

UNIVERSIDAD DE ORIENTE
NUCLEO DE ANZOATEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACION Y SISTEMAS



**“ACTUALIZACIÓN Y EXPANSIÓN DEL AMBIENTE VIRTUAL
PROTOTIPO DEL DEPARTAMENTO DE COMPUTACIÓN Y
SISTEMAS DE LA UNIVERSIDAD DE ORIENTE NÚCLEO DE
ANZOÁTEGUI.”**

REALIZADO POR:

Millán Chirinos, José Luis

**Trabajo de Grado presentado como requisito parcial para optar al Título de:
Ingeniero en Computación**

BARCELONA, Abril de 2009.
UNIVERSIDAD DE ORIENTE
NUCLEO DE ANZOATEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACION Y SISTEMAS



**“ACTUALIZACIÓN Y EXPANSIÓN DEL AMBIENTE VIRTUAL
PROTOTIPO DEL DEPARTAMENTO DE COMPUTACIÓN Y
SISTEMAS DE LA UNIVERSIDAD DE ORIENTE NÚCLEO DE
ANZOÁTEGUI.”**

JURADO CALIFICADOR:

Ing. Pedro Dorta
Asesor Académico

Ing. Claudio Cortínez
Jurado Principal

Ing. Víctor Mújica
Jurado Principal

Trabajo de Grado presentado como requisito parcial para optar al Título de:
Ingeniero en Computación

BARCELONA, Abril de 2009.
UNIVERSIDAD DE ORIENTE
NUCLEO DE ANZOATEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACION Y SISTEMAS



**“ACTUALIZACIÓN Y EXPANSIÓN DEL AMBIENTE VIRTUAL
PROTOTIPO DEL DEPARTAMENTO DE COMPUTACIÓN Y
SISTEMAS DE LA UNIVERSIDAD DE ORIENTE NÚCLEO DE
ANZOÁTEGUI.”**

Ing. Pedro Dorta
Asesor Académico

**Trabajo de Grado presentado como requisito parcial para optar al Título de:
Ingeniero en Computación**

BARCELONA, Abril de 2009.

RESOLUCIÓN

Artículo N° 44.

Del Reglamento de Trabajo de Grado.

“Los Trabajos de Grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario”.

RESUMEN

Este trabajo se plantea de la experiencia obtenida del desarrollo del proyecto AVUDONA_1, en el cual, se construyo un Ambiente Virtual del Departamento de computación y sistema, usando el lenguaje VRML para crear un Mundo virtual interactivo tridimensional. De igual forma siguiendo este planteamiento surge la inquietud de desarrollar este proyecto que permitió extender y actualizar el ambiente virtual de la Universidad de Oriente Núcleo de Anzoátegui. Agregando nuevas escenas y realizando cambios tanto en las tecnologías aplicadas como las escenas que sufrieron cambios en su homónimo real. En este Ambiente virtual, AVUDONA_2, el visitante se limitara a moverse libremente por las escenas del mundo virtual, pudiendo solo actuar en la activación de determinados sucesos como la apertura de una puerta, etc. De esta forma permitir conocer las instalaciones de la Universidad a distancia utilizando no solo medios multimedia como CD-ROM/DVD-RW, sino también, a través de la tecnología WWW o Internet.

Para el desarrollo de este trabaja se aplico ULM para el analizar, recolección, diseño de toda los datos pertinente, combinada a la metodología utilizada para el desarrollo de ambiente virtuales, la cual, permitido distribución del proyecto dividiéndolo en varias etapas que facilitaron el desarrollo y culminación de una maneras más eficiente. Se utilizo Auto CAD para data de planos en 2D, para la diseño de las imágenes tridimensionales 3D Studio MAX y para la vinculación entre las escenas dinámicas (animaciones) y estáticas se utilizo VRML y JavaScript.

Palabra Clave: AVUDONA, AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NÚCLEO DE ANZOÁTEGUI.

DEDICATORIA

En primer lugar a Dios todo poderoso, por haber dado la fuerza y tolerancia para cumplir esta meta tan anhelada.

A mí querida e incomparable madre Nieves Chirinos, este logro también es tuyo mereces esto y muchas cosas más, para ti con todo mi amor y cariño.

A mí hermana Yolimar Millán que siempre estuvo pendiente de mí, este trabajo también es para ti y sirva de guía para que veas que si se puede. Te quiero Mucho.

A mis abuelas paterna y materna que ya no están físicamente conmigo a ellas mi más profundo reconocimiento por su lucha de vida esto es para ustedes donde quieran que estén.

Y cada unas de las personas que directa e indirectamente hicieron posible cumplir esta meta, para todos ustedes.

AGRADECIMIENTO

Para comenzar deseo en nombre de Jesús dar las gracias a dios quien fue mi guía para alcanzar el fin de esta etapa de mi vida, su fuerza divina y bendición es lo que me ayudo a lograr esta meta. GLORIA A DIOS...AMEN.

Existe una persona muy importante para mí que gracias a su sacrificio, su esfuerzo, su luchar en la vida me enseñó a nunca rendirme ante ningún obstáculo. Sr. Nieves Chirinos, mi madre, gracia por todo el sacrificio y tu apoyo incondicional, en esto que se convertido en una verdadera carrera en mi vida a ti te debo todo lo que soy como persona y el profesional que voy hacer, tú has sido el pilar de vida a ti te agradezco de corazón todo tu amor. GRACIA MADRE...no habrá nada sobre esta tierra para pagarte lo que soy. Te amo.

A mi hermana, Yolimar Millán por su sacrificio y su apoyo incondicional también forma parte de esa base fundamental para lograr esta meta, gracia hermana tu comprensión y apoyo fue importante para mí. Espero pronto leer tus agradecimiento en tu trabajo de grado.

Agradezco a mi querida amiga Brisaida Granado y familia, sus amigas que en el tiempo olvido sus nombres pero no su noble gesto en brindarme su apoyo al principio de mi carrera universitaria. Gracias de todo corazón.

A mi primo hermano Pedro Yáñez, Hilda Granado y sus hijos Maikel y Pierina, por su apoyo y ayuda, fue importante, gracia por su tolerancia y recibirme en hogar.

A los profesores que desde mis comienzos en la universidad hasta el final de mi carrera estuvieron presentes con su entereza en las aulas, sus sabios consejos y apoyo académico.

A mis amigos y compañeros universitario por su apoyo y consejos en los momentos oportunos.

A la Universidad de Oriente brindarme la oportunidad de pertenecer y poder decir soy UDISTA, gracias a ti recibí grandes conocimientos de lucha y perseverancia fue muy importante para alcanzar esta meta.

A la Sr. Isabel Tineo y familia por su hospedaje en su residencia, y tolerancia en los momentos difíciles gracia por su amistad nunca la olvidare.

Al Ingeniero Pedro Dorta mi asesor académico por su amistad y paciencia durante el desarrollo de mi proyecto. GRACIA MI PANA.

Anabel Velásquez, su prima Mary Carmen y todas sus amigas que en los últimos momentos de mi carrera universitaria fueron de gran apoyo emocional, alentándome cuando se presentaban momentos difíciles.

A mis compañeros de la residencia “La Madama”, por estar pendiente en la culminación de mí proyecto de trabajo de grado.

A todos y cada unas de aquellas personas que no menciono, pero que de algún modo me apoyaron con un grano de arena e hicieron posible la culminación de mi carrera universitaria. GRACIAS TOTALES...

INDICE DE CONTENIDO

RESOLUCIÓN	I
RESUMEN.....	II
DEDICATORIA	III
AGRADECIMIENTO	IV
INDICE DE CONTENIDO	VI
INDICE DE TABLAS	XV
INDICE DE FIGURAS.....	XVI
CAPITULO I – INTRODUCCIÓN	1
1.1 UNIVERSIDAD DE ORIENTE NÚCLEO DE ANZOÁTEGUI	1
1.2 PLANTEAMIENTO DEL PROBLEMA	3
1.3 OBJETIVOS	10
1.3.1 Objetivo General	10
1.3.2 Objetivos Específicos.....	10
CAPITULO II – MARCO TEÓRICO	12
2.1 ANTECEDENTES	12
2.2 EL LENGUAJE DE MODELADO DE MUNDOS VIRTUALES EN 3D VRML (LENGUAJE PARA MODELADO DE REALIDAD VIRTUAL).	14
2.2.1 Introducción al VRML	14
2.2.2 VRML – Definición Inicial.....	15
2.2.3 VRML – Definiciones Complementarías [10]	16
2.2.4 Como Opera el VRML [10]	16
2.2.5 Características del VRML [10]	18

2.2.6	VRML 1.0 – Características. [10]	19
2.2.7	VRLM 2.0: Mundos en Movimiento – Características. [10].....	20
2.2.8	Pros y Contras del VRML. [10]	21
2.2.8.1	Pros.....	21
2.2.8.2	Contras	22
2.2.9	Diferencias entre VRML 1.0 y VRML 2.0. [10].....	23
2.2.10	Mundos Virtuales en VRML. [10]	24
2.2.10.1	Consideraciones Iniciales.....	24
2.2.10.2	Creación de un Mundo Virtual.....	25
2.2.10.3	Componentes de un Mundo Virtual.	25
2.2.10.4	Pasos a Cumplir en la Construcción de un Mundo Virtual.....	26
2.2.11	VRLM 2.0 Comando Básico.	27
2.2.11.1	Componente de un Fichero.	28
2.2.11.2	Sintaxis Básica.	29
2.3	LA REALIDAD VIRTUAL RV [1]	35
2.3.1	Diferenciación Entre Ciberespacio Y Realidad Virtual	36
2.3.2	Características Básicas De La Realidad Virtual	36
2.3.3	Formas de Realidad Virtual.....	37
2.3.3.1	Ventanas Acopladas Visualmente o Sistemas de Inmersión	38
2.3.3.2	Sistemas Desktop De RV.....	38
2.3.3.3	Segunda Persona	39
2.3.3.4	Cabina de Simulación	40
2.3.3.5	Realidad Aumentada	40
2.3.3.6	Telepresencia	41
2.3.3.7	Realidad Virtual de Escritorio.....	41

2.3.4	Algunos Aspectos Importantes que hay que Tomar en Cuenta al Utilizar Rv	42
2.3.4.1	El Aspecto de la Inmersión y Percepción Visual	42
2.3.4.2	Consideraciones Básicas Para el Usuario: Implicaciones y Aplicaciones	42
2.4	AMBIENTES VIRTUALES AV	44
2.4.1	Componentes de los Ambientes Virtuales.....	45
2.4.2	Tipos De Mundos Virtuales.....	46
2.5	MÉTODOS DE TRABAJO EN REALIDAD VIRTUAL.	47
2.6	SISTEMAS DE BASE DE DATOS.	52
2.6.1	Hablemos de sistemas.....	52
2.6.2	Base de datos.	52
2.6.3	Componentes principales de una Base de Datos	53
2.6.4	Modelos de Base de Datos	54
2.6.5	Sistemas Administrador de Base de Datos (DBMS).....	55
2.6.6	Objetivo de un Sistema Administrador de Base de Datos.....	56
2.6.7	Base de Datos Multimedia en la WWW.....	57
2.6.8	La Base De Datos En Los Ambientes Virtuales.....	58
2.6.8.1	Métodos de Almacenamiento.....	58
2.7	EL PARADIGMA DE LA PROGRAMACION ORIENTADA A OBJETO.	59
2.7.1	El Enfoque Orientado a Objeto.	60
2.7.2	Ventajas De Usar El Enfoque O.O. [1]	62
2.7.3	Proceso Unificado.	63

2.7.4	Ciclo de Vida.....	64
2.7.5	Fases. [3].....	65
2.7.6	Lenguaje De Modelado Unificado (UML).....	66
2.7.6.1	Diagrama De Clases.....	68
2.7.6.2	Diagramas de Objetos	72
2.7.6.3	Diagrama de Componentes	74
2.7.6.4	Diagrama de Casos De Uso	75
2.7.6.5	Diagramas de Interacción.....	78
2.8	HERRAMIENTAS A UTILIZAR EN EL DISEÑO 3D.	85
2.8.1	3D Studio MAX [4].....	85
2.9	AUTOCAD [5]	86
2.9.1	El Cad	86
2.9.2	El AutoCad	86
2.9.3	Aplicaciones con AutoCad	86
2.10	COREL DRAW 10[6]	87
2.10.1	Características.....	87
2.10.2	Requisito mínimo del Corel Draw.....	88
2.11	VRMLPAD	89
2.11.1	Funciones.....	89
2.11.2	Características.....	89
2.11.3	Requisito mínimo de VrmlPad	90
2.12	OTRAS DEFINICIONES DE INTERÉS.	90
CAPITULO III – FASE DE INICIO		95
3.1	INTRODUCCIÓN	95

3.2	PLANIFICACIÓN DE LA FASE DE INICIO	96
3.3	Requisitos	97
3.3.1	Comprensión de los Requisitos	97
3.3.2	Análisis del Ambiente Virtual actual AVUDONA_1 (AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI VERSION 1).....	98
3.3.2.1	Objetivos del Ambiente Virtual AVUDONA_1	98
3.3.2.2	Características AVUDONA_1	98
3.3.2.3	Nivel de Detalle de AVUDONA_1	99
3.3.2.4	División AVUDONA_1.....	100
3.4	ANÁLISIS DEL AMBIENTE VIRTUAL PROPUESTO AVUDONA_2 (AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI VERSION 2)	101
3.5	DEFINICIÓN DEL NIVEL DE DETALLES DE AVUDONA_2	102
3.5.1	Muestreo de datos para definir el nivel de detalle de AVUDONA_2.....	103
3.5.2	Planos 2D, Planos de Levantamiento y Fachada de la Estructura Física de la Universidad.	104
3.6	MODELO DE DOMINIO	107
3.6.1	Contexto de AVUDONA_2	108
3.7	Requisitos Funcionales de AVUDONA_2	110
3.8	Requisitos no Funcionales de AVUDONA_2	111
3.9	Modelo de caso de uso	113
3.9.1	Identificación y Descripción de los Actores.....	113
3.9.2	Identificación De Los Casos De Uso De AVUDONA_2.....	114
3.9.3	Descripción de los Caso de Uso de AVUDONA_2	116

3.10	Requisitos Adicionales	117
3.10.1	Requisitos Adicionales de Avudona_2.....	118
3.11	Análisis	119
3.11.1	Diagrama de Casos de Análisis Para el Caso de Uso Moverse en el Ambiente Virtual.....	120
3.11.2	Diagrama De Colaboración.....	122
3.11.2.1	Diagrama de colaboración de Análisis del caso de uso Moverse en el AV.....	122
3.12	Diseño de AVUDONA_2.	124
3.13	Arquitectura de AVUDONA_2	124
3.13.1	Interfaz de AVUDONA_2.....	126
CAPITULO IV – FASE DE ELABORACION.....		129
4.1	INTRODUCCIÓN	129
4.2	Planificación de la Fase de Elaboración	130
4.3	Requisitos	131
4.3.1	Caso de Uso Detallado de AVUDONA_2.....	131
4.3.1.1	Descripción del Flujo de Sucesos del Caso de Uso “Moverse en el AV” (Figura 4.1).....	131
4.3.1.2	Descripción del flujo de datos del caso de uso “Animación” (Figura 4.2).....	133
4.3.1.3	Descripción del flujo de datos del caso de uso “Abrir Puertas del Departamento de Computación y Sistema” (Figura 4.3).....	135
4.4	ANÁLISIS	137
4.4.1	Diagrama de Caso de Análisis del Caso de Uso “Animación”.....	137
4.4.2	Diagrama de Clase de Análisis del Caso De Uso “Abrir Puertas De Computación Y Sistema”......	138

4.4.3	Diagrama de Colaboración	139
4.4.3.1	Diagramas de Colaboración “Animacion” de AVUDONA_2.	140
4.4.3.2	Diagramas de Colaboración “Abrir Puerta Dpto. Comp. y Sist” de AVUDONA_2.....	143
4.4.4	Diagrama de Paquete.....	145
4.5	DISEÑO	146
4.5.1	Diseño Detallado de AVODONA_2	146
4.5.1.1	Antecedentes y Propiedades Acerca de los Objetos a Modelar	147
4.5.1.2	Recolección de Muestras de las Texturas de los Objetos en el Entorno Real.....	148
4.5.1.3	Identificación de los objetos que van hacer modelados en VRML 148	
4.5.1.4	Iteración de una Escena VRML para el Diseño	152
4.5.1.5	Estructura de Ensamblajes de AVUDONA_2	156
4.5.2	Diagrama de Clase de Diseño.....	158
4.5.2.1	Descripción de las Clases de Diseño AVUDONA_2	159
4.6	DIAGRAMA DE SECUENCIAS	160
4.7	ARQUITECTURA LÓGICA DE AVUDONA_2	164
4.8	DISEÑO DEL SISTEMA DE ALMACENAMIENTO DE AVUDONA_2 (BASE DE DATOS).	166
CAPITULO V – FASE DE CONSTRUCCIÓN.....		169
5.1	Introducción	169
5.2	Planificación de la fase de construcción	169
5.3	Codificación de AVUDONA_2.	170
5.3.1	Construcción de Objetos de Manera Individual.	171
5.3.2	Construcción de las Zonas a Partir de los Objetos Creados.	173

5.3.3	Aplicación de los Mapas de Texturas a los Objetos.....	174
5.3.4	Incorporación de Luces Para Efectos de Profundidad y Sombras...	175
5.3.5	Realización de la Conversión de las Zonas a Código VRML.	176
5.3.6	Utilizar Vrmypad 2.1 Para Aplicación de Técnicas de Depuración al Código Vrmml Resultante Para Evitar la Pérdida de Detalles en la Conversión.	180
5.3.7	Incorporación de la Animación a los Objetos Cuyo Comportamiento es Constante.	183
5.3.8	Incorporación de la Animación a los Objetos Cuyo Comportamiento Depende de la Interacción con el Usuario o Cualquier otro Factor Externo Variante.	188
5.3.8.1	Animación Ardilla.....	188
5.3.8.2	Animación Centro de Fotocopiados de Ingeniería.....	191
5.3.8.3	Animación Puerta de Cubículo	197
5.3.9	Agregar los Efectos Sonoros.	198
5.3.9.1	Definición del Sonido Abrir Puertas del Ambiente Virtual.....	198
5.3.9.2	Definición del Sonido Límite del Ambiente Virtual.....	200
5.3.10	Acoplar las Actualizaciones y Nuevas Zonas con AVDONA_1.	200
5.3.11	Inclusión de las Guías de Navegación Interactivas Para los Usuarios.	201
5.3.12	Definición de la Forma de Desplazamiento de los Usuarios en el Ambiente Virtual.	202
5.3.13	Establecimiento de las Características del Clima (Cielo, Piso, Niebla).	204
5.3.14	Regulación de la Rapidez de Desplazamiento de los Usuarios por el Ambiente Virtual.	206

5.4	publicación de avudona_2	206
5.4.1	Interfaz HTML de AVUDONA_2.	207
CAPITULO VI – FASE DE TRANSICION		211
6.1	introducción	211
6.2	Planificación de la fase de transición	211
6.3	Prueba	212
6.3.1	PNC Aplicada al Edificio Decanato Perteneiente a la Zona 5.....	212
6.3.2	PNC Aplicada a la Animación Información Perteneiente a la Zona 6 213	
6.3.3	PNC Aplicada a la Animación luz Fachada Entrada de la Universidad Perteneiente a las Actualizaciones.....	214
6.4	Integración	215
6.5	Realizar la Prueba de integración	222
6.5.1	PNC Aplicada a la Intergracion de AVUDONA_2.....	222
CONCLUStoNES		225
RECOMENDACIONES		227
BIBLIOGRAFÍA		228
ANEXO.....		230

INDICE DE TABLAS

Tabla 2.1 Comparación Entre la Versión 1.0 y Versión 2.0.....	24
Tabla 2.2 Nodos que Soporta el VRML 2.0	31
Tabla 2.3. Nodos que Soporta el VRML 2.0	31
Tabla 3.1 Descripción de Planos.	105
Tabla 3.2 Descripción de los Objetos Dinámicos.....	106
Tabla 3.3 Identificación y Descripción de los Actores de AVUDONA_2.....	114
Tabla 3.4 Caso de uso “Moverse por el Ambiente Virtual”	116
Tabla 3.5 Caso de uso “Animación”	116
Tabla 4.1 Fotografías del Ambiente Real de la UDO a Modelar	147
Tabla 4.2 Identificación de Objetos (1/3).....	149
Tabla 4.3 Identificación de Objetos (2/3).....	150
Tabla 4.4 Identificación de Objetos (3/3).....	151
Tabla 5.1 Escena con LOD (Nivel de Detalle).....	182
Tabla 5.2 Textura de la Panorama Día	205
Tabla 5.3 Textura de la Panorama Noche.....	206

INDICE DE FIGURAS

Figura 2.1 Sistema de Coordenada, eje X (Amarillo), eje Y (Azul), eje Z (Rojo).	28
Figura 2.2 Estructura de un Nodo.	30
Figura 2.3 Un <i>Touchsensor</i> Envía un Solo Evento <i>Touchtime</i> (Sftime) a un Nodo Script Siempre que se Active. el Nodo <i>Script</i> Convierte el Sftime en un Sfbool y Manda el Resultado (True O False) al Campo <i>Enabled</i> del <i>Timesensor</i> que Comenzará Entonces (O Detendrá) su Marcha.	32
Figura 2.4 (a), (b), (c) <i>Secuencia de Imágenes Recreando el Programa Anterior, Rebote de la Pelota.</i>	34
Figura 2.5 Estructura del Proceso Unificado	64
Figura 2.6 Representación de un Diagrama de Clase.	71
Figura 2.7 Elementos que Forman Parte del Diagramas de Clase	72
Figura 2.8 Representación de un Diagrama de Objetos.....	73
Figura 2.9 Representación de un Diagrama de Componente de Windows.....	74
Figura 2.10 Ejemplo de la Relación en un Diagrama de Uso.	76
Figura 2.11 Ejemplo de Diagrama de Uso (Modelado de Contexto).	77
Figura 2.12 Representación Diagrama De Uso (Modelado de Requisito).....	78
Figura 2.13 Elementos que Forman el Diagrama de Secuencia.....	81
Figura 2.14: Ejemplo de Diagrama de Secuencia con Condiciones Que Generan Caminos Alternativos En Tiempo De Ejecución.	81
Figura 2.15 Elementos que Conforman un Diagrama de Estados.	84
Figura 3.1 Modelo de Dominio del AVUDONA_2.....	108
Figura 3.2 Modelo De Caso De Uso De AVUDONA_2.	115
Figura 3.3 Clase de Análisis “Moverse en el Ambiente Virtual”.	120
Figura 3.4 Diagrama de Colaboración de Análisis “Moverse en el AV”	123

Figura 3.5 Arquitectura de AVUDONA_2	126
Figura 3.6 Interfaz de Controles de AVUDONA_2.....	128
Figura 4.1 Caso de Uso “Moveirse en el AV”	132
Figura 4.2 Caso de Uso “Animación”.....	134
Figura 4.3 Caso de uso “Abrir Puerta del Departamento de Computación y Sistema”	135
Figura 4.4 Clase de Análisis “Animación”	138
Figura 4.5 Diagrama de Clase de Análisis del Caso de Uso Abrir Puertas de Comp. y Sistema.	139
Figura 4.6 Diagrama de Colaboración del Caso de Uso Animación.	140
Figura 4.7 Diagrama de Colaboración General de AVUDONA_2.....	143
Figura 4.8 Paquete de Análisis del Sistema AVUDONA_2.....	145
Figura 4.9 Textura Aplicada en el Diseño.	148
Figura 4.10 Componente del Visualizador VRML.....	152
Figura 4.11 Diagrama del Nodo Ardilla.	153
Figura 4.12 Diagrama del Nodo <i>PuetraSalaMicro</i>	155
Figura 4.13 Estructura de Ensamblaje de Nodo Ardilla	156
Figura 4.14 Diagrama de Ensamblaje del Nodo SalaMicroRC	157
Figura 4.15 Diagrama de Clase de AVUDONA_2.....	158
Figura 4.16 Diagrama de Secuencia del Recorrido Virtual	162
Figura 4.17 Diagrama de Secuencia de Animación, interacción con el usuario.....	163
Figura 4.18 Diagrama de Secuencias de Animación, Interacción con el Sistema	164
Figura 4.19 Arquitectura Lógica de AVUDONA_2.....	165
Figura 4.20 Modelo Conceptual de la Base de Datos del AVUDONA_2.....	168
Figura 5.1 Importación del Planos 2D a la Herramienta 3D Studio MAX.	171
Figura 5.2 Construcción del Objeto “Símbolo de Concreto” Plaza Decanato.....	172
Figura 5.3 Construcción del Objeto “Edificio Decanato”.....	172
Figura 5.4 Actualización y Construcción del “Portón” de la Entrada Principal	173
Figura 5.5 Zona 5 del Ambiente Virtual AVUDONA_2.....	174

Figura 5.6 Aplicación de Textura Ladrillo a la Pared Interior de la Sala “RENE CABRERA”	175
Figura 5.7 Efecto Luz Reflectores Entrada Universidad de Oriente.....	176
Figura 5.8 Ilustración Código AVUDONA_2 en VrmIPad 2.1	201
Figura 5.9 Interfaz Provisional para activar a AVUDONA_2.....	208
Figura 6.1 Integración del Código del Sub-Mundo de la Zona 5 a AVUDONA_1 en VrmIPad 2.1	216
Figura 6.2 Diagrama de Integración de AVUDONA_2.....	221

CAPITULO I – INTRODUCCIÓN

1.1 UNIVERSIDAD DE ORIENTE NÚCLEO DE ANZOÁTEGUI

La Universidad de Oriente fue creada el 21 de noviembre de 1958, mediante el Decreto Ley Nro. 459 dictado por la Junta de Gobierno presidida por el Dr. Edgard Sanabria, siendo Ministro de Educación el Dr. Rafael Pizani.

En su concepción, la UDO se define como un sistema de Educación Superior al servicio del país con objetivos comunes a las demás universidades venezolanas y del mundo.

Sin embargo, la UDO tiene fines propios derivados fundamentalmente de las características particulares de la región Insular, Nor-oriental y Sur del país, en cuyo desarrollo integral está comprometida.

La orientación general de la UDO, desde su creación, ha hecho de ella una institución de Educación Superior con organización diferente de las demás en Venezuela, con flexibilidad suficiente para ensayar diversos modelos estructurales y operacionales, desarrollando investigación científica, docencia y extensión en todos los aspectos del conocimiento, que contemplan sus programas educativos de pre y postgrado.

La UDO es un sistema regional de Educación Superior cuyo campus tiene su sede en los Núcleos Universitarios ubicados en los estados Anzoátegui, Bolívar,

Monagas, Nueva Esparta y Sucre, asumiendo así la responsabilidad de la Educación Universitaria en toda la región Insular, Nor-oriental y Sur del país.

El Núcleo de Anzoátegui, con una población estudiantil cercana a los 14 mil estudiantes, tiene su sede en Puerto La Cruz. Está formado por la Escuela de Ingeniería y Ciencias Aplicadas, la Escuela de Ciencias Administrativas, la Escuela de Medicina, el Instituto de Investigación y Desarrollo y una Unidad de Estudios Básicos.

El Departamento de Computación y Sistemas forma parte de la Escuela de Ingeniería y Ciencias Aplicadas de la Universidad de Oriente, Núcleo de Anzoátegui. El Departamento ofrece las carreras de Ingeniería en Computación e Ingeniería de Sistemas a nivel de pregrado.

El desarrollo académico del Departamento se inicia con la elaboración de los pensum de estudios de ambas carreras, estos son presentados ante el Consejo Universitario, el cual aprueba y ordena la ejecución de ambos pensum en la Escuela de Ingeniería y Ciencias Aplicadas del Núcleo de Anzoátegui, según resolución CU-02189 y CU-02289 de fecha 13-07-89.

La actividad académica se inicia el 15 de enero de 1990, con la ejecución de las asignaturas correspondientes al tercer semestre de cada uno de los pensum, luego se fue agregando un semestre en cada lapso académico, hasta completar los 10 semestres requeridos.

La creación formal del Departamento de Computación y Sistemas se aprueba por el Consejo Universitario en su reunión del 14-06-94, según resolución CU-861. Así mismo, el Consejo Nacional de Universidades, autoriza a la Universidad de

Oriente a otorgar los títulos de Ingeniero en Computación e Ingeniero de Sistemas, según comunicación de fecha 11-12-92.

1.2 PLANTEAMIENTO DEL PROBLEMA

Desde que la humanidad comenzó a utilizar la computadora como herramienta para la creación de medios de comunicación, estas se han convertido en una gran plataforma capaz de actuar como un “metamedio”, permitiendo englobar todas las alternativas existentes (transmisión de voz, sonidos, imagen, documentos, etc.), creando una sociedad con base en la comunicación y cooperación de diferentes grupos de individuos. En esencia, superando todas las limitaciones de cada una por separadas. Esta idea se concretó en la conformación de una gigantesca red de redes, que unió a todo el mundo, la famosa “INTERNET”.

La evolución y rápida expansión a todos los niveles de la tecnología aplicada para Internet ha permitido la incorporación de numerosas aplicaciones con la finalidad de hacer cada vez más atractivo para el usuario la experiencia de navegar por “INTERNET”. Estos factores tecnológicos han facilitado a los programadores utilizar todo este avance técnico en el diseño de innovadoras páginas WEB con interfaces graficas muy dinámicas incluyendo una variedad de formatos. En una página WEB se puede incluir diferentes tipos de información: imágenes, sonido, animaciones, videos, texto, imágenes tomadas en tiempo real, etc. Debido esencialmente a la aparición de nuevas necesidades por parte de los usuarios y al avance tecnológico desarrollado para el “INTERNET”, ha permitido el surgimiento de una particular interfaz que mejora la interacción humano-computadora como lo son los Ambientes Virtuales.

La “Realidad Virtual”, o los Ambientes Virtuales como otros también prefieren llamarlos, es la tecnología (software y hardware) que permite sumergir a un usuario en un ambiente tridimensional generado por el computador, de forma interactiva y autónoma, en tiempo real. En donde el usuario visualiza, manipulan e interactúan con mundos que no tiene existencia real, sino aparente, o lo que simula a su homónimo real. El disponer de mundos virtuales en las páginas Web posibilita en la actualidad un gran número de nuevas aplicaciones para las instituciones que utilizan el Internet en el ámbito corporativo.

Por tal motivo en la Universidad de Oriente, se desarrollo para ser incluido en su página Web de la misma un Ambiente Virtual prototipo en el departamento de Computación y Sistemas del Núcleo de Anzoátegui, AVUDONA1 (Ambiente Virtual de la Universidad de Oriente del Núcleo de Anzoátegui), innovando y actualizando a nivel de interfaz la presentación de su imagen a través de los medio que utiliza en el ámbito local, nacional y mundial. Este trabajo se desarrollo por los estudiantes Arcia Maican Jackson Ramón y Benítez González Allen Darío, como trabajo de grado para optar al título de Ingeniero en Computación. Para su desarrollo se utilizó una metodología estructurada que todavía no se ha estandarizado, tomada como referencia de toda una de serie de investigaciones sobre la secuencialidad de acciones en la creación de Ambientes Virtuales; los conceptos de la Ingeniería de Software orientada a objeto como punto base y los estereotipos definidos por el lenguaje unificado de modelado (UML).

Para conocer las necesidades de actualización de AVUDONA1 se utilizo una serie de técnicas de ingeniería de software que permitieron análisis, investigar, determinar y recolectar todas las información sobre las variables que son o no perceptibles por el usuario, evaluando el comportamiento del sistema, tanto en los detalles de navegación, como en el diseño de las imágenes 3D y los espacio físico reales que se involucraron en el recorrido virtual.

En el diseño del Ambiente Virtual se utilizaron programas de diseño de planos como Autocad, programas para el diseño tridimensional como 3D Studio Max y para la implementación del mundo Virtual en la página Web utilizaron el lenguaje VRML y programas para el tratamiento de texturas como Photoshop.

Para la codificación de AVUDONA versión 1.0 se utilizaron las herramientas para convertir las escenas 3D al código VRML (Lenguaje Modelado de Realidad Virtual) y JavaScript para crear cambios dinámicos en las escenas, son lenguaje de programación con una lógica sencilla, legible y fácil de entender.

La Universidad de Oriente Núcleo de Anzoátegui, con sede en Puerto La Cruz, Está formado por:

- ➔ La Escuela de Ingeniería y Ciencias Aplicadas.
- ➔ La Escuela de Ciencias Administrativas.
- ➔ La Escuela de Ciencias de la Salud.
- ➔ El Instituto de Investigación y Desarrollo.
- ➔ Unidad de Estudios Básicos.
- ➔ Edificio Académicos.
- ➔ Edificio Administrativos.
- ➔ Instalaciones de Recreación y Deportivas.

En el desarrollo de AVUDONA versión 1.0 solo lograron diseñar desde la fachada de la entrada principal del Núcleo hasta las instalaciones del Departamento de Computación y Sistemas. La presentación de la versión 1.0 se realizó al estilo de CD-ROM multimedia, por ser un tema novedoso. Conllevando así a restricciones en cuanto a su uso y acceso siendo este el principal limitante para ser incluido en la página Web de la Universidad y así permitir el conocimiento de las instalaciones a distancia.

Considerando estas deficiencias surge la necesidad de desarrollar la versión 2.0 de AVUDONA, el cual, se mantendrá como prototipo, ya que la finalidad es incrementar parcialmente el recorrido virtual, agregando nuevas escenas y actualizando los cambios que han ocurrido en las instalaciones físicas del Departamento de Computación y Sistemas. Facilitando la búsqueda y motivación, para lograr la virtualización de la Universidad y su página Web. Ajustando estos cambios a la evolución que viene presentando la tecnología aplicada al desarrollo de Ambientes Virtuales. Además que sirva como un gran aporte informativo sobre la realidad virtual y mundos virtuales se refiere.

En el desarrollo de la versión 2.0 AVUDONA, primeramente, se realizarán unas series de análisis de las debilidades, oportunidades, fortalezas, y amenazas de AVUDONA versión 1.0. Para la elaboración del recorrido virtual se fusionarán las metodologías de proceso unificado y la referencia conocida para crear Ambientes Virtuales. Para el diseño de las escenas se utilizaran las herramientas AutoCad, para diseños de los planos y 3D Studio Max, para el diseño tridimensional. En la implementación del mundo virtual en la página Web se utilizara los lenguajes de programación VRML, JavaScrip, la metodología para Sistemas gestores de Base de Datos y lenguaje de consultas estandarizado como el SQL (structured query language) para la implementación de la Base de Datos y las herramientas para el tratamiento de textura Corel Draw 10.

El propósito del presente proyecto es realizar la actualización y expansión de la versión 1.0 de AVUDONA, esto se logrará desarrollando nuevas escenas e integrándolas al Ambiente Virtual. Crear nuevas escenas permite expandir el recorrido virtual y realizar aquellos cambios que hayan ocurrido en las instalaciones física del Departamento de Computación y Sistemas, así ir conformando el Ambiente Virtual del Núcleo hasta alcanzar incorporar todas sus instalaciones, el cual, permita que la página Web del mismo posea una interfaz más atractiva e innovadora acorde

con las nuevas tecnologías, mejorando la interacción (por medio de la Web) entre los usuarios y la Institución, y permita el conocimiento de las instalaciones a distancia.

En la versión 2.0 de AVUDONA se actualizarán todas las escenas que hayan tenido cambios en su homónimo real y se agregará nuevas escenas al Ambiente Virtual, estas escenas que serán integradas son:

De la Fachada del Núcleo:

- ➔ Actualizar portón y El Jardín de la fachada de la Universidad.

De la Escuela de Ingeniería Y Ciencias Aplicadas:

- ➔ Actualizar sala de Micros “RENE CABRERA”.
- ➔ Aula de Ingeniería de Computación y Sistemas (AICS).
- ➔ Sala de Micro II.
- ➔ Jefatura del Departamento de Computación y Sistemas.
- ➔ Cubículos del Departamento de Computación y Sistemas (cubículos 1; 2; 3).
- ➔ Centro de Fotocopiados.
- ➔ Escuela de Ingeniería Industrial (fachada exterior).
- ➔ Anfiteatro (fachada exterior).
- ➔ Pasillos de accesos a las distintas instalaciones.

De Instituciones de Investigación y Desarrollo, Académicos y Administrativos:

- ➔ Decanato, Biblioteca (fachada exterior).
- ➔ Plaza central, plaza del Decanato, estacionamientos.
- ➔ Pasillos de accesos a las distintas instalaciones.

Es importante realizar la actualización de la versión 1.0 de AVUDONA, porque radica primordialmente en presentar una extendida tendencia a la virtualización de la página Web de la Universidad. Por otra parte, permite crear nuevos objetos, donde cada escena se compone de objetos, para posteriormente ser agregada al recorrido y permitan expandir el Ambiente Virtual del Núcleo. Al mismo tiempo se aplicaran todos los avances tecnológicos que han surgiendo en cuanto a realidad virtual y Ambientes Virtuales se refiere, adaptando a las nuevas necesidades en el medio de presentación de las escenas, mediante cambios en la técnicas de almacenamiento y como generar de los objetos.

Cabe destacar que esta investigación está orientada a realizar un aporte tanto práctico como tecnológico a la Universidad, al país y al mundo; actualmente el tema de la Realidad Virtual está representando lo último en tecnología de interfaz y está siendo aplicada a cualquier campo. La consecuencia de este proyecto es la de impulsar a las instituciones superiores a actualizar la calidad de la información integral que se le imparte al estudiantado. Además, se busca modelar una metodología que potencie el desarrollo de habilidades y destrezas asociadas al uso de las técnicas de la realidad virtual.

La actualización de AVUDONA está motivada principalmente, por ir siguiendo los avances tecnológicos mundiales, que permita la competitividad de los egresados de esta universidad con cualquiera otra en todo el mundo, brindando una imagen innovadora, en donde, se realizan trabajos no sólo prácticos sino científicos, con una capacidad de actualización completamente visible.

El alcance de la versión 2.0 de AVUDONA trascenderá a un recorrido de las instalaciones de la Universidad a distancia, en línea, utilizando tecnología Word Wide Web e Internet como medio de transporte, siendo requisito una computadora con los requerimientos mínimos para la visualización de interfaz 3D, un MODEM conectado

a una línea telefónica y un programa denominado navegador, teniendo como ventajas ofrecer el fácil acceso al usuario desde cualquier lugar y hora.

En el recorrido virtual AVUDONA versión 2.0, el usuario se encontrará una representación virtual de su homónimo real, donde podrá recorrer y visualizar desde varias perspectivas cada unas de las escenas que conforman el Ambiente Virtual.

1.3 OBJETIVOS

1.3.1 Objetivo General

Actualizar el Ambiente Virtual prototipo del Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui, incluyendo nuevas escenas que permitan expandir el recorrido virtual, aplicando las nuevas tecnologías y herramientas que han surgido para el desarrollo de Ambientes Virtuales e interfaz 3D.

1.3.2 Objetivos Específicos.

1. Realizar el análisis y especificación de los nuevos requerimientos funcionales y no funcionales de AVUDONA versión 1.0, para hacer una descripción de su nueva versión.
2. Analizar el diseño del Ambiente Virtual AVUDONA versión 1.0, a fin de estudiar todas las factibilidades de los elementos que fueron incluidos, así comprobar todos los recursos técnicos y tecnológicos que fueron implicados, para determinar el nivel de detalle de la nueva versión.
3. Analizar las zonas en la que fue dividido el Ambiente Virtual, para verificar cuales necesitan actualización y definir las nuevas zonas que serán agregadas.

4. Elaborar una lista de los tipos de objetos que existen, asociada a las estructuras físicas del núcleo, sus características, para crearlos de manera individual utilizando las herramientas de diseño 3D.
5. Aplicar y adaptar los detalles de realismo a los objetos creados, utilizando la herramienta de diseño 3D (Mapas de Textura, Efectos de Profundidad y Sombra).
6. Desarrollar el código VRML de las escenas 3D de cada zona creada, utilizando las herramientas de diseño 3D y crear los efectos de animación a los objetos de las escenas.
7. Realizar las pruebas de cada uno de las escenas creadas para ser integrada en su respectiva zona aplicando técnicas de desarrollo incremental para obtener una versión preliminar.
8. Definir los detalles de navegación, en cuanto a la forma, rapidez de desplazamiento y condiciones del clima (Cielo, Piso, Niebla) en el Ambiente Virtual.
9. Elaborar la base de datos que justifique la estabilidad e integridad de las zonas que conforma el Ambiente Virtual, utilizando Sistemas Base de Datos para interfaz VRML
10. Realizar las pruebas de funcionamiento del Ambiente Virtual completo.
11. Realizar el manual de usuario y mantenimiento del Ambiente Virtual.

CAPITULO II – MARCO TEÓRICO

2.1 ANTECEDENTES

La aplicación de Ambiente Virtuales en las páginas Web, estas representando hoy en día una impresionante evolución que se proyecta en todas las direcciones posibles, lo cual permite desarrollar un sin fin de proyectos, enmarcados hacia el diseño de interfaz tridimensionales.

En la actualidad en la Universidad de Oriente, Núcleo de Anzoátegui, se utiliza la Internet para ofrecer información sobre la institución a través de su página Web. Por tal motivo se ha elaborado y están desarrollando trabajos de grado, basado en Ambientes Virtuales, el cual, puedan ser incluidos en su página Web y permita proporcionar al mundo una imagen innovadora acorde con los avances tecnológicos mundiales, marcando así el paso a seguir para la virtualización de nuestra página Web. De esta forma se establece una base teórica para la realización de futuros trabajos relacionados con el tema. Reflejando, así, la competitividad de los egresados de esta universidad con cualquiera otro en todo el mundo.

Todo esta búsqueda ha permitido recolectar toda la referencias que sirvan para el desarrollo de este proyecto; iniciando con una búsqueda de fuentes de información, efectuando revisiones bibliográficas con el fin de establecer un marco de referencia basado: primero, en los trabajos realizados en el área de Ambientes Virtuales dentro y fuera de la U.D.O.; y segundo, en los conocimientos previos que deben tomarse en cuenta para desarrollar el proyecto. Los trabajos que fueron tomados como marco de referencia; se nombran a continuación:

Del Departamento de Computación Y sistemas de la Universidad de Oriente (UDO), Núcleo de Anzoátegui Venezuela, el trabajo de grado que titula; **“Desarrollo de Ambiente Virtual prototipo del Departamento de computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui”**, elaborado por: Arcia M. Jackson R. y Benítez G. Allen D.

Del Centro Virtual del Instituto Cervantes en el Museo de Prado, Madrid España, el trabajo titulado: **“Salón de Reinos en Realidad Virtual”**, elaborado por: Luis Morán y Yolanda Arribas.

De la Facultad de Arquitectura y Urbanismo de la Universidad Central de Venezuela (UCV), el proyecto de titulación; **“Centro Virtual de Arte Digital Venezolano”**, elaborado por: Cesar Salguero y Ernesto Bueno.

Del Departamento de Ingeniería en Sistemas Computacionales de la Universidad de las Américas – Puebla, el trabajo de tesis titulado: **“Aplicación de mapas cognitivos difuso a VRML en un Ambiente Virtual”**, elaborado por: Von Raesfeld Porras.

De la Facultad de Arquitectura y Diseño en la Universidad del Zulia. Venezuela. (LUZ), el proyecto titulado: **“DESARROLLO Y APLICABILIDAD DE MENÚS VIRTUALES EN VRML”**, elaborado por: Gabriela Bustos L. y Iván Burgos P.

2.2 EL LENGUAJE DE MODELADO DE MUNDOS VIRTUALES EN 3D VRML (LENGUAJE PARA MODELADO DE REALIDAD VIRTUAL).

2.2.1 Introducción al VRML

En términos conceptuales podemos concebir al VRML (Lenguaje de Modelado de Realidad Virtual) como un subconjunto de recursos perteneciente a esa gran familia de tecnologías conocida como Realidad Virtual. Dentro de esa óptica abarcamos un gran escalón en cuanto al avance de los medios interactivos y multi-sensoriales apoyados por computadora en la búsqueda de nuevas y más completas formas de visualizar nuestra realidad e, inclusive de visualizar otras realidades fruto de nuestra imaginación, nuestro ingenio y nuestro talento.

Igual que el HTML nos sirve para maquetar páginas Web, VRML sirve para crear mundos en tres dimensiones a los que accedemos utilizando nuestro navegador, igual que si visitásemos una página Web cualquiera, con la salvedad que nuestras visitas no se limitan a ver un simple texto y fotografías, sino que nos permite ver todo tipo de objetos y construcciones en 3D por los que podemos pasear o interactuar.

Este modo de visitar sitios en Internet es mucho más avanzado y posee grandes ventajas. Para empezar la navegación se desarrolla de una manera mucho más intuitiva, dado que la forma de actuar dentro del mundo virtual es similar a la de la vida real. Podemos movernos en todas las direcciones, no solo izquierda y derecha sino también adelante, atrás, arriba y abajo. Podemos tratar con los objetos como en la vida misma, tocarlos, arrastrarlos, etc. y en general todo lo que puedas imaginar.

Por último, destacamos el hecho de que la incorporación del VRML abre puertas al uso tridimensional del Ciberespacio, siguiendo nuestra forma natural de ver, hacer las cosas y restringiendo, de paso, el caos de comunicación bidimensional que nos invade progresivamente con la incorporación con propósitos de navegación y visualización de las cada vez mas aceptadas Metáforas Arquitectónicas como recursos para la navegación, la visualización, la comprensión de objetos y mundos ciberespaciales. A la vez el VRML nos ofrece la fascinante posibilidad de una creciente capacidad de intercomunicarnos en el nuevo medio, tanto con los recursos tecnológicos ofrecidos por la época como con los otros seres humanos que asisten a esta vigorosa colonización del medio conocido como Ciberespacio, en las postrimerías del presente Milenio.

2.2.2 VRML – Definición Inicial.

"El Lenguaje de Modelación de Realidad Virtual (VRML) es aquel orientado a la descripción de simulaciones interactivas de multiparticipación en mundos virtuales conectados vía Internet hipervinculados con la WWW (World Wide Web). Todos los aspectos de representación, interacción e interconexión en la red pueden ser especificados utilizando VRML. Es la intención de sus diseñadores que VRML pueda convertirse en el lenguaje estándar para simulación interactiva dentro de la WWW". [10]

2.2.3 VRML – Definiciones Complementarias [10]

Otras definiciones aparecidas posteriormente y que pueden ayudar a enriquecer el contexto de la anteriormente citada son:

"VRML es un lenguaje abierto y extensible, estandarizado para la industria, orientado a la descripción de escenas o mundos 3D que permite la visualización de mundos tridimensionales distribuidos e interactivos enriquecidos mediante la incorporación de texto, imágenes, animación, sonido, música e incluso video".

"VRML es un lenguaje de programación en la Internet que permite describir ambientes tridimensionales a través de los cuales se navega en forma análoga a lo que sería el espacio real y casi con la misma facilidad".

"VRML es un lenguaje para escribir aplicaciones tridimensionales del HTML. Constituye el estándar en toda la WWW y está abriendo el camino para la Realidad Virtual en la Internet. Las paginas elaboradas en VRML pueden "bajarse" (download) y sus contenidos entonces visualizados, rotados y manipulados, permitiendo el desplazamiento a través de los espacios virtuales creados".

2.2.4 Como Opera el VRML [10]

El elemento programático básico del VRML se denomina NODO. Existen nodos de distintas clases como veremos posteriormente. Un nodo equivale a una variedad de objeto y sus características están definidas por las funciones que aloja.

Los Nodos están ordenados según estructuras jerárquicas denominados "GRAFOS DE ESCENA" (scene graphs).

Los objetos empleados en VRML están basados en polígonos tridimensionales básicos. Aun cuando entendemos usualmente, como una forma bidimensional, en un espacio tridimensional le confiere la connotación de "polígono tridimensional". Los programadores de formas pueden posteriormente cubrir las formas poligonales con texturas y patrones gráficos tales como piel, grama o rayados y añadir fuentes de luz y sombras que busquen a las formas aparecer como personas reales, como paisajes o con objetos.

También pueden construir objetos "biblioteca"("librería") basados en estos prototipos iniciales para simplificar grandemente las tareas de programación y construcción de mundos. Los programadores pueden crear simulaciones interactivas de casi cualquier cosa. Utilizando editores o herramientas de construcción interactiva de mundos es posible obtener listados parciales de las actividades en curso lo que permite al usuario no especializado constatar los comandos y efectos de su interacción y su eventual corrección.

VRML permite que información procedente de otras fuentes pueda ser fácilmente integrada en un mundo tridimensional interactivo. También es "escalable" a través de diversa plataformas que van desde PCs, Macs y estaciones de trabajo "high-end". Cuando los ficheros de VRML se hacen muy extensos ello se debe al esfuerzo por capturar movimiento, animación, sonido o video. Todo esto se simplificara y reducirá sustancialmente a medida que el concepto de "medio confluyente" (streaming media) se operativice haciéndose realidad.

2.2.5 Características del VRML [10]

- Es un formato de descripción de escenas tridimensionales dotado de conectores (plug-ins) que suministran accesibilidad en redes de naturaleza multiusuario.
- Requiere de un visualizador (Browser) especial para mostrar aquellos gráficos que simulan Realidad Virtual bajo la forma de "ambientes" o "mundos" a través de los cuales el usuario puede moverse e interactuar con los objetos que contienen. Estos tiene la capacidad de poder vincularse a documentos, otros objetos e, inclusive, conectarse a otros mundos 3D.
- Es una máquina de representación tridimensional en tiempo real, la misma tecnología empleada en la Realidad Virtual tradicional (RV).
- Un fichero (file) VRML reside en un sitio determinado de la Red al igual que lo hace un fichero HTML. Es de carácter textual o textual comprimido y su nombre posee una terminación ".wrl" (World, mundo), ".wrl.gz" o ".wrz". Al contenido de un fichero VRML se le llama un "mundo".
- Los Fichero son automáticamente "bajado" (downloaded) al Browser de la Web. Cuando este browser detecta un archivo en VRML con las terminaciones anteriormente citadas, procede a ordenar a la computadora la activación del visualizador (browser, viewer) especial para mundos virtuales.
- Existen tres categorías de visualizadores ("browsers") VRML: autónomos o independientes (stand alone), auxiliares (helpers) y conectados (plug-ins).

- Comparte el protocolo HTTP y el URL con el HTML pero en el proceso de construcción de documentos son totalmente diferentes.
- Diferencia mayúscula de minúsculas (es 'case-sensitive').
- Cada fichero VRML contiene exactamente un nodo VRML, al cual se denomina 'GRAFO DE ESCENA' (scene graph). Puede ser un nodo de grupo y alojar cualquier cantidad de otros nodos. Todo nodo posee las siguientes características:
 - Los PARAMETROS que distinguen a un nodo de otros nodos se llaman CAMPOS. Un nodo puede poseer cero o más campos.
 - Un NOMBRE para identificar el nodo. Esto permite reutilizarlo y referirse a ellos, es una exigencia obligatoria deben ser de una sola palabra y únicos.
 - NODOS "HIJOS"; Los nodos de grupo pueden poseer cero o más nodos "hijos".

2.2.6 VRML 1.0 – Características. [10]

- Muy limitado comportamiento interactivo.
- Es posible atribuir diversas propiedades a los objetos (nodos), como color y brillo.
- El grafo de escena VRML está constituido por una jerarquía de nodos que representan la totalidad de los aspectos ESTATICOS de una escena tridimensional:

- Geometría
- Materiales
- Texturas
- Transformaciones geométricas.
- Luces.
- Puntos de vista.
- Estructuras anidadas.

2.2.7 VRML 2.0: Mundos en Movimiento – Características. [10]

La denominación de Moving World resume todo lo que el VRML 2.0 es capaz de hacer: Movimiento y dinamización de mundos. Ejemplos de estas capacidades potenciales son:

- Luces que se encienden y apagan virtualmente.
- Los sonidos incrementan gradualmente a medida que el usuario se acerca.
- Puertas virtuales que se abren o cierran automáticamente al percibir la cercanía del usuario.
- Personas virtuales que se mueven entre rutas.
- Conectan y controlan acciones, reacciones y animaciones por medio de los nodos llamados eventos, rutas, sensores, interpoladores.
- Hacer referencia un applet de Java externo que permite a los desarrolladores extender el comportamiento y la dinámicas de los VRML, se denomina, nodos guiones (scripts).
- Dos nodos especiales de VRML, Proto y ExternProto pueden también extender dinámicamente al VRML de cualquier sitio en la Web.

- Mejora la apariencia del VRML al suministrar gradientes y escenarios texturizados, sonidos vinculados a diferentes ubicaciones dentro del mundo.
- Permite "mapear" videos MPEG-1 sobre objetos elegidos arbitrariamente utilizando un nodo denominado MoviTexture.
- Permite hacer contornos de terrenos, extrusiones, detección de colisiones, neblina y texto permanentemente legible por el usuario.
- Visualización 3D de la base de datos en tiempo real.
- Desplazamiento (walk-through) a través de espacios virtuales.
- Multi-participación en juegos de Realidad Virtual.
- Interacción en ambientes multimediativos de conversación chats.
- Estudio de geografías fotorealísticas 3D.
- Interconexión de mundos de gran tamaño.
- Los autores pueden insertar/añadir o reemplazar totalmente una escena.
- Interpoladores que facilitan animación e interacción.

2.2.8 Pros y Contras del VRML. [10]

2.2.8.1 Pros

Los propulsores del VRML alegan que el lenguaje ofrece las siguientes ventajas:

- Ofrece independencia de plataformas. Tecnología "abierta". Estándar ISO.
- Se apoya en un conjunto de recursos gratuitos o de "shareware". (Visualizadores, constructores, herramientas varias).
- Ofrece facilidad de aprendizaje.

- Incrementa la capacidad de comunicación interactiva en Red.
- Ofrece rapidez de construcción de mundos sencillos.
- Supone manejo más compacto de información. Y mayor facilidad en la transferencia de información.
- Aporta una facilidad de visualización.
- Constituye un escalón hacia el manejo de información en 4D.
- Representa una evolución de los medios, y esto no tiene marcha atrás.
- Muchas limitaciones actuales van desapareciendo con el incremento progresivo del poder y velocidad de procesamiento de las computadoras y el incremento en el ancho de banda que se prevé en los próximos años.
- Integración progresiva de matemática/ciencia/tecnología/arte/diseño.
- Constituye un recurso importante de apoyo para la publicidad en la feroz competencia comercial que se apoya en el diseño de páginas WEB.
- Promueve un acercamiento entre clientes y profesionales en aquellas áreas que dependen de la modelación tridimensional para la visualización de situaciones y productos complejos.

2.2.8.2 Contras

Entre las debilidades que se detectan en el desarrollo uso actual del VRML se encuentran:

- Está aún distante de desarrollar el potencial que promete.
- Se encuentra sujeto la deformación intencional de estándares, por parte de empresas vendedoras.
- Las nuevas versiones del VRML evidencian creciente complejidad en su contenido.

- El incremento de poder y de amplitud de banda requerido para la visualización (browsing) de mundos virtuales es elevado.
- Presenta dificultades de conciliación con respecto al CAD y sus objetivos de producción y en la precisión de escala ofrecida actualmente por el VRML para armonizar objetos muy pequeños y muy grandes.
- Estaríamos mejor refinando y optimizando los medios ya existentes.
- No alcanza aún la calidad de resolución que ofrece la representación de objetos y escenas tridimensionales con herramientas de computación grafica más convencionales.
- La navegación de espacios tridimensionales no pre-programados en base a "ratón" es dificultosa de dominar con fluidez.

No es un avance autentico. Obedece a los intereses y beneficios de los grupos de Silicon Valley.

2.2.9 Diferencias entre VRML 1.0 y VRML 2.0. [10]

El VRML 2.0 representa un avance en muchos frentes relación a la versión VRML 1.0. Esto evidentemente origina un número de modificaciones que alejan operativamente la compatibilidad entre las dos versiones: Surgen nuevos nodos, se eliminan algunos existentes, se modifican otras.

Es importante destacar la conveniencia de disponer de dos tipos de browsers (visualizadores): uno para los modelos creados en VRML 1.0 y otro para modelos en VRML 2.0. En la Tabla 2.1 presentamos seguidamente un resumen de las principales diferencias entre las dos versiones.

Tabla 2.1 Comparación Entre la Versión 1.0 y Versión 2.0.

Aspectos	VRML1.0 (1995)	VRML 2.0 (1996 – 97)
Interacción Individual	Objetos Estáticos	Con Objetos
Animación	Primaria	Compleja
Uso del Java	No	Si
Comportamiento	No	Si
Sonido 3D	No	Si
Iluminación	Baja Calidad	Mejor Calidad
Sensores	No	Si
Extensibilidad	Muy Limitada	Admite Java
Color	Baja Calidad	Mejor Calidad
Avatares	No	Incorpora Concepto
Lenguaje de guiones (Scripts)	No	No Especificados

2.2.10 Mundos Virtuales en VRML. [10]

2.2.10.1 Consideraciones Iniciales.

Dejada atrás ya la fase de conceptos elementales sobre el tema de VRML, se supere los sencillos ensayos iniciales requeridos para una familiarización inicial con el VRML, se pasa a una etapa de aplicación aceptando el hecho de que enfrentaran el reto de construir mundos de mayor tamaño y complejidad. Será necesario entonces precisar procedimientos metodológicos, utilizar ciertas herramientas complementarias, esencialmente modeladores gráficos y de escena y prestar atención a la experiencia de los que antes han recorrido el camino.

2.2.10.2 Creación de un Mundo Virtual

Existen tres modalidades de construcción de mundos virtuales:

- Crear y editar, a mano, un fichero texto en VRML.
- Utilizar un programa de conversión para convertir un fichero 3D existente, no-VRML, a VRML.
- Utilizar un paquete de Software para crear modelos y posicionarlos dentro de un mundo.

Debido a que VRML es aún una nueva tecnología hay altas posibilidades de que, aun en el caso de utilizar una herramienta de diseño o convertidor para crear un mundo virtual, será necesario introducir manualmente ligeras modificaciones al listado ASCII del mundo. En consecuencia, resulta aconsejable que el estudiante se familiarice con los conceptos básicos y la sintaxis del VRML.

2.2.10.3 Componentes de un Mundo Virtual.

Todo mundo virtual debe poseer necesariamente:

- Una ESCENA dentro de la cual desarrollarse.
 - Escala.
- Un conjunto de objetos gráficos y sus características.

- Forma
- Color
- Textura
- Comportamiento
- Ubicación.

➤ Unas condiciones ambientales entre las que figuran:

- Luz
- Sonido.

2.2.10.4 Pasos a Cumplir en la Construcción de un Mundo Virtual.

Existen varios caminos. Como ejemplo adoptamos las siguientes secuencias:

- Concepción inicial de mundo, sus componentes y comportamiento.
- Utilización de un modelador geométrico para construir objetos del mundo.
- Exportar los OBJETOS contruidos a ficheros VRML.
- Recolectar en los repositorios VRML, si fuere requerido, modelos menores, que pueden estar en diferentes tipos de formato 3D.
- Recolectar y/o construir las texturas que van a utilizarse.
- Utilizar algunos convertidores (converters) para poder convertir ficheros de VRML 1.0 a 2.0; ficheros 3D y DXF a VRML y demás.
- Agrupar objetos en una ESCENA
 - Iluminarlos
 - Texturarlos

- Añadir comportamientos.
 - Preparar cámaras o las vistas de cámaras (viewpoint cameras).
- Ajustar (tweak) la escala de objetos y escena de modo tal que se reduzca el riesgo de operar con el producto directo del convertidor que muchas veces adopta un tamaño excesivo y lento para "cargar" (load) y representar (render).
 - Probar el resultado y realizar los ajustes necesarios.
 - Hacer accesible el mundo al público general y publicitarlo a través de las "máquinas de búsqueda" (search engines).

2.2.11 VRML 2.0 Comando Básico.

El lenguaje VRML 2.0 será el soporte en este proyecto, aparte de la descripción de conceptos clave para la comprensión del mismo, y las nuevas opciones. Es capaz de representar Objetos estáticos y/o animados, soporta hipervínculos enlaces con sonidos películas e imágenes. De aquí en adelante las referencias a la Versión 2.0 se harán únicamente escribiendo VRML.

La unidad de medición no está definida por el VRML. En consecuencia, se asume que todas las distancias lineales están en metros y los ángulos en radianes, así que, para hacer la rotación tendrías que multiplicar el número de grados por **$\pi/180$** . Por su parte, las unidades de tiempo se especifican en el espacio de color RGB (Red-Green-Blue) y están regidos al rango 0 a 1.

Utiliza un sistema cartesiano de coordenadas tridimensional, el eje-X crece hacia la derecha, eje-Y crece hacia izquierda, eje-Z crece hacia arriba (ver Figura 1). Las escenas pueden contener un número arbitrario de sistemas de coordenadas

locales, definidos por los campos de transformación en los nodos Transform y Billboard.

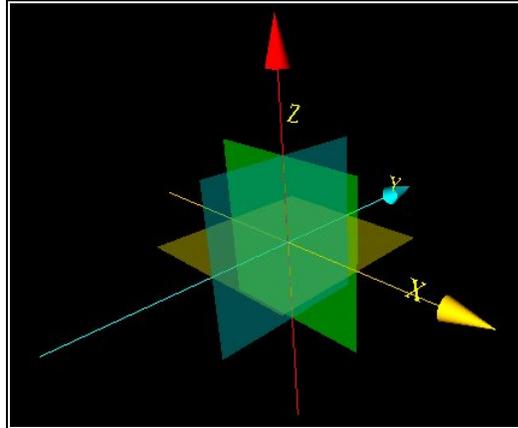


Figura 2.1 Sistema de Coordenada, eje X (Amarillo), eje Y (Azul), eje Z (Rojo).

2.2.11.1 Componente de un Fichero.

Un fichero VRML puede contener cuatro tipos de componentes:

- ENCABEZADO ("header").
- PROTOTIPOS.
- NODOS: FORMAS, INTERPOLADORES, SENSORES Y GUIONES ("Scripts").
- RUTAS O ENRRUTAMIENTO DE EVENTOS.

No todos los ficheros deben contener todos estos componentes. De hecho, el único obligatoriamente requerido es el encabezado. Por otra parte, y a nivel más desagregado, un fichero VRML puede contener: comentarios, nodos y sus nombres, campos y sus valores.

2.2.11.2 Sintaxis Básica.

➤ Encabezado y Comentarios de Programas en VRML

Todo programa en VRML debe iniciarse de la siguiente forma:

```
#VRML V2.0 utf8
```

La parte del **utf8** dice al navegador qué es un *standard* de caracteres internacionales *UCS (Universal Character Set)*, cadenas de texto que debe emplear, *Transformation Format, 8-bit*. Es una extensión al código ASCII. Los comentarios comienzan con el carácter “#”, por ejemplo;

```
#Archivo VRML así se escribe un comentario.
```

➤ Prototipos

Los prototipos permiten que el usuario extienda el conjunto de tipos de nodos de VRML. La definición de éstos puede ser incluida en el archivo en el que son usados o en otros archivos (estos prototipos se conocen como externos) y ésta puede estar dada en términos de los nodos VRML o usando mecanismos de extensión específicos al *browser*. Algunos nodos VRML generan eventos en respuesta a cambios en el Mundo Virtual, interacción del usuario con éste, o el paso del tiempo durante la ejecución del programa VRML. [1]

➤ Nodos (Grafos de Escenas)

Un mundo VRML está hecho a partir de nodos, que son tipos de objeto que describen el contenido de las escenas (formas, luces, sonidos, etc.), ver figura 2.2. Dentro de estos nodos se escribe lo siguiente:

```
[DEF <nombre>] <tipo_nodo > {conjunto de < campo > < valor >}
```

- **DEF** se utiliza para definir un nombre a un tipo de nodo.
- *<nombre>* sirve para hacer referencia al nodo en otras partes del programa.
- El tipo identifica al nodo (*Shape, Light, ElevationGrid, etc.*).
- Un campo describe atributos del nodo.
 - ✓ Nombre del campo (*height, radius, coordIndex, etc.*).
 - ✓ Tipo de dato asociado (*SFInt32, SFFloat, etc.*).
 - ✓ Valor por defecto.

Los elementos encerrados entre *[]* son opcionales y pueden omitirse en la definición del nodo.

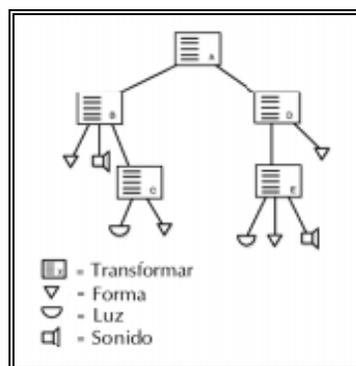


Figura 2.2 Estructura de un Nodo.

➤ **Grupo de Nodos.**

Existen un gran número de nodos que a continuación nombraremos los más importantes en las tablas 2.2 y tabla 2.3, siguientes

Tabla 2.2 Nodos que Soporta el VRML 2.0

Nodos comunes	Apariencia	Nodos de agrupamiento	Sensores
AudioClip	Appearance	Anchor	CylinderSensor
DirectionalLight	FonStyle	Billboard	PlaneSensor
Pointlight	ImageTexture	Collision	ProximitySensor
Script	Material	Group	SphereSensor
Shape	MovieTexture	Transform	TimeSensor
Sound	PixelTexture		TouchSensor
Spotlight	TextureTransform		VisibilitySensor
Worldinfo			

Tabla 2.3. Nodos que Soporta el VRML 2.0

GEOMETRÍA	INTERPOLADORES	AGRUPAMIENTOS ESPECIALES	NODOS DE ACTIVACIÓN EXCLUYENTE
Box	ColorInterpolator	Inline	Background
Cone	CoordinateInterpolator	LOD	Fog
Cylinder	NormalInterpolator	Switch	NavigationInfo
ElevationGrid	OrientationInterpolator		Viewpoint
Extrusion	PositionInterpolator	PROPIEDADES GEOMÉTRICAS	
IndexedFaceSet	ScalarInterpolator	Color	
IndexedLineSet		Coordinate	
PointSet		Normal	
Sphere		TextureCoordinate	
Text			

➤ Enrutamiento de Eventos.

Necesitamos saber un poco más sobre los eventos mismos. Los eventos son mensajes que ligan los elementos de la escena. Todo lo que se mueve o interactúa en VRML lo hace debido a los eventos. Son la clave de todo este asunto. Un evento posee dos partes: el propio mensaje, que es un valor o dato de un cierto tipo, y un *time stamp*, o estampa de tiempo. El valor del mensaje puede ser de cualquier tipo,

por ejemplo *SFTime*, *SFString*, *MFNode*. Si los nodos pueden manejarlo, tú puedes enviarlos. La estampa de tiempo es un valor especial sobre el que no tienes ningún control. Es un valor que corresponde al momento preciso en que el evento se produjo, no cuando el evento ocurrió para ser generado por el nodo. Cada nodo tiene:

field <tipo> <nombre_campo> campo donde se almacena el valor de un dato

eventIn <tipo> <nombre_eventIn> evento de entrada

eventOut <tipo> <nombre_eventOut> evento de salida

exposedField <tipo> <nombre_campo> campo que equivale a
field <tipo> <nombre_campo>,

eventIn <tipo> **set_<nombre_campo>**

eventOut <tipo> <nombre_campo> **_changed**

Si un evento se genera, y esto causa otro evento en otro nodo, y así continuamente una y otra vez, se dice que se está produciendo una *casca de eventos*. Para entender un poco más ver figura 2.3, se representa gráficamente un modelo complejo de un enrutamiento de evento iniciado en el nodo *TouchSensor*.

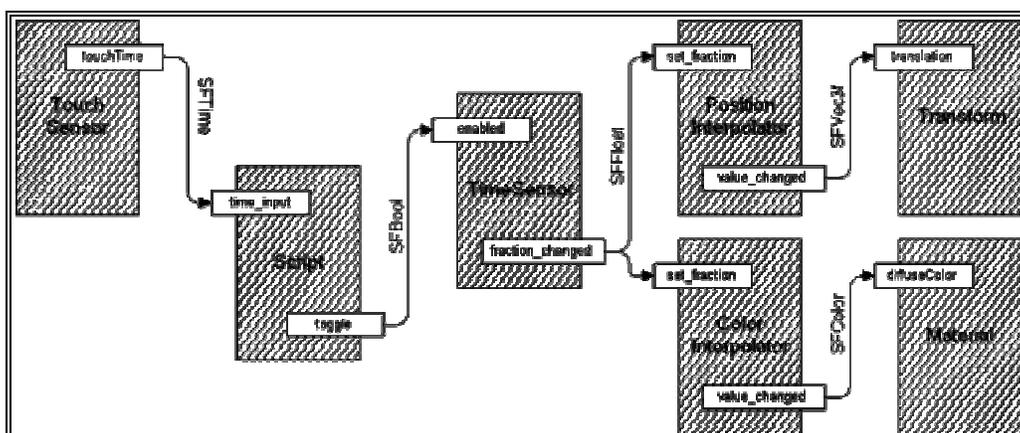


Figura 2.3 Un *Touchsensor* Envía un Solo Evento *Touchtime* (*Sftime*) a un Nodo *Script* Siempre que se Active. el Nodo *Script* Convierte el *Sftime* en un *Sfbool* y Manda el Resultado (True O False) al Campo *Enabled* del *Timesensor* que Comenzará Entonces (O Detendrá) su Marcha.

✦ **Incorporación del Java al VRML.**

El lenguaje JAVA, que es el enfoque más divulgado hasta los momentos, está basado en el empleo de pequeñas aplicaciones denominadas "applets" que interactúan gráficamente con las páginas Web. Estos applets son bajados ("downloaded") del servidor a la máquina del usuario y activados localmente. Su integración en los diversos "browsers" del VRML permitiría generar modelos activos que responderían a eventos y no simplemente a comandos predefinidos (ver Programa 2.1, Figura 2.4). Y ello a su vez abriría el camino a futuras aplicaciones incorporando Inteligencia Artificial.

```

1-  #VRML V2.0 utf8
2-  #Para cada instante de tiempo recibido como evento de entrada, el nodo devolverá un vector
3-  #con la posición de la pelota en ese instante
4-  DEF Ball Transform { . . . }
5-  DEF Clock TimeSensor { . . . }
6-  DEF Bouncer Script {
7-      field SFFloat bounceHeight 3.0
8-      eventIn SFFloat set_fraction
9-      eventOut SFVec3f value_changed
10-     url "javascript:
11-         function set_fraction( frac, tm ) {
12-             y = 4.0 * bounceHeight * frac * (1.0 - frac);
13-             value_changed[0] = 0.0;
14-             value_changed[1] = y;
15-             value_changed[2] = 0.0;
16-         }"
17-     }
18-  ROUTE Clock.fraction_changed TO Bouncer.set_fraction
20-  ROUTE Bouncer.value_changed TO Ball.set_translation

```

Programa 2.1 Fragmento de programa VRML que utiliza el nodo **Script** para el movimiento de una pelota al rebotar contra el suelo. También muestra como es el enrutamiento de un evento.

*Rebote1(a)**Rebote2(b)**Rebote3(c)*

Figura 2.4 (a), (b), (c) *Secuencia de Imágenes Recreando el Programa Anterior, Rebote de la Pelota.*

➤ **Reglas para la Definición de Campos y Eventos**

Las reglas para nombrar campos ("fields"), campos expuestos ("ExposedFields"), salidas de eventos ("eventOuts") y entradas de eventos ("EventsIns") para nodos construidos son las siguientes:

- Todos aquellos nombres que contienen múltiples palabras deben iniciarse con letra minúscula y la primera letra de cada palabra subsiguiente se escribirá con mayúscula (por ejemplo, bboxCenter), con la excepción hecha de get_ y _changed.
- Todos aquellos "eventIns" obedecieran al prefijo "set_", con la excepción de aquellos conocidos como "addChilden" y "removeChildren".
- A todos los eventos del tipo "eventOuts" se le añadirá el sufijo "_changed", con excepción de aquellos eventos del tipo SFBool. En ese caso los eventos boléanos deberán iniciarse con la palabra "is" para mayor legibilidad.
- Todos aquellos "eventIns" y "eventOuts" del tipo SFTime no deberán utilizar el preffijo"set_" o el sufijo "_changed".

2.3 LA REALIDAD VIRTUAL RV [1]

El diccionario define a la palabra virtual como “que existe o resulta en esencia o efecto pero no como forma, nombre o hecho real”, y a la palabra realidad como a “la cualidad o estado de ser real o verdadero”. Es fácil imaginar el campo de acción que tendrán con esta definición los filósofos y abogados de hoy día. En cualquier caso, tecnológicamente hablando, la RV ha sido definida de varias maneras específicas. A continuación se muestran algunas de estas definiciones:

- Es un entorno en tres dimensiones sintetizado por computadora en el que varios participantes acoplados de forma adecuada pueden atraer y manipular elementos físicos simulados en el entorno y, de alguna manera, relacionarse con las presentaciones de otras personas pasadas, presentes, ficticias o con criaturas inventadas.
- Es un sistema interactivo computarizado tan rápido e intuitivo que la computadora desaparece de la mente del usuario, dejando como real el entorno generado por la computadora, por lo que puede ser un mundo de animación en el que nos podemos adentrar.
- Es aquella forma de trabajo donde el hombre quiere interactuar totalmente con la computadora, generando ésta espacios virtuales donde el humano puede desempeñar sus labores y comunicarse con la computadora a través de efectores o dispositivos de interacción.

2.3.1 Diferenciación Entre Ciberespacio Y Realidad Virtual

Ciberespacio: Término creado por William Gibson, en 1984 publica su novela de ciencia-ficción "Neromancer", en la cual describía los espacios tridimensionales sintetizados por computadora. Comúnmente se utiliza este término en lugar del de Realidad Virtual, pero se distinguen uno del otro de la siguiente manera: la Realidad Virtual incluye experiencias, y el Ciberespacio sólo visualiza información y la accesa. A través de la historia se han hecho intentos de capturar la esencia de una experiencia y destilarla para así permitir la disponibilidad al público para disfrutar y analizar. Por medio de la experiencia directa del teatro, música y pintura, la gente ha sido capaz de percibir tanto las experiencias reales e imaginarias de otros mundos, otras épocas, nuevas ideas y nuevas perspectivas de antiguas ideas. Las computadoras y la RV son los primeros esquemas para cambiar la forma de cómo la gente observa, debate y relaciona el conocimiento.

2.3.2 Características Básicas De La Realidad Virtual

Se toman como características básicas de un sistema de RV las siguientes:

Interacción: Rasgos que permiten al usuario manipular el curso de la acción dentro de una aplicación de RV, permitiendo que el sistema responda a los estímulos de la persona que lo utiliza; creando interdependencia entre ellos. Existen dos aspectos únicos de interacción en un sistema de RV. El primero de ellos es la navegación, que es la habilidad del usuario para moverse independientemente alrededor del sistema. Las restricciones para este aspecto las coloca el inventor del software, que permite varios grados de libertad, si se puede volar o no, caminar,

nadar, etcétera. El otro punto importante de la navegación es el posicionamiento del punto de vistas del usuario. El usuario se puede mirar a sí mismo (a través de los ojos de alguien más), o puede moverse a través de cualquier aplicación observando 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 sistema interactúan con el usuario para intercambiar energía o información.

➤ **Inmersión:** Esta palabra significa bloquear toda distracción y enfocarse selectivamente sólo en la información u operación sobre la cual se trabaja. Posee dos atributos importantes, el primero de ellos es su habilidad para enfocar la atención del usuario, y el segundo es que convierte una base de datos en experiencias, estimulando de esta manera el sistema natural de aprendizaje humano (las experiencias personales).

➤ **Tridimensionalidad:** Esta es una característica básica para cualquier sistema llamado de Realidad Virtual, tiene que ver directamente con la manipulación de los sentidos del usuario, principalmente la visión, para dar forma al espacio virtual; los componentes del sistema se muestran al usuario en las tres dimensiones del mundo real, en el sentido del espacio que ocupan, y los sonidos tienen efectos estereofónicos (direccionalidad).

2.3.3 Formas de Realidad Virtual

Hay una variedad de diferentes concepciones y sistemas llamados entre ellos Realidad Virtual. Un sistema de Realidad Virtual debe tener las características básicas antes definidas, no es suficiente si se tienen una o dos de ellas, deben estar

las tres presentes. Tomando en cuenta las características básicas los tipos de RV existentes son:

2.3.3.1 Ventanas Acopladas Visualmente o Sistemas de Inmersión

Esta es la clase de sistema de inmersión que se asocia más a menudo con Realidad Virtual. Este sistema se basa en colocar las muestras directamente enfrente 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 (Head Mounted Display) 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 Realidad Virtual son los juegos que hicieron popular esta técnica de computación en los últimos años.

2.3.3.2 Sistemas Desktop De RV

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 3 dimensiones los exploradores pueden viajar en cualquiera dirección dentro de estos mundos, los ejemplos característicos de estos ambientes son los simuladores de vuelo para computadora, la mayoría de los juegos de alto nivel de realismo para computadora. En resumen, los sistemas desktop RV muestran mundos tridimensionales a través de pantallas de 2D. Algunos comprenderán interfaces sofisticadas, como guantes, controles, cabinas customizadas, pero todos tendrán en común la característica antes mencionada.

2.3.3.3 Segunda Persona

En este tipo de RV una imagen en movimiento del usuario se proyecta junto con otras imágenes en una extensa pantalla donde el usuario puede verse a sí mismo como si estuviese en la escena. En esencia los usuarios se miran ellos mismos como proyectados hacia ese mundo. También pueden pintar diseños de colores en el aire, o hacer cualquier movimiento que el sistema reacciona en tiempo real.

A diferencia de los sistemas de inmersión los sistemas en segunda persona (o unencumbered systems) involucran percepciones y respuestas en tiempo real a las acciones de los humanos involucrados, quienes están liberados o no están sometidos al uso de cascos, guantes, HMD's, alambres o cualquier otro tipo de interface intrusivo. Los sistemas de inmersión simulan las percepciones del mundo real, el viajero sabe que está ahí porque los sonidos e imágenes de esa realidad particular responden de manera similar a como responden los del mundo real a los movimientos de la cabeza. Sin embargo, en los sistemas en segunda persona, el explorador sabe que está dentro de ese mundo porque se ve a sí mismo dentro de la escena. Es decir es un integrante más de la escena. Para lograr esto el participante es ubicado frente a una pantalla de video, en la cual se proyecta la imagen misma del participante, pero con otra imagen utilizada como fondo o ambiente, entonces el participante visualiza en la pantalla el mundo tridimensional completo.

Mediante un software que realiza detección de contornos es posible realizar manipulaciones dentro de la escena, las cuales son visualizadas en la pantalla. Más que imitar las sensaciones del mundo real, un sistema de segunda persona cambia las reglas y aplica la vieja noción de "ver para creer" para inducir la sensación de presencia. Una aplicación es el juego virtual de golf, en el cual el jugador se ve a sí mismo sobre un campo de golf golpeando una pelota virtual. La aplicación más

famosa de esto es el popular programa de televisión "nick arcade", en el cual los niños participantes viven, juegan video juegos con ellos mismos como personajes del juego.

2.3.3.4 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 etc.), 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.

2.3.3.5 Realidad Aumentada

Esta se logra cuando una persona se confía del mundo real como línea de referencia, pero utiliza visores de cristal transparentes u otros medios inmersivos para aumentar la realidad, superponiendo esquemas, diagramas, textos, referencias, etc. Como ejemplo la Boeing está explorando la posibilidad de 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.

2.3.3.6 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. Es una experiencia psicológica que ocurre cuando la tecnología de simulación funciona lo suficientemente bien como para convencer al usuario de que está en otro mundo.

La telepresencia requiere que los estímulos adecuados provoquen las respuestas adecuadas en el usuario. Esto es en esencia, la proyección de una mente humana a un lugar remoto. La proyección de varias mentes a un sitio remoto puede ser efectuada si los participantes comparten el mismo espacio virtual. En lugares devastados donde ha habido un desalojo de personal, por ejemplo, se puede consultar a equipos localizados de expertos que quizás están lejos los unos de los otros. Juntos, ayudados por una vista común del lugar y la situación, son capaces de decidir la mejor manera de actuar antes de reunir gente y recursos para la tarea.

2.3.3.7 Realidad Virtual de Escritorio

Tablero de Realidad Virtual es una subinstalación del tradicional sistema de RV. En lugar de utilizar cascos para mostrar la información visual utiliza un monitor grande de computadora o un sistema de proyección. Algunos sistemas de este tipo permiten al usuario ver una imagen de tres dimensiones en sus monitores, pero utilizando lentes de cristal y pantalla de LCD o pantallas de cristal líquido.

2.3.4 Algunos Aspectos Importantes que hay que Tomar en Cuenta al Utilizar Rv

2.3.4.1 El Aspecto de la Inmersión y Percepción Visual

El diseño de presentaciones visuales para un sistema de RV es un asunto complicado debido a que el sistema visual humano es muy sensitivo a toda anomalía en la percepción de imágenes y escenas en movimiento. Además, dentro del campo de la percepción y focalización de los estímulos que recibe el ser humano a través de los sentidos, no podemos desligarnos de la interpretación que hacemos de aquella información captada del medio ambiente, así como de la supresión de la información que entendemos no es importante.

En resumen, el éxito en el uso y aplicación de la Realidad Virtual depende de programas tridimensionales con tiempo real de respuesta, así como escenarios que estimulen o motiven al usuario a utilizar el mismo. La percepción y campo visual considerados importantes para el sentido de inmersión reflejan la necesidad de percibir un ambiente que estimule su uso, proporcione seguridad y proteja su salud.

2.3.4.2 Consideraciones Básicas Para el Usuario: Implicaciones y Aplicaciones

Un aspecto importante en el desempeño del ser humano en un sistema RV es el efecto de las diferencias entre usuario, donde se han encontrado diferencias significativas. Las características del usuario influyen significativamente en la experiencia dentro de la Realidad Virtual, por tanto, deben identificarse para diseñar

sistemas de Realidad Virtual que se acomoden a las necesidades únicas del usuario. Para este propósito es necesario examinar estudios sobre "human-computer interacción" (HCI). En el HCI una de las principales características del usuario es la adaptación en el diseño de acuerdo al nivel de la experiencia. El nivel de experiencia influye en las destrezas del usuario, las habilidades en las cuales se predice su desempeño, y la manera en la cual el usuario entiende y organiza la tarea. Al examinar las influencias de las experiencias en HVEI ("Human-Virtual Environments Interaction/Interfase"), se puede predecir que la experiencia puede afectar las destrezas que el usuario tiene para interactuar con el sistema y la manera en la cual él mentalmente representa el sistema. La implicación refleja que se debe diseñar equipos externos para dichos sistemas apropiados al nivel de destreza y conocimiento de la población que utilizará el sistema, lo que es un reto para los desarrolladores de estos equipos.

Las aptitudes técnicas (visualización espacial, orientación memoria espacial, "spatial scanning", por ejemplo) son generalmente significativas en predecir el desempeño en la HCI. En efecto, individuos con baja puntuación en pruebas de memoria espacial generalmente presentan mayor tiempo para el desempeño de tareas, más tiempo para realizar el trabajo y producen una mayor cantidad de intentos y error. Las dificultades experimentadas en sujetos con bajo nivel espacial están particularmente relacionadas a problemas en el sistema de navegación donde los usuarios frecuentemente reportan que "están perdidos" con los sistemas jerárquicos del menú. Estos hallazgos son particularmente relevantes para los sistemas virtuales que demandan grandes destrezas de navegación. De hecho, usuarios experimentados o conocedores, también señalan sentirse "perdidos" en dichos sistemas.

En la búsqueda de accesibilidad para cualquier producto o sistema, uno de las aproximaciones más básicas en el diseño (de productos/sistemas) es el rango de habilidades y necesidades de adaptación o de tecnología asistida. Esta aproximación

es particularmente relevante para el futuro desarrollo de los sistemas de Realidad Virtual, y los mismos presentan dos principios fundamentales. Primero, la Realidad Virtual debe ser usada como una metáfora que permita fácilmente proveer alternativas físicas o metáforas sensoriales. Segundo, la naturaleza de la Realidad Virtual y la habilidad para alterar la realidad construida debe aportar nuevas oportunidades para proveer acceso al mundo. Básicamente, la Realidad Virtual provee oportunidades para recrear un mundo o un ambiente que acerque el vínculo de las tareas y las habilidades del usuario. También provee varias alternativas de control (equipos externos) y de presentación. Por ejemplo, se pueden tomar varias formas para representar el nivel de volumen, tal como una barra, marcas, impresión (printed), o un número hablado (o sintetizado por computadora); pero no se tienen las mismas opciones al presentar la sonrisa de la Mona Lisa.

2.4 AMBIENTES VIRTUALES AV

Los AV como su nombre lo indica son lugares, sitios, ambientes no físico sino abstractos. Realizados aplicando toda la tecnología de la RV, que pueden ser basados en la realidad, si el ambiente existe físicamente como lo es la Universidad de Oriente Núcleo de Anzoátegui, o artificiales como los juegos de video. A continuación se muestran algunos conceptos más formales sobre AV:

➤ Los AV o Mundos Virtuales MV pueden representar cualquier mundo tridimensional que puede ser real o abstracto. Esto incluye sistemas reales como edificios, aeronaves, sitios de excavación, anatomía humana, reconstrucción de crímenes, sistemas solares, y muchas más. De sistemas abstractos se puede incluir

campos magnéticos, modelos moleculares, sistemas matemáticos, acústica de auditorios, densidad de población y muchos más.

➤ Un MV es una base de datos gráficos interactivos, explorable y visualizable en tiempo real en forma de imágenes tridimensionales de síntesis capaces de provocar una sensación de inmersión de la imagen. En sus formas más complejas, el entorno virtual es un verdadero "espacio de síntesis", en el que uno tiene la sensación de moverse "físicamente".

Los MV son nuevos "Laberintos". Nos enfrentan a experiencias nuevas del espacio y del cuerpo y a paradojas de un género nuevo. Pero sobre todo debido al desarrollo técnico de lo virtual y de lo sintético nos obligan a aprender, a ver de otra forma las imágenes para comprenderlas mejor, y también captar los modelos debajo de las apariencias para ver de verdad. Mundos que enlazan las realidades y las apariencias, las imágenes y los modelos. Gracias a los MV, descubrimos horizontes nuevos y sutiles entre los ámbitos de la concepción y de la percepción. Así, se abre un gran campo de investigación a través de una fenomenología de lo virtual.

2.4.1 Componentes de los Ambientes Virtuales

Consta básicamente de tres componentes que son:

1. Un sistema computador que genera los AV.
2. Uno o más operadores humanos.
3. Una interfaz que permita la interacción de los usuarios con el AV.

2.4.2 Tipos De Mundos Virtuales

Se puede decir que existen tres tipos básicos de MV 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.

2.5 MÉTODOS DE TRABAJO EN REALIDAD VIRTUAL.

Existen propuestas de diferentes autores en lo que respecta el desarrollo de aplicaciones en realidad virtual. Aquí describiremos un Enfoque Planteado por GONZALO VELEZ. Este investigador venezolano plantea una secuencia de pasos a seguir. Describir la aplicación por desarrollar en aspectos tales como: objetos, escenarios, reglas y tramas, de manera muy esquematizada en un anteproyecto, permite:

- Identificar interacciones e interfaces.
- Especificar hardware y software.
- Crear y/o importar objetos y escenarios.
- Definir atributos.
- Establecer sus posiciones.
- Incorporar comportamiento interactivo.
- Añadir animación, luces y cámaras.
- Operar y ajustar experimentalmente e incorporarlo al producto acabado.

La propuesta de este autor se basa en un ciclo que se repite hasta que se logra un producto deseado. Por otra parte, según Vélez, existen dos visiones para desarrollar un sistema de RV, que son:

- **Top-down:** Donde lo principal es el problema. Se estructura el sistema sobre la base de preguntas básicas, estableciendo los atributos de los objetos, el comportamiento, la iluminación y otros atributos. Establecer las propiedades del espacio virtual que contendrá los objetos para luego ir desarrollando en detalle las

especificaciones, es decir, mirar la aplicación desde arriba, en forma general para luego ir a los niveles de detalle.

➤ **Bottom-up:** Corresponde a definir primeramente los detalles de la aplicación hasta llegar al sistema global. Aquí se toman en cuenta primeramente los sistemas coordinados y los objetos con sus características de rotación, traslación y escalamiento, las especificaciones de color, sombra y textura para cada objeto y finalmente se tratan los límites, la iluminación y las características del ambiente simulado.

Al desarrollar una aplicación, se proponen las siguientes interrogantes a resolver: ¿Cuál es el uso que se le dará al mundo virtual? ¿Cuál es el propósito? El mundo virtual ¿admite cambios? ¿Cuántos escenarios diferentes se van a incorporar? ¿Cómo se van a utilizar los escenarios y sus elementos? ¿Qué cualidades deben tener los objetos? ¿Cuál es el grado de interacción con los elementos que pueblan el mundo? ¿Existen leyes que controlan el mundo, particularmente la gravedad?

Agrega que existen ciertos factores que dan el grado de interés que presenta un mundo virtual ante un “cibernauta” como son: los eventos, el realismo, la animación y la capacidad de respuesta. En un mundo virtual, en cierto modo, uno es un pequeño Dios, creador de un pequeño mundo en el cual las personas y las cosas que están allí, uno las hizo. Según Vélez, es necesario considerar ciertas recomendaciones, para la creación de ellos. Estas son:

➤ El mundo debe ser orientado con criterio de baja resolución. Nunca debe pensarse en un mundo que se base en una resolución muy alta, porque es costoso en tiempo y memoria. Se debe pensar en comenzar en forma sencilla y luego sofisticar.

- Los objetos deben ser simples. No tiene sentido hacer objetos complicados, porque esto es contradictorio con la fluidez del mundo. Se pueden hacer cosas muy creativas sin la necesidad de ser complejo. Se debe tener la capacidad de ser sintético describiendo objetos, esto es un arte.
- Discriminar niveles de detalles, porque se pueden tener cosas de cerca y distintas de lejos.
- Pocas superficies curvas, ya que estas consumen enormes cantidades de memoria y además porque se ve mucho el efecto escalera que producen las curvas.
- Texturizado selectivo, porque la textura, siendo muy útil, también consume demasiada memoria. Existen texturas dinámicas, como por ejemplo para representar agua o lluvia. Las posibilidades son variadas y buenas, pero hay que saberlas utilizar.
- Colores simples, imitando dentro de lo posible a la naturaleza. Los colores muy especiales no tienen importancia en el desarrollo del mundo.
- Iluminación variada, es decir, si pongo una luz fija en un sitio va a ser fastidioso, monótono y aburrido. Se debe jugar con las luces para provocar la sensación de variación.
- Sonidos armónicos. Deben armonizar con el propósito de la aplicación. Por ejemplo, si existe un reloj de muralla se debe colocar un sonido de reloj y no otro.
- Música suave, no muy compleja, porque quita tiempo y recursos.
- Los sonidos y voces deben ser buenos, ya que el sonido consume enormes cantidades de memoria.

- Incorporar movimientos, porque estos generan interés.
- Incorporar seres inteligentes, porque los seres que habitan esos mundos no deben ser predecibles.
- Imitar, en lo posible, a la realidad, porque el ser humano establece una contraparte con lo que conoce, con su experiencia vivida.
- Premiar al usuario, esto incentiva su interés.
- Mantener el propósito inicial.

En la práctica, la creación de un mundo virtual dependerá de la naturaleza del problema y de la experiencia del diseñador, es así que Vélez considera una secuencia de etapas que pueda ser aplicable a diferentes situaciones. Dicha secuencia es:

➤ **Estudio de factibilidad. Aquí se analiza:**

- Alcance y objetivos del mundo virtual.
- Requerimientos del mundo virtual.
- Herramientas requeridas.
- Personal, tiempo, costos.

➤ **Diseño del mundo virtual.**

Elaboración de especificaciones y de criterios de evaluación detallados respecto del mundo virtual. Deberá definirse:

- El mundo virtual está únicamente comprometido por las respuestas que debe dar a las acciones desarrolladas por el usuario.

- El mundo virtual posee una línea de acción propia (“guión”), pero única que culmina en un determinado desenlace.
- Que el guión del mundo virtual ofrece múltiples líneas de acción alternativas, las cuales puede navegar el usuario. Los factores mencionados se desarrollan “fuera del computador” y son preparativos por las actividades que involucran el contacto con el computador.

➤ **Construcción del mundo virtual.**

Desarrollo del modelo del mundo virtual en función de las especificaciones elaboradas en la fase anterior. Debido a la necesidad de contar con mejores métodos de comunicación, se vienen desarrollando recursos tales como “manos virtuales” y comandos de voz que faciliten la acción del usuario.

➤ **Prueba y control de calidad.**

Verificación de que el mundo virtual desarrollado esté conforme con los estándares e incorporar todas las especificaciones planteadas.

➤ **Ensayo piloto.**

Comprobación del comportamiento del mundo virtual a través de usuarios “de confianza”, en un ámbito limitado. Correcciones y ajustes.

➤ **Distribución.**

Hacer accesible el mundo virtual desarrollado a todos aquellos usuarios que deban tenerlo.

➤ **Evaluación y ajustes.**

Transcurrido un tiempo prudencial de uso, recoger todas las observaciones formuladas e incorporar correcciones y ajustes requeridos para una nueva versión, determinados aspectos tales como si las metas trazadas originalmente fueron

realmente alcanzadas y si el proceso de construcción del mundo pudo haber sido realizado de manera más eficiente.

2.6 SISTEMAS DE BASE DE DATOS.

2.6.1 Hablemos de sistemas.

Conjunto de entidades caracterizadas por ciertos atributos, que tienen relaciones entre sí y están localizadas en un cierto ambiente, de acuerdo con un cierto objetivo.

2.6.2 Base de datos.

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos.

2.6.3 Componentes principales de una Base de Datos

➤ **Datos**

Los datos son la Base de Datos propiamente dicha.

➤ **Hardware**

El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.

➤ **Software**

Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.

➤ **Usuarios**

Existen tres clases de usuarios relacionados con una Base de Datos:

- El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
- El usuario final, quien accesa la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
- El administrador de la Base de Datos (DBA: Data Base Administrator), quien se encarga del control general del Sistema de Base de Datos.

2.6.4 Modelos de Base de Datos

➤ Bases de datos Jerárquicas

Estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres se le conoce como raíz, y a los nodos que no tienen hijos se les conoce como hojas. Una de las principales limitaciones de este modelo, es su incapacidad de representar eficientemente la redundancia de datos.

➤ Bases de datos de red

Este es un modelo ligeramente distinto del jerárquico, en donde su diferencia fundamental es la modificación del concepto de un nodo, permitiendo que un mismo nodo tenga varios padres (algo no permitido en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos.

➤ Bases de datos relacionales

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. Este modelo pasa por un proceso al que se le conoce como *Normalización de una base de datos*, se usan un grupo de tablas para representar los datos y las relaciones entre ellos.

➤ **El modelo Entidad – Relación**

Este modelo está basado en una percepción del mundo real, que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es todo aquello que exhibe autonomía, diferenciación y existencia en el mundo real que es distinguible de otros objetos. Por ejemplo cada persona es una entidad y las cuentas bancarias pueden ser consideradas entidades.

➤ **Bases de datos Orientadas a Objetos (BBOO)**

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Las BDOO permiten el desarrollo y mantenimiento de aplicaciones complejas ya que se puede utilizar un mismo modelo conceptual y así aplicado al análisis, diseño y programación. Es una base de datos que incorpora todos los conceptos importantes de la programación orientada a objetos:

- *Encapsulación* Ocultar datos del resto de los datos, impidiendo así accesos incorrectos o conflictos.
- *Herencia* Reusabilidad del código.
- *Polimorfismo* Sobrecarga de operadores o de métodos.

2.6.5 Sistemas Administrador de Base de Datos (DBMS)

El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un lenguaje de manipulación de datos (DML: Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language).

2.6.6 Objetivo de un Sistema Administrador de Base de Datos

Dado que los sistemas administradores de base de datos, además de tener como objetivo fundamental almacenar e independizar las aplicaciones de la estructura física de los datos, buscan facilitar y simplificar las tareas de desarrollo y mantenimiento de sistemas de información, puede afirmarse que sus objetivos generales son:

➔ **Disminuir la redundancia de datos**

En los sistemas tradicionales, dado que cada aplicación tiene sus propios archivos privados, es común que un mismo dato se encuentre repetido en diferentes archivos. En un ambiente de base de datos, las redundancias de este tipo pueden ser minimizadas, con las consecuentes ventajas de simplicidad y flexibilidad en el uso y búsqueda de la información.

➔ **Eliminar la inconsistencia entre datos redundantes**

Si algunas veces no es posible evitar la redundancia, la tecnología de base de datos brinda facilidades que pueden ser utilizadas por las aplicaciones con el fin de evitar que los datos redundantes sean inconsistentes de tal forma que si, un dato es redundante en dos o más tipos de registros, sus valores coincidan.

➔ **Compartir datos entre múltiples usuarios**

En un ambiente de base de datos ofrece facilidades que permiten que la información sea compartida por diversos usuarios y aplicaciones, aún cuando existan requerimientos diferentes o conflictivos.

➔ **Establecer estándares y procedimientos**

Un sistema de manejo o gestión de base de datos permite establecer estándares y normas para el uso de los datos y brinda facilidades para que puedan ponerse en

práctica procedimientos generales que regulen el acceso, el uso y la representación de los datos.

➔ **Establecer normas de seguridad**

Un manejador de bases de datos permite aplicar restricciones de seguridad para tener acceso a la información, con el fin de proteger la confidencialidad de la misma.

➔ **Proteger la integridad de los datos**

De manera análoga, un manejador de bases de datos ofrece facilidades para proteger la integridad de los datos, evitando que puedan ser destruidos accidental o intencionalmente, y evitando que puedan ser alterados por personas no autorizadas.

➔ **Independizar los programas de las estructuras de datos**

La tecnología de base de datos busca independizar las aplicaciones de la estrategia de acceso a los datos, de tal forma que pueda modificarse una base de datos sin que ello requiera modificar los programas de aplicación que utilizan esa base de datos.

2.6.7 Base de Datos Multimedia en la WWW

Las bases de datos multimedia han empezado a enfrentarse a dos de los problemas más espinosos que se les presentaban a los desarrolladores Web: controlar la proliferación de información digital y generar páginas Web complejas e interactivas. Ahora cuando las grandes compañías están almacenando y publicando masivamente información multimedia en CD-ROM, intranet y web, los productores

de bases de datos tratan de aumentar las capacidades del modelo relacional de sus respectivos servidores para mejorar el tratamiento de tipos de datos complejos.

Todo esto debido a los *Rdbms* que al enfrentarse al proceso de imágenes, vídeo, sonido o datos espaciales, esto se atascan. En este sentido hay cuatro tipos de bases de datos compitiendo por la atención de los Webmasters y los productores multimedia:

- Los tradicionales modelos relacionales (*Rdbms*) de *Oracle*, *IBM*, etc.;
- Las Bases de datos orientadas a objetos (*Odbms*);
- Los híbridos o Bases de datos relacionales orientados a objetos (*Ordbms*), como *Illustra*, del que se hablará posteriormente, y un número creciente de bases de datos desarrolladas específicamente (*Cinebase* por ejemplo).

2.6.8 La Base De Datos En Los Ambientes Virtuales

Contiene la descripción del mundo, como los objetos que se encuentran dentro de él, los guiones que describen las acciones de los objetos, variables de ambiente como iluminación, controles y dispositivos de hardware

2.6.8.1 Métodos de Almacenamiento

La descripción del mundo puede ser almacenada de varias formas, ya sea en un sólo archivo, un conjunto de ellos o una base de datos. Lo más común es utilizar varios archivos, así, la información de cada objeto (forma, guiones, etc.) estará

contenida en uno o más. Hay sistemas que cuentan con un archivo de configuración que determina las conexiones requeridas con la interfaz de hardware.

Algunos sistemas cargan desde el inicio la base de datos completa, otros van cargando la información que necesitan en cada momento. Algo que puede contribuir al aprovechamiento es el uso de un sistema real de bases de datos. Una base de datos orientada a objetos podría ser una gran ayuda para un sistema de MV, aunque la mayoría de los proyectos no los usan, puesto que para ellos no es importante la administración de su propia base de datos.

La práctica más común es almacenar los datos como archivos ASCII principalmente o archivos binarios. También hay sistemas que incluyen ésta información en el archivo de la aplicación misma.

2.7 EL PARADIGMA DE LA PROGRAMACION ORIENTADA A OBJETO.

La Programación Orientada a Objetos (OOP por sus siglas en inglés de *Object Oriented Programming*) como paradigma, "es una forma de pensar, una filosofía, de la cual surge una cultura nueva que incorpora técnicas y metodologías diferentes. Pero estas técnicas y metodologías, y la cultura misma, provienen del paradigma, no lo hacen. La OOP como paradigma es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose por sí mismo, y comunicándose con los demás por medio de mensajes" [3].

Se debe distinguir que la OOP como paradigma (enfoque o manera de visualizar la realidad) y como metodología (colección de características para la

ingeniería de software) no es la misma cosa. Sin embargo, la publicidad nos confunde asociando la OOP más a una metodología, que al paradigma. De aquí que "el interés en la OOP radica más en los mecanismos que aporta para la construcción de programas que en aprovechar un esquema alterno para el modelado de procesos computacionales" [3].

La Programación Orientada a Objetos desde el punto de vista computacional "es un método de implementación en el cuál los programas son organizados como grupos cooperativos de objetos, cada uno de los cuales representa una instancia de alguna clase, y estas clases, todas son miembros de una jerarquía de clases unidas vía relaciones de herencia" [3].

2.7.1 El Enfoque Orientado a Objeto.

El enfoque de orientación por objetos (O.O.) es un paradigma que también cubre el ciclo de vida del software y que permite tener un mayor acercamiento al mundo que se modela y cómo funciona este mundo. En algunas metodologías de Ingeniería de Software (IS) se habla del análisis, diseño y desarrollo como tres procesos independientes cuya mezcla tiene como resultado final una aplicación que satisface X o Y necesidades.

El problema que presenta esta disyunción está dado por el modo como se trabaja normalmente en cada uno de estos procesos. El lenguaje que manejan, los alcances y el resultado final de cada uno de ellos pueden afectar el resultado final global. Al tratarlos como entes independientes, los mecanismos para acomodar y traducir la información producida por cada proceso para que pueda ser

"efectivamente usada" genera potenciales fallos a interpretar de determinada manera la información allí contenida. [1]

El enfoque O.O. busca resarcir las deficiencias que se presentan en cada una de las etapas del ciclo de vida de la IS convencional, permitiendo obtener una mejor representación del mundo y de los requerimientos particulares de una aplicación en dicho mundo. Este enfoque puede ser aplicado indistintamente al análisis, diseño o desarrollo de una aplicación. No es estrictamente necesario usar el enfoque en todas las etapas del ciclo de vida de una aplicación. Si se desea, se puede elaborar un buen análisis y diseño O.O., aún cuando la implementación no necesariamente siga el mismo esquema. Sin embargo, es una excelente alternativa usar O.O. en todo el ciclo de vida, buscando aprovechar al máximo todas las bondades de este nuevo paradigma. [1]

En el enfoque OO las propiedades del objeto son claves. Los principios del modelo **OO** son: abstracción, encapsulación, modularidad y jerarquía, fundamentalmente, y en menor grado tipificación (typing), concurrencia, persistencia. [18] dice que si un modelo que se dice OO no contiene alguno de los primeros cuatro elementos, entonces no es OO.

Durante la definición de objetos del mundo se pueden usar los mecanismos de herencia y polimorfismo, para aprovechar las características y comportamiento de algunos objetos básicos, extendiéndolos para conseguir objetos con un comportamiento más específico. Además se puede usar otra importante característica llamada "reutilización de código", definiendo objetos que pueden ser usados en futuros desarrollos.

Por último, el hecho de modelar el mundo y no únicamente los datos necesarios para determinada aplicación, permiten crear diversas aplicaciones sobre la misma

información sin repetir los procesos de análisis de los mismos. Esto ofrece la posibilidad de dedicarse a cumplir con los requerimientos de la aplicación basándose en las facilidades que ofrecen los objetos del mundo ya modelados.

2.7.2 Ventajas De Usar El Enfoque O.O. [1]

Las ventajas de usar el enfoque O.O. se traducen en mejoramientos de calidad a lo largo del ciclo de vida de una aplicación, facilitando además el mantenimiento y la creación de nuevas versiones que extiendan el programa.

Al disminuir las barreras entre las etapas de análisis, diseño y desarrollo, se garantiza que se está hablando de las mismas cosas y en los mismos términos desde el comienzo del análisis hasta el final de la etapa de implementación. Esto evita inconsistencias y permite verificar que las cosas están claramente definidas y cumplen con todos los requerimientos, incluso antes de escribir una línea de código del programa. Las características anteriormente mencionadas (encapsulamiento, herencia, reutilización) permiten crear un software mucho más robusto.

Por último, el hecho de modelar el mundo y no únicamente los datos necesarios para determinada aplicación, permiten crear diversas aplicaciones sobre la misma información sin repetir los procesos de análisis de los mismos. Esto ofrece la posibilidad de dedicarse a cumplir con los requerimientos de la aplicación basándose en las facilidades que ofrecen los objetos del mundo ya modelados.

Se pueden enunciar varios beneficios de la aproximación orientada por objetos:

➔ **Reutilización de Software:** permite describir clases y objetos que podrán ser usados en otras aplicaciones.

- **Estabilidad:** el diseñador piensa en términos de comportamiento de objetos, no en detalles de bajo nivel; diseño rápido y de alta calidad, puesto que se concentra en satisfacer los requerimientos y no en detalles técnicos.
- **Integridad:** facilidad de programación al usar efectivamente toda la información de la fase de diseño, poniéndola en términos de un lenguaje específico.
- **Facilidad de Mantenimiento:** dado que al tener el modelo del mundo, es fácil realizar mantenimiento en términos de objetos, atributos y métodos de los mismos.
- **Independencia en el Diseño:** el diseño de un software se puede hacer independientemente de plataformas, software y hardware.

2.7.3 Proceso Unificado.

Es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología orientada a objetos en el desarrollo de software de misión crítica en una variedad de industrias por la compañía Rational, donde confluyen “los tres amigos” como se llaman a sí mismos o los tres grandes OO: Grady Booch, James Rumbaugh e Ivar Jacobson. [3].

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el

sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

2.7.4 Ciclo de Vida

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez, se divide en cuatro fases, como se muestra en la figura 2.5. A través de una secuencia de modelos, los implicados visualizan lo que está sucediendo en esas fases. Dentro de cada fase, los directores o los desarrolladores pueden descomponer adicionalmente el trabajo en iteraciones con sus incrementos resultantes. La columna izquierda representa los flujos de trabajos: requisitos, análisis, diseño, implementación y prueba. Las curvas son una aproximación de hasta dónde se llevan a cabo los flujos de trabajo en cada fases.

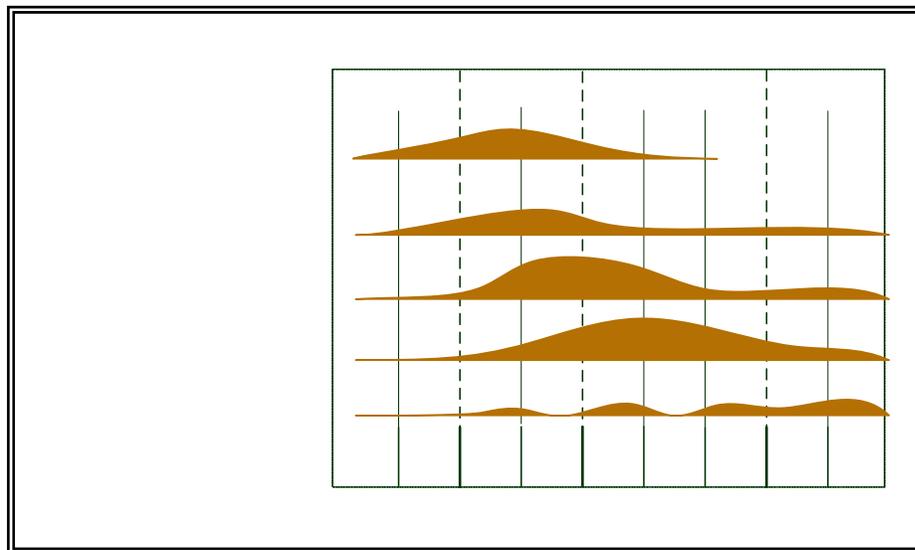


Figura 2.5 Estructura del Proceso Unificado

2.7.5 Fases. [3]

➤ **Inicio.** Se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. Esencialmente, esta fase responde a las siguientes preguntas: ¿Cuáles son las principales funciones del sistema?, ¿Cómo podría ser la arquitectura del sistema?, ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

La Respuesta a la primera pregunta se encuentra en un modelo de casos de uso simplificado que contenga los casos de uso más críticos. Cuando se tiene, la arquitectura es provisional, y consiste típicamente en un simple esbozo que muestra los subsistemas más importantes. En esta fase se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

➤ **Fase de elaboración.** Se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. La relación entre la arquitectura del sistema y el propio sistema. Por tanto, la arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan su totalidad. Esto implica que hay vistas arquitectónicas del modelo de caso de uso, de análisis, diseño, implementación y despliegue. Durante esta fase del desarrollo, se realizan los casos de uso más críticos que se identificaron en la fase de inicio.

➤ **Fase de construcción.** En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. Al final de esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han

acordado. La pregunta decisiva es: ¿Cubre el producto las necesidades de los usuarios de manera suficiente como para hacer una primera entrega?

➤ **Fase de Transición:** Cubre el periodo durante el cual el producto se convierte en versión beta. En la versión beta un número de usuarios con experiencia prueban el producto e informan de defectos y deficiencias. Los desarrolladores corrigen el problema e incorporan algunas de las mejores sugerencias en una versión general dirigida a la totalidad de la comunidad de usuarios. La fase de transición conlleva actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y corrección de los defectos que se encuentran en la entrega.

2.7.6 Lenguaje De Modelado Unificado (UML)

El Lenguaje de Modelado Unificado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML es una especificación de notación orientada a objetos. Se basa en las anteriores especificaciones Booch, Rumbaugh y Coad-Yourdon. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

La explicación se basará en los diagramas, en lugar de en vistas o anotación, ya que son estos la esencia de UML. Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas. Las vistas existentes en UML son:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- Vista de diseño: Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- Vista de procesos: Se forma con los diagramas de la vista de diseño. Recalcando las clases y objetos referentes a procesos.
- Vista de implementación: Se forma con los diagramas de componentes, colaboración, estados y actividades.
- Vista de despliegue: Se forma con los diagramas de despliegue, interacción, estados y actividades.
- Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.
- Los diagramas estáticos son:
 - Diagrama de clases.
 - Diagrama de objetos.
 - Diagrama de componentes.
 - Diagrama de despliegue.
 - Diagrama de casos de uso.
- Los diagramas dinámicos son:
 - Diagramas de Iteración o también llamados; Diagrama de secuencia y Diagrama de colaboración.

- Diagrama de estados.
- Diagrama de actividades.

2.7.6.1 Diagrama De Clases

Forma parte de la vista estática del sistema. En el diagrama de clases es donde definiremos las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización. Es decir, es donde daremos rienda suelta a nuestros conocimientos de diseño orientado a objetos, definiendo las clases e implementando las ya típicas relaciones de herencia y agregación. Un diagrama de clase está compuesto por los siguientes elementos:

➤ **La Clase**

Una clase está representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos.

- Cada clase debe tener un nombre único, que las diferencie de las otras.
- Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Los atributos pueden representarse solo mostrando su nombre y su tipo, e incluso su valor por defecto.
- Un método u operación es la implementación de un servicio de la clase, que muestra un comportamiento común a todos los objetos. En resumen es una función que le indica a las instancias de la clase que hagan algo.

➤ **Relaciones entre clases**

Existen tres relaciones diferentes entre clases, Dependencias, Generalización y Asociación. En las relaciones se habla de una clase destino y de una clase origen. El origen es desde la que se realiza la acción de relacionar. Es decir desde donde parte la flecha, hasta el destino o quien recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones.

✚ **Dependencias**

Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada. Con la dependencia mostramos que un cambio en la clase utilizada puede afectar al funcionamiento de la clase utilizadora, pero no al contrario. Aunque las dependencias se pueden crear tal cual, es decir sin ningún estereotipo (palabra al lado de la línea que representa la dependencia) UML permite dar más significado a las dependencias, es decir concretar más, mediante el uso de estereotipos. Estereotipos de relación Clase-objeto.

- ✓ *Bind*: La clase utilizada es una plantilla, y necesita de parámetros para ser utilizada, con Bind se indica que la clase se instancia con los parámetros pasándole datos reales para sus parámetros.
- ✓ *Derive*: Se utiliza al indicar relaciones entre dos atributos, indica que el valor de un atributo depende directamente del valor de otro. Es decir el atributo edad depende directamente del atributo Fecha nacimiento.
- ✓ *Friend*: Especifica una visibilidad especial sobre la clase relacionada. Es decir podrá ver las interioridades de la clase destino.
- ✓ *InstanceOF*: Indica que el objeto origen es una instancia del destino.
- ✓ *Instantiate*: indica que el origen crea instancias del destino.

- ✓ *PowerType*: indica que el destino es un contenedor de objetos del origen, o de sus hijos.
- ✓ *Refine*: se utiliza para indicar que una clase es la misma que otra, pero más refinada, es decir dos vistas de la misma clase, la destino con mayor detalle.

✚ Generalización

Pues es la herencia, donde tenemos una o varias clases padre o superclase o madre, y una clase hija o subclase. UML soporta tanto herencia simple como herencia múltiple. Aunque la representación común es suficiente en el 99.73% de los casos UML nos permite modificar la relación de Generalización con un estereotipo y dos restricciones.

- ✓ Estereotipo de generalización.
 - Implementation*: El hijo hereda la implementación del padre, sin publicar ni soportar sus interfaces.
- ✓ Restricciones de generalización.
 - Complete*: La generalización ya no permite más hijos.
 - Incomplete*: Podemos incorporar más hijos a la generalización.
 - Disjoint*: solo puede tener un tipo en tiempo de ejecución, una instancia del padre solo podrá ser de un tipo de hijo.
 - Overlapping*: puede cambiar de tipo durante su vida, una instancia del padre puede ir cambiando de tipo entre los de sus hijos.

■ Asociación

Especifica que los objetos de una clase están relacionados con los elementos de otra clase. Se representa mediante una línea continua, que une las dos clases. Podemos indicar el nombre, multiplicidad en los extremos, su rol, y agregación. En la figura 2.6 se han creado cuatro clases. La clase principal es Usuario, que tiene dos clases hijas UsuarioADM y UsuarioINF.

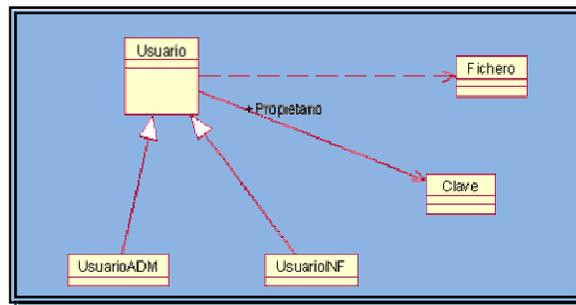


Figura 2.6 Representación de un Diagrama de Clase.

El usuario mantiene una relación de asociación con la clase Clave, se indica que es propietario de una clave, o de un número indeterminado de ellas. Se le crea también una relación de dependencia con la clase Perfil, es decir las instancias de usuario contendrán como miembro una instancia de Perfil.

En la figura 2.7 están todos los elementos que pueden ser usados para representar un diagrama de clase.

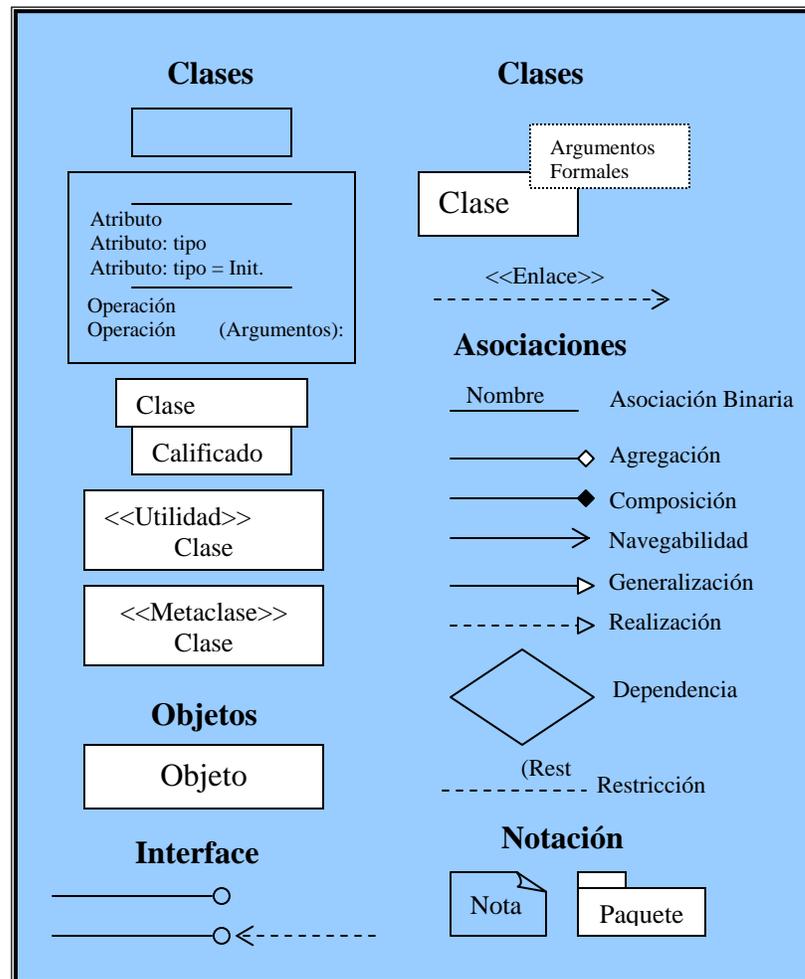


Figura 2.7 Elementos que Forman Parte del Diagramas de Clase

2.7.6.2 Diagramas de Objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Muestra a los objetos y sus relaciones, pero en un momento concreto del sistema. Estos diagramas contienen objetos y

enlaces. En los diagramas de objetos también se pueden incorporar clases, para mostrar la clase de la que es un objeto representado.

En este diagrama se muestra un estado del diagrama de eventos. Para realizar el diagrama de objetos primero se debe decidir qué situación queremos representar del sistema. Es decir si disponemos de un sistema de mensajería, deberemos decidir que representaremos el sistema con dos mensajes entrantes, los dos para diferentes departamentos, dejando un departamento inactivo. Para la siguiente figura 2.8;

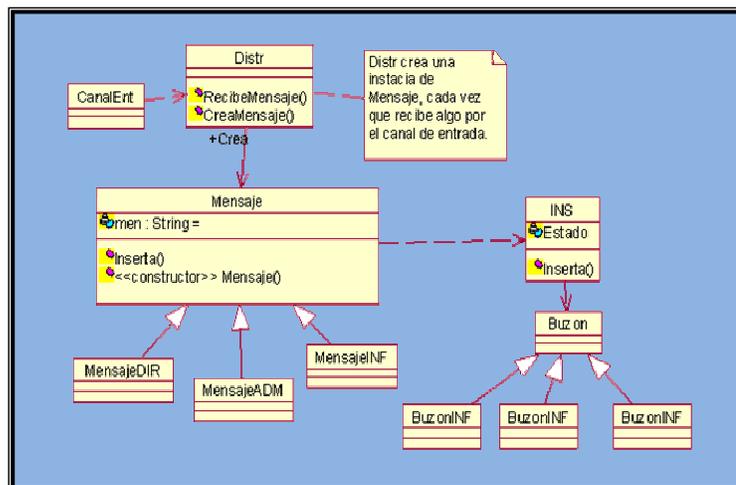


Figura 2.8 Representación de un Diagrama de Objetos.

Tendríamos un diagrama de objetos con dos instancias de `Mensaje`, más concretamente con una instancia de `MensajeDIR` y otra de `MensajeADM`, con todos sus atributos valorados. También tendríamos una instancia de cada una de las otras clases que deban tener instancia. Como `CanalEnt`, `INS`, `Distr`, y el `Buzon` correspondiente a la instancia de mensaje que se esté instanciando. En la instancia de la clase `INS` se deberá mostrar en su miembro `Estado`, que está ocupado realizando una inserción. En un diseño no podemos encontrar con multitud de diagramas de objetos, cada uno de ellos representando diferentes estados del sistema.

2.7.6.3 Diagrama de Componentes

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. Entre las cuales tenemos:

- Ejecutables: Nos facilita la distribución de ejecutables a los clientes. Documenta sus necesidades y dependencias. Si disponemos de un ejecutable que solo se necesita al mismo para funcionar no necesitaremos el diagrama de componentes.
- Código Fuentes: Se utiliza para documentar las dependencias de los diferentes ficheros de código fuente. Un ejecutable, o librería es una combinación de estos ficheros, y al mostrar la dependencia entre ellos obtenemos una visión de las partes necesarias para la creación del ejecutable o librería.

En la figura 2.9 (a) tenemos un componente del sistema de Windows. En el diagrama de componentes de Windows debe salir este componente, ya que sin el sistema no funcionaría y en esta otra figura 2.9 (b) tenemos el mismo componente, pero indicamos que dispone de un interface. Al ser una Dll el interface nos da acceso a su contenido. Esto nos hace pensar que la representación anterior es incorrecta, pero no es así solo corresponde a un nivel diferente de detalle.

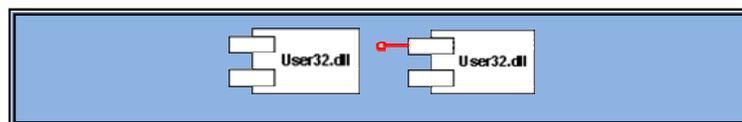


Figura 2.9 Representación de un Diagrama de Componente de Windows.

2.7.6.4 Diagrama de Casos De Uso

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase. De forma que se pueda conocer cómo responde esa parte del sistema. El diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema, ya que solo especifica cómo deben comportarse y no como están implementadas las partes que define. Por ello es un buen sistema de documentar partes del código que deban ser reutilizables por otros desarrolladores. El diagrama también puede ser utilizado para que los expertos de dominio se comuniquen con los informáticos sin llegar a niveles de complejidad. Un caso de uso especifica un requerimiento funcional, es decir indica esta parte debe hacer esto cuando pase esto.

En el diagrama nos encontramos con diferentes figuras que pueden mantener diversas relaciones entre ellas:

➔ Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad. Los casos de uso pueden tener relaciones con otros casos de uso. Sus relaciones son:

- Include: Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso.
- Extends: Una relación de un caso de Uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.
- Generalization: Es la típica relación de herencia.

➔ Actores: se representan por un muñeco. Sus relaciones son:

- Communicates: Comunica un actor con un caso de uso, o con otro actor.

- Parte del sistema (System boundary): Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

En la figura 2.10 tenemos un ejemplo; encontramos tres casos de usos Crear producto utiliza Validar producto, y Crear pack productos es una especialización de Crear productos

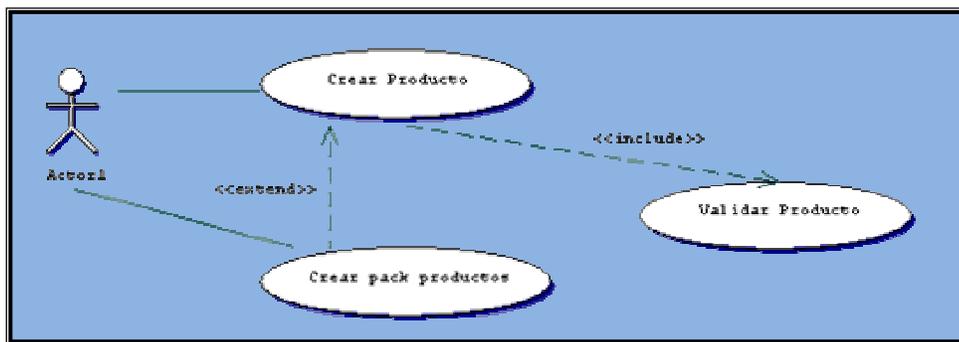


Figura 2.10 Ejemplo de la Relación en un Diagrama de Uso.

Podemos emplear el diagrama de dos formas diferentes, para modelar el contexto de un sistema, y para modelar los requisitos del sistema.

2.7.6.4.1 Modelado Del Contexto

Se debe modelar la relación del sistema con los elementos externos, ya que son estos elementos los que forman el contexto del sistema. Los pasos a seguir son: (Ver figura 2.11)

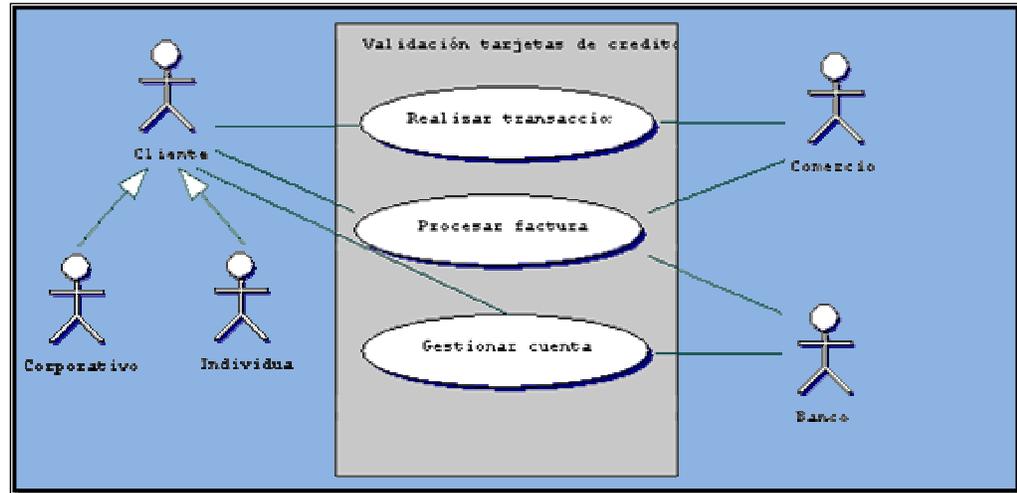


Figura 2.11 Ejemplo de Diagrama de Uso (Modelado de Contexto).

- Identificar los actores que interactúan con el sistema.
- Organizar a los actores.
- Especificar sus vías de comunicación con el sistema.

2.7.6.4.2 Modelado De Requisitos

La función principal, o la más conocida del diagrama de casos de uso es documentar los requisitos del sistema, o de una parte de él. Estos establecen un contrato entre el sistema y su exterior, definen lo que se espera que realice el sistema, sin definir su funcionamiento interno. Es el paso siguiente al modelado del contexto, no indica relaciones entre autores, tan solo indica cuales deben ser las funcionalidades (requisitos) del sistema. Se incorporan los casos de uso necesarios que no son visibles desde los usuarios del sistema. Para modelar los requisitos es recomendable:

- Establecer su contexto, para lo que también podemos usar un diagrama de casos de uso.
- Identificar las necesidades de los elementos del contexto (Actores).
- Nombrar esas necesidades, y darles forma de caso de uso.
- Identificar que casos de uso pueden ser especializaciones de otros, o buscar especializaciones comunes para los casos de uso ya encontrados.

En la figura 2.12 podemos ver se incluyen nuevos casos de uso que no son visibles por ninguno de los actores del sistema, pero que son necesarios para el correcto funcionamiento.

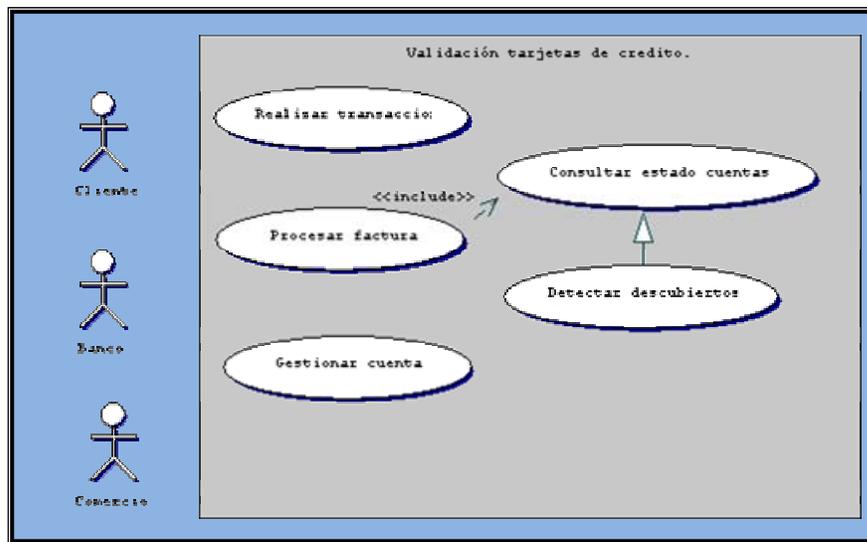


Figura 2.12 Representación Diagrama De Uso (Modelado de Requisito).

2.7.6.5 Diagramas de Interacción.

Muestran las interacciones ocurridas entre los objetos que forman parte de un sistema o escenario. Estos diagramas forman parte del modelado dinámico del

sistema y tienen la intención de verificar todo lo acontecido con el sistema desde el momento de su inicio hasta su culminación. Fueron desarrollados con el firme propósito de apreciar el tiempo de vida del o de los procesos que lo forman y la manera de interactuar entre ellos. En dichos diagramas se pueden definir tres elementos muy esenciales, que son: estados, eventos y actividades cada uno de los cuales son inherentes a la evolución completa de un proceso dado, es decir, los eventos pueden provocar cambios de estado en cualquier proceso y esto redundará a su vez en un cambio de actividades en el mismo. Dentro de los diagramas más importantes de comportamiento o interacción se tiene los siguientes.

2.7.6.5.1 Diagramas De Secuencia.

Los diagramas de secuencia muestran la interacción entre los componentes de una aplicación (objetos) en un intervalo de tiempo especificado. Estos diagramas presentan a los objetos que participan en la interacción y los mensajes que ellos intercambian organizados en una secuencia de tiempo. [1]

El diagrama de secuencia, como tal, facilita la visualización de los diagramas de casos de uso, pues pretende describir detalladamente todos y cada uno de los mensajes que son intercambiados entre los objetos que forman al sistema. Estos diagramas son bidimensionales.

Los objetos tienen un tiempo de vida para su ejecución. Este tiempo de vida se representa por medio de una Línea de Vida del Objeto, que no es más que una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos. El rectángulo de encabezado contiene el nombre del objeto y el de su clase, y se representa así: NombreObjeto:

_____ . Los rectángulos a través de la línea representan los momentos en que el objeto se encuentra activo. [1]

La activación de un objeto muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto, como se explico anteriormente.

Otra constante en este diagrama lo constituyen los mensajes. El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta. [1]

Por otro lado, se puede presentar el caso de condiciones con caminos alternativos en tiempo de ejecución, es decir, si la aplicación maneja procesos concurrentes pueden expresarse, en un diagrama de secuencia, diferentes alternativas de ejecución dependiendo de la lógica asociada con dicha aplicación. Estas alternativas pueden representar condiciones en la ejecución o diferentes hilos de ejecución (threads). [1]

Todo objeto tiene un final de actividades, es en este momento cuando se activa el destructor del mismo, es decir, han finalizado las operaciones para dicho objeto. En un diagrama de secuencia, y más aún, en la aplicación que este representa, es común encontrar objetos que terminan su ciclo de actividades y por lo tanto, para ser destruidos se debe señalar con una “X” al final de la línea de ejecución para dicho objeto, como se muestra en la figura 2.13. [1]

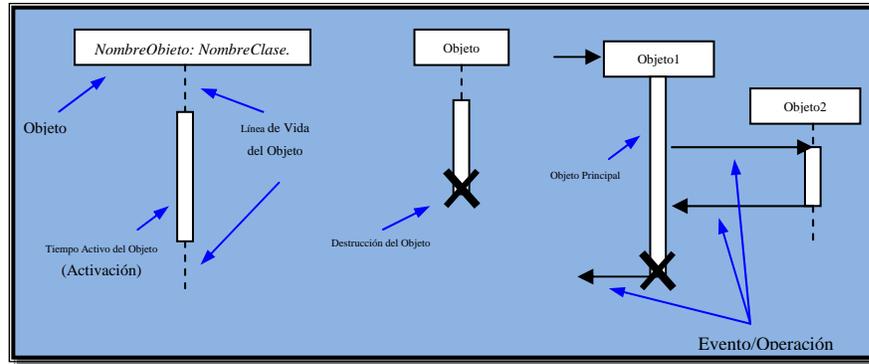


Figura 2.13 Elementos que Forman el Diagrama de Secuencia.

En el ejemplo de la figura 2.14 se muestran dos casos. Obj1 muestra una condición al enviar un mensaje a Obj3 o a Obj2, dependiendo de si $x > 0$ o $x < 0$. Estas dos líneas de ejecución se vuelven a unir más adelante, indicando el fin del condicional. Por otra parte Obj4 muestra dos posibles operaciones dependiendo de la condición anterior ($x > 0$ o $x < 0$). Ya que se presentan en el mismo instante de tiempo, se requiere dividir la línea del objeto en dos (esta misma representación se utiliza para el caso de dos hilos de ejecución). [1]

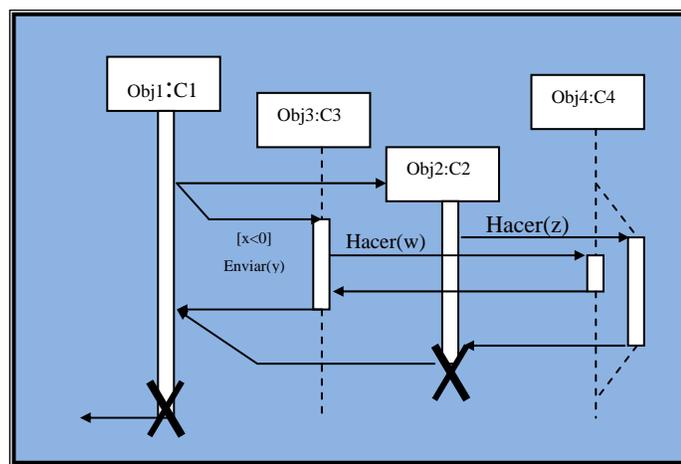


Figura 2.14: Ejemplo de Diagrama de Secuencia con Condiciones Que Generan Caminos Alternativos En Tiempo De Ejecución.

2.7.6.5.2 Diagramas de Colaboración

Muestran la interacción entre varios objetos y los enlaces que existen entre ellos. Un diagrama de colaboración puede especificar un contrato entre objetos, parte esencial para la descripción de un patrón de diseño y en el mismo se representan las interacciones entre los objetos organizados alrededor de los objetos y sus vinculaciones. [1]

A diferencia de un diagrama de secuencias, un diagrama de colaboraciones muestra las relaciones entre los objetos y no la secuencia en el tiempo en que se producen los mensajes. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en forma diferente. Estos diagramas están constituidos por objetos, enlaces y mensajes. [1].

2.7.6.5.3 Diagrama de Estado.

Están constituidos por una serie de estados, los cuales identifican períodos de tiempo del objeto (no instantáneo) en el cual el mismo está esperando alguna operación. Esta operación viene dada por medio de un *evento* que estimula o induce al objeto a cambiar a otro estado o continuar en el mismo. [1]

Los *Estados* se representa mediante un rectángulo con los bordes redondeados, y los mismos pueden tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado. [1]

Un concepto importante dentro de estos diagramas es el de *evento*, que no es más que una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de verdadero o falso.
- Recepción de una señal de otro objeto en el modelo.
- Recepción de un mensaje.
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.

Otro concepto necesario que se debe tener claro es el de *mensajes*, pues, además de mostrarse y comprobarse la transición de estados por medio de eventos, puede representarse el momento en el cual se envían mensajes a otros objetos. Los mensajes son eventos más complejos, puesto que llevan una información implícita de la acción o acciones que deben realizar.

Finalmente el concepto de *transición* es otro punto de importancia en los diagramas de estado. La transición ocurre cuando un objeto o una interacción pasa de un estado a otro por la ocurrencia de un evento. Existen varios tipos de transiciones entre objetos, entre las más conocidas están: Transiciones Simples, Internas y Complejas. Los elementos que forman un diagrama de estados pueden ser vistos en la figura 2.15. [1]

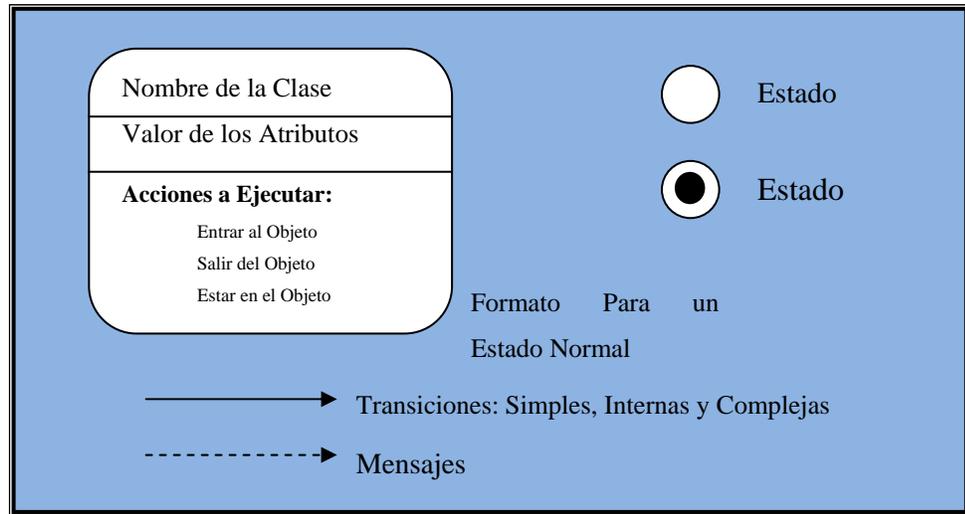


Figura 2.15 Elementos que Conforman un Diagrama de Estados.

2.7.6.5.4 Diagrama de Actividad.

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual aproximadamente todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos. Los diagramas de actividad se utilizan para mostrar el flujo de operaciones que se desencadenan en un procesamiento interno del sistema. Son similares a los diagramas de flujo de otras metodologías OO, es decir, generalmente modelan los pasos de un algoritmo o rutina desarrollada en un método o procedimiento de programa. [1]

2.8 HERRAMIENTAS A UTILIZAR EN EL DISEÑO 3D.

2.8.1 3D Studio MAX [4]

El 3DStudio MAX es una de las soluciones en el ámbito del diseño por ordenador recientemente creada, entre sus características principales tenemos la posibilidad de analizar el desarrollo de proyectos 3D en tiempo real, enlazar o importar datos y archivos de tipo CAD, incluyendo los más conocidos como son .DWG y .DXF de AutoCAD e incluso crear presentaciones multimedia interactivas.

3DStudio MAX es el complemento ideal de AutoCAD para la visualización y animación de los diseños y está especialmente concebido para el diseño de exteriores de edificios, espacios de interior, diseño de soluciones estructurales y visualización de productos industriales.

Es un software completo de modelado, renderizado, y animación completamente profesional y vanguardista, basado en la tecnología desarrollada por Kinetix para 3DStudio MAX, pero adaptado al estilo y necesidades de una oficina técnica especializada. Gracias a su entorno gráfico, centrado en la productividad para Arquitectura y Construcción, el diseño es un trabajo vital y creativo que facilita enormemente a los especialistas en estas aplicaciones tanto el desarrollo de proyectos en la oficina como la comunicación con los clientes.

El 3DStudio Max posee las siguientes Características que facilitan el diseño de entorno gráficos, Facilidad en el Manejo de Objetos, Entorno Gráfico Funcional e

Interactivo, Render Interactivo, Entorno de Trabajo Intuitivo y Flexible, Módulo de Animación Integrado y Sencillo.

2.9 AUTOCAD [5]

2.9.1 El Cad

Diseño Asistido por Computadora (CAD). Existen numerosos paquetes de CAD como IBMCAD, AutoCAD, etc. Su propósito es ayudar al diseño y graficación apoyándose en la computadora.

2.9.2 El AutoCad

Es una herramienta de aplicación CAD. La velocidad y precisión con la cual se pueden preparar dibujos utilizando una computadora ofrece un fenomenal avance sobre métodos tradicionales de dibujo.

2.9.3 Aplicaciones con AutoCad

- Dibujos Arquitectónicos.
- Diseño de interiores y fácil planeación.
- Diagramas de flujo y organizacionales.
- Dibujos para aplicaciones de Ingeniería electrónica, química, civil, mecánica, automotriz y aeroespacial.

- Mapas topográficos
- Ilustraciones técnicas y diagramas de ensamble.
- Logotipos para compañías.

2.10 COREL DRAW 10[6]

Es uno de los programas de edición gráfica más potentes y versátiles que existen. Permite crear una gran variedad de efectos y un gran número de objetos. Junto al Corel Draw también vienen otros programas gráficos como el Corel Photopaint, que sirve para retoque gráfico, el Corel Capture que sirve para hacer capturas de pantallas, o el Corel Texture que sirve para hacer texturas para los rellenos.

Hay que tener en cuenta que cuando se guarda un archivo de Corel Draw lo hace con la extensión .cdr, pero también trabajar con archivos en otros formatos como el .jpg o .gif.

2.10.1 Características

- Previsualización RealTime: Puedes ver una presentación preliminar de opciones y efectos de formato antes de aplicarlos a tu documento.
- Clasificador de páginas: Crea miniaturas de todas las páginas de un documento a la vez y vuelve a ordenar las páginas fácilmente.

- Formatos de archivos gráficos vectoriales: La nueva compatibilidad con formatos vectoriales SVG y SWF de Macromedia Flash™ te permite crear gráficos vectoriales compactos para Web.
- Documentos en varios idiomas: Coloca texto en varios idiomas dentro del mismo cuadro de texto.
- Compatibilidad con formatos de archivo: Intercambia archivos Macromedia® Freehand™ 8 y Microsoft® Visio® 5. Aprovecha además los nuevos filtros de importación y exportación DXF y DWG de AutoCAD.

Publicar como PDF 1.3 y PDF/X: Disfruta de nuevas funciones, tales como la posibilidad de añadir marcas de corte, barras de calibración, información de archivos, escalas de densito-metro y marcas de registro. Puedes también acelerar el flujo de trabajo con tamaños más reducidos de archivo y una representación más rápida y precisa.

2.10.2 Requisito mínimo del Corel Draw

- Windows® 98, Windows 2000, Windows NT® 4.0 o Windows Millenium.
- 64 MB de RAM (se recomiendan 128 MB).
- Pentium® 200.
- Unidad de CD-ROM de doble velocidad.
- Ratón o tableta.
- Monitor SVGA.
- 160 MB de espacio en el disco duro.

2.11 VRMLPAD

Es un editor específico desarrollado por **Parallel Graphics**, la misma empresa que desarrolla el visualizador Cortona. VrmIPad es una herramienta potente y flexible que te permitirá diseñar y desarrollar contenidos profesionales VRML. Podrás usar VrmIPad para crear mundos VRML para publicarlos en tu Web. VrmIPad soporta totalmente las especificaciones de VRML97.

2.11.1 Funciones

- El acceso tanto a archivos remotos como locales; soporte avanzado de búsquedas y reemplazos; trucos de ayuda para la sintaxis; rápido auto-completado; y detección dinámica de errores entre otras funciones más.

- Podrás ver, editar y navegar a través de las diferentes partes del documento. VrmIPad te permite organizar y optimizar tu escena con dependencias para publicarla en Internet, colocación de tus archivos en tu servidor Web, o envío por e-mail.

- Podrás pre-visualizar las escenas en los navegadores que tengas instalados y los visores VRML.

2.11.2 Características

- Pre-visualización de nodos individuales.

- No podrás grabar archivos más grandes de 64K.
- No podrás copiar o cortar selecciones de más de 32K al portapapeles.
- Sesiones limitadas a 2 minutos.

2.11.3 Requisito mínimo de VrmIPad

- S.O.: Win95/98/NT/ME/2000/XP

2.12 OTRAS DEFINICIONES DE INTERÉS.

- **Ambiente.**

En términos de Realidad Virtual, un modelo generado por computadora que puede ser explorado por un participante.

- **Browser.**

Programa usado para visualizar documentos WWW en Internet. Los más utilizados en estos momentos son Internet Explorer y Mozilla.

- **Campos.**

Partes integrantes de un NODO. Definiciones opcionales de los atributos de un nodo.

- **Detección de Colisiones.**

Acto que permite impedir el movimiento a través de objetos "sólidos" en un ambiente virtual.

➤ **Espacio 3D**

Un área real o imaginaria que puede ser definida en términos de un sistema de coordenadas X, Y, Z y de espacios ocupados (volúmenes).

➤ **Evento.**

Algo que ocurre. En VRML 2.0 un EVENTO se origina cuando la interacción o el movimiento de un participante u objeto "dispara" otra interacción o evento en el fichero de escenas.

➤ **Interactivo**

Que tiene rasgos que permite al usuario influenciar o manipular el curso de la acción; permite una interdependencia entre el usuario y el sistema.

➤ **Java.**

Lenguaje orientado a objeto que permite la manipulación de documentos en la Internet. Se persigue la incorporación del JAVA en el VRML 2.0 para el cumplimiento de diversas funciones que abran el camino a la creatividad.

➤ **Java Script.**

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas Java Script van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente; como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, etc.

➤ **Lod.**

(Level of Detail, nivel de detalle) La cantidad de detalle asignado a un objeto y a sus propiedades.

➤ **Mud/Moo.**

Ambientes multiusuario generalmente ligados a juegos.

➤ **Multiusuario.**

Se refiere a aquellos participantes representados por avatares dentro de un ambiente que se encuentran facultados para comunicarse entre ellos mediante tecnologías de texto y/o sonido. La capacidad MU es uno de los componentes clave para crear comunidades 3D en el Ciberespacio.

➤ **Mundo Virtual.**

Cualquier fichero en VRML con terminación .wrl y variantes.

➤ **Percepción**

La percepción es un proceso de organización e interpretación de información sensorial que se lleva a cabo en el cerebro y cuyo propósito es brindar significado a esa información que entra por nuestros sentidos. Tanto la sensación como la percepción son procesos inseparables. Cuando el cerebro recibe información sensorial de los nervios aferentes, por ejemplo, dicha información es automáticamente interpretada. Por tanto, muchos psicólogos se refieren a la sensación y a la percepción como un sistema unificado de procesamiento de información.

➤ **Plug – ins**

Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica. Ésta aplicación adicional es ejecutada por la aplicación principal. Los plugins típicos tienen la

función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar e-mail y filtrar imágenes de programas gráficos.

➔ **Primitivas.**

Cubo, cilindro, cono, esfera, texto.

➔ **Realidad**

Existencia real y efectiva de una cosa. Verdad, ingenuidad, sinceridad.

➔ **Sensores.**

Son recursos que proveen mecanismos para que el participante pueda interactuar con objetos en el "mundo" al cual ingresa.

➔ **Simulación**

Un proceso o aparato para generar condiciones de ensayo que se aproxima a las condiciones reales u operacionales, por ejemplo, el uso de simuladores de vuelo para entrenar pilotos; un sistema de imitación o que se usa para simular algo distinto.

➔ **Texturas.**

Detalles o imágenes mapeadas en superficies.

➔ **Tiempo Real**

Tiempo entre la entrada de los datos y la salida de la solución, donde la respuesta a la entrada es lo suficientemente rápida como para afectar las entradas posteriores.

➔ **Virtual**

Que tiene "virtud" para producir un efecto, aunque no lo produce de frecuente. Implícito, tácito. Que tiene existencia aparente y no real.

➤ **Virtualización**

Proceso mediante el cual un humano interpreta una impresión sensorial como un objeto en un entorno distinto al entorno en el que el objeto existe físicamente.

CAPITULO III – FASE DE INICIO

3.1 INTRODUCCIÓN

La mayoría de los proyectos requieren de una etapa inicial breve en la que se estudian los siguientes tipos de preguntas: “¿Cuál es la visión y el análisis del negocio para este proyecto?, ¿Es viable?, ¿Comprar y/o conseguir?, Estimación aproximada del costo, ¿Deberíamos abordarlo o no seguir?”.

Para definir la visión y obtener una estimación del orden de magnitud es necesario llevar a cabo alguna exploración de los requisitos. Sin embargo, el objetivo de la etapa de inicio no es definir todos los requisitos, o generar una estimación creíble o plan de proyecto. Aún a riesgo de simplificar demasiado, la idea es hacer la investigación justa para formar una opinión racional y justificable del propósito global y la viabilidad del nuevo sistema potencial, y decidir si merece la pena invertir en un estudio más profundo.

Por lo tanto, el Proceso Unificado de Desarrollo de Software se estructura en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada una de éstas se desarrolla a través de cinco flujos de trabajo: requisitos, análisis, diseño, implementación y prueba.

La Fase de Inicio es la más importantes en el desarrollo del proyecto, debido a que en esta fase se definirá y acordará el alcance del proyecto con las personas involucradas y responsables del sistema propuesto, identificando los requisitos, desarrollando el modelo de dominio y el modelo de casos de uso para los principales

procesos, así como también se identificarán los riesgos asociados al proyecto. A partir del modelo de casos de uso y la lista de riesgos, se determinará qué casos de uso deben implementarse y se establecerá la primera visión de la arquitectura candidata. Con base en la información previa se realizará el proceso de planificación general y un plan de trabajo detallado para la siguiente fase, así como el plan para la siguiente iteración de ser necesario. De esta manera podrá establecerse la viabilidad del sistema y se identificarán los posibles obstáculos que pongan en duda el desarrollo o culminación del proyecto.

3.2 PLANIFICACIÓN DE LA FASE DE INICIO

El propósito del desarrollo de esta fase es establecer una visión común inicial de los objetivos del proyecto, solo se pondrán en marcha los 3 primeros flujos de trabajo o actividades que se llevan a cabo en la aplicación del Proceso Unificado de Desarrollo de software, los cuales son la captura de requisitos, análisis y Diseño. Debido a esto no existen disciplinas o roles correspondiente a las etapas de implementación y pruebas. El comienzo de esta etapa se basa en la comprensión del contexto del sistema actual, esta comprensión se logra con el uso del Diagrama del modelo de Dominio del sistema y el modelo de casos de uso para los principales procesos, así como también se identificarán los riesgos asociados al proyecto. Esto permitiría la extracción de los requisitos funcionales y no funcionales para establecer una arquitectura candidata del sistema. En el capítulo 1, (Marco Teórico) ya se establecieron los requerimientos generales que un ambiente virtual debe contemplar en su primera etapa de desarrollo tomado de las metodología o secuencias de acciones para el diseño de un mundo virtual.

Se continúa con el análisis, aquí se refinan los casos de uso encontrados en el flujo precedente en forma de Clases de Análisis, Clases de Colaboraciones y Paquetes. El Diseño, la Implementación y la Prueba se tratan en fases posteriores. Los casos de uso tratados en esta iteración permiten cubrir las necesidades de la fase. Finalmente se establece la conclusión de la Fase de Inicio.

3.3 REQUISITOS

3.3.1 Comprensión de los Requisitos

Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar lo que se necesita realmente, de manera que tenga un significado claro.

Para establecer los requisitos iniciales (no muy explícitos), se consideró necesario implementar el modelo de dominio del sistema y modelo de casos de uso para obtener el modelo conceptual del negocio y así estudiar los requerimientos fundamentales del sistema.

3.3.2 Análisis del Ambiente Virtual actual AVUDONA_1 (AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI VERSION 1)

Es un Ambiente Virtual de las instalaciones de la UDO Núcleo de Anzoátegui, específicamente las que se observan al realizar el recorrido desde la entrada de la Universidad hasta llegar al Departamento de Computación y Sistemas. Interactuando y sintiendo sensación de presencia de estar moviéndose libremente en la Universidad, siendo algunas de las opciones que contiene este sistema.

3.3.2.1 Objetivos del Ambiente Virtual AVUDONA_1

- Permitir al usuario de Internet realizar el recorrido virtual por las instalaciones especificadas en el punto anterior.

- Demostrar que la metodología que se establece para la realización del Ambiente Virtual permite el logro del mismo de una manera satisfactoria y así poder aplicarla en la realización de futuros trabajos similares dentro de la Universidad.

3.3.2.2 Características AVUDONA_1

- El usuario del AV experimenta telepresencia limitada por estímulos visuales y auditivos dados por el computador.

- Es del tipo MUNDO REAL porque se obedecen las leyes físicas (Gravedad y Colisión) y además posee animaciones.
- Los guiones de comportamiento de los objetos animados son del tipo Activados por Gatillo, el cual permite activar las animaciones dado un evento por el usuario. Evitando la realización de las mismas cuando el usuario no las pueda apreciar.
- La interfaz gráfica son de dos dimensiones dados por los controles del Browser VRML y en tres dimensiones dadas por las interacciones con el Ambiente Virtual.
- La presentación o almacenamiento del AV se realizo desde un dispositivo de multimedia (CD-ROM).

3.3.2.3 Nivel de Detalle de AVUDONA_1

El nivel de detalle del Ambiente Virtual está demarcado por tres factores bien definidos.

- El primer factor es el tiempo de desarrollo. Para lograr un nivel de realismo semejante al del Mundo Real se requieren de varios rediseños de las escenas, esto implica que para desarrollar los objetos gráficos del AV se requieren de gran cantidad de tiempo de trabajo. Por esto se debe tomando en cuenta el estimado del tiempo utilizado para el desarrollo AVUDONA_1.
- El segundo factor es la velocidad. La frecuencia de los fotogramas con que se presentan las escenas debe ser mayor de 10 fotogramas por segundo, para no perder la continuidad de las escenas. La causas que producen la disminución de este factor son:

la cantidad de polígonos con el que se dibujan los objetos del entorno virtual, el número de texturas que se utilicen, las animaciones incluidas, los efectos de luz y sombras que sean aplicados. Por lo que se implementan sólo los objetos y efectos necesarios permitan lograr el mayor realismo del AV y no afecten en gran manera el número de fotogramas por segundo. Este factor es el más limitante por la importancia que tiene el grado de inmersión de los usuarios en los AV.

➡ El tercer factor viene dado por el tamaño de los archivos que componen el AV. Dichos archivos deben ser de un tamaño no muy grande para lograr el mayor rendimiento al descargarlos de la red.

3.3.2.4 División AVUDONA_1

Para AVUDONA_1 se dividió el AV en 4 zonas, el cual permito aislar físicamente unas de otras para ser diseñadas en forma paralela. El Ambiente Virtual se dividió en:

- ➡ Zona 1: Fachada de la Entrada Principal del Núcleo.
- ➡ Zona 2: Tramo Central que va Desde la Entrada Principal de la UDO Hasta los Pasillos y las Instalaciones de Ingeniería.
- ➡ Zona 3: Pasillos que llevan a las distintas Instalaciones del Departamento de Computación y Sistemas (Sala de Micros, Cubículos, Secretaría, Laboratorios).
- ➡ Zona 4: Las Aulas de Ingeniería, Departamento de Computación y Sistemas, Dirección de Escuela de Ingeniería.

3.4 ANÁLISIS DEL AMBIENTE VIRTUAL PROPUESTO AVUDONA_2 (AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI VERSION 2)

La propuesta de una nueva versión de AVUDONA, está fundamentada principalmente a la expansión del recorrido en el ambiente virtual y actualizaciones de los cambios que hayan ocurrido en su homónimo real de las escenas ya creadas, considerando incluir todos los avances tecnológicos que han surgido en la aplicación de interfaces 3D.

Puesto que es una aplicación desarrollada para un entorno Web, permite que se sea presentada y accedida desde un navegador Web, utilizando sistema manejador de base de datos para controlar el almacenamiento y acceso a los archivos VRML del ambiente virtual, siendo esto unos de los cambios significativo, ya que, permite el cambio de la presentación desde un dispositivos multimedia CD-ROM, como se realizo en AVUDONA_1, el cual, limita que pueda ser aplicación en la página Web de la Universidad y de esta forma lograr la vitalización de la misma.

Las Herramientas utilizadas fueron 3D Studio Max 7 y AutoCad para el diseño de las imágenes 3D, Corel Draw 10 para editar las texturas que se usaron en el diseño de las escenas, VRML 2.0(Lenguaje de Modelado de Realidad Virtual) para la creación del mundo virtual, JavaScript y Java (Que es un Lenguaje de Programación Orientada a Objetos) para crear los guiones de animación, SQL (Lenguaje de consulta Estructurado) como manejador de la base de datos y herramientas de almacén para archivos VRML y muy importante para la visualización del ambiente virtual Cortona® VRML client.

Con las aplicación de estas herramientas y apoyado en la metodología de proceso unificado UML, será posible desarrolla el sistema virtual, combinado con la metodología que ha venido utilizando para la creación de mundos virtuales se llevara a cabo una series de trabajos distribuidos en fase que nos permitirá cumplir con el objetivo y el éxito del proyecto. Siguiendo con el planteamiento de una metodología que permita el logro de futuros trabajo dentro de la Universidad.

El desarrollo del ambiente virtual será del tipo *Mundo Real* obedeciendo las leyes físicas (gravedad y colisión), donde el usuario podrá forma parte a través de la realidad virtual, *Telepresencia*, el cual, permitirá al usuario viajar por todo las escenas del mundo virtual, visualizando a través del monitor del computador y teniendo como dispositivos de control los periféricos (teclado, mouse) y los controles que ofrece la interfaz del Browser VRML y serán en 2D. Las animaciones o guiones de comportamiento de los objetos serán activados por efecto de gatillo, solo cuando el usuario provoque el evento, evitando que ocurra cuando no pueda apreciarlas.

3.5 DEFINICIÓN DEL NIVEL DE DETALLES DE AVUDONA_2

El nivel de detalle para la nueva versión AVUDONA, deberá estar enmarcados por los tres factores presentados en su primera versión, ya que hacen una definición del realismo que debe dar al ambiente virtual. Esto con el objetivo de lograr la armonía entres la versión 1 y 2 de AVUDONA, descrita en función del tiempo de trabajo, velocidad de navegación y tamaño de los archivos que componen el ambiente virtual.

Para continuar con el análisis, vamos a considerar los paso planteado en AVUDONA versión 1 de dividir en zonas el ambiente virtual, considerando que subdividir las en módulos nos permitirá aislarla físicamente unas de la otras de manera que pueda ser diseñada en forma paralela. Para AVUDONA_2 agregaremos dos nuevas zonas para alcanzar dividir el ambiente virtual en 6 zonas, las cuales se expresan a continuación:

- Zona 1: Fachada de la Entrada Principal del Núcleo.
- Zona 2: Tramo Central que va Desde la Entrada Principal de la UDO Hasta los Pasillos que llevan a las Instalaciones de Ingeniería (hasta el Auditorio) y Decanato.
- Zona 3: Pasillos que llevan a las distintas Instalaciones del Departamento de Computación y Sistemas (Salas de Micros, Cubículos, Secretaría, Laboratorios), Departamento de Industrial y Auditorio, Decanato, Plaza Decanato, Biblioteca “Luis García Pelissier” y Plaza Central “11 de Julio”.
- Zona 4: Las Aulas de Ingeniería (desde la I-1 hasta I-9), Departamento de Computación y Sistemas, Dirección de Escuela de Ingeniería, Centro de Fotocopiados, Sala de Navegación de la Esc. de Ingeniería.
- Zona 5: Las Aulas de Ingeniería (desde la I-10 hasta I-12), Departamento de Industrial, Auditorio, Baños y Aéreas de Servicio de Ingeniería.
- Zona 6: Biblioteca “Luis García Pelissier”, Plaza Central, Decanato, Plaza Decanato, jardines y alrededores.

3.5.1 Muestreo de datos para definir el nivel de detalle de AVUDONA_2

Durante esta etapa se procederá a recolectar toda la información y antecedentes acerca del objeto real que se desea modelar. Esta recolección varía dependiendo del objeto. Apoyado en los métodos obtenidos de trabajos realizados para el desarrollo de

ambientes virtuales y los aplicados en AVUDONA en su versión 1. Para el caso particular de objetos geométricos estáticos (muebles, casas, edificios, jardines, pasillos, escaleras, bancos), se recopilaron información referente a planos (elevaciones, de planta, cortes), trabajos previos en Autocad, fotografías digitales.

Para el caso de los objetos dinámicos (puertas, Lámparas), cabe decir, que son todos aquellos eventos transmitidos entre los objetos que forman el mundo virtual, a través de una ruta, que permiten a éstos presentar interacciones traducidas en movimientos, sonidos, o algún otro fenómeno perceptible para el usuario, ya sea en respuesta a eventos internos (un objeto cuyo movimiento afecte a otro) o externos (un click). Para esto se procedió a recolectar toda la información pertinente y así tener un pleno conocimiento de los mecanismos que gobiernan el movimiento de cada uno de los componentes.

3.5.2 Planos 2D, Planos de Levantamiento y Fachada de la Estructura Física de la Universidad.

A continuación se muestra la tabla 3.1. Contiene una breve descripción de los datos pertinentes a las zonas que serán actualizadas, las nuevas que van ser incluidas en el ambiente virtual, los formatos en los cuales se encuentran, el código asignado a los mismos y la fuente de donde se obtuvieron y la tabla 3.2, describe los objetos dinámicos, su comportamiento y eventos que los activan.

Tabla 3.1 Descripción de Planos.

DESCRIPCIÓN	FORMATO	FUENTE	CÓDIGO
De la zona 1: Foto de Fachada de la Entrada. Actualizaciones	Fotografías Digitales		Ninguno
De la zona 3, 4, 5, 6: Planos conjuntos, pasillos que llevan a las distintas instalaciones, solo (Decanato, Biblioteca, hasta el Auditorio)	Impreso, Autocad y Fotografías digitales	Planos y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-C.V.2-02
De la zona 4: Plano Sección Aulas de Ingeniería, Sala de Micros, Departamento, Sala de Navegación, Centro de Fotocopias.	Impreso , Autocad y Fotografías digitales	Planos y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-S.A.1-01
De la zona 4 y 5: Plano de las Seccione Aulas de Ingeniería I-10, I-12, Sala de Micros “René Cabrera”, Departamento Comp. y Sist. Fachadas Este, Oeste, Norte, Sur.	Impreso, Autocad y Fotografías Digitales	Planos y archivos pertenecientes al Departamento de Planta Física de la Universidad.	A-6
De la zona 4 y 5: Plano de las Secciones de Aulas de Ingeniería Cortes A-A,B-B,C-C,D-D	Impreso	Planos pertenecientes al Departamento de Planta Física de la Universidad.	NA-S.A.1-03
De la zona 6: Plano de la Plaza central “11 de Julio”.	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-PC-01
De la zona 6: Planos de la Plaza del Decanato (Cortes Monumento Central)	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-PDC-02
De la zona 6: Planos del Edificio del Decanato planta baja1, planta alta, techo y cortes ()	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-DC-01 NA-DC-04 NA-DC-05

De la zona 6: Planos del edificio de la Biblioteca Central, planta alta, sótano, cortes ()	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	A-3 A-4 A-5 A-8
De la zona 5: Planos Planta Ingeniería Industrial, fachada.	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-EI-02
De la zona 6: Planos del edificio del Auditorio, planta baja , techo, cortes y fachada	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	NA-A-03

Tabla 3.2 Descripción de los Objetos Dinámicos

OBJETO	DESCRIPCIÓN	FORMATO	FUENTE	EVENTO
Simulación Ardilla	De la Zona 1: simulación ardilla que habitan los jardines de la universidad	Fotografías Digitales, videos	Visual Propia	Evento activado por la presencias del usuario. Acción desplazarse.
Puertas Sala de micro “RENE CABRERA”	De la zona 2: Planos de la sala de micro, altura, ancho y grosor	Impreso, Autocad y Fotografías Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	Activado por el usuario a través de un click. Acción abrir
Puerta Centro de Fotocopias	De la zona 3: (Actualizaciones) Planos de la Escuela de Ingeniería	Impreso, Autocad y Digitales	Plano y archivos pertenecientes al Departamento de Planta Física de la Universidad.	Activado cuando el usuario hace click sobre la puerta. Acción abrir/cerrar

3.6 MODELO DE DOMINIO

Partiendo del estudio actual y apoyando en la metodología orientada a objetos, podemos decir, que la esencia del análisis o investigación es la descomposición de un dominio de interés de clases conceptuales individuales u objetos (las cosas de las que somos conscientes). Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.

El proceso unificado define un modelo de dominio como uno de los artefactos que podrían crearse en la disciplina del modelado del negocio.

Utilizando la notación UML, un modelo dominio no refleja el comportamiento funcional del sistema, es decir, no muestra funciones del software ya que tiene un fin de tratamiento y comprensión del problema, plantea el sistema desde los objetos más representativos de sus funcionalidades, tangibles o no. Y se pueden mostrar:

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

En la Figura 3.1, se muestra el Modelo del Dominio de ambiente virtual donde se muestran los objetos más importantes dentro del contexto del sistema, en su nueva versión que llamaremos AVUDONA _2.

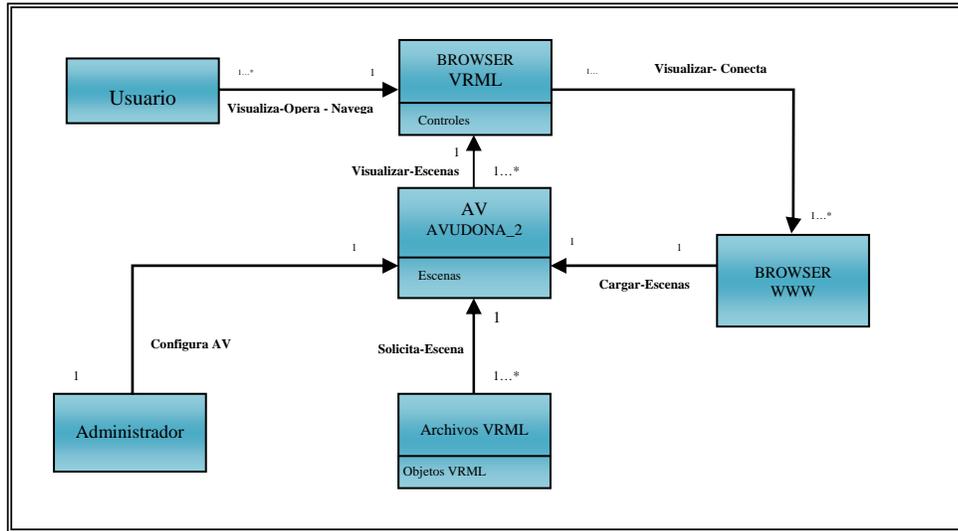


Figura 3.1 Modelo de Dominio del AVUDONA_2.

3.6.1 Contexto de AVUDONA_2

Cuando hablamos de contexto nos referimos a las circunstancias y condiciones que rodean un evento determinado. Haciendo uso de este concepto se determinó el contexto del presente sistema iniciando con el análisis del procedimiento que se deben llevar a cabo para visualizar el ambiente virtual utilizando mecanismos a través del WWW y algunos navegadores o visualizadores VRML que se conecta con el *browser* WWW a través de un API.

Los usuarios recorren el Ambiente Virtual de diferente manera permitido por el AV. Al solicitar visualizar el AV a través de un navegador web este se interrelaciona con el visualizador WWW (Browser WWW), el cual, carga los archivos VRML, las escenas del AV, conéctase al Browser VRML mediante un API y de esta manera el usuario visualiza las escenas tridimensionales (3D) que representan el homónimo real, desde la *Entrada del Nucleo*, todo el recorrido hasta llegar a la *Escuela de*

Computacion y Sistemas, Industrial y Anfiteatro, así como también la *Biblioteca Central y Decanato*. Cabe destacar que los detalles no son importantes en este momento. Solo se describe un modelo bastante simplificado del sistema en sí. Las clases conceptuales usadas en este modelo se describen brevemente a continuación:

- **Usuario:** esta clase representa todos los usuarios en general que inicie un recorrido o navegan por el Ambiente Virtual.
- **Administrados:** esta clase representa a un usuario que tendrá acceso al sistema y podrá llevar a cabo operaciones de mantenimiento y actualización del Ambiente Virtual.
- **Browser VRML:** esta clase conceptualiza la interfaz que permite al usuario interactuar con el Ambiente Virtual. Cada vez que un usuario desee navegar por AV este visualiza las escenas transferidas desde el archivo de VRML a través de la clase conceptual Browser WWW.
- **Browser WWW:** es la clase que representa la interrelación entre el usuario y el *Browser VRML*. Permite la relación de las escenas con los archivos VRML del AV.
- **AV AVUDONA_2:** esta clase es la total y completa representación de las escenas del Ambiente Virtual que llamamos AVUDONA en su versión 2. Permite la interrelación entre los archivos VRML, el browser www y el browser VRML.
- **Archivos VRML:** clase que conceptualiza el almacén donde están almacenados los Objetos VRML y cada una de las escenas que representan el Ambiente Virtual.

3.7 REQUISITOS FUNCIONALES DE AVUDONA_2

Los requisitos funcionales son los que se encargan de definir lo que la herramienta de software debe hacer. Define los alcances del sistema en cuanto a las acciones que debe de realizar, y en cuanto a la transferencia de datos entre todas los diferente procesos del Ambiente Virtual.

En el caso de este proyecto, los principales requerimientos funcionales son los siguientes:

➔ **Sensibilidad a la presencia de los usuarios.** Este proyecto gira en torno a la posibilidad de poder visitar las instalaciones de la Universidad de Oriente Nucleo de Anzoategui a distancia por medio de la WWW a traves de un Ambiente Virtual. Este debe tener la posibilidad de reaccionar a las acciones de un usuario que se encuentre realizando el recorrido, este debe tener una cierta interacción y comunicación entre los usuarios del ambiente virtual y el ambiente virtual entre si mismo. VRML 2.0 proporciona dentro de sus especificaciones, diferentes nodos que permiten a un mundo virtual, o un objeto contenido dentro de un mundo, el disparar ciertas acciones como respuestas a eventos provocado por el usuario. Estos eventos deben ser capturados y procesados por el mundo virtual, para producir respuesta al usuario en el entorno virtual.

➔ **Funcionalidad Global.** Este ambiente virtual debe permitir al usuario desplazarse por todas las escenas que lo conforman. Este recorrido se inicia desde la entrada del núcleo hasta todas y cada una de las instalaciones representada en el ambiente virtual ofreciendo la sensación de estar y conocer las instalaciones de la universidad, en ciertas escenas del recorrido el usuario podrá visualizar eventos

provocado por la interacción con el ambiente virtual permitiendo experimentar la sensación de tener presencia dentro del recorrido o ambiente virtual activo. La gran mayoría de la funcionalidad del ambiente estará dada a través de las opciones funcionales que ofrece el VRML en su versión 2.0, y en aquellos casos donde se requiera funcionalidad más compleja, se completará VRML a través de JavaScript y Java.

➤ **Alcance.** Con anterioridad ya se ha mencionado que el objetivo de este trabajo es modelar mediante un ambiente virtual las instalaciones de la Universidad de Oriente Núcleo de Anzoátegui el cual tiene como objetivo motivar a la virtualización de la institución y establecer una metodología que permita guiar en el desarrollo de ambientes virtuales.

Los mencionados anteriormente son los requerimientos funcionales particulares de este proyecto, sin embargo, en el capítulo 2, (Marco Teórico), se discuten requerimientos generales que deben contemplarse dentro del desarrollo de todo ambiente virtual. En este caso en particular vamos a establecer, los requerimientos no funcionales en función de los requerimientos funcionales.

3.8 REQUISITOS NO FUNCIONALES DE AVUDONA_2

Los requerimientos no funcionales son aquellos que definen lo que la herramienta de software debe tener en cuanto el nivel de detalle, sensación, operabilidad y mantenimiento.

➤ **Características de la población objetivo.** Se refiere a cuestiones como al tipo de usuario al cual va está dirigido el ambiente virtual, características y habilidades que debe presentar no son completamente necesario en cuanto a un conocimiento básico o experiencia con la realidad virtual. La habilidad, es el atributo importante del usuario para interactuar o moverse independientemente alrededor del ambiente virtual, solo debe tener un previo conocimiento con computadoras y videos juegos. El usuario tendrá la experiencia navegar por un ambiente virtual dinámico compuesto por escenas que representan el mundo real en tres dimensiones, estimulando el sentido del espacio que ocupan, y sonidos de efectos estereofónicos. Todo esto basados sobre “human-computer interation” (HCI) que nos ofrece las características de adaptar el diseño al nivel de experiencia del usuario final.

➤ **Justificación de los medios interactivos a utilizar.** Se ha determinado que el uso de sistemas informáticos para resolver problemas, se debe tomarse en cuenta siempre y cuando no exista otros mecanismos más viables. El caso de este proyecto, se tiene varias justificaciones que puedes apreciarse desde varios puntos de vista. El desarrollo dentro de las ciencias computacionales se caracterizan por llevar actividades manuales a procesos automáticos con el fin de facilitar el trabajo. En el capítulo II, se mencionan todas las ventajas que se manifiestan al lograr virtualizar un mundo real usando Realidad Virtual, y son precisamente estas ventajas que buscamos incorporar en la creación de AVUDONA_2. Otro aspecto importante que muestra una imagen 2D o 3D en una pantalla de computadora en lugar de proyectarla a un HMD a través de Sistemas Desktop De RV. Puesto que representan mundos de tres dimensiones los exploradores pueden viajar en cualquiera dirección dentro de estos mundos. En resumen, los sistemas desktop RV muestran mundos tridimensionales a través de pantallas de 2D utilizando la plataforma del WEB como presentador.

➤ **Operabilidad.** Es necesario proporcionar a los usuarios intrucciones claras que permita comprender la manera de interactuar en el sistema virutal. En vista de que no

todos los usuario posee la misma habilidad y destrezas en ambiente virtuale se debe proporcionar un sistema de ayuda en donde se explique detalladamente los controles y como utilizarlos para desplazarse por el ambiente virtual sin quitarle el atractivo al recorrido virtual.

3.9 MODELO DE CASO DE USO

El Modelo de casos de Uso se compone por varios objetos que intervienen en los procesos que un sistema es capaz de ejecutar, esta orientados a la representación grafica de requerimientos en este caso de AVUDONA_2. Los casos de uso son graficas de fragmentos de funcionalidad que el sistema ofrece cuando un usuario interactúa con una entidad y el sistema. Para conseguir un buen modelo de caso de uso, es necesario realizar la identificación de los actores del sistema, los casos de uso y describirlos para mostrar gráficamente la funcionalidad del sistema.

3.9.1 Identificación y Descripción de los Actores

Esta es una actividad que se realiza para encontrar los actores que interactúan con el sistema. Es de mucha utilidad para establecer las permisologías de usuario, la relación entre actores y una visión clara sobre el alcance de cada actor o entidad que interactúe con el sistema, sean software o hardware externos en cuestión. En la tabla 3.3 se muestran los actores principales de AVUDONA_2, identificados durante esta primera fase.

Tabla 3.3 Identificación y Descripción de los Actores de AVUDONA_2.

Actor	Descripción
Usuario	Este representa al usuario en general que navega por el ambiente virtual, el cual podrá recorrer de diferentes formas que permite el sistema, ya sea, Viajando, Moviéndose en el Ambiente Virtual, Entrando a las diferentes Salas, modificando la configuración para el recorrido y acceder a la ayuda del ambiente virtual en el momento que lo necesite.
Administrador	Este actor tendrá cierto privilegio dentro del sistema, ya que su principal función es realizar operaciones, tales como, agregar nuevas escenas, cambiar el comportamiento y funcionamiento del ambiente virtual por medio del módulo <i>Configuración</i> .
Browser VRML	El actor Browser VRML, que no es más que la herramienta de software encargado de relacionar a los usuarios con el Ambiente Virtual, es decir, permite la visualización del mundo, la interacción de los usuarios con los objetos presentes en el mundo, la activación y ejecución de animaciones, además de permitir la transmisión e interrelación del mundo con la WWW (si es remoto) por medio de la conexión con el Browser WWW a través de un API.
SMBD VRML <<Sistema>>	Es el Software encargado de la conexión con la base de datos, y el almacén que contiene la descripción del mundo, como los objetos que se encuentran dentro de él, los guiones y variables de ambiente.

3.9.2 Identificación De Los Casos De Uso De AVUDONA_2

Según los requisitos que hemos manejados hasta este punto, en cuanto a la funcionalidad del sistema y lo que debe manejar el sistema para presentar al usuario como interactúa con el ambiente virtual mediante procesos visuales representados por diagramas de caso de uso, como se muestra en la figura 3.2. Cada usuario tendrá acceso a uno o varios casos de uso determinado, de acuerdo a la función que este

manejo. Todo caso de uso individual representa las necesidades esenciales de AVUDONA_2, de tal manera de que nos aproximemos a la representación real de las necesidades del usuario.

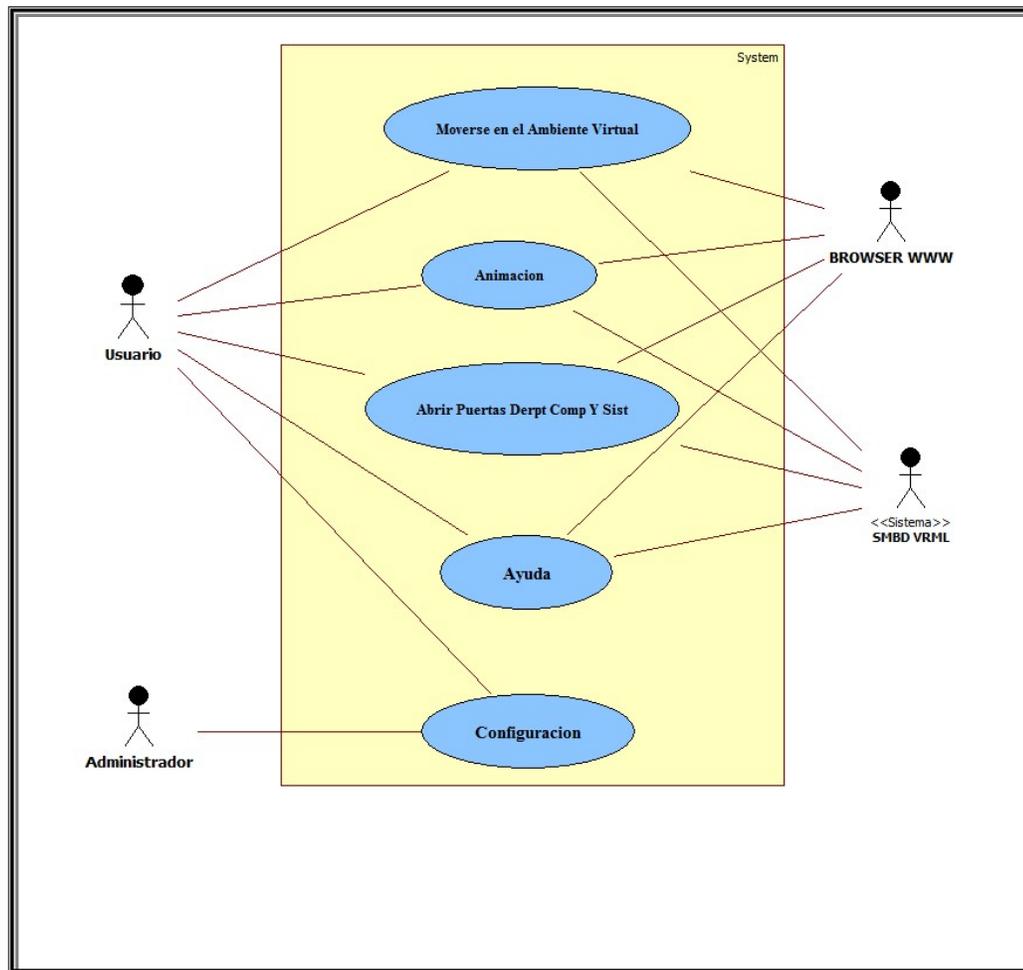


Figura 3.2 Modelo De Caso De Uso De AVUDONA_2.

3.9.3 Descripción de los Caso de Uso de AVUDONA_2

Tabla 3.4 Caso de uso “Moverse por el Ambiente Virtual”

Caso de Uso	Moverse por el Ambiente Virtual
Descripción	Estés proceso le permite al usuario, una vez que cargue el Ambiente Virtual, moverse libremente por cada una de las escenas que han sido representadas utilizando el menú punto de vista y los controles para la navegación 3D (Cortona VRML client).
Actores Implicados	Usuario Browser VRML SMBD VRML

Tabla 3.5 Caso de uso “Animación”

Caso de Uso	Animación
Descripción	Este permite que el usuario experimente o perciba en ciertos lugares del ambiente virtual las animaciones del mismo. Solo serán activadas únicamente por la virtual presencia del usuario.
Actores Implicados	Usuario Browser VRML SMBD VRML

Tabla 3.6 Caso de uso “Abrir Puerta del Dpto. Computación y Sistemas”

Caso de Uso	Abrir Puertas Dpto. Computación y Sistemas
Descripción	Una vez que el usuario inicia el recorrido, dentro del Ambiente Virtual existes escenas que el usuario no podrá visualizar sino accede a ellas mediante este proceso, permitiendo que el usuario pueda visualizar esas escenas (habitaciones) sin perder la ubicación en el Ambiente Virtual.
Actores Implicados	Usuario Browser VRML SMBD VRML

Tabla 3.7 Caso de uso “Ayuda”

Caso de Uso	AYUDA
Descripción	La información de cómo utilizar los controles en el visualizador, el dispositivo configurado para navegar, de cómo configurar el ambiente virtual para un mejor funcionamiento, las escenas representadas en el ambiente virtual y documentación d cada una de ellas, es reflejada en este proceso y puede accederse desde lugares estratégicos del recorrido por el usuario.
Actores Implicados	Usuario Browser VRML SMBD VRML

Tabla 3.8 Caso de uso “Configuración”

Caso de Uso	CONFIGURACION
Descripción	El usuario podrá acceder a configurar el visualizar según los requerimientos Software/Hardware utilizados para navegar. Desde cambiar la resolución y el tamaño de la presentación, la velocidad y los controles de navegación. Para el caso de actor Administrador tendrá un acceso exclusivo, donde podrá realizar modificación tanto como de las escenas como de configuración para posibles actualizaciones del Ambiente virtual.
Actores Implicados	Usuario Administrador

3.10 REQUISITOS ADICIONALES

Hay que tener en claro que los requisitos adicionales son fundamentalmente no funcionales que no pueden reflejarse en ningún caso de uso, pero de manera no especifica se asocian a ellos teniendo un impacto sobre vario o ninguno. Estos se obtienen a través de una lista de requisitos, y se utiliza durante el análisis y le diseño,

tomando en cuenta propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Permitiendo obtener básicamente limitaciones arquitectónicas y de rendimiento del sistema propuesto.

3.10.1 Requisitos Adicionales de Avudona_2.

➤ **Requisito hardware.** El ambiente virtual solo podrá ser visualizado en PC que cumpla con un requerimiento mínimo (procesado P4 1.2 MHz, memoria RAM de 256 Mb, tarjeta grafica de 256 Mb, monitos SVGA, mouse, teclado y cornetas).

➤ **Requisito software.** Los sistema operativo compatible para la visualizar el ambiente virtual son; Microsoft Windows (2000, XP, Vista) o Linux (cualquier distribución). Programas para navegación Web entre ellos tenemos: Microsoft Internet Explorer 6.0(o superior) y Mozilla Firefox 2.0 (o superior). Un Browser VRML necesario para visualizar ambiente virtual; Cortona® VRML Client 5.1 (o superior), esta interfaz se ejecuta como un plug – in y permite al usuario interactuar con el ambiente virtual. Este ultimo debe estar instalado en el computador de lo contrario no se podrá visualizar.

➤ **Facilidad de aprendizaje.** El ambiente virtual propuesto debe disponer al usuario todas las instrucciones necesarias para proceso de aprendizaje en el uso y mantenimiento del ambiente virtual de forma práctica, consistente y flexible.

3.11 ANÁLISIS

Para que el desarrollo de este proyecto concluya con éxito, es de suma importancia que antes de comenzar a codificar el programa que construirá la aplicación de software completa, se tenga en cuenta una completa y plena comprensión de los requisitos del software. Para esto, en este flujo de trabajo analizaremos los requisitos que se describieron en los flujos de trabajo anteriores, descubriéndolos, refinándolos, modelándolos y especificándolos. Este análisis nos permite conseguir una comprensión más precisa de los requisitos mediante un modelo de datos, flujo de información y control del comportamiento operativo del sistema de manera que nos ayude a estructurar el sistema entero incluyendo su arquitectura.

UML (Proceso Unificado de Desarrollo) respalda el modelo de análisis a través de una representación más formal por medio del uso de tres estereotipos tales como: clase de entidad, clase de interfaz y clase de control. Esto son estándares de UML y nos ayuda a proporcionar una vista interna del sistema, orientado más al lenguaje del desarrollador.

El resultado de este flujo es el modelo de análisis, proporciona una visión que argumenta el Proceso Unificado, es decir, la fundamentación en los casos de uso, dando forma a la arquitectura para soportar la funcionalidad. A continuación se presentará los diagrama de casos de análisis para los caso de uso de AVUDONA_2, donde se describirá como se lleva a cabo y se ejecuta en términos de las clases de análisis, de sus objetos, las interacciones entre objetos, los enlaces y los mensajes que envían esos enlaces.

3.11.1 Diagrama de Casos de Análisis Para el Caso de Uso Moverse en el Ambiente Virtual

Este diagrama de clases de análisis muestra de forma más detallada cómo interactúan las interfaces, los gestores y las entidades en el módulo *Moverse por el Ambiente Virtual*. Una vez que el usuario carga el ambiente virtual tiene como opción comenzar el recorrido directamente desde las diferentes escenas establecidas en el Ambiente Virtual o desplazarse libremente por las escenas de AVUDONA_2, es decir, el usuario puede tener más control al recorrer las escenas de su interés con más rapidez y eficiencias. En la figura 3.3 se muestra el diagrama clase de análisis *Moverse en el Ambiente Virtual*.

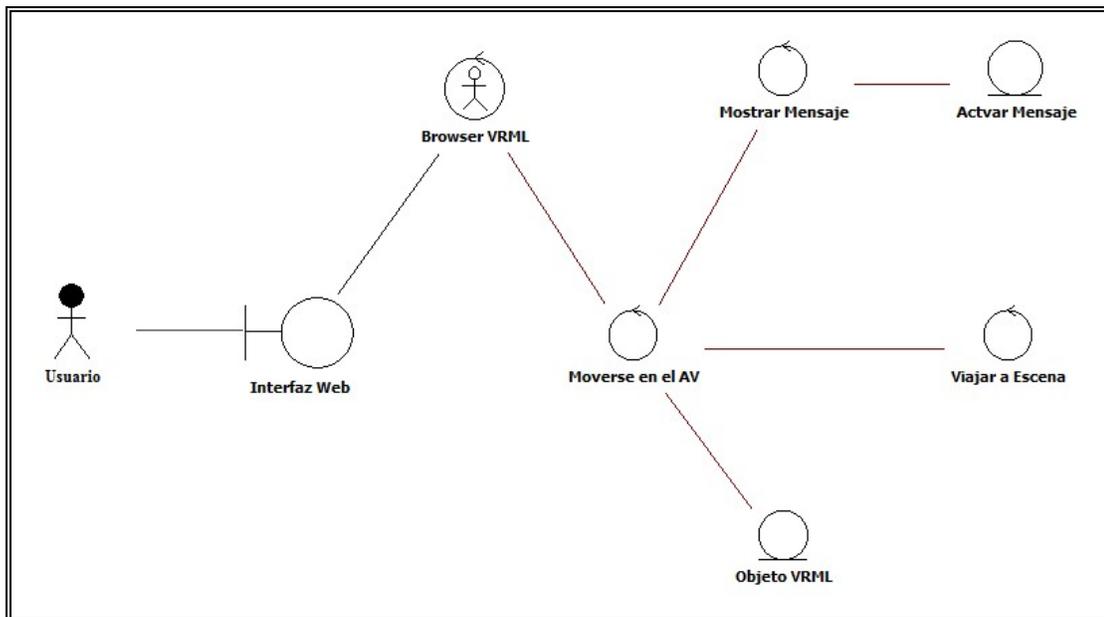


Figura 3.3 Clase de Análisis “Moverse en el Ambiente Virtual”.

La *Interfaz Web* es portal de WWW en donde el usuario invoca o activa el recorrido virtual, AVUDONA_2, este se comunica con el Browser VRML a través de API y visualiza el Ambiente Virtual.

La clase de control *Browser VRML* una vez cargado el ambiente le permite al usuario visualizar y desplazarse por las escenas que conforman el Ambiente Virtual.

La clase de control *Moverse en el AV*, es la que permite al usuario moverse libremente en el ambiente virtual e iniciar el recorrido desde las escenas establecidos en AVUDONA_2.

Las clase de control Viajar a Cámaras (Entrada Universidad, Plaza Bolívar, Aulas, Sala de Micro “René Cabrera”, Entrada Dpto. Comp. y Sist., Auditorio e industrial, Plaza Central, Plaza Decanato, Decanato, Biblioteca, Entrada Sala de Navegación, Entrada Centro de Copiado) son puntos de vistas ubicado estratégicamente dentro del ambiente virtual para que el usuario se desplace directamente a esos escenas y pueda comenzar o continuar el recorrido.

La clase entidad *Objetos VRML*, contiene la información referente al comportamiento del ambiente virtual, tales como, los enrutadores que llegan a los objetos Nodos Script, animaciones, interpolación y todo las escenas 3D que actúan en el funcionamiento e interacción con el usuario en el recorrido.

La clase de control *Mostrar Mensaje*, se encarga de gestionar todos los mensajes que son visibles cuando el usuario realiza el recorrido por el Ambiente Virtual de la clase entidad *Activar Mensaje*.

3.11.2 Diagrama De Colaboración

En esta fase del flujo de trabajo mediante el diagrama de colaboración vamos a destacar la organización estructural de los objetos que envía y reciben mensajes. Esta representación de su interacción nos permite identificar los requisitos y responsabilidades sobre los objetos, lo cual significa que se puede describir cómo se instancia la realización de caso de uso.

3.11.2.1 Diagrama de colaboración de Análisis del caso de uso Moverse en el AV

A continuación en la figura 3.4 se muestra la representación grafica de la colaboración entre los objetos del caso de uso Moverse en el Ambiente Virtual. Este modulo muestra el trabajo fundamental para que el ambiente virtual cumpla con los requerimientos del sistema, por eso es sometido a pruebas ya que representa un funcionamiento parcial de AVUDONA_2.

El diagrama de colaboración representa al igual que el diagrama de casos de análisis, las clases de interfaz, control y entidad, desde el momento que el usuario solicita visualizar el ambiente virtual, los mensajes de interacción que existe entre ellos cuando se llevan a cabo los procesos. Cuando el usuario desea *Activar el recorrido virtual* (mensaje:1) por medio de la clase de interfaz de Web, este suministra la información de *Cargar el Ambiente virtual* (mensaje:2) a la clase de control *Browser VRML*, que a su vez ordena *Cargar Escenas* (mensaje:3), y es recibido por la clase de control *Moverse en el AV*, luego según el usuario, *Activa Puntos de Vista* (mensaje:5) aquí el clase de control *Viajar a Escena* activa los punto de vistas (*Entrada Universidad, Plaza Bolívar, Aulas, Sala de Micro “René Cabrera”*,

Entrada Dpto. Comp. y Sist., Auditorio e industrial, Plaza Central, Plaza Decanato, Decanato, Biblioteca “Luis García Pelisser” , Entrada Sala de Navegación, Entrada Centro de Copiado) y devuelve el *Cambiar Punto de Vista* (mensaje:8) o *Activar Mensaje* (mensaje:6) a través de la clase control *Mostrar Mensaje* y así *Mostrar Mensaje* (mensaje. 9) de la clase entidad *Activar Mensaje*, para posteriormente *Solicitando los Objetos VRML* (mensaje:4), a través de la clase de clase entidad *Objeto VRML*, y devuelve las *Modificación de los Objeto* (mensaje:7) a la clase de control *Moverse en el AV*, enviado la *Visualizar Escenas* (mensaje: 10) al *Browser VRML*.

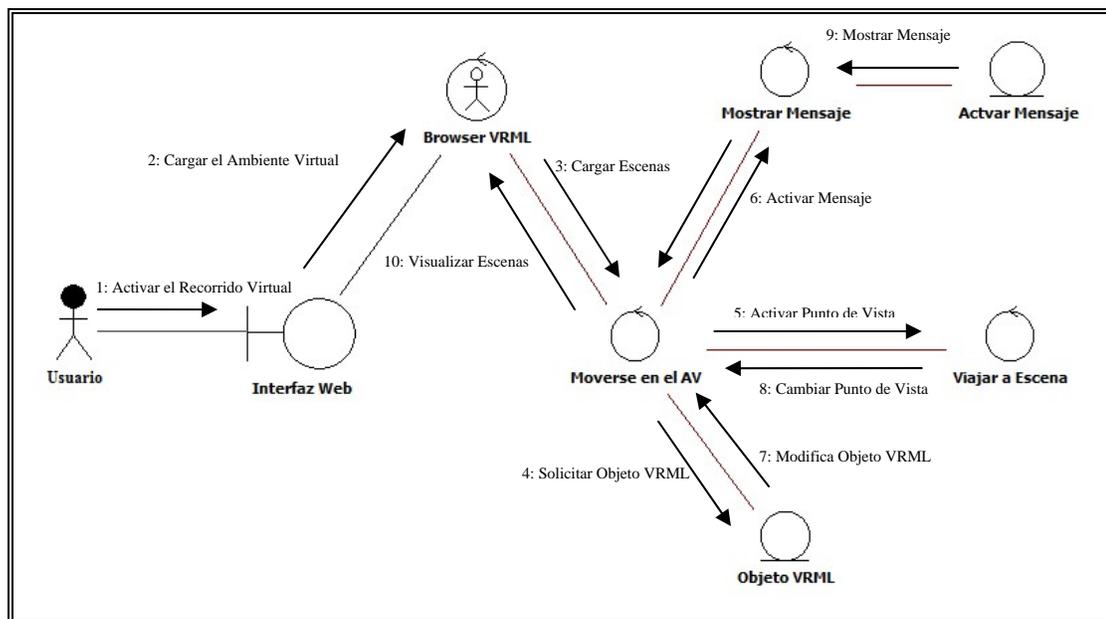


Figura 3.4 Diagrama de Colaboración de Análisis “Moverse en el AV”

3.12 DISEÑO DE AVUDONA_2.

El diseño es el primer paso hacia la fase de desarrollo de cualquier producto o sistema de ingeniería. Para la fase inicial, el flujo de diseño no es imperativo realizar estas tareas ya que no poseemos todos los detalles del complejo sistema que se construirá, lo que obtenemos es de manera tentativa es la arquitectura de AVUDONA_2, de una forma muy general y a su vez candidata como base para la elaboración del sistema. Todo basando en los requisitos obtenidos y análisis realizados, de igual forma se planteo la interfaz del usuario que el ambiente virtual podría presentar, dejando para la futura fase de elaboración lo concerniente al diseño de la base de datos. A continuación se trata el tema con más detalle.

3.13 ARQUITECTURA DE AVUDONA_2

En el diseño de la arquitectura se definió las relaciones que existe entre los principales elementos estructurales del programa. Para una herramienta de software basada en el desarrollo e implementación de ambientes virtuales éste es un aspecto fundamental dado que en esta representación del diseño se establece la estructura modular del software que se desarrolla. Dado que este proyecto pretende proponer una metodología para la construcción del ambiente virtual de la Universidad de Oriente Núcleo de Anzoátegui, la codificación y generación de ambiente y entornos virtuales es esencial.

Cuando se utiliza VRML 2.0 es necesario codificar cada instrucción que crearán un objeto determinado con sus propias características y atributos. Por tal

motivo debemos darle un alto nivel de modularidad a los mundos virtuales dado que cada objeto pueda manipularse, adaptarse y que su mantenimiento sea una tarea bastante sencilla.

La arquitectura de AVUDONA_2, se propone como una herramienta aplicativa dentro de un contexto real, está pensada de acuerdo a las propiedades estructurales y extra-funcionales que describen aspectos arquitectónicos como parte de un buen diseño. Estos aspectos representan los componentes del sistema, como se empaquetan y la manera de interactuar unos con los otros, ya que es necesario poder identificarlos individualmente que al unirse entre sí forman AVUDONA_2. Este proyecto propone un ambiente virtual integrado que forma una aplicación conjunta que puede presentarse apropiadamente al usuario final.

Como ya se mencionó anteriormente, VRML permite modularidad a través de ofrecer la posibilidad de crear los componentes de una escena virtual por separado. En la figura 3.5 un modelo que muestra la cohesión general que existe entre los diferentes elementos de la herramienta de software que este estudio implementa.

Los componentes más básicos de AVUDONA_2, son los nodos del lenguaje VRML 2.0 y los Scripts de comportamiento en Java y Java Script, estos crean y definen objetos de VRML que tiene sus propios atributos y características. Un conjunto o agrupaciones de objetos VRML forman archivos *.wlr en base a los objetos que los conforman pueden describir elementos más complejos de una escena virtual. Los ambientes virtuales importan una serie de archivos *.wrl que una vez integrados forman el ambiente virtual, que se incrusta en un documento HTML a través de Browser que es lo que da la interfaz con el usuario. De esta manera se tiene una modularidad total con respecto a la arquitectura del software, ya que para modificar un objeto se redefine ya sea el nodo(s) o el Script(s) que lo describe. De igual manera para modificar un archivo *.wrl puede manipularse individualmente los

objetos que los conforman, y de esta manera se puede dar mantenimiento al ambiente virtual.

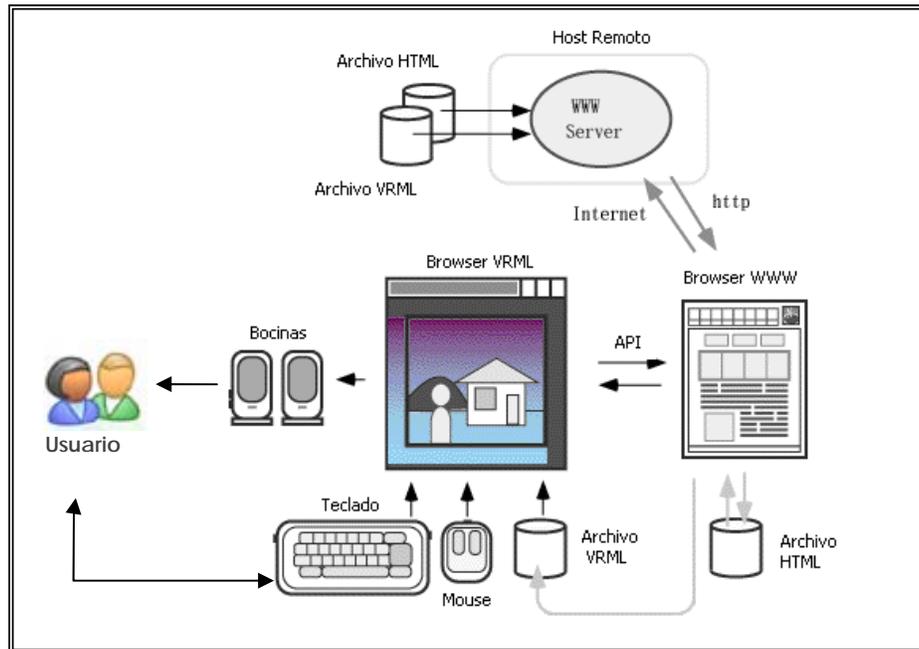


Figura 3.5 Arquitectura de AVUDONA_2

3.13.1 Interfaz de AVUDONA_2

El diseño de la interfaz describe como se comunica el software consigo mismo, con los sistemas que operan con él, y con los usuarios que lo emplean. Normalmente el proceso de diseño de la interfaz de AVUDONA_2 no se requiere del todo de la colaboración con los usuarios finales del mismo, ya que estudios hasta el momento ha establecido que la población objetivo no tiene experiencia previa en cuanto a ambiente virtual se refiere. Por ende, hay algunos elementos de la interfaz con el usuario que no solo son los más convenientes sino también los más apropiados. Las

consideraciones técnicas que tienen que tomarse en cuenta para desarrollar la interfaz son las siguientes

- VRML 2.0 tiene que ser desplegado por un VRML browser
- Un VRML browser es un plug-in que se integra a una página Web regular
- El VRML browser tiene comandos especiales para manipular las escenas virtuales
- Una escena virtual desarrollada en VRML 2.0 es, esencialmente, un documento presentable en una página Web que esta descrita en tres dimensiones (3D).

Otro aspecto que es de sumamente importante es el hecho de que el VRML browser por si solo tienes controles avanzados para navegar dentro de una escena virtual en tres dimensiones. Según el análisis realizado las interfaces se puede presentar de las siguientes maneras:

- Nodos d VRML 2.0 se comunican con otros nodos.
- Nodos que se comunican con scripts de comportamientos descritos en Java o en Java Script.
- Browser o visualizador, donde el usuario controla e interactúa con el ambiente virtual.

En la figura 3.6 se muestra la interfaz donde se visualizar el ambiente y los controles para desplazarse por el ambiente virtual.

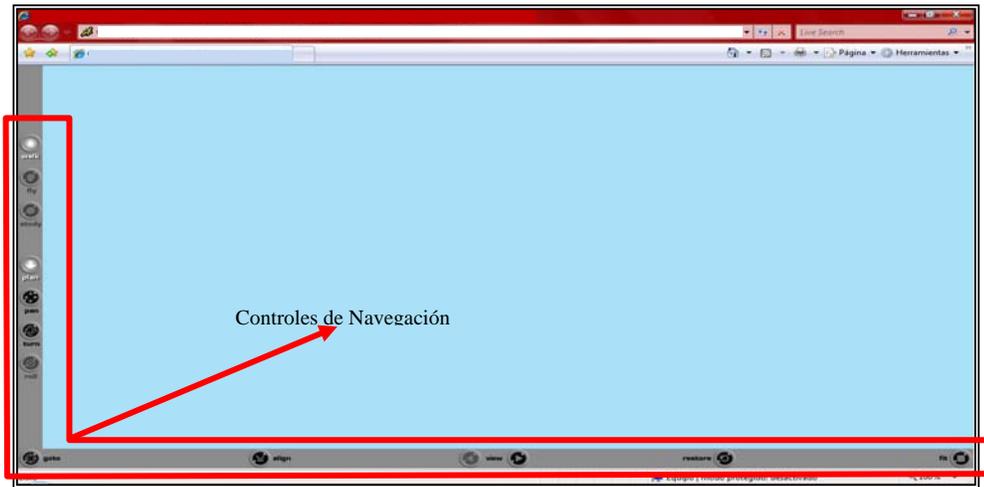


Figura 3.6 Interfaz de Controles de AVUDONA_2

CAPITULO IV – FASE DE ELABORACION

4.1 INTRODUCCIÓN

La fase de elaboración es la serie inicial de iteraciones durante las que el equipo lleva a cabo un estudio serio, implementa (programas y pruebas) el núcleo central de la arquitectura, aclara la mayoría de los requisitos, y aborda las cuestiones de alto riesgo. En el Proceso Unificado, el “riesgo” incluye valor del negocio. Por tanto, el trabajo inicial podría incluir la implementación de los escenarios que se consideran importantes, pero que no son especialmente arriesgados desde el punto de vista técnico.

La elaboración no se corresponde con una fase de diseño o una fase en la que se desarrollan completamente los modelos preparándolos para que se implementen en la etapa de construcción. Durante esta fase, no se están creando prototipos desechables; sino que el código y el diseño son proporciones del sistema final con calidad de producción.

Durante el desarrollo de esta fase, se realizaron los flujos de trabajo de análisis, diseño e implementación. Se terminaron de refinar los requisitos para que su diseño sea el más adecuado, y así fortalecer la arquitectura base del sistema.

4.2 PLANIFICACIÓN DE LA FASE DE ELABORACIÓN

En esta etapas, hemos logro una meta, haber finalizado la fase inicio obteniendo una arquitectura candidata que se quiere implementar, parte de su análisis y esbozos del diseño, teniendo así una base importante del proyecto. Se estima culminar esta fase con una sola iteración. El sistema en desarrollo es extenso y complejo, principalmente en la etapa de diseño de los objetos 3D, esta representa el bloque de construcción básico del proyecto, el cual, cubre la mayor parte del tiempo de trabajo en el desarrollo del proyecto.

Como hemos venido mencionando anteriormente, este un sistema basado en Ambientes virtuales, donde cabe destacar que el proceso unificado de desarrollo de software no contempla el análisis, diseño e implementación para sistemas basados en ambientes virtuales de una forma explícita. Debido a esto se realizo un estudio y evaluación de las mejores herramientas existente, hasta los momentos, para el desarrollo de ambientes virtuales.

Debido que aun no existe una metodología que cumplan con una filosofía de trabajo conocida y aceptada formalmente (como ocurre con la programación convencional) para la construcción de ambientes virtuales, y en vista al auge que la realidad virtual estas presentado con la infinidad de aplicaciones hacia todos los campos, esto ha permito que se desarrollaran una series de pasos a seguir que permiten realizar los estudios necesario, desde el análisis hasta el diseño y publicación del ambiente virtual a desarrollar, sustentado en la tecnología VRML (lenguaje modelado para realidad virtual).

Continuando con el desarrollo del proyecto, se recopilan los requisitos que queden pendientes mediante el Modelo de casos de uso, aunque en la fase de inicio se obtuvo un modelo parcialmente completo de este. La siguiente actividad del flujo de trabajo de análisis será describir la realización de casos de uso detallados, ya sea para modificar o agregar nuevas escenas al ambiente virtual, realizando análisis y restricciones. Al final de esta fase se trabajará en el diseño de la base de datos aplicada para VRML. Para obtener una arquitectura que sustentará al sistema y un diseño conceptual del ambiente virtual bastante sólido para proceder a la codificación de las mismas.

4.3 REQUISITOS

4.3.1 Caso de Uso Detallado de AVUDONA_2

Para una visualización más detallada de los procesos representados de forma general en el diagrama de contexto, se presenta a continuación la estructura interna de los procesos más representativos del sistema a través de sus en casos de uso y completar totalmente la descripción del sistema en base a su operabilidad.

4.3.1.1 Descripción del Flujo de Sucesos del Caso de Uso “Moverse en el AV” (Figura 4.1)

Nombre del caso 3: **“Moverse en el Ambiente Virtual”**

Actores: Usuario, Browser VRML, SMBD VRML<<Sistema>>

Descripción: Este proceso es el que le permite al usuario moverse libremente por el ambiente virtual, utilizando los controles del navegador “Cortona VRML client”, así mediante un desplazamiento sucesivo visitar cada una de las escenas que conforman AVUDONA_2 y además puede ir directamente a escenas que fueron definidas en el menú Punto de Vista de la barra de control.

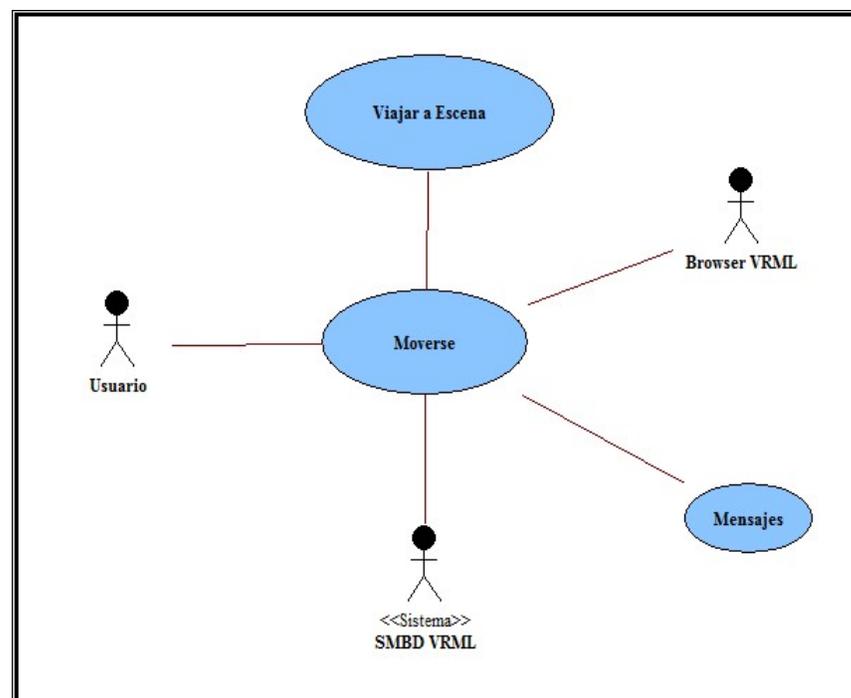


Figura 4.1 Caso de Uso “Moverse en el AV”

Pre – condición: Una vez cargado el ambiente el usuario puede desplazarse libremente por el ambiente o ir directamente activando los Puntos de Vista, usando los controles disponible del navegador y tener acceso la información si esta sobre la escenas definidas previamente.

Flujo de Eventos:

Flujo Principal:

1 – El usuario activa o invoca el ambiente virtual

2 – Cuando se carga el ambiente virtual el usuario, se desplaza por el ambiente virtual utilizando el menú en las barras del navegador.

3 – Si posición de usuario está dentro de la zona de información, entonces Muestra información de la escena.

4 – Si usuario sale de la zona de información, entonces Oculta información.

5 – Si usuario Activa un punto de Vista del menú (Entrada UDO o Plaza Bolívar o Plaza “11 de julio” o Sala “RENE CABRERA” o Aula Ing. o Plaza Decanato o Dpto. Computación y Sistemas o Centro de Fotocopiado o Sala de Navegación de Ingeniería o Auditorio)

, entonces

Activa punto de Vista Seleccionado

6 – Fin de caso de uso.

Flujo Alternativo:

1 – El Usuario tiene la opción de solicitar ayuda acerca de cómo manejar el navegador del ambiente virtual.

4.3.1.2 Descripción del flujo de datos del caso de uso “Animación” (Figura 4.2)

Nombre del caso 2: **Animación.**

Actores: Usuario, Browser VRML, SMBD VRML.

Descripción: En este caso uso representa las todas las **Animaciones** del Ambiente Virtual, las que son gobernadas por el sistema y la que ocurre cuando el usuario interactúa con los **Sensores** que disparan las animaciones.

Pre – condición: Una vez cargado el Ambiente Virtual y depende la posición del usuario puede visualizar e interactuar con las diferentes animaciones.

Flujo de Eventos:

Flujo principal:

1 – El usuario activa o invoca el ambiente virtual.

2 – Si hora del Sistemas es día o es noche, entonces

Día colocar panorama día

Noche colocar panorama noche

2 – Mientras usuario se desplaza, Si posición del usuario es igual a la posición de la animación, entonces

Disparar o activar la Animación

Repetir;

3 – El usuario visualiza la animación.

4 – Final del caso de uso.

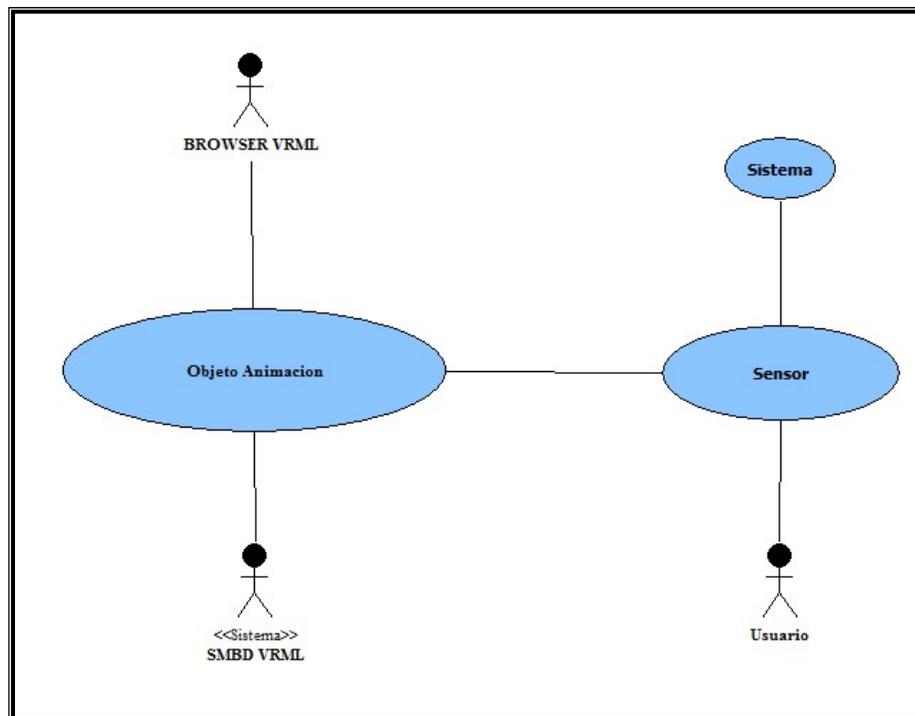


Figura 4.2 Caso de Uso "Animación".

4.3.1.3 Descripción del flujo de datos del caso de uso “Abrir Puertas del Departamento de Computación y Sistema” (Figura 4.3)

Nombre del caso 4: **Abrir Puertas del Departamento de Computación y Sistema.**

Actores: Usuario, Browser VRML, SMBD VRML.

Descripción: Este proceso le permite al usuario entrar en escenas que no tiene una visualización directamente, son escenas cerradas (habitaciones) y con solo hacer click en el objeto que permite el enlace entras en la escena.

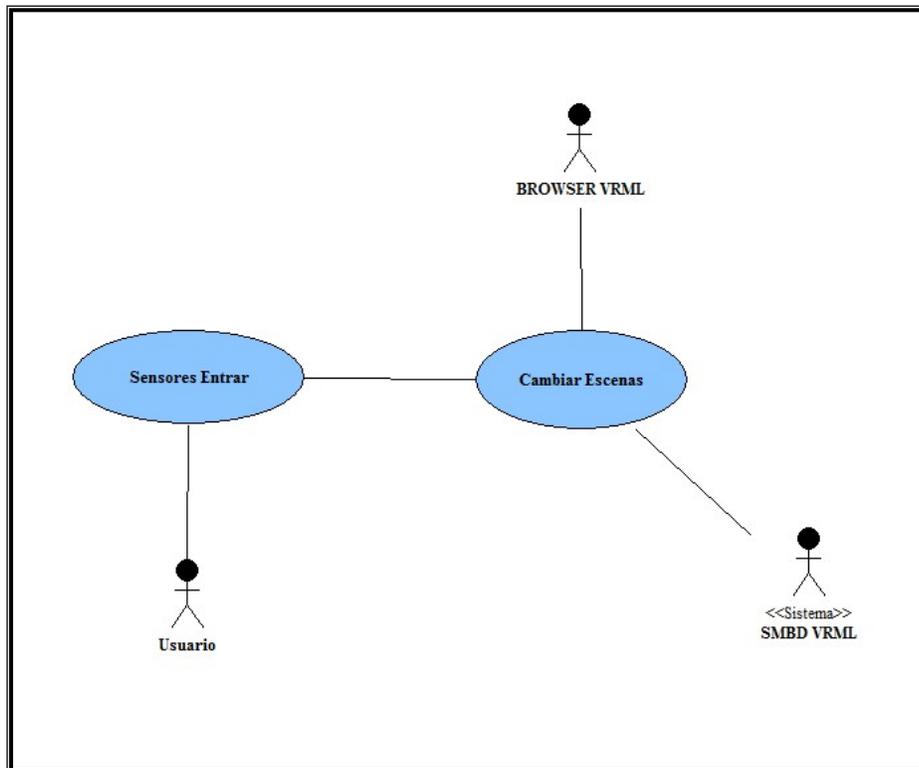


Figura 4.3 Caso de uso “Abrir Puerta del Departamento de Computación y Sistema”

Pre-condición: una vez que usuario carga el Ambiente Virtual, durante el recorrido tiene la opción de conocer o entrar en escenas si solo hace click sobre la representación de la puerta de entrada a la habitación y así cambia de escena.

Flujo de Eventos:

Flujo primario:

1 – *El usuario activa o invoca el ambiente virtual.*

2 – *Mientras el usuario se desplaza por el ambiente virtual, entonces,*

4 – *Si, objeto puerta es la entrada a la escena y usuario hace click sobre la puerta, para **Cambiar de Escena**, entonces*

Si Click Sobre Objeto Puerta Cubículo 1, entonces

Cambia Escena (AVUDONA_2 por Cubículo 1);

Si Click Sobre Objeto Puerta Cubículo 2, entonces

Cambia Escena (AVUDONA_2 por Cubículo 2);

Si Click Sobre Objeto Puerta Cubículo 3, entonces

Cambia Escena (AVUDONA_2 por Cubículo 3);

Si Click Sobre Objeto Puerta Cubículo Jefatura Dpto. Comp. y Sist., entonces

Cambia Escena (AVUDONA_2 por Jefatura Dpto. Comp. y Sist.);

Si Click Sobre Objeto Puerta sala de Navegación., entonces

Cambia Escena (AVUDONA_2 por Sala de Navegación.);

Si Click Sobre Objeto Puerta Sala “RENE CABRERA”, entonces

Cambia Escena (AVUDONA_2 por Sala “RENE CABRERA”)

5 – *Si usuario esta dentro de la escena y hace click en la objeto puerta sale de la escena.*

6 – *Si objeto puerta no es la entrada a una escena no pasa nada, sigue desplazándose por el ambiente*

7 – *Fin del caso de uso.*

4.4 ANÁLISIS

Continuando con el flujo de trabajo de análisis se tomará como entrada los casos de uso identificados en la sección precedente, de esta manera se plantearán las clases de análisis de alguno de ellos, seguidamente se establecerán sus diagramas de colaboración y por último los paquetes de análisis. Dichos artefactos se obtienen con la finalidad de precisar y fundamentar la línea base de la arquitectura ejecutable.

4.4.1 Diagrama de Caso de Análisis del Caso de Uso “Animación”

Este diagrama de caso de análisis muestra de forma detallada como interactúa la interfaz, los gestores y entidades en el modulo Animación cuando se representa una animación en el ambiente virtual, son objetos dinámico que cambian sus propiedades y atributos cuando son activado por algún factor externo o interno del recorrido virtual, reaccionan de la misma manera para producir las animaciones y se describe en la figura 4.5, a continuación.

El *Gestor Sensor_Disparador_Objeto* del tipo clase de control, recibe un evento generado por el usuario cuando se está *Moviendo en el AV* e indica que se debe producir una animación enviando un mensaje utilizando los *Enrutadores*.

Esta clase control *Enrutador_Objeto_Activo*, establece el enlaces de los mensaje entre los objetos involucrados en la animación.

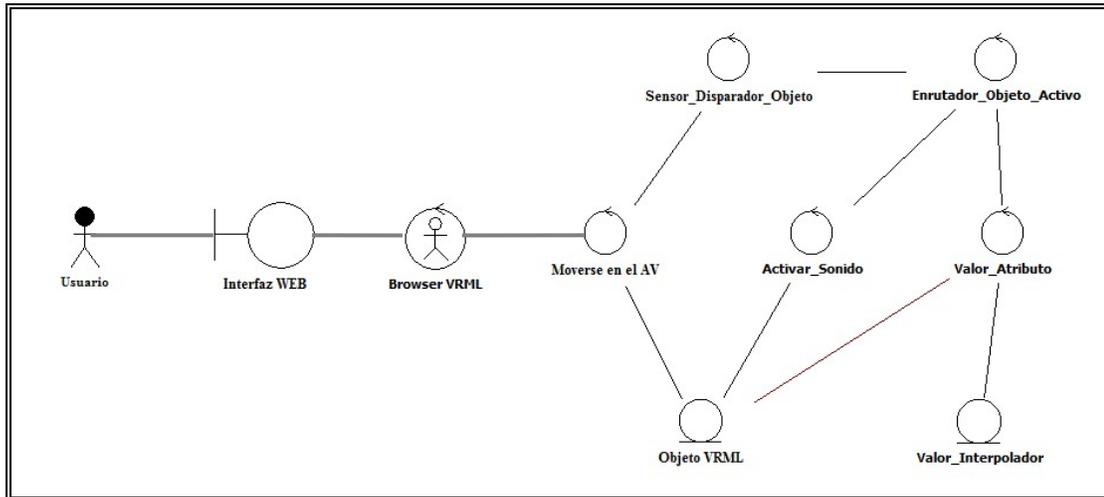


Figura 4.4 Clase de Análisis “Animación”

En la clase de control *Valor_Atributo*, genera los valores necesario a partir del valor un valor inicial enviado por *Sensor_Disparador_Objeto*, se generan consecutivamente cada intervalo de tiempo, enviados a los valore de *Interpolación* del objetos encargados de hacer las modificaciones a los objetos VRML involucrados en la animación, este proceso terminara cuando se generen todos los valores del intervalo.

La clase de control Activar sonido, gestiona los sonidos que vas ser reproducido si el objeto VRML tiene relación o enlaces con el objeto dinámico.

4.4.2 Diagrama de Clase de Análisis del Caso De Uso “Abrir Puertas De Computación Y Sistema”.

Este diagrama describe los procesos cuando el usuario esta navegando por el ambiente virtual y toma la decisión de entra en los ambientes: *Cubículo 1, 2, 3, Sala de Micro “René Cabrera”, Sala de Navegación, Jefatura del Dpto. de Comp. y Sist.*

Cuando el usuario decide entrar en las mencionadas escenas a través de la clase de control *Sensor_Disparador_Objeto*, que no están a simple vista en el ambiente virtual. A través de *Cambiar_Escenas*, una vez identificada en la entidad *Valor_Escena* realiza el cambio de *AVUDONA* por la escena solicita de la entidad *Objeto VRML*, todos este proceso ocurre para todos las escenas mencionadas arriba. En la figura 4.6 se muestra el diagrama de clase de análisis.

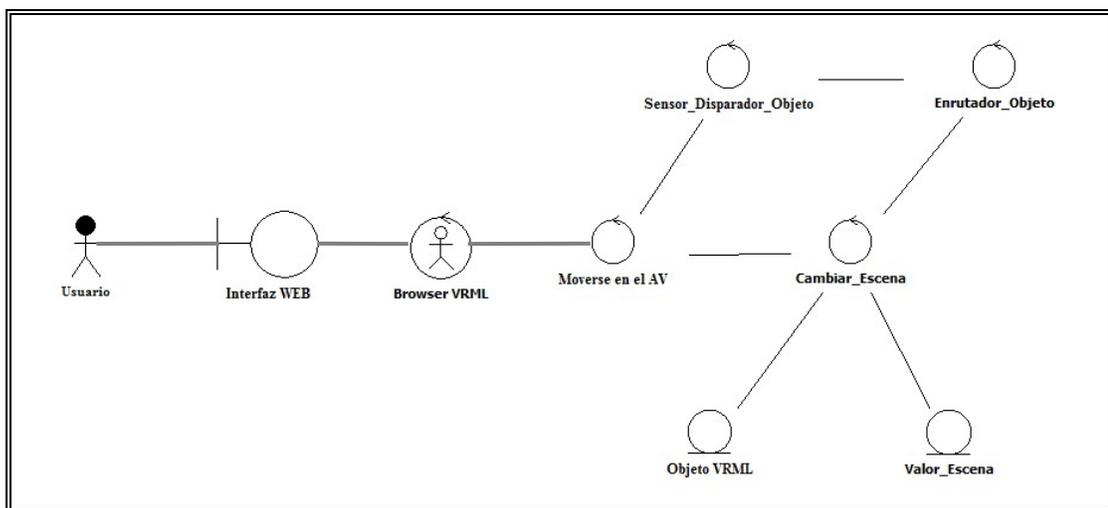


Figura 4.5 Diagrama de Clase de Análisis del Caso de Uso Abrir Puertas de Comp. y Sistema.

4.4.3 Diagrama de Colaboración

En un diagrama de colaboración, mostramos las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces y se centra en estudiar los enlaces que denotan el propósito del objeto invocante en la interacción con el objeto invocado. A continuación se muestra los diagramas de colaboración de *AVUDONA_2* de los módulos *Animación* y el de *Abrir Puerta de Comp. Y Sist.*

4.4.3.1 Diagramas de Colaboración “Animación” de AVUDONA_2

A Continuación se muestra el diagrama de colaboración del modulo Animación, donde se describe la relación y se puede observar en la Figura 4.6.

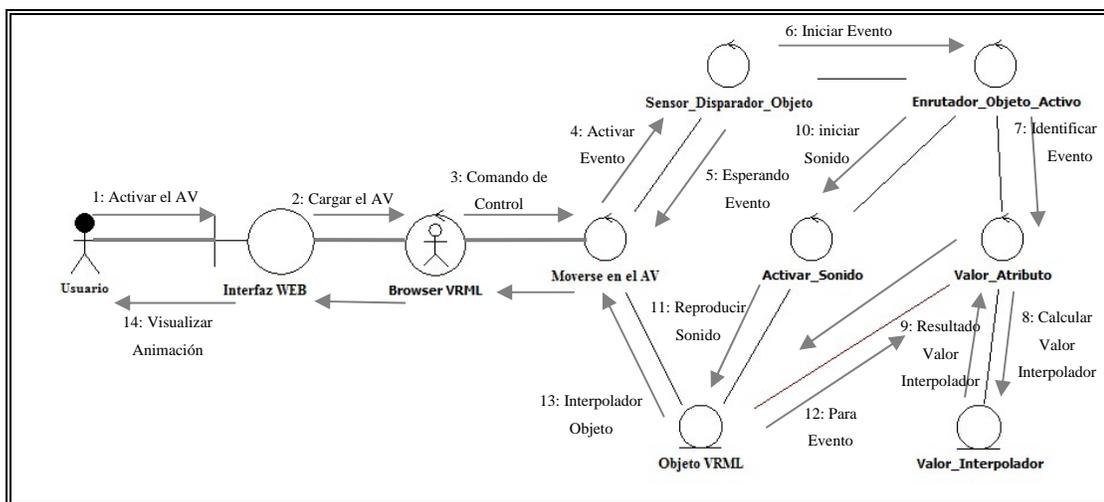


Figura 4.6 Diagrama de Colaboración del Caso de Uso Animación.

Tomando en cuenta la información que nos ha suministrado los diagramas de casos de análisis acerca de los procesos que se presentan en el sistema, este diagrama podemos ver la interacción que existe entre cada uno de los procesos que se presenta en el ambiente virtual, específicamente cuando se activa una animación, y los mensajes que ellos envían y reciben para ser efectiva esta interacción.

Mensajes:

1: *Activar el AV*: el usuario activa el recorrido virtual a través de interfaz web, este a su vez se comunica con el Browser VRML que es el encargado de gestionar e indica que se debe cargar el ambiente virtual.

2: *Cargar el AV*: este mensaje es suministrado al Browser VRML desde la interfaz WEB, indicando activar el control para cargar y visualizar el ambiente virtual, a través de este control el usuario interactúa con el sistema o mundo virtual.

3: *Comando de Control*: este mensaje representa las señales de la barra de navegación del Browser VRML que el usuario activa cuando interactuar con el Ambiente virtual.

4: *Activar Evento*: este mensaje es enviado hacia la clase de control *Sensor_Disparador_objeto*, ocurre una vez que usuario activa el evento ya sea por acción de un clic o simple presencia en la escena.

5: *Esperando Evento*: Cuando la clase de control *Sensor_Disparador_Objeto* detecta la presencia del usuario envía un mensaje que se encuentra sobre un sensor y puede activar un evento.

6: *Iniciar Evento*: este mensaje identifica el momento en que ocurre el evento para posteriormente activar los sucesos que controlan el evento mediante el control *Enrrutador_Evento_Activo*.

7: *Identificando Evento*: este mensaje provocado por la clase de control *Enrrutador_Evento_Activo*, el cual, identifica la ruta para enlazar los objetos para pasar los valores que son necesarios para el control del evento.

8: *Calcular_Valor_Interpolador*: son las coordenadas correspondientes al evento asignadas por la clase de control *Valor del Atributo*.

9: *Resultado_Valor_Interpolador*: es el mensaje correspondiente con toda la información del evento, así de esta forma se realizan las modificaciones del objeto que representa la animación para que sea visualizado por el usuario.

10: *Iniciar Sonido*: este mensaje indica a la clase de control *Activar_Sonido* que el evento tiene un sonido Adjunto.

11: *Reproducir_Sonido*: este mensaje envía la señal para que se reproduzca el sonido correspondiente al objeto involucrado en el evento.

12: *Parar_Evento*: este mensaje ocurre y es enviado por el *Objeto VRML* involucrado en el evento cuando llega al final de sus coordenadas de interpolación.

13: *Interpolador_Objeto*: este mensaje ocurre como respuesta a las coordenadas del Objeto VRML y así realizar las modificaciones en la posición y animación del objeto.

14: *Visualizar Animación*: este mensaje presenta el final del proceso, aquí el usuario podrá percibir la animación del Objeto que fue activado, a través de la *Interfaz Web*.

4.4.3.2 Diagramas de Colaboración “Abrir Puerta Dpto. Comp. y Sist” de AVUDONA_2

En la figura 4.7 se muestra el proceso que ocurre cuando el usuario que realiza el recorrido virtual de AVUDONA_2 desea entrar en los espacios no visible directamente del ambiente virtual, por tal motivo debe activar la visión de las escenas definida para tal fin. A continuación podemos observar la interacción entre cada uno de los proceso.

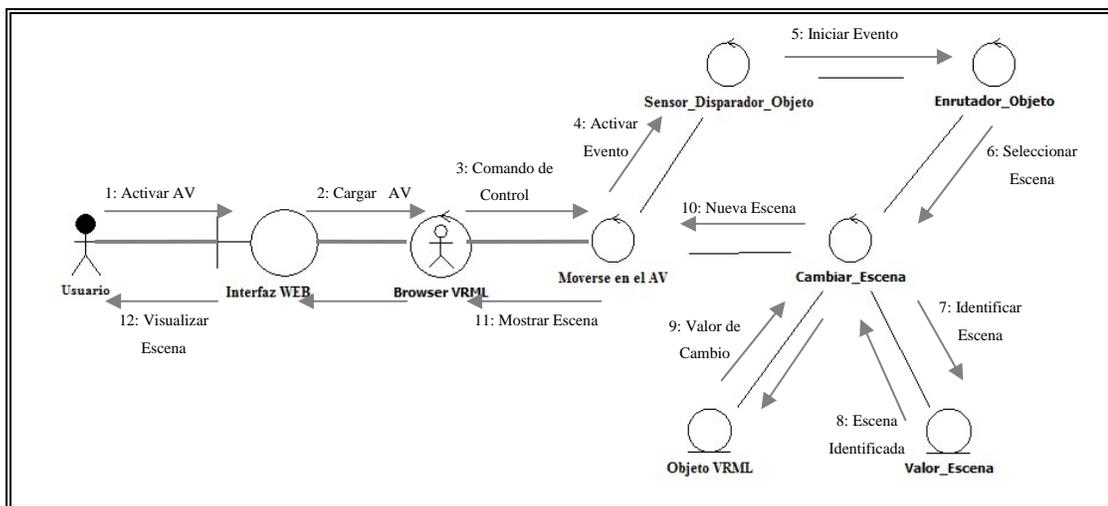


Figura 4.7 Diagrama de Colaboración General de AVUDONA_2

Mensajes:

Los mensaje 1, mensaje 2, mensaje 3, mensaje 4 y mensaje 5, tienen la misma función que cumplen en el procedimiento del diagrama anterior. Se comenzará la explicación desde el siguiente mensaje.

6: *Seleccione_Escena*: una vez que el usuario a cargado y está realizando el recorrido virtual, ha seleccionado entrar en unas de las escenas definida dentro del ambiente virtual a través del *Sensor_Disparador_Objeto* el *Enrutador_Objeto* envía el mensaje para establecer el enlaces entre los diferentes objetos que se van involucrar en el evento.

7: *Identificar_Escena*: este mensaje ocurre una vez que la clase de control *Cambiar_Escena*, identifica la escena que el usuario quiere visualizar.

8: *Escena_Identificada*: este mensaje corresponde a los valores de los *Objeto VRML* de la escena que se desea cargar.

9: *Valor_Cambio*: este mensaje contiene todo los valores y atributos de la escena para que la clase de control *Cambiar_Escena* realice el cambio y active la escena que el usuario desea visualizar.

10: *Nueva_Escena*: mensaje que envía la *Cambiar_Escena* a *Moverse en AV* indicando que se va carga nueva escena.

11: *Mostrar_Escena*: este mensaje representa el proceso que ocurre cuando se está cargando la escena en el *Browser VRML*.

12: *Visualizar_Escena*: mensaje visual en esta parte del proceso el usuario esta visualizando la escena que desea observar.

4.4.4 Diagrama de Paquete

El análisis realizado a los principales caso de uso, nos han permite establecer los mecanismo ideales para lograr la implementación en función de sus necesidades colectivas, a través de la formación de paquetes que determina sustentar una organización para así dar un soporte al diseño de la arquitectura. Este proceso permite agruparlos en paquetes generales que representa el sistema en su totalidad.

Los paquetes de análisis obtenidos para este sistema se pueden observar en la figura 4.8. Según sus casos de uso y sus respectiva relación.

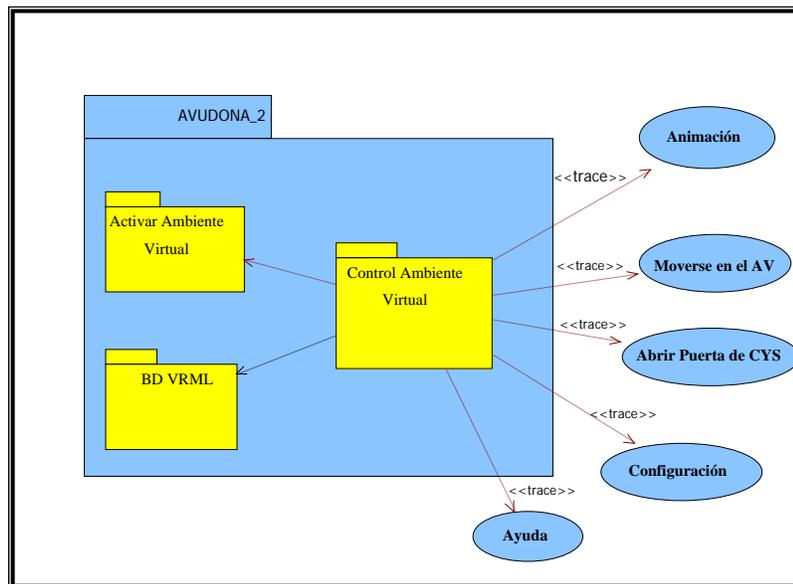


Figura 4.8 Paquete de Análisis del Sistema AVUDONA_2

Los procesos analizados en la realización de los casos de uso son, los paquetes Activar Ambiente virtual, Control del Ambiente Virtual y el paquete de BD Servicio, que representa la base de datos del ambiente virtual, y la funcionalidad necesaria para la conexión y almacenamiento de los Archivos VRML. El paquete Control de

Ambiente virtual tiene dependencia total de esta ya necesitas de los archivos 3D para visualizar el ambiente virtual. De igual forma tiene dependencia del paquete Activar Ambiente Virtual, ya que de aquí es donde el usuario activar el recorrido virtual.

4.5 DISEÑO

A partir de este flujo de trabajo se procederá a diseñar el modelo del sistema, tomando en cuenta el análisis hecho anteriormente. Este diseño nos permitió dar soporte a todos los requisitos funcionales, no funcionales y aquellos requisitos adicionales para así desarrollar la arquitectura del sistema. Esta fase nos proporciona una comprensión más detallada de los requisitos e impone una estructura del sistema creando una entrada apropiada y un punto de partida para las actividades de implementación subsiguiente. Se mostrara un diseño de la arquitectura y diseño de la base de datos para mundo 3D, y el diseño detallado del ambiente virtual utilizando pasos o metodologías VRML (Lenguaje Modelado de Realidad Virtual) que permite comprender la estructura de los objetos que forman el ambiente virtual.

4.5.1 Diseño Detallado de AVODONA_2

En esta parte del diseño se determinan y estudia toda la información que sea necesaria para el modelamiento de todos los objetos que formaran el ambiente virtual, sus propiedades específicas como, texturas, color, dimensiones, etc. Se recolectaron todas las muestra de los objetos que sea necesaria y haya quedado pendiente o necesitan ser modificados.

4.5.1.1 Antecedentes y Propiedades Acerca de los Objetos a Modelar

En la recolección se obtuvo la información de todos los detalles reales y propiedades adecuados para modelar los objetos estáticos, así como, dinámicos. Esto se logro realizando un recorrido por las diferentes zonas de la Universidad de Orienten, con el fin de determinar la información necesaria de los objetos a ser incluidos en el Ambiente virtual.

Se considero necesario realizar la captura de los objetos a través de fotografías digitales, para así tener de una forma permanente los atributos de las escenas al momento de diseñar los objetos 3D. En la tabla 4.1 se muestra algunas de las imágenes capturadas donde se determinan los antecedentes mencionados, acerca de las localidades a diseñar.

Tabla 4.1 Fotografías del Ambiente Real de la UDO a Modelar

Fotografía 1	Fotografía 2	Fotografía 3
		
De la zona 1: Jardín de la entrada de la UDO	De la zona 6: Edificio del Decanato; fachadas sur	De la zona 6: plaza central "11 de Julio"
Fotografía 4	Fotografía 5	Fotografía 6
		
De la zona 6: Edif. Decanato; Escaleras de acceso al piso 2	De la zona 4: Sala de navegacion de la escuela de Ingenieria.	De la zona 5: Auditorio de la UDO; entradas pricipales

4.5.1.2 Recolección de Muestras de las Texturas de los Objetos en el Entorno Real

Una vez identificados los objetos que van a formar el ambiente virtual, se procedió a tomar muestra de las texturas de cada uno, tomando en cuenta el nivel de detalle establecido en los análisis anteriores, para aplicar en algunos objetos de cada escena del ambiente virtual. En la figura 4.9 se muestra algunas texturas capturada por cámara digital e internet, cabe destacar que estas imágenes serán sometidas a procesamiento para facilitar la aplicación de las mismas y así darle una mejor apariencia de las que presenta estructura real de las escenas.



Figura 4.9 Textura Aplicada en el Diseño.

4.5.1.3 Identificación de los objetos que van hacer modelados en VRML

La identificación de objetos en el caso del modelado para VRML sigue un camino similar al análisis orientado a objetos. En primer lugar los objetos fueron

identificados a partir del análisis hecho a la especificación del modelo, y en segundo de los antecedentes recogidos a partir del muestreo. A continuación se pueden apreciar en la tabla 4.2 de algunos objetos, en donde, se define su estructura al nivel más bajo. Los eventos o sucesos definiendo claramente una series de movimientos y las señales que indique el punto de partida del suceso o evento.

Tabla 4.2 Identificación de Objetos (1/3)

Objeto	Descripción	Evento	Señal	Color o Textura
Portón	De la zona 1: Actualización portón de la entra principal del UDO	Estático	ninguna	Blanco
ardilla	De la zona 2: Actualización del desplazamiento al subir palmera	Dinámico, Simulación de una ardilla al correr para subir a un árbol	Activado por la presencia del usuario	Marrón
Monitor	De la zona 4: monitor de las computadoras diferente salas de CYS	Estático	Ninguna	Negro
Poste con lámpara	De la zona 5: ubicados en la adyacencias de la plaza “11 de Julio”	Estático	Ninguno	Color verde para el poste y blanco para la lámpara

Tabla 4.3 Identificación de Objetos (2/3)

reflectores	De la zona 1: reflectores que iluminan la entrada principal de la UDO	Dinámicos, simulación de luz que ilumina entrada principal UDO	Controlado por el sistema	Color negro y gris
Escalera	De la zona 5: escaleras principales de acceso al edif. Decanto	Estático	Ninguno	Gris y rojo
Biblioteca "Luis García Pelissier"	De la zona 5: edificio principal de la Biblioteca "Luis García Pelissier"	Estáticos	Ninguno	Anuncio tecnología Educativa, color Rojo, blanco, Gris
Ayuda	Se encuentra en la zona 1, 2, 5,6	Dinámico, dispone información de ayuda sobre la navegación en el ambiente virtual	Controlado por el usuario	Textura signo de Interrogación
Edificio Decanato	De la zona 5: Sede principal de actividades Administrativas y Académicas	Estático	Ninguno	Textura Logo UDO, vidrio, colores Rojo, Blanco, Marrón, Gris
Letra "D"	De la zona 5: Primera letra del anuncio que identifica al edif. Decanato	Estático	Ninguno	Azul

Tabla 4.4 Identificación de Objetos (3/3)

Pilar pasillo	De la zona 3: pilares de hierro que sostienen el techo del pasillo que va hacia la Esc. de Básico	Estático	ninguno	Marrón
Puerta	De la zona 4: Puerta de entrada a la sala “RENE CABRERA”	Dinámico, movimiento de rotación simulación al abrir una puerta	Con un click del mouse sobre la puerta	Marrón
Puerta 2	De la zona 4: puerta del centro de fotocopias de la Esc. de ING.	Dinámico, movimiento de rotación, simulación al abrir/cerrar una puerta	Con un click del mouse sobre la puerta	Negra
Escritorios	De la zona 4: escritorios de las diferentes salas micro y cubículos de profesores de CYS.	Estáticos	Ninguno	Textura de madera
Puerta 3	De la zona 4: puerta de entra a la aula AICS	Estático	ninguno	Blanca, manilla gris, vidrio negro
Placa plaza “11 de Julio”	De la zona 5: placa ubicada en el jardín central de la plaza	Estático	Ninguno	Textura placa plaza “11 de Julio”

4.5.1.4 Iteración de una Escena VRML para el Diseño

Siguiendo con diseño de AVUDONA_2, una vez conocido algunos de los objetos que van a conformar las escenas del ambiente virtual, sus atributos, eventos y conceptos claves que hay que tener en cuenta para el diseño adecuado. Se realizó una representación gráfica del VRML expresado bajo un mecanismo conocido como visualizador quien se encarga de desplegar las formas y sonidos en la escena gráfica, y suministrar soporte a los formas de interacción del usuario con el ambiente, en la figura 4.10 se muestra los componente que conforman el visualizador.

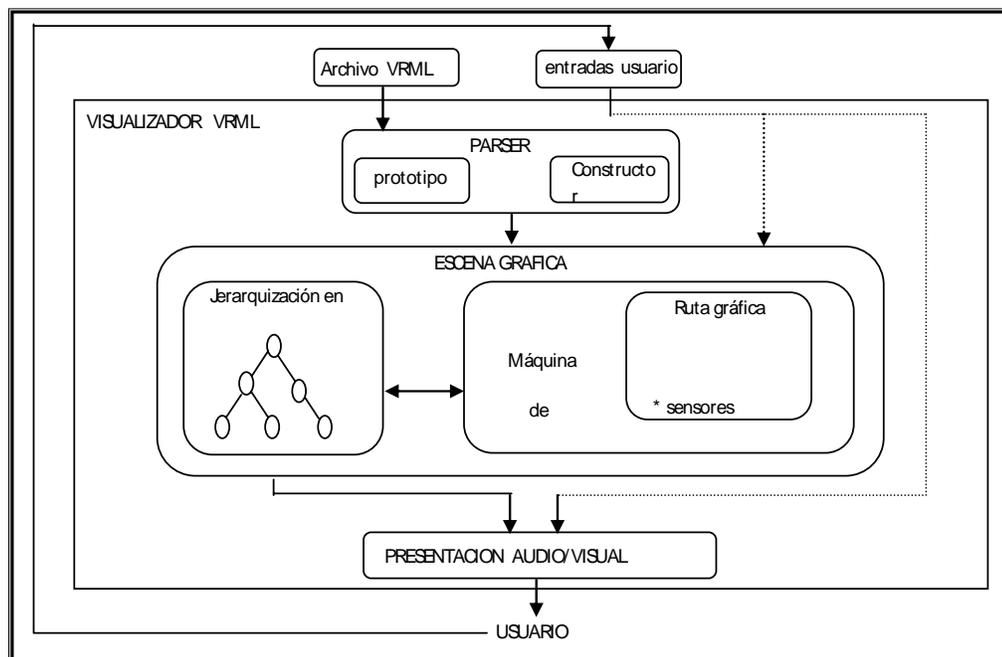


Figura 4.10 Componente del Visualizador VRML

Una vez entendido los componentes esenciales del visualizador VRML, tales como, el componente *Parser* que lee el archivo VRML y crea la escena gráfica, la *escena gráfica* que es una composición jerárquica de nodos y conexión entre los mismos, se encarga de manejar los eventos y cambios que se presenten en la escena

producto de los enrutamientos que tengan esos eventos y que modifique las propiedades de los nodos en escena, *el componente de presentación audiovisual* render de los objetos 3D y manejo del sonido, encargado de la realimentación de la escena con las posibles modificaciones que se presenten, conocido esto procedimos a realizar el diseño adecuado, tomando en cuenta todos los conceptos clave de las escenas graficas, nodos, campos, eventos y rutas.

Siguiendo adelante con esta etapa, a continuación representamos la estructura de algunos objetos del ambiente virtual, el cual, propone una notación que permite analizar gráficamente la interacción entre los diversos objetos. Debido a lo complejo de las escenas y el gran número de objeto, la notación VRML la representamos a un bajo nivel. En la figura 4.11 el diagrama del nodo *Ardilla*.

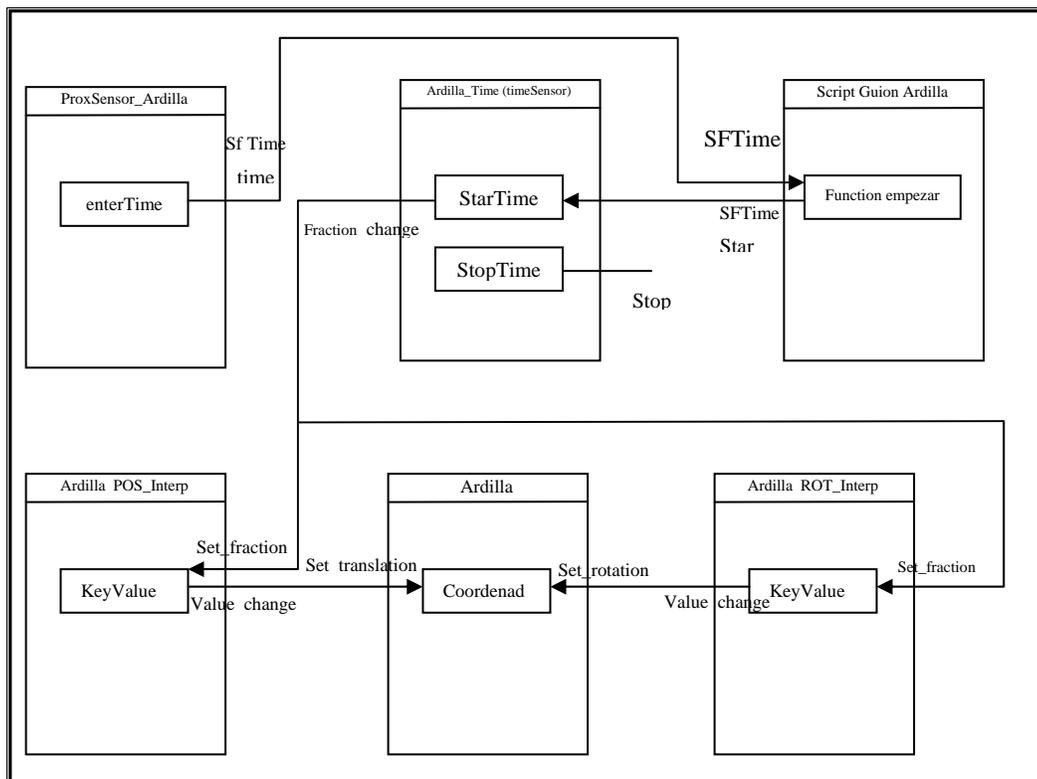


Figura 4.11 Diagrama del Nodo Ardilla.

Cada cuadro o ficha forma parte del nodo ardilla, una vez que el usuario comienza el desplazamiento en el ambiente virtual, puede interactuar con este nodo, cuando el sensor *ProxSensor_Ardilla* detecta la presencia del usuario este se activa enviando una señal de tiempo al nodo *TimeSensor_Ardilla_Time* indicando que debe comenzar la animación, a su vez este se encarga de controlar el tiempo de la animación e indica los valores sucesivos que permiten interpolar la translación y rotación del objeto.

Otro nodo importante que utilizamos para representar gráficamente y describe el comportamiento entre los nodos es *PuertaSalaMicro*. En la figura 4.12 está el diseño gráfico de dicho nodo.

La iteración de estos nodos son los que permite simular la animación de una puerta al abrir, específicamente la puerta de la Sala “RENE CABRERA”. Cuando el usuario tiene visible en el ambiente virtual el objeto que representa la puerta de acceso a la sala de micros y desee entrar a dicha sala, solo con hacer un click sobre el objeto *PuertaSalaMicro* de inmediato se desencadena una serie de eventos que le permite al usuario percibir, primero, la puerta abriéndose y segundo, la ubicación inmediata del usuario al interior de la sala de micro. Los nodos que intervienen en el mecanismo se describen a continuación; tenemos un nodo sensor, *EntrarSala-SENSOR* este identifica el evento exterior (un click), enviando una señal de tiempo al nodo *PuertaSalaMicro-TIMER* indicándole que debe comenzar la animación, esta señal de inicio es gestionada y transformada por el Script *Ir_Sala*, el cual, envía cada valor necesario, a través de los eventos, **output** enviados a los nodo *Swith* y *SwithI* respectivamente para activar la escena del interior de la sala de micro y desactivar el resto del ambiente, **cámara** este evento le permite al usuario ubicarse dentro de la escena a través de la cámara *SalaM*, por medio del evento **fieldofView** y al mismo tiempo desactivar el resto de las cámara por medio del Script *ApagarCamara*, **Auxiliar_fraccion** valore sucesivos enviados al nodo *PuertaSalaMicro* en donde

ocurre los evento que permite interpolar las coordenadas de translación y rotación respectivamente del objeto de esta forma el usuario pueda percibir la animación.

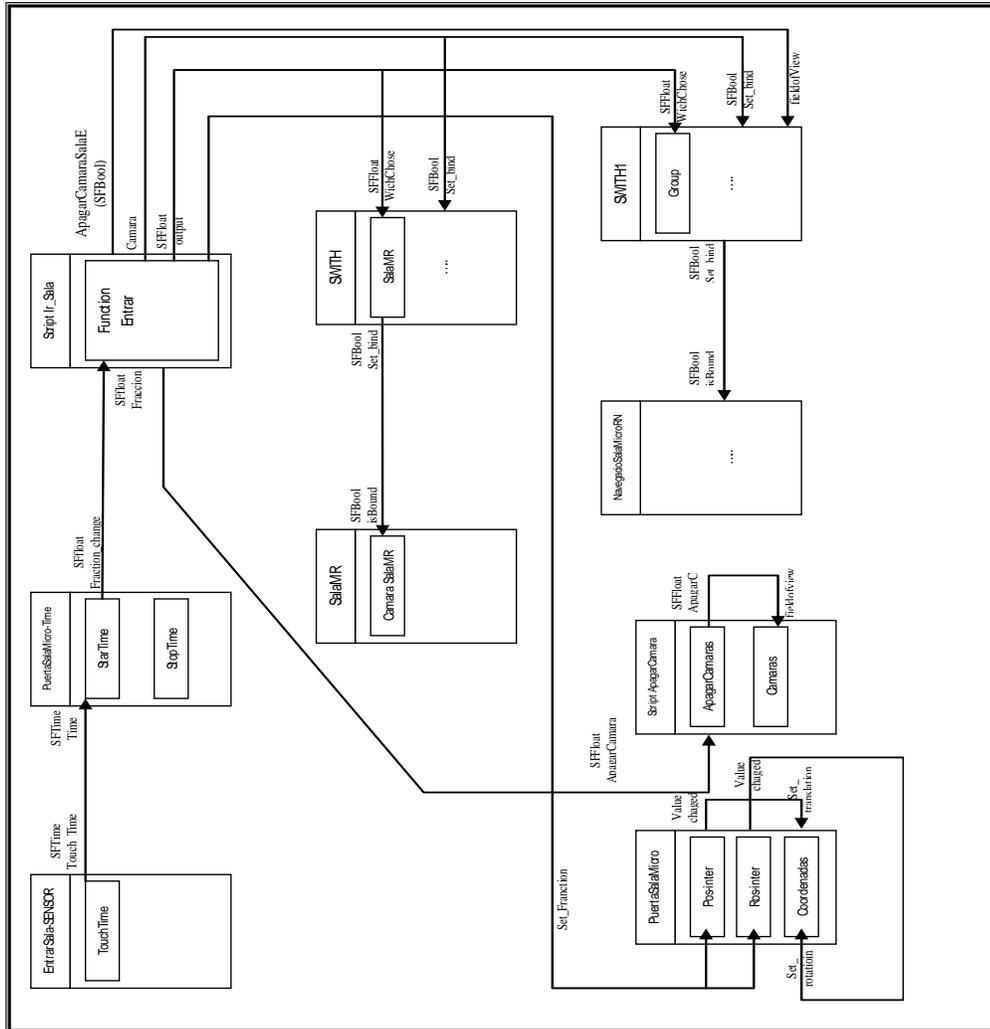


Figura 4.12 Diagrama del Nodo *PuetraSalaMicro*.

4.5.1.5 Estructura de Ensamblajes de AVUDONA_2

Como ya se ha mencionado, dentro del diseño para VRML se puede apreciar que un objeto identificado en las etapas primarias del análisis puede ser descompuesto en varias partes constitutivas y que pueden también ser definidas en sí mismas como objetos. Estas estructuras de este tipo son denominadas estructuras de ensamblaje y permiten definir las instancias entre los objetos, las modificaciones a los valores de los campos y atributos sin extrapolar a VRML.

Estructura de Ensamblaje del Nodo Ardilla. En la figura 4.13, cuando el usuario activa mediante la estructura *ProxSensorArdilla* la simulación de la ardilla corriendo, todo los nodo que instancia a la estructura *CuerpoArdilla* están supeditada a realizar el mismo desplazamiento y cambios en sus atributo.

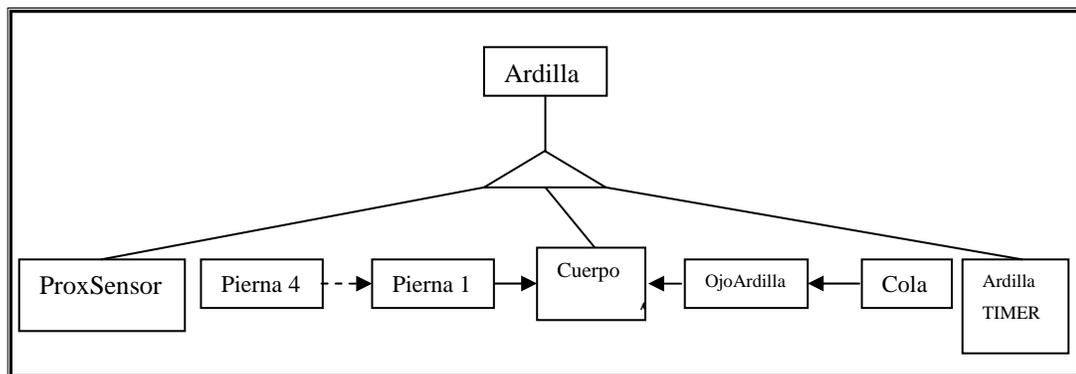


Figura 4.13 Estructura de Ensamblaje de Nodo Ardilla

Estructura de Ensamblaje de Nodo SalaM (sala “RENE CABRERA”), esta grafica de la figura 4.14, describe dentro del contexto VRML el diseño de los objetos que van a formar la escena, tomado como referencia de su homónimo real, sus atributo y el nivel jerárquico que representa.

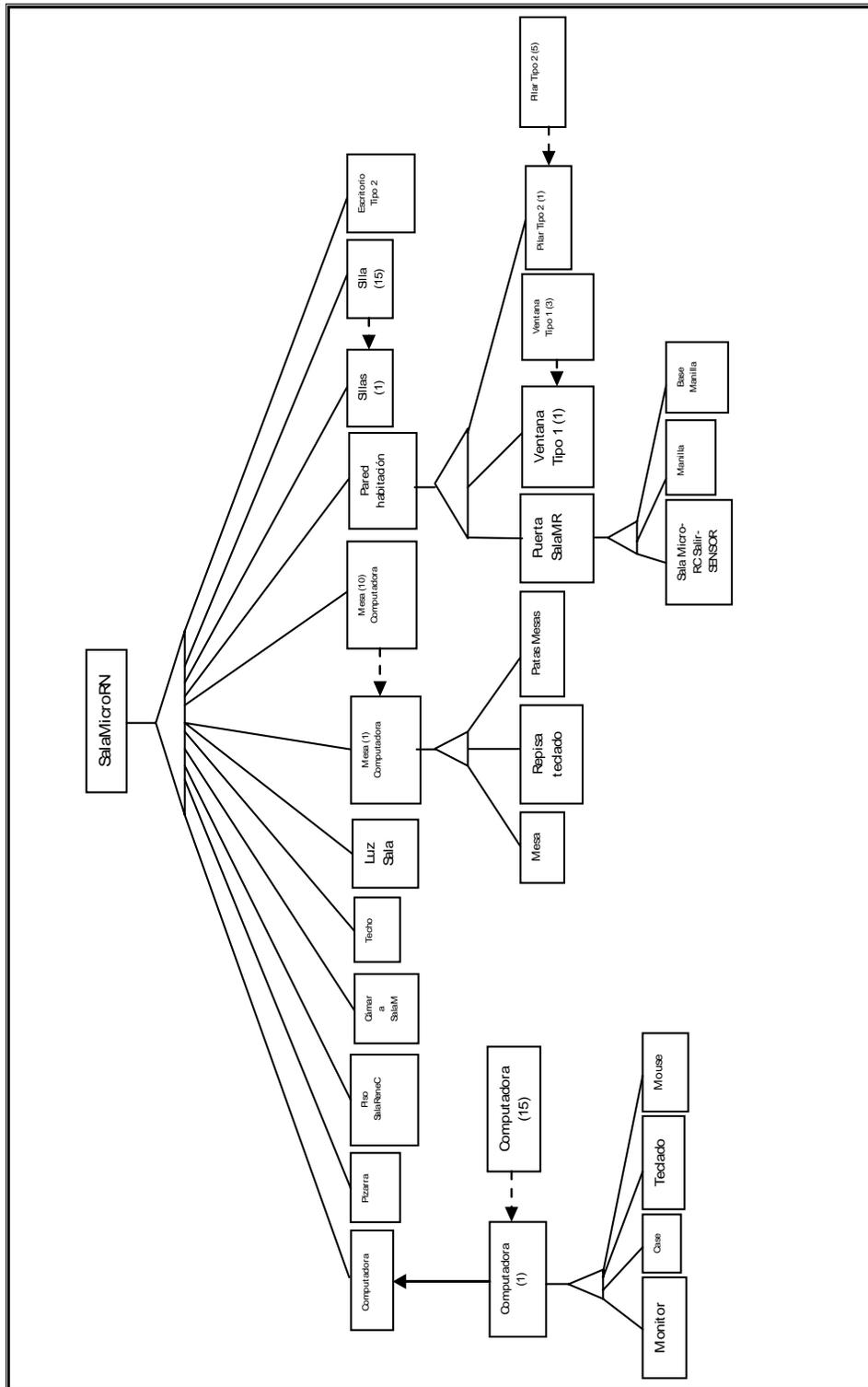


Figura 4.14 Diagrama de Ensamblaje del Nodo SalaMicroRC

4.5.2 Diagrama de Clase de Diseño

Siguiendo con el planteamiento de UML, el diagrama de clase representa la estructura estática del software, representando cada una de las clases que intervienen en cada proceso, sus atributos y relaciones, proporcionando la funcionalidad del sistema. Como lo hemos mencionado anteriormente el VRML, es un lenguaje modelado donde las declaraciones de las clase no están asequibles al programador. Están implícitamente en el lenguaje a si como las funciones que interpretan la ejecución. En la figura 4.15 se muestra el diagrama de clase de AVUDONA_2, donde existe una relación entre una estructuras principal y las demás, cabe destacar que todas descripción física del ambiente virtual se encuentra en la clase principal.

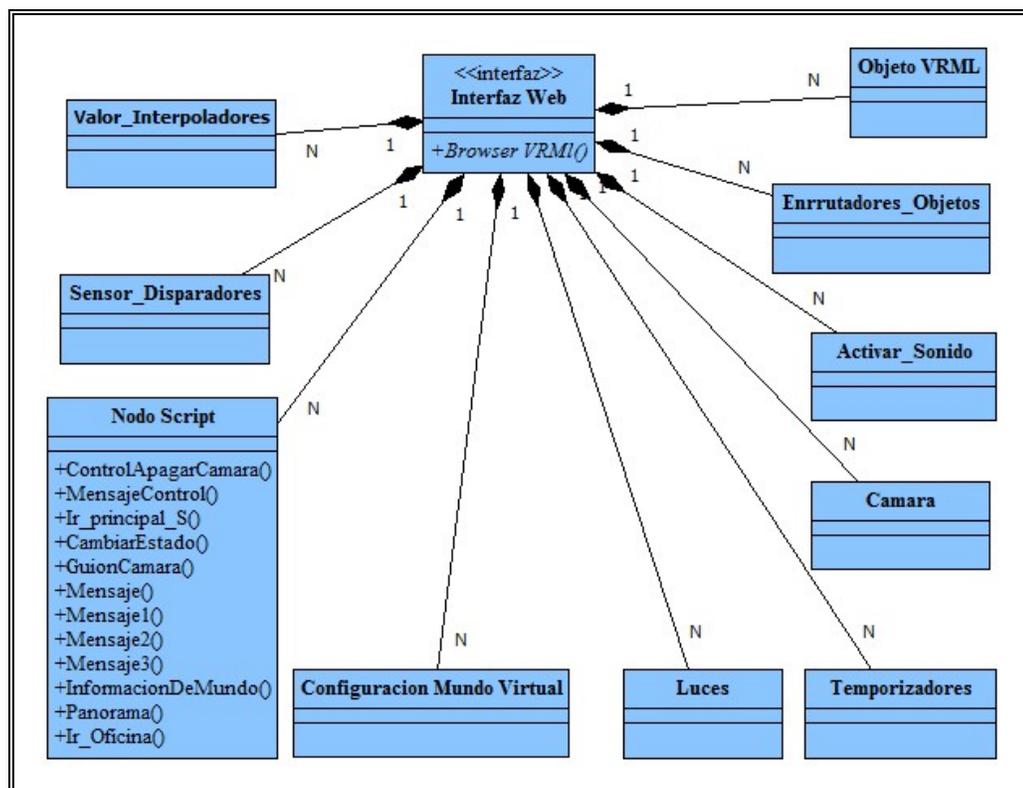


Figura 4.15 Diagrama de Clase de AVUDONA_2

4.5.2.1 Descripción de las Clases de Diseño AVUDONA_2

- **Clase Interfaz Web:** representa la interfaz que permite al usuario activar el Ambiente virtual cumple con la función de ejecutar todo el procedimiento de interpretar y visualizar el Ambiente Virtual descrito con la unión de todas las instancias de las demás clases.

- **Clase Activar_Sonido:** representa los objetos en los que se definen los nodos de sonidos que serán utilizado en las animaciones del ambiente virtual.

- **Clase Valor_Interpoladores:** Objetos que contiene todas las coordenadas y valores necesario para realizar las modificaciones en los atributos de las escenas y permitan la animaciones de los objetos, tales como coordenadas de translación y rotación.

- **Clase Temporizadores:** representa el grupo de objetos que permiten el control en la ejecución de las animaciones. Actúa como motor en el movimiento de los objetos 3D, a través un pulso de tiempo establecido para cada objeto, permitiendo las condiciones de cambios.

- **Clase Enrutadores_Objeto:** esta clase permiten establecer las rutas que controlan los eventos entre los objetos que componen las animaciones.

- **Clase Sensor_Disparadores:** conjunto de nodos que permiten activar los eventos en las escenas. Esto actúa como sensores que vigilan la acción del usuario dentro del ambiente virtual.

- **Clase Nodo Script:** representa el grupo de nodos que permite que el ambiente virtual posea una especie de interacción o variantes de las escenas. Estas condiciones incluyen una serie de rutinas escritas en lenguajes Script el cual permite incorporar en el VRML una programación lógica compleja.

- **Clase Objetos VRML:** representa la diversidad de figuras geométricas en el código VRML utilizada para construir todos los objetos 3D incluidos en el Ambiente Virtual.

- **Clase Luces:** representa la descripción de los efectos de luz y sus características, utilizados para darle iluminación a todos las escenas del ambiente virtual.

- **Clase Configuración del Mundo virtual:** permite declarar los valores de los parámetros modificables de la descripción del AV. Son los valores generales por defecto que contiene los Ambientes Virtuales.

- **Clase Cámaras:** son un conjunto de objetos especiales que representan una perspectiva de visión para el usuario dentro del Ambiente Virtual.

4.6 DIAGRAMA DE SECUENCIAS

Los Diagramas de Secuencia son conocidos como Diagramas de Interacción dado que muestran la interacción que se establece en un conjunto de objetos y sus relaciones, además de los mensajes que estos se envían. Describen como interactúa el usuario con los objetos y las escenas del ambiente virtual, las instancias del usuario

para generar las animaciones y permitir al usuario percibir la sensación de estar recorriendo el ambiente virtual.

Aquí lo importante es encontrar secuencias de interacciones ordenadas en el tiempo de carácter dinámico que se puedan presentar en un determinado escenario VRML. A continuación se muestran los Diagramas de Secuencia más importantes del AVUDONA_2.

El **Recorrido Virtual** se describe como el proceso que permite la sensibilidad del usuario dentro del entorno virtual cuando se encuentra navegando. En la figura 4.16 se muestra las acciones al desplazarse físicamente por el Ambiente Virtual.

Una vez que el usuario carga el ambiente virtual, la clase interfaz web modela la perspectiva de la escena que se está observando, cada vez que se desplazarse utilizando los comandos del navegador, dicha clase solicita el cálculo de la nueva perspectiva según el desplazamiento del usuario dentro del ambiente virtual y actualiza la información de cada objetos (luces, objetos 3D y cámaras) que conforma la escena, de esta forma simular el caminar del usuario dentro del mundo virtual.

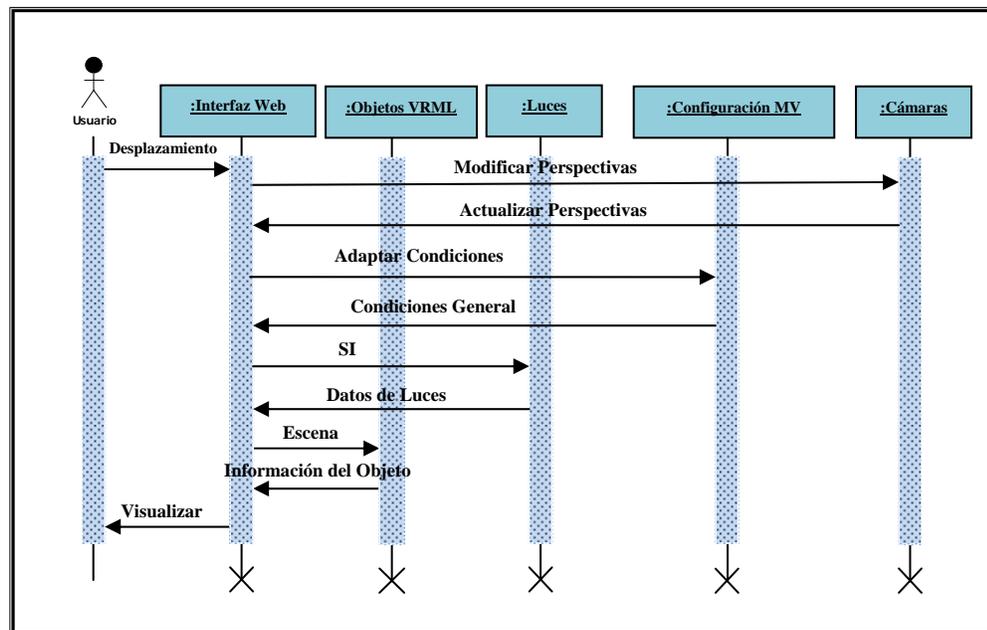


Figura 4.16 Diagrama de Secuencia del Recorrido Virtual

Dentro de grupo de **Animaciones** en el ambiente virtual existen dos tipos de animaciones, la cual, el usuario puede interactuar con las mismas. **Tipo 1**, el usuario interactúa con los objetos activando el evento al detectar su presencia en la escena, en la figura 4.17 se muestra dicho diagrama de secuencia. Una vez detectada la presencia del usuario el objeto *Sensor_Disparadores* recibe el evento y generan una señal *evento activado*, el cual mediante un objeto del tipo *Enrutadores_Objeto* es direccionado al objeto del tipo *Nodo Script* correspondiente y al nodo de *Activar_Sonidos* para efecto de Audio al comenzar la animación. Este nodo es el que determina según las condiciones establecidas en el *Nodo Script* si comienza o no la animación. Una vez validada las condiciones este envía todas las señales o valores (*Star*) correspondientes para iniciar la animación. En donde el objeto *Temporizadores* recibe la señal a través del objeto *Enrutadores_Objeto* y comienza a generar los valores consecutivos cada intervalo de tiempo y enviados al objeto *Valor_Interpoladores*, creando un ciclo de valores correspondiente a cada atributo del *Objeto VRML* incluidos en la animación. En todo este mecanismo el objeto tipo

Enrutadores_Objeto es quien encamina todos los eventos que se involucran en esta animación. Este proceso terminará cuando el objeto del tipo *Temporizadores* termine de generar el intervalo de valores.

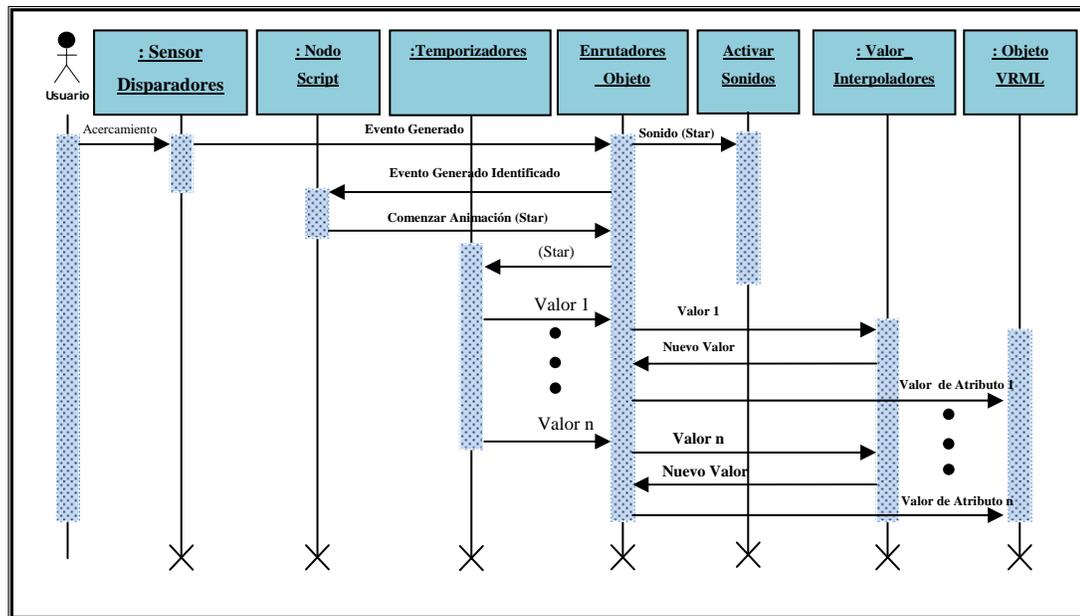


Figura 4.17 Diagrama de Secuencia de Animación, interacción con el usuario.

Existe otra que a diferencia de la animación **Tipo 1**, esta se inicia cuando el usuario realiza el recorrido se acerca a la escena de la animación y la activa a través de un evento externo (un click), el mecanismo de control es exactamente igual, no es necesario representar gráficamente el diagrama secuencial ya que la figura 4.18 anterior queda demostrado la acción del mecanismo que hace posible los cambios dinámicos que permite al usuario percibir la dicha animación.

Los siguientes **Tipo 2** de animación son simples, ya que son estrictamente controladas por el sistema a través de eventos internos que gestiona el funcionamiento del ambiente virtual. En la figura 4.18 podemos observar el diagrama que representa la animación tipo 2.

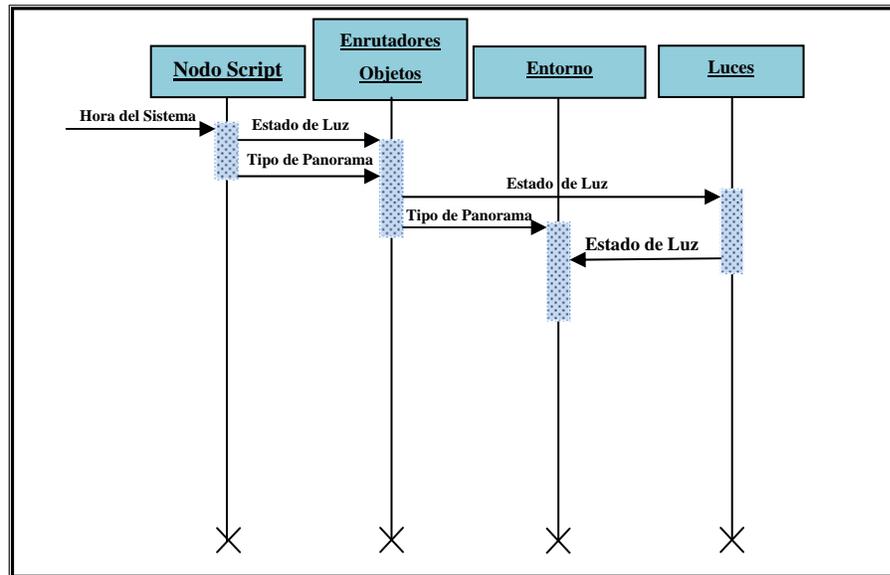


Figura 4.18 Diagrama de Secuencias de Animación, Interacción con el Sistema

Una vez cargado el ambiente virtual el sistema establece la hora y el *Nodo Script* determina, uno, el *Estado de Luces* (Apagadas o Encendidas) y, dos, el paisaje que rodea el ambiente virtual (Dia o Noche), enviando esta información al objeto *Enrutadores_Objeto* que es el que direcciona los valores a sus respectivos objetos para obtener los atributos necesario y cambiar estado en el *Entorno* del ambiente.

4.7 ARQUITECTURA LÓGICA DE AVUDONA_2

En la figura 4.19 se muestra la arquitectura del sistema AVUDONA_2 utilizando diagrama de capas y está compuesta en cuatro que son:

- Capa Específica de la Aplicación.

- Capa General de la Aplicación.
- Capa Intermedia.
- Capa del Software del Sistema.

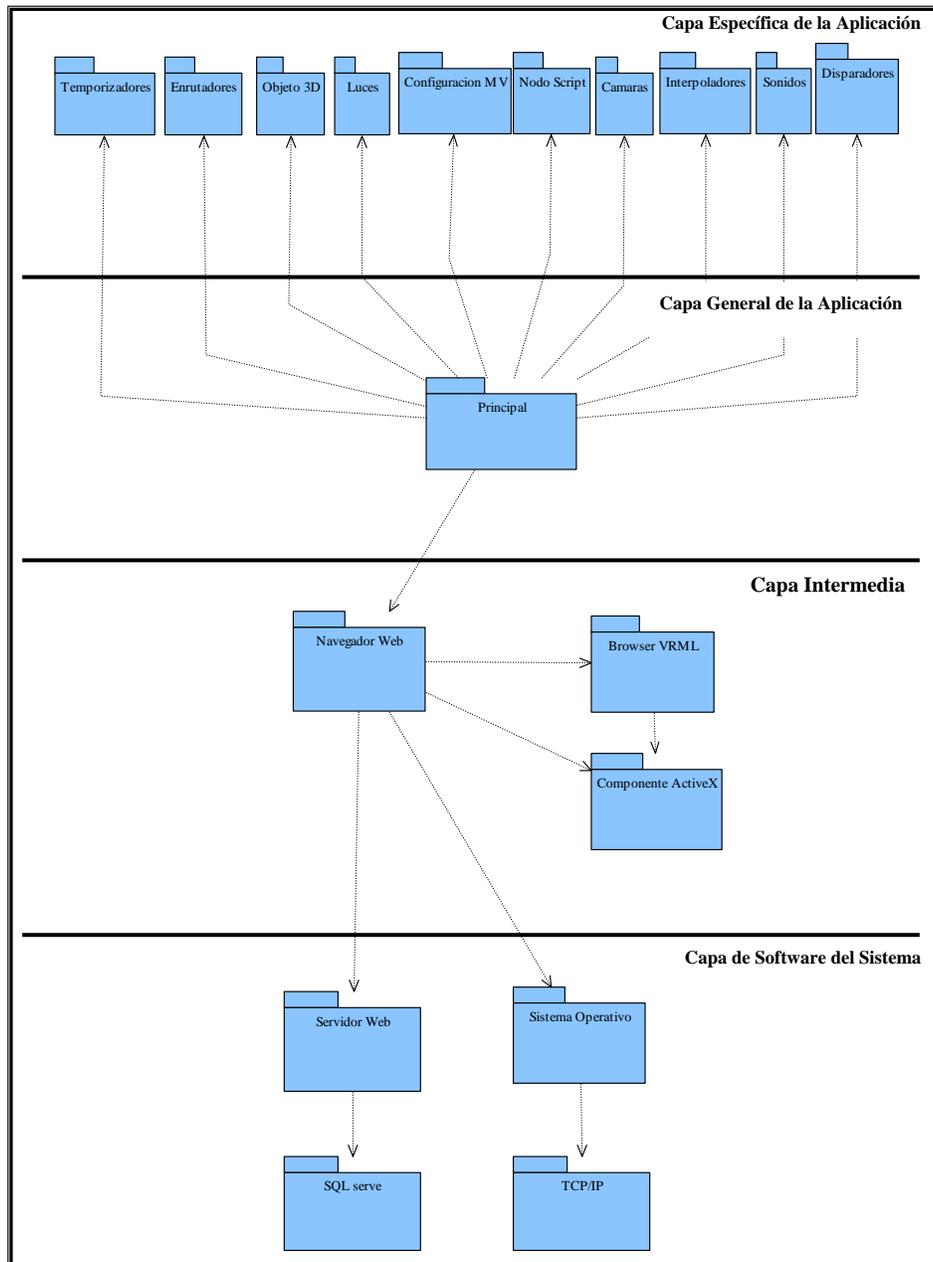


Figura 4.19 Arquitectura Lógica de AVUDONA_2

4.8 DISEÑO DEL SISTEMA DE ALMACENAMIENTO DE AVUDONA_2 (BASE DE DATOS).

Si hay algo por demás de potente dentro del ámbito de la aplicación computacional es el manejo de información a través de base de datos. Las bases de datos son estructuras lógicas que agrupan grandes cantidades de información de manera que esta pueda ser accedida fácilmente y con fines específicos.

Para realizar el diseño de la base de datos que sustente el acceso al mundo virtual, se utilizo el modelo relacional, el cual, permite una clara estructuración de la información, correcta presentación de las relaciones, manejo apropiado y fácil al usuario final, permitiendo consultar la información específica de forma sencilla.

Para la aplicación de la base de datos para AVUDONA_2, utilizamos un método, el cual, almacenamos la ruta de acceso al directorio donde está ubicado el archivo (.wrl) en la base de datos de esta forma sea más relajada y funcionara mejor, respecto a lo que significa implementar una base de dato que almacene los objeto de la escenas del mundo virtual.

Durante el proceso de definición de la estructura de las tablas, se encontraron 3 tablas que conforman el sistema de datos relacionadas entre sí, almacenando datos relacionado con la ruta de acceso a los archivos del ambiente virtual. En la notación del modelo relacional, los campos clave de las tablas o relaciones son señalados mediante un subrayado.

Tabla Ambiente_Virtual. Contiene la información acerca de la ruta de acceso al directorio donde se ubica el archivo AVUDONA_25_Actualizacion.wrl, estas

compuesta por un campo ***Id_AmbienteVirtual*** almacena información del índice de la tabla es un valor tipo entero y es generado por el SMBD, el campo ***nombre Archivo*** contiene los datos referente al nombre del Archivo que pertenece a la ruta de acceso almacenada es la clave principal de la tabla y es un valor tipo carácter, y el campo ***Ruta_Acceso*** almacena los datos referente a la ruta de acceso al archivo (.wrl) del mundo virtual es un dato de tipo carácter largo.

Esquema:

Ambiente_Virtual: (Id_AmbienteVirtual: int, nombre Archivo: char (25), Ruta_Acceso: varchar (25))

Tabla Imagen. Contiene la información acerca de las rutas de acceso al directorio donde se ubica las imágenes utilizadas como texturas en al ambiente virtual está compuesta por un campo ***Id_imagen*** es el índice de la tabla es un valor tipo entero y es generado por el SMBD, ***Id_AmbienteVirtual*** es la clave foránea que permite establecer la relación con la tabla Ambiente Virtual, ***nombre imagen*** contiene los datos referente al nombre de la imagen es la clave principal y es tipo carácter, ***rutaAcceso_Imagen*** este campo contiene los daros referente a la ruta de acceso al directorio del archivo imagen que son utilizado en el mundo virtual es dato de tipo carácter.

Esquema:

Imagen: (Id_imagen: int, Id_AmbienteVirtual, nombre imagen: char (25), rutaAcceso_Imagen: char (25))

Tabla Sonido. Contiene la información acerca de la ruta de acceso al directorio donde se ubica los archivos de audio utilizado para reproducir sonidos en el ambiente

virtual. Está compuesta por un campo *Id_sonido* es el índice de la tabla es un dato tipo entero y es generado por el SMBD, el campo *Id_AmbienteVirtual* es la clave foránea del la tabla, el campo *nombre_sonido* que contiene los datos referente al nombre del archivo de audio es de tipo carácter, el campo *rutaAcceso_sonido* se almacena los datos que hacen referencia a la ruta de acceso al directorio donde se encuentra los archivos de audio y los datos son de tipo carácter.

Esquema:

Sonido: (Id_sonido: int, Id_AmbienteVirtual, nombre_sonido: char (25), rutaAcceso_sonido: char (25)).

En la Figura 4.20, se puede observar el modelo Conceptual de la Base de Datos para la Actualización Ambiente Virtual Prototipo del Departamento de Computación y Sistema de la Universidad de Oriente Núcleo de Anzoátegui (AVUDONA_2).

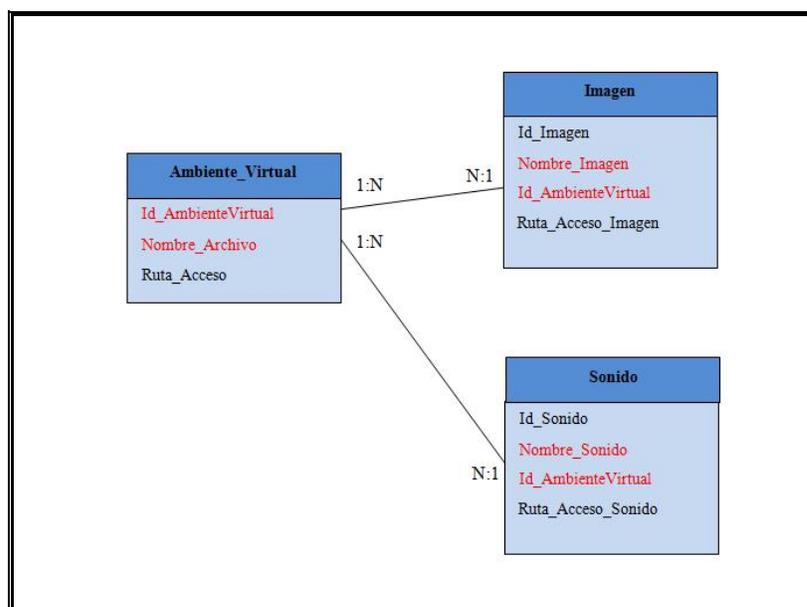


Figura 4.20 Modelo Conceptual de la Base de Datos del AVUDONA_2

CAPITULO V – FASE DE CONSTRUCCIÓN

5.1 INTRODUCCIÓN

Continuando con la metodología de Proceso unificado esta fase de construcción se procede a desarrollar todos los componentes del software que hará operativa cada caso de uso realizado en la fase de elaboración y de inicio, así reflejar la versión beta y posteriormente llegar a la versión final e incremento del software.

5.2 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN

El objetivo en esta fase es obtener una solución de software operativa, para lograr el desarrollo análogo del proyecto, enfocado a la codificación de AVUDONA_2. Para llevar a cabo la construcción del ambiente virtual, una vez, hecho con cuidado los pasos anteriores realizados en la fase de inicio y elaboración, proseguimos con el modelado en base a las especificaciones entregadas, comprendiendo y modelando los fenómenos de dependencia de movimientos y organización de los componentes en forma jerárquica presentes en el desarrollo del proyecto de realidad virtual. Dicho esto, reimplementamos unas series de etapas a seguir para lograr la codificación del ambiente virtual:

Con la Herramienta *3D Studio MAX* se realizan las primeras cinco etapas.

- ✦ Construcción de objetos de manera individual.

- Construcción de las zonas a partir de los objetos creados.
- Aplicación de los mapas de texturas a los objetos.
- Incorporación de luces para efectos de profundidad y sombras.
- Realización de la conversión de las zonas a código VRML.

A partir de la siguiente etapa se trabaja directamente con código **VRML** utilizando *VrmlPad 2.1*.

- Utilizar Vrmlpad 2.1 para aplicación de técnicas de depuración al Código VRML resultante para evitar la pérdida de detalles en la conversión.
- Incorporación de la animación a los objetos cuyo comportamiento es constante.
- Incorporación de la animación a los objetos cuyo comportamiento depende de la interacción con el usuario o cualquier otro factor externo variante.
- Inclusión de los efectos sonoros.
- Acoplar las actualizaciones y nuevas zonas con AVDONA_1.
- Inclusión de las guías de navegación interactivas para los usuarios.
- Definición de la forma de desplazamiento de los Usuarios en el Ambiente Virtual.
- Establecimiento de las características del clima (Cielo, Piso, Niebla).
- Regulación de la rapidez de desplazamiento de los Usuarios por el Ambiente Virtual.

5.3 CODIFICACIÓN DE AVUDONA_2.

Digitación de los objetos utilizando la herramienta 3D Studio MAX se realiza de la siguiente manera:

5.3.1 Construcción de Objetos de Manera Individual.

Para la construcción de los correspondientes objetos de la lista definida en el capítulo anterior, utilizamos el programa de modelado 3D Studio MAX. En primer lugar importamos los planos 2D de la Zona a la que corresponde el objeto a modelar, a través de la opción que ofrece 3D Studio MAX de importar archivos en formatos AutoCad, para así obtener las dimensiones exactas y la ubicación del objeto en la Zona. En la figura 5.1 se puede observa los planos 2D importados.

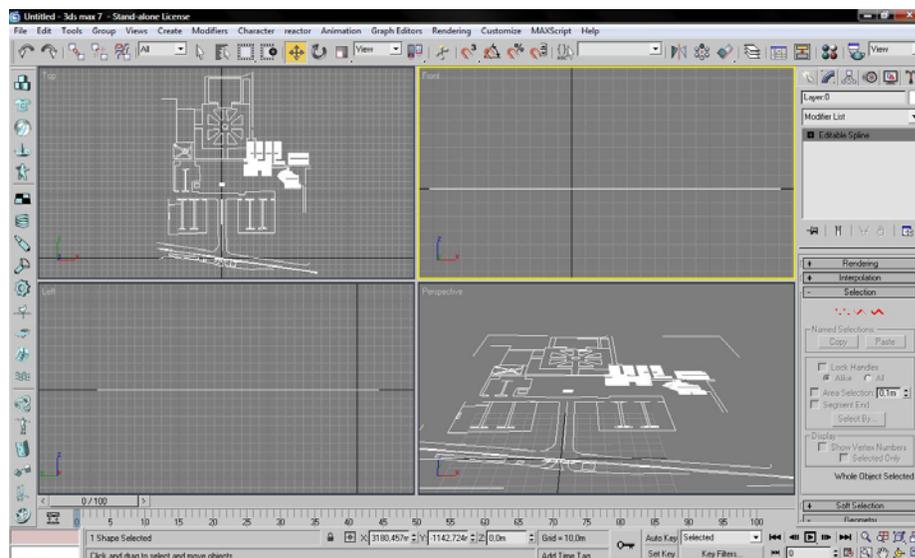


Figura 5.1 Importación del Planos 2D a la Herramienta 3D Studio MAX.

Obtenido los planos 2D se procede a la construcción del objeto como tal, utilizando las distintas herramientas que ofrece 3D Studio MAX. Dependiendo de la naturaleza del objeto y la complejidad para modela se utilizaron estándares primitivos (Cubo, Esferas, rectángulos, etc...), extendido como (Polígonos, etc) y objetos compuestos (operados boolean, lotf, terrain), window, door's, splines (objetos bidimensionales círculos, líneas, etc.), modificadores (extrude, venit, etc.), lights, cameras, editor de texturas y control de animaciones, etc.

En la siguiente figura 5.2 se muestra la construcción del objeto “Símbolo de Concreto” centro decorativo de la plaza del Decanato.

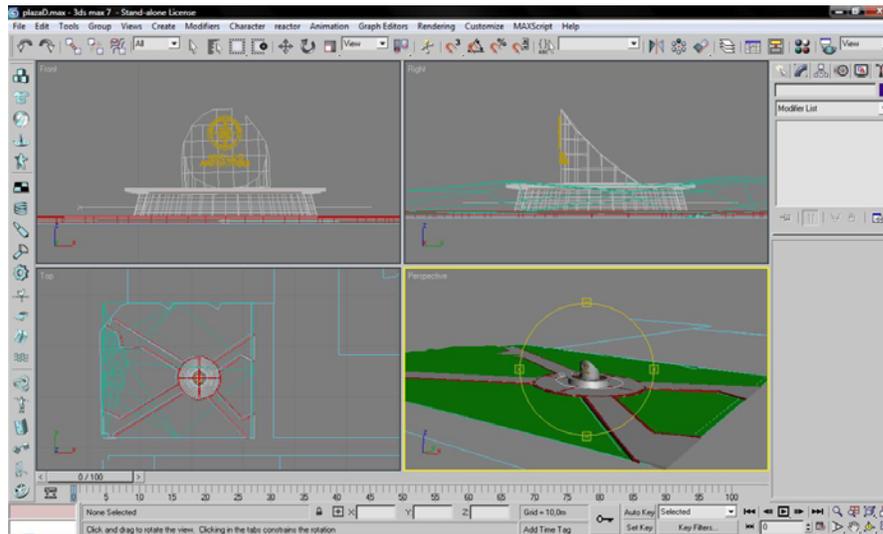


Figura 5.2 Construcción del Objeto “Símbolo de Concreto” Plaza Decanato

De igual manera se construyo de manera individual el siguiente objeto “Edificio Decanato” ilustrado en la figura 5.3 a continuación.

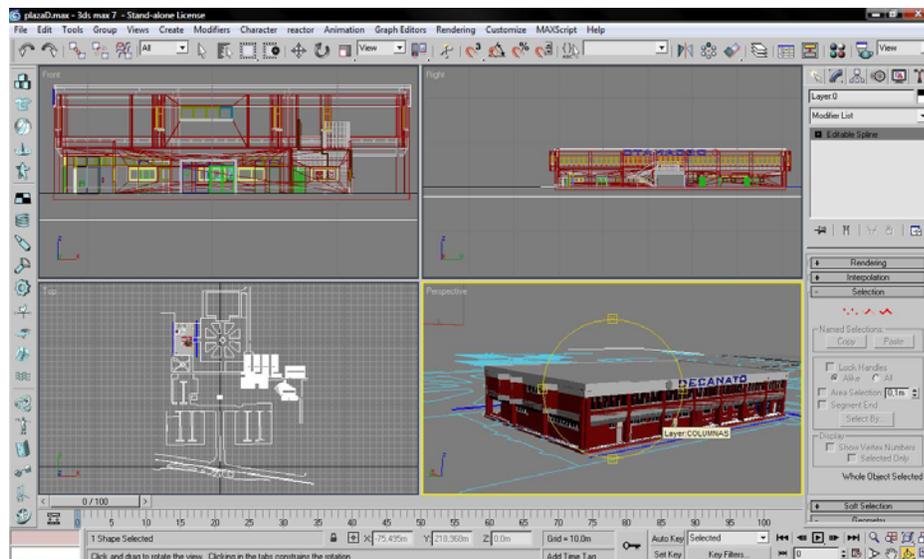


Figura 5.3 Construcción del Objeto “Edificio Decanato”

Siguiendo en la construcción individual de cada objeto, realizamos actualizaciones en aquellas escenas que sufrieron cambio en su homónimo real tal como el “Portón” de la entrada principal de la UDO, ilustrado en la figura 5.4 a continuación.

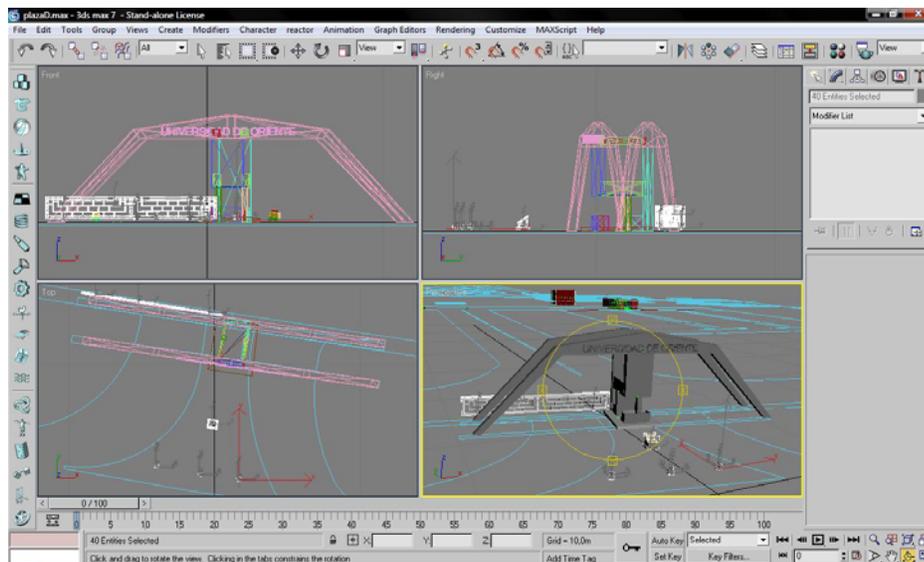


Figura 5.4 Actualización y Construcción del “Portón” de la Entrada Principal

5.3.2 Construcción de las Zonas a Partir de los Objetos Creados.

Una vez terminado la construcción de todos y cada unos de los objetos que van a conformar el recorrido virtual, procedimos a incorporar progresivamente a cada objeto en su respectiva ubicación. Sabemos que el ambiente virtual fue dividido relativamente en zonas, y lo que hemos logrado obtener hasta el momento es una cantidad de objetos en sus posiciones correspondientes.

Partiendo de una escena y una de las nuevas zonas cualquiera comenzamos a incluir las diferentes escenas hasta completar la zona en cuestión. Esto se realizara de igual manera para cada unas de las diferentes zonas del recorrido, sean nuevas escenas o actualizaciones.

Utilizando la opción *Meger*, que ofrece 3D Studio MAX se logro cumplir este objetivo. En la figura 5.5 podemos observa la zona 5 ya completamente formada.

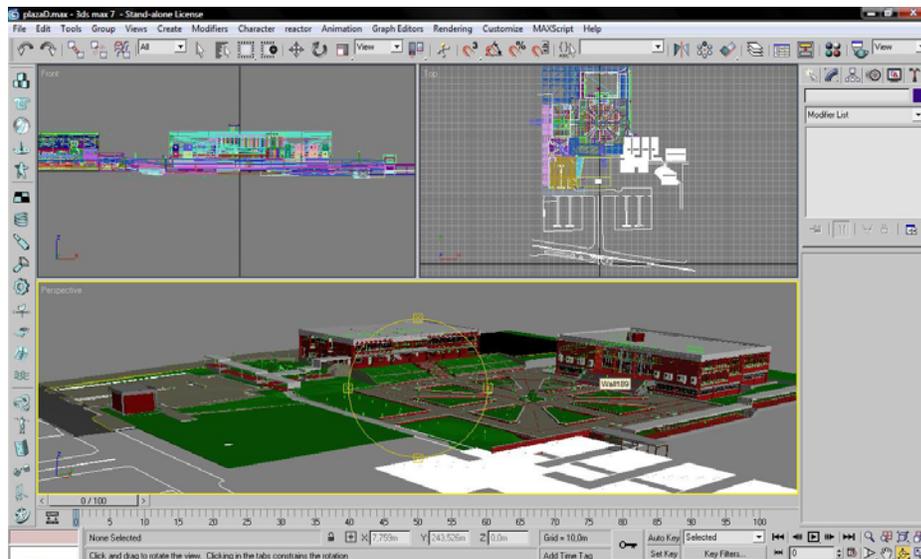


Figura 5.5 Zona 5 del Ambiente Virtual AVUDONA_2.

5.3.3 Aplicación de los Mapas de Texturas a los Objetos.

El uso de texturas aplicadas en los objetos, permite dar un aproximado realismo en algunas de las escenas del ambiente virtual. 3D Studio MAX nos ofreció una poderosa herramienta que nos permite editar, crear y aplicar textura a los objetos, llamada “*Material Editor*” o editor de material, de tal forma, usamos la misma para

aplicar la textura ladrillo al objeto “Pared” y “Mesas de Computadoras” de la SALA “*RENE CABRERA*”. Obteniendo la escena ilustrada en la figura 5.6.

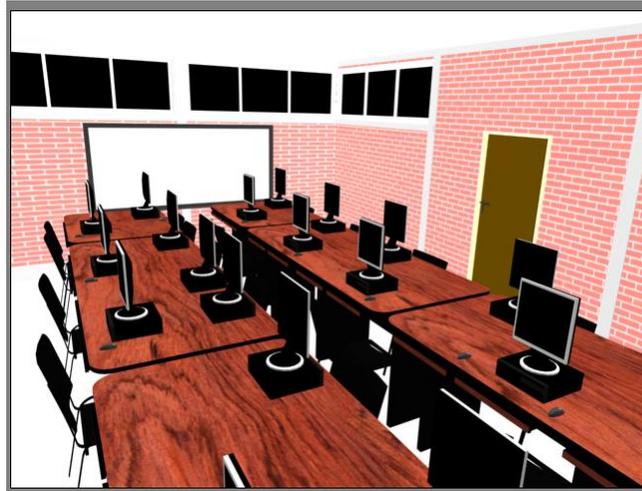


Figura 5.6 Aplicación de Textura Ladrillo a la Pared Interior de la Sala “*RENE CABRERA*”

5.3.4 Incorporación de Luces Para Efectos de Profundidad y Sombras

Aplicar el efecto de luz en el ambiente virtual permite definir la iluminación de las escenas del ambiente virtual, el cual es lógico. La herramienta de diseño 3D Studio MAX dispone de varios tipo de luces entre ellas tenemos: Omni, Target Spot, Target Direct, Free Spot, Free Direct, Sunlight y Ambient. Para el caso de AVUDONA_2 se utilizo dos tipos de luces, Omni para el efecto de la luz ambiente (Sol) y Target Spot para simular la luz de los reflectores de la entrada de la UDO. En la figura 5.7 se observa cómo fueron definidas las luces de la entrada de la UDO para lograr el efecto deseado.

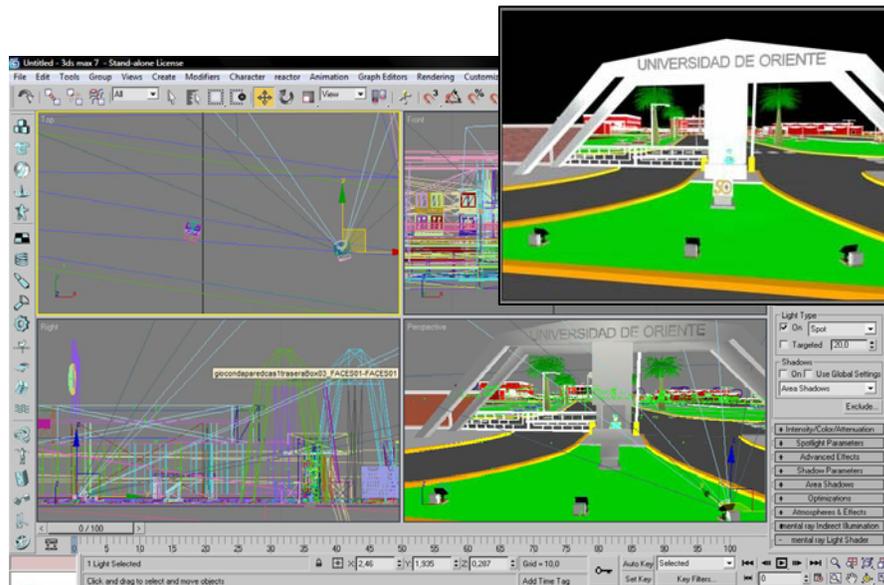


Figura 5.7 Efecto Luz Reflectores Entrada Universidad de Oriente.

5.3.5 Realización de la Conversión de las Zonas a Código VRML.

Hasta el momento se ha trabajado modelando todos los objetos 3D de cada una de las zonas. Partiendo de este punto con la ayuda de las herramientas que ofrece *3D Studio MAX*, que permite a partir de un formato estándar (ej. .max, .dxf, .3ds), transformar el archivo a VRML directamente, sin necesidad de ningún paso adicional. Realizado el proceso de conversión es preciso aplicar las herramientas necesarias para corregir cualquier dificultad que surja en el proceso de conversión. El resultado ideal sería aquel en donde los objetos pasarán herramienta por herramienta sin ningún cambio perceptible. *VrmlPad* es la herramienta que permite visualizar los archivos VRML (.wrl). A Continuación se muestra parte del código resultante de la conversión de la zona 5:

```

#VRML V2.0 utf8
# Produced by 3D Studio MAX VRML97 exporter, Version 7, Revision 0,65
# MAX File: AVUDONA_2_ACTUALIZACION, Date: Sat Nov 01 19:55:55 2008

DEF info WorldInfo {
  title "AVUDONA 2 VRML97 "
  info ["Trabajo de Grado Universidad de Oriente:"
    "ACTUALIZACION Y EXPACION DEL AMBIENTE VIRTUAL PROTOTIPO DE"
    "LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI"]
}

DEF Zona6Cerca Group {
  children [
    DEF BoxTecho Transform {
      translation 131.5 -0.7812 -116.9
      children [
        Transform {
          translation 0 0.2176 0
          children [
            Shape {
              appearance Appearance {
                material Material {
                  diffuseColor 0.8902 0.8863 0.8902
                  ambientIntensity 0.2
                  specularColor 0 0 0
                  shininess 0.06615
                  transparency 0
                }
              }
              geometry Box { size 0.1 0.4352 9.664 }
            }
          ]
        }
      ]
    }
  ]
}

DEF V2Grp06 Transform {
  translation 101.5 -2.027 -121.4
  rotation -1 0 0 -1.571
  children [
    DEF Wall01 Transform {
      translation 2.744 -24.9 0
      children [
        DEF Wall01_FACES01 Transform {
          translation 0 0 0
          children [
            Shape {
              appearance Appearance {
                material Material {
                  diffuseColor 0.5255 0.02353 0.02353
                  ambientIntensity 0.2
                  specularColor 0 0 0
                  shininess 0.06615
                  transparency 0
                }
              }
            }
          ]
        }
      ]
    }
  ]
}

```

```

    }
  }
+  geometry DEF Wall01_FACES01-FACES IndexedFaceSet {
    }
  ]
}
},
DEF Wall02 Transform {
  translation 13.36 -3.961 0
  children [
    DEF Wall02_FACES01 Transform {
      translation 0 0 0
      children [
+     Shape {
    }
  ]
}
},
DEF Wall04 Transform {
  translation 13.46 -3.824 -0.04383
  children [
    DEF Wall04_FACES01 Transform {
      translation 0 0 0
      children [
+     Shape {
    }
  ]
}
},
DEF Wall04 Transform {
  translation 13.46 -3.824 -0.04383
  children [
    DEF Wall04_FACES01 Transform {
      translation 0 0 0
      children [
+     Shape {
    }
  ]
}
},
DEF Wall05 Transform {
  translation 2.744 -24.9 1.383
  children [
    DEF Wall05_FACES01 Transform {
      translation 0 0 0
      children [

```

```

+     Shape {
        }
    ]
}
]
},
DEF Wall07 Transform {
    translation 2.814 -24.96 -3.336
    children [
        DEF Wall07_FACES01 Transform {
            translation 0 0 0
            children [
+         Shape {
                }
            ]
        }
    ]
}
DEF Wall08 Transform {
    translation 3.585 -27.22 0
    children [
        DEF Wall09_FACES01 Transform {
            }
        ]
    ]
},
DEF Wall09 Transform {
},
DEF Wall10 Transform {
},
DEF Wall11 Transform {
},
DEF Wall12 Transform {
},
DEF Wall13 Transform {
},
DEF Wall14 Transform {
},
DEF Wall15 Transform {
},
DEF Wall16 Transform {
},
DEF Wall17 Transform {
},
DEF Wall18 Transform {
},
DEF Wall20 Transform {
},
DEF Wall21 Transform {
},

```

```

DEF Box08 Transform {
},
DEF Box09 Transform {
},
DEF Box10 Transform {
  translation 20.35 -19.9 0
  rotation 0 0 -1 -0.672
  children [
    DEF Box03_FACES03 Transform {
      translation 0 0 0
      children [
        Shape {
        }
      ]
    }
  ]
},

```

A partir de la siguiente etapa se trabaja directamente con código **VRML** utilizando *VrmlPad 2.1*.

5.3.6 Utilizar Vrmlpad 2.1 Para Aplicación de Técnicas de Depuración al Código Vrml Resultante Para Evitar la Pérdida de Detalles en la Conversión.

Para el desarrollo de esta etapa es necesario tener previo conocimiento del lenguaje VRML 2.0, ya que, desde aquí se comienza a edita el código resultante de la exportación realizada en 3D Studio Max. El proceso de optimización se llevara a cabo con el VrmlPad 2.1 que es un editor de programación VRML, cabe destacar que, VrmlPad es una herramienta que incluye funciones de edición visual, permite identificar errores, realizar compilación del código y ejecutar proceso de depuración, habilidades y apoyo a la escena con el árbol de recursos, y operaciones ahorro de

tiempo. Retomando el proceso aplicado en el proyecto anterior procedemos a optimizar el código siguiendo los siguientes pasos:

Verificar que en la conversión no haya ocurrido algún error que cause que algún objeto de la escena no se visualice o sea distinto al creado en 3D Studio Max.

Puede ser que al exportar algún objeto no quede en su lugar, o algunas de sus partes. Este es un error común en la exportación de 3D Studio Max. Aunque este proceso, el de crear en 3D Studio exportar y ver si es lo que queremos en VRML, es recomendable irlo haciendo cada vez que realizamos algún objeto, también es necesario en el proceso de integración de las zonas y al final cuando se completa el recorrido virtual. Esto se debe a que a veces al incluir nuevos objetos puede existir algún desajuste entre su relación de tamaño, color, etc.

Reutilizar el código usando el nodo DEF/USE, creando los llamados arreglos de objetos para los que se repiten muchas veces dentro de las escenas. La instanciación que realiza 3D Studio Max es muy limitada y en realidad no reutiliza por completo la definición de un objeto. Claro, hacemos énfasis en los objetos que tienen una definición extensa y ocupa un espacio significativo. En el Ambiente Virtual desarrollado se realizaron arreglos de los Postes de luz y lámparas, las Palmeras de la entrada, Mesitas de la entrada, sillas de sala de micro y de navegación y pilares de pasillos.

Aplicar el nivel de detalles (LOD) a los nodos con diseño complejos para aumentar el rendimiento del recorrido.

Utilizar el nodo LOD (nivel de detalles) cuando el recorrido es extenso y existen estructuras con cierta complejidad de detalles que no se perciben a distancia es muy útil para disminuir el retardo de procesamiento, utilizando texturas que represente el detalle y nodos que se transpongan a medida que el usuario se acerca.

Sin embargo, la realización de varios nodos de una misma cosa aumenta la cantidad de código total y a su vez el tamaño de ficheros. Para evitar esta contradicción, tomamos en cuentas varias condiciones:

- Aquellas zonas que durante el recorrido no son visible. A medida que el usuario se acerca van apareciendo en la escena.
- Zonas que son visible a distancia y no es necesario un completo detalle y van incrementando a medida que el usuario se acerca.
- Escenas visibles a distancia y se utilizan textura para mostrar los detalles, a medida que se acerca el usuario se transpone la escena.

En tabla 5.1 se nombran las escenas del Ambiente Virtual donde fue implementada la herramienta LOD.

Tabla 5.1 Escena con LOD (Nivel de Detalle).

Escena	Escena Nivel Bajo	Escena Nivel Alto	Área de Detalle
Aulas de Ingeniería	Aulas_Lejos	Aulas_Cerca	80 unidades Métricas
Edif. Biblioteca	BiblioVirtual	Biblioteca	80 unidades Métricas
Edif. Decanato	DecaVirtual	Decanato	75 unidades Métricas
Zona 6	Zona6Lejos	Zona6Cerca	150 unidades Métricas
Zona 5	Zona5Lejos	Zona5Cerca	151 unidades Métricas

Para realizar esta optimización primero se tomó en cuenta que partes de la zona era capaz de observar mientras recorría desde la entrada hasta cierta distancia de ella. Se construyó un nodo llamado *_Lejos* que sólo muestra los detalles que percibe el usuario desde lejos de las escena y al nodo original detallado se le dio el nombre de *_Cerca*. En la sintaxis del nodo LOD se colocan los nodos secuencialmente y las distancias desde donde serán válidos. Pero los nodos pueden ser un archivo aparte o cualquier nodo de tipo Group. Se escogió utilizar un nodo Group en vez de colocar

los nodos aparte para evitar que el Browser muestre la escena restante mientras se cargue las aulas y afecte la inmersión del usuario cuando por ejemplo se sale de la sala de micros.

5.3.7 Incorporación de la Animación a los Objetos Cuyo Comportamiento es Constante.

Dentro de esta categoría está la animación Ayuda, representada por un objeto en forma de moneda con el signo de interrogación, la cual mantiene un movimiento de giro sobre un eje fijo. Este objeto permite que el usuario pueda tener acceso a la ayuda de cómo usarlos los controles de navegación. A continuación se muestra de cómo se define dicho nodo la declaración de coordenadas y sus interpoladores:

```

Anchor {
  description "Haga Click Para Obtener ayuda"
  parameter [ "target=main_frame" ]
  url "../Tesis/Diseño Tesis/Ayuda/AyudaMarcos.htm"
  children [
    DEF Ayuda Transform {
      translation 10.151 0.8539 -7.764
      scale 0.4261 0.4261 0.4261
      rotation -1 0 0 -1.571
      scale 0.7312 0.7312 0.7312
      children [
        Shape {
          appearance Appearance {
            material Material {
              diffuseColor 0.5882 0.5882 0.5882
              ambientIntensity 1.0

```

```

specularColor 0 0 0
shininess 0.145
transparency 0
}
texture ImageTexture {
    url "../Tesis/Diseño Tesis/Texturas/Ayuda.jpg"
}
}
geometry DEF Ayuda-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Ayuda-COORD Coordinate { point [
        0.6365 0 0, 0.6197 0 -0.1458, 0.5717 0 -0.2798, 0.4966 0 -0.398,
        0.398 0 -0.4966, 0.2798 0 -0.5717, 0.1458 0 -0.6197,
        0 0 -0.6365, -0.1458 0 -0.6197, -0.2798 0 -0.5717,
        -0.398 0 -0.4966, -0.4966 0 -0.398, -0.5717 0 -0.2798,
        -0.6197 0 -0.1458, -0.6365 0 0, -0.6197 0 0.1458, -0.5717 0 0.2798,
        -0.4966 0 0.398, -0.398 0 0.4966, -0.2798 0 0.5717,
        -0.1458 0 0.6197, 0 0 0.6365, 0.1458 0 0.6197, 0.2798 0 0.5717,
        0.398 0 0.4966, 0.4966 0 0.398, 0.5717 0 0.2798, 0.6197 0 0.1458,
        0.6365 0 0, 0.6197 0 -0.1458, 0.5717 0 -0.2798, 0.4966 0 -0.398,
        0.398 0 -0.4966, 0.2798 0 -0.5717, 0.1458 0 -0.6197,
        0 0 -0.6365, -0.1458 0 -0.6197, -0.2798 0 -0.5717,
        -0.398 0 -0.4966, -0.4966 0 -0.398, -0.5717 0 -0.2798,
        -0.6197 0 -0.1458, -0.6365 0 0, -0.6197 0 0.1458, -0.5717 0 0.2798,
        -0.4966 0 0.398, -0.398 0 0.4966, -0.2798 0 0.5717,
        -0.1458 0 0.6197, 0 0 0.6365, 0.1458 0 0.6197, 0.2798 0 0.5717,
        0.398 0 0.4966, 0.4966 0 0.398, 0.5717 0 0.2798, 0.6197 0 0.1458]
    }
    texCoord DEF Ayuda-TEXCOORD TextureCoordinate { point [
        0 0, 0.03679 0, 0.07243 0, 0.1075 0, 0.1425 0, 0.1776 0,
        0.2132 0, 0.25 0, 0.2868 0, 0.3224 0, 0.3575 0, 0.3925 0,
        0.4276 0, 0.4632 0, 0.5 0, 0.5368 0, 0.5724 0, 0.6075 0,
        0.6425 0, 0.6776 0, 0.7132 0, 0.75 0, 0.7868 0, 0.8224 0,

```

0.8575 0, 0.8925 0, 0.9276 0, 0.9632 0, 1 0, 0 1, 0.03679 1,
 0.07243 1, 0.1075 1, 0.1425 1, 0.1776 1, 0.2132 1,
 0.25 1, 0.2868 1, 0.3224 1, 0.3575 1, 0.3925 1, 0.4276 1,
 0.4632 1, 0.5 1, 0.5368 1, 0.5724 1, 0.6075 1, 0.6425 1,
 0.6776 1, 0.7132 1, 0.75 1, 0.7868 1, 0.8224 1, 0.8575 1,
 0.8925 1, 0.9276 1, 0.9632 1, 1 1, 1 0.5, 0.9868 0.6146,
 0.9491 0.7198, 0.8901 0.8126, 0.8126 0.8901, 0.7198 0.9491,
 0.6146 0.9868, 0.5 1, 0.3854 0.9868, 0.2802 0.9491,
 0.1874 0.8901, 0.1099 0.8126, 0.05086 0.7198, 0.01322 0.6146,
 0 0.5, 0.01322 0.3854, 0.05086 0.2802, 0.1099 0.1874,
 0.1874 0.1099, 0.2802 0.05086, 0.3854 0.01322, 0.5 0,
 0.6146 0.01322, 0.7198 0.05086, 0.8126 0.1099, 0.8901 0.1874,
 0.9491 0.2802, 0.9868 0.3854, 1 0.5, 0.9868 0.6146,
 0.9491 0.7198, 0.8901 0.8126, 0.8126 0.8901, 0.7198 0.9491,
 0.6146 0.9868, 0.5 1, 0.3854 0.9868, 0.2802 0.9491,
 0.1874 0.8901, 0.1099 0.8126, 0.05086 0.7198, 0.01322 0.6146,
 0 0.5, 0.01322 0.3854, 0.05086 0.2802, 0.1099 0.1874,
 0.1874 0.1099, 0.2802 0.05086, 0.3854 0.01322, 0.5 0,
 0.6146 0.01322, 0.7198 0.05086, 0.8126 0.1099, 0.8901 0.1874,
 0.9491 0.2802, 0.9868 0.3854]

}

coordIndex [

29, 28, 0, -1, 0, 1, 29, -1, 30, 29, 1, -1, 1, 2, 30, -1,
 31, 30, 2, -1, 2, 3, 31, -1, 32, 31, 3, -1, 3, 4, 32, -1,
 33, 32, 4, -1, 4, 5, 33, -1, 34, 33, 5, -1, 5, 6, 34, -1,
 35, 34, 6, -1, 6, 7, 35, -1, 36, 35, 7, -1, 7, 8, 36, -1,
 37, 36, 8, -1, 8, 9, 37, -1, 38, 37, 9, -1, 9, 10, 38, -1,
 39, 38, 10, -1, 10, 11, 39, -1, 40, 39, 11, -1, 11, 12, 40, -1,
 41, 40, 12, -1, 12, 13, 41, -1, 42, 41, 13, -1, 13, 14, 42, -1,
 43, 42, 14, -1, 14, 15, 43, -1, 44, 43, 15, -1, 15, 16, 44, -1,
 45, 44, 16, -1, 16, 17, 45, -1, 46, 45, 17, -1, 17, 18, 46, -1,
 47, 46, 18, -1, 18, 19, 47, -1, 48, 47, 19, -1, 19, 20, 48, -1,
 49, 48, 20, -1, 20, 21, 49, -1, 50, 49, 21, -1, 21, 22, 50, -1,
 51, 50, 22, -1, 22, 23, 51, -1, 52, 51, 23, -1, 23, 24, 52, -1,

53, 52, 24, -1, 24, 25, 53, -1, 54, 53, 25, -1, 25, 26, 54, -1,
 55, 54, 26, -1, 26, 27, 55, -1, 28, 55, 27, -1, 27, 0, 28, -1,
 17, 16, 15, -1, 17, 15, 14, -1, 17, 14, 13, -1, 17, 13, 12, -1,
 17, 12, 11, -1, 17, 11, 10, -1, 17, 10, 9, -1, 17, 9, 8, -1,
 17, 8, 7, -1, 17, 7, 6, -1, 17, 6, 5, -1, 17, 5, 4, -1,
 17, 4, 3, -1, 17, 3, 2, -1, 17, 2, 1, -1, 17, 1, 0, -1,
 17, 0, 27, -1, 17, 27, 26, -1, 17, 26, 25, -1, 17, 25, 24, -1,
 18, 17, 24, -1, 19, 18, 24, -1, 19, 24, 23, -1, 20, 19, 23, -1,
 20, 23, 22, -1, 20, 22, 21, -1, 48, 49, 50, -1, 48, 50, 51, -1,
 43, 44, 45, -1, 42, 43, 45, -1, 41, 42, 45, -1, 40, 41, 45, -1,
 39, 40, 45, -1, 38, 39, 45, -1, 37, 38, 45, -1, 36, 37, 45, -1,
 35, 36, 45, -1, 34, 35, 45, -1, 33, 34, 45, -1, 32, 33, 45, -1,
 31, 32, 45, -1, 30, 31, 45, -1, 29, 30, 45, -1, 28, 29, 45, -1,
 55, 28, 45, -1, 54, 55, 45, -1, 53, 54, 45, -1, 52, 53, 45, -1,
 52, 45, 46, -1, 52, 46, 47, -1, 51, 52, 47, -1, 48, 51, 47, -1]

texCoordIndex [

30, 29, 0, -1, 0, 1, 30, -1, 31, 30, 1, -1, 1, 2, 31, -1,
 32, 31, 2, -1, 2, 3, 32, -1, 33, 32, 3, -1, 3, 4, 33, -1,
 34, 33, 4, -1, 4, 5, 34, -1, 35, 34, 5, -1, 5, 6, 35, -1,
 36, 35, 6, -1, 6, 7, 36, -1, 37, 36, 7, -1, 7, 8, 37, -1,
 38, 37, 8, -1, 8, 9, 38, -1, 39, 38, 9, -1, 9, 10, 39, -1,
 40, 39, 10, -1, 10, 11, 40, -1, 41, 40, 11, -1, 11, 12, 41, -1,
 42, 41, 12, -1, 12, 13, 42, -1, 43, 42, 13, -1, 13, 14, 43, -1,
 44, 43, 14, -1, 14, 15, 44, -1, 45, 44, 15, -1, 15, 16, 45, -1,
 46, 45, 16, -1, 16, 17, 46, -1, 47, 46, 17, -1, 17, 18, 47, -1,
 48, 47, 18, -1, 18, 19, 48, -1, 49, 48, 19, -1, 19, 20, 49, -1,
 50, 49, 20, -1, 20, 21, 50, -1, 51, 50, 21, -1, 21, 22, 51, -1,
 52, 51, 22, -1, 22, 23, 52, -1, 53, 52, 23, -1, 23, 24, 53, -1,
 54, 53, 24, -1, 24, 25, 54, -1, 55, 54, 25, -1, 25, 26, 55, -1,
 56, 55, 26, -1, 26, 27, 56, -1, 57, 56, 27, -1, 27, 28, 57, -1,
 75, 74, 73, -1, 75, 73, 72, -1, 75, 72, 71, -1, 75, 71, 70, -1,
 75, 70, 69, -1, 75, 69, 68, -1, 75, 68, 67, -1, 75, 67, 66, -1,
 75, 66, 65, -1, 75, 65, 64, -1, 75, 64, 63, -1, 75, 63, 62, -1,
 75, 62, 61, -1, 75, 61, 60, -1, 75, 60, 59, -1, 75, 59, 58, -1,

```

75, 58, 85, -1, 75, 85, 84, -1, 75, 84, 83, -1, 75, 83, 82, -1,
76, 75, 82, -1, 77, 76, 82, -1, 77, 82, 81, -1, 78, 77, 81, -1,
78, 81, 80, -1, 78, 80, 79, -1, 106, 107, 108, -1,
106, 108, 109, -1, 101, 102, 103, -1, 100, 101, 103, -1,
99, 100, 103, -1, 98, 99, 103, -1, 97, 98, 103, -1,
96, 97, 103, -1, 95, 96, 103, -1, 94, 95, 103, -1,
93, 94, 103, -1, 92, 93, 103, -1, 91, 92, 103, -1,
90, 91, 103, -1, 89, 90, 103, -1, 88, 89, 103, -1,
87, 88, 103, -1, 86, 87, 103, -1, 113, 86, 103, -1,
112, 113, 103, -1, 111, 112, 103, -1, 110, 111, 103, -1,
110, 103, 104, -1, 110, 104, 105, -1, 109, 110, 105, -1,
106, 109, 105, -1]
} } ]      }      ]      }

```

Para que fuera posible el continuo movimiento de la moneda de ayuda, se obtuvieron las siguientes coordenadas por cada 0.06 seg para cada ciclo de intervalo de 10 seg dando un total de 19 coordenadas de orientación (rotación).

```

DEF Orientacion_Ayuda OrientationInterpolator {
key [0, 0.06, 0.12, 0.18, 0.24, 0.3, 0.36, 0.42, 0.48, 0.54, 0.6,
0.66, 0.72, 0.78, 0.84, 0.9, 0.96, 1, ]
keyValue [-1 0 0 -1.571, -0.9655 0.1842 -0.1842 -1.606, -0.8725 0.3455 -0.3455 -1.707,
-0.7442 0.4723 -0.4723 -1.862, -0.6015 0.5649 -0.5649 -2.059,
0.457 -0.629 0.629 -3.999, 0.3157 -0.6709 0.6709 -3.753,
0.1786 -0.6957 0.6957 -3.495, 0.04444 -0.7064 0.7064 -3.23,
-0.08897 -0.7043 0.7043 -2.964, -0.2239 -0.6892 0.6892 -2.701,
-0.3623 -0.6591 0.6591 -2.446, -0.5049 -0.6103 0.6103 -2.206,
-0.6497 -0.5375 0.5375 -1.989, -0.7895 -0.434 0.434 -1.805,
-0.9087 -0.2952 0.2952 -1.666, -0.9844 -0.1244 0.1244 -1.587,
-1 0 0 -1.571,] }

```

La definición del objeto temporizador de esta animación tiene las siguientes características Tiempo por ciclo 10 seg , debe repetirse cada ciclo (loop TRUE):

```

DEF Ayuda-TIMER TimeSensor {
  cycleInterval 10
  loop TRUE
}

```

5.3.8 Incorporación de la Animación a los Objetos Cuyo Comportamiento Depende de la Interacción con el Usuario o Cualquier otro Factor Externo Variante.

Dentro de este grupo de animaciones se encuentran aquellas que permiten la interacción con el usuario sea mediante acercamiento o activando un evento. Para esto se utilizó un número de sensores que determinan cuando el usuario activa la animación. A continuación se desarrollarán las animaciones que contiene el Ambiente Virtual.

5.3.8.1 Animación Ardilla.

Para este caso en particular la Animación Ardilla, que realiza un movimiento desde la grama hasta la cima de la palmera, se realizaron modificaciones del recorrido generando un número de nuevas coordenadas, tanto de rotación, como de posición. Teniendo como resultado los siguientes objetos interpoladores, cada 0.022 seg realiza un movimiento durante un tiempo total de 4.667 seg. Se define de la siguiente manera:

```

DEF Ardilla-POS-INTERP PositionInterpolator {

```

```

key [0, 0.02143, 0.04286, 0.06429, 0.08571, 0.1071, 0.1286, 0.15,
0.1714, 0.1929, 0.2143, 0.2357, 0.2571, 0.2786, 0.3, 0.3214,
0.3429, 0.3643, 0.3857, 0.4071, 0.4286, 0.45, 0.4714, 0.4929,
0.5143, 0.5357, 0.5571, 0.5786, 0.6, 0.6214, 0.6429, ]
keyValue [0.1012 0 -0.001915, 0.1012 0.01559 -0.1191, 0.1012 0.04754 -0.4027,
0.1012 0.0736 -0.7505, 0.1012 0.07942 -1.13, 0.1012 0.08031 -1.62,
0.1012 0.08114 -2.042, 0.1012 0.08988 -2.217, 0.1012 0.1602 -2.275,
0.1012 0.2846 -2.302, 0.1012 0.4345 -2.318, 0.1012 0.5823 -2.342,
0.1012 0.7249 -2.361, 0.1012 0.89 -2.377, 0.1012 1.107 -2.399,
0.1012 1.399 -2.435, 0.1012 1.754 -2.479, 0.1012 2.141 -2.527,
0.1012 2.533 -2.576, 0.1012 2.906 -2.622, 0.1012 3.298 -2.673,
0.1012 3.673 -2.72, 0.1012 3.973 -2.753, 0.1012 4.225 -2.774,
0.1012 4.484 -2.793, 0.1012 4.737 -2.809, 0.1012 4.967 -2.823,
0.1012 5.158 -2.833, 0.1012 5.296 -2.841, 0.1012 5.365 -2.844,
0.1012 5.37 -2.844, ] },
DEF Ardilla-ROT-INTERP OrientationInterpolator {
key [0, 0.02143, 0.04286, 0.06429, 0.08571, 0.1071, 0.1286, 0.15,
0.1714, 0.1929, 0.2143, 0.2357, 0.2571, 0.2786, 0.3, 0.3214,
]
keyValue [1 0 0 0, 0 1 0 -0.3433, 0 1 0 -1.03, 0 1 0 -1.545,
-0.0008853 1 0.000929 -1.59, -0.01058 0.9999 0.01088 -1.59,
-0.03995 0.9984 0.04082 -1.591, -0.09449 0.9883 0.1196 -1.625,
-0.1719 0.9115 0.3736 -1.923, 0.3568 -0.7437 -0.5654 -4.115,
0.5306 -0.6081 -0.5905 -4.143, 0.5679 -0.5833 -0.5807 -4.178,
0.5496 -0.6006 -0.5807 -4.176, -0.5319 0.63 0.5658 -2.068,
-0.5339 0.6449 0.5469 -2.017, -0.5348 0.6457 0.545 -2.012,
] }

```

La definición del objeto temporizador de esta animación tiene las siguientes características Tiempo por ciclo 4,667 seg y debe realizar un sólo ciclo (loop FALSE):

```

DEF ardilla-TIMER TimeSensor {

```

```

loop FALSE
cycleInterval 4.667
}

```

Los disparador son de tipo ProximitySensor o sensor de acercamiento, el cual permite definir un perímetro cuadrado, si el usuario entra en esa área céntrica a la posición inicial del objeto ardilla generará una señal (***ProxSensorArdilla.enterTime de tipo TIME***) y activa el evento. Se define de la siguiente manera:

```

DEF ProxSensorArdilla-ROOT Transform {
translation 13.04 0.005 -41.99
children [
DEF ProxSensorArdilla ProximitySensor {
enabled TRUE
center 0 3.5 0
size 12 12 12
}}

```

Esta señal generada por el Disparador será recibida por el objeto de tipo Nodos Script llamado GUION_Ardilla el cual verificará si la ardilla se encuentra en la grama y debe subir la palma o si ya subió. En el caso que deba comenzar a moverse y subir la palma enviará la señal que activará el Objeto de tipo motor, mediante la variable de salida Star, para comenzar el movimiento.

```

DEF GUION_Ardilla Script
{
eventIn SFTime empezar
eventOut SFTime star
eventOut SFBool Lp1
url
"vrmlscript:
function empezar(f, tm)

```

```

{
  if(!Lp1)
  {
    Lp1 = TRUE;
    star = f;      }      }      " }

```

La comunicación entre los eventos es controlada mediante los objetos tipo Enrutadores:

```

ROUTE ProxSensorArdilla.enterTime TO GUION_Ardilla.empezar
ROUTE GUION_Ardilla.star TO   ardilla-TIMER.startTime
ROUTE ardilla-TIMER.fraction_changed TO Ardilla-POS-INTERP.set_fraction
ROUTE Ardilla-POS-INTERP.value_changed TO Ardilla.set_translation
ROUTE ardilla-TIMER.fraction_changed TO Ardilla-ROT-INTERP.set_fraction
ROUTE Ardilla-ROT-INTERP.value_changed TO Ardilla.set_rotation

```

5.3.8.2 Animación Centro de Fotocopiados de Ingeniería

Esta animación representa el movimiento de la puerta del centro de fotocopia al Abrir/Cerrar. Este tipo de animación es activada por el usuario a través de un click con el mouse, el disparador de tipo TouchTime detecta el evento y comienza a generar una serie de suceso que controla la animación. Este objeto en particular por cada puerta tiene definido una zona de acercamiento para abrir la puerta desde adentro únicamente cuando el usuario active el ProximitySensor y este a su vez activa el TouchTime correspondiente a cada puerta, cada una son independiente pero la animación es en conjunto. A continuación se muestran los interpoladores de orientación (rotación) para esta animación, por cada puerta, cabe destacar que cada puerta es de doble pivot.

La definición del objeto temporizador para esta sesión de la puerta (salir) tiene la siguiente característica tiempo de ciclo 10 seg y solo realizara un solo ciclo (loop FALSE):

```
DEF PivotDoor14-TIMER TimeSensor { loop FALSE cycleInterval 10 }
```

El objeto ProxSensorPuerta01, permite la definición de la zona, donde el usuario debe estar para poder activar y abrir la puerta, tiene la siguiente característica, un área (ancho: 2.469, alto: 2.386, largo: 1.359) y punto céntrico en el eje z en 1.193 y siempre activo (enabled TRUE):

```
DEF ProxSensorPuerta01 Transform {
  translation 62.15 -1.049 -140.6
  children [
    DEF ProxSensorPuerta01 ProximitySensor {
      enabled TRUE
      center 0 1.193 0
      size 2.469 2.386 1.359
    } ]}
```

Los interpoladores de orientación (rotación) generado para esta animación son las siguientes:

```
DEF PivotDoor02_FACES01_FACES01-ROT-INTERP OrientationInterpolator {
  key [0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08,
    0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17,
    0.18, 0.19, 0.2, 0.22, 0.23, 0.26, 0.27, 0.3, 0.31,
    0.34, 0.35, 0.39, 0.4, 0.43, 0.44, 0.47, 0.48, 0.51,
    0.52, 0.55, 0.56, 0.58, 0.59, 0.61, 0.62, 0.63, 0.64,
    0.65, 0.66, 0.67, 0.68, 0.69, 0.7, 0.71, 0.72, 0.73,
```

0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8, 0.81, 0.82,
 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9, 0.91,
 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1,
]
 keyValue [0 -1 0.0004027 -3.141, 0 -1 0.0004027 -3.185,
 2.909e-005 -1 0.0004027 -3.285, 5.043e-005 -1 0.0004027 -3.39,
 6.543e-005 -1 0.0004027 -3.463, 7.765e-005 -1 0.0004027 -3.522,
 9.092e-005 -1 0.0004027 -3.585, 0.0001073 -1 0.0004027 -3.662,
 0.0001248 -1 0.0004027 -3.742, 0.0001461 -1 0.0004027 -3.837,
 0.0001776 -1 0.0004027 -3.972, -0.0002393 1 -0.0004027 -2.069,
 -0.0003278 1 -0.0004026 -1.775, -0.0003983 1 -0.0004026 -1.582,
 -0.0004047 1 -0.0004026 -1.566, -0.0004047 1 -0.0004026 -1.566,
 -0.0004043 1 -0.0004026 -1.567, -0.0004039 1 -0.0004026 -1.568,
 -0.0004035 1 -0.0004026 -1.569, -0.0004035 1 -0.0004026 -1.569,
 -0.0004031 1 -0.0004026 -1.57, -0.0004031 1 -0.0004026 -1.57,
 -0.0004027 1 -0.0004026 -1.571, -0.0004027 1 -0.0004026 -1.571,
 -0.0004023 1 -0.0004026 -1.572, -0.0004023 1 -0.0004026 -1.572,
 -0.0004019 1 -0.0004026 -1.573, -0.0004019 1 -0.0004026 -1.573,
 -0.0004015 1 -0.0004026 -1.574, -0.0004015 1 -0.0004026 -1.574,
 -0.0004011 1 -0.0004026 -1.575, -0.0004011 1 -0.0004026 -1.575,
 -0.0004007 1 -0.0004026 -1.576, -0.0004007 1 -0.0004026 -1.576,
 -0.0004003 1 -0.0004026 -1.577, -0.0004003 1 -0.0004026 -1.577,
 -0.0003999 1 -0.0004026 -1.578, -0.0003999 1 -0.0004026 -1.578,
 -0.0003995 1 -0.0004026 -1.579, -0.0003995 1 -0.0004026 -1.579,
 -0.0003991 1 -0.0004026 -1.58, -0.0003991 1 -0.0004026 -1.58,
 -0.0003987 1 -0.0004026 -1.581, -0.0003987 1 -0.0004026 -1.581,
 -0.0003959 1 -0.0004026 -1.588, -0.0003826 1 -0.0004026 -1.622,
 -0.0003632 1 -0.0004026 -1.674, -0.0003405 1 -0.0004026 -1.738,
 -0.0003177 1 -0.0004026 -1.806, -0.0002976 1 -0.0004026 -1.869,
 -0.0002823 1 -0.0004026 -1.919, -0.0002704 1 -0.0004026 -1.959,
 -0.0002598 1 -0.0004026 -1.996, -0.0002502 1 -0.0004026 -2.03,
 -0.0002411 1 -0.0004027 -2.063, 0.0002322 -1 0.0004027 -4.187,
 0.0002236 -1 0.0004027 -4.155, 0.000215 -1 0.0004027 -4.122,
 0.0002063 -1 0.0004027 -4.088, 0.0001975 -1 0.0004027 -4.053,

0.0001881 -1 0.0004027 -4.015, 0.0001784 -1 0.0004027 -3.975,
 0.0001683 -1 0.0004027 -3.933, 0.0001582 -1 0.0004027 -3.89,
 0.0001479 -1 0.0004027 -3.845, 0.0001377 -1 0.0004027 -3.8,
 0.0001277 -1 0.0004027 -3.755, 0.0001178 -1 0.0004027 -3.71,
 0.0001077 -1 0.0004027 -3.664, 9.831e-005 -1 0.0004027 -3.62,
 8.876e-005 -1 0.0004027 -3.575, 7.891e-005 -1 0.0004027 -3.528,
 6.767e-005 -1 0.0004027 -3.474, 5.593e-005 -1 0.0004027 -3.417,
 4.425e-005 -1 0.0004027 -3.36, 3.287e-005 -1 0.0004027 -3.304,
 2.235e-005 -1 0.0004027 -3.252, 1.35e-005 -1 0.0004027 -3.208,
 0 -1 0.0004027 -3.173, 0 -1 0.0004027 -3.15,
 0 -1 0.0004027 -3.141,] }

DEF PivotDoor02_FACES01_FACES02-ROT-INTERP OrientationInterpolator {

key [0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08,
0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17,
0.18, 0.19, 0.2, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26,
0.27, 0.28, 0.29, 0.3, 0.31, 0.32, 0.33, 0.34, 0.35,
0.36, 0.37, 0.38, 0.39, 0.4, 0.41, 0.42, 0.43, 0.44,
0.45, 0.46, 0.47, 0.48, 0.49, 0.5, 0.51, 0.52, 0.53,
0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62,
0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7, 0.71,
0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8,
0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89,
0.9, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98,
0.99, 1,]

keyValue [-1 -0.0002039 0.004285 -3.141, -0.9998 -0.000195 -0.01781 -3.141,
-0.9977 -0.0001748 -0.06737 -3.141, -0.9928 -0.0001529 -0.1196 -3.141,
-0.9878 -0.0001375 -0.1558 -3.141, -0.9827 -0.0001247 -0.1852 -3.141,
-0.9764 -0.0001112 -0.2159 -3.141, -0.9674 -9.435e-005 -0.2534 -3.141,
-0.9565 -7.681e-005 -0.2918 -3.141, -0.9415 -5.572e-005 -0.337 -3.141,
-0.9167 -2.565e-005 -0.3997 -3.141, -0.862 2.826e-005 -0.507 -3.141,
-0.7783 9.348e-005 -0.6279 -3.141, -0.7139 0.0001354 -0.7003 -3.141,
-0.7083 0.0001388 -0.7059 -3.141, -0.7085 0.0001387 -0.7057 -3.141,
-0.7087 0.0001386 -0.7055 -3.141, -0.709 0.0001384 -0.7052 -3.141,

-0.7093 0.0001382 -0.7049 -3.141, -0.7096 0.000138 -0.7046 -3.141,
-0.7098 0.0001379 -0.7044 -3.141, -0.7099 0.0001378 -0.7043 -3.141,
-0.71 0.0001378 -0.7042 -3.141, -0.7101 0.0001377 -0.7041 -3.141,
-0.7102 0.0001377 -0.704 -3.141, -0.7103 0.0001376 -0.7039 -3.141,
-0.7104 0.0001375 -0.7038 -3.141, -0.7105 0.0001375 -0.7037 -3.141,
-0.7105 0.0001375 -0.7037 -3.141, -0.7106 0.0001374 -0.7036 -3.141,
-0.7107 0.0001374 -0.7035 -3.141, -0.7108 0.0001373 -0.7034 -3.141,
-0.7109 0.0001372 -0.7033 -3.141, -0.711 0.0001372 -0.7032 -3.141,
-0.711 0.0001372 -0.7032 -3.141, -0.7111 0.0001371 -0.7031 -3.141,
-0.7112 0.0001371 -0.703 -3.141, -0.7113 0.000137 -0.7029 -3.141,
-0.7114 0.0001369 -0.7028 -3.141, -0.7114 0.0001369 -0.7028 -3.141,
-0.7115 0.0001369 -0.7027 -3.141, -0.7116 0.0001368 -0.7026 -3.141,
-0.7117 0.0001368 -0.7025 -3.141, -0.7117 0.0001367 -0.7024 -3.141,
-0.7118 0.0001367 -0.7024 -3.141, -0.7119 0.0001366 -0.7023 -3.141,
-0.712 0.0001366 -0.7022 -3.141, -0.7121 0.0001365 -0.7021 -3.141,
-0.7122 0.0001365 -0.702 -3.141, -0.7122 0.0001364 -0.7019 -3.141,
-0.7123 0.0001364 -0.7018 -3.141, -0.7124 0.0001363 -0.7018 -3.141,
-0.7125 0.0001363 -0.7017 -3.141, -0.7126 0.0001362 -0.7016 -3.141,
-0.7127 0.0001362 -0.7015 -3.141, -0.7128 0.0001361 -0.7014 -3.141,
-0.7129 0.000136 -0.7012 -3.141, -0.713 0.0001359 -0.7011 -3.141,
-0.7131 0.0001359 -0.701 -3.141, -0.7133 0.0001358 -0.7009 -3.141,
-0.7134 0.0001357 -0.7008 -3.141, -0.7135 0.0001357 -0.7007 -3.141,
-0.7136 0.0001356 -0.7005 -3.141, -0.7138 0.0001355 -0.7004 -3.141,
-0.7163 0.0001339 -0.6978 -3.141, -0.7278 0.0001268 -0.6858 -3.141,
-0.7456 0.0001155 -0.6664 -3.141, -0.7665 0.0001016 -0.6422 -3.141,
-0.7878 8.678e-005 -0.6159 -3.141, -0.8068 7.289e-005 -0.5908 -3.141,
-0.8214 6.179e-005 -0.5703 -3.141, -0.8325 5.3e-005 -0.554 -3.141,
-0.8426 4.482e-005 -0.5386 -3.141, -0.8518 3.708e-005 -0.5239 -3.141,
-0.8604 2.967e-005 -0.5097 -3.141, -0.8685 2.244e-005 -0.4957 -3.141,
-0.8764 1.52e-005 -0.4816 -3.141, -0.8842 0 -0.4672 -3.141,
-0.8919 0 -0.4522 -3.141, -0.8998 0 -0.4363 -3.141,
-0.9079 -1.612e-005 -0.4191 -3.141, -0.9162 -2.515e-005 -0.4007 -3.141,
-0.9244 -3.454e-005 -0.3814 -3.141, -0.9324 -4.415e-005 -0.3614 -3.141,
-0.9402 -5.399e-005 -0.3406 -3.141, -0.9476 -6.394e-005 -0.3195 -3.141,

```
-0.9546 -7.394e-005 -0.298 -3.141, -0.9611 -8.397e-005 -0.2762 -3.141,
-0.9671 -9.391e-005 -0.2544 -3.141, -0.9726 -0.0001037 -0.2325 -3.141,
-0.9775 -0.0001133 -0.211 -3.141, -0.9822 -0.0001236 -0.1878 -3.141,
-0.9869 -0.000135 -0.1615 -3.141, -0.9911 -0.0001471 -0.1333 -3.141,
-0.9945 -0.0001593 -0.1046 -3.141, -0.997 -0.0001709 -0.07678 -3.141,
-0.9987 -0.0001815 -0.05105 -3.141, -0.9996 -0.0001906 -0.02881 -3.141,
-0.9999 -0.0001977 -0.01127 -3.141, -1 -0.0002023 0.0001981 -3.141,
-1 -0.0002039 0.00431 -3.141, ] }
```

El objeto TouchSensor, su definición es la siguiente:

```
DEF TouchSensor05Puerta02S-SENSOR TouchSensor { enabled FALSE };
```

```
DEF TouchSensor06Puerta01S-SENSOR TouchSensor { enabled TRUE };
```

A continuación el nodo Script que controla el motor de animación, verificando y cambiando el estado a través de la variable de salida ActivePue:

```
DEF GuionActivarPuerta Script
{
  eventIn SFTime ActivarPuerta
  eventOut SFBool ActiPue
  url
  "vrmlscript:
    function ActivarPuerta(tm)
    {
      if(!ActiPue)
        ActiPue = TRUE;
      else
        ActiPue = FALSE;
    }
  "
```

Y los objetos Enrutadores para este evento son generados de la siguiente manera:

```
ROUTE ProxSensorPuerta01.enterTime TO GuionActivarPuerta.ActivarPuerta
ROUTE GuionActivarPuerta.ActiPue TO TouchSensor05Puerta02S-SENSOR.enabled
ROUTE ProxSensorPuerta01.exitTime TO GuionActivarPuerta.ActivarPuerta
ROUTE GuionActivarPuerta.ActiPue TO TouchSensor05Puerta02S-SENSOR.enabled
```

```
ROUTE ProxSensorPuerta02.enterTime TO GuionActivarPuerta.ActivarPuerta
ROUTE GuionActivarPuerta.ActiPue TO TouchSensor06Puerta01S-SENSOR.enabled
ROUTE ProxSensorPuerta02.exitTime TO GuionActivarPuerta.ActivarPuerta
ROUTE GuionActivarPuerta.ActiPue TO TouchSensor06Puerta01S-SENSOR.enabled
```

```
ROUTE PivotDoor14-TIMER.fraction_changed TO PivotDoor02_FACES01_FACES01-ROT-
INTERP.set_fraction
ROUTE PivotDoor02_FACES01_FACES01-ROT-INTERP.value_changed TO
PivotDoor02_FACES01_FACES01.set_rotation
```

```
ROUTE PivotDoor14-TIMER.fraction_changed TO PivotDoor02_FACES01_FACES02-ROT-
INTERP.set_fraction
ROUTE PivotDoor02_FACES01_FACES02-ROT-INTERP.value_changed TO
PivotDoor02_FACES01_FACES02.set_rotation
```

5.3.8.3 Animación Puerta de Cubículo

Esta animación tiene cierto parecido a la animación del centro de fotocopia en la forma como el usuario interactúa para activar la misma. El disparador tipo TouchSensor genera la cadena de evento y como resultado las puertas de los diferentes cubículos (Cubículos de Profesores 1, 2, 3, Sala “RENE CABRERA, Sala de Navegación de Ingeniería y Jefatura de CYS) cambia de estado de cerrado a medio

abrir, siendo este el punto de entrada a las respectivas salas a través de un cambio de escena controlado por sus nodo Script respectivos.

5.3.9 Agregar los Efectos Sonoros.

Para incluir sonido dentro del recorrido virtual, hicimos las siguientes definiciones de los siguientes nodos:

➤ Sound

Este nodo se usa para especificar la posición y orientación de una fuente de sonido en el mundo. El nodo Sound se coloca en un cierto lugar, definiendo una área en donde se podrá oír el sonido dependiendo del lugar en que se coloque en la escena respecto al sonido.

➤ AundioClip

El nodo AudioClip especifica qué sonara por el nodo sound, y cuándo, lugar de almacenamiento, velocidad y adicional se puede agregar una breve descripción del tipo de sonido.

5.3.9.1 Definición del Sonido Abrir Puertas del Ambiente Virtual

Para lograr el buen funcionamiento del sonido tomamos en consideración las siguientes características, el formato del audio es “.wav”, la velocidad fue establecida por defecto por el nodo AudioClipAbrirPuerta con un loop FALSE ya solo sonara una vez, el archivo de sonido utilizado “Logoff.wav”. Se define de la siguiente manera:

```

DEF AudioClipAbrirPuerta AudioClip {
    description ""
    url "C:\sound\LOGOFF.wav"
    pitch 1
    loop FALSE
    startTime 1
}

```

Una vez definido el nodo AudioClip, se establece la relación del nodo Sound para que pueda ser reproducido y ser percibido por el usuario, la definición es de la siguiente manera con las siguientes características, con un alcance de 1.107 metros con una prioridad de 1 para máxima prioridad de sonido e intensidad de volumen de 1 y sonido siempre desvaneciente.

```

DEF SoundAbrirPuerta2 Transform {
    translation 62.1 0 -152.4
    children [
        DEF SoundAbrirPuerta02 Sound {
            direction 0 0 1
            intensity 1
            location 0 0 0
            maxBack 1
            maxFront 1.107
            minBack 1
            minFront 1
            priority 1
            spatialize TRUE
            source
            USE AudioClipAbrirPuerta    } ]]

```

5.3.9.2 Definición del Sonido Límite del Ambiente Virtual

Este efecto sonoro es el producido cuando el usuario colisiona contra el límite de navegación del ambiente virtual, al igual que el sonido de abrir puerta realizamos la definición de un nodo Sound con características de un máximo de alcance de 1500 metro para garantizar que sea percibido por el usuario en cual escena del ambiente virtual y un AudioClip con un archivo de audio tipo “.wav” llamado “boing_1.wav”. La definición final es la siguiente:

```

Sound {
    minFront 10
    minBack 10
    maxFront 1500
    maxBack 1500
    source DEF SonidoColision AudioClip {
        url "../sound/boing_1.wav"
    }
}

```

5.3.10 Acoplar las Actualizaciones y Nuevas Zonas con AVDONA_1.

Realizada la conversión al código VRML de cada unas de las nuevas zonas del recorrido virtual y terminado por separado una optimización parcial de la misma, se procedió a incluir y acoplar con la versión 1 de AVUDONA utilizando técnicas de desarrollo incremental, el cual, nos permite usar el sistema y corregir los fallos al distribuir el incremento del ambiente virtual. Es importante señalar que no existe herramienta VRML "perfecta", cualquier proyecto VRML por complejo que se precie

requerirá un adecuado ajuste "a mano", en los archivos VRML generados por éstas. Utilizando VrmIPad realizamos todos los ajustes necesarios no solo en las escenas, sino que también a los objetos. Abrimos el archivo de AVUDONA_1 con el editor VrmIPad y a través de un simple “copy” y “paste” incluimos el código de las nuevas escenas, se compila y se ejecuta obteniendo una versión parcial de AVUDONA, cabe destacar que, previamente se confirmó que el código de la escena a incluir funcione a la perfección. En la figura 5.8 se puede observar cuando el código del Decanato ya incluido en AVUDONA_2.

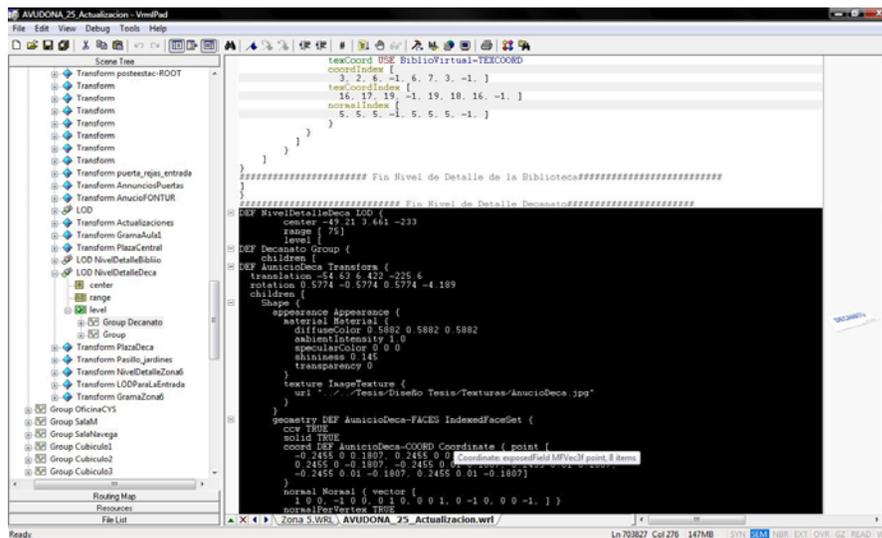


Figura 5.8 Ilustración Código AVUDONA_2 en VrmIPad 2.1

5.3.11 Inclusión de las Guías de Navegación Interactivas Para los Usuarios.

La guía de navegación para AVUDONA_2, estará conformada de cuatro maneras, primero, a través de mensajes de texto que se presentarán en la barra de sistema cuando el usuario se encuentre en algún punto del Ambiente Virtual y se espera alguna acción del mismo (un click, eje. Para que abra las puertas del

Departamento), segundo, mensajes informativos que ofrece al usuario una breve descripción de la escena que está visitando, esta información aparece de manera automática con la presencia del usuario en la escena en una ventana translúcida, tercero, utilizando la cámara guiada del menú de punto de vista que ayudara al usuario a realizar un recorrido automático por AVUDONA_2, y cuarto, un enlace a una página HTML que ofrece información del ambiente virtual y ayuda sobre los controles de navegación, el cual, el usuario podrá desplegar con solo hacer click en las monedas con el signo de interrogación ubicada en escenas estratégicas del ambiente virtual.

5.3.12 Definición de la Forma de Desplazamiento de los Usuarios en el Ambiente Virtual.

Tomando en cuenta que el ambiente virtual desarrollado es basado en una representación real, en donde la fuerza de gravedad afecta al usuario al desplazarse por el ambiente virtual. Definimos el nodo NavigationInfo con las siguientes características:

➔ **VisibilityLimit:** Se puede especificar además un límite de visibilidad al mundo. Esta es la distancia desde el usuario tras la cual nada será dibujado en pantalla. Se puede usar para aumentar la velocidad de ejecución en escenarios muy grandes. Si está determinado a 0 0 0 (tal y como viene por defecto), no habrá límite de visibilidad. En este caso no se modifica el valor por defecto ya que el Ambiente virtual no posee distancias tan amplias en relación con el usuario que no puedan ser vista por el usuario.

➤ **Headlight:** Establecida por defecto es TRUE, de modo que se ilumina las escenas por una especie de faros de coche. En el Ambiente Virtual esta luz es desactivada debido a que la iluminación de la escena está definida ya por los Tipos de luces incluidos.

➤ **Type:** El usuario se puede desplazar por el Ambiente Virtual de la siguientes formas básicas:

- "WALK", Donde el usuario camina normalmente siendo afectado por la gravedad, y
- "NONE", donde el usuario no tiene control de movimiento en absoluto. Este último es utilizado cuando el usuario activa la cámara guiada.

En el Ambiente Virtual se definió que el usuario tendrá un desplazamiento de tipo “Walk” para que sienta una mayor sensación de inmersión al verse afectado como en la realidad por la fuerza de gravedad.

➤ **AvatarSize:** Permite establecer el área física de presencia del usuario (o *avatar*) durante el recorrido. Tiene tres valores de escala. Estos representan, en orden, al avatar:

- Radio
- Altura de los ojos
- Altura de las rodillas

Para AVUDONA_2 se estableció estos valores de navegación para el AvatarSize de la siguiente manera:

RADIO USUARIO = 0.25.

ALTURA DE LOS OJOS = 1.6.

ALTURA MAXIMA QUE ESCALA = 0.75.

➤ **Speed:** La velocidad de desplazamiento del usuario. Por defecto su valor es de 1. Los valores de 2, 3, 4, 5... son valores que multiplican la velocidad de desplazamiento. En este caso el valor que se le colocó fue de 2.

La definición total del nodo fue la siguiente:

```
DEF NavegacionPrincipal NavigationInfo {
  avatarSize [0.25, 1.6, 0.75]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}
```

5.3.13 Establecimiento de las Características del Clima (Cielo, Piso, Niebla).

Establecer las condiciones climáticas que van a prevalecer en AVUDONA_2, es de suma importancia, ya que permite incrementar el realismo del ambiente virtual, así logra una inmersión total del usuario. El nodo Background permite llevar a cavo esta característica estableciendo un panorama que represente el día y otro la noche.

Los campos backUrl, bottomUrl, frontUrl, leftUrl, rightUrl, y topUrl especifican un juego de imágenes que definen un panorama de background entre la tierra/cielo backdrop y la geometría de la escena. El panorama consta de seis imágenes, cada una de las cuales es mapeada en una cara de un cubo infinitamente

grande, contenido dentro de las esferas del backdrop y centrada en el sistema coordenado local. Las imágenes son aplicadas individualmente a cada cara del cubo, las texturas utilizada están definidas en la tabla 5.2 y tabla 5.3.

El nodo Background queda definido de las siguientes maneras respectivamente:

```

DEF PanoramaDia Background {
  groundColor [0.66 0.88 0.97]
  groundAngle 1.5 skyColor [0.66 0.88 0.97]
  skyAngle 1.64
  backUrl ""
  frontUrl "../img/PanoramaDia2.jpg"
  rightUrl ""
  leftUrl "../img/PanoramaDiaIzq2.jpg"
  bottomUrl ""
}

DEF PanoramaNoche Background {
  groundColor [0 0 0]
  groundAngle 1.5 skyColor [0 0 0]
  skyAngle 1.64
  backUrl ""
  frontUrl "../img/PanoramaNoche2.jpg"
  rightUrl ""
  leftUrl "../img/PanoramaNoceIzq2.jpg"
  bottomUrl ""
}

```

Tabla 5.2 Textura de la Panorama Día

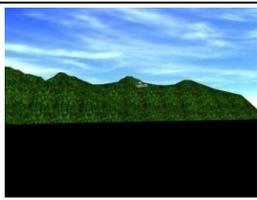
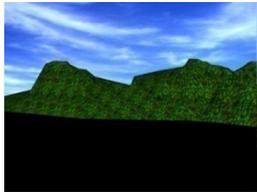
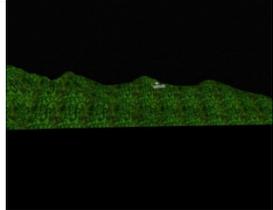
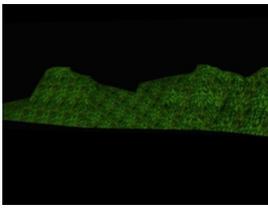
BackUrl	BottomUrl	FrontUrl
		
LeftUrl	RightUrl	TopUrl
		

Tabla 5.3 Textura de la Panorama Noche

BackUrl	BottomUrl	FrontUrl
		
LeftUrl	RightUrl	TopUrl
		

5.3.14 Regulación de la Rapidez de Desplazamiento de los Usuarios por el Ambiente Virtual.

Para establecer la rapidez para la nueva versión de AVUDONA_2, acorde con el tamaño del ambiente virtual y el grado de inmersión definimos un valor mínimo de 2, establecido como predeterminado. Este valor puede ser cambiado por el usuario el cual se recomienda un valor óptimo máximo de 5.

5.4 PUBLICACIÓN DE AVUDONA_2

En esta parte del proyecto representa la parte final de la fase de construcción, la implementación. Se termina de desarrollar los componentes y codificar el mundo

virtual, su relación con la base de datos, definir y explicar cómo se activar el mundo virtual. Para lograr el cumplimiento de esta fase seguimos los siguientes pasos, primero, simplemente ubicamos el archivo (.wrl) en el directorio(dirección física de almacenamiento sea local o remota), tomando como referencia a la ruta de acceso almacenada en la base de datos, segundo se crea una introducción hacia el mundo virtual mediante una página web(HTML) que explique qué es lo que va a encontrar, con qué browser se consigue un mejor resultado para explorarlo, y en qué versión del VRML está hecha. Todo esto le permitirá al usuario no cargar mundos que a la postre le resulten poco interesantes, o intentar cargar un mundo usando un browser incompatible con la versión para la cual éste ha sido.

A continuación se mostrara la interfaz que permite activar el mundo virtual en forma de plantillas y una breve explicación del código fuente.

5.4.1 Interfaz HTML de AVUDONA_2.

Para la publicación e implantación del ambiente virtual se diseño una interfaz web provisional que permite al usuario realizar el enlace con AVUDONA_2, la cual solo tendrá valides para efecto de presentación ya que dicho proyecto está planteado para la pagina web oficial de la Universidad de Oriente Núcleo de Anzoátegui. Esta interfaz cumple específicamente con la función de permite al usuario activar el ambiente virtual desde la misma, por tal motivo se presenta una interfaz sencilla compuesta por tipo de fuente, tablas, imágenes y clara instrucciones de cómo activar y visualizar el ambiente virtual. A continuación se presenta el código fuente de la página, y en la figura 5.9 se puede observar la vista gráfica de la misma.



Figura 5.9 Interfaz Provisional para activar a AVUDONA_2.

Código Fuente:

```

<!--Este codigo no forma de la estructura funcional de AVUDONA-2 solo es utilizado
como página Web provisional para activar el ambiente virtual y enlazar con la base de datos--
>
<html>
<head>
<title>Pagina Ambiente Virtual</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#000000" >
<font color="#FFFFFF">

<?php
// codigo Php utilizado unicamente para realizar la conexión con la base de datos
// en la interfaz provisional para activar el ambiente virtual.
// aquí se realiza la conexión con la base de datos SOFIA y se realiza la consulta
$conectando = mssql_connect('SOFIA','SA,') or die ('Fallo la Conexión con la Base de
Datos');

```

```

mssql_select_db('AVUDONA') or ('Fallo... Base de Ddato no existe');
$resultaDato = mssql_query("select * from Ambiente_Virtual ",$conectando) or
die('Fallo...Dato no Existe');
$DatoColumna = mssql_fetch_array($resultaDato);
// fin del codigo
?>
</font>
<table border="0" height="0" align="center" bordercolor="#FFFFFF"
bgcolor="#000000">
<tr>
<td colspan="2" height="156">
<div align="center"></div>
</td>
</tr>
<tr>
<td height="33" colspan="2">
<div align="center"><font color="#FFCC00">AMBIENTE VIRTUAL DE LA
UNIVERSIDAD
DE ORIENTE NUCLEO ANZOATEGUI</font></div>
</td>
</tr>
<tr>
<td height="20">
<div align="center"><font size="-1"></font></div>
</td>
<td height="20">
<div align="left"><font size="-1" color="#CCCCCC">La visualizaci&oacute;n
de los modelos 3D en VRML require la instalaci&oacute;n de un plug-in
(Cliente Cortona) haga click en el siguiente enlace</font><font size="-1"
color="#999999">
</font><font size="-1"><a href="http://www.cortona3d.com/cortona">Descargar
visualizador</a> </font></div>

```

```

</td>
</tr>
<tr>
<td colspan="2" height="269">
<div align="left">
<table width="99%" border="0" bgcolor="#000000" bordercolor="#000000">
<tr>
<!-- en esta parte del codigo utilizando el comando echo incluimos el dato de la
consulta
con la ruta de acceso al archivo (.wrl)-->
<td width="40%" height="266"><a href="<?php echo $DatoColumna['ruta_Acceso']
?>" tiile=" Haga klik para activas" target="_blank">
</a></td>
<td width="60%" height="266"> <div align="justify"><b><font
color="#CCCCCC">Ambiente
Virtual, UDO-ANZ</font></b><font color="#CCCCCC">. En el recorrido
virtual se podra visualizar una represenatacion de las instalaciones
del recinto universitario desde la Entrada Principal, Edificio
de Biblioteca &quot;Luis Garcia Pellisier&quot; y Decanato hasta
la Escuela de Ingeniería (Dpto Ingeniería Computación y Sistema,
Dpto Ingeniería Industrial) y sus alrededores. Realizado por Br:
José Millán. /Tama&ntilde;o 24 Mb/ </font></div></td>
</tr>
</table>
<div align="left"><font color="#FFCC33"><b>Para activar el Ambiente virtual
haga click sobre la imagen.</b></font></div>
</div>
</td>
</tr>
</table>
</body>
</html>

```

CAPITULO VI – FASE DE TRANSICION

6.1 INTRODUCCIÓN

Esta fase se centra en implantar el producto en su entorno de operación, ejecutando las pruebas diseñadas para verificar así su correcto funcionamiento o posibles fallas que se puedan presentar y deban de corregirse. El software se entrega a los usuarios para realizar pruebas beta, así se dará el visto bueno al sistema dando como finalizado el desarrollo del proyecto.

6.2 PLANIFICACIÓN DE LA FASE DE TRANSICIÓN

En la parte anterior se obtuvo el diseño de cada zona que van a conformar la nueva versión de AVUDONA_2, y se realizó la conversión cada una al código VRML, terminando cada una por separado dando una optimización parcial de la misma. En esta fase el principal objetivo es acoplar las actualizaciones y nuevas zonas con AVUDONA_1. Para ello se realizó la prueba de cada zona con la finalidad de descubrir y corregir los posibles defectos que se puedan presentar en cada escena, posteriormente ser acoplado utilizando técnicas de desarrollo incremental y presentar la primera versión beta de AVUDONA_2.

6.3 PRUEBA

Las pruebas realizadas aplicadas al recorrido virtual tuvieron una finalidad, descubrir posibles errores en cuanto al diseño y la codificación. Es importante señalar que no existe herramienta VRML "perfecta", cualquier proyecto VRML por complejo que se precie requerirá un adecuado ajuste "a mano", en los archivos VRML generados por éstas. Por el tipo de características del sistema se debe considerar que las únicas entradas al sistema son del tipo visual o de interfaz con el usuario y no necesitan validación, tal como, el teclado y mouse, generando salidas visuales y auditivas. Dadas estas consideraciones las pruebas se realizaron tomando en cuenta, la visualización y la audición. En resumen las pruebas permitirán apreciar si el resultado obtenido es el esperado o no y si efectivamente se ha logrado una representación reconocible con el modelo original.

Considerando la magnitud del ambiente virtual desarrollado solo se representa una escena por cada nueva zona y para una actualización. A continuación se realizara las pruebas de las escenas del ambiente virtual utilizando la técnica de PNC (Prueba de la Caja Negra).

6.3.1 PNC Aplicada al Edificio Decanato Perteneciente a la Zona 5

➔ Escena: Decanato.

➔ Descripción: Representa el edificio del Decanato del núcleo, sede de la mayoría de las actividades administrativas y académica más importante

➤ Verificación

- Alineación de los objetos de la escena (pilares, ventana, puertas, etc,...) y exacta ubicación. Se logra comprobando el objeto una vez hecha la conversión de 3D Max a código VRML, corrigiendo las pequeñas distorsión de los objetos. Esto se repite hasta logra la completa alineación de todos los objetos de la escena.

- Color de los objetos tras la aplicación del efecto de luz que pierdan la armonía del color con respecto al objeto real. El color aplicado en la etapa de diseño al realizar la conversión sufre alteraciones del color, para ello es necesario observar los colores de cada objeto y realizar la corrección.

- Posición de la escena en el ambiente virtual. En este objetivo se debe observar que la orientación del objeto es la correcta según los ejes (x, y, z) respectivo aplicado en el diseño.

➤ Resultado: Satisfactorio

6.3.2 PNC Aplicada a la Animación Información Perteneciente a la Zona 6

➤ Escena: Animación Información

➤ Descripción: Animación que le permite al usuario recibir información en relación a la zona que está visitando en el ambiente virtual.

➤ Verificación

- Activación de la animación. Comprobar que los nodos de proximidad estén en la ubicación correcta para que el usuario visualice la información, sino es correcta se debe cambiar la ubicación hasta logra la posición donde se desee que se muestre la información.

- Visualización de la información. Verificar que se puede realizar una clara lectura de contenido de información y que corresponda a la escena o zona visitada por el usuario. También se debe tomar en cuenta la distancia de la información respecto al avatar del usuario y realice el seguimiento al usuario mientras se esté moviendo en la escena.

➤ Resultado: Satisfactorio.

6.3.3 PNC Aplicada a la Animación luz Fachada Entrada de la Universidad Perteneciente a las Actualizaciones

➤ Escena: Animación Luz Entrada Principal.

➤ Descripción: Es la animación que representa la iluminación de los faros de luz de la fachada de la entrada de la universidad. El encendido/apagado es controlado por el sistema. Esta animación forma parte de la actualización de AVUDONA.

➤ Verificación:

- Visualizar que la representación de la iluminación se activa automáticamente por el sistema según la hora en que el usuario visita el

ambiente virtual e ilumine el arco de la entrada de la universidad, el cual, se debe verificar la directriz de foco de luz sea correcto y corregir si es necesario.

- Verifica forma, color de la representación de los faros luz, que tenga la forma de los faros utilizados para iluminar la entrada de la universidad en el entorno real y no tenga forma entre montadas que distorsione la representación del objeto. El color sea el correcto según el objeto real. Estas propiedades del objeto se revisaran todas las veces necesario hasta alcanzar el preforma del objeto.

- La posición correcta según su ubicación en el entorno real, esta subpruebas debe repetirse hasta logra la posición correcta del objeto, cambiando las coordenadas en el nodo que representa el objeto.

➡ Resultado: Satisfactorio.

6.4 INTEGRACIÓN

Una vez realizada la conversión al código VRML de cada uno de los objetos por separado y terminado la optimización parcial de la misma, se procedió a incluir y acoplar cada escena en su respectiva zona, para formar sub-mundo virtuales aplicando el método de Hebras seleccionada de las técnicas de desarrollo incremental, el cual, nos permite usar el sistema y corregir los fallos al distribuir el incremento en el sistema del ambiente virtual.

Realizada la completa integración de cada sub-mundo que formara parte de AVUDONA_2, procedimos a realizar la integración del nuevo sub-mundo y las actualizaciones con las escenas de AVUDONA_1. Utilizando *VrmlPad* realizamos todos los ajustes necesarios no solo en las escenas, sino que también a los objetos. Abrimos el archivo de AVUDONA_1 con el editor *VrmlPad* y a través de un simple “copy” y “paste” incluimos el código de los sub-mundos, se compila y se ejecuta obteniendo una versión parcial de AVUDONA, cabe destacar que, previamente se confirmo que el código de la escena a incluir funcione a la perfección. En la figura 6.1 se puede observar cuando el código del Decanato ya incluido en AVUDONA_2.

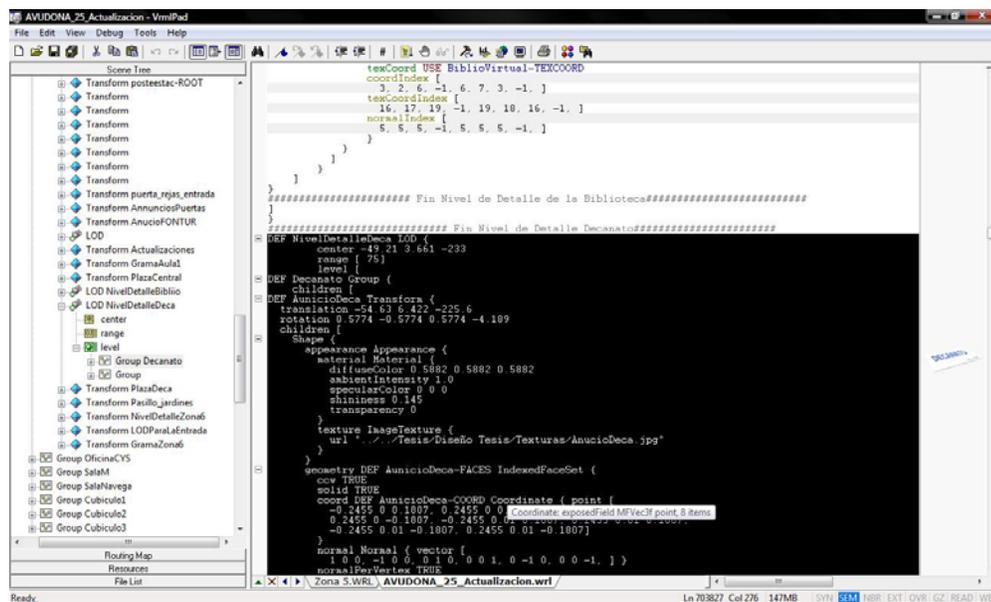


Figura 6.1 Integración del Código del Sub-Mundo de la Zona 5 a AVUDONA_1 en VrmlPad 2.1

A continuación se describe las fases del método aplicado en el proceso de integración en el desarrollo del ambiente virtual.

Fase 1: Sub-Mundo Virtual Plaza Central “11 de Julio”-Edificio Decanato-Biblioteca (zona 5). Este sub-mundo se encarga de agrupar los objetos de la zona 5. Los objetos son: Edificio Decanato, Edificio Biblioteca, Plaza Central “11 de Julio”,

Plaza Decanato, Casilla de transformadores, Escaleras, Pasillo de acceso, Estacionamientos, Poste de iluminación, Jardines, Ventanas, Puertas. Cada uno desarrollado por separado y considerado como una escena del sub-mundo. Terminado cada escena se continuó con el ensamblaje del sub-mundo virtual, agrando de una manera progresiva cada escena, así lograr una integración satisfactoria y poder observar que se logro dar el nivel de realismo deseado según homónimo, al realizar un recorrido virtual al sub-mundo donde se aprecia los edificio Decanato, pasillos de acceso, poste de iluminación, etc.

Fase 2: Sub-Mundo Virtual Auditorio-Depto. Ingeniería Industrial-Edificio de Servicios (zona 6). Este sub-mundo agrupa y controla los objetos correspondientes a la zona 6. Los objetos son: Auditorio, Depto. Ing. Industrial, Edif. De Servicios, Pasillo de acceso, Caminaderas, Jardines, Aulas Ingeniería (I-10, I-11, I-12), Puertas, Ventanas. Cada uno desarrollado por separado e integrado progresivamente al sub-mundo, logrando un satisfactorio ensamblaje al realizar el recorrido virtual.

Fase 3: Sub-Mundo Virtual Actualización Fachada Entrada Principal del Núcleo (zona 1). Este sub-mundo agrupa las actualizaciones que se realizaron en algunas escenas de la zona 1. Las escenas actualizadas son: Portón Entrada y agregados: Faros de Iluminación, Monumento “50 Aniversario”. Cada uno diseñado por separado, en escenas diferentes. Posteriormente fueron intercambiadas los objetos y agregado en la escena, logrando la actualizar el sub-mundo en relación al aspecto real de los elementos que componen la fachada del Núcleo.

Fase 4: Sub-Mundo Virtual Actualización Camino Central del Nucleó (Zona 2). En este sub-mundo se realiza la actualización de los objetos; Palmera, Fecha del Monumento de la Plaza Bolívar, Faros del monumento y Acercade, asociado a la zona 2. Su diseño se realizo en escenas diferentes para luego ser

integrada en el sub-mundo. Cambiando los objetos respectivos dando una apariencias más actual. Logrando una actualización satisfactoria.

Fase 5: Sub-Mundo Virtual Actualización Pasillo del Depto. De Computación y Sistema (zona 3). Este sub-mundo asocia las actualizaciones a los objetos a la zona 3. El objeto; Banco Pasillo. Fue diseñado en una escena diferente y agregado al sub-mundo zona 3 logrando una actualización satisfactoria.

Fase 6: Sub-Mundo Virtual Actualización Depto.-Aulas-Escuela de Ingeniería (Zona 4). Este sub-mundo asocia las actualizaciones a los objetos de la zona 4. Los objetos actualizados son: Pasillos de Accesos, Sala “RENE CABRERA”, Cubículo de Profesores (1, 2, 3), Sala de Navegación, Aula de Computación y Sistema (AICS), Centro de Fotocopias de Ingeniería, FONTUR. Cada unos de estos objetos se desarrollaron por separado en escenas diferentes, luego fueron agregando uno a uno al sub-mundo de la zona 4 logrando el nivel de actualización satisfactoria acorde con los aspectos que involucran la escena real.

Fase 5: Sub-Mundo Virtual Actualización Animaciones. Este procedimiento se encarga de asociar las actualizaciones de las animaciones del mundo virtual. Su integración tiene 3 fases: en esta se realiza la unión de la animación Movimiento Puerta Centro de Fotocopia y Sonido Puerta del Centro de Fotocopia para formar la Animación Puerta Centro de Fotocopia, además se realiza la actualización de Animación Aridilla-Nueva Ruta, Animación Información de Escena, Animación Cielo-Nueva Textura, Animación Icono de Ayuda, Animación Puerta Sala “RENE CABRERA”, Animación Puerta Cubículo 1, Animación Puerta Cubículo 2, Animación Puerta Cubículo 3, Animación Puerta Jefatura CYS, Animación Puerta Sala de Navegación de Ingeniería dando un resultado satisfactorio.

Fase 6: Unión Movimiento Puerta Centro de Fotocopia y Sonido Puerta De Fotocopia. Esta fase contribuye de manera satisfactoria la integración para forma la Animación Puerta Centro de Fotocopia e integrar la fase Animación.

Fase 7: Actualización Ayuda. Este procedimiento se encarga de asociar las actualizaciones de los procesos de la Ayuda Interactiva y Ayuda General del Sistema. Fueron desarrollados de forma independiente y posteriormente integrados de manera satisfactoria.

Fase 9: Sub-Mundo Actualización Virtual Fachada-Camino Central se forma al integrar el Sub-Mundo Virtual Actualización de la Fachada de Entrada Principal del Núcleo con el Sub-Mundo Virtual Actualización Camino Central del Núcleo, obteniendo resultando satisfactorio.

Fase 10: Sub-Mundo Virtual Actualización Fachada-Camino Central-Pasillos se forma al integrar el Sub-Mundo Virtual Actualización Pasillos del Dpto. de Computación y Sistemas, al Sub-Mundo Virtual Actualización Fachada-Camino Central, obteniendo resultando satisfactorio.

Fase 11: Sub-Mundo Virtual Actualización AVUDONA-Pasivo. Se forma al integrarle al Sub-Mundo Virtual Actualización Fachada-Camino-Central el Sub-Mundo Virtual Actualización Dpto.-Aulas y Escuela de Ingeniería completando el Ambiente Virtual pasivo (todos los objetos pero sin animaciones). Obteniendo resultado fue satisfactorio.

Fase 12: Sub-Mundo Virtual AVUDONA-Animado. Se forma al integrarle al Sub-Mundo Virtual Actualización AVUDONA-Pasivo el conjunto de Actualizaciones de las Animaciones, completando el Ambiente Virtual (pero sin la ayuda). El resultado fue satisfactorio, ya que se logra el nivel de realismo deseado al

integrar a los objetos de este sub-mundo y las animaciones. Al viajar por el sub-mundo resultante el usuario podrá observar a los objetos con sus respectivas animaciones.

Fase 13: Sub-Mundo Actualización AVUDONA_1. Se forma al integrarle al Sub-Mundo Virtual Actualización AVUDONA-Animado el conjunto Actualización Ayuda completando todas las actualizaciones del Ambiente Virtual en su totalidad. Obteniendo un resultado satisfactorio.

Fase 14: Sub-Mundo Actualización AVUDONA-AVUDONA_1. Se forma al integrarle Actualización Mundo AVUDONA el conjunto Mundo AVUDONA_1 completando parcialmente el Ambiente Virtual. El resultado fue satisfactorio, cabe destacar que esta hasta los momentos tenemos una versión parcial de AVUDONA.

Fase 15: Mundo Virtual AVUDONA_2. Esta es la etapa final de la integración, se forma agrupando los sub-mundo virtual Sub-Mundo Virtual Plaza Central “11 de Julio”-Edificio Decanato-Biblioteca el sub-mundo virtual Auditorio-Depto. Ingeniería Industrial-Edificio de Servicios con el conjunto de actualizaciones realizadas a AVUDONA versión 1 y completar en su totalidad el ambiente virtual AVUDONA_2 en su primera versión beta.

El resultado de la integración de AVUDONA_2 se puede visualizar en la grafica de la figura 6.2.

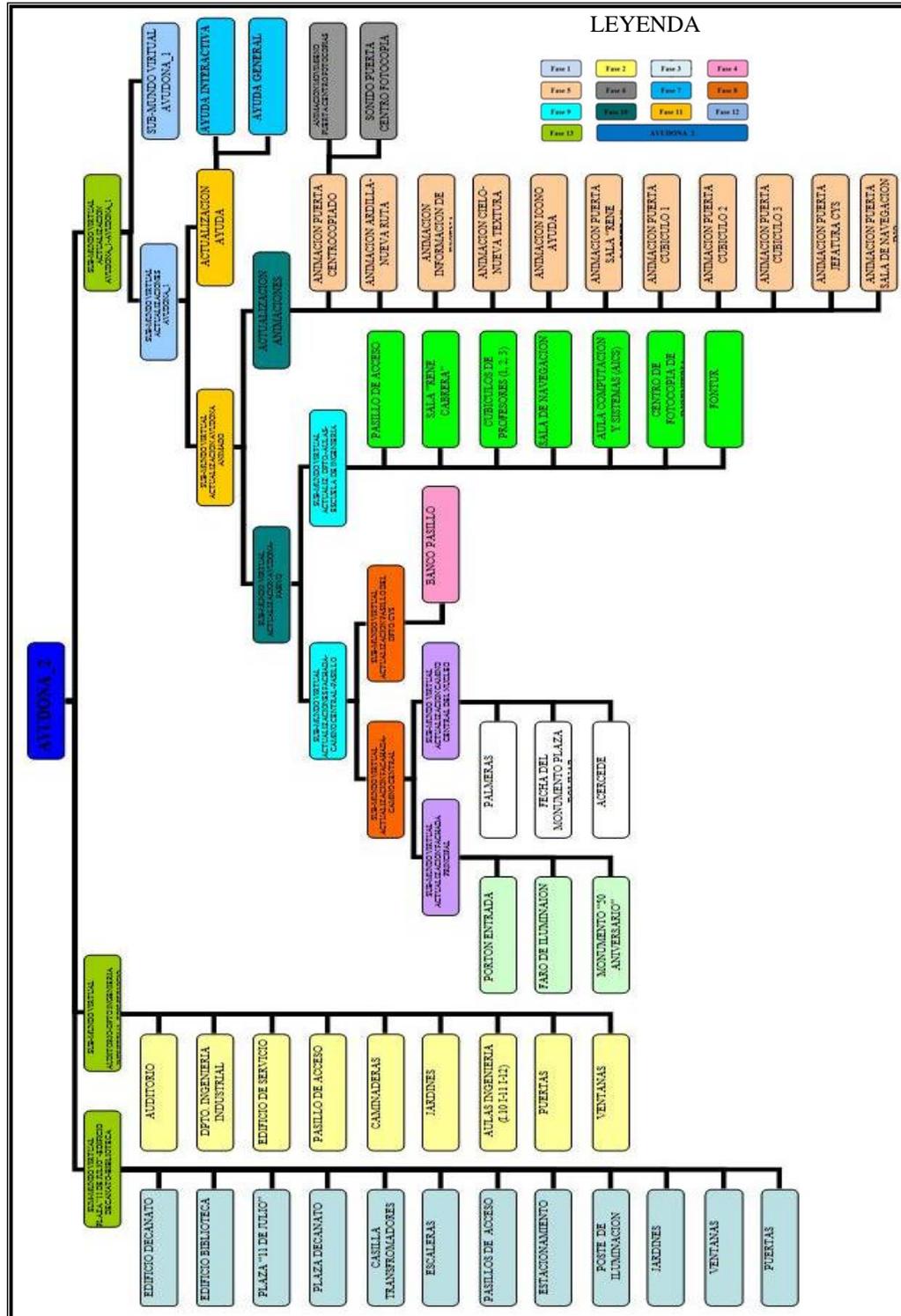


Figura 6.2 Diagrama de Integración de AVUDONA_2

6.5 REALIZAR LA PRUEBA DE INTEGRACIÓN

La prueba de integración realizada se basó en la utilización de sistema del ambiente virtual una vez terminado en su totalidad y haciendo una revisión final de las especificaciones. Se procedió a realizar las pruebas bajo un ambiente el cual permita al usuario final interactuar con el ambiente virtual, teniendo como resultado el perfecto funcionamiento de AVUDONA_2 y a su vez estableciéndose una correcta interacción entre este y sus usuarios. Existió un gran entendimiento y comprensión del mundo virtual con cada uno de los usuarios, los cuales se mostraron complacidos al realizar la navegación identificando las instalaciones reales del Núcleo representado en AVUDONA_2, presentado un fácil manejo de los controles de navegación. Por ello se puede decir que las pruebas concluyeron satisfactoriamente y sin ningún inconveniente que amerite alguna corrección.

6.5.1 PNC Aplicada a la Intergracion de AVUDONA_2

➤ Descripción: AVUDOAN_2 (AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI VERSION).

➤ Verificación:

- Sensibilidad a la presencia del usuario, se logro que los objetos dinámicos definidos en el ambiente virtual reaccionaran a la acción del usuario que se encuentra realizando el recorrido, presentando una adecuada interacción y

comunicación entre el usuario el ambiente virtual y el ambiente virtual entre sí mismo, produciendo la respuesta esperada.

- La alineación de las escenas que forman el ambiente virtual (zonas 1, zona 2, zona 3, zona 4, zona 5, zona 6) es la correcta. Se logro visualizar las escenas una vez integradas al ambiente virtual, corrigiendo las pequeñas distorsiones, esto se repite hasta alcanza la completa alineación de las zonas.

- El color de los objeto tras la integración y aplicación del efecto de luz fue logrado exitosamente con respecto al objeto real y los objetos de las demás zonas del ambiente virtual, se realizaron las correcciones necesaria en aquellos escenas que no armonía con el restos de las escenas

- La posición de las zonas en el ambiente virtual. Se verifica la correcta orientación de las zonas según los ejes (x, y, z).

- El funcionamiento global. Se observo el correcto desplazamiento del usuario en las zonas que conforma el ambiente virtual, velocidad adecuada que permita una clara per sección de las escenas, eventos durante el recorrido y ofrecer la sensación de estar y conocer las instalaciones de la universidad.

- La operabilidad, verificar si las instrucciones para interactuar con el ambiente virtual proporciona a los usuario una clara y fácil comprensión, según su habilidad y destreza. El sistema de ayuda explica detalladamente los controles y como utilizar para realizar el recorrido virtual manteniendo el atractivo del ambiente virtual.

- La población objetivo, se comprobó que las características y habilidades que debe presentar el usuario no son completamente necesario el conocimiento

básico o experiencia con la realidad virtual, solo tener la habilidad o un previo conocimiento con computadoras y videos juegos para interactuar o moverse alrededor del ambiente virtual.

➔ Resultado: Satisfactorio.

CONCLUSIONES

- Este proyecto de Realidad Virtual aplicada a la virtualización de la Universidad de Oriente Núcleo de Anzoátegui, permitió analizar lo que es la realidad virtual y los requerimientos que tienen que tomarse en cuenta para generar aplicaciones computacionales basada en ambientes virtuales.
- La técnica utilizada permitió sistemáticamente aplicar unas series de pasos, dividiendo por etapas el complicado trabajo de modelar ambientes virtuales.
- Se comprobó que la aplicación de la metodología utilizada para el desarrollo de ambientes virtuales, permitió cumplir con todos objetivos planteados.
- Aplicar el estándar UML combinada con la metodología aplicada para desarrollar ambientes virtuales facilito realizar el análisis y diseño del sistema, utilizando diversas herramientas de uso abstracto que permitiera de manera graficas representa el ambiente virtual con el uso de sus diferentes diagramas.
- La utilización de planos 2D reales y digitalizados de la universidad, facilitaron el diseño del ambiente virtual, ofreciendo información sobre las dimensiones, ubicación y escala de los objetos, acercándolos a una escala real.
- Se utilizaron tecnologías modernas y accesibles, que permitieron encontrar la manera de hacerla trabajar conjuntamente para generar las escenas virtuales capaces de reaccionar y responder a las reacciones del usuario.

- La utilización 3D Studio MAX, siendo una herramienta para el diseño 3D, permitió crear y exportar a VRML de manera rápida, precisa y eficiente todos los objetos que conforman el ambiente virtual, facilitando un mayor logro en la calidad de las escenas.

- El lenguaje VRML (*Virtual Reality Modeling Language*) o Lenguaje Modelado de Realidad Virtual, en su especificación 2.0, cumplió con los requerimientos para la realización del ambiente virtual ya que permitió implementar satisfactoriamente las características específicas en su diseño.

- La aplicación de técnicas de optimización, LOD (Nivel de Detalle), permitieron disminuir de manera significativa el retardo innecesario en el procesamiento de las escenas del ambiente virtual, logrando una mayor calidad.

- Se realizó un diseño efectivo de la base de datos mediante técnicas del modelo relacional que facilitó la manera en que se gestionan el acceso al ambiente virtual.

- Se obtuvo un resultado satisfactorio en las pruebas funcionales realizadas permitiendo obtener una retroalimentación sobre el ambiente virtual, garantizando el correcto funcionamiento y la calidad del mismo.

- Se logró un potencial crecimiento del ambiente virtual ofreciendo una interfaz más innovadora que permitió mejorando aspectos visuales, auditivos y táctiles, siendo más atractivo para el usuario final.

RECOMENDACIONES

- Continuar con la virtualización de la Universidad de Oriente Núcleo de Anzoátegui, aplicando e implementando un Ambiente Virtual distribuido de cada uno de las dependencias administrativas y académicas que conforma la Universidad, de tal forma que mejore el acceso a las instalaciones virtuales a distancia a través de la página web de la Universidad.
- Desarrollar una metodología estándar para el desarrollo de ambientes virtuales, sustentado en el VRML 2.0 (Lenguaje Modelado de Realidad Virtual) que permita ser utilizada para el desarrollo de nuevos proyectos similares, ya que el desarrollo de AVUDONA_2 fue satisfactorio.
- Aplicar la Realidad Virtual para el desarrollo de herramienta virtual que puedan ser utilizadas en los laboratorios de Electrónicas y Diseño Usando Microprocesamiento (DUM) del Departamento de Computación y Sistema.
- Desarrollar técnica que mejore la implementación de una base de datos que permita la conectividad con el VRML y generar las escenas de manera dinámicas del objeto que se desea modelar.
- Se recomienda utilizar el Browser Cortona Cliente 4.1 de Parallel Graphics, los navegador Internet Exploret versión 6 en adelante o Netscape Communicator 4.5 en adelante y como entorno físico (Hardware) se recomienda usar, computadora Pentium 4, con un mínimo de memoria 256 Mb RAM y tarjeta grafica de 128Mb mínimo.

BIBLIOGRAFÍA

1. Abásolo M. **“Creación de Mundos Virtuales con VRML 97”**. Universidad Nacional del Centro de la Provincia de Buenos Aires. (1997). <http://www.virtual.unal.edu.co/cursos/artes/2003259/cursovrml/indice.htm>.
2. Arcia J. y Benítez A. **“Desarrollar Un Ambiente Virtual Prototipo De La Universidad De Oriente Núcleo De Anzoátegui”**. Tesis (1996).
3. D. Burgos. **“3D Studio Max Práctico. Guía de aprendizaje”**. Editorial McGraw-Hill, (1997).
4. Domínguez. **“Auto CAD 14. Iniciación y referencia”**. Editorial McGraw-Hill. (1997).
5. Gonzalo Vélez. **“Curso VRML”. [ZONA iAZ, CAD]”**. (1998). <http://www.datarq.fadu.uba.ar/sigradi/expo/gvelez/mirror-iaz/entrega10.html>.
6. José Emilio. **“Metodología de Desarrollo VRML”**. <http://www.jose-emilio.com/estudios/mlmetodologia.htm>
7. K. Jamsa **“VRML Biblioteca del Programador”**, Editorial McGraw-Hill, (1998).
8. Lamar Craig. **“UML Y PATRONES Introducción al Análisis y diseño Orientado a Objeto”**. Editorial Pentrice Hall. (1999).
9. Larijani, Casey. **“Realidad Virtual”**, Editorial McGraw-Hill, (1994).

10. **“Modelos VRML de la CUC”**. Universidad Central de Venezuela FAU http://copred.rect.ucv.ve/4DAction/SubWeb_VRML. (2006)
11. N. Fulton. **“3D Studio VIZ 3.0 & 3D Studio MAX 3.0”**, Editorial Reviews (2000).
12. Parallel Graphics. **“Vrml Pad 2.0 & 1 Extrusión Editor 1.0”**. Download. mca. <http://www.parallelgraphics.com/products/vrmlpad>.
13. Peral Marín, Sergio Alejandro. **“Integración de HTML, VRML y Java Script para la Navegación en 3 Dimensiones con Sonido Ambiental bajo el Protocolo http”**. Universidad de las Américas, Puebla. Primavera 1997.
14. Pressman Roger. **“Ingeniería de Software un Enfoque Practico”**. Editorial McGraw-Hill. 4º Edición. (1998)
15. Sánchez Santamaría, Héctor. **“Diseño e Implementación de un Sistema de Realidad Virtual Orientado a Internet”**, Universidad del País Vasco, España, (1998).
16. Steve Bain. **“Guía Oficial de Corel Draw 10”**. Editorial McGraw-Hill. 1º edición. 2001
17. <http://www.sc.ehu.es/ccwgamoia/docencia/Hector.htm>

ANEXO

<i>Anexo 1. Manual de Usuario</i>	229
<i>Anexo 2. Manual de Mantenimiento</i>	239

*Anexo 1. Manual de Usuario***COMO NAVEGAR EN EL AMBIENTE VIRTUAL AVUDONA_2**

Para navegar en el Ambiente Virtual AVUDONA_2, es necesario conocer a nivel del usuario como se maneja el mundo virtual. Se debe utilizar el ratón y el teclado del computador para poder moverse con soltura, los cuales servirán para simular el comportamiento de los humanos en la vida real. El monitor del computador como medio que permite visualizar las escenas del mundo.

Para lograr la visualización del Ambiente Virtual se debe utilizar los visores de VRML (Lenguaje Modelado de Realidad Virtual), estos son programas que dan un entorno de navegación para poder dirigirse a cualquier punto e incluso poder interactuar con los objetos de este mundo. El visor usando en AVUDONA_2 es el Cortona VRML Client 4.1.

A continuación se presenta las características del visor 3D, necesario para poder recorrer el Ambiente Virtual.

Los controles de navegador

Cuando se entra en un mundo 3D, Cortona VRML Client 4.1 se lanza automáticamente dentro del navegador. Los controles del navegador se encuentran divididos en dos partes la barra de herramienta y la ventana 3D que muestra el mundo.

Barras de herramientas:

La barra de herramientas vertical, que contiene botones para especificar el tipo de navegación en un mundo.

La barra de herramientas horizontal, que contiene botones predefinidos con las acciones para cambiar su posición en un mundo.

La ventana de 3D, que muestra mundo VRML.

También hay menú emergente, que puede acceder pulsando el botón derecho del ratón mientras el puntero está sobre las barras de herramientas o la ventana 3D. En la figura 1-MU se puede observar una grafica de los controles.

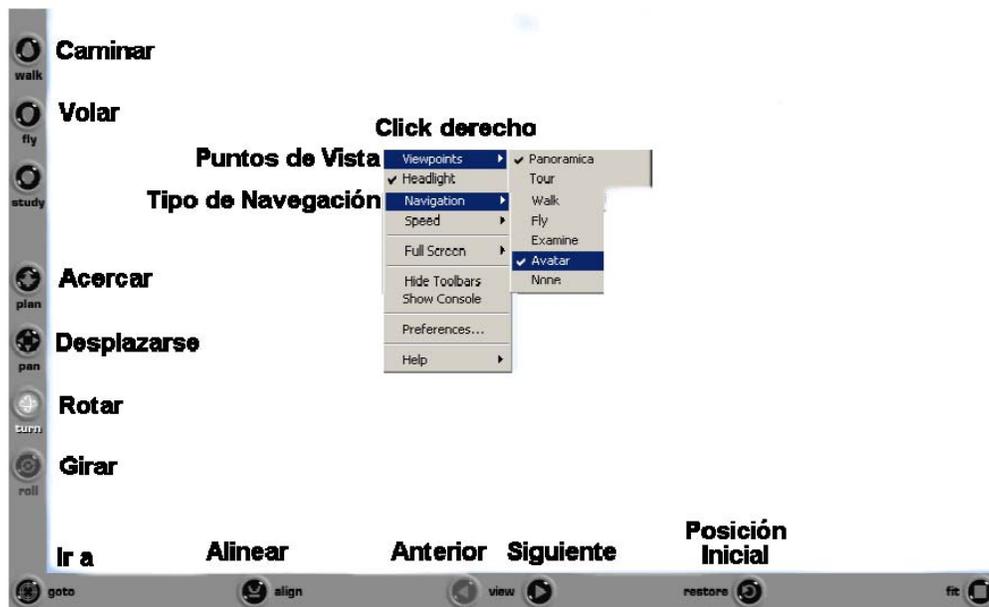


Figura 1-MU Controles del Navegador Cortona VRML Client

Navegando en el Cortona

Moverse a través de un espacio 3D es similar a mover una cámara. Piense en una cámara de video que captura las imágenes en el mundo real y los convierte en los signos electrónicos que se ven en una pantalla; tiene una posición y orientación, y estos son continuamente los movimientos que se aplican. Usted es independiente en el mundo en cuanto a posición y orientación. Use los mandos de la cámara en la Barra de Herramientas vertical para moverse a través del Espacio 3D.

Este concepto asume que hay una persona real viendo y actuando recíprocamente con el Mundo Virtual. El Mundo posee diversos puntos de vista (o cámaras) - lugares interesantes que el usuario podría desear ver. Sólo un punto de vista puede estar activo en un momento. Esta sección describe los mecanismos que el Cortona VRML Cliente mantiene navegando en un espacio tridimensional. Cabe destacar que algunos controles de la Barra de Herramienta estas desactivado, debido que tienen ningún uso para el tipo de ambiente desarrollado, donde se está simulando el efecto de gravedad.

Uso de Puntos de vista

Un punto de vista describe una posición y orientación para ver la escena. El VRML Cortona probablemente va permite guiar al usuario a los mejores punto de vista en mundo virtual. No todos los mundos contienen puntos de vista, pero cuando lo hacen, usted puede utilizarlos.

Para activar un punto de vista, utiliza los siguientes:

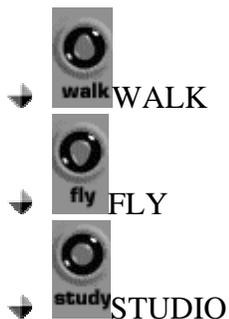


Haga clic en la vista en la barra de herramientas horizontal o puntos de vista de elegir en el menú emergente y, a continuación, seleccione una cámara de la lista de los puntos de vista predefinidos.

En los botones de flechas haga click para pasar a la siguiente o anterior cámara. También puede pulsar Page Up o Page Down botón del teclado.

Moverse por el ambiente virtual

El navegador ofrece tres modos principales para navegar:



Puede cambiar el modo de navegación haciendo click en los botones de la barra de herramientas vertical.

Para este Ambiente virtual AVUDONA 2.0 el navegador solo permite un modo de navegación para simular el efecto gravedad.

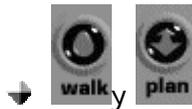
Dispone en la pantalla señales de navegación.

Usted puede navegar con el ratón, el teclado, o ambas cosas. Para desplazarse por el ambiente virtual utilice el ratón realizando los siguientes pasos:

1. Coloque el puntero en cualquier lugar de la ventana 3D y pulse el botón izquierdo del ratón.
2. Mueva el ratón hacia abajo mientras mantiene presionado el botón izquierdo. La dirección en la que usted arrastrar el ratón determina el movimiento de la cámara.
3. Suelte el botón izquierdo del ratón para dejar de moverse.

Nota: La distancia que arrastrar el ratón determina la velocidad con la que la cámara se mueve. Si te dejas de mover el ratón, la cámara seguirá avanzando hasta que suelte el botón del ratón. Para acelerar la cámara o el movimiento de rotación, pulse SHIFT, CTRL + SHIFT o CTRL.

El modo de navegación tiene opciones que determinan el posible movimiento de cámara y su orientación. Tenga en cuenta que para AVUDONA 2.0 se especifico el paradigma de navegación debe utilizado para las escenas. Estas opciones al combinar con el modo de navegación permiten:



Use Walk + plan para moverse horizontalmente por el ambiente virtual. El puntero de ratón cambia según el desplazamiento:

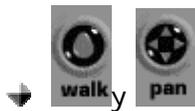
↑ Acercarse - avanzar hacia delante.

↓ Retroceder - avanzar hacia atrás.

↻ Derecho - giro a la derecha.

↻ Izquierda - giro a la izquierda.

Nota: Mueva el ratón hacia delante o hacia atrás mientras mantiene pulsada la tecla **espacio** a su vez hacia arriba, hacia abajo. Mueva el ratón hacia la izquierda o derecha mientras mantiene pulsada la tecla **ALT** para mover hacia la izquierda o la derecha.



Use Walk + pan para moverse horizontalmente por el ambiente virtual. El puntero de raton cambia según el desplazamiento:

↑ Acercarse - mover hacia delante

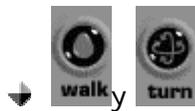
↓ Retroceder - moverse hacia atrás

⇒ Derecho - mover hacia la derecha

⇐ Izquierda - mover a la izquierda

Nota: Cuando usted mueve la cámara derecha o la izquierda, el mundo parece moverse en la dirección opuesta.

Debido a las características de AVUDONA 2.0 en cuanto al paradigma de navegación definido, se recomienda utilizar solo los dos modos explicados anteriormente.



Use Walk + turn para cambiar el ángulo de la cámara en un ambiente virtual.

 Alzar - Girar hacia arriba.

 Bajar - Girar hacia abajo.

 Derecho - Girar a la derecha.

 Izquierda - Girar a la izquierda.



Utilice GOTO para moverse cerca de un objeto en el ambiente virtual. Haga click sobre GOTO en la barra de herramientas horizontal y, a continuación, haga clic en un objeto del ambiente virtual. Usted pasar directamente a él.

Restore, Fit, and Align (Restaurar, Apropriado, y Alinear).

Cortona VRML Cliente ofrece tres mecanismos que pueden ayudar a re-orientar la cámara si usted ha perdido su camino en el ambiente virtual. A diferencia de las herramientas de navegación, estos botones predefinidos invocar las acciones que se llevan a cabo al hacer clic sobre ellos.



Utilizar Restaurar para automáticamente volver a la carga del mundo activo desde la posición inicial de la cámara activa.



Use fit para tener hacer plenamente una vista de todas las escenas del mundo virtual en el VRML Cortona Cliente ventana 3D. Esta es una función del navegador cortona, este control no es recomendado utilizar por el paradigma de navegación (no se recomienda utilizar este control por que desconfigura el ambiente virtual).



Utilice **ALIGN** para restaurar la altura de la cámara según el eje horizontal longitudinal y en paralelo a la escena o plano horizontal.

Usando el teclado para navegar.

Usted puede controlar la cámara usando comandos de teclado. La funcionalidad de los botones corresponde a los movimientos de su ratón y depende del tipo de navegación.

- ➔ **Flecha arriba** - Acercarse.
- ➔ **Flecha Abajo** - Avanzar hacia atrás.
- ➔ **Flecha derecha** - Gire a la derecha.
- ➔ **Flecha izquierda** - Gire a la izquierda.
- ➔ **Flecha Arriba en el teclado numérico** - Acercarse.
- ➔ **Flecha Abajo en el teclado numérico** - Avanzar hacia atrás.
- ➔ **Flecha Derecha en el teclado numérico** - mover a la derecha.
- ➔ **Flecha Izquierda en el teclado numérico** - mover a la izquierda.
- ➔ **7 en la teclado numérico** - Gire a la cámara hacia abajo en torno a su eje horizontal.
- ➔ **9 en el teclado numérico** - Gire la cámara hacia arriba alrededor de su eje horizontal.
- ➔ **1 en el teclado numérico** - Inclinación a la derecha. Nota: No está disponible en modo MARCHA.
- ➔ **3 en el teclado numérico** - Inclinación a izquierdo. Nota: No está disponible en modo MARCHA.
- ➔ **1 en el teclado alfanumérico** - inclinación a la derecha. Nota: No está disponible en modo MARCHA.
- ➔ **2 en el teclado alfanumérico** - inclinación a la izquierda. Nota: No está disponible en modo MARCHA.
- ➔ **Page Down** - Cámara siguiente de la lista de punto de vista.
- ➔ **Page Up** - Cámara anterior de la lista de punto de vista.

✦ **Pausa** - Activa o desactiva CylinderSensor, PlaneSensor, SphereSensor, y TouchSensor en Cortona VRML Cliente.

Para acelerar la cámara o el movimiento de rotación: Pulse **SHIFT**, **CTRL** o **SHIFT + CTRL** y una de las claves más arriba simultáneamente.

Las teclas **ALT** y **ESPACIO** permiten al usuario cambiar rápidamente la opción de navegación PAN y Turn en consecuencia en el modo de navegación FLY

Anexo 2. Manual de Mantenimiento

AVUDONA (AMBIENTE VIRTUAL PROTOTIPO DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI)

Es un ambiente virtual en cual el usuario puede visitar las instalaciones de la universidad utilizando tecnología WWW o internet. Está diseñado para ser publicada en página web de la Universidad y ser acezado por el usuario visualizar a través de un Browser que permite automáticamente cargar el archivo AVUDONA_25_ACTUALIZACION.

ZONAS Y ESCENAS REPRESENTADA EN AVUDONA_2

En el ambiente virtual, se considero subdividir las en módulos o zonas para separar físicamente unas de las otras y sea más sencillo diseñar en forma paralela. Para AVUDONA_2 se dividió el ambiente virtual en 6 zonas, las cuales se expresan a continuación:

- Zona 1: Fachada de la Entrada Principal del Núcleo.

- Zona 2: Tramo Central que va Desde la Entrada Principal de la UDO Hasta los Pasillos que llevan a las Instalaciones de Ingeniería (hasta el Auditorio) y Decanato.

- Zona 3: Pasillos que llevan a las distintas Instalaciones del Departamento de Computación y Sistemas (Salas de Micros, Cubículos, Secretaría, Laboratorios),

Departamento de Industrial y Auditorio, Decanato, Plaza Decanato, Biblioteca “Luis García Pelissier” y Plaza Central “11 de Julio”.

➤ Zona 4: Las Aulas de Ingeniería (desde la I-1 hasta I-9), Departamento de Computación y Sistemas, Dirección de Escuela de Ingeniería, Centro de Fotocopiados, Sala de Navegación de la Esc. de Ingeniería.

➤ Zona 5: Las Aulas de Ingeniería (desde la I-10 hasta I-12), Departamento de Industrial, Auditorio, Baños y Aéreas de Servicio de Ingeniería.

➤ Zona 6: Biblioteca “Luis García Pelissier”, Plaza Central, Decanato, Plaza Decanato, jardines y alrededores.

MIODELO DE CASO DE USO

Esta orientados a la representación grafica de requerimientos en este caso de AVUDONA_2. Los casos de uso son graficas de fragmentos de funcionalidad que el sistema ofrece cuando un usuario interactúa con una entidad y el sistema. Figura 1.

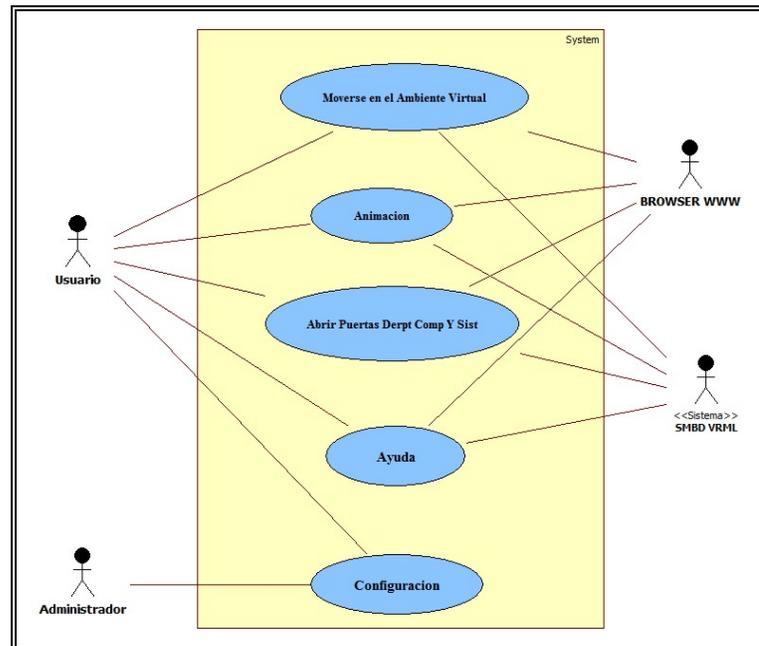


Figura 1. Caso de Uso AVUDONA_2

Identificación y Descripción de los Actores

Tabla 1 Identificación y Descripción de los Actores de AVUDONA_2.

Actor	Descripción
Usuario	Este representa al usuario en general que navega por el ambiente virtual, el cual podrá recorrer de diferentes formas que permite el sistema, ya sea, Viajando, Moviéndose en el Ambiente Virtual, Entrando a las diferente Salas, modificando la configuración para el recorrido y acceder a la ayuda del ambiente virtual en el momento que lo necesite.
Administrador	Este actor tendrá cierto privilegio dentro el sistema, ya que su principal función es realizar operaciones, tales como, agregar nuevas escenas, cambiar el comportamiento y funcionamiento del ambiente virtual por medio del modulo <i>Configuración</i> .
Browser VRML	El actor Browser VRML, que no es más que la herramienta de

	software encargado de relacionar a los usuarios con el Ambiente Virtual, es decir, permite la visualización del mundo, la interacción de los usuarios con los objetos presentes en el mundo, la activación y ejecución de animaciones, además de permitir la transmisión e interrelación del mundo con la WWW (si es remoto) por medio de la conexión con el Browser WWW a través de un API.
SMBD VRML <<Sistema>>	Es el Software encargado de la conexión con la base de datos, y el almacén que contiene la descripción del mundo, como los objetos que se encuentran dentro de él, los guiones y variables de ambiente.

Archivos de AVUDONA_2

El directorio donde se encuentra almacenado los archivos son:

Carpeta **AVUDONA 2**:

Contiene todos los archivos necesarios para el funcionamiento del Ambiente Virtual, en esta carpeta se encuentra:

Carpeta **AVUDONA 2**. Contiene (dos) 2 archivos, **AVUDONA_25_ACTUALIZACION.wrl**, archivo VRML, Ambiente Virtual. **1008-2.htm**, interfaz que permite el acceso al archivo **AVUDONA_25_ACTUALIZACION.wrl**

Carpeta **htm**, directorio que almacena los archivo de ayuda utilizados en **AVUDONA_25_ACTUALIZACION.wrl**, este se referencia desde el código VRML.

Carpeta **img**, directorio que almacena todas las texturas e imágenes utilizadas en el Ambiente Virtual.

Carpeta ***round***, directorio que almacena todas los archivos de audio utilizados en el Ambiente Virtual.

Archivo ***PresentacionFinal.php***, este archivo es la interfaz utilizada para realizar la presentación del Ambiente Virtual, a través de esta página web el usuario activa o carga el archivo AVUDONA_25_ACTUALIZACION.wrl. A continuación se mostrara el código:

```

<!--Este código no forma de la estructura funcional de AVUDONA-2 solo es utilizado
como página Web provisional para activar el ambiente virtual y enlazar con la base de datos-->
<!-- Interfaz que permite el acceso a archivo *.wrl -->
<html>
<head>
<title>Pagina Ambiente Virtual</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#000000" >
<font color="#FFFFFF">

<?php
// codigo Php utilizado unicamente para realizar la conexion con la base de datos
// en la interfaz provisional para activar el ambiente virtual.
// aqui se realiza la conexion con la base de datos SOFIA y se realiza la consulta
$connectando = mssql_connect('SOFIA','SA','') or die ('Fallo la Conexion con la Base de Datos');
mssql_select_db('AVUDONA') or ('Fallo... Base de Ddato no existe');
$resultaDato = mssql_query("select * from Ambiente_Virtual ",$connectando) or
die('Fallo...Dato no Existe');
$DatoColumna = mssql_fetch_array($resultaDato);
// fin del codigo
?>
</font>
<table border="0" height="0" align="center" bordercolor="#FFFFFF" bgcolor="#000000">
<tr>

```

```

        <td colspan="2" height="156">
            <div align="center"></div>
        </td>
    </tr>
    <tr>
        <td height="33" colspan="2">
            <div align="center"><font color="#FFCC00">AMBIENTE VIRTUAL DE LA
UNIVERSIDAD
            DE ORIENTE NUCLEO ANZOATEGUI</font></div>
        </td>
    </tr>
    <tr>
        <td height="20">
            <div align="center"><font size="-1"></font></div>
        </td>
        <td height="20">
            <div align="left"><font size="-1" color="#CCCCCC">La visualizaci&oacute;n
            de los modelos 3D en VRML require la instalaci&oacute;n de un plug-in
            (Cliente Cortona) haga click en el siguiente enlace</font><font size="-1"
color="#999999">
                </font><font size="-1"><a href="http://www.cortona3d.com/cortona">Descargar
                visualizador</a> </font></div>
        </td>
    </tr>
    <tr>
        <td colspan="2" height="269">
            <div align="left">
                <table width="99%" border="0" bgcolor="#000000" bordercolor="#000000">
                    <tr>

```

El siguiente código hacer la consulta a la Base de Datos que tiene almacenado la ruta de acceso del archivo que permite el enlace con el archivo

AVUDONA_25_ACTUALIZACION.wrl. La ruta de acceso debe corresponde al directorio específico donde se encuentra almacenado el archivo .wrl (Ambiente Virtual)

<!-- en esta parte del código utilizando el comando echo incluimos el dato de la consulta con la ruta de acceso al archivo (.wrl)-->

```

        <td width="40%" height="266"><a href="<?php echo $DatoColumna[ruta_Acceso']
?>" title=" Haga clic para activas" target="_top">
        </a></td>
        <td width="60%" height="266">
        <div align="justify"><b><font
color="#CCCCCC">Ambiente
        Virtual, UDO-ANZ</font></b><font color="#CCCCCC">. En el recorrido
        virtual se podrá visualizar una representación de las instalaciones
        del recinto universitario desde la Entrada Principal, Edificio
        de Biblioteca &quot;Luis Garcia Pellisier&quot; y Decanato hasta
        la Escuela de Ingeniería (Dpto Ingeniería Computación y Sistema,
        Dpto Ingeniería Industrial) y sus alrededores. Realizado por Br:
        José Millán. /Tama&ntilde;o 24 Mb/ </font></div></td>
    </tr>
</table>
<div align="left"><font color="#FFCC33"><b>Para activar el Ambiente virtual
haga click sobre la imagen.</b></font></div>
</div>
</td>
</tr>
</table>
</body>
</html>

```

A continuación se muestra parte del código de la interfaz que permite el enlace con el archivo que AVUDONA_25_ACTUALIZACION.wrl.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

<html>

Embed instrucción utilizada por Cliente VRML Cortona en el documento HTML para mostrar el Mundo Virtual.

<body>

<!--esta intrccion le permite el acceso al archivo AVUDONA_25_Actualizacion.wrl a través del documento HTML -->

<embed src="AVUDONA_25_Actualizacion.wrl" width="100%" height="100%"></embed>

</body>

</html>

EL LENGUAJE DE MODELADO DE MUNDOS VIRTUALES EN 3D, VRML (LENGUAJE PARA MODELADO DE REALIDAD VIRTUAL)

"El Lenguaje de Modelación de Realidad Virtual (VRML) es aquel orientado a la descripción de simulaciones interactivas de multiparticipación en mundos virtuales conectados vía Internet hipervinculados con la WWW (World Wide Web). Todos los aspectos de representación, interacción e interconexión en la red pueden ser especificados utilizando VRML. Es la intención de sus diseñadores que VRML pueda convertirse en el lenguaje estándar para simulación interactiva dentro de la WWW".

Componente de un Fichero

Un fichero VRML puede contener cuatro tipos de componentes:

- ENCABEZADO ("header").
- PROTOTIPOS.
- NODOS: FORMAS, INTERPOLADORES, SENSORES Y GUIONES ("Scripts").
- RUTAS O ENRRUTAMIENTO DE EVENTOS.

No todos los ficheros deben contener todos estos componentes. De hecho, el único obligatoriamente requerido es el encabezado. Por otra parte, y a nivel más desagregado, un fichero VRML puede contener: comentarios, nodos y sus nombres, campos y sus valores.

Sintaxis Básica.

A continuación podrá observar el código de *AVUDONA_25_Actualizacion.wrl*.

El encabezado

Es obligatorio iniciar con la siguiente línea:

```
#VRML V2.0 utf8
# Produced by 3D Studio MAX VRML97 exporter, Version 7, Revision 0,65
# MAX File: AVUDONA_2_ACTUALIZACION, Date: Sat Nov 01 19:55:55 2008
```

Información referencian sobre el diseño de Ambiente, Titulo, Autor, Institución si es el caso;

```
DEF info WorldInfo {
  title "AVUDONA 2 VRML97 "
```

```

info ["Trabajo de Grado Universidad de Oriente:"
      "ACTUALIZACION Y EXPACION DEL AMBIENTE VIRTUAL PROTOTIPO DE LA"
      "UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI"]
}

```

Nodos (Grafos de Escenas)

Son los objetos que describen el contenido de las escenas. Describen el nodo de navegación del usuario:

```

DEF NavegacionPrincipal NavigationInfo {
  avatarSize [0.25, 1.6, 0.75]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}

DEF NavegacionOficina NavigationInfo {
  avatarSize [0.25, 1.9, 0.75]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}

DEF NavegacionSalaMicroRN NavigationInfo {
  avatarSize [0.1, 1.6, 0.5]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}

```

```

DEF SalaNavegacion NavigationInfo {
  avatarSize [0.25, 1.6, 0.5]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}
DEF NavegacionCubiculo1 NavigationInfo {
  avatarSize [0.1, 1.6, 0.5]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}
DEF NavegacionCubiculo2 NavigationInfo {
  avatarSize [0.1, 1.6, 0.5]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}
DEF NavegacionCubiculo3 NavigationInfo {
  avatarSize [0.1, 1.6, 0.5]
  headlight FALSE
  speed 2
  type "WALK"
  visibilityLimit 0
}

```

Describen el fondo o Entorno del Ambiente virtual;

```

# PANORAMAS
DEF PanoramaDia Background {
  groundColor [0.66 0.88 0.97]

```

```

    groundAngle 1.5 skyColor [0.66 0.88 0.97]
    skyAngle 1.64
    backUrl ""
    frontUrl "../img/PanoramaDia2.jpg"
    rightUrl ""
    leftUrl "../img/PanoramaDiaIzq2.jpg"
    bottomUrl ""
}
DEF PanoramaNoche Background {
    groundColor [0 0 0]
    groundAngle 1.5 skyColor [0 0 0]
    skyAngle 1.64
    backUrl ""
    frontUrl "../img/PanoramaNoche2.jpg"
    rightUrl ""
    leftUrl "../img/PanoramaNoceIzq2.jpg"
    bottomUrl ""
}
}

```

Los nodos a continuación describen las escenas que conforman el Ambiente Virtual, Luces, Edificios, Puertas, Pared, Jardines, Cámaras, Nodo de Agrupamiento, etc.

```

DEF Camara_Entrada_UDO Viewpoint {
    position 5.422 1.5 13.51
    orientation 0 1 0 0
    fieldOfView 0.7363
    description "Camara Entrada UDO"
}
DEF Camara_Plaza_Bolivar Viewpoint {
    position 3.294 1.5 -94.42
    orientation 1 0 0 0
    fieldOfView 0.7363
}

```

```
description "Camara Plaza Bolivar"  
}  
DEF CameraPlazaCentral01 Viewpoint {  
position 4.921 2.346 -170.5  
orientation 0 0 0  
fieldOfView 0.7363  
description "Camera Plaza Central"  
}  
DEF CameraPlazaDecanato01 Viewpoint {  
position -49.88 3.064 -171.5  
orientation 0 -1 0 -0.8954  
fieldOfView 0.7363  
description "Camera Plaza Decanato"  
}  
DEF SalaMicroRC Viewpoint {  
position 29.56 0.2 -148.5  
orientation 0 1 0 -1.571  
fieldOfView 0.7363  
description "Sala de Micros 'RENE CABRERA"  
}  
DEF Camara_Aulas_Ing_ Viewpoint {  
position 53.52 0.5 -167  
orientation 0 1 0 -0.6299  
fieldOfView 0.7363  
description "Camara Aulas Ing."  
}  
DEF CameraCentroFotoCopia Viewpoint {  
position 62.1 0.23 -133  
orientation 1 0 0  
fieldOfView 0.7363  
description "Camera Centro FotoCopiado"  
}  
DEF Jefatura Viewpoint {  
position 53.14 0.5 -158.8
```

```

orientation 0 -1 0 -3.142
fieldOfView 0.7363
description "Camara Dpto Comp. y Sist. "
}
DEF CameraSalaNavegacion Viewpoint {
position 73.73 0.5386 -155.1
orientation 0 -1 0 -3.142
fieldOfView 0.7363
description "Camera Sala de Navegacion"
}
DEF CameraAnfiteatro01 Viewpoint {
position 87.89 -0.577 -159.6
orientation 0 -1 0 -4.101
fieldOfView 0.7363
description "Camera Anfiteatro"
}
DEF CameraDecanato01 Viewpoint {
position -98.809 5.102 -221.2
orientation 0 -1 0 -4.712
fieldOfView 0.7363
description "Camera Edif. Decanato"
}
DEF Acercade Viewpoint {
position 7.665 1.4 -159.3
orientation 0 1 0 -0.0004022
fieldOfView 0.7363
description "Acercade"
}
DEF Oficina Viewpoint {
position 53.73 0.854 -149.3
orientation 0 -1 0 -1.577
fieldOfView 0.6486
}
DEF Cubiculo01 Viewpoint {

```

```
position 53.73 0.8537 -148
orientation 0 -1 0 -1.577
fieldOfView 0.6486
}
```

```
DEF Cubiculo02 Viewpoint {
position 52.6 0.8537 -148
orientation 0 1 0 -1.566
fieldOfView 0.6486
}
```

```
DEF Cubiculo03 Viewpoint {
position 52.6 0.8537 -149.3
orientation 0 1 0 -1.566
fieldOfView 0.6486
}
```

```
DEF Omni701 PointLight {
intensity .6
color 0.7059 0.7059 0.7059
location -473.1 75 -615.2
on TRUE
radius 713
}
```

```
DEF Omni702 PointLight {
intensity .6
color 0.7059 0.7059 0.7059
location 460.7 75 -600.4
on TRUE
radius 713
}
```

```
DEF Omni703 PointLight {
intensity .6
color 0.7059 0.7059 0.7059
location 460.7 75 259.6
on TRUE
}
```

```

    radius 713
}
DEF Omni704 PointLight {
    intensity .6
    color 0.7059 0.7059 0.7059
    location -473.1 75 244.8
    on TRUE
    radius 713
}
DEF Omni705 PointLight {
    intensity .8
    color 0.7059 0.7059 0.7059
    location 210.5 -79.4 -125.1
    on TRUE
    radius 713
}
DEF Omni706 PointLight {
    intensity .8
    color 0.7059 0.7059 0.7059
    location -134.3 -79.4 -124.9
    on TRUE
    radius 713
}

```

Nodo de Agrupamiento OficinaCYS, objeto que representa la oficina principal del Departamento de Computación Y sistemas.

```

##### Oficina Jefe CYS #####
DEF OficinaCYS Group{
    children[
    Transform {
    translation 48.35 0 -150.3
    children [

```

```
DEF vision_jefe Viewpoint {
position -50.54 1.6 -52.27
orientation -6.534e-005 -0.9999 0.01319 -3.132
fieldOfView 0.6024
description "vision_jefe"
}
DEF camOficina Viewpoint {
position -50.29 2 -45.51
orientation 0 -1 0 -0.005326
fieldOfView 0.6024
description "camOficina"
}
DEF Omni15 PointLight {
intensity 1
color 1 1 1
location -48.97 10.07 -35.9
on TRUE
radius 68.68
}
DEF Omni16 PointLight {
intensity 1
color 1 1 1
location -38.65 10.07 -59.06
on TRUE
radius 68.68
}
DEF Omni03 PointLight {
intensity 1
color 1 1 1
location -59.04 10.07 -59.26
on TRUE
radius 68.68
}
DEF Omni07 PointLight {
```

```

intensity 0.5
color 1 1 1
location -48.9 -12.56 -48.82
on TRUE
radius 68.68
}
DEF pared_der Transform {
translation -54.11 2.43 -54.69
rotation -1 0 0 -1.571
scale 1.78 1.78 1.78
children [
DEF pared_der-TIMER TimeSensor { loop TRUE cycleInterval 3.333 },
Shape {
appearance Appearance {
material Material {
diffuseColor 0.5 0.5 0.5
ambientIntensity 0.1
specularColor 0 0 0
shininess 0.05
transparency 0
}
texture ImageTexture {
url "../img/ladrillo.JPG"
}
}
geometry DEF pared_der-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF pared_der-COORD Coordinate { point [
-0.05 0 1.38, 0.05 0 1.38, -0.05 0 -1.38, 0.05 0 -1.38, -0.05 5.5 1.38,
0.05 5.5 1.38, -0.05 5.5 -1.38, 0.05 5.5 -1.38]
}
texCoord DEF pared_der-TEXCOORD TextureCoordinate { point [
-0.39 -0.1141, 1.39 -0.1141, -0.39 1.114, -0.5098 1.39, 2.058 -0.1141,

```

```

-1.058 -0.1141, 2.058 1.114, -0.5098 -1.058, 1.551 -0.1141, 1.551 1.114,
1.51 1.114, 1.51 1.114, 1.51 -0.1141, 1.551 -0.1141, -0.5512 -0.1141,
-0.5098 -0.1141, -0.5098 1.114, -0.5098 1.114, -0.5512 1.114,
-0.5512 -0.1141, 1.551 1.39, 1.51 1.39, 1.51 -1.058, 1.51 -1.058,
1.551 -1.058, 1.551 1.39, 1.39 -0.1141, 1.39 1.114, -1.058 1.114,
-1.058 1.114, -0.5098 1.39, -0.5512 1.39, -0.5512 -1.058, -0.5512 -1.058,
-0.39 1.114, 2.058 -0.1141]
}
coordIndex [
0, 2, 3, -1, 3, 1, 0, -1, 4, 5, 7, -1, 7, 6, 4, -1, 0, 1, 5, -1,
5, 4, 0, -1, 1, 3, 7, -1, 7, 5, 1, -1, 3, 2, 6, -1, 6, 7, 3, -1,
2, 0, 4, -1, 4, 6, 2, -1]
texCoordIndex [
8, 9, 10, -1, 11, 12, 13, -1, 14, 15, 16, -1, 17, 18, 19, -1,
20, 21, 22, -1, 23, 24, 25, -1, 26, 27, 28, -1, 29, 5, 1, -1,
30, 31, 32, -1, 33, 7, 3, -1, 34, 0, 35, -1, 4, 6, 2, -1]
]]]
DEF pared_izq Transform {
translation -45.26 2.43 -54.69
rotation -1 0 0 -1.571
scale 1.78 1.78 1.78
children [
DEF pared_izq-TIMER TimeSensor { loop TRUE cycleInterval 3.333 },
Shape {
appearance Appearance {
material Material {
diffuseColor 0.5 0.5 0.5
ambientIntensity 0.1
specularColor 0 0 0
shininess 0.05
transparency 0
}
texture ImageTexture {
url "../img/ladrillo.JPG"

```

```

}}
geometry DEF pared_izq-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF pared_izq-COORD Coordinate { point [
-0.05 0 1.38, 0.05 0 1.38, -0.05 0 -1.38, 0.05 0 -1.38, -0.05 5.5 1.38,
0.05 5.5 1.38, -0.05 5.5 -1.38, 0.05 5.5 -1.38]
}
texCoord DEF pared_izq-TEXCOORD TextureCoordinate { point [
-0.39 -0.1141, 1.39 -0.1141, -0.39 1.114, 1.551 1.39, 2.058 -0.1141,
-1.058 -0.1141, 2.058 1.114, 1.551 -1.058, -0.5098 -0.1141, -0.5098 1.114,
-0.5512 1.114, -0.5512 1.114, -0.5512 -0.1141, -0.5098 -0.1141,
1.51 -0.1141, 1.551 -0.1141, 1.551 1.114, 1.551 1.114, 1.51 1.114,
1.51 -0.1141, -0.5098 1.39, -0.5512 1.39, -0.5512 -1.058, -0.5512 -1.058,
-0.5098 -1.058, -0.5098 1.39, 1.39 -0.1141, 1.39 1.114, -1.058 1.114,
-1.058 1.114, 1.551 1.39, 1.51 1.39, 1.51 -1.058, 1.51 -1.058,
-0.39 1.114, 2.058 -0.1141]
}
coordIndex [
0, 2, 3, -1, 3, 1, 0, -1, 4, 5, 7, -1, 7, 6, 4, -1, 0, 1, 5, -1,
5, 4, 0, -1, 1, 3, 7, -1, 7, 5, 1, -1, 3, 2, 6, -1, 6, 7, 3, -1,
2, 0, 4, -1, 4, 6, 2, -1]
texCoordIndex [
8, 9, 10, -1, 11, 12, 13, -1, 14, 15, 16, -1, 17, 18, 19, -1,
20, 21, 22, -1, 23, 24, 25, -1, 26, 27, 28, -1, 29, 5, 1, -1,
30, 31, 32, -1, 33, 7, 3, -1, 34, 0, 35, -1, 4, 6, 2, -1]
}}}]
DEF techo Transform {
translation -49.69 4.702 -49.8
scale 1.78 1.78 1.78
children [
Transform {
translation 0 0.05 0
children [

```

```
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0.8471 0.8471 0.8471
    }
  }
  geometry Box { size 5 0.1 5.5 }
} } }
DEF lampara Transform {
  translation -49.69 4.613 -51.58
  scale 1.78 1.78 1.78
  children [
    Transform {
      translation 0 0.025 0
      children [
        Shape {
          appearance Appearance {
            material Material {
              diffuseColor 1 1 1
              ambientIntensity 0.1115
              specularColor 0.045 0.045 0.045
              shininess 0.2875
              transparency 0
              emissiveColor 1 1 1
            }
          }
          geometry Box { size 1 0.05 2 }
        } } }
      DEF lampara_0 Transform {
        translation -49.45 1.572 -50.74
        rotation 0 1 0 -1.553
        scale 0.6141 0.6141 0.6141
        children [
          DEF BASE_BLACK Transform {
            translation 0 -0.525 0
```

```

children [
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0.2275 0.2275 0.2275
        ambientIntensity 0.2275
        specularColor 0.09098 0.09098 0.09098
        shininess 0.297
        transparency 0
      }
    }
    geometry DEF BASE_BLACK-FACES IndexedFaceSet {
      ccw TRUE
      solid TRUE
      coord DEF BASE_BLACK-COORD Coordinate { point [...]
    }
    coordIndex [...]
  }
}],
DEF BASE_CHROM Transform {
  translation 0 -0.525 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.749 0.749 0.749
          ambientIntensity 0.749
          specularColor 0.6891 0.6891 0.6891
          shininess 0.7625
          transparency 0
        }
      }
      + geometry DEF BASE_CHROM-FACES IndexedFaceSet {
        ...
      }
    }
  ]
},
DEF BASE_RED_P Transform {

```

```

translation 0 -0.525 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.8314 0 0
ambientIntensity 0.2771
specularColor 0.6069 0 0
shininess 0.544
transparency 0
}
}
geometry DEF BASE_RED_P-FACES IndexedFaceSet {
...}
coordIndex [...]
}}]}}
DEF papelera Transform {
translation -47.37 0.6018 -52.63
children [
DEF BANDBOT Transform {
translation -1.276 -0.587 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.007843 0.0902
ambientIntensity 0
specularColor 0.87 0.87 0.87
shininess 0.753
transparency 0
}
}
geometry DEF BANDBOT-FACES IndexedFaceSet {
ccw TRUE

```

```
solid TRUE
coord DEF BANDBOT-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF BARREL Transform {
translation -1.276 -0.587 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
}
geometry DEF BARREL-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF BARREL-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF TRASHBIN Transform {
translation -1.276 -0.587 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
```

```

    shininess 0.753
    transparency 0
  }
}
geometry DEF TRASHBIN-FACES IndexedFaceSet {
  ccw TRUE
  solid TRUE
  coord DEF TRASHBIN-COORD Coordinate { point [...]
  }
  coordIndex [...]
  }},
DEF Cylinder01 Transform {
  translation -1.267 0.3806 -0.001869
  children [
  Transform {
    translation 0 0.025 0
    children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.6745 0.6745 0.6745
          ambientIntensity 0
          specularColor 0.91 0.91 0.91
          shininess 0.753
          transparency 0
        }
      }
    }
  ]
}
geometry Cylinder { radius 0.26 height 0.05 }
} ] ] ] ]
DEF Escritorio Transform {
  translation -50.5 0.6178 -50.66
  rotation 0 -1 0 -3.142
  scale 2.5 1.7 1.942
  children [

```

```

DEF ASAS_____ Transform {
translation -0.01157 -0.08447 0.008392
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.898 0.898 0.898
ambientIntensity 0.898
specularColor 0.5837 0.5837 0.5837
shininess 0.4775
transparency 0
}
}
geometry DEF ASAS_____ -FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF ASAS_____ -COORD Coordinate { point [...]
}
coordIndex [...]
}}],
DEF ASAS_____01 Transform {
translation -0.8367 -0.08493 0.008392
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.898 0.898 0.898
ambientIntensity 0.898
specularColor 0.5837 0.5837 0.5837
shininess 0.4775
transparency 0
}}
geometry USE ASAS_____ -FACES
}],

```

```

DEF basedel Transform {
translation -0.002775 -0.008497 -0.2608
rotation 0 -0.7071 0.7071 -3.142
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
geometry DEF basedel-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF basedel-COORD Coordinate { point [...]
}
texCoord DEF basedel-TEXCOORD TextureCoordinate { point [...]
}
coordIndex [...]
texCoordIndex [...]
}}},
DEF Baseder Transform {
translation 0.6497 -0.01354 -0.0647
rotation -0.5774 0.5774 -0.5774 -2.094
children [
Shape {
appearance Appearance {
material Material {

```

```

diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
geometry DEF Baseder-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Baseder-COORD Coordinate { point [...]
}
texCoord DEF Baseder-TEXCOORD TextureCoordinate { point [...]
}
coordIndex [...]
texCoordIndex [...]
}}},
DEF BaseIzq Transform {
translation -0.6307 -0.01354 -0.0647
rotation -0.5774 0.5774 -0.5774 -2.094
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {

```

```

url "../img/Madera.jpg"
}
}
geometry USE Baseder-FACES
}},
DEF Box02 Transform {
translation 0.4172 -0.05566 -0.07416
children [
Transform {
translation 0 0.2 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.4392 0.6627 0.6627
}
}
geometry Box { size 0.4415 0.4 0.3673 }
}} ]}],
DEF CARA_CAJ_N Transform {
translation -0.007912 -0.08905 0.008392
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
}
}

```

```

}
geometry DEF CARA_CAJ_N-FACES IndexedFaceSet {
  ccw TRUE
  solid TRUE
  coord DEF CARA_CAJ_N-COORD Coordinate { point [...]
}
  coordIndex [... ]
}}},
DEF CARA_CAJ_N01 Transform {
  translation -0.8331 -0.08951 0.008392
  children [
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0.6745 0.6745 0.6745
        ambientIntensity 0
        specularColor 0.91 0.91 0.91
        shininess 0.753
        transparency 0
      }
      texture ImageTexture {
        url "../img/Madera.jpg"
      }
    }
  }
  geometry USE CARA_CAJ_N-FACES
}}},
DEF gavetas Transform {
  translation 0.4172 -0.05566 -0.07416
  children [
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0.6745 0.6745 0.6745
        ambientIntensity 0

```

```

specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
geometry DEF gavetas-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF gavetas-COORD Coordinate { point [...]
}
texCoord DEF gavetas-TEXCOORD TextureCoordinate { point [...]
}
coordIndex [...]
texCoordIndex [...]
}}},
DEF gavetas01 Transform {
translation -0.4135 -0.05566 -0.07416
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
}
}

```

```

geometry USE gavetas-FACES
}},
DEF tabla_arriba Transform {
translation 0 0.3484 0.001753
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6745 0.6745 0.6745
ambientIntensity 0
specularColor 0.91 0.91 0.91
shininess 0.753
transparency 0
}
texture ImageTexture {
url "../img/Madera.jpg"
}
}
geometry DEF tabla_arriba-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF tabla_arriba-COORD Coordinate { point [..]
}
texCoord DEF tabla_arriba-TEXCOORD TextureCoordinate { point [...]
}
coordIndex [...]
texCoordIndex [...]
}}}]
DEF monitor Transform {
translation -51.55 1.616 -50.44
rotation 0 1 0 -0.7069
scale 0.024 0.024 0.024
scaleOrientation 0 1 0 -0.05525
children [

```

```

DEF MonitorB523 Transform {
translation -4.891 -10.73 -14.1
rotation 0 -0.7402 0.6724 -3.142
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.8941 0.7216 0.6
}
}
geometry DEF MonitorB523-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF MonitorB523-COORD Coordinate { point [...]
}
coordIndex [...]
}}],
DEF MonitorB524 Transform {
translation -2.575 -10.73 -14.1
rotation 0 -0.7402 0.6724 -3.142
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.8941 0.7216 0.6
}
}
geometry USE MonitorB523-FACES
}],
DEF MonitorE131 Transform {
translation -0.7874 -10.85 -14.11
rotation 0 -0.7402 0.6724 -3.142
children [
Shape {

```

```

appearance Appearance {
material Material {
diffuseColor 0.8941 0.7216 0.6
}
}
geometry DEF MonitorE131-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF MonitorE131-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF MonitorB525 Transform {
translation -3.719 -10.73 -14.1
rotation 0 -0.7402 0.6724 -3.142
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.8941 0.7216 0.6
}}
geometry USE MonitorB523-FACES
}},
DEF MonitorB527 Transform {
},
DEF MonitorB528 Transform {
},
DEF MonitorB529 Transform {
},
DEF MonitorB530 Transform {
},
DEF MonitoL69 Transform {
},
DEF MonitorE134 Transform {

```

```
},  
DEF Monitor953 Transform {  
},  
DEF Monitor977 Transform {  
},  
DEF Monitor981 Transform {  
},  
DEF Monitor982 Transform {  
},  
DEF Monitor983 Transform {  
},  
DEF MonitorPS70 Transform {  
},  
DEF Text71 Transform {  
},  
DEF Monitor991 Transform {  
},  
DEF MonitorB566 Transform {  
},  
DEF monitorbase Transform {  
},  
DEF monitorcaja Transform {  
},  
DEF monitoresferabase Transform {  
},  
DEF raton Transform {  
translation -51.74 1.285 -51.04  
children [  
DEF Alfombrill Transform {  
translation -8.392e-005 -0.02175 9.537e-005  
rotation 0 -1 0 -0.8465  
scale 0.15 0.15 0.15  
children [  
Shape {
```

```

appearance Appearance {
material Material {
diffuseColor 0.08627 0.08627 0.08627
ambientIntensity 0
specularColor 0 0 0
shininess 0.5535
transparency 0
}
}
geometry DEF Alfombrill-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Alfombrill-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF Alfombrill Transform {
translation -8.392e-005 -0.02175 9.537e-005
rotation 0 -1 0 -0.8465
scale 0.15 0.15 0.15
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.09412 0.5412 0.7804
ambientIntensity 0
specularColor 0.19 0.19 0.19
shininess 0.6105
transparency 0
}
}
geometry DEF Alfombrill-FACES IndexedFaceSet {
ccw TRUE
solid TRUE

```

```

coord DEF Alfombrill-COORD Coordinate { point [...]
}
coordIndex [...]
}}}],
DEF MOUSE Transform {
translation -8.392e-005 -0.02175 9.537e-005
rotation 0 -1 0 -0.8465
scale 0.15 0.15 0.15
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.6902 0.6902 0.6549
ambientIntensity 0
specularColor 0.69 0.69 0.69
shininess 0.753
transparency 0.07
}
}
geometry DEF MOUSE-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF MOUSE-COORD Coordinate { point [...]
}
coordIndex [...]
}}}}],
DEF silla Transform {
translation -50.5 1.092 -51.85
scale 1.344 1.344 1.344
children [
DEF Chair12_C2 Transform {
translation 0 -0.82 0
children [
Shape {

```

```

appearance Appearance {
material Material {
diffuseColor 0.2039 0.2039 0.2039
ambientIntensity 0
specularColor 0.69 0.69 0.69
shininess 0.753
transparency 0
}
}
geometry DEF Chair12_C2-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Chair12_C2-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF Chair12_C1 Transform {
translation 0 -0.9282 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.4 0 0.1451
ambientIntensity 0
specularColor 0.14 0.14 0.14
shininess 0.4585
transparency 0
}
}
geometry DEF Chair12_C1-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Chair12_C1-COORD Coordinate { point [...]
}

```

```

coordIndex [...]
}}}],
DEF Chair12_CH Transform {
translation 0 -0.8048 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0 0 0
ambientIntensity 0
specularColor 0.92 0.92 0.92
shininess 0.753
transparency 0
}
}
geometry DEF Chair12_CH-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Chair12_CH-COORD Coordinate { point[...]
}
coordIndex [...]
}}}],
DEF portamanos Transform {
translation -0.4749 0.01507 0.02837
scale 0.744 0.744 0.744
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.02745 0.02353 0.003922
}
}
geometry DEF portamanos-FACES IndexedFaceSet {
ccw TRUE

```

```

solid TRUE
coord DEF portamos-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF portamosizq Transform {
},
DEF TECLADO Transform {
translation -51.16 1.286 -50.86
rotation 0 -1 0 -2.426
scale 0.4662 0.4662 0.4662
children [
DEF keybd_spac Transform {
translation 0.0001164 -0.07779 -0.0002707
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.5922 0.5843 0.5608
ambientIntensity 0
specularColor 0.19 0.19 0.19
shininess 0.753
transparency 0
}
}
geometry DEF keybd_spac-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF keybd_spac-COORD Coordinate { point [...]
}
coordIndex [...]
}}},
DEF keybd_pad Transform {
translation 0.0001164 -0.07779 -0.0002707

```

```

children [
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0.8078 0.8 0.7843
        ambientIntensity 0
        specularColor 0.19 0.19 0.19
        shininess 0.753
        transparency 0
      }
    }
    geometry DEF keybd_pad-FACES IndexedFaceSet {
      ccw TRUE
      solid TRUE
      coord DEF keybd_pad-COORD Coordinate { point [...]
    }
    coordIndex [...]
  }
}],
DEF keybd_main Transform {
  translation 0.0001164 -0.07779 -0.0002707
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.8078 0.8 0.7843
          ambientIntensity 0
          specularColor 0.19 0.19 0.19
          shininess 0.753
          transparency 0
        }
      }
      geometry DEF keybd_main-FACES IndexedFaceSet {
        ccw TRUE
        solid TRUE

```

```

coord DEF keybd_main-COORD Coordinate { point [...]
}
coordIndex [...]
}],
DEF keybd_home Transform {
translation 0.0001164 -0.07779 -0.0002707
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.5922 0.5843 0.5608
ambientIntensity 0
specularColor 0.19 0.19 0.19
shininess 0.753
transparency 0
}
}
geometry DEF keybd_home-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF keybd_home-COORD Coordinate { point [...]
}
coordIndex [...]
}],
DEF keybd_fnct Transform {
translation 0.0001164 -0.07779 -0.0002707
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.5922 0.5843 0.5608
ambientIntensity 0
specularColor 0.19 0.19 0.19
shininess 0.753

```

```

transparency 0
}
}
geometry DEF keybd_fnct-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF keybd_fnct-COORD Coordinate { point [...]
}
coordIndex [...]
}}}],
DEF keybd_botm Transform {
},
DEF BASE Transform {
translation -0.007264 -0.04552 0.002252
rotation 1 0 0 0
scale 2.145 2.145 2.145
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.9176 0.9176 0.9176
}
}
geometry DEF BASE-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF BASE-COORD Coordinate { point [...]
}
coordIndex [...]
}}]]}
DEF PISO Transform {
translation -49.77 0.002459 -49.68
children [
DEF PISO-TIMER TimeSensor { loop TRUE cycleInterval 3.333 },

```

```

Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0.5 0.5 0.5
      ambientIntensity 0.1
      specularColor 0.81 0.81 0.81
      shininess 0.2875
      transparency 0
    }
    texture ImageTexture {
      url "../img/Con5_b.JPG"
    }
    textureTransform TextureTransform {
      center 0.5 0.5
      scale 1.5 1.5
    }
  }
  geometry DEF PISO-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF PISO-COORD Coordinate { point [...]
  }
    texCoord DEF PISO-TEXCOORD TextureCoordinate { point [...]
  }
    coordIndex [
    2, 3, 0, -1, 1, 2, 0, -1]
    texCoordIndex [
    2, 3, 0, -1, 1, 2, 0, -1]
  ]}]
  DEF archivador Transform {
    translation -51.32 1.386 -53.1
    scale 1.7 1.548 1.548
    children [
    DEF LOCK_MEDIU Transform {

```

```

translation 0.001927 -0.89 -0.007194
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.4706 0.4706 0.5882
ambientIntensity 0.5098
specularColor 0.4329 0.4329 0.5412
shininess 0.7625
transparency 0
}
}
geometry DEF LOCK_MEDIU-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF LOCK_MEDIU-COORD Coordinate { point [...]
}
coordIndex [...]
}}],
DEF gavetas_0 Transform {
translation 0.001148 0.002447 0.183
children [
DEF gaveta0 Transform {
},

```

Enrutamiento de Eventos.

Permite el enlace entre los objeto para realizar un evento en al ambiente virtual, estos son mensaje que ligan a los eventos. A continuación algunos errutadores utilizado en el Ambiente Virtual.

```
#####Ruta de Animacion de Jefatura de CYS#####
```

```

ROUTE Volver_principal_O.touchTime TO Ir_Principal_O.Volver
ROUTE Ir_Principal_O.ApagarCamaraVolverV TO ControlApagarCamara.ApagarCamara
ROUTE Ir_Principal_O.AuxiliarFraccionV TO Ir_oficina.Entrar
ROUTE Ir_Principal_O.ApagarCamaraSalaV TO camOficina.fieldOfView
ROUTE Ir_Principal_O.ApagarCamaraSalaV TO vision_jefe.fieldOfView
ROUTE Ir_Principal_O.output TO SWITCH.whichChoice
ROUTE Ir_Principal_O.output TO SWITCH1.whichChoice
ROUTE Ir_Principal_O.camara TO Oficina.set_bind
ROUTE Ir_Principal_O.camara TO NavegacionPrincipal.set_bind

ROUTE Entrar_Oficina-SENSOR.touchTime TO PuertaJefatura-TIMER.startTime
ROUTE PuertaJefatura-TIMER.fraction_changed TO Ir_oficina.Entrar
ROUTE Ir_oficina.ApagarCamaraEntrarO TO ControlApagarCamara.ApagarCamara
ROUTE Ir_oficina.ApagarCamaraOE TO camOficina.fieldOfView
ROUTE Ir_oficina.ApagarCamaraOE TO vision_jefe.fieldOfView
ROUTE Ir_oficina.output TO SWITCH.whichChoice
ROUTE Ir_oficina.output1 TO SWITCH1.whichChoice
ROUTE Ir_oficina.camara TO camOficina.set_bind
ROUTE Ir_oficina.camara TO NavegacionOficina.set_bind

```

Nodo Script

Esto nodo permite el control de los eventos.

```

DEF Ir_Principal_O Script {
  eventIn SFTIME Volver
  eventOut SFInt32 output
  eventOut SFBool camara
  eventOut SFFloat ApagarCamaraSalaV
  eventOut SFFloat ApagarCamaraVolverV
  eventOut SFFloat AuxiliarFraccionV
  url "javascript:
  function initialize() {
    output = 0;
  }
  function Volver(value, time) {
    ApagarCamaraVolverV = 0.7363;

```

```

        AuxiliarFraccionV = 0;
        output = 0;
        camara = TRUE;
        ApagarCamaraSalaV = 0.0;
    }
    "
}

DEF Ir_oficina Script {
    eventIn SFFloat Entrar
    eventOut SFInt32 output
    eventOut SFInt32 output1
    eventOut SFBool camara
    eventOut SFFloat AuxiliarFraccionO
    eventOut SFFloat ApagarCamaraOE
    eventOut SFFloat ApagarCamaraEntrarO
    url "javascript:
        function Entrar(FraccionO) {
            AuxiliarFraccionO = FraccionO;
            if (FraccionO == 1.0)
            {
                ApagarCamaraOE = 0.8985;
                ApagarCamaraEntrarO = 0;
                output = 1;
                output1 = 2;
                camara = TRUE;
            }
        }"
}

```

A continuacion solo se listara los encabeza de los script importantes que controlan los eventos.

Script PANORAMA

Script ControlApagarCamara

Script Principal_S

Script Ir_Principal_Cubiculo3

Script Entrar_Cubiculo3

Script MensajeControl

<i>Script Ir_Sala</i>	<i>Script GuionCamara</i>
<i>Script Ir_Zona6_SalaN</i>	<i>Script CambiarEstado</i>
<i>Script Ir_SalaNave</i>	<i>Script Mensaje</i>
<i>Script Ir_Principal_O</i>	<i>Script Mensaje2</i>
<i>Script Ir_Oficina</i>	<i>Script Mensaje3</i>
<i>Script Entrar_Cubiculo1</i>	<i>Script Mensaje4</i>
<i>Script Ir_Principal_Cubiculo2</i>	<i>Script Mensaje5</i>
<i>Script Entrar_Cubiculo2</i>	<i>Script InformacionDeMundo</i>

Sistema de almacenamiento de AVUDONA_2 (BASA DE DAOS).

EL sistema utilizado para el almacenamiento del archivo que describe el Ambiente Virtual, es similar a unas de las técnicas utilizadas para almacenar imágenes en una base de datos en interfaces WEB, el cual, solo guardan la ruta de acceso de la imagen y a través del documento HTML se muestra la imagen. El sistema de Base de Datos utilizado SQL aplicando el modelo relación, en la figura 2 a continuación nos muestra la elación grafica de la base de dato y sus campos.

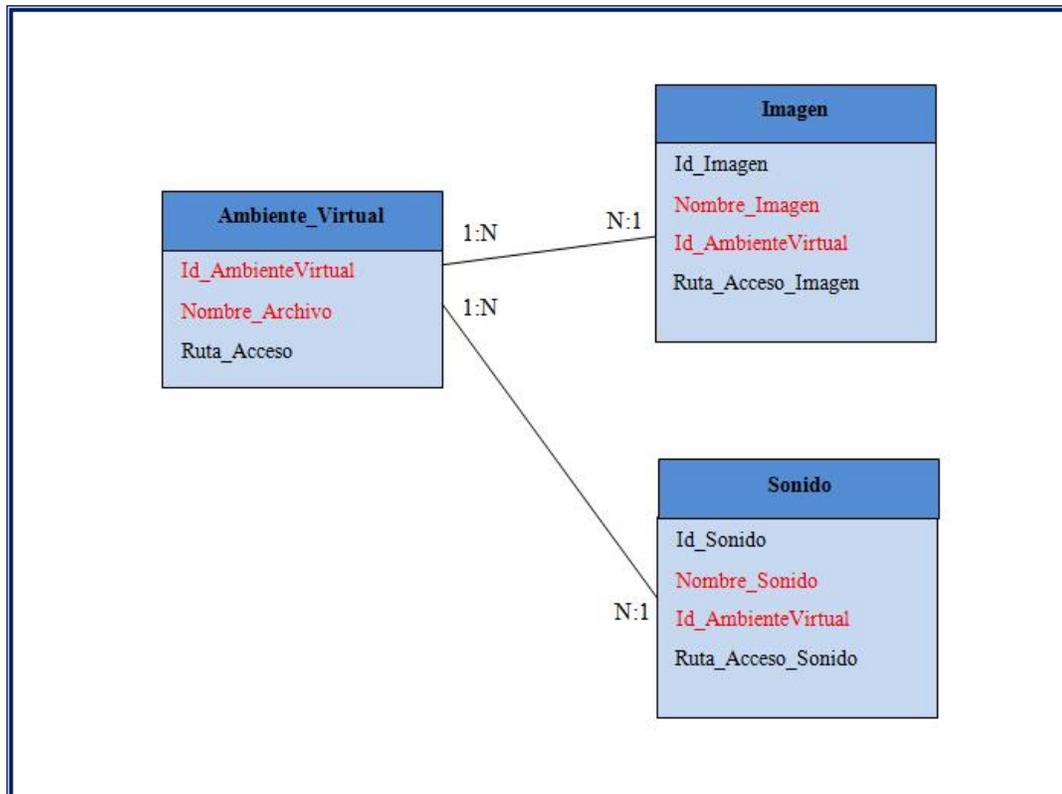


Figura 2. Modelo Conceptual de la Base de Datos del AVUDONA_2

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

TÍTULO	"ACTUALIZACION Y EXPACION DEL AMBIENTE VIRTUAL PROTOTIPO DE DEPARTAMENTO DE COMPUTACION Y SISTEMAS DE LA UNIVERSIDAD DE ORIENTE NUCLEO DE ANZOATEGUI."
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
José Luis Millán Chirinos	CVLAC: 11.656.968 E MAIL: josemillanchinet@gmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

_____ AVUDONA: AMBIENTE VIRTUAL DE LA UNIVERSIDAD DE
ORIENTE NUCLEO DE ANZOATEGUI _____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
Ingeniería y Ciencias Aplicadas	Ingeniero en Computación

RESUMEN (ABSTRACT):

El desarrollo de estés proyecto permitió extender y actualizar el Ambiente Virtual de la Universidad de Núcleo de Anzoátegui, agregando nuevas escenas y realizando cambios tanto en las tecnologías aplicadas así como en las escenas que sufrieron cambios homónimo real. En este Ambiente Virtual AVUDONA 2, el Visitante se limitara a moverse libremente por las escenas del mundo virtual, desde la Entrada de la Universidad, Edificio Decanato, Biblioteca "Luis García Pelissier", Escuela de Ingeniería hasta el Auditorio. Permittiendo conocerlas a distancias utilizando no solo medios multimedia como CD-ROM/DVD-RW, sino también a través de la tecnología WWW o internet.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**CONTRIBUIDORES:**

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Pedro Dorta	ROL	CA	<input checked="" type="checkbox"/> AS	TU	JU
	CVLAC:	12.914.617			
	E_MAIL	dortap@hotmail.com			
	E_MAIL				
Víctor Mújica	ROL	CA	AS	TU	<input checked="" type="checkbox"/> JU
	CVLAC:	14.054.907			
	E_MAIL	vmujicaudo@yahoo.com			
	E_MAIL				
Claudio Cortinez	ROL	CA	AS	TU	<input checked="" type="checkbox"/> JU
	CVLAC:	12.553.334			
	E_MAIL	Cl_cortinez@cantv.net			
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	05	07
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**ARCHIVO (S):**

NOMBRE DE ARCHIVO	TIPO MIME
TESIS: InformeAVUDONA2	Aplication/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H
 I J K L M N O P Q R S T U V W X Y Z . a b c d e f g h i j k l m n o p q r s t u
 v w x y z . 0 1 2 3 4 5 6 7 8 9 .

ALCANCE

ESPACIAL: Dpto. Computación y Sistemas UDO (OPCIONAL)

TEMPORAL: 12 mese y medios (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pre – Grado

ÁREA DE ESTUDIO:

Dpto. Computación y Sistemas

INSTITUCIÓN:

UDO – Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**DERECHOS**

Articulo N° 44

Del Reglamento de Trabajo de Grado

"Los trabajos de grado son exclusiva propiedad de la Universidad y solo podrán ser usados a otros fines con el consentimiento del consejo de Núcleo respectivo, quien lo participara al Consejo Universitario."

JOSE LUIS MILLAN CHIRONOS

AUTOR

AUTOR

AUTOR

PEDRO DORTA

TUTOR

VICTOR MUJICA

JURADO

CLAUDIO CORTINEZ

JURADO

JOSE BASTARDO

POR LA SUBCOMISION DE TESIS