \text{minuto} - 360)$$

$$\text{cont2} = (\text{hora} * 60) + \text{minuto} + 180$$

Con las fórmulas para obtener la iluminación, las fórmulas para los contadores y con el uso de la siguiente tabla, que reúne los parámetros antes obtenidos, exceptuando las fórmulas, se tiene completo el diseño para la iluminación.

Hora	R	D	A	ΔR	ΔD	ΔA	CTE
6:00	R(-10,45,0)	(225,175,175)	(210,140,140)	0.14444	0.11111	0.06667	
				0.1	0	0.22222	---
				0	0.07778	0.22222	
13:30	R(-75,0,0)	(175,175,210)	(240,240,240)	0.14444	0.25556	0.13333	
				0.1	0.25556	0.13333	450
				0	0.26667	0.03333	
21:00	R(-10,-45,0)	(60,60,90)	(180,180,255)	0.24074	0	0	
				0.16667	0	0	---
				0	0	0	
1:30	R(-75,0,0)	(60,60,90)	(180,180,255)	0.24074	0.61111	0.11111	
				0.16667	0.42592	0.14814	270
				0	0.31481	0.42592	

TABLA VI.8

Diseño completo de la iluminación

VI.3.6 DISEÑO DEL AUDIO

Se ha comprobado que la inclusión de sonido en la presentación de imágenes o de video, mejora la percepción de estos elementos, es por ello que hemos incluido tres fuentes diferentes de audio en el paseo virtual.

La primera de ellas corresponde al audio nombrado “Ambiente”, que como su nombre lo indica, simula un ambiente virtual donde podemos escuchar el viento, las aves, los grillos, etc.

La segunda fuente de audio la hemos llamado “Música”, este canal de audio contiene diversos fragmentos de melodías prehispánicas que utilizamos como música de fondo.

El tercer canal que hemos incluido en el paseo virtual, es el llamado “Voz”. Dicho canal es el más interesante y completo de los tres existentes, ya que por medio de este canal podemos incluir información acerca de la historia y arquitectura teotihuacana.

Este canal se activa cuando entramos a una sección de la zona arqueológica en la que existe información audible para ser presentada. De ésta forma el PV adquiere características de tutorial multimedia al mismo tiempo que presenta en tiempo real el entorno tridimensional reconstruido para la zona arqueológica.

Cabe mencionar que los tres canales de audio presentes en el programa son independientes entre sí y cada uno de ellos tiene su propio control para modular el volumen del canal correspondiente.

Los canales “Ambiente” y “Música” crecen (Fade In) o decrecen (Fade Out) en su volumen dependiendo de las transiciones programadas para los diferentes momentos del día.

A diferencia del diseño de la iluminación, en el diseño del audio los eventos son dependientes de las coordenadas espaciales y no dependen del tiempo virtual generado por la función `Timer()`.

Para los canales “Ambiente” y “Música” no se realizó su diseño de audio, debido a que no requerían de éste y de realizarlo, únicamente hubiera sido una tarea infructuosa, ya que únicamente se actualizan ambos canales dependiendo la hora del día.

Para el canal de “Voz”, se presenta la siguiente tabla, necesaria para programar los eventos de activación de los audios, con dicha tabla queda completo el diseño del audio:

Zona	Sección	Intervalo en X	Intervalo en Z	Intervalo en Y	Audio (*.Wav)
Ciudadela	Plaza	3000<X<3500	-----	-----	pvcidud1
Ciudadela	Pirámide Quetzalcoatl	7250<X<9100	-870<Z<820	-----	pvquetzal
Calzada1	Edificios Superpuestos	X<-1060	-3715<Z<-1650	-----	pvedifsuper
Calzada1	Plaza Oeste	X<-1060	Z<-3720	-----	pvcjtoeste
Calzada2	Grupo Vikingo	X>620	-1270<Z<0	-----	pvvikingo
Plaza Sol	Plaza	640<X<950	-6160<Z<-75	-----	pvplazasol
Plaza Sol	Pirámide Del Sol	2500<X<8000	-4220<Z<2050	-----	pvpirsol
Mural Puma	Mural del Puma	500<X<650	-2900<Z<-2600	-----	pvpuma
Plaza Luna	Plaza	X>-520	-250<Z<0	-----	pvplazaluna
Plaza Luna	Pirámide de la Luna	-510<X<650	Z<-3300	-----	pvpir luna
Cjto. Palacios	Mariposas Quetzal	-1050<X<-450	-750<Z<80	Y>140	pvmari2
Cjto. Palacios	Patio de los Jaguares	-1700<X<-1120	-630<Z<-425	Y<130	pvjaguares
Cjto. Palacios	Caracoles Emplumados	-1080<X<-875	-635<Z<135	Y<130	pvcaracoles

TABLA VI.9
Diseño del audio par el canal “Voz”

VI.3.7 DISEÑO DEL AMBIENTE

El diseño del ambiente se realiza de manera muy sencilla. La primera de dos tareas a realizar consiste en actualizar el mapeado del mundo dependiendo de la hora del día.

El mundo virtual está formado por las caras interiores de un cubo de dimensiones muy grandes, cada una de las seis caras de éste cubo será mapeada con una imagen diferente de la otras. Se diseñaron imágenes para cada una de las secciones de la zona arqueológica (siete en total), ya que la vista del ambiente en cada una de las secciones, es diferente a la vista de las otras seis.

Para dar el efecto de avance del tiempo, se tuvieron que diseñar cinco diferentes momentos del día en cada una de las siete secciones arqueológicas y para cada una de las cuatro caras verticales del cubo, es decir 5x7x4 que da un total de 140 imágenes más cinco imágenes correspondientes a la cara superior del cubo, da un total de 145 imágenes diferentes para la ambientación de los mundos virtuales.

Sin embargo si únicamente actualizamos la imagen dependiendo del momento del día, notaríamos que la transición de una foto a la otra es inmediata, lo que produce un efecto irreal, ya que veríamos que el entorno cambia inmediatamente su luminosidad.

Sería imposible obtener una imagen para cada minuto del día y así poder animar el ambiente logrando la continuidad y animación del entorno, es por ello que para el diseño del ambiente y específicamente para el diseño de las transiciones, utilizamos imágenes con PNG con canal alfa de transparencia para traslaparlas una sobre otra, desvaneciendo la superior poco a poco, para que la inferior se vaya mezclando con la primera y lograr así el efecto de desvanecimiento y de continuidad en el ambiente.

Para manejar la transparencia de las imágenes, se utiliza un rango de 0 a 100 unidades, donde 100 significa totalmente opaco y cero totalmente transparente; si dividimos el día en intervalos de dos horas, tenemos que consumir las cien unidades en este intervalo de tiempo, para que al final del mismo, la imagen superior sea totalmente transparente y podamos ver la inferior de manera nítida, posteriormente en el siguiente intervalo, la imagen inferior ahora será superior y como inferior tendremos otra imagen que represente otro momento del día.

Si dividimos las 100 unidades de transparencia entre los 120 minutos del intervalo obtenemos la constante de atenuación con un valor de $5/6$ o 0.83333 .

En el momento en que se carga una sección de la zona arqueológica, se almacenan las texturas en 25 variables $tN... tN4...tOe...tO4... t...t4$, para representar el ambiente de dicha sección a cualquier hora del día, de esta forma no tenemos que cargar siempre las 145 imágenes de los ambientes. Aunque tenemos las texturas del ambiente almacenadas en 25 variables, cuando se cargan en tiempo real únicamente ejecutaremos 10 de ellas, cinco que corresponden a las caras interiores del cubo (exceptuando la inferior) en el intervalo presente en ese instante y las otras cinco que corresponden al siguiente intervalo.

$$\text{Atenuación} = 0.83333 \text{ [unidades/min]}$$

Intervalo	Imágenes Superiores	Imágenes Inferiores
06:00 -- 8:00	tN, tS, tE, tOe, t	tN1, tS1, tE1, tO1, t1
8:00 --10:00	tN1, tS1, tE1, tO1, t1	tN3, tS3, tE3, tO3, t3
10:00 -- 12:00	tN3, tS3, tE3, tO3, t3	tN2, tS2, tE2, tO2, t2
12:00 – 13:00	tN2, tS2, tE2, tO2, t2	tN2, tS2, tE2, tO2, t2
13:00 – 15:00	tN2, tS2, tE2, tO2, t2	tN3, tS3, tE3, tO3, t3
15:00 – 17:00	tN3, tS3, tE3, tO3, t3	tN1, tS1, tE1, tO1, t1
17:00 – 19:00	tN1, tS1, tE1, tO1, t1	tN, tS, tE, tOe, t
19:00 – 21:00	tN, tS, tE, tOe, t	tN4, tS4, tE4, tO4, t4
21:00 – 5:00	tN4, tS4, tE4, tO4, t4	tN4, tS4, tE4, tO4, t4
5:00 – 6:00	tN4, tS4, tE4, tO4, t4	tN, tS, tE, tOe, t

TABLA VI.10
Diseño del Ambiente

Hasta este momento hemos hablado del diseño de eventos como la iluminación y el ambiente, que dependen del tiempo y no del espacio. Por lo tanto debemos hablar del formato que maneja nuestro tiempo virtual, y que a continuación se presenta:

Tiempo: HH : MM AM/PM

A nuestro tiempo virtual podremos asignarle tres velocidades diferentes (lenta, media y rápida), también podemos detenerlo en cualquier momento, y actualizarlo asignándole cualquier hora que deseemos.

VI.3.8 DISEÑO DE LAS TRANSICIONES

El modelado de la zona arqueológica de Teotihuacan tuvo que ser dividido en siete secciones debido a que el número de polígonos del modelo completo es muy grande, y por lo tanto es imposible procesar en tiempo real todo el modelo.

Debido a esto decidimos que las transiciones entre las diferentes secciones pudieran realizarse de dos formas diferentes; la primera de ellas, de manera secuencial, como si estuviéramos caminando a lo largo de la calzada de los muertos en dirección sur a norte; la segunda es de manera aleatoria, no importando donde nos encontremos, podemos “saltar” a cualquier otra sección en el momento que se requiera.

Los eventos de transición secuencial son función de las coordenadas espaciales, no así la transición aleatoria que se activa cuando seleccionamos una de las siete ligas presentes en el menú del PV. Por lo antes expuesto, únicamente realizamos la tabla de diseño para las transiciones secuenciales, que a continuación presentamos:

Zona	Coordenadas Siguiete Sección	Coordenadas Anterior Sección	Rango de Movimientos
Ciudadela	$-10000 < X < 10000$	-----	$-300 < X < 10100$
	$Z < -7000$		$Z < 4000$
Calzada1	$Z < -5250$	$Z > 300$	$-2730 < X < 3000$
Calzada2	$Z < -5700$	$Z > 480$	$-1200 < X < 2550$
Plaza Sol	$Z < -6800$	$Z > 235$	$-4675 < X < 8800$
Mural del Puma	$Z < 4700$	$Z > 300$	$-4400 < X < 4150$
Plaza Luna	$X < -1440$	$Z > 1000$	$-4000 < X < 4200$
	$-300 < Z < 500$		$Z > -7500$
Conjunto Palacios	-----	$X > 150$	$X > -1880$
			$-740 < Z < 225$

TABLA VI.11
Diseño de las transiciones.

Observamos en la tercera columna, los rangos de movimiento correspondientes a cada una de las secciones de la zona arqueológica, estos con el fin de limitar el movimiento a lo largo y ancho de cada sección; ya que no es útil la navegación fuera del rango de interés.

Los rangos se encuentran alojados en el plano XZ, y sólo dentro de estos intervalos el usuario podrá navegar. En el momento que el usuario pretenda seguir su camino fuera del rango, automáticamente se detendrá para poder seguir se recorrido en otra dirección.

VI.4 DISEÑO ARQUITECTÓNICO

Su objetivo es desarrollar la estructura modular del programa y representar las relaciones de control entre módulos, así como el flujo de la información.

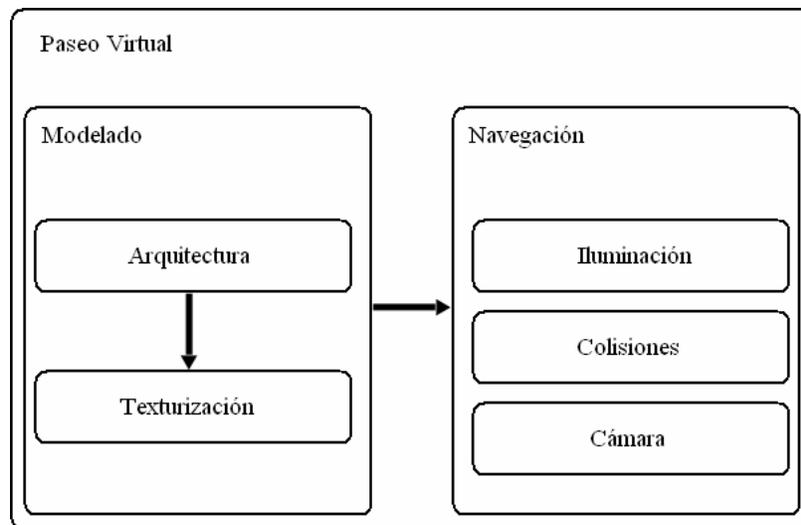


ILUSTRACIÓN VI.11
Esquema del Diseño Arquitectónico

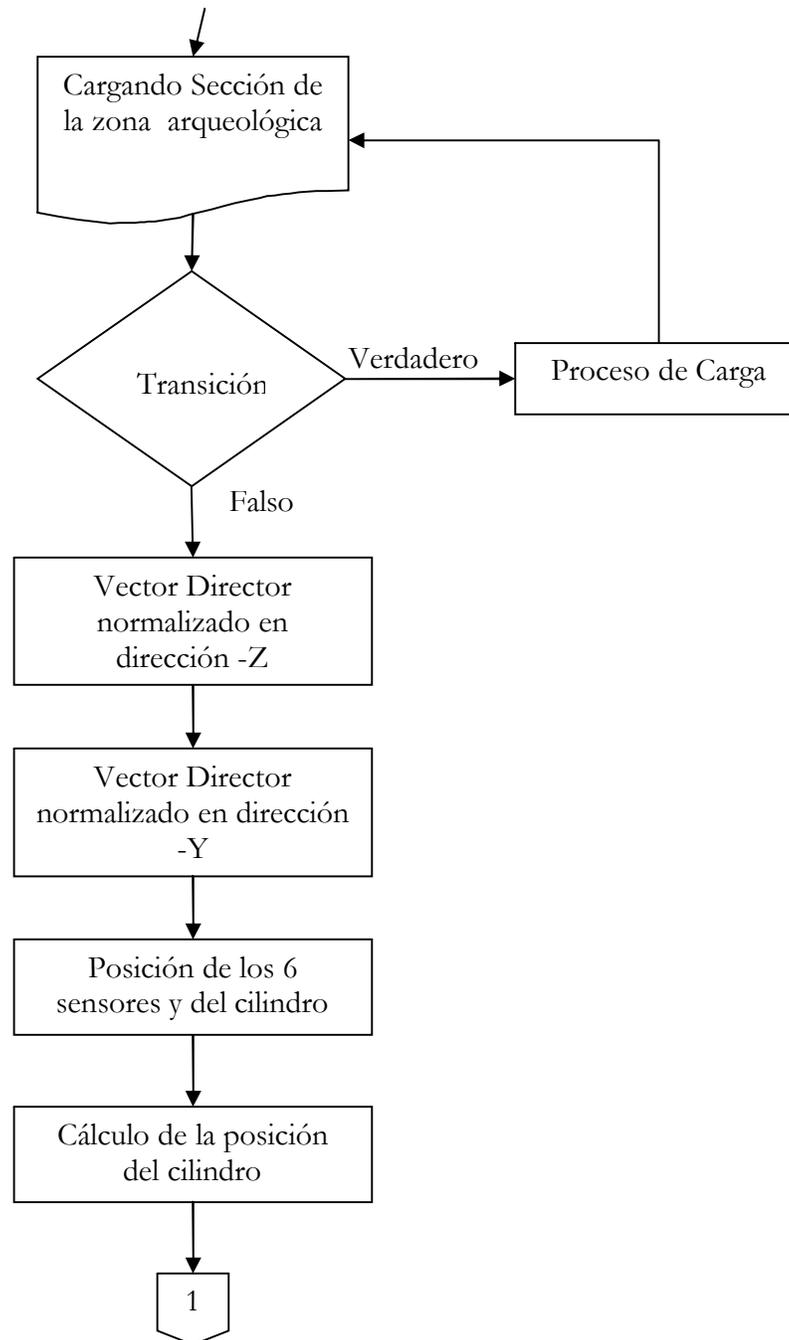
En el esquema anterior se muestran dos grandes grupos: el modelado y la navegación, el primer grupo se subdivide en arquitectura y texturización, por su parte el segundo grupo se subdivide en iluminación, colisiones y cámara. Cada uno de ellos se explicó con anterioridad.

Lo que nos muestra el diagrama anterior es la forma en la que interactúan los módulos, es decir como pasa la información de uno a otro.

Haciendo una analogía con la arquitectura de un edificio, podemos decir que, nos interesa saber como están formadas cada una de las partes que integran al edificio y como interactúan entre sí, para obtener el máximo provecho del lugar; de igual forma en el diseño arquitectónico, nos interesa saber como están diseñados cada uno de los módulos que integran al proyecto final, y como interactúan dichos módulos entre sí, para entender como funciona parte por parte el software y diseñarlo de manera eficaz, para obtener el mayor provecho posible.

VI.5 DISEÑO PROCEDURAL

El diseño procedural o de la programación nos proporciona información importante para la etapa de codificación. Por medio de herramientas gráficas o de forma escrita, podemos representar cualquier algoritmo, de manera que quede perfectamente definido, sin ambigüedades y con un número finito de pasos, para obtener la mejor solución al problema planteado. A continuación se presenta el diagrama de flujo para la detección de colisiones:



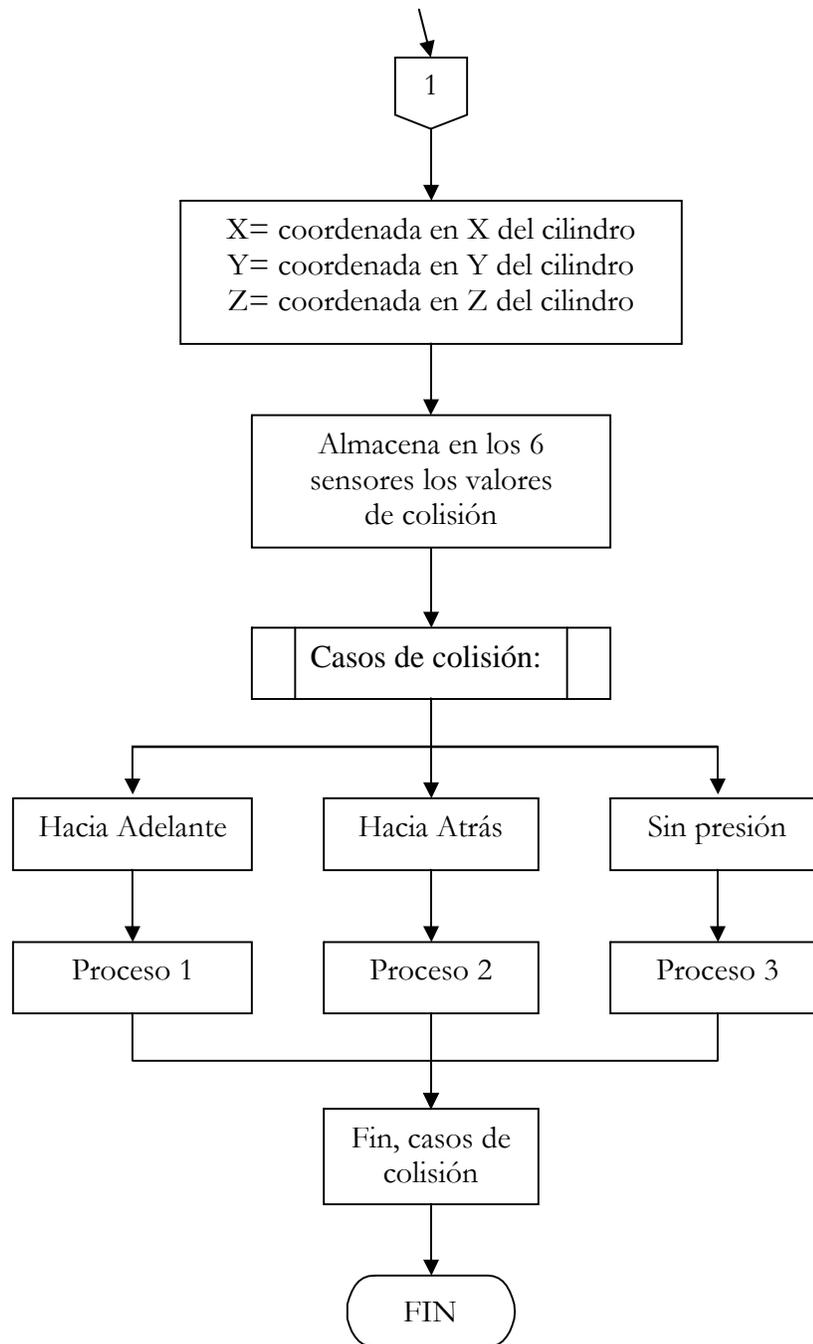


DIAGRAMA VI.11

Diagrama de Flujo para la detección de colisiones

Sin embargo, el diagrama de flujo no es la única herramienta empleada para el diseño procedural. A continuación representamos la sección de código perteneciente al script del manejo de cámara encargada de la generación del tiempo virtual, representada con el diagrama de Chapin:

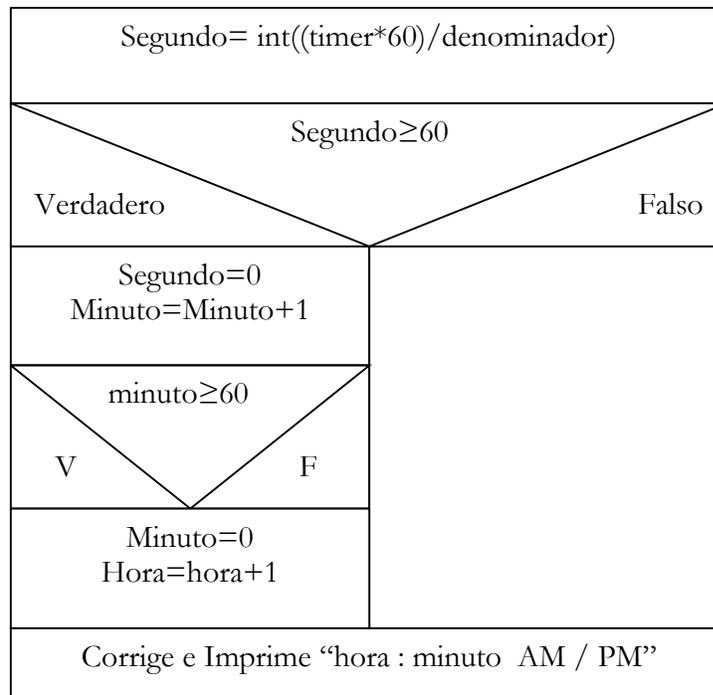


DIAGRAMA VI.12

Diagrama de Chapin para generar la hora virtual.

La última forma de representar un algoritmo, es de forma escrita, y es conocida como “pseudo código”. Por medio de este representaremos a la sección de código perteneciente al script de cámara y que se encarga de los movimientos de la misma:

Si la tecla presionada es **adelante y atrás** entonces
 Velocidad= velocidad mínima
 Fin Si

Si la tecla presionada es **adelante** entonces
 Velocidad= velocidad + aceleración
 activaColisión= Verdadero
 signo=1
 Fin Si

Si la tecla presionada es **atrás** entonces
 Velocidad= velocidad + aceleración
 activaColisión= Verdadero
 signo=-1
 Fin Si

Continúa Pseudo código.

Si la tecla presionada es **“d”** entonces
Velocidad= velocidad - desaceleración
Fin Si

Si la tecla presionada es **“a”** entonces
rotaciónDeCámara (3°,0°,0°)
Fin Si

Si la tecla presionada es **“a”** entonces
rotaciónDeCámara (-3°,0°,0°)
Fin Si

Si la tecla presionada es **izquierda** entonces
rotaciónDeCámara (0°,5°,0°)
Fin Si

Si la tecla presionada es **derecha** entonces
rotaciónDeCámara (0°,-5°,0°)
Fin Si

El diseño procedural, está basado en el teorema propuesto por Edgar Dijkstra a fines de los 60s, demuestra que cualquier algoritmo se puede expresar con 3 tipos de instrucciones: la secuencia, la condición y la repetición, e inició lo que se conoce como programación estructurada.

VI.6 DISEÑO DE LA INTERFASE

Debido a que el número de polígonos procesados en tiempo real debe ser muy bajo (al rededor de 10000 polígonos como máximo para Shockwave 3D), decidimos particionar la zona arqueológica en 7 diferentes salas, estas son:

- Ciudadela
- Calzada 1 (comprende edificios superpuestos, plaza oeste y la parte correspondiente de la calzada de los muertos)
- Calzada 2 (grupo vikingo y la parte correspondiente de la calzada de los muertos)
- Plaza del Sol
- Mural del Puma
- Plaza de la Luna
- Conjunto de Palacios (Palacio de las Mariposas Quetzal, Patio de los Jaguares y Subestructura de los Caracoles emplumados)

Elementos que requerían un alto nivel de detalle para ser modelados, como las almenas, las cabezas de Quetzalcoatl y Tlaloc, no fueron incluidos debido al alto costo que hubiera representado, en lugar de modelar dichos elementos, estos fueron sustituidos por imágenes, obteniéndose mejores resultados en cuanto a la fluidez del PV y a la apariencia.

El acceso a cada una de las secciones se lleva a cabo de dos formas diferentes; de manera secuencial, como si estuviéramos caminando a lo largo de la zona arqueológica en dirección Sur a Norte o viceversa; la segunda es de manera aleatoria ya que en algunas ocasiones necesitamos ir de un lugar a otro muy lejano y esto nos tomaría mucho tiempo al realizarlo de manera secuencial, es por esto que eligiendo por medio de una de las siete ligas que aparecen en la pantalla, podemos elegir cualquier sección.

Hablaremos ahora de las interfaces diseñadas para el PV y su complemento multimedia, enfocándonos mayormente a la interfaz del PV. A continuación se muestra una lista con las interfaces diseñadas.

Menú principal

Paseo Virtual
Historia
Fotos
Juegos

A continuación presentamos de forma gráfica, la explicación de los elementos que componen la interfase del menú principal:



ILUSTRACIÓN VI.12
Interfase del Menú Principal

- **PASEO VIRTUAL.-** Permite acceder a la interfase del paseo virtual, comenzando el recorrido en la Ciudadela.
- **HISTORIA.-** Nos muestra distintos puntos importantes que destacaron dentro de la cultura teotihuacana.
- **FOTOS.-** Tiene la finalidad de realizar un comparativo de distintas lugares de la zona arqueológica, en como se encuentran en la actualidad y la reconstrucción que se realizó.
- **JUEGOS.-** Ésta sección se diseño para explotar las habilidades y destrezas haciendo alusión a distintos rompecabezas de la zona arqueológica de Teotihuacan.
- **CRÉDITOS.-** Se indican las distintas referencias que colaboraron y ayudaron de cierta forma a complementar nuestro proyecto.
- **AGRADECIMIENTOS.-** Se hace referencia a aquellas personas o instancias que tienen un motivo especial en la realización de este proyecto.
- **SALIR.-** Da por concluido la ejecución de la aplicación.

Ahora presentamos de forma gráfica, la explicación de los elementos que componen la interfase del paseo virtual:

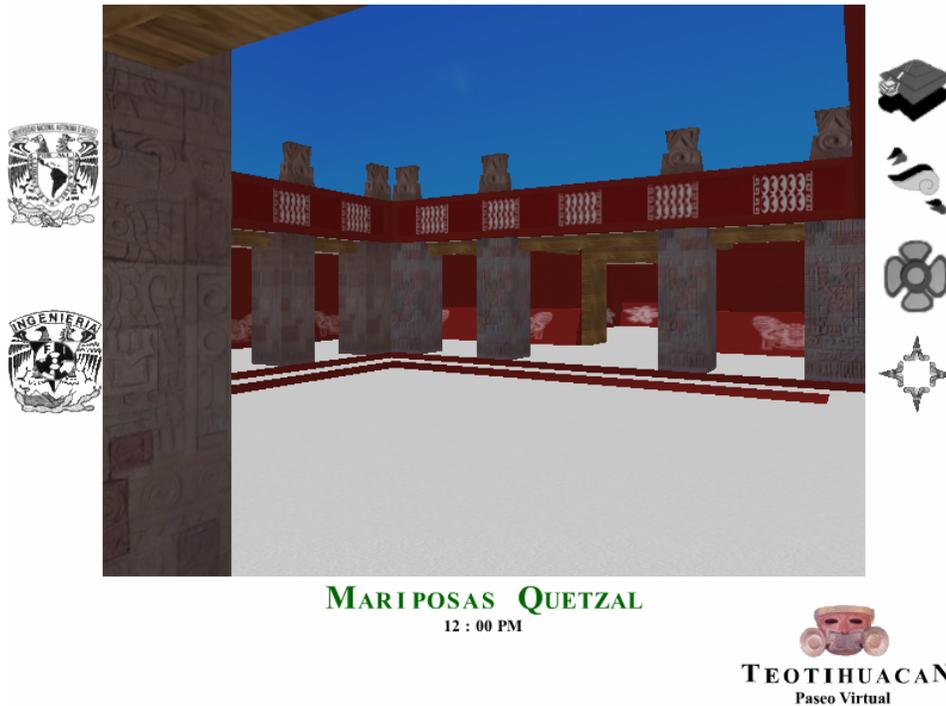


ILUSTRACIÓN VI.13
Interfase del Paseo Virtual

Para cada una de las diferentes opciones, los menús se despliegan oprimiendo los íconos correspondientes, donde dichos menús así como sus íconos se presentan a continuación:



La selección de cualquier liga, nos lleva hacia la sección elegida.



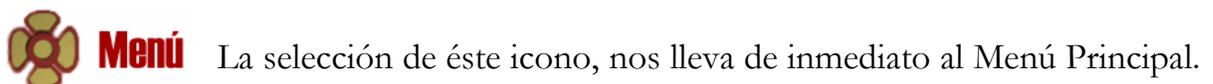
Muestra la información correspondiente a las teclas utilizadas en el paseo virtual.



Con los tres canales, podemos modular el sonido del paseo virtual.



Sirve para actualizar la hora y modular la velocidad el reloj.



 **CAPÍTULO VII** 
DESARROLLO

Habiendo concluido las fases de análisis y diseño, continuaremos el desarrollo del software con la fase de Codificación. Al igual que en el capítulo anterior dividiremos el desarrollo en dos partes, la primera corresponde a la codificación de la arquitectura, y la segunda corresponde a la codificación de la navegación.

Para la primer parte hablaremos del desarrollo de la arquitectura en lugar de codificación, ya que este primer término es mucho más apropiado.

VII.1 DESARROLLO DE LA ARQUITECTURA

El proceso para la creación de una animación mediante 3D pasa por varias etapas:

1. Modelado de las diferentes formas presentes en la escena.
2. Aplicación de una textura a cada una de esas formas.
3. Definición de las fuentes de luz tanto en su iluminación como en su ubicación.
4. Especificación de las cámaras virtuales que se van a utilizar.
5. Procesado de toda la información por parte de la computadora.

1. MODELADO

El primer paso para realizar la creación de una escena consiste en modelar cada uno de los cuerpos presentes mediante formas simples, como planos, cajas, esferas, cilindros, etc. o mediante formas compuestas, realizando operaciones booleanas, extrusiones de formas planas, etc. La unidad mínima para definir un plano es un triángulo, y sirve para medir la perfección de cualquier modelo, es decir, que mientras un objeto tenga más triángulos, mayor será el detalle de este. Lo único que tenemos que cuidar al realizar el modelado, es encontrar el balance entre la calidad o “apariencia” de los modelos y la “eficiencia” debido al número de polígonos que tengan todos los objetos que en conjunto formen el mundo virtual.

Es muy importante que los modelos se vean lo mejor posible, sin que esto afecte el rendimiento a la hora de navegar. Los modelos son generados con un shader estándar como textura, es por ello que el siguiente paso después del modelado, es la texturización.

2. TEXTURIZADO

La texturización consiste en aplicar a cada triángulo (a cada plano) un aspecto, que se denomina textura. Los programas de 3D incorporan diferentes texturas para simular casi cualquier superficie (metal, piedra, agua, madera...), sin embargo las texturas mencionadas no son apropiadas para nuestros modelos, es por ello que generamos distintas texturas, con el objetivo de simular un ambiente de reconstrucción en la zona arqueológica.

Al aplicar una textura, a cada píxel de cada triángulo se le asigna un color RGB (rojo, verde y azul). Algunos programas (incluyendo 3D Studio) codifican cada píxel con cuatro valores, tres de los cuales son el color y el cuarto aporta la información de transparencia que tiene ese punto, denominado “Canal Alpha”. Ese cuarto valor es especialmente indicado para producir efectos de niebla o simulación de agua y cristal, en los que interviene la transparencia.

Para explicar los procesos de modelado y de texturización, explicaremos la creación de una pirámide con talud, tablero, escaleras y templo. Posteriormente explicaremos la aplicación de las texturas a dicha pirámide. Este ejemplo es aplicable para cualquier otro modelo generado dentro de 3D Studio Max.

3. ILUMINACIÓN

Cuando tenemos que definir una escena no sólo hay que realizar los elementos que la conforman. También hay que definir las fuentes de luz (iluminación) y desde luego donde la estamos observando (cámaras)

La correcta iluminación de la escena es uno de los factores más importantes a la hora de evaluar el resultado artístico del proyecto.

4. CAMARAS

Su función es definir los puntos de vista de la escena que estamos tratando. En ellas podemos modificar la posición, rotación y el acercamiento, así como el ancho focal.

5. RENDERING

Se tiene que crear la escena, darle una textura, definir las luces y las cámaras y definir el movimiento. Pero para que ese movimiento sea efectivo hay que realizar el cálculo de cada uno de los cuadros que van a componer la secuencia. Para cada cuadro hay que calcular la nueva posición de los cuerpos, la nueva posición de la cámara y la iluminación de toda la escena.

El proceso de rendering consta de dos etapas:

1. Geometrización. Consiste en calcular la forma de cada uno de los objetos presentes en la escena y de aplicar el número de triángulos correspondientes a cada modelo.
2. Rasterización. Consiste en calcular el color final de cada uno de los píxeles de la escena y proyectar la imagen en 3D en la pantalla (2D).

Dependiendo de la plataforma que utilicemos para el cálculo, el proceso de rendering será más o menos lento. Pero siempre los tiempos de cálculo de rendering son muy elevados.

Simplificando el desarrollo del modelo de la zona arqueológica, lo podemos resumir en dos grandes pasos, los cuales son: la creación de objetos, donde se explicará como se generan los objetos y que comandos y herramientas se utilizan; y la aplicación de las texturas al modelo, donde se explicara como se realiza dicha tarea, como se genera el archivo de materiales, su aplicación al modelo, y como se realiza el mapeo a cada polígono del modelado.

A continuación se explicara de manera detallada cada uno de estos dos puntos, para hacer referencia en ellos que herramientas y que procedimientos siguieron a lo largo del desarrollo del modelo.

VII.1.1 CREACIÓN DE OBJETOS

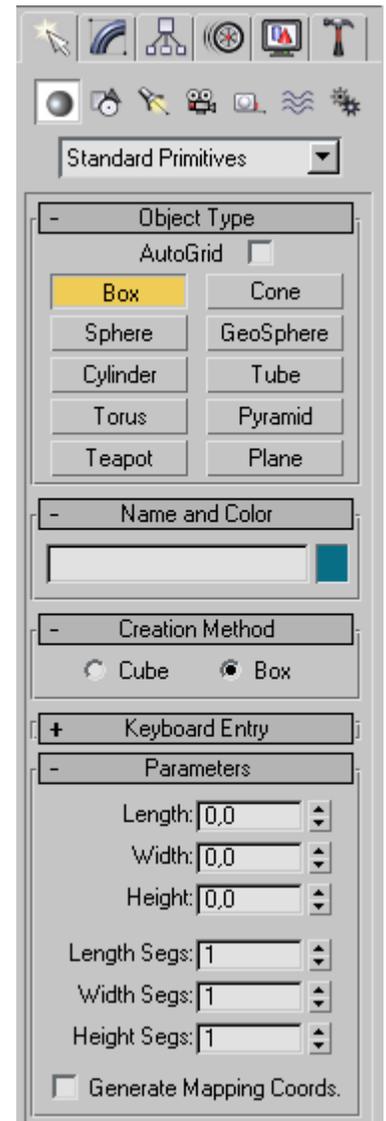
La herramienta principal que se utiliza para la creación de objetos que integran el modelo de la zona arqueológica es el panel de comandos Crear, a continuación se describirá como esta integrado y como funciona.

INTERFAZ DEL PANEL DE COMANDOS CREAR

Los controles del panel de comandos Crear varían según el tipo de objeto creado. Sin embargo, algunos controles siempre están presentes y otros son comunes a diversos tipos de objetos. El panel de comandos Crear cuenta con un conjunto de características básicas comunes a todos los tipos de objetos, las cuales son: Categoría, Tipo de Objeto, Nombre y Color, Método de Creación, Parámetros, y Entradas de Teclado.

A continuación se explicaran brevemente las funciones utilizadas:

- Categoría: Los botones superiores acceden a las siete categorías básicas de objetos.
- Subcategoría. El campo de texto desplegable que se encuentra debajo de los botones de categoría acceden a las distintas subcategorías dependiendo de la categoría seleccionada.
- Tipo de objeto: Se muestran el conjunto de objetos básicos que hay en cada categoría.
- Nombre y color: En este apartado se muestra el nombre y el color que se asigna al objeto.
- Método de creación: En muchas ocasiones hay más de una forma de crear un objeto, tal es el caso de crear un objeto mediante el centro o la arista del mismo. Si se desea cambiar el método, ha de hacerse antes de crearlo.
- Entradas del teclado: Permite la creación de objetos mediante entradas del teclado.
- Parámetros: La mayoría de los objetos presentan esta persiana, donde se muestran los parámetros de creación, es posible definir algunos parámetros antes de crear el objeto, mientras que otros sólo permiten ajustarlos en los ya creados.



CATEGORÍAS DE OBJETOS

En el panel de comandos Crear se presentan una serie de categorías básicas, siete en total, de las cuales se explicarán cuatro únicamente, debido a que son las utilizadas para el desarrollo del modelo, a continuación se explicarán brevemente:



Geometría: Presenta una serie de subcategorías, como son, Primitivas estándar (caja, esfera, etc), objetos de composición (se incluyen booleanos y transformaciones), etcétera.



Formas: Sólo incluye una subcategoría, que es Splines, donde se incluyen objetos 2D como línea, círculo, arco, etc., que si se unen podrán crear objetos 3D mediante extrusión o torneado.



Luces: Sólo tiene una subcategoría, que es predeterminado. Se presentan algunos tipos, como son omni, direccional, foco con objetivo y foco libre.



Cámaras: Al igual que las luces, tiene una sola subcategoría. Las cámaras proporcionan un punto de vista sobre la escena. Puede ser Free Camera o Target Camera.

Cabe mencionar que de estas siete categorías que contiene el panel de comandos Crear, solamente haremos uso de 4 de ellas, las cuales son las cuatro primeras, Geometría (para la creación de objetos geométricos, básicamente cajas y planos), Formas (para la creación de formas geométricas, básicamente líneas), Luces (para la creación de luces, básicamente omni) y Cámaras (para la creación de cámaras, básicamente libres).

Un aspecto importante a resaltar, es que la creación tanto de luces como de cámaras dentro de 3D Studio MAX, únicamente tendrán la finalidad de apoyo. La integración de estos objetos se realizará en Director 8.5 que utilizaremos para completar el PV por la zona arqueológica.

SUBCATEGORÍAS

Dependiendo de la categoría previamente seleccionada, va a depender el número y tipos de subcategorías que se tengan como opciones. Y como ya se mencionó anteriormente solamente haremos uso de cuatro categorías y dentro de cada una de ellas haremos uso de unas cuantas subcategorías. Para la categoría Geometría, haremos uso de la subcategoría Primitivas Estándar para crear objetos primitivos como cajas y planos, también utilizaremos la subcategoría Objetos Compuestos para crear objetos booleanos.

Para la categoría de Formas (Shapes), utilizamos la subcategoría Splines para generar líneas. Para las Luces y las Cámaras únicamente utilizaremos una subcategoría en cada una de ellas la cual es Estándar. Para el caso de las Luces se crearán luces Omnidireccionales y para las Cámaras, creamos cámaras libres.

VII.1.1.1 CREACIÓN DE LA GEOMETRÍA

Solamente haremos referencia a la creación de un elemento de la zona arqueológica virtual, debido a que el modelo ejemplifica cualquier caso que se pueda presentar para la elaboración de la geometría (pirámide con talud, tablero, templo y escaleras) y hacemos referencia a todas las herramientas que se aplicaron durante el desarrollo del modelo completo.

Cabe destacar que para el modelo se consideraron las dimensiones reales de los elementos de la zona arqueológica, de esta forma una unidad dentro del programa de 3D Studio MAX equivale a un metro real.

La pirámide está conformada por tres partes: cuerpo, escaleras y templo. Y la generación de cada una de estas va a estar integrada de la siguiente manera:

CUERPO

Generar **caja** y aplicar la propiedad **edit mesh**

Generar **caja** y aplicar el objeto compuesto **boolean**

ESCALERA

Generar **caja** y aplicar la propiedad **edit mesh**

Generar **línea** y aplicar la propiedad **extrude**

TEMPLO

Generar su contorno en su vista superior, utilizando una **línea** y posteriormente aplicar **extrude** para obtener la altura del templo.

CUERPO

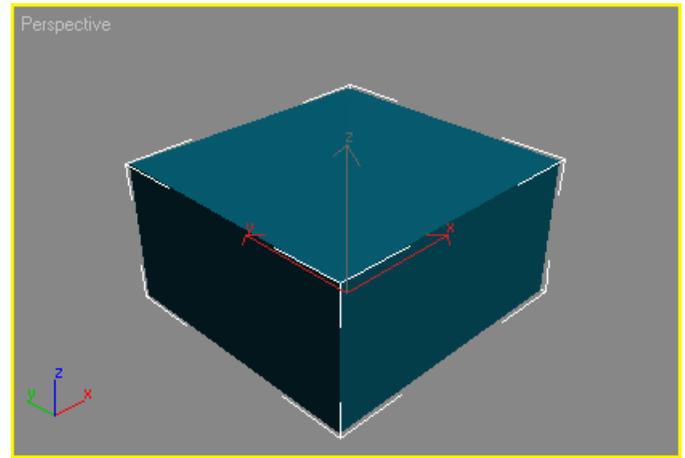
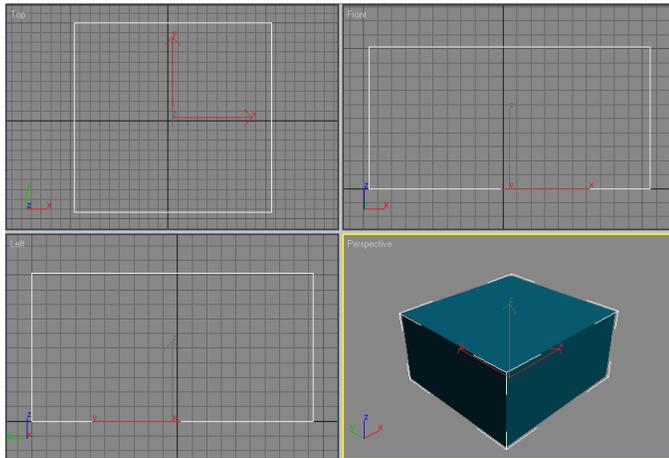
El cuerpo de la pirámide va a estar integrado por un talud y un tablero, que son los elementos básicos en la arquitectura Teotihuacana. El talud es generado con una caja de dimensiones, 30 unidades de largo, 30 unidades de ancho y 0.5 unidades de alto.

A continuación se explicara los pasos a seguir para generar una caja:

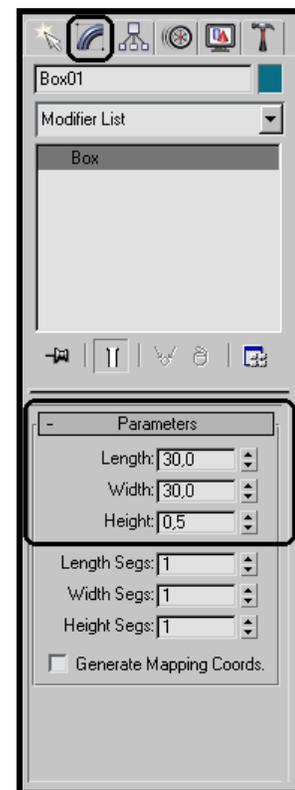
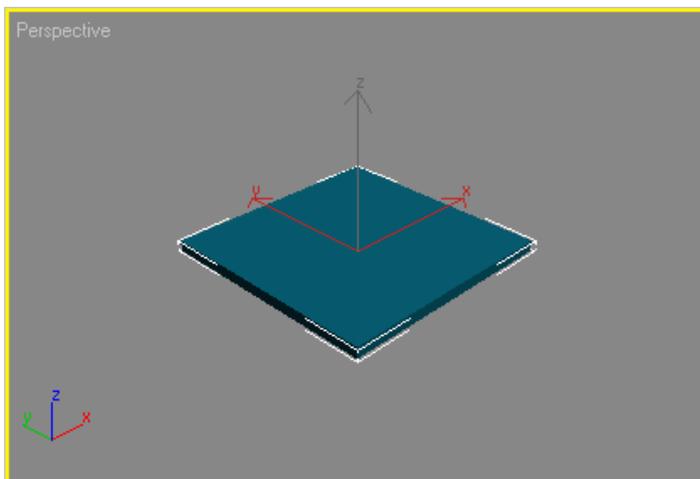
1. Elegir la categoría de objetos mediante los botones de la parte superior del panel de comandos Crear Geometría. En algunas categorías hay que designar la subcategoría que en este caso corresponde a la categoría Geometría (Geometry), y dentro de ésta, la subcategoría Primitivas Estándar (Standard Primitives).
2. Presionar un botón para seleccionar un tipo de objeto, en este caso Caja (Box). El panel de comandos presenta las opciones de cada tipo de objeto, como los parámetros.
3. Para crear el objeto se debe arrastrar el ratón y elegir en un puerto de vista, mientras se realiza esto, el objeto se sitúa en la cuadrícula activa del puerto de vista. Cuando se suelta el ratón se establece la longitud y anchura. Para definir la altura se arrastra el ratón hacia arriba o abajo y llegado el punto deseado, se presiona el botón derecho del ratón.
4. Cuando esté terminado y aún seleccionado se pueden cambiar los parámetros del mismo en una o más opciones específicas de ese objeto estableciendo nuevas unidades. Así, se puede dar una mayor altura, anchura, longitud, etc a la caja.

Los parámetros del objeto primitivo estándar caja son los siguientes:

Primitiva	Punto XYZ	Parámetros
Caja	Centro de la base	Longitud, anchura y altura

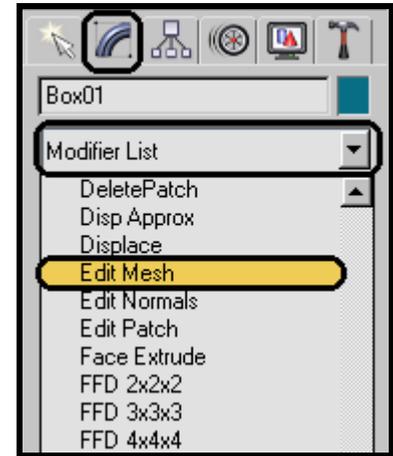


Una vez generada la caja nos auxiliaremos del panel de comandos Modificar (Modify) para de esta forma poder modificar los parámetros de la caja generada de acuerdo a nuestras necesidades, las dimensiones serán: 30X30X0.5.



El siguiente paso es deformar la caja para obtener el talud del cuerpo de la pirámide y esto lo haremos utilizando el panel de comandos Modificar, aplicando el modificador **Malla Editable (Edit Mesh)**. Con la finalidad de que la caja pueda ser manipulada y deformarla de acuerdo con nuestras necesidades.

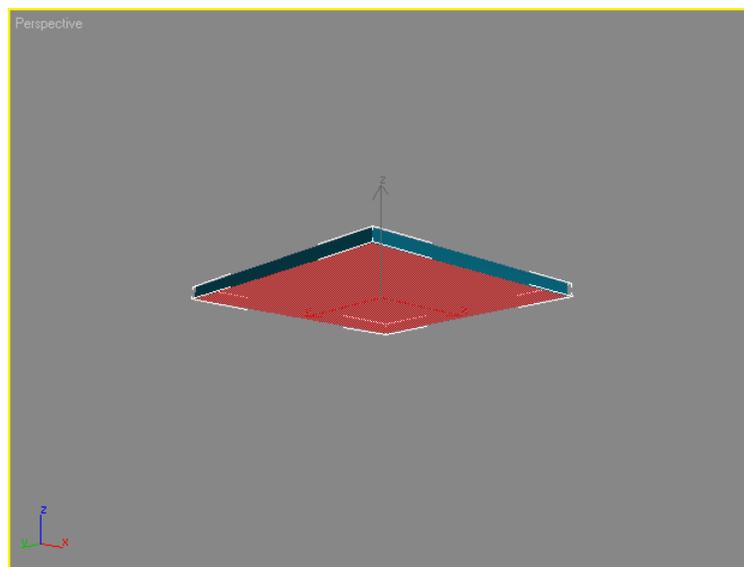
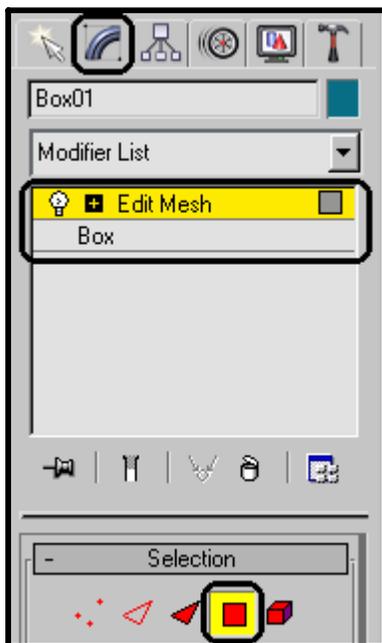
Hecho lo anterior, deberá aparecer en la pila del objeto, el modificador antes especificado, lo que significa que la transformación ha sido asignada de manera correcta.



Antes de aplicar cualquier transformación a un objeto, es importante situar el pivote de los objetos en el centro del mismo, ya que las transformaciones geométricas, (translación, rotación y/o escalamiento) que apliquemos a este objeto van referenciadas siempre con respecto de su pivote.

 Para no tener complicaciones al aplicar transformaciones geométricas, utilizaremos la herramienta **Punto Pivote Centro (Pivot Point Center)**, para que todos los objetos de nuestra escena tengan su correspondiente punto de pivote en su respectivo centro.

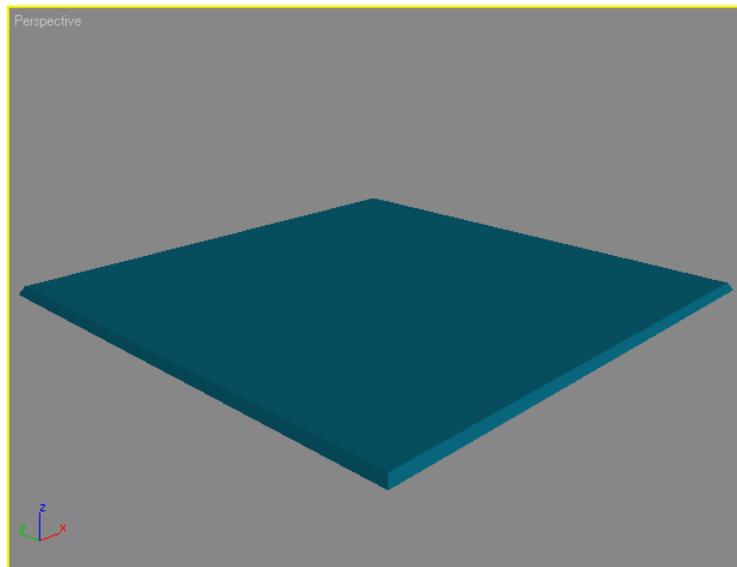
 Mediante el comando, selección de polígonos, escogemos el polígono de la base (polígono inferior), el cual debe resaltarse en color rojo.



 Habiendo seleccionado el polígono aplicaremos la herramienta **Seleccionar y Escalar**, oprimiendo el botón derecho del ratón sobre dicho comando, para acceder a la ventana de escalamiento e insertar los parámetros requeridos. De esta forma el escalamiento se realiza de manera exacta, a diferencia del escalamiento manual que es inexacto.

Los parámetros serán, para 'X' y 'Y' 102 unidades, es decir, que el polígono inferior lo estamos escalando al 102% o mejor dicho lo aumentamos en un 2%.

De esta forma finalmente obtenemos el primer talud del cuerpo de nuestra pirámide.



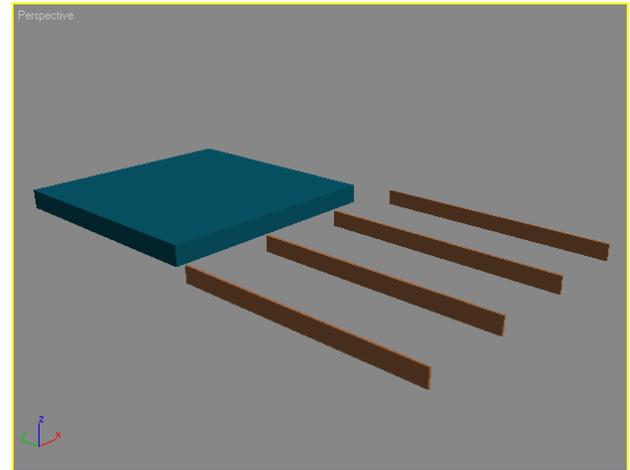
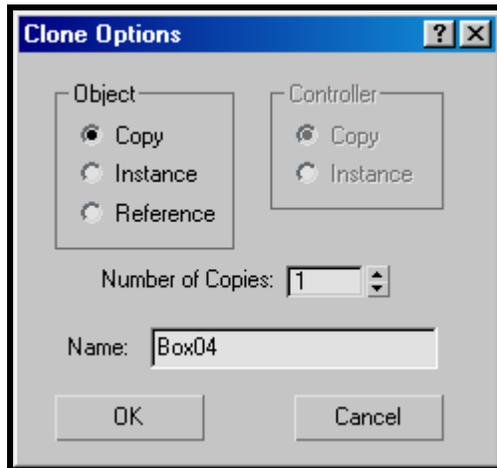
El segundo paso consistirá en obtener el tablero. Este también se obtendrá mediante la generación de una caja de las mismas dimensiones que la generada para el talud en cuanto a largo y ancho; lo que cambiaría es la altura que será de 3 unidades. De esta forma, la caja que se generará es de 30X30X3.

El siguiente paso consiste en realizar los cuatro tableros del cuerpo de la pirámide, que se encuentran en las cuatro caras laterales del objeto. Para generar los huecos de los tableros es necesario realizar una operación booleana de sustracción, es decir, tendremos que intersectar al tablero con otras cuatro cajas previamente generadas.

Para lograr esto, utilizaremos nuevamente el comando Crear, en la categoría de Geometría, y en la subcategoría de Objetos Compuestos (Compound Objects), seleccionamos el comando Boleano (Boolean), para generar la sustracción y de esta forma generar los tableros de la caja.

Los pasos que se deben de seguir para obtener el tablero del cuerpo son los siguientes:

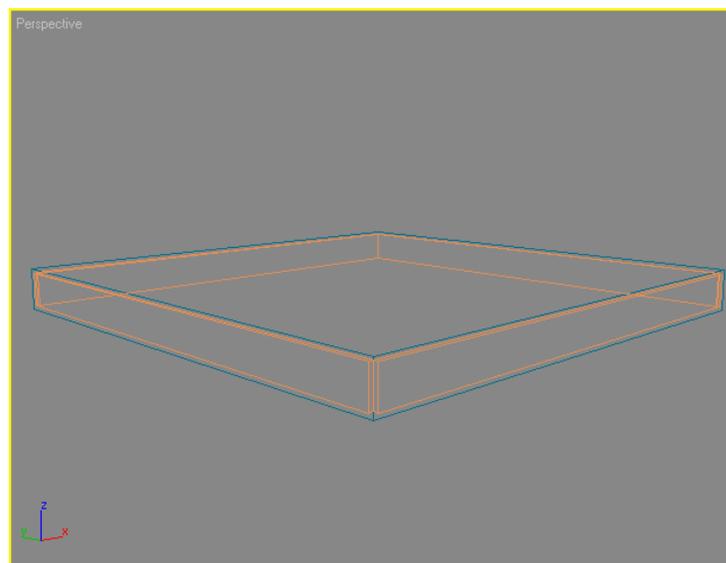
1. Generar una caja de dimensiones 30X30X3
2. Generar cuatro cajas de dimensiones 29 metros de largo, 0.5 de ancho y 2.0 de alto. (En este caso utilizamos la opción **Clonar Objetos**, ya que podemos generar una sola caja con estas dimensiones, y posteriormente clonar 3 cajas adicionales).



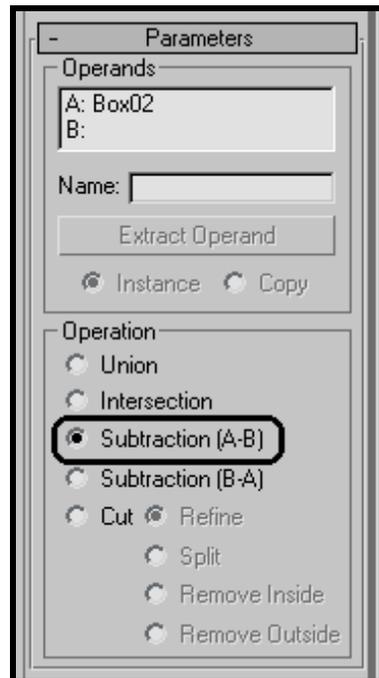
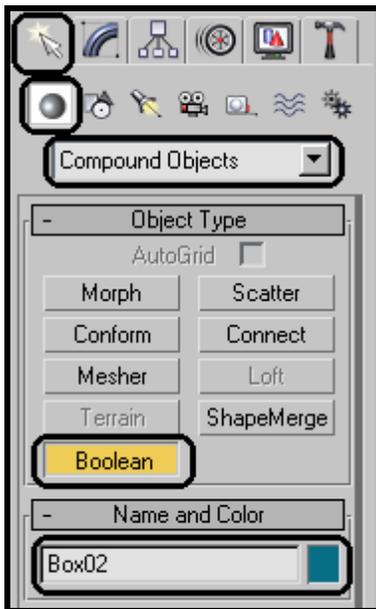
3. Cada una de las cuatro cajas generadas deberá ser alineada con su correspondiente cara del tablero, de tal forma que tomen la posición del hueco que se realizara.



Para ello, utilizamos la opción de **Alinear Objetos**. Para alinear un objeto con respecto de otro, es oportuno mencionar que el objeto seleccionado es el objeto de referencia y el objeto que se selecciona después de seleccionar el comando Alinear es el objeto que se va a alinear. Este proceso varía dependiendo el puerto de vista donde se realice la alineación de los objetos, además hay que tener en cuenta que la alineación se realiza utilizando el sistema de coordenadas del puerto de vista que es diferente del sistema coordenado general.

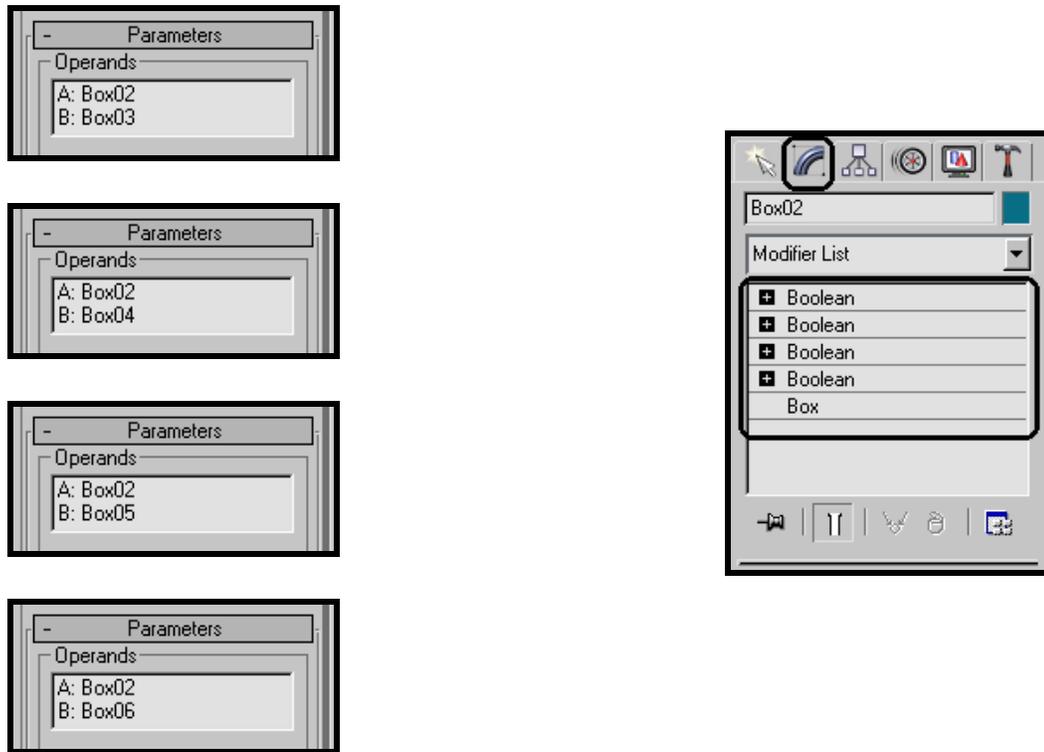


4. A continuación realizaremos las cuatro operaciones booleanas de sustracción. Únicamente tenemos que seleccionar la caja principal y posteriormente acceder a la categoría Geometría, en la subcategoría Objetos Compuestos, y elegimos la opción del comando **Boolean**.
5. Elegimos dentro de las opciones de **Operación (Operation)**, la sustracción (A-B). Lo anterior quiere decir que se va a obtener como resultado, la sustracción del operando A (caja principal Box02) con el objeto seleccionado en el siguiente punto, que en este caso será el operando B (caja secundaria Box03).
6. Debemos indicar cual es el operador con el que se realizará la operación booleana. Este operador, es el segundo en la operación booleana, ya que el primer operando es la caja principal. Así que para seleccionar el segundo operando, hay que elegir Pick Operand B, y de esta forma con el puntero del ratón debemos indicar cual es el segundo operando (operando B).



Realizaremos cuatro operaciones booleanas, donde el operando A es el tablero (Box02), y el operando B son las cajas secundarias que servirán para hacer los huecos al tablero (Box03, Box04, Box05, Box06).

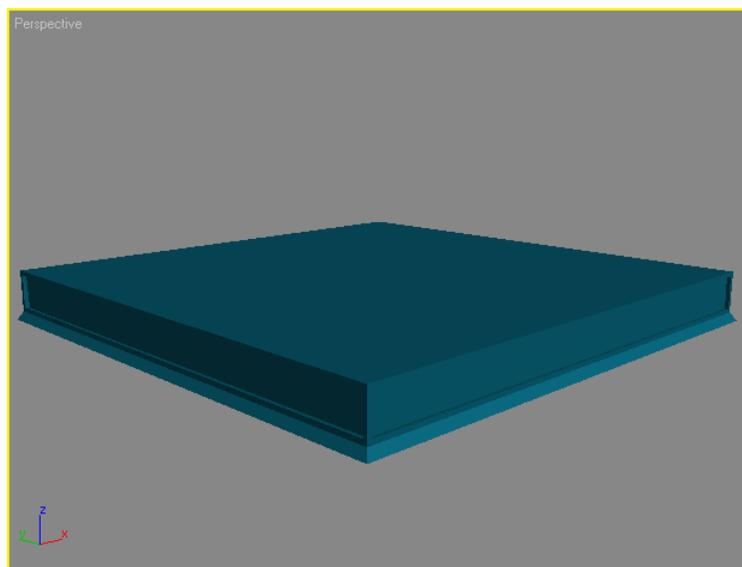
Una vez efectuada la operación, tenemos como resultado un solo objeto, con los huecos correspondientes a los tableros. En la pila del comando Modificar encontramos indicadas las cuatro operaciones booleanas.



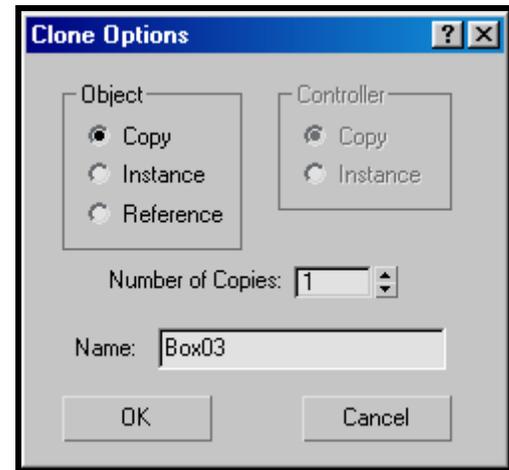
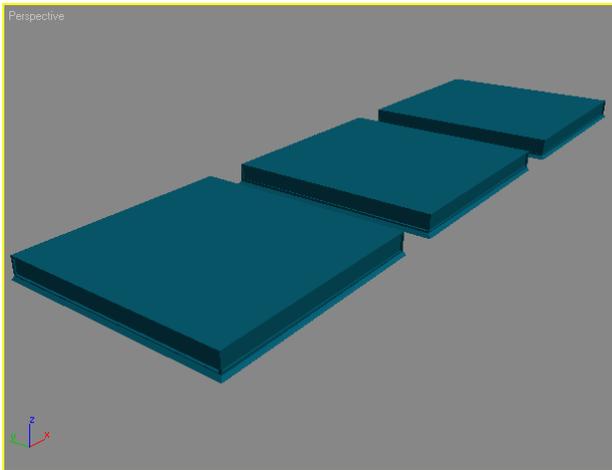
Habiendo realizado los seis pasos que se explicaron con anterioridad, hemos obtenido el tablero del primer cuerpo de la pirámide.

En éste momento contamos con el talud y el tablero del primer nivel de la pirámide. Pero dichos elementos no se encuentran en su posición correcta, para resolver éste problema, debemos alinear ambos objetos utilizando la herramienta **Alinear Objetos**.

En la vista de Perspectiva alinearemos ambos elementos, colocando el talud en la parte inferior del tablero. Finalmente, tenemos que alinear el centro del tablero con respecto al centro del talud en los ejes `X` y `Y`; y en el eje `Z` alinearemos la parte mínima del tablero con respecto del máximo en Z del talud. A continuación se muestra el resultado de la alineación:

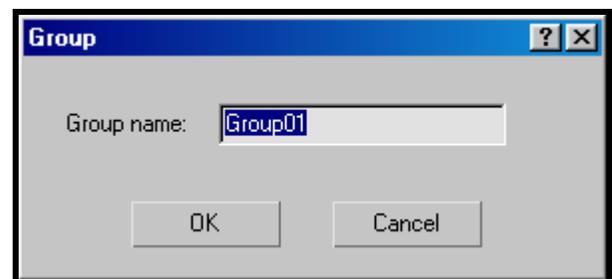


Finalmente para tener el cuerpo completo de la pirámide, lo único que resta es clonar el talud y el tablero haciendo uso de la herramienta **Clonar Objetos** con la finalidad de crear otros dos objetos idénticos.

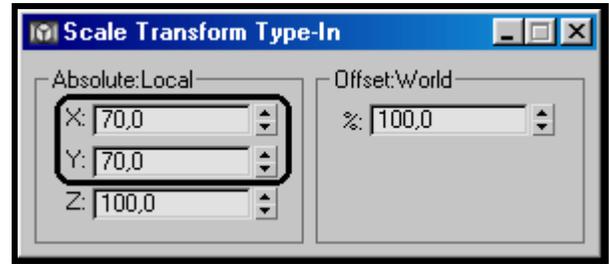
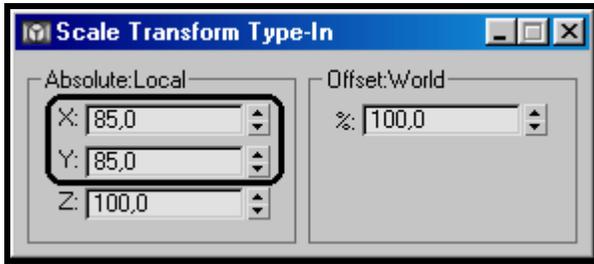


Habiendo realizado la clonación del primer nivel, contamos con tres pares, integrados por talud y tablero. Seleccionaremos un par para escalarlo y posicionarlo arriba del primer nivel para formar el segundo y tercer nivel, para ello utilizamos la herramienta **Seleccionar y Escalar**.

Antes de realizar la transformación geométrica de escalamiento, se recomienda agrupar al talud con su respectivo tablero por medio del menú **Grupo**. De esta manera los parámetros de escalamiento que asignemos, afectarán a los dos objetos como si fueran uno solo.

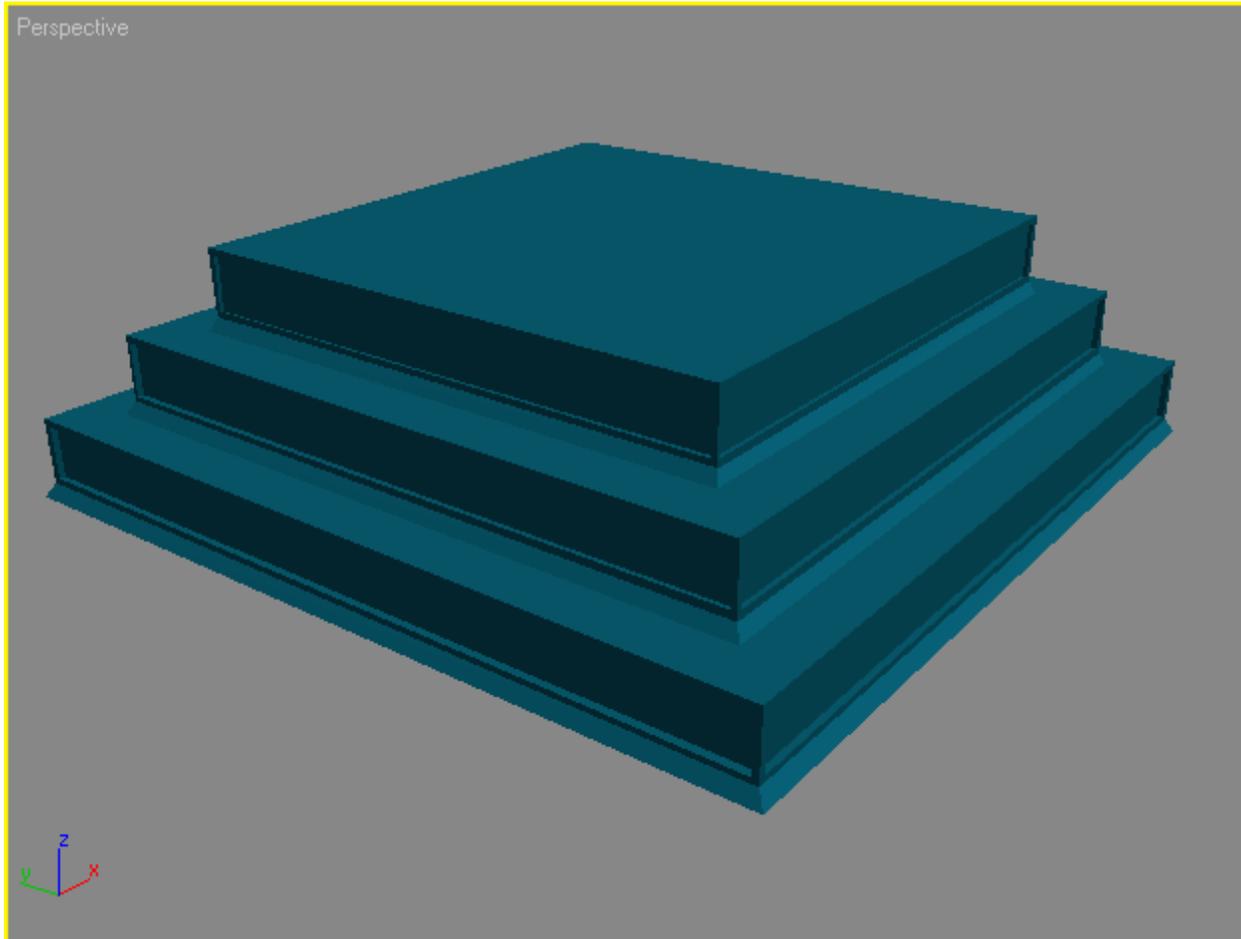


Aplicaremos un factor de escalamiento igual a 0.85 pero únicamente en los ejes `X` (largo) y `Y` (ancho), para no afectarlo en su altura, y posteriormente seleccionamos otro par para aplicarle un factor de escalamiento igual a 0.70, de igual manera solamente en los ejes `X` y `Y`, y el otro no lo afectamos, y de esta manera logramos tener tres pares de objetos con la misma altura pero con dimensiones escaladas proporcionalmente en largo y ancho.



A continuación tenemos que alinear cada uno de los grupos, uno encima de otro, utilizando la herramienta **Alinear Objetos**, para obtener el resultado que se muestra a continuación.

Entonces el resultado final del cuerpo de la pirámide quedaría de la siguiente manera:



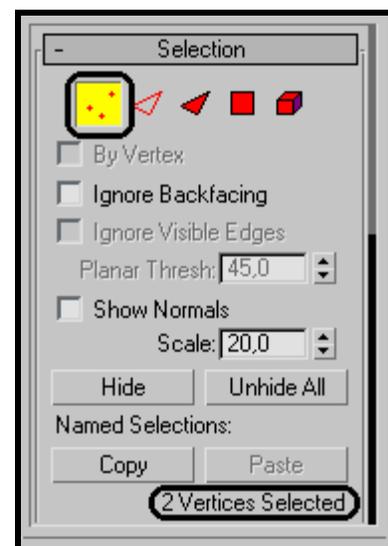
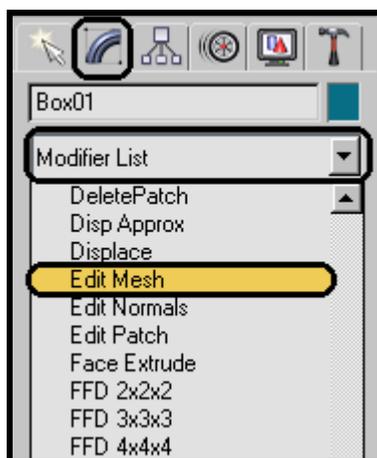
ESCALERAS

Ahora realizaremos la generación de la escalera, la cual esta integrada por tres partes, los escalones por un lado y por otro lado por los barandales y la rampa. La generación de los barandales van de la mano para la generación de la rampa, para este caso utilizaremos la generación de cajas, mientras para el caso de los escalones lo que utilizaremos es la generación de estos por medio de una línea.

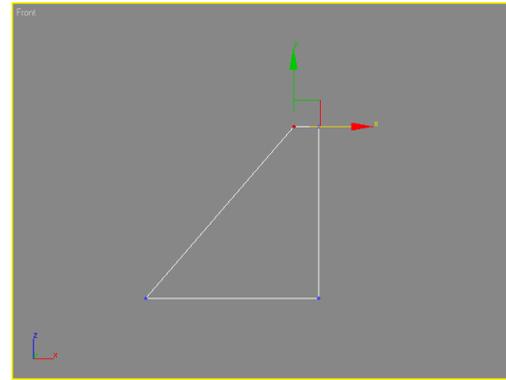
Entonces primeramente realizaremos los barandales teniendo en cuenta dos aspectos, que son la altura que en este caso vamos a considerar la altura del cuerpo de la pirámide que esta compuesto por tres taludes y tres tableros, donde cada talud tiene una altura de 0.5 mts., y cada tablero tiene una altura de 2.5 mts., por lo que la altura de las escaleras deben ser de 9.0 mts. Y el otro aspecto importante es la inclinación o pendiente de los barandales, ya que dicha pendiente es la referencia de inclinación de la rampa así como de los escalones y entonces para lograr una inclinación aproximadamente de 45° el largo de los barandales debe ser igual a 9.0 mts.. El ancho de los barandales es una dimensión arbitraria, en este caso va a ser igual a 1 metro.

Para generar el barandal seguiremos los siguientes pasos:

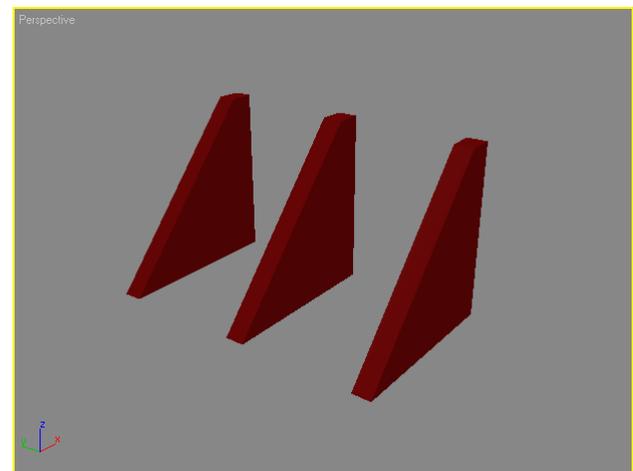
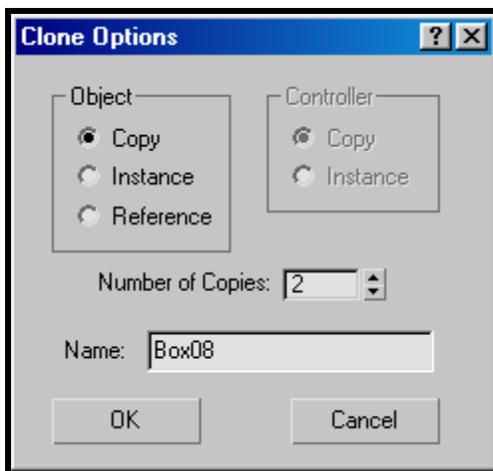
1. Generar una caja de dimensiones (1, 9, 9) de largo, ancho y alto respectivamente.
2. Haciendo uso del comando Modificar, le asignaremos a la caja el modificador **Malla Editable**, para poder manipular la caja y deformarla como sea requerido.
3. Dentro de las opciones de Malla Editable, elegimos la opción de selección por **Vértices (Vertex)**, para seleccionar un par de ellos ubicados en la parte superior de la caja.



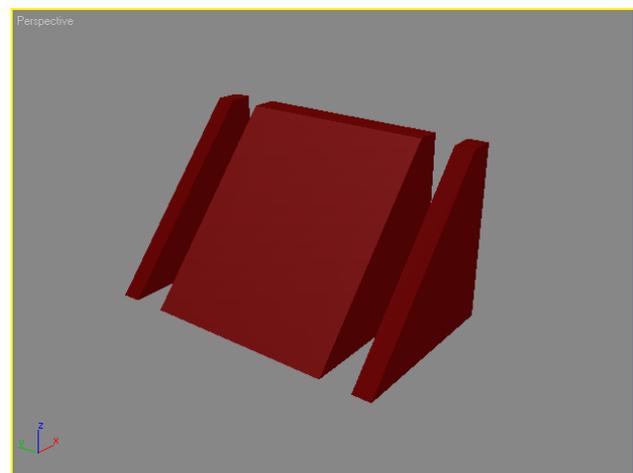
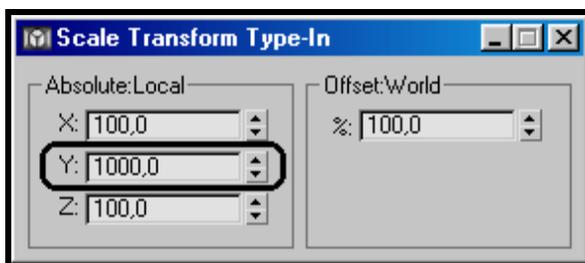
4.  Seleccionando los dos vértices superiores, utilizaremos la herramienta de translación, **Seleccionar y Mover** para trasladar los vértices en dirección hacia los otros dos vértices superiores, destacando que únicamente la translación se realiza sobre el eje 'Y', de tal forma que obtengamos una pendiente arbitraria, pero cercana a los 45°.



5. Ahora clonaremos el objeto que obtuvimos dos veces, uno de ellos servirá para tener el otro barandal, y el otro objeto será la rampa de la escalera.

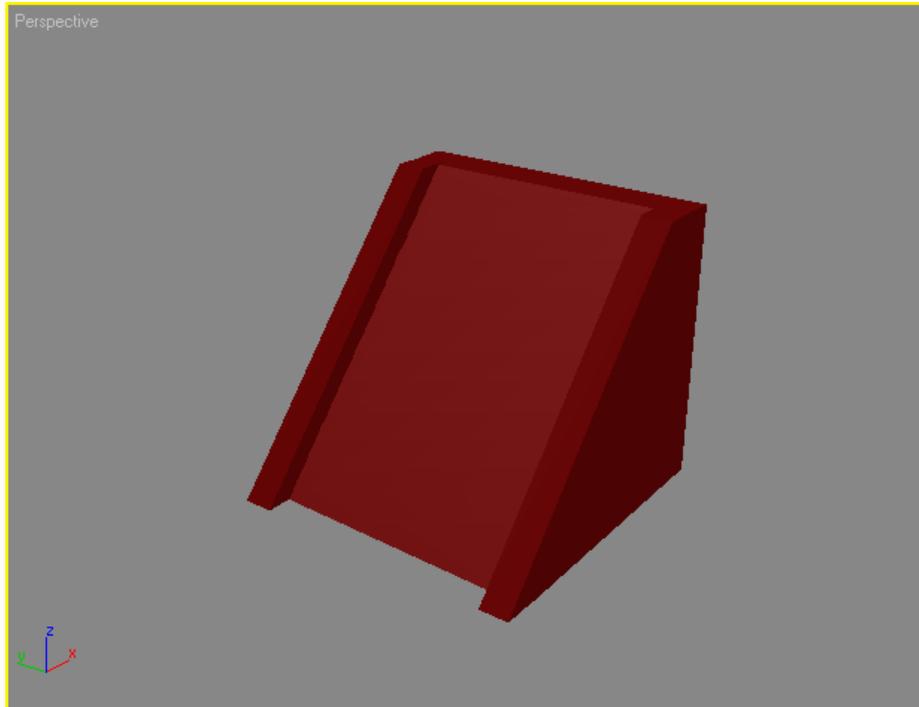


6. Le aplicaremos un escalamiento del 1000% sobre el eje 'Y', al elemento clonado que servirá de rampa, para que ésta sea 10 veces más grande que los barandales. Para ello utilizamos la herramienta **Seleccionar y Escalar**.



7. Los tres objetos tienen las dimensiones correctas, pero no cuentan con las posiciones adecuadas. Primeramente realizaremos una pequeña translación del objeto de la rampa sobre el eje `X`, desplazando hacia atrás con referencia de los barandales, de tal forma que estos últimos sobresalgan un poco respecto de la rampa.

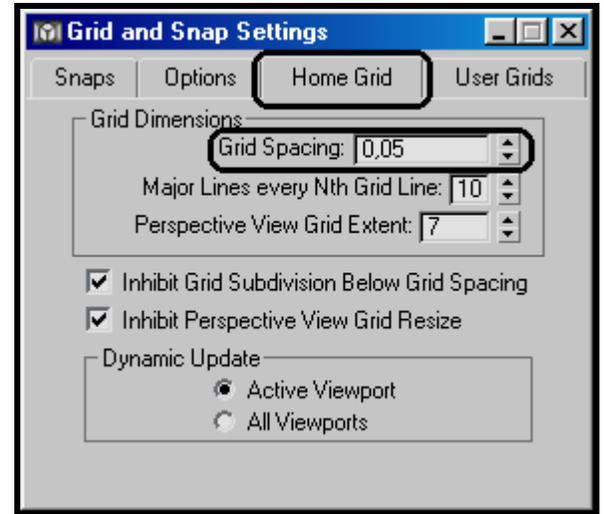
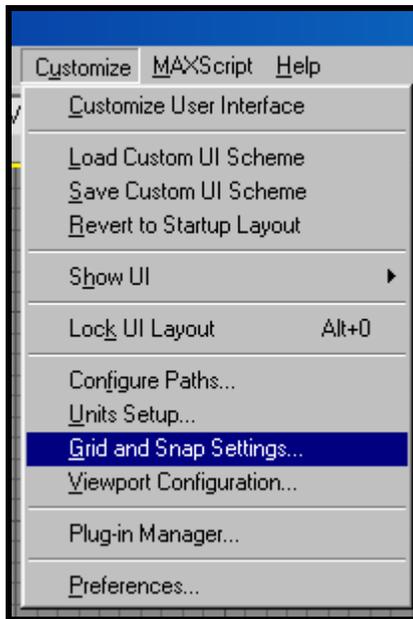
Únicamente faltaría alinear los tres objetos entre sí sobre el eje `Y`, ya que sobre los ejes `X` y `Z` se encuentran alineados al haber sido clonados sobre el eje `Y`. Utilizando la herramienta **Alinear**, alinearemos los tres objetos para que queden juntos.



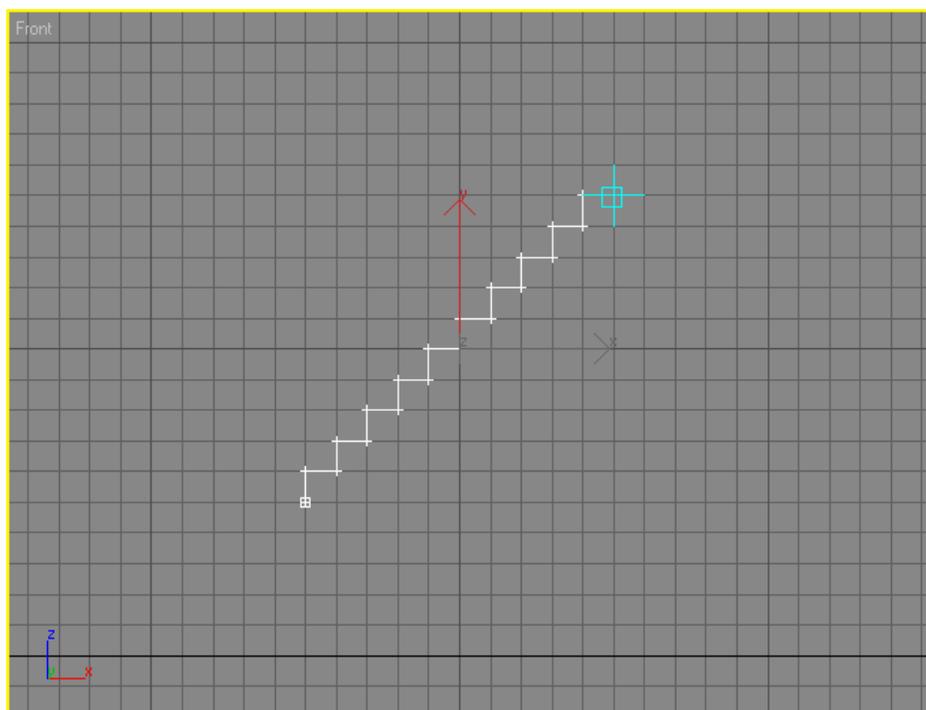
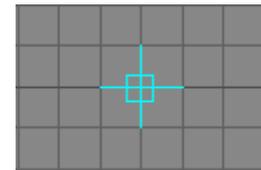
El siguiente paso es realizar los escalones, para esto generamos una línea apoyados en la cuadrícula de un puerto de vista, para generar el contorno de los escalones como si se estuvieran viendo su perfil, y apoyados al mismo tiempo en la rampa.

 A la hora de generar la línea, es importante utilizar la herramienta **Snap Toggle**, para restringir el trazado de la línea únicamente sobre la cuadrícula. Observamos que el puntero del ratón aparece iluminado en cualquiera de los vértices de la cuadrícula, esto indica, que solamente en dichos puntos podremos realizar el trazado de la línea.

Para poder adaptar la cuadrícula o el grid de un determinado puerto de vista (viewport) seguiremos las siguientes pasos, en primer lugar acceder dentro de la barra de menús, al menú **Customize**, posteriormente acceder en la opción **Grid and Snap Settings....** Dentro de esta opción, aparecerá una ventana con varias pestañas, debemos seleccionar la pestaña **Home Grid**, y dentro de esta opción manipularemos la opción de **Grid Spacing**, y le asignaremos un valor igual a 0.05.

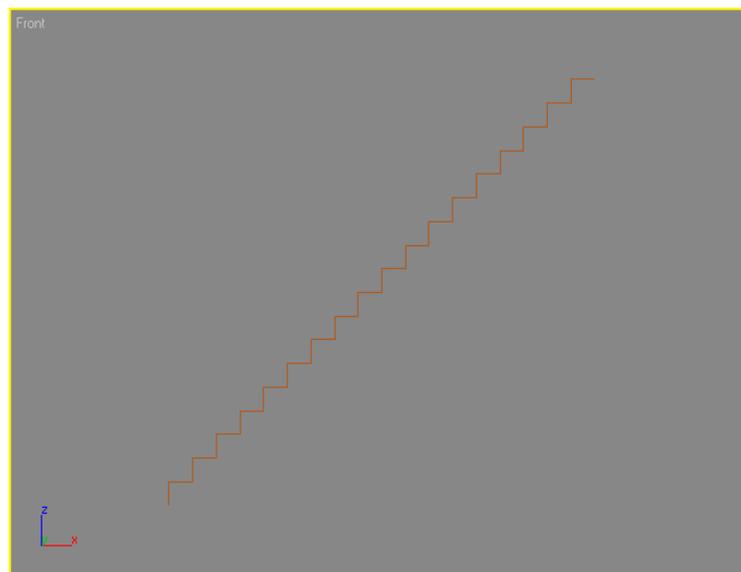
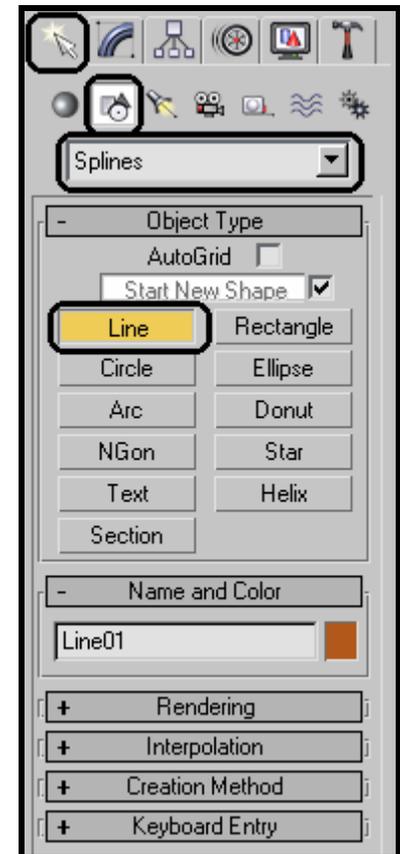


Ahora para el caso de la herramienta Snap Toggle seleccionamos el correspondiente icono, y podremos observar que el puntero del ratón aparece iluminado el correspondiente vértice del grid por donde tracemos la respectiva línea.

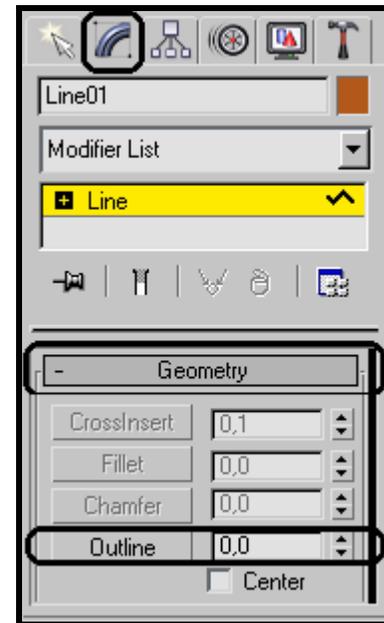
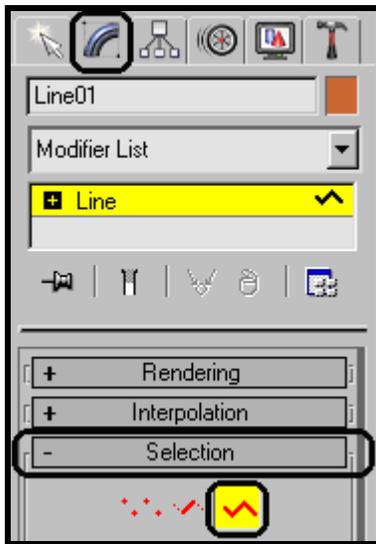


De manera precisa se explicarán, los pasos para generar una línea:

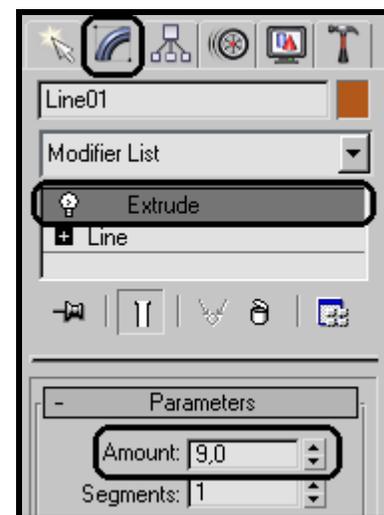
1. Elegir la categoría de objetos mediante los botones de la parte superior del panel de comandos Crear Formas. En algunas categorías hay que designar la subcategoría que en este caso corresponde a la categoría Formas (Shapes), y dentro de ésta, la subcategoría Splines (Splines).
2. Presionar un botón para seleccionar un tipo de objeto, en este caso Line (Línea).
3. Para crear el objeto se debe arrastrar el ratón y elegimos un puerto de vista, mientras se realiza esto, el objeto se sitúa en la cuadrícula activa del puerto de vista. Cuando se suelta el ratón se establece un segmento de la línea. Para definir otro segmento de la misma línea, se vuelve a presionar el botón derecho del ratón y se arrastra el ratón hacia arriba o abajo y llegado el punto deseado, se presiona el botón derecho del ratón; y así sucesivamente hasta determinar todos los segmentos que integra dicha línea, y para concluir se debe presionar el botón derecho del ratón.



Habiendo generado el contorno de los escalones, tenemos que aplicarle la opción **Outline**, para asignarle un determinado grosor, y que los escalones generados tengan volumen. Para acceder a esta opción del objeto línea, accedemos al comando **Modificar** y en la opción **Selección** elegimos el comando **Spline**. Dentro de las propiedades accedemos a la opción **Geometría** para seleccionar **Outline**, cuyo valor será 0.025.



Lo que resta es extruir o “darle altura” a la línea par completar el proceso de generación de los escalones. Para asignar este modificador, elegimos el comando **Modificar** y agregando en la Lista de Modificadores la propiedad **Extrude**, la cual nos permitirá darle volumen a la línea. El volumen dependerá del parámetro **Amount** que en nuestro ejemplo tiene un valor de 9.



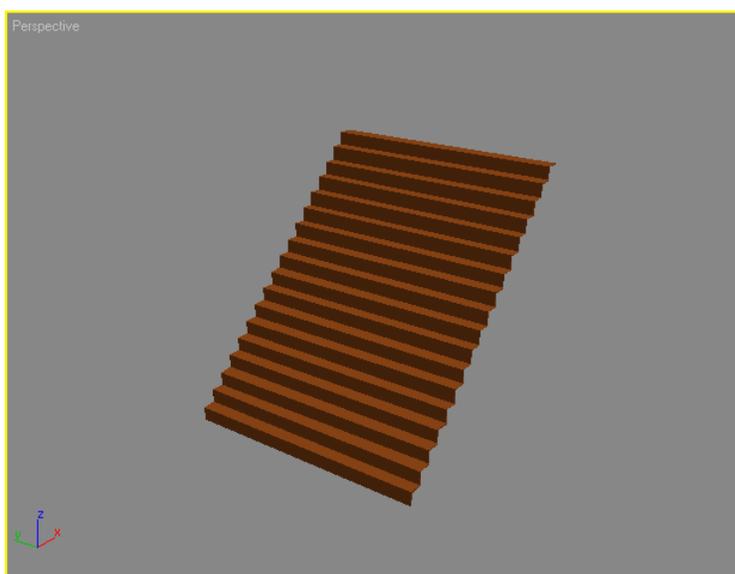
La línea ya fue generada pero, puede existir la posibilidad que la generación de la línea que simulan los escalones de la escalera no concuerda con la altura de la rampa y los barandales y ni con la pendiente de estas. Así que debemos realizar algunos ajustes de manera manual y aproximada, utilizando algunas herramientas para poder adaptar bien los escalones con las rampas y los respectivos barandales.

Para esto nos podemos auxiliar de la herramienta de **Seleccionar y Escalar**, para que por medio de esta lograr adaptar los escalones con la altura de la rampa y barandales, así como con la

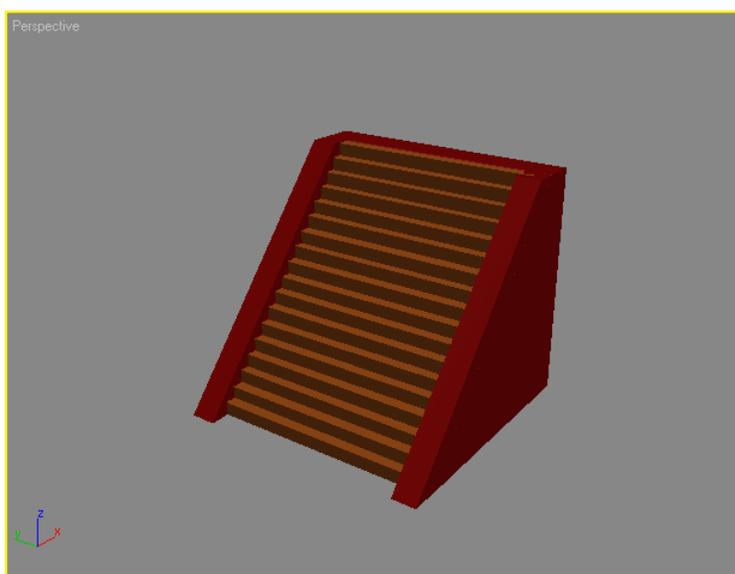
pendiente de estas. Considerando algún puerto de vista donde se puedan observar simultáneamente la rampa con los barandales y los escalones de manera lateral, en el caso de la altura podemos escalar los escalones en el eje `Y` para adaptar de manera aproximada la altura de los escalones con la rampa; y para el caso de tratar de igualar también de manera aproximada la pendiente de los escalones, podemos escalar estos en el eje `X`.

Por ultimo podemos utilizar la herramienta de **Seleccionar y Mover** y **Alinear Objetos** para colocar a los escalones en la posición final de estos con respecto a la rampa.

Finalmente los escalones los podemos visualizar de la siguiente manera:



Y las escaleras completas quedan de la siguiente manera:



TEMPLO

Desafortunadamente los templos de las pirámides teotihuacanas fueron destruidos después de la conquista de México, por los españoles, así que la única información que conseguimos fue en la maqueta del museo de sitio de Teotihuacan.

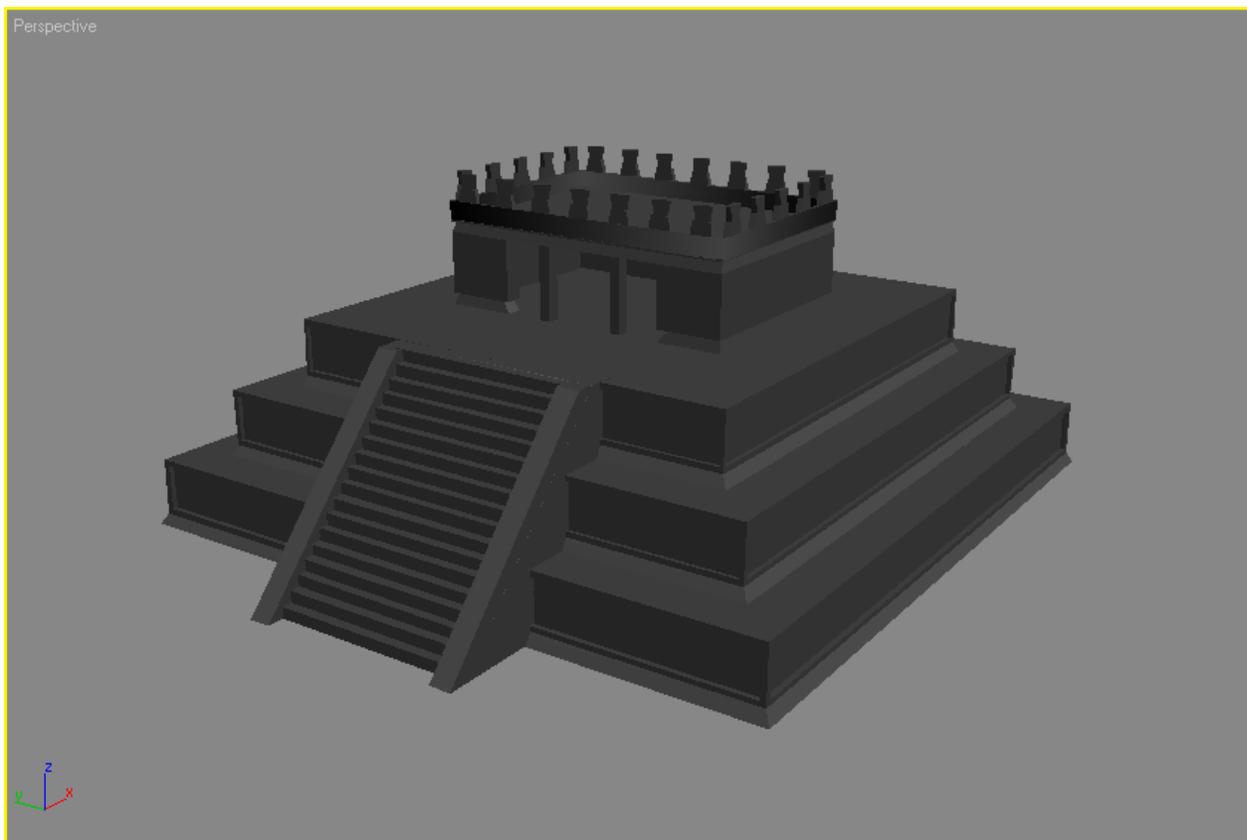
Existen muchas formas para construir un objeto tridimensional. En el caso de nuestro templo, seguimos la misma idea que utilizamos para la construcción de los escalones.

En la vista superior del objeto, trazamos su contorno y le dimos grosor, para que al momento de extruirlo o “levantar” el contorno, se genere un objeto 3D con volumen.

El techo y las columnas de la puerta fueron generados mediante cajas de diferentes dimensiones. El zoclo y el antetecho fueron generados con el mismo contorno del templo, pero se modificaron las posiciones de los vértices superiores aplicando la propiedad de **Malla Editable** para lograr las pendientes que queríamos, las cuales por cierto fueron arbitrarias debido a la carencia de información.

Por último mencionaremos que existen diferentes dimensiones para los templos a lo largo de la zona arqueológica. Lo que hicimos fue utilizar el mismo templo y únicamente aplicarle su escalamiento respectivo para que adquiriera las dimensiones necesarias.

En resumen podemos decir que para la creación de objetos geométricos, la metodología aplicada para el resto de la arquitectura de la zona arqueológica, queda totalmente representada con la explicación anteriormente definida.



VII.1.1.2 PREPARAR LA GEOMETRÍA PARA LA APLICACIÓN DE TEXTURAS.

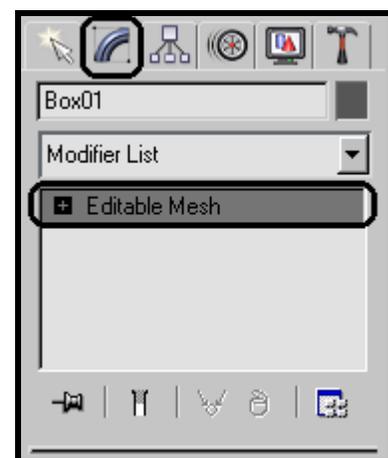
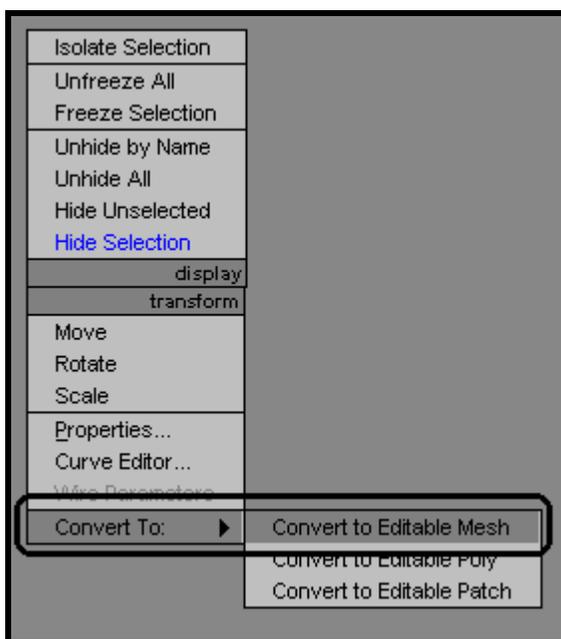
Hasta este momento contamos con nuestro modelo terminado. Lo que resta es preparar al objeto para la aplicación de texturas, y esto lo realizaremos manipulando dicho archivo con la ayuda de un par de herramientas, que son: **Convert to Editable Mesh (Convertir a Malla Editable)** y **Attach List (Lista de Ligas)**, que a continuación se explicarán cada una de ellas.

- Convert to Editable Mesh (Convertir a Malla Editable)

El convertir toda la geometría en malla editable nos ayuda a manipular de mejor manera nuestra geometría, ya que de esta forma la geometría creada ya no va a ser referencia de acuerdo con los tipos de objetos con la que fue generada, sino que ahora se va a considerar una malla editable, la cual nos va a permitir una mejor manipulación de esta, sobre todo para su manejo a la hora de la aplicación de las texturas, ya que nos va a permitir seleccionar vértices, caras, polígonos y elementos, según sea nuestra preferencia y necesidad.

Para convertir nuestra geometría en una malla editable, hay que seguir los siguientes pasos:

1. Seleccionar todos los objetos de nuestro proyecto. Una vez seleccionados, sus contornos aparecerán resaltados con color blanco. Presionaremos el botón derecho del Mouse para desplegar el menú de “despliegue y transformar”.
2. En la sección “transformar” (transform) seleccionamos “Convertir a:” (Convert to) y después “Convertir a Malla Editable” (Convert to editable Mesh).



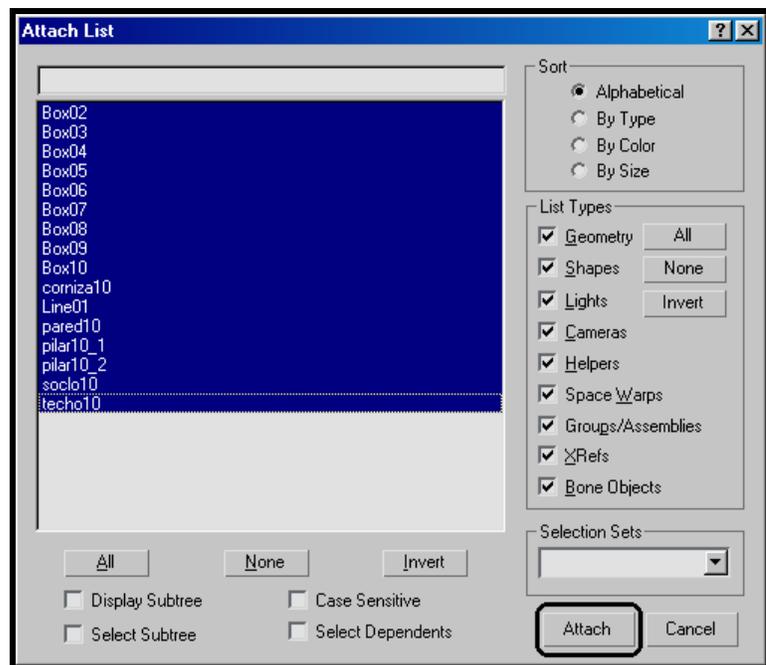
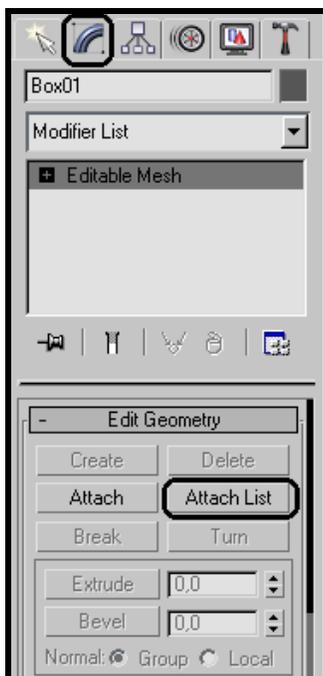
- Attach List

El objetivo de utilizar la herramienta de Attach List es con la finalidad de que obtengamos un solo objeto el cual va a ser el que finalmente queremos aplicarle la textura.

Entonces podemos partir del hecho de que tenemos varios objetos pero estos tiene la característica particular que cada uno de estos objetos son una malla editable, así que cuando apliquemos la herramienta Attach de un objeto con varios objetos, es ligar varios objetos a un objeto base o objeto principal, para que de esta forma podamos obtener finalmente un solo objeto el cual es un objeto que es malla editable. Un aspecto importante es considerar que al objeto base, no se le ha aplicado previamente una transformación geométrica.

Para realizar todo el proceso anterior, se deben de seguir los siguientes pasos:

1. Seleccionar un objeto, el cual debe tener la característica.
2. Acceder al panel de comandos Modificar (Modify), y dentro de la opción de Editar Geometría (Edit Geometry), presionar el botón que indica la opción **Attach List**.
3. Aparecerá una ventana en la cual se enlistan todos los elementos que conforman a nuestra geometría, excepto el elemento que previamente seleccionamos. Lo cual quiere decir que podemos ligar a este elemento con los elementos que aparecen en dicha lista, entonces se seleccionan todos los elementos de dicha lista y se selecciona el comando Attach.
4. Finalmente podemos obtener un solo elemento o en su defecto varios elementos, el cual o los cuales son ya una malla editable, y están listos para aplicarles las texturas respectivas.



NOTA: Los objetos generados en 3D Studio Max y que pertenecen a un mismo mundo, no deben exceder los 10000 polígonos, ya que al ser exportados a Director 8.5, si los objetos exceden la cantidad mencionada, serán interpretados erróneamente.

VII.1.2 APLICACIÓN DE LAS TEXTURAS

VII.1.2.1 DISEÑO Y APLICACIÓN DE MATERIALES

Todos los modelos que tenemos, tienen algunas características en común, como su naturaleza (son mallas editables), y que les serán aplicadas mas de una textura a cada uno de ellos. Como ejemplo podríamos citar el de un cubo, cada una de las seis caras tendrá una textura diferente, por lo que contaríamos con seis texturas para ser aplicadas a sus respectivas caras.

Lo primero que realizaremos es numerar las seis caras, esta numeración se realiza de manera arbitraria, pero tiene que ver con el índice del material que le será asignado (ID Material).

Para trabajar correctamente la asignación de texturas, tenemos que generar un material Multi/SubObject, es decir que dicho material estará a su vez compuesto por materiales con un índice único.

En resumen dentro de la creación de los objetos que conformaran el modelo de la zona arqueológica, con apoyo del panel de comandos Crear, y de acuerdo con las herramientas que utilizaremos dentro de este panel, solamente haremos referencia a la creación de un ejemplo de todo el modelo que se genero para el paseo virtual, pero con la característica que desarrollando, en este caso una pirámide con talud y tablero así como con su templo, podemos hacer referencia a todas las herramientas que se aplicaron durante el desarrollo de todo el modelo, y con este ejemplo queda bien referenciado y explicado como se realizo la generación de todo el modelado.

Lo anterior lo podemos desglosar que la pirámide esta conformada por tres partes que son: cuerpo, escaleras y templo. Y la generación de cada una de estas va a estar integrada de la siguiente manera:

Generar el **archivo de materiales** utilizando la herramienta **Material Editor**
Asignar el **archivo de materiales** utilizando la herramienta **Material Editor**
Asignar el **ID (identificador)** utilizando el panel de comandos **Modificar**

Aplicar el modificador **UVW Map** utilizando el panel de comandos **Modificar**
Aplicar el comando **Collapse** utilizando el panel de comandos **Utilidades**

PROPIEDADES BÁSICAS DE LOS MATERIALES

Los materiales están definidos por una serie de parámetros que determinan su aspecto final, una opción interesante es la sombra, que determina la forma en que el material brilla bajo la incidencia de la luz, en nuestro caso utilizaremos la opción **Blinn**. Además tenemos que hacer mención que las texturas serán incluidas en la componente difusa del material para que puedan ser interpretadas correctamente.

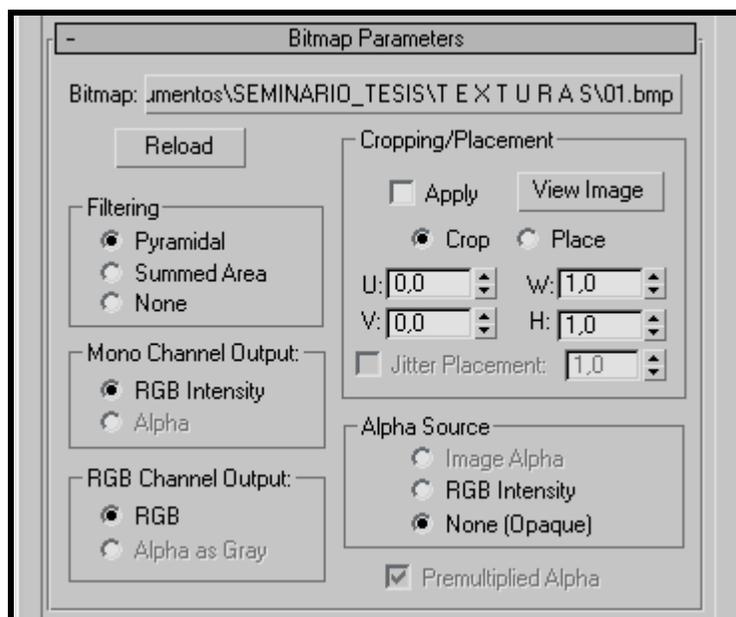
CANALES DE LOS MAPAS

Existen tres canales para la asignación de mapas de bits, los cuales se explican a continuación, aunque nosotros únicamente hayamos utilizado la componente difusa.

- **Difuso**. Aplica el mapa sobre el objeto para que la imagen se coloque en torno al objeto según las coordenadas de mapeado.
- **Ambiental**—Aplica el mapa seleccionado sobre las zonas del objeto que queden sin iluminar.
- **Especular**—Aplica el mapa seleccionado sobre las partes que reciban luz especular.

IMAGEN BITMAP

Las imágenes que serán incluidas en la componente difusa del material, son bitmaps, o archivos de imagen que quedan definidas por su número de píxeles así como por su altura y anchura. El botón Imagen bitmap permite controlar en todo momento el nombre del archivo de imagen que se esté aplicando. En nuestro caso aplicamos únicamente imágenes con formato BMP.



MATERIALES

Los materiales describen cómo un objeto refleja o transmite luz. Dentro de un material, los mapas pueden simular texturas, diseños aplicados, reflexiones, refracciones, y otros efectos. (También los mapas pueden usarse como ambientes y proyecciones de luces). El Editor de Materiales es la herramienta que se utiliza para crear, alterar, y aplicar los materiales en una escena.

TIPOS DE MATERIALES

Los materiales crean un realismo mayor en una escena. Un material describe cómo un objeto refleja o transmite luz. Se asignan los materiales a objetos individuales o a selección de sets; una sola escena puede contener muchos materiales diferentes. Otros tipos de materiales entran en la categoría de Materiales Compuestos.

MATERIALES COMPUESTOS

Los materiales compuestos son aquellos que combinan otros materiales de alguna manera; combinan dos o más submateriales. Los materiales compuestos son similares a los mapas compuestos, pero existen a nivel material. Aplicando un material compuesto a un objeto crea un efecto compuesto que a menudo usa el mapping. Se cargan o crean materiales compuestos utilizando el Navegador Material/Map.

Los diferentes tipos de materiales crean diferentes efectos, se comportan de manera particular, o son proporcionados de tal de manera de combinar los múltiples materiales.

TIPO DE MATERIALES COMPUESTOS. MATERIAL MULTI/SUBOBJECT

Permite asignar distintos materiales a un mismo objeto.

Contiene una lista de materiales que pueden asignarse a diferentes partes de la geometría de un mismo objeto. El método consiste en seleccionar las diferentes caras del objeto en el nivel subobjeto y asignarles distintos números ID de material. Cada material corresponderá a un único ID de material.

A continuación se mencionan las características principales del material Multi/SubObject:

- El material Multi/SubObject asigna múltiples materiales a un solo objeto, basado en los valores de ID de los materiales.
- Los IDs del submaterial no dependen del orden de la lista, y se pueden introducir nuevos valores de ID en la columna correspondiente.

VII.1.2.1.1 Crear el archivo de materiales

Con el botón Obtener material  del editor de materiales accedemos al Visor de materiales / mapas, desde el cual podremos escoger materiales y/o mapas de diferentes bibliotecas o materiales nuevos que se encuentren en la escena y asignarlos a la ranura activa.

En general, cuando se crea un nuevo material y se aplica a un objeto, se siguen estos pasos:

1.  Seleccionar **Get Material** en la barra de herramientas del Editor de Materiales. El Navegador Material/Map es desplegado. O bien elegir en una hendidura de muestra activa, e introducir un nombre para el material que se está a punto de diseñar.
2. Escoger el tipo de material presionando doble el botón derecho del ratón en la lista tipo de material, o arrastrar el material a una hendidura de muestra. El Editor de Materiales ahora despliega los controles para el nuevo material.
3. Asignar los mapas a los componentes que se quieren mapear, y ajustar sus parámetros.
4. Guardar el material.

VII.1.2.1.2 Asignación del id a la geometría

A toda nuestra geometría es necesario asignarle a cada polígono el id del material asignado, por esto es necesario identificar de la lista de nuestro material Multi/SubObjet que id le corresponde a cada polígono o a cada elemento geométrico de acuerdo con nuestro diseño de asignación de texturas, por lo que podemos decir que tenemos un solo material asignado a nuestra geometría pero en realidad tenemos muchos materiales de acuerdo a la lista por lo que cada material se identifica con el id el cual se asigna a los polígonos de la geometría.

VII.1.2.1.3 Asignar el archivo de materiales

Una vez diseñados las texturas y almacenadas dentro del material Multi/Subobjeto, es necesario su aplicación a los objetos modelados, para esto utilizamos el botón Asignar materiales del editor de materiales, accederemos al Visor de materiales/mapas, desde el cual podremos escoger materiales y/o mapas de diferentes bibliotecas o materiales nuevos que se encuentren en la escena y asignarlos a la ranura activa.



Para asignar el material a un objeto de la escena, lo haremos pulsando al botón **Asignar material a la selección** o arrastrando directamente el material desde la ranura hasta el objeto.

En este momento tenemos el material Multi/ SubObject generado, además de tener cada una de las caras del modelo numeradas con un ID diferente para la asignación del material que corresponda a dicho ID.

VII.1.2.2 MODIFICADOR UVW MAP

Aplicando las coordenadas de mapeo a un objeto, el modificador UVW Map controla cómo mapear y procesar los materiales que aparecen sobre la superficie de éste. Las coordenadas de mapeo especifican cómo los bitmaps son proyectados hacia un objeto. El sistema de coordenadas UVW es similar al sistema de coordenadas XYZ. Los ejes U y V de un bitmap corresponden a los ejes X y Y. El eje W corresponde al eje Z, y es generalmente usado para los mapas procesados.

Se utiliza el modificador UVW Map para:

- Aplicar uno de los siete tipos de coordenadas de mapeo a un objeto sobre un canal de mapeo en específico.
- Transformar el gizmo de mapeo para ajustar la colocación del mapa.
- Aplicar las coordenadas de mapeo a un objeto sin coordenadas de mapeo.
- Aplicar el mapeo al nivel del subobjeto.

Para aplicar de manera correcta las texturas a los objetos modelados, será necesario utilizar el modificador llamado **UVW Mapping**, el cuál se explica a grandes rasgos a continuación.

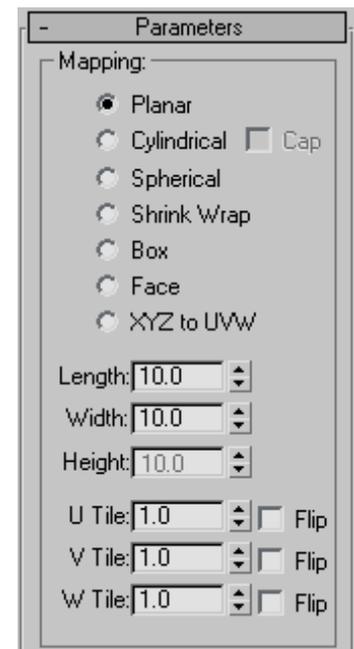
En el Modificador Pila, en el Nivel del Gizmo se habilitan las transformaciones del gizmo, es decir, se puede mover, escalar, y rotar el gizmo en los viewports para posicionar el mapa.

- **Mapping (Mapeo).** Determina el tipo de coordenadas de mapeo a usar. Los diferentes tipos de mapeos son distinguidos por cómo el mapa se proyecta geoméricamente hacia el objeto y cómo la proyección interactúa con las superficies del objeto. Éste puede ser Planar, Cilíndrico, Esférico Shrink Wrap, Box, Face o XYZ to UVW.

- **Length, Width, Height (Largo,Ancho,Alto).** Especifica las dimensiones del gizmo UVW Map. La escala predefinida del icono del mapping está definido por la dimensión más grande del objeto cuando se aplica el modificador.

- **U Tile, V Tile, W Tile (Repetición U, Repetición V, Repetición W).** Permite especificar los valores para las dimensiones del UVW map, para el tiling de la imagen. Éstos son valores punto-flotante; se pueden animar estos valores para cambiar el tiling del mapa con el tiempo.

- **Flip.** Invierte la imagen sobre el eje dado.



VII.1.2.2.1 Asignar el modificador UVW Map

El siguiente paso es saber de que forma va a ser aplicada la textura y en que posición, así como el número de veces que se repetirá si es que así sucede, la opción por defecto de repetición es uno. Para hacer todo lo anterior es necesario utilizar el modificador llamado UVW Mapping.

Se pueden controlar el tipo de coordenadas de mapeo y la colocación del gizmo de mapeo para cada bitmap en un material que usa múltiples bitmaps para asignar los canales de mapeo explícitos a los bitmaps. En el Editor de Materiales se asigna un número de canal diferente a cada mapa, entonces se agregan múltiples modificadores UVW Map al modificador pila del objeto, cada modificador UVW Map es colocado en un canal de mapeo diferente. Para cambiar el tipo de mapeo o la colocación del gizmo para un bitmap en particular, se selecciona uno de los modificadores UVW Map en el modificador pila y cambiar los parámetros.

Para aplicar el modificador UVW Map se siguen los siguientes pasos:

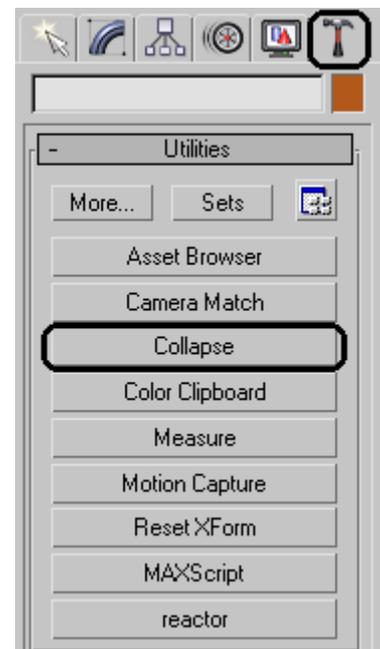
1. Asignar un material de mapeado al objeto.
2. En el panel Modificar, agregar el modificador UVW Map desde la lista de Modificadores.
3. Ajustar los parámetros de mapeo de acuerdo a las necesidades deseadas, esto es, determinar el tipo de mapeo, ya sea planar o box, determinar las dimensiones del gizmo y determinar el tiling o repetición.

Por default, el modificador UVW Map utiliza el mapeo planar en canal de mapeo 1. Se pueden cambiar el tipo de mapeo, así como el canal de mapeo de acuerdo a las necesidades. Existen siete tipos de coordenadas de mapeo, noventa y nueve canales de mapeo, controles de tiling, y controles de tamaño y orientación del gizmo de mapeo.

VII.1.2.2.2 Aplicar el comando Collapse (Colapzar)

Después de aplicar el modificador UVW Mapping a un objeto o a una serie de objetos, y que la asignación de los parámetros a dicho modificador ha sido la adecuada, el siguiente paso es la aplicación del comando Collapse.

La finalidad de aplicar el comando Collapse (Colapsar), es quitar de la pila del objeto correspondiente el modificador UVW Mapping, interpretando que se colapsa el objeto, esto es, que al objeto se le incorpora o se incrusta en este caso el modificador UVW Mapping.

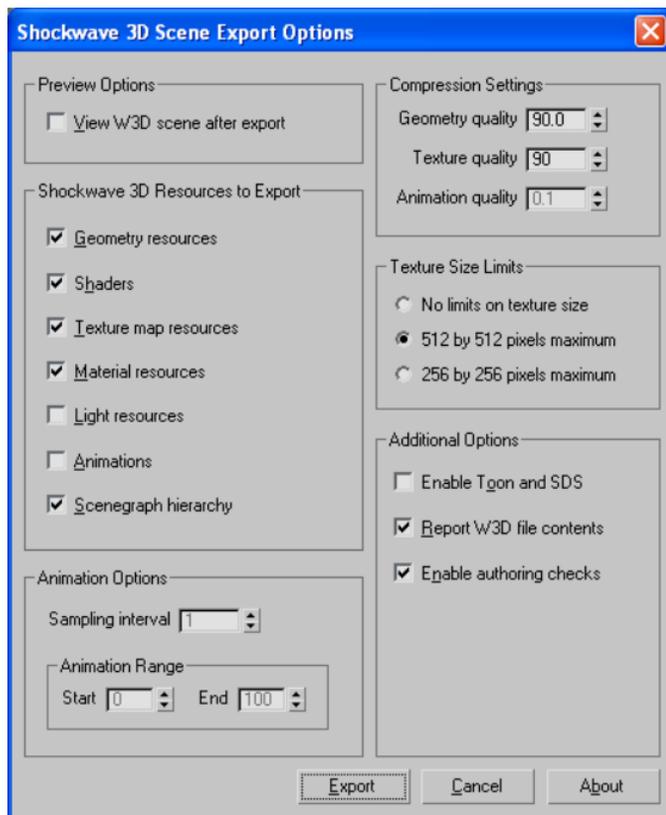


VII.1.3 EXPORTACIÓN

Para exportar nuestros modelos de 3D Studio Max a Director 8.5, utilizaremos la interfaz tridimensional de Director, llamada Shockwave 3D. El formato de archivos que lee Shockwave es *.w3d, el cual no podemos exportar desde 3D Studio hasta no haber descargado el plug-in para exportar modelos que se encuentra en el sitio Web de Macromedia.

Una vez hecho lo antes mencionado, y con nuestro modelo terminado (la malla completa, las texturas y su correcto mapeado), podemos exportar nuestro mundo eligiendo la opción Exporta del menú Archivo (File) y guardando el proyecto como archivo W3D.

A continuación explicaremos el funcionamiento de la pantalla que sirve para introducir los parámetros de exportación:



La ventana se divide en 6 secciones de las cuales describiremos 3 que son las de mayor importancia.

Shockwave 3D Resources to Export. Sirve para indicar que recursos queremos sean exportados; señalando cualquier checkbox habilitamos el recurso elegido. Nosotros únicamente exportamos la geometría, las texturas, shaders, materiales, y que se conserve la jerarquía entre objetos.

Compression Settings. Indicamos la calidad de la geometría y de las texturas del proyecto. Los valores óptimos para un menor tamaño de archivo y una visualización correcta fueron de 90 en ambos casos.

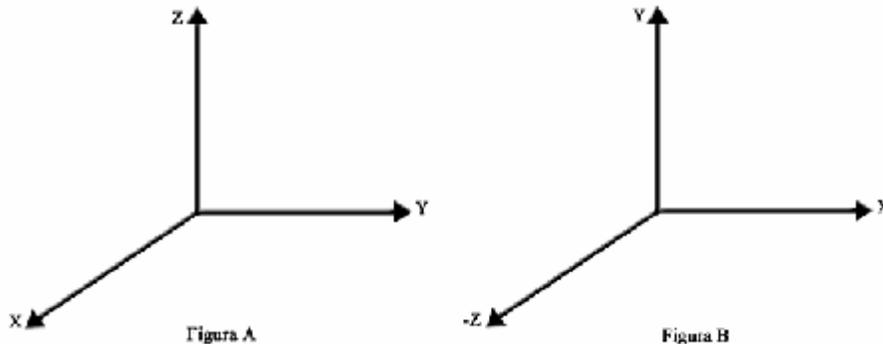
Texture Size Limits. Elegimos la segunda opción, para que ninguna textura exceda los 512x512 píxeles. En caso de rebasar el tamaño, la textura será ignorada.

Los motivos por los cuales elegimos esta opción son; en primer lugar, para una ejecución mas fluida en tiempo real, hay que recordar que mientras mayores sean las texturas, mayor será el procesamiento en tiempo real y más lenta será su ejecución pues consume mas memoria; y en segundo lugar porque existen algunas tarjetas de video que no soportan mas allá de los 512x512 píxeles en una imagen, y el objetivo es que el PV pueda verse en la mayoría de maquinas.

VII.1.4 CARACTERÍSTICAS DE LAS PLATAFORMAS.

A continuación mencionaremos algunas características importantes que se consideraron a la hora de realizar la exportación de 3D Studio max 5 a Shockwave 3D para ser incluido en Director 8.5.

Diferencias Canónicas en los ejes



3D Studio maneja el sistema coordenado de ejes ilustrado en la figura A, mientras que Shockwave maneja el sistema coordenado de ejes que se muestra en la figura B. Lo que hicimos fue rotar nuestro sistema coordenado A 90° en dirección positiva del eje X y posteriormente 90° en dirección positiva del eje Y, para que ambos coincidieran.

Nombramiento de los objetos del mundo 3D

Es muy importante el nombre que le demos a los objetos que conforman el ambiente en tres dimensiones, ya que posteriormente utilizaremos dichos nombres para hacer referencia a los modelos en tiempo real por medio de Lingo 3D; además es mucho más fácil referirse a los modelos por un nombre que los caracterice y nos lleve a recordar su función.

Luces y Cámaras

La incursión de cámaras e iluminación la realizaremos en tiempo real por medio de código en Lingo 3D; por este motivo el modelo de 3D Studio no cuenta con iluminación ni tampoco con cámaras, únicamente exportaremos el modelo con la geometría y las texturas con su respectivo mapeo.

A partir de este momento estamos listos para trabajar con nuestro archivo en formato **W3D**, realizar la programación 3D en tiempo real y poder integrarlo a la aplicación multimedia.

VII.2 DESARROLLO DE LA NAVEGACIÓN

Lo primero que tenemos que hacer es importar el mundo 3D, que se encuentra en formato Shockwave 3D (W3D). Una vez realizado lo anterior, tenemos ya como miembros del reparto de la película a los elementos Shockwave 3D, que serán nuestros mundos virtuales tridimensionales.

El siguiente paso es programar los eventos que sucederán en tiempo real dentro de nuestro mundo virtual tridimensional.

Comenzaremos explicando el método de detección de colisiones, posteriormente el manejo de la cámara, la animación de iluminación y audio, y finalmente la generación del ambiente.

El desarrollo de la navegación se refiere prácticamente a la codificación de la misma; en total para la navegación contamos con 10 scripts para su funcionamiento. Éstos scripts son: Preparar Película, Mantén hasta que estés listo, Descarga y limpieza del mundo, Llamada a inicio, Inicio, Creación de Primitivas Básicas, Manejo de Cámara, Manejo de Colisiones, Función mapeo, Función iluminación.

Por cuestión de espacio, tiempo y simplicidad, explicaremos únicamente la función principal del script de Manejo de Colisiones, llamada CheckForHits().

Explicaremos por medio de comentarios en el código, la forma en que funciona el script para el Manejo de Colisiones.

CÓDIGO PARA LA DETECCIÓN DE COLISIONES

```
-- Función que se encarga de generar los datos de la colisión y llevar a cabo una reacción hacia la colisión
on checkForHits(me)
```

```
-- activehit es la bandera para indicar donde ocurrió la colisión, en este caso aún no existe colisión
```

```
activehit = false
```

```
if banderamundo=0 then
```

```
-- establece la dirección del cilindro en el mundo virtual por medio de un vector director normalizado
```

```
dirvect = -(scene.model("mycyl").getworldtransform().zaxis)
```

```
dirvect.normalize()
```

```
-- vector director normalizado para el sensor superior, que sirve para ajustar las alturas
```

```
vectordirector=-(scene.model("mycyl").getworldtransform().yaxis)
```

```
vectordirector.normalize()
```

```
-- encuentra la posición de los 6 sensores (cajas)
```

```
leftEdgepos = scene.model("leftedge").getworldtransform().position
```

```

rightEdgepos = scene.model("rightedge").getworldtransform().position
leftDownpos = scene.model("leftdown").getworldtransform().position
rightDownpos = scene.model("rightdown").getworldtransform().position
updownpos=scene.model("updown").getworldtransform().position
updownpos2=scene.model("updown2").getworldtransform().position

-- encuentra la posición del cilindro
coordcil=scene.model("mycyl").getworldtransform().position

-- almacena en las variables equis, ye y zeta, las componentes del vector de posición para el cilindro
equis = coordcil.x
ye = coordcil.y
zeta = coordcil.z

-- determina si se ha colisionado con alguno de los 6 sensores y los almacena en las variables asociadas
lefthit = scene.modelsunderray(leftedgepos, dirvect, 1, #detailed)
righthit = scene.modelsunderray(rightedgepos, dirvect, 1, #detailed)
leftdownhit = scene.modelsunderray(leftdownpos, -dirvect, 1, #detailed)
rightdownhit = scene.modelsunderray(rightdownpos, -dirvect, 1, #detailed)
updownhit = scene.modelsunderray(updownpos, vectordirector,1, #detailed)
updownhit2= scene.modelsunderray(updownpos2, vectordirector,1, #detailed)

--cuando se oprime la tecla hacia adelante
if keypressed(126)then

-- encuentra la distancia del piso al sensor superior
updownDistance = updownhit[1][#distance]

-- la variable diferencia sirve para ajustar la altura del cilindro
-- la constante 23 es el número que arrojó el sensor superior en su estado inicial
diferencia = 23-updowndistance

-- si ha ocurrido una colisión en el sensor izquierdo entonces...
if lefthit[1][#model].name <> "safeplane" then

-- guarda la distancia a la cual colisiono
leftDistance = lefthit[1][#distance]

-- ajusta la altura en caso de ser necesario
scene.model("mycyl").translate(0,diferencia,0,#self)
else

-- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
leftDistance = pmaxsafe * 5
end if

-- si ha ocurrido una colisión en el sensor derecho entonces...
if righthit[1][#model].name <> "safeplane" then

```

```

-- guarda la distancia a la cual colisiono
leftDistance = pmaxsafe * 5

-- ajusta la altura en caso de ser necesario
rightDistance = righthit[1][#distance]
scene.model("mycyl").translate(0,diferencia,0,#self)
else

-- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
rightDistance = pmaxsafe * 5
end if

-- ajusta la altura del cilindro cuando no existe colisión
if (rightdistance >= 100 and leftdistance >= 100) then
  scene.model("mycyl").translate(0,diferencia,0,#self)
end if

-- determina si cualquiera de los dos sensores laterales colisionó con algo
if leftdistance <= pmaxsafe or rightdistance <= pmaxsafe then

-- siempre se considerará primero la menor distancia de colisión, en este caso
-- si la menor distancia es la del sensor derecho, entonces...
if leftDistance > rightdistance then

  -- almacena para la bandera de colisión, el dato del sensor derecho
  activeHit = righthit.duplicate()
  turn = #right
else

  -- si no, almacena para la bandera de colisión el dato del sensor izquierdo
  activeHit = lefthit.duplicate()
  turn = #left
end if
end if

-- si se produjo una colisión, entonces...
if activehit <> false then

-- calculo para la desaceleración del cilindro.
adecel = activehit[1][#distance]
decel = power((log(pmaxsafe * 2) - log(adecel)), 4)
driver.braking(decel)

-- calculo del ángulo para la colisión del cilindro
i = activehit[1][#isectnormal]
theta = ((i.dot(dirvect)) * 57.295779576)
theta = theta / float(2)
case(turn) of

```

```

#right:
  theta = -theta
#left:
  nothing
end case

-- si el ángulo de colisión es menor que 26° y mayor a -26°, entonces...
if theta > -26 and theta < 26 then

  -- ocurre un desvío para que la trayectoria sea paralela al objeto colisionado
  driver.deflect(theta)
end if
end if
else

--cuando se oprime la tecla hacia atrás
if keypressed(125)then

  -- encuentra la distancia del piso al sensor superior
  updownDistance2 = updownhit2[1][#distance]

  -- la variable diferencia sirve para ajustar la altura del cilindro
  -- la constante 23 es el número que arrojó el sensor superior en su estado inicial
  diferencia2 = 23-updowndistance2

  -- si ha ocurrido una colisión en el sensor izquierdo entonces...
  if leftdownhit[1][#model].name <> "safeplane2" then

    -- guarda la distancia a la cual colisiono
    leftdownDistance = leftdownthit[1][#distance]

    -- ajusta la altura en caso de ser necesario
    scene.model("mycyl").translate(0,diferencia2,0,#self)
  else

    -- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
    leftdownDistance = pmaxsafe * 5
  end if

  -- si ha ocurrido una colisión en el sensor derecho entonces...
  if rightdownhit[1][#model].name <> "safeplane2" then

    -- guarda la distancia a la cual colisiono
    rightdownDistance = rightdownhit[1][#distance]

    -- ajusta la altura en caso de ser necesario
    scene.model("mycyl").translate(0,diferencia2,0,#self)
  else

```

```

-- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
rightdownDistance = pmaxsafe * 5
end if

-- ajusta la altura del cilindro cuando no existe colisión
if (rightdowndistance >= 100 and leftdowndistance >= 100) then
  scene.model("mycyl").translate(0,diferencia2,0,#self)
end if

-- determina si cualquiera de los dos sensores laterales colisionó con algo
if leftdowndistance <= pmaxsafe or rightdowndistance <= pmaxsafe then

  -- siempre se considerará primero la menor distancia de colisión, en este caso
  -- si la menor distancia es la del sensor derecho, entonces...
  if leftdownDistance > rightdowndistance then

    -- almacena para la bandera de colisión, el dato del sensor derecho
    activeHit2 = rightdownhit.duplicate()
    turn = #right
  else

    -- si no, almacena para la bandera de colisión el dato del sensor izquierdo
    activeHit2 = leftdownhit.duplicate()
    turn = #left
  end if
end if

-- si se produjo una colisión, entonces...
if activehit2 <> false then

  -- calculo para la desaceleración del cilindro.
  adecel = activehit2[1][#distance]
  decel = power((log(pmaxsafe * 2) - log(adecel)), 4)
  driver.braking(decel)

  -- calculo del ángulo para la colisión del cilindro
  i = activehit2[1][#isectnormal]
  theta = ((i.dot(dirvect)) * 57.295779576)
  theta = theta / float(2)
  case(turn) of
    #right:
      theta = -theta
    #left:
      nothing
  end case

  -- si el ángulo de colisión es menor que 26° y mayor a -26°, entonces...
  if theta > -26 and theta < 26 then

```

```

-- ocurre un desvío para que la trayectoria sea paralela al objeto colisionado
driver.deflect(theta)
end if
end if
else

--cuando no se oprime ninguna tecla...
if signo=1 then

-- encuentra la distancia del piso al sensor superior
updownDistance = updownhit[1][#distance]

-- la variable diferencia sirve para ajustar la altura del cilindro
-- la constante 23 es el número que arrojó el sensor superior en su estado inicial
diferencia = 23-updowndistance

-- si ha ocurrido una colisión en el sensor izquierdo entonces...
if lefthit[1][#model].name <> "safeplane" then

-- guarda la distancia a la cual colisiono
leftDistance = lefthit[1][#distance]
-- ajusta la altura en caso de ser necesario
scene.model("mycyl").translate(0,diferencia,0,#self)
else

-- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
leftDistance = pmaxsafe * 5
end if

-- si ha ocurrido una colisión en el sensor derecho entonces...
if righthit[1][#model].name <> "safeplane" then

-- guarda la distancia a la cual colisiono
rightDistance = righthit[1][#distance]

-- ajusta la altura en caso de ser necesario
scene.model("mycyl").translate(0,diferencia,0,#self)
else

-- en caso contrario calcula una distancia muy grande para saber que no hubo colisión
rightDistance = pmaxsafe * 5
end if

-- ajusta la altura del cilindro cuando no existe colisión
if (rightdistance>=100 and leftdistance>=100) then
scene.model("mycyl").translate(0,diferencia,0,#self)
end if

-- determina si cualquiera de los dos sensores laterales colisionó con algo
if leftdistance <= pmaxsafe or rightdistance <= pmaxsafe then

```

```

-- siempre se considerará primero la menor distancia de colisión, en este caso
-- si la menor distancia es la del sensor derecho, entonces...
if leftDistance > rightdistance then

    -- almacena para la bandera de colisión, el dato del sensor derecho
    activeHit = righthit.duplicate()
    turn = #right
else

    -- si no, almacena para la bandera de colisión el dato del sensor izquierdo
    activeHit = lefthit.duplicate()
    turn = #left
end if
end if

-- si se produjo una colisión, entonces...
if activehit <> false then

    -- calculo para la desaceleración del cilindro.
    adecel = activehit[1][#distance]
    decel = power((log(pmaxsafe * 2) - log(adecel)), 4)
    driver.braking(decel)

    -- calculo del ángulo para la colisión del cilindro
    i = activehit[1][#isectnormal]
    theta = ((i.dot(dirvect)) * 57.295779576)
    theta = theta / float(2)
    case(turn) of
        #right:
            theta = -theta
        #left:
            nothing
    end case

    -- si el ángulo de colisión es menor que 26° y mayor a -26°, entonces...
    if theta > -26 and theta < 26 then

        -- ocurre un desvío para que la trayectoria sea paralela al objeto colisionado
        driver.deflect(theta)
    end if
end if
end if
end if
end if
end if
end

```

 **CAPÍTULO VIII** 
PRUEBAS Y MANTENIMIENTO

VIII.1 PRUEBAS

VIII. 1.1 PRUEBAS PARA EL MODELADO

Para la sección de modelado, al momento de ir desarrollando las secciones de nuestra zona arqueológica virtual, primeramente necesitábamos saber si la geometría estaba quedando bien es decir, si las pirámides, las escaleras, las calzadas y demás elementos que conforman las diferentes secciones del PV estaban ubicados en su posición correcta y con sus dimensiones exactas.

Cuando los modelos geométricos estaban completos, la siguiente tarea era aplicar las texturas de forma correcta, con el número de repeticiones exacto, en la orientación requerida y con la imagen que mas convenía, para simular un ambiente “realista” y que simulara de la mejor manera como era Teotihuacan en aquella época.

La forma en la que realizábamos las pruebas para el modelado, consistió únicamente en el “render” de los modelos, obteniendo una imagen de dicha geometría con su textura y con una iluminación ambiental generada por default por 3D Studio Max. De esta forma nosotros podíamos percibir si los resultados obtenidos eran o no los deseados. Si éstos no lo eran, nos veíamos en la necesidad de modificar la o las secciones del mundo virtual que hubiésemos determinado incorrectas.

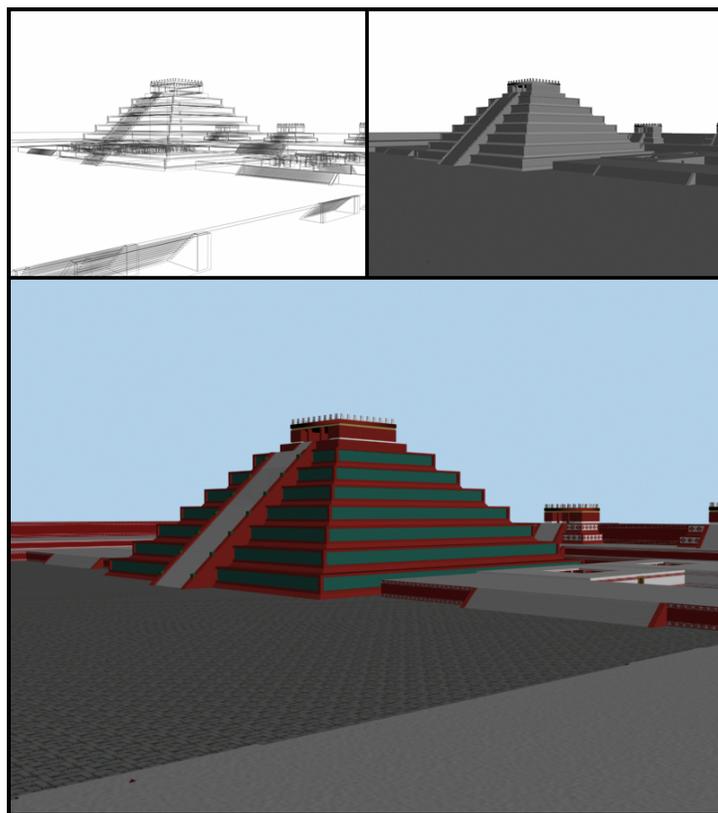


ILUSTRACIÓN VIII.1
Pruebas para el modelado

VIII.1.2 PRUEBAS PARA LA NAVEGACIÓN

En lo tocante a la navegación, las pruebas consistieron en la ejecución parcial de nuestro programa. Cuando trabajamos con un proyecto en Director, se dice que nos encontramos en modo de autoría, o modo de autor, donde tenemos la posibilidad de modificar el código que esté incorrecto, agregar o quitar miembros del reparto, o realizar cualquier otra opción que el proyecto requiera.

En ésta fase de autoría, debemos resaltar que los programas son interpretados por Director, es decir, que cuando detecta el primer error en la interpretación de la codificación detiene dicho proceso, por lo que no podremos visualizar todos los errores de la codificación, sino hasta que el programa sea ejecutado, además no se genera ningún archivo ejecutable, y cuenta con las demás características de un lenguaje interpretado, que son conocidas.

Una vez concluidas las pruebas, cuando hemos determinado que nuestro programa está completo y funcionando perfectamente, podemos realizar la compilación de nuestro proyecto para generar un archivo ejecutable, que se pueda leer en cualquier máquina, es decir, que nuestro paquete sea portable, y pueda funcionar perfectamente en todas las máquinas que cuenten con el hardware requerido y con el sistema operativo Windows en versiones 98 o posteriores. Además tenemos la ventaja de ocultar nuestro código y evitar que sea copiado ilegalmente o que la idea sea robada para ser utilizada en proyectos posteriores y lucrar con ello.

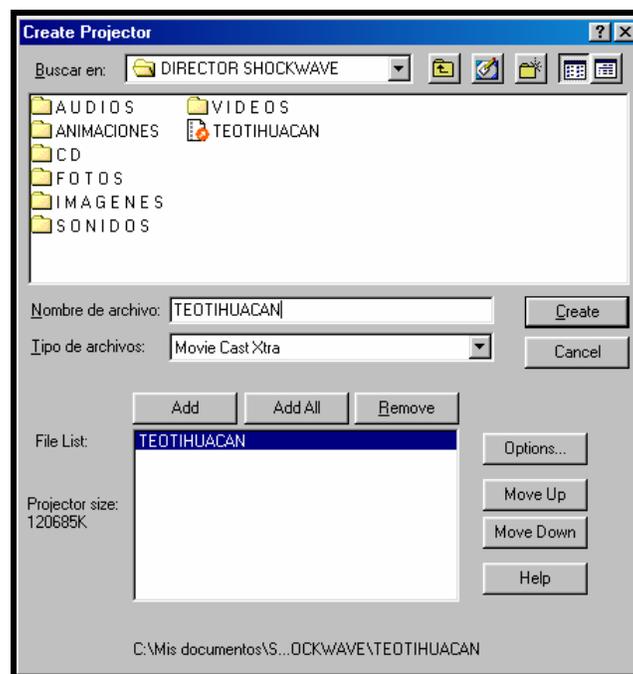


ILUSTRACIÓN VIII.2

Ventana para la creación del proyector (archivo ejecutable)

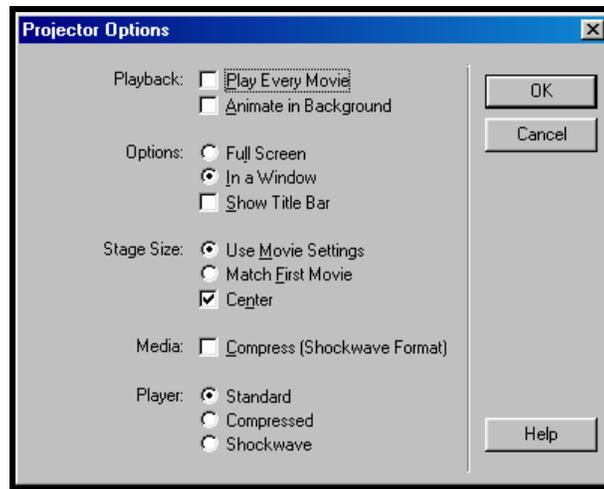


ILUSTRACIÓN VIII.3
Ventana de opciones del proyector

VIII.2 MANTENIMIENTO

Hasta este momento, no se le ha dado mantenimiento a nuestro programa, sin embargo está diseñado de forma modular, o integrado por módulos, de tal manera que podemos modificar cualquiera de sus partes o agregar nuevas, sin que esto afecte el funcionamiento del software, y de una forma sencilla y rápida.

Otra gran ventaja que tiene nuestro proyecto al haber sido desarrollado de manera modular, es que podemos extender nuestra plataforma hacia otros proyectos orientados a los recorridos virtuales.

Por ejemplo, podíamos implementar de manera sencilla, desarrollando únicamente el modelado de cualquier otra zona arqueológica, un PV en el derruido Templo Mayor de Tenochtitlan, y hacer que recobre la vida y transportarnos al pasado haciendo uso de nuestra aplicación, o cualquier otro recorrido virtual aunque no esté relacionado con la arqueología.

Imaginemos el gran potencial de nuestra aplicación, ya que de igual forma que hemos realizado el recorrido a través de la zona arqueológica de Teotihuacan, podríamos realizar un viaje intergaláctico, o un recorrido intramolecular, o cualquier proyecto imaginable que se pueda modelar de manera sencilla, sin que se necesite de una gran capacidad de cómputo y con la gran ventaja que será accesible para la mayor parte de los usuarios que cuente con un equipo de computo.



CAPÍTULO IX

CONCLUSIONES



“Nacionalizar la ciencia, mexicanizar el saber”**Justo Sierra**

En la medida en que una nación genere su propia tecnología, podrá realizar una verdadera revolución, desarrollándose en todos los aspectos y alcanzando un mejor nivel de vida para la mayor parte de su sociedad.

El objetivo de la ingeniería es plantear y desarrollar soluciones prácticas en beneficio de la sociedad. En la actualidad uno de los principales problemas que afronta la sociedad mexicana es la deficiente educación de los jóvenes mexicanos. Una de las principales aportaciones de este trabajo de tesis es llevar la Realidad Virtual a la educación, de esta forma los estudiantes podrán alcanzar un mejor dominio, retención y generalización de los nuevos conocimientos, ya que se involucrarán activamente en la construcción del conocimiento. La Realidad Virtual hará que los estudiantes sean generadores del conocimiento y no únicamente receptores, como hasta ahora lo han sido.

En este trabajo de tesis se utilizan plataformas de programación y de diseño para generar un programa de Realidad Virtual novedoso, ya que liga a la multimedia y a la Realidad Virtual con la historia de México; y versátil, ya que es aplicable a muchas áreas del conocimiento así como también lo es en el entretenimiento o como difusión turística, por mencionar sólo algunas áreas.

Con las herramientas de cómputo necesarias y los conocimientos adquiridos a lo largo de nuestra carrera pudimos desarrollar de manera práctica un paseo virtual en la zona arqueológica de Teotihuacan, logrando el objetivo propuesto al inicio de nuestro trabajo de tesis. Primero reconstruimos los edificios de la zona, posteriormente editamos los materiales y finalmente realizamos la programación para poder navegar en el mundo virtual; de manera alternativa se agregaron elementos multimedia para que el programa fuera amigable y didáctico, debido a que el paseo virtual está enfocado a la educación.

Nuestro programa puede aplicarse a cualquier tipo de proyecto, que pueda modelarse de manera sencilla, de esta forma podríamos utilizar nuestra plataforma para generar un Paseo Virtual en cualquier área del conocimiento, por ejemplo un viaje al sistema planetario solar, dentro de la pirámide de Palenque, o en la zona del Templo Mayor, por citar sólo algunos ejemplos.

De inicio en los objetivos se había planteado el desarrollo del paseo virtual haciendo uso de cualquier game engine, sin embargo los resultados obtenidos no fueron los idóneos. Algunos problemas que se presentaron al utilizar diversos game engines fueron los siguientes: la interfaz de diseño era complicada (sobre todo en el Unreal Tournament), la capacidad para importar modelos generados en 3D Studio Max es limitada, ya que se modifica la malla y no permite un mapeado correcto de las texturas (lo que hubiese implicado el doble de trabajo ya que el mapeado se había generado dentro de 3D Studio Max); aunque el game engine de Unreal

Tournament generaba las sombras de los objetos en escena, no tenía capacidad de exclusión para los elementos que conforman el mundo (esto para excluir la caja que forma el ambiente y que no se viera afectada por la iluminación general, dando el efecto de inmersión ambiental), y quizá el problema más grande, la poca documentación sobre cuestiones específicas para los editores correspondientes a los video juegos.

Al cambiar de un game engine a Lingo 3D ganamos en interactividad, amigabilidad y personalización, otra ventaja es que pudimos generar un ejecutable independiente de cualquier aplicación que se puede correr en cualquier máquina, a diferencia de la aplicación generada sobre el game engine, la cual necesita forzosamente del video juego para poder funcionar; la documentación existente para programar en Lingo 3D fue mucho mayor ya que contamos con libros, tutoriales en la web y la ayuda de Director.

Cabe mencionar que tuvimos algunas dificultades para obtener una buena referencia de donde extraer la información y desarrollar un modelo adecuado aproximado al real. De inicio pensamos que existía información digitalizada de la zona arqueológica, para ello acudimos a instancias como el Instituto de Investigaciones Antropológicas de la UNAM, el Centro de Estudios Teotihuacanos, el Museo Nacional de Antropología y las oficinas centrales del INAH, pero desgraciadamente aún no se cuenta con un acervo de información a la altura de los avances tecnológicos actuales.

Entre las referencias obtenidas, podemos citar los planos realizados entre los años 60's y 70's por Rene Millon, éstos planos incluyen referencia topográfica y arquitectónica, pero debido al tamaño de estos es prácticamente imposible obtener información acerca de las dimensiones de los elementos que conforman la zona arqueológica.

También obtuvimos planos de levantamientos arqueológicos, donde se muestra únicamente la forma de la excavación, ofrendas encontradas, y demás elementos que tengan que registrar los arqueólogos. Ninguna de las referencias antes citadas nos proporcionó información útil para el desarrollo de nuestro modelo, motivo por el cual decidimos tomar como fuente de información a la maqueta del Museo de Sitio de Teotihuacan.

Al tomar las fotografías de la maqueta se cometen errores de aproximación a las medidas reales, sin embargo, es suficiente para cubrir nuestro objetivo, ya que no requerimos las medidas y posiciones exactas de cada uno de los elementos; lo que requerimos es mostrarlos de tal forma que permita la navegación y que sus semejanza con la realidad sea la mejor posible.

Debido a las grandes dimensiones de la maqueta y al acceso restringido, hay zonas que se dificultó obtener fotografías debido a su lejanía, al no contar con puntos de perspectiva, por lo que no se contó con fotografías frontales, laterales, y superiores de la edificaciones para obtener las proyecciones correspondientes y determinar sus dimensiones de ancho, largo y alto, pero esto no fue posible debido a las complicaciones mencionadas; por lo que solo realizamos las proyecciones de los edificios grandes, que son la Pirámide del Sol, de la Luna y de Quetzalcoatl.

Así que no podríamos haber encontrado una mejor referencia que esta maqueta, ya que podemos considerarla como un modelo físico, y nuestra tarea consistió en llevar ésta representación, a un modelo virtual.

Pasando a otro punto, un aspecto de gran relevancia es que cualquier computadora en la actualidad debe correr correctamente el programa. La velocidad de ejecución dependerá de los recursos de la computadora es decir, la velocidad del microprocesador, la memoria RAM y el tipo de tarjeta de video principalmente. Con esto la mayoría de las escuelas que cuenten con equipo de cómputo reciente podrán hacer uso del programa, así como la gente que tenga en sus hogares u oficinas su computadora.

La principal aportación de nuestro trabajo de tesis es el haber utilizado un algoritmo existente, modificarlo de acuerdo a nuestras necesidades y de esta forma realizar de manera sencilla una plataforma de paseos virtuales que permita navegar al usuario libremente (como en un video juego) mientras conoce y aprende sobre cualquier tópico. El Paseo Virtual puede enfocarse a la educación, a cualquier otra área del conocimiento o inclusive al entretenimiento.

La expansión e inmersión de la Realidad Virtual en nuestras vidas será cada vez mayor, por eso la importancia de adentrarnos en esta tecnología, para que en el futuro seamos partícipes de ella. Podremos simular sismos y diseñar mejores estructuras que nos permitan evitar catástrofes, entenderemos como se generan los huracanes, como están formadas las nubes, manipularemos estructuras moleculares para entenderlas mejor, los médicos aprenderán a operar en modelos virtuales, y veremos como está formado el universo.

Tendremos que incursionar en industrias que no han sido explotadas en nuestro país, como es el caso de los videojuegos y de las películas animadas por computadora, ambas con un gran futuro por delante.

Esperamos proyectar nuestro trabajo de tesis para que en un futuro se pueda desarrollar de manera comercial en conjunción con cualquier instituto de investigación, con el INAH o con cualquier otra institución interesada en el desarrollo de éste tipo de Paseos Virtuales, y se pueda concretar un proyecto profesional de alta trascendencia.

Estamos convencidos que este trabajo servirá como base para que cualquier persona que esté interesada en el desarrollo de Paseos Virtuales, pueda mejorar el mismo o bien implementar un nuevo proyecto utilizando las herramientas que los avances tecnológicos le otorguen, pudiendo incursionar en nuevas líneas de investigación.



CAPÍTULO X
GLOSARIO



3D Lingo. Es una rama de Lingo dedicada a controlar entornos en 3D en Director. Por tanto, 3D Lingo amplía las posibilidades de Lingo.

Bitmap. Un bitmap es una imagen producida por una matriz fija de píxeles coloreados, como un mosaico. Se pueden usar los bitmaps como las texturas para los materiales.

Booleano. Combina dos objetos, realizando una operación Booleana sobre ellos. Los dos objetos originales son los operandos y el objeto Booleano es el resultado de la operación.

CAD. Computer-Aided Design (Diseño Asistido por Computadora).

Cámara. Es uno de los nodos básicos del entorno en 3D. La cámara se considera como la vista del usuario en una escena en 3D. Sin una cámara, el usuario no puede ver el mundo en 3D.

Canal Alfa. Hace referencia a la información almacenada dentro de un mapa de bits cuyo objetivo es controlar la cantidad de transparencia por píxel. A los canales alfa se hace referencia utilizando su profundidad de bits, que describe cuantos niveles de transparencia son capaces de admitir. El canal alfa utilizado en nuestras imágenes fue de 8 bits de profundidad y proporciona 256 niveles de transparencia.

Coordenadas de Mapeo. Las coordenadas de mapeo especifican la colocación, orientación, y escala de un mapa en la geometría. Si se aplica un material mapeado a un objeto que no tiene ninguna coordenada de mapeo, el Render asigna las coordenadas de mapeo predefinidas. Las coordenadas de mapeo incorporadas se diseñan para cada tipo objeto. Las coordenadas de mapeo caja coloca un mapa doble en cada uno de sus seis lados. Para el cilindro, la imagen se envuelve una vez alrededor de sus lados. Una esfera tiene la imagen envuelta una vez alrededor de la esfera.

Coordenadas del Mundo. Cuando calculamos situaciones en un mundo en 3D, tendremos que medir desde un punto de partida. Si el punto de partida es el origen del mundo en 3D, estamos realizando las medidas en coordenadas del mundo.

Coordenadas Locales. Cuando medimos ubicaciones en un mundo en 3D, tendremos que medir desde un punto de partida. Si este punto de partida no es el origen del mundo en 3D, tal como un nodo, las medidas que estamos tomando se encuentran en coordenadas locales.

D3D (Director 3D). Los desarrolladores hacen referencia normalmente al proceso de programación o desarrollo de contenido 3D en Director como D3D.

Deformación de Mallas. Es un proceso de la manipulación de la geometría de un modelo utilizando cualquier transformador geométrico para su manipulación.

Detección de Colisión. Un proceso que exige mucha memoria y que comprueba si los modelos están o no intercectándose.

Editable Mesh. Una malla editable es un tipo de objeto deformable. Una malla editable es un trimesh: es decir, usa los polígonos triangulares. Las mallas editables son útiles para crear objetos simples, objetos bajo poligonales o mallas de control para MeshSmooth y modelado de HSDS. Se pueden convertir NURBS y superficies patch a una malla del editable. Las mallas editables requieren poca memoria, y es un método natural de modelar con objetos poligonales.

Escalamiento. Uno de los tres tipos principales de transformaciones utilizado para aumentar o disminuir el tamaño de la geometría.

Formato de Archivo BMP. Formato de imagen de Microsoft Windows, tanto para 8 bits como para 24 bits.

Formato de Archivo DIR. Formato de un archivo correspondiente a una aplicación o programa desarrollado en la plataforma de Director de Macromedia.

Formato de Archivo EXE. Formato de un archivo correspondiente a una aplicación o programa ejecutable.

Formato de Archivo GIF. Formato gráfico desarrollado por Comuserve y destinado en un principio a imágenes de 8 bits (256 colores).

Formato de Archivo JPG. Formato de imagen comprimida, por lo que no ocupa mucho espacio en el disco y por lo tanto se descarga con mayor velocidad.

Formato de Archivo MAX. Es un formato de archivo nativo de 3D Studio Max, el cuál sirve para almacenar modelos tridimensionales, luces, cámaras, ayudantes, animaciones, etc.

Formato de Archivo PNG. (Portable Network Graphics) es un formato de archivo de imagen desarrollado para el uso con la Internet y Web. Con la característica principal que cuenta con 8 bits del canal alfa para poder asignarle una canal de transparencia a la respectiva imagen.

Formato de Archivo W3D. Es un formato de archivo nativo de Director utilizado para importar modelos y movimientos creados en programas de modelaje externo.

Gizmo. Muchos modificadores desplegados en los puertos de vista como una caja que inicialmente rodea al objeto seleccionado llamada gizmo. Un gizmo actúa como un tipo de recipiente que transfiere la modificación hacia el objeto al cual está ligado.

Se puede mover, escalar, y rotar un gizmo como se haría sobre cualquier objeto, alterando el efecto del modificador en el objeto.

HMD. Head Mounted Display. Interfase de los sistemas de RV que genera la inmersión del usuario. Consiste en un casco con pantallas integradas.

Luz Ambiental. Se utiliza para aproximarse a un nivel general de la luz en un entorno. Concretamente, el uso de la luz ambiental es un intento de aproximarse a la cantidad de luz reflejada dentro de una escena.

Materiales. Es un conjunto de datos que se asignan a la superficie o caras de un objeto para que este aparenten algo cuando son renderizados. Los materiales afectan el color de los objetos, su brillo y su opacidad. Un material estándar contiene una componente ambiental, difusa, y especular. Se pueden asignar mapas a las distintas componentes de un material estándar.

Materiales Compuestos. Los materiales compuestos permiten crear un material compuesto o integrado por dos o más submateriales. El material compuesto Multi/subobjeto permite asignar los diferentes materiales a los subobjetos diferentes, al nivel del subobjeto de su geometría.

Malla (Mesh). Es un tipo especial de recurso aplicado a los modelos que permite construir y deformar la geometría en sus distintos niveles; vértice, arista, cara, polígono y objeto.

Mapas. Las imágenes que se asignan a los materiales son llamadas mapas. Los programas de software de imágenes proveen varios tipos de mapas diferentes. Ellos incluyen bitmaps estándares (como archivos .bmp, .jpg, o .tga), mapas procesales, como el Checker o Marble, y sistemas de procesamiento de imágenes como composiciones y sistemas de enmascarado.

Mip – Mapping. Proceso que crea varias versiones sucesivamente más pequeñas de una textura. Esto permite al render en 3D elegir una versión de la textura que tenga el tamaño adecuado, dado que el tamaño de la textura varía dependiendo de su ubicación en el entorno en 3D.

Modelo. Es uno de los nodos principales de un entorno en 3D. Los modelos se utilizan para representar los objetos de una escena.

Modificador Pila. Es la llave para manejar todos los aspectos de la modificación del objeto.

Modificadores. Modifica la estructura geométrica de un objeto, deformándolo de alguna manera. Los modificadores hacen los cambios en la geometría que permanece en efecto hasta que se ajuste o anule el modificador.

NearFiltering. Es un proceso que resta nitidez ligeramente a las texturas cuando se encuentran cerca de la cámara, esto sirve para evitar el pixelado de la textura.

Objeto Tiempo de Espera. Objeto especial de código que actúa como temporizador. Se utiliza para disparar un evento determinado una vez que ha transcurrido una cierta cantidad de tiempo.

Pixel. Es el elemento más pequeño de una imagen, es un solo punto en una imagen gráfica. Los monitores gráficos despliegan imágenes dividiendo la pantalla en miles (o millones) de pixeles, arreglados en filas y columnas.

Plug-In. Es una característica o funcionalidad proporcionada para un programa independiente o componente, es decir, extiende la funcionalidad de un programa. Estos pueden ser proporcionados por distintos medios, por ejemplo por diseñadores de software independientes.

Primitivos. Son los cinco tipos de recursos básicos incorporados tanto a los modeladores tridimensionales como a Lingo, concretamente, estos cinco primitivos son: cajas, cilindros, esferas, planos y sistemas de partículas.

Profundidad de Bits. Es un término utilizado para hacer referencia al número de colores disponibles para un monitor, un mapa de bits o un programa.

PV. Abreviatura de Paseo Virtual.

Renderización (Render). Es un proceso de representación de los entornos en 3D, donde se observan los modelos geométricos texturizados, con iluminación, etcétera, vistos desde el punto de vista de la cámara.

Rotación. Es uno de los tres tipos de transformación. Es el proceso de girar los nodos de un elemento geométrico respecto a otro nodo dentro del entorno 3D.

RV. Abreviatura de Realidad Virtual.

Shader. Es un elemento de los entornos en 3D que se utiliza para definir las cualidades de la superficie, como su color o su brillo.

SubObject. El subobjeto es un subconjunto de la geometría del un objeto. Muchos objetos tienen varios tipos de subobjetos que se pueden trabajar independientemente. Por ejemplo, un subobjeto de un objeto malla editable son vértices, bordes, caras, polígonos, y elementos. Para acceder a los subobjetos, hay que acceder por medio del panel Modificar.

En el nivel subobjeto se puede seleccionar los subobjetos, transformar las selecciones, aplicar modificadores, y así sucesivamente.

SW3D (Shockwave 3D). Término que se usa normalmente para hacer referencia a todo el proceso y el producto envuelto en la creación de proyectos en 3D en tiempo real en Director.

Textura. Elemento de los entornos en 3D que se utiliza en combinación con un shader para definir las cualidades materiales de un modelo.

Transformación. Término que se utiliza para hacer referencia al proceso que orienta, coloca y escala los nodos en el entorno en 3D.

Cuando se crea cualquier objeto, se graba la información de su posición, rotación, y escala en una tabla interna llamada matriz de transformación. Subsecuentemente los ajustes de posición, rotación, y escalamiento son llamadas transformaciones.

Cada transformación geométrica tiene como referencia un sistema de coordenadas ubicado en cualquier elemento o región geométrica, así como en el sistema de coordenadas predefinido. Un objeto puede tener solamente una transformación geométrica en un mismo instante de tiempo.

Traslación. Es uno de los tres tipos de transformación. Es el proceso de posicionar los nodos dentro del entorno 3D.

UVW Mapping (Mapeo UVW). Es el proceso de aplicar un shader a un modelo. Los shaders se aplican a los modelos en el espacio en 3D relativo a estos. Para evitar la confusión entre las coordenadas de la aplicación de mapas y las coordenadas normales, se utilizan las letras UVW, aunque técnicamente se trata de las coordenadas XYZ.

La mayoría de los mapas de materiales son planos 2D asignados a una superficie 3D. Por consiguiente, el sistema de coordenadas usado para describir la colocación y transformación de los mapas es diferente del los ejes X, Y, y Z coordenadas usadas en el espacio 3D. La coordenada de profundidad W para un plano 2D, se utiliza para poder voltear la orientación de un mapa, relativo a su geometría. La coordenada de W también tiene un significado para los materiales de 3 dimensiones.



CAPÍTULO XI

BIBLIOGRAFÍA



- BERNAL y García Pimentel, Ignacio., “Guía Oficial: Teotihuacan”, Salvat, Instituto Nacional de Antropología e Historia, México, 1985
- BURGOS, Daniel. De-León, Luz., “Director 8.x / Lingo. Práctico. Guía de Aprendizaje.”, Editorial Mc Graw Hill / Interamericana de España, España, 2000
- CATANESE, Paul., “Director 8.5. Fundamentos de Programación 3D”, Editorial Anaya Multimedia, España, 2002
- DE LA FUENTE, Beatriz., “Pintura Mural Prehispánica en México. Teotihuacan.”, Instituto de Investigaciones Estéticas, UNAM, Volumen I, Tomo II, México, 2001
- DE LA FUENTE, Beatriz., “Pintura Mural Prehispánica”, Instituto de Investigaciones Estéticas, UNAM, CONACULTA, México, 2001
- FOLEY, James D., Van Dam, Andries., Feiner, Steven K., Hughes, John F., Phillips, Richard L., “Introducción a la Graficación por Computador”, Editorial Addison-Wesley Iberoamericana, EUA, 1996
- GAMIO, Manuel., “Guía para visitar la ciudad arqueológica de Teotihuacan”, Talleres Gráficos de la Nación, México, 1929
- HEARN, Donald., Baker, Pauline M., “Graficas por Computadora”, Editorial Prentice Hall Iberoamericana, México, 1995
- LAWRENCE Pfleeger, Shari. “Ingeniería de Software. Teoría y Práctica.”, Editorial Printice Hall y Pearson Educacion, Brasil, 2002
- MARQUINA, Ignacio., “Arquitectura Prehispanica”, Instituto Nacional de Antropología e Historia, México, 1964
- MATOS Moctezuma, Eduardo., “Museo de Sitio de Teotihuacan”, Instituto Nacional de Antropología e Historia, México, 1993
- MILLON, Rene., “The Teotihuacan Map”, Austin, University of Texas, EUA, 1973
- PARRA Márquez, Juan Carlos., García Alvarado, Rodrigo., Santelices Malfanti, Iván., “Introducción Practica a la Realidad Virtual”, Ediciones UBB, Chile, 2001
- PRESSMAN, Roger S., “Ingeniería de Software. Un Enfoque Práctico”, Editorial Mc Graw Hill, 4a.edición, México, 1997

- SOMMERVILLE, Ian., “Ingeniería de Software”, Editorial Addison-Wesley Iberoamericana, México, 1988
- UPSTILL, Steve., “The RenderMan Companion. A Programmer’s Guide to realistic Computer Graphics.”, Editorial Addison-Wesley Publishing Company, EUA, 1990
- VARIOS, “Enciclopedia de México”, Tomo XIII, Editorial Sabeca International Investment C., México, 2000
- VARIOS, “Enciclopedia Universal Ilustrada. Europeo-Americana.”, Editores ESPASA-CALPE, S.A., Madrid, España, 1928, Tomo LX
- WATT, Alan., “3D Computer Graphics”, Editorial Addison-Wesley, Tercera Edición, Gran Bretaña, 1993

TESIS

- AROCHE Juárez, Carlos Federico., León Estrada, Moisés Alfredo., Vidal Garibay, Francisco Arturo., Valiente Gómez, Santiago Igor. (Asesor), “La Ingeniería de Software orientada hacia los Video Juegos”, Facultad de Ingeniería, UNAM, México, 2003
- CANCHE Rodríguez, Claudia., Santos Jaimes, Germán. (Asesor), “Creación de un Sistema de Realidad Virtual”, Facultad de Ingeniería, UNAM, México, 1999
- CARRILLO Silva, Urbano Cesar., Domínguez Osorno, Guillermo., Zaldivar Zamorategui, Orlando. (Asesor), “Sistema Documental en Realidad Virtual sobre la Cultura Maya”, Facultad de Ingeniería, UNAM, México, 2000
- GUZMÁN Velásquez, Carlos Alberto., Alva de la Selva, Alma Rosa. (Asesor), “Aspectos Generales sobre la Realidad Virtual y sus Principales Aplicaciones”, Escuela Nacional de Estudios Profesionales Acatlán, UNAM, México, 1997
- TENORIO Sánchez, Angélica., Torre Aguirre, Elba., Romero Ruiz, Rubén. (Asesor), “Desarrollo y Construcción de un Sistema de Realidad Virtual”, Escuela Nacional de Estudios Profesionales Acatlán, UNAM, México, 1997
- VALIENTE Gómez, Santiago Igor., Carreón Granados, Juan José. (Asesor), “Realidad Virtual: Panorámica y Aplicaciones”, Facultad de Ingeniería, UNAM, México, 1997

REVISTAS

- Guías Arqueológicas. Sitios Arqueológicos de Estado de México. Teotihuacan., Eduardo Moctezuma Matos, Editorial México Desconocido, 2002
- Arqueología Mexicana. Teotihuacan Ciudad de Misterios., Editorial Raíces, Vol.XI, No.64
- Pasajes de la Historia IV. El Milenio Teotihuacano., Eduardo Moctezuma Matos, Editorial México Desconocido CONACULTA, 2000
- Muy Interesante, Editorial Televisa, Año XX, No. 3, México, Pág. 36-39
- 3D World, Editorial Prensa Técnica, Año 6, No. 57, España
- 3D Todo 3D, Editorial MC Ediciones, No. 26, España

SITIOS DE INTERNET

Realidad Virtual

- <http://www.radiouniversidad.org/secciones/reportajes/tecnologia/>
- <http://www.ucm.es/info/multidoc/multidoc/revista/num8/hilera-oton.html>
- <http://www.utp.ac.pa/seccion/topicos/realidad/>
- <http://www.sav.us.es/pixelbit/articulos/n15/n15art/art151.htm>
- <http://zeus.dci.ubiobio.cl/~sigradi/libros/>

Ingeniería de Software

- <http://www.inf.udec.cl/~ingsoft/apunte2001.htm>
- <http://zeswish.homeip.net/apuntes3/ingenieria/index.php>
- <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
- <http://148.225.83.24/ic2/ic2apuntes/index.html>
- <http://148.202.148.5/cursos/cc321/fundamentos/temario.htm>

3D Studio MAX

- <http://www.3dcafe.com>
- <http://www.3dimens.org>
- <http://www.3dlinks.com>
- <http://www.unav.es/cti/manuales/3DStudioMax/indice.html>
- <http://www.infor.uva.es/~descuder/docencia/IG/3DStudio/IndicePrincipal.htm>

OTRAS REFERENCIAS

- Teotihuacan. Guía Arqueológica para Niños y Jóvenes., Escalante Gonzalbo, Pablo., Becerra, Iris., Ediciones Tecolote, CONACULTA, México, 2002
- Teotihuacan. Estado de México. Vela, Enrique., Instituto Nacional de Antropología e Historia. CONACULTA, México. 1992
- Gaceta. Órgano Informativo de la Universidad Nacional Autónoma de México., Ciudad Universitaria, 11 de Marzo de 2004, Número 3703