



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS
PARA EL DESARROLLO DE *SOFTWARE* CON INTERFAZ 3D**

Fausto Américo Durini Castillo

Asesorado por el Ing. Bayron Wosvely López López

Guatemala, abril de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS
PARA EL DESARROLLO DE SOFTWARE CON INTERFACES EN 3D**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

FAUSTO AMÉRICO DURINI CASTILLO

ASESORADO POR EL ING. BAYRON WOSVELY LÓPEZ LÓPEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, ABRIL DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	P.A. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Freiry Javier Gramajo López
EXAMINADOR	Ing. Edgar René Ornelyz Hoil
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS PARA EL DESARROLLO DE *SOFTWARE* CON INTERFAZ EN 3D

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2010.

Fausto Américo Durini Castillo

Guatemala, 14 de febrero de 2011

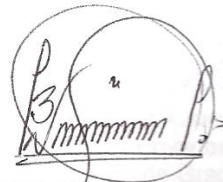
Ing. Carlos Azurdia
Revisor de Tesis
Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero:

Por este medio hago de su conocimiento que he revisado el trabajo de investigación titulado "EVALUCACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS PARA EL DESARROLLO DE SOFTWARE CON INTERFAZ 3D" desarrollado por el estudiante FAUSTO AMÉRICO DURINI CASTILLO, quien se identifica con número de carné 200010606, el cual ha estado desarrollando y que a la fecha ha completado, cumpliendo con los objetivos planteados para ésta investigación. Por lo anterior manifiesto mi aprobación por el esfuerzo, dedicación y resultados obtenidos por el estudiante para este trabajo de investigación.

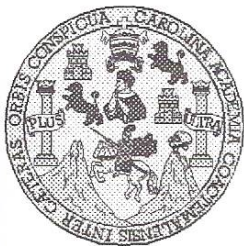
Sin otro particular y agradeciendo la oportunidad de colaboración a la educación e investigación universitaria,

Atentamente:



Bayron Wosvely López López
Ingeniero en Ciencias y Sistemas
Colegiado Número 4403

Bayron Wosvely López López
INGENIERO EN CIENCIAS Y SISTEMAS
Colegiado 4403



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 25 de Febrero de 2011

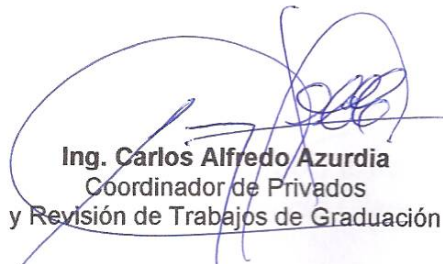
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **FAUSTO AMÉRICO DURINI CASTILLO**, carné **2000-10606**, titulado: **“EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS PARA EL DESARROLLO DE SOFTWARE CON INTERFAZ 3D”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS PARA EL DESARROLLO DE SOFTWARE CON INTERFAZ 3D”**, presentado por el estudiante **FAUSTO AMÉRICO DURINI CASTILLO**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería Ciencias y Sistemas



Guatemala, 07 de abril 2011



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **EVALUACIÓN DE TECNOLOGÍAS Y REQUERIMIENTOS PARA EL DESARROLLO DE SOFTWARE CON INTERFACES EN 3D**, presentado por el estudiante universitario **Fausto Américo Castillo**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos
DECANO



Guatemala, abril de 2011

/cc

ACTO QUE DEDICO A:

Mis papás	Por todo su amor y cariño que siempre me han dado.
Mis hermanas	Por todo lo que me han enseñado, querido y apoyado.
Mi familia	Por su apoyo incondicional en todas mis metas.
Mis amigos y compañeros de vida	Por haber completado juntos las etapas de esta gran carrera.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	III
GLOSARIO	V
RESUMEN	IX
OBJETIVOS	XI
INTRODUCCIÓN	XIII
1 TENDENCIAS TECNOLÓGICAS DEL SOFTWARE Y SUS INTERFACES	1
1.1 Conceptos	1
1.2 Evolución del <i>Hardware</i> vs. <i>Software</i>	2
1.3 Interfaces de una dimensión (1D)	4
1.4 Interfaces GUI o de dos dimensiones (2D)	6
1.5 Diferencia entre interfaces de una y dos dimensiones	10
1.6 Clasificación de las interfaces de usuario	11
1.7 Interfaces en 3D	12
2 REQUERIMIENTOS DE COMPORTAMIENTO Y MANEJO DE UNA INTERFAZ EN 3D APLICADOS A TRAVÉS DE METÁFORAS	15
2.1 Requerimientos fundamentales de comportamiento	16
2.2 Aplicación de una metáfora	28
2.3 Diagramas de metáforas	31
2.4 Evaluación de metáforas	33
2.5 Manejo de metáforas	36
3 FACTORES TECNOLÓGICOS Y REQUERIMIENTOS DE FUNCIONAMIENTO INTERNO DE UNA INTERFAZ EN 3D	39
3.1 Tecnologías de <i>hardware</i>	40
3.1.1 Dispositivos de entrada	40

3.1.2	Dispositivos de procesamiento	47
3.1.3	Dispositivos de almacenamiento	49
3.1.4	Dispositivos de salida	52
3.2	Tecnologías de <i>software</i>	52
3.3	Requerimientos de funcionamiento interno	57
4	IMPACTO TECNOLÓGICO DEL DESARROLLO DE <i>SOFTWARE</i> CON INTERFAZ EN 3D	65
4.1	¿Qué hace posible el desarrollo de interfaces en 3D?	66
4.2	¿Qué tecnologías quedarán obsoletas?	67
4.3	¿Qué problemas se minimizan o resuelven?	69
4.4	¿Qué ventajas posee el <i>software</i> con interfaz en 3D?	72
4.5	¿Qué obstáculos posee frente a otras tecnologías?	75
4.6	¿Cuál es el futuro de las interfaces en 3D?	78
5	LINEAMIENTOS DE EVALUACIÓN PARA APLICACIÓN DE MÉTRICAS PARA EL ANÁLISIS Y DESARROLLO DE APLICACIONES CON TERCERA DIMENSIÓN	85
5.1	Lineamientos para requerimientos de interfaces en 3D	86
5.2	Lineamientos para evaluación de prototipos de interfaces en 3D	92
5.2.1	Evaluación de interacción	93
5.2.2	Evaluación de usuario	102
	CONCLUSIONES	105
	RECOMENDACIONES	107
	REFERENCIAS	109
	BIBLIOGRAFÍA	111
	ANEXOS	113

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Ejemplo línea de comandos MS-DOS	5
2.	Ejemplo aplicación con una CUI	6
3.	Icono de impresora	17
4.	Icono de la papelera de reciclaje	18
5.	Ejemplo del proyecto <i>Looking Glass</i>	25
6.	Ejemplo de la metáfora de galería de tareas	30
7.	Ejemplo de plantilla de ficha descriptiva	31
8.	Ejemplo de diagrama de relación de metáforas	32
9.	Ejemplo de árbol de metáforas según clasificación	32
10.	Ejemplo de evaluación de requerimientos fundamentales	35
11.	Ejemplo de diagrama de evaluación de metáforas	36
12.	Diagrama de interacción de usuario y dispositivos de <i>hardware</i>	40
13.	Ejemplo de <i>mouse</i> 3D	44
14.	Ejemplo de guante de movimientos	45
15.	Ejemplo de reconocimiento gráfico	46
16.	Relación de lenguajes de programación para 3DUI	55
17.	Ejemplo de funcionamiento de <i>Tactile 3D</i>	56
18.	Ejemplo de aplicación científica	80
19.	Ejemplo simple del efecto libro	82
20.	Aplicación de lineamientos de requerimientos de software con interfaz en 3D	87
21.	Estructura de la descripción general del <i>software</i> con interfaz en 3D	88
22.	Ejemplo de descripción general del <i>software</i> con interfaz en 3D	89
23.	Estructura de definición de metáforas y objetos	90
24.	Ejemplo de definición de metáforas y objetos	90

25. Ejemplo de mediciones de evaluación de interacción	94
26. Ejemplo de gráfica de promedio de usuarios que completaron una tarea	100
27. Ejemplo de gráfica de promedio de usuarios que completaron una tarea para la lectura del libro	101

TABLAS

I. Cronología de <i>hardware</i> vs. sistema operativo	3
II. Cronología de las GUI	7
III. Interfaces independientes para Unix o Linux	10
IV. Tipos de navegación	19
V. Cronología de <i>hardware</i> Vs. sistema operativo	20
VI. Tareas básicas de selección y manipulación	20
VII. Clasificación de dispositivos de entrada según el envío de datos	41
VIII. Implementaciones tecnológicas de sensores de tacto	42
IX. Clases de dispositivos de salida	52
X. Métricas de medición de efectividad	95
XI. Métricas de medición de eficiencia	96
XII. Métricas de medición de satisfacción	97
XIII. Usuarios que completaron una tarea (0% ó 100%)	98
XIV. Ejemplo usuarios que completaron una tarea (0% ó 100%) para la lectura del libro	99
XV. Promedio de usuarios que completaron una tarea (0% ó 100%)	99
XVI. Ejemplo del promedio de usuarios que completaron una tarea (0% ó 100%), para la lectura del libro	99
XVII. Tiempo transcurrido para completar cada tarea (en minutos)	102

GLOSARIO

API	Interfaz de programación de aplicaciones.
GDI	<i>Graphics Device Interface</i> , interfaz de dispositivos para gráficos. Lenguaje de gráficos Windows utilizado para proveer salida a la pantalla e impresora. Las aplicaciones llaman a las funciones GDI en Windows para presentación e impresión.
GUI	<i>Graphic User Interface</i> , interfaz gráfica de usuario es una interfaz de usuario que utiliza ventanas y gráficas como parte básica de su interacción.
Java	Plataforma de <i>software</i> desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.
Java3D	Extensión de java orientada a la creación de elementos en tercera dimensión.

JDIC	<i>JDesktop Integration Components</i> , herramienta de Sun que integra completamente las aplicaciones multiplataforma basadas en Java con los entornos de escritorio nativos.
JDNC	<i>JDesktop Network Components</i> , herramienta de Sun para el desarrollo de aplicaciones en red para entornos de escritorio.
MPEG	<i>Moving Picture Experts Group</i> , estándar de compresión para ficheros de audio y vídeo que facilita la transmisión en la red de archivos con este tipo de contenidos y su almacenamiento digital.
OPENGL	<i>Open Graphics Library</i> , biblioteca de gráficos abierta; biblioteca gráfica desarrollada originalmente por <i>Silicon Graphics Incorporated</i> (SGI).
RGL	Librería basada en paquetes estadísticos para la creación de métodos sofisticados de visualización en tercera dimensión.
Sistema operativo	Conjunto de programas que administran los recursos de la computadora, y permite crea el medio de comunicación entre el usuario y el <i>hardware</i> .

UI	<i>User Interface</i> , interfaz de usuario
VE	<i>Virtual Environment</i> , es una representación del mundo real representado gráficamente.
VRML	<i>Virtual Reality Modeling Language</i> , formato de archivo normalizado que tiene como objetivo la representación de gráficos interactivos tridimensionales; diseñado particularmente para su empleo en la web.
WIMP	<i>Windows – Icons – Menus – Pointers</i> , ventanas – iconos –menús – punteros, método de interacción que el usuario utiliza para interactuar con algunas GUI actuales.
X3D	Lenguaje XML 3D basado en VRML.
XML	<i>Extended Markup Language</i> .

RESUMEN

Dentro de la era tecnológica actual, el nuevo *hardware* sigue aumentando y mejorando sus capacidades, con el objetivo de obtener nuevos elementos que faciliten la satisfacción de los usuarios. Como todo cambio constante, posee varios efectos, uno de ellos es la innovación para crear nuevas generaciones de *software* que tratan de explotar al máximo todos los recursos y satisfacer los requerimientos de los usuarios.

Durante esta investigación se enmarca el cambio de las interfaces de *software* considerando su tendencia histórica, la capacidad de convertirse en un *software* funcional y bien diseñado, a través de los requerimientos de comportamiento, y la aplicación de metáforas que permiten capturar la realidad para convertirse en el diseño del *software*.

Describe las tecnologías de *hardware* y *software* que poseen la capacidad para ejecutar y/o desarrollar el *software* con interfaz en tercera dimensión, los requerimientos de funcionamiento interno que deben seguirse durante la construcción del *software* para que optimicen el uso de los recursos y permita al usuario experimentar la tercera dimensión, con el mayor realismo posible sin que el desempeño se vea afectado.

Toda nueva tecnología de *software* implica nuevas tendencias de desarrollo, las cuales generan un impacto tecnológico, el cual es descrito a través de varios puntos de vista como: los adelantos que la hacen posible, las tecnologías que dejan obsoletas, las ventajas y obstáculos que presenta, así como una reseña de lo que logrará una vez que sea una tecnología madura.

No se pretende crear ningún tipo de metodología de desarrollo o de implementación, pero sí generar instrumentos de análisis, diseño, desarrollo y evaluación de *software*, que permitan adherirse a metodologías actuales de desarrollo de *software* para la implementación de esta tecnología.

OBJETIVOS

General

Evaluar tecnologías, requerimientos de comportamiento, funcionamiento interno y manejo, para el desarrollo de *software* con interfaz en 3D.

Específicos

1. Establecer la tendencia histórica de las interfaces de una y dos dimensiones, para comprender el cambio a una interfaz en 3D.
2. Identificar los requerimientos de comportamiento y manejo de una interfaz en 3D, que permitan al usuario utilizar la percepción natural de las cosas.
3. Identificar los factores tecnológicos y requerimientos de funcionamiento interno de una interfaz en 3D, que sirvan para su ejecución y optimización.
4. Determinar el impacto tecnológico de los factores y requerimientos, que demuestren las ventajas y desventajas del desarrollo de *software* con interfaz en 3D.
5. Identificar lineamientos de evaluación para la aplicación de métricas para el análisis y desarrollo de aplicaciones con tercera dimensión.

INTRODUCCIÓN

Con la evolución del *hardware* y *software*, se ha mejorado la interacción con la computadora, creando el concepto de interfaz, el cual con el pasar de los años ha utilizado para su visualización o despliegue, sistemas gráficos de una y dos dimensiones. En esta investigación se realiza una evaluación de tecnologías para la aplicación del cambio a interfaces en tercera dimensión.

Considerando que el *software*, además del cumplimiento de sus objetivos, aplique lineamientos para esta nueva tecnología, conocidos como metáforas, requerimientos de comportamiento y funcionamiento interno durante su diseño y desarrollo, brindan el máximo realismo posible en los entornos que crean y permiten a la personas, simular el sentimiento del realismo con un buen desempeño.

1 TENDENCIAS TECNOLÓGICAS DEL SOFTWARE Y SUS INTERFACES

1.1 Conceptos

El *software*, es el componente intangible en una computadora que actúa como medio de comunicación para realizar distintas tareas en ella. El *software* al igual que cualquier otro elemento que se relacione con la tecnología computacional, ha tendido a seguir una trayectoria de evolución constante, con el único y claro objetivo de crear nuevas tecnologías que mejoren y faciliten la vida del hombre en todo sentido. El término de *software* incluye, dentro de su definición, a los programas o aplicaciones que se ejecutan sobre los sistemas operativos y a los sistemas operativos mismos.

Durante esta evolución los programas de *software* han mejorado la forma de interacción con la computadora, creando el concepto de interfaz, el cual se definirá para mostrar el enfoque principal de este trabajo de investigación, el cual tratará de explicar el impacto del desarrollo de interfaces en tercera dimensión, con o sin semejanza con la realidad.

- Interfaz

“Conjunto de comandos y métodos que permiten comunicarse a cualquier programa con un elemento interno o externo a él”¹.

¹ 3DUI (3D User Interface). Organización dedicada al diseño y evaluación de interfaces en tercera dimensión. [ref. 18 de agosto de 2010]. Disponible en Web: <<http://www.3dui.org>>

En general, la interfaz es el medio interno de ejecución, el cual puede ser utilizado a través de una interfaz de usuario para interactuar directamente con el *hardware*.

- Interfaz de Usuario

“Desde el punto de vista espacial, la interfaz se define como el lugar de la interacción, es decir, el espacio donde se desarrolla el intercambio de datos o información entre el usuario y el sistema, creado por el diseñador”².

El objetivo técnico de las interfaces de usuario es el adaptar la complejidad de los sistemas y las capacidades del ser humano. Éste es el componente esencial para que se puedan realizar las tareas en los sistemas operativos y aplicaciones que se ejecutan en ellos.

1.2 Evolución del *Hardware* vs. *Software*

El *hardware* es el elemento que marca las condiciones para desarrollar el *software* que se va a ejecutar en él, y es la razón principal para establecer el comportamiento y funcionamiento de los programas que existen actualmente.

Como punto principal de la comparación, se enmarcarán a los sistemas operativos y su desarrollo en los últimos años, tomando en cuenta el *hardware* para el cual fueron desarrollados.

² IEEE VGTC (Visualization & Graphics Technical Committee). [ref. 30 de septiembre de 2010]. Disponible en Web:<<http://www.cs.sfu.ca/~vis/Track/3DUI.html>>

Tabla I. **Cronología de *hardware* vs. sistema operativo**

Tiempo	Hardware	Sistema operativo
Primera generación 1945 – 1955	Tubos de vacío y páneles de control	No existía un sistema operativo.
Segunda generación 1955 -1965	Transistores (tarjetas perforadas)	De trabajo único: se dedica a preparar el sistema para la llegada de trabajos. Por lotes: automatiza una secuencia de operaciones que lee, interpreta y ejecuta.
Tercera generación 1965 -1980	Circuitos integrados	Multiprogramación: más de un programa residente en memoria principal al mismo tiempo. Multiprocesamiento: más de un procesador. Multiusuario: permite acceder a varios usuarios a un mismo computador mediante terminales interactivas.
Cuarta generación 1980 – Actual	Microprocesador	De red: el usuario es consciente de la existencia de computadoras interconectadas e indica con cuál va a trabajar. Distribuido: el usuario no es consciente de las computadoras conectadas, porque actúan como solo una. Multihilo: capacidad de ejecutar más de un hilo por núcleo. Multinúcleo: capacidad de albergar a núcleos en un procesador físico.
Quinta generación Actual - Futuro	Inteligencia artificial – No oficial –	En desarrollo, únicamente en diseño y modelos simples de implementación.

Fuente: María Encarnación, Rufino González, Sistemas operativos, Pág.3

Dentro de los dispositivos de *hardware*, el que ha presentado un crecimiento constante en sus capacidades de procesamiento es el microprocesador, tal es el caso que actualmente existen procesadores con más de un núcleo interno y éste, con capacidad de manejar varios hilos en cada uno.

Además, los microprocesadores ya no están restringidos solamente como parte de los CPU, ahora son también una parte esencial de las tarjetas de video y de otros componentes para procesamiento de datos.

Con la combinación tecnológica de la cuarta generación, se están desarrollando aplicaciones capaces de administrar, balancear y monitorear la carga de procesamiento de datos, basado en “M” computadoras con “N” microprocesadores trabajando en paralelo, generando una capacidad de procesamiento de datos calculable, solamente a través del poder adquisitivo para tener esta tecnología.

1.3 Interfaces de una dimensión (1D)

Todos los sistemas operativos de primera, segunda y parte de la tercera generación, incluían una interfaz de usuario que hoy en día se denomina CUI (*Command-line User Interface*), acrónimo de interfaz de usuario de línea de comando. Dentro de los ejemplos más famosos de este tipo de interfaces se pueden nombrar a: MS-DOS (Sistema operativo de lotes), consola de Linux.

Figura 1. Ejemplo línea de comandos MS-DOS

```
BL          <DIR>          26.06.01    21:07
PROGIS     <DIR>          28.06.01    18:11
NECMOUSE   <DIR>          05.07.01    21:38
TRANSIT    <DIR>          14.07.01    19:38
CORELDRAW  <DIR>          19.09.01    21:35
COMMAND    COM           57.377 31.05.94    6:22
WINA20     386           9.349 31.05.94    6:22
CONFIG     OLD           147 12.06.01    17:44
AUTOEXEC   OLD           230 22.06.01    21:05
NETWIN     BAT            42 01.07.01    15:45
TREEINFO   MCD           539 23.06.01    13:02
CONFIG     SYS           208 30.06.01    15:45
AUTOEXEC   BAT           292 05.08.01    22:41
DEUTSCH    PDF           861.570 10.10.00    14:52
CHKLIST    MS            135 27.11.01    22:00
          37 Datei(en)          929.889 Byte
          50.778.112 Byte frei

C:\>ver
MS-DOS Version 6.22

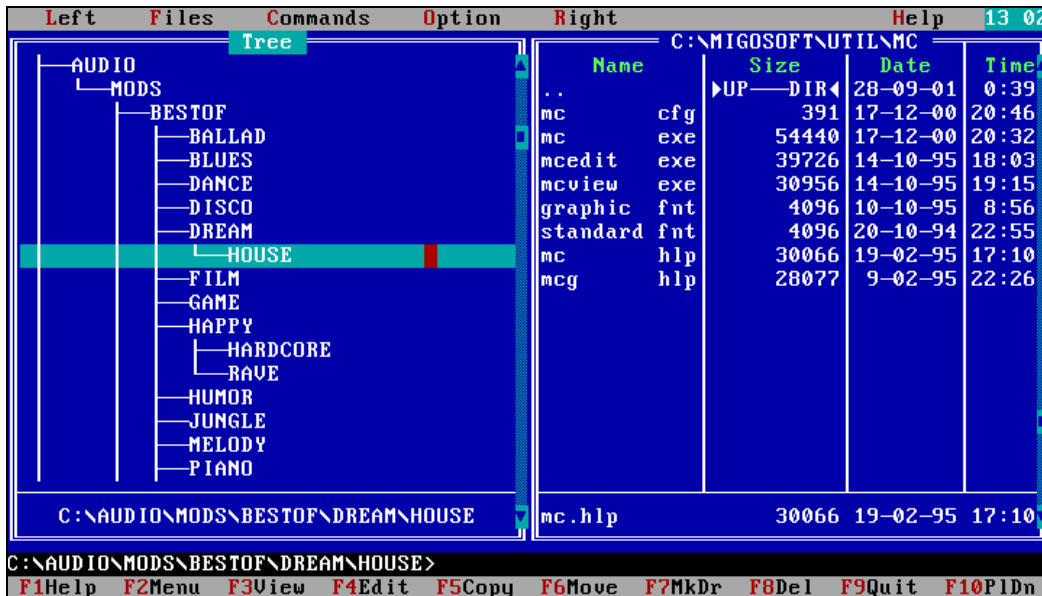
C:\>_
```

Fuente: www.winhistory.de/pics/big/dos622.gif

Todas las interfaces CUI son consideradas interfaces de una dimensión. En el paso de CUI a GUI se desarrollaron algunas interfaces que daban los primeros pasos en mejorar la organización del diseño, manteniendo un aspecto de línea de comando, pero implementando las siguientes características.

- Mejoras producidas por la introducción de menús simples: la máquina los muestra y el usuario introduce el código asociado.
- Primeras herramientas de carácter gráfico (mouse).
- Introducción de cadenas de acceso rápido, cuando hay muchos grupos de órdenes.
- Creación de las primeras formas para ingreso de datos.

Figura 2. Ejemplo aplicación con una CUI



Fuente: <http://migo.sixbit.org/software/dos/screenshots/mc.gif>

En la línea de desarrollo de los sistemas, se da paso a una nueva era de interfaces de usuario de mejor visualización, por medio de gráficos bidimensionales llamada GUI (*Graphic User Interface*).

1.4 Interfaces GUI o de dos dimensiones (2D)

Las interfaces en dos dimensiones o GUI, son las más utilizadas en el mercado del *software* actualmente, están constituidas por un pánel o formulario principal, el cual contiene objetos con formas que permiten interactuar utilizando los ejes “X” y “Y” para la organización de la información y objetos de control que permiten ejecutar tareas personalizadas según la programación que se realice para cada uno.

La mayoría de los lenguajes de programación permiten la creación de interfaces de dos dimensiones con objetos predefinidos, a través de sus entornos de programación IDE, lo cual ha permitido el desarrollo y estudio para el diseño de interfaces que permiten dar a los usuarios comodidad y ergonomía en el uso de las aplicaciones. Dentro de las interfaces de 2 dimensiones más famosas se puede mencionar la de *Microsoft Windows 3.1* y *Microsoft Windows 95*.

- Evolución cronológica de las GUI

La evolución de las interfaces de usuario corre en paralelo con la de los sistemas operativos, ya que de hecho la interfaz constituye actualmente uno de los principales elementos de un sistema operativo, por ser el encargado de soportar la ejecución de los programas. La siguiente tabla muestra el año en que la GUI fue puesta en funcionamiento.

Tabla II. **Cronología de las GUI**

Año	GUI
1973	<i>Xerox PARC Alto</i>
1980	<i>Three Rivers Computer Corporation Perq</i>
1981	<i>Xerox PARC Star</i>
1983	<i>Apple Lisa (Presenta gran influencia de los sistemas Xerox PARC)</i> <i>Visi Corp Visi On</i>
1984	<i>MIT Project X (Participantes Sun, IBM, HP, y DEC)</i> <i>Apple Macintosh</i> <i>Digital Research GEM (su uso conocido fue en computadoras Atari ST)</i>
1985	<i>Amiga Workbench 1.0</i> <i>Windows 1.0</i> <i>Geos (Para Commodore 64 y Apple II)</i>

Continúa Tabla II

1986	<i>Primera versión comercial de X para UNIX</i>
1987	<i>Color Macintosh Windows 2.03 Acorn Arthur</i>
1988	<i>NeXT Apple GS/OS IBM OS/2 Version 1.10 Presentation Manager</i>
1990	<i>Windows 3.0 (Primera GUI exitosa de Microsoft) Amiga Workbench 2 PC-GEOS</i>
1992	<i>Windows 3.1/3.11 IBM OS/2 Version 2.0 con Shell de Workplace Amiga Workbench 3</i>
1993	<i>Windows for Workgroups 3.11, NT 3.1/3.5.</i>
1994	<i>XFree86, en la implementación de código abierto de X11R6 (Linux adquiere su propia GUI) QNX Photon microGUI</i>
1995	<i>Windows 95 / Microsoft Bob BeOS</i>
1996	<i>Windows NT 4.0 OS/2 Warp 4 con Shell de Workplace</i>
1997	<i>Apple Mac OS 8</i>
1998	<i>Windows 98</i>
1999	<i>RISC OS 4</i>
2000	<i>Windows 2000</i>
2001	<i>Mac OS X released Windows XP</i>
2003	<i>Windows 2003</i>
2007	<i>Windows Vista Sun Glass iPhone OS 1.0</i>

Continúa Tabla II

2008	<i>Windows 2008</i> <i>IPhone OS 2.0</i>
2009	<i>Windows 7</i> <i>Palm webOS</i> <i>IPhone OS 3.0</i>
2010	<i>Windows Phone 7</i> <i>Windows Mobile 7</i>

Fuente: <http://toastystech.com/guis/guitimeline.html>, GUI Timeline

Todas las interfaces de usuario gráficas que se mencionan en la tabla anterior, presentan las siguientes características.

- Aparición de los sistemas gráficos que presenta ventanas, íconos, menús y elementos apuntadores.
- Visualizan diferentes tipos de información simultánea.
- Sistemas de menús desplegables más sencillos de usar.
- Íconos y elementos de asociación gráfica que reducen el uso del teclado para especificar órdenes.

Cabe destacar que para los sistemas operativos Linux o variantes de Unix, existen interfaces gráficas independientes, las cuales en su mayoría trabajan con la tecnología *X Window System*. A continuación se listan algunas interfaces gráficas de ese tipo y se remarcarán las que están impulsando las interfaces en 3D.

Tabla III. Interfaces independientes para Unix o Linux

Interfaz	Características
BERYL (3DUI)	Proyecto para la rama Quininstorm de Compiz, el cual incluye componentes de 3D para el manejo en cubo del entorno de escritorio.
COMPIZ (3DUI)	De los primeros gestores de ventanas para Unix, el cual aprovecha la aceleración del OPENGL.
GNOME	El proyecto GNOME (<i>GNU Network Object Model Environment</i>), surgió en agosto de 1997 como proyecto liderado por los mexicanos.
KDE	KDE (<i>K Desktop Environment</i>), es un entorno de escritorio gráfico e infraestructura de desarrollo para sistemas.

Los gestores de ventanas mencionados, al igual que el sistema operativos, fueron creados por proyectos que promueven la distribución gratuita de *software*, por ello existe un gran número de versiones, y por su código abierto variaciones.

1.5 Diferencia entre interfaces de una y dos dimensiones

A continuación se mencionan las diferencias más elementales que permitieron la clasificación de las interfaces de una y dos dimensiones:

- Las CUI por ser gráficamente simples no consumen muchos recursos del equipo de cómputo donde se ejecuta.
- El manejo de las CUI está basado en el conocimiento del lenguaje estricto que posee, es decir, que para su manejo el usuario debe conocer las instrucciones que puede ejecutar.

- Las CUI son más fáciles de desarrollar que las GUI, por su sencillez gráfica.
- Las GUI presentan un ambiente más atractivo y fácil de usar para usuarios inexpertos; sin embargo, durante su desarrollo requiere más tiempo en diseño para cumplir con su objetivo.
- Las interfaces gráficas de usuarios permiten la realización de tareas más complejas por la facilidad que presentan las gráficas.
- Las GUI consumen un porcentaje más alto en los recursos de la computadora que las CUI.
- Las GUI actuales utilizan una estructuración lógica para la localización de sus opciones; sin embargo, no incluyen los objetos con motricidad y despliegue real que aceleren la percepción.

1.6 Clasificación de las interfaces de usuario

Existen varias clasificaciones de interfaces de usuario, según la perspectiva que se desee evaluar. A continuación se describe la clasificación general de las interfaces de usuario a través de la interacción que el usuario ejerce sobre ella, esta propuesta fue hecha por Hammer en 1983, y divide a las interfaces en 2 tipos.

- Descriptivas

El usuario introduce una orden o instrucción que la interfaz interpretará para su ejecución. Las interfaces de usuario descriptivas, se pueden delimitar como todas las CUI, ya que en general éstas son líneas de comando. Cabe resaltar que no existe sistema operativo que no contenga este tipo de interfaz como parte de él.

- *Libre de forma*: utiliza un lenguaje casi natural, el cual es relativamente libre de forma. Este tipo de interfaz es utilizado en su mayoría por lenguajes para inteligencia artificial.
- *Sintaxis estricta*: utiliza un lenguaje definido, el cual es un conjunto de instrucciones estructuradas con parámetros para su ejecución; tal es el caso de las interfaces de consola de UNIX.

- Selectivas

El usuario selecciona una de las diversas alternativas presentadas por la interfaz. En general todas las interfaces selectivas se resumen a todas las GUI actuales, por presentar un conjunto de componentes visuales de fácil acceso y manejo.

1.7 Interfaces en 3D

Una interfaz en tercera dimensión será considerada como una interfaz de usuario, en la cual existen objetos reales con comportamiento y manejo lo más aproximado a la realidad. Estos poseerán características que permitirán realizar las tareas específicas a los usuarios, de forma intuitiva, utilizando la experiencia y conocimiento natural del hombre.

En el cine se han creado muchas escenas, en la cuales se observan aplicaciones holográficas que permite buscar y ver archivos a través del toque de los objetos reflejados en el holograma. Con el desarrollo de las interfaces en 3D y sus dispositivos específicos estas escenas podrán llevarse a la realidad, permitiendo inclusive un despliegue holográfico a tamaño real de los objetos.

Además, una interfaz en 3D debe tener una incorporación completa de objetos reales, los cuales deben manejarse a semejanza de la realidad, incluyendo su movimiento y reacción. Para ello deben aplicarse requerimientos de comportamiento y manejo que permiten dar lineamientos para el desarrollo de una interfaz de usuario en tercera dimensión. Una aplicación práctica de una interfaz en tercera dimensión sería un sistema de comercio electrónico, el cual podría mostrar los productos en anaqueles tal y como sucede en la realidad en un supermercado.

Las interpretaciones sobre el comportamiento, función y manejo de una interfaz en 3D, son variadas; se ha incorporado el movimiento de objetos geométricos básicos a algunas funciones de sistemas operativos, la mayoría para movilizarse dentro de varias áreas de trabajo, comúnmente conocidas como “Escritorio”; sin embargo, no son totalmente una interfaz en 3D.

En los siguientes capítulos, se detallaran los requerimientos de comportamiento y de funcionamiento interno para el desarrollo de una interfaz de este tipo, y se generará un modelo de evaluación de métricas para analizar el impacto durante el análisis y desarrollo de *software*. Además, se detallarán las tecnologías que actualmente permiten el desarrollo de las interfaces en 3D, y cómo el uso del mouse y otros dispositivos, deberá ser reemplazado por otros dispositivos de entrada específicos, que agreguen la variable de profundidad, reconocida científicamente como el eje z.

La variedad de características que estos dispositivos esperan poseer, pretenden adecuarse a las necesidades de todos los usuarios, incluyendo aquellos que posean alguna discapacidad.

2 REQUERIMIENTOS DE COMPORTAMIENTO Y MANEJO DE UNA INTERFAZ EN 3D APLICADOS A TRAVÉS DE METÁFORAS

Una interfaz de usuario en 3D, al igual que cualquier otro tipo de interfaz, debe poseer claramente definidos los objetivos de su creación, teniendo una estricta relación con las funciones objetivo del *software* donde será utilizada. En el caso de las interfaces en 3D, se recomienda que además del cumplimiento de sus objetivos, se apliquen los requerimientos de comportamiento durante su diseño y desarrollo.

Los requerimientos de comportamiento se definen como un conjunto de estándares que garantizan la mayor eficacia en la usabilidad que el usuario obtiene de una interfaz de *software*. Además de los requerimientos más importantes que se tratarán, se encuentra la aplicación de metáforas para la creación de interfaces.

Una metáfora en este contexto se conoce como la relación o semejanza que los objetos de una interfaz en tercera dimensión tienen hacia objetos de la realidad, comparando su estructura, manejo y comportamiento. Como ejemplo funcional dentro de los sistemas operativos actuales se encuentra la “Metáfora del Escritorio” utilizada en su mayoría por Microsoft y Macintosh, la cual consiste en pequeñas carpetas con ficheros dentro de ellas, que se pueden arrastrar para poder moverlos entre ellas.

Este capítulo se centra en la identificación de los requerimientos de comportamiento y manejo, que son utilizados en la aplicación de metáforas para el diseño formal de interfaces en tercera dimensión.

2.1 Requerimientos fundamentales de comportamiento

Proporcionan lineamientos de interacción, basados en las mejores prácticas del manejo y estructura de las interfaces de *software*. Permiten bajo una correcta aplicación que los usuarios puedan explotar las capacidades del *software*, de forma sencilla e intuitiva. Dentro de sus principales ventajas a nivel de usuario pueden mencionarse:

- Personalización de las interfaces para amigabilidad del *software*
- Maximización de la usabilidad del *software*
- Minimización de la curva de aprendizaje para uso o manejo del *software*

Como parte del análisis y diseño del *software*, en la construcción de las interfaces en tercera dimensión permiten:

- Optimización del capital humano para desarrollo del *software*
- Interoperabilidad entre los componentes visuales que conforman la interfaz en tercera dimensión

A continuación se definirán los requerimientos fundamentales de comportamiento que deben aplicarse en la creación de interfaces en tercera dimensión.

a. Invitación

Se define como la relación entre objeto y usuario, permite que los usuarios puedan reconocer la tarea que realiza el objeto a través del reconocimiento visual del mismo. Cualquier objeto que se incluya como parte de una interfaz debe poseer su razón de existir. Una interfaz en tercera dimensión implica incluir objetos reales que realicen las mismas acciones o acciones semejantes.

Un caso muy sencillo es el uso de la impresora, cuando el usuario quiere enviar a imprimir, busca generalmente un icono en forma de impresora, porque sabe de forma abstracta que con este ícono va poder llevar a cabo su tarea de impresión de un documento, fotografía, etc.

Figura 3. **Icono de impresora**



Fuente: <http://yolvi.files.wordpress.com/2009/05/impresora-epson.jpg>

b. Proximidad

Este requerimiento define la capacidad del *software* para desplegar las opciones de uso común, de tal forma que el usuario no supere 3 órdenes para ejecutar su tarea. Una orden puede entenderse actualmente como un clic realizado con el mouse o bien oprimir una o varias teclas simultáneamente del teclado.

Un ejemplo muy práctico de la aplicación de este requerimiento, sería la “Papelera de Reciclaje” de los sistemas operativo Microsoft, la cual se encuentra creada en el escritorio o área principal de trabajo, y permite acceder a todos aquellos archivo eliminados dentro del sistema operativo.

Figura 4. **Icono de la papelera de reciclaje**



Fuente:

http://4.bp.blogspot.com/_AGSpYDvydis/SwgLuSMctRI/AAAAAAAAAR1U/azliGa183sU/s1600/papelera+de+reciclaje.png

Debe permitir que los usuarios con menos experiencia tengan acceso a las opciones más utilizadas de forma rápida y sencilla, mientras que las opciones avanzadas deben quedar más alejadas para los usuarios expertos. Un ejemplo de este requerimiento puede ser un mecánico, quien desea tener más a la mano su herramienta más utilizada.

c. Técnicas de interacción

Son técnicas que tratan de entender las capacidades humanas del movimiento, comunicación, percepción y razonamiento, para mapearse a través de la programación de *software* y generar en la interfaz de usuario canales de comunicación que produzcan la máxima interacción con el usuario. A continuación se describen según el orden en que el usuario las percibe dentro de una interfaz:

- Navegación
 - Selección y manipulación
 - Sistemas de control
- **Navegación**

Representa el soporte del conocimiento espacial, es decir, proveyendo un movimiento eficiente y cómodo entre los distintos objetos de la interfaz, para que los usuarios puedan centrarse en ejecutar sus tareas más fácilmente. La navegación dentro de un ambiente en tercera dimensión, posee una gran diferencia de sus antecesoras, debe considerar la profundidad en la que se encuentran los objetos.

La navegación estará dividida en dos componentes:

Tabla IV. **Tipos de navegación**

Componente	Descripción
Recorrido	Reconocido como el motor de la interfaz, se encarga de garantizar que exista movimiento o traslado entre los objetos interfaz de usuario.
Camino de búsqueda	Garantiza que el movimiento y traslado generado por el recorrido se realice de forma cognoscitiva o con sentido.

Con el trabajo sistemático, ambos componentes generan 3 tipos de tareas dentro de la navegación:

Tabla V. **Cronología de *hardware* vs. sistema operativo**

Tarea	Definición
Exploración	Reconocido como la capacidad de movilizarse entre los objetos interfaz de usuario.
Búsqueda	Se da cuando el objetivo principal es localizar un objeto para ejecutar una acción específica.
Manejabilidad	Se considera una tarea de poco rango y movimiento de alta precisión usado para localizar un punto donde se tenga el mejor desempeño para una tarea en particular. Una forma fácil de comprender esta última categoría es cuando se posiciona el puntero del mouse en el botón para ejecutar una acción en una aplicación.

- **Selección y manipulación**

La selección y manipulación, son acciones de la interacción que complementan a la navegación, dentro de las interfaces en tercera dimensión deben proveer al menos tres tareas básicas, que se describen a continuación:

Tabla VI. **Tareas básicas de selección y manipulación**

Tarea	Definición
Posicionamiento sobre objetos	Reconocido como el motor de la interfaz, se encarga de garantizar que exista movimiento o traslado entre de los objetos interfaz de usuario.
Selección de objetos	Permite remarcar o hacer sobresalir un objeto de la interfaz, en el cual el puntero ya se encuentre posicionado.
Rotación o traslado de objetos	Permite rotar o mover un objeto seleccionado, hacia alguna ubicación de espacio dentro de la interfaz.

Las tareas descritas anteriormente han sido elegidas como básicas, considerando la metáfora de la capacidad de la mano, ya que como toda representación de la realidad ésta provee de la interacción sobre objetos de uso común, y permite al ser humano interactuar con los objetos. Un ejemplo práctico, es ubicarse dentro de un escritorio y buscar un expediente o archivo, como normalmente se hace hasta dejarlo frente a la vista para revisarlo.

- **Sistema de control**

Dentro de las técnicas de interacción, existe un monitor el cual controla todos los parámetros de la configuración de la interacción, permitiendo cambiarlos entre varios estilos de entrada, conocidos como perfiles, que el usuario puede controlar para mejorar su funcionamiento. Tal es el caso de la velocidad del movimiento, de los objetos, de la selección, la calidad de detalle visual de los movimientos, sensibilización de las señales de los dispositivos de entrada, etc., un ejemplo claro del sistema de control es el panel de control del sistema operativo Windows.

d. Predicción

Específica con claridad el funcionamiento de los objetos y controles creados en la interfaz, retomando, un ejemplo de predicción en las GUI de 2 dimensiones se da al crear los accesos directos, porque estos deben representar claramente el enlace al programa o archivo al cual hacen referencia. La predicción en la interfaz de tercera dimensión también es conocida como la familiarización, la cual induce al usuario a tener claro que pueden hacer los objetos incluidos según la realidad que representen. Como ejemplo el objeto libro, el cual claramente al distinguirlo, se sabe que tiene un contenido para lectura.

e. Simplicidad

Se caracteriza por abstraer los aspectos básicos de comportamiento y función de los objetos, para ser adaptados y creados dentro de una interfaz en tercera dimensión. Es decir, si se desea leer un libro y se incluyera como un objeto, se debería considerar.

- Buscar y seleccionar un libro de una librería, a través del título de su lomo o de su portada
- Abrir el libro para lectura
- Cambiar de página para continuar la lectura
- Ir a un número de página
- Cerrar el libro y guardar el libro

Sin embargo, el hacerlo simple, no implica que se pierda los elementos esenciales de un libro, como el número de página o la sensación de darle vuelta a la siguiente página, para realizar una búsqueda por página.

Un objeto en tercera dimensión debe ser simple y sencillo de utilizar, pero considerando todos los detalles que lo acerquen a la realidad para un buen funcionamiento. Un correcto uso de la simplicidad aprovecha todos los recursos disponibles para la interfaz en tercera dimensión.

f. Consistencia

Se encarga de asegurar la uniformidad del estilo, estructura, función y distribución de la interfaz de usuario.

Para crear la consistencia en una interfaz 3D, el primer punto a considerar es la dimensión de los objetos, es decir, para asegurar el tamaño correcto de los objetos se recomienda aplicar una escala de medida real con relación de proporción entre objetos.

Una vez creado un objeto en su totalidad, su estilo y función deben ser los mismos, no importando la ubicación donde se encuentren, ya que este es un principio de la memoria humana que relaciona los objetos con su función.

La aplicación de la consistencia aplica de igual forma a los controles de navegación. El estilo de un objeto o un control incluye:

- Tamaño
- Proporción
- Rotación
- Movimientos
- Aproximación
- Luz
- Sombra
- Color
- Textura
- Fuente

En una GUI de segunda dimensión, la consistencia ha llegado a una madurez muy alta y los objetos únicamente requieren la aplicación de estándares. A diferencia de los objetos para una interfaz en 3D, la consistencia debe validarse a través del contraste.

g. Contraste

Evalúa la consistencia, para hacerlo toma todos los objetos y controles y los contrasta con la realidad, tomando como base dos preguntas esenciales:

- ¿Qué objetivo tiene?
- ¿Qué parte de la realidad se pretende representar?

Con la primera pregunta se confirma que el objeto sea el adecuado para cumplir con la función objetivo y cumpla con la función esperada o definida, y en la segunda se ve si está aplicando la realidad al objeto.

h. Retroalimentación

La retroalimentación se presenta cuando se quiere representar el estado de los controles o de los objetos dentro de la interfaz, logrando una mayor comunicación con el usuario. Un ejemplo sencillo de entender se da cuando el puntero del *mouse* se posiciona sobre algún ícono, y éste cambia de forma para indicar que el puntero se encuentra sobre él, a este tipo de retroalimentación se le llama lectura de mapas sensibles.

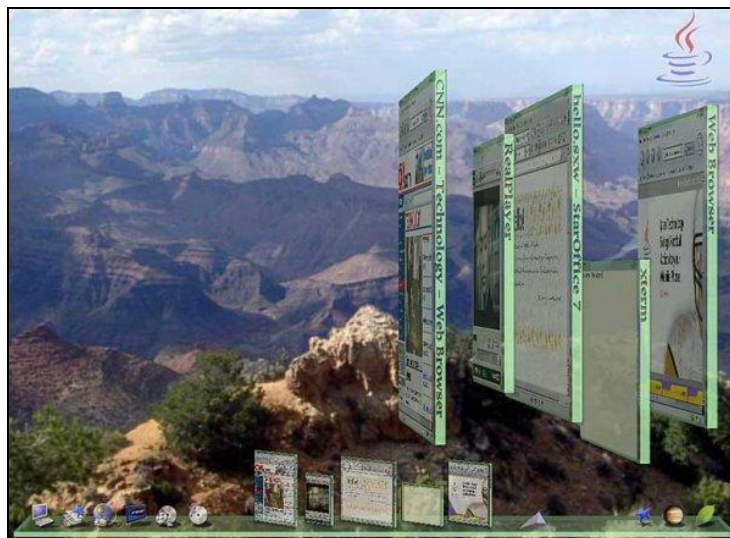
Con los mapas sensibles se utiliza un control de estados para los objetos, con el propósito de recopilar información que pueda ser utilizada para pronosticar con qué frecuencia el usuario utiliza el objeto, tanto en tiempo como número de veces. Esto permite personalizar el *software* de tal forma que el usuario tendrá todas sus opciones a la mano, de forma inteligente, a través de criterios de uso y visita de objetos.

i. Objetos de interfaz real

Cuando se habla del mundo real, se está hablando de realidad virtual, un factor que ha venido creciendo desde hace unos años, y el cual se ha puesto en práctica en cinematografía y juegos, donde se han creado mundos tan reales que nadie notaría la diferencia.

Este es el objetivo del *software* con interfaces en tercera dimensión, tratar de llevar a cabo todas las tareas con la mayor semejanza al mundo real. La facilidad y comodidad de tener objetos más reales brinda más comodidad y más confianza en el uso del *software*. “*Looking Glass* el proyecto de *Sun Microsystems*, presenta en su versión de beta, un ambiente natural en el cual se pueden utilizar aún algunas aplicaciones de dos dimensiones con funcionamiento como la galería de tareas”³.

Figura 5. Ejemplo del proyecto *Looking Glass*



Fuente: http://www.gentoo.org/images/gwn/20050124_looking-glass.jpg

³ HEISS, Janice J. Going 3D Project Looking Glass. Oracle. Octubre 2004. [ref. 10 de agosto de 2010]. Disponible en Web: <<http://java.sun.com/developer/technicalArticles/J2SE/Desktop/lookingglass/>>

j. Control y libertad por el usuario

El usuario debe sentir que es capaz de controlar la interfaz con facilidad y utilizar su sentido común y sus capacidades como ser humano para manejarlo con completa libertad, tratando de que necesite la menor cantidad de conocimientos de tecnología, sin dejar escapar las tareas que debe realizar como usuario. Algunas de estas capacidades ya han sido explotadas para dar al hombre comodidades aplicadas en algunas tecnologías. A continuación se mencionan, para su aplicación, dentro de las interfaces en tercera dimensión:

- Reconocimiento de movimientos
- Reconocimiento de la presión del tacto
- Reconocimiento de voz
- Reconocimiento del pensamiento

Sin embargo, es preciso mencionar que algunas de ellas aún se encuentran como prototipos, o bien no tiene una aplicación directa y formal para equipos de cómputo.

- **Reconocimiento de movimientos**

Todos los objetos de una interfaz tienen funciones específicas, mientras se interactúa con uno, deben implementarse métodos de inteligencia artificial como redes neuronales y/o lenguaje natural, que permitan al usuario ejecutar acciones a través del movimiento.

Existen famosas consolas de juegos como el *Wii* y el *Play Station 3*, que permiten interpretar los movimientos, sin embargo, estos ya están programados.

El propósito del reconocimiento es que, además de las acciones predefinidas, tenga la capacidad de aprender a ejecutar esas acciones. El ejemplo a este requerimiento, dentro de interfaces de dos dimensiones, serían los macros de *Microsoft Office*, los cuales se programan para ser ejecutados muchas veces por el usuario.

- **Reconocimiento de la presión del tacto**

Existen dispositivos de entrada que permiten utilizar la tecnología “*Touchscreen*” en interfaces de usuario, las cuales a través de una pantalla táctil, son capaces de detectar los movimientos y la acción una interfaz en 3D; sin embargo, existe aún un problema pendiente para probar en su totalidad esta tecnología en una interfaz 3D, la detección de la presión del tacto en pantalla para detectar, la profundidad y la fuerza con que el cursor debe moverse en pantalla.

- **Reconocimiento de voz**

Durante los últimos años, el hombre ha buscado a través de la tecnología ayudar a los seres humanos que por naturaleza tiene dificultad del movimiento, esto ha provocado la creación de tecnologías que permitan reconocer instrucciones de voz. Sin embargo, en un mundo globalizado las tecnologías de punta y la comodidad, han generado que se avance más rápidamente. En los sistemas operativos como *Macintosh* o *Microsoft Windows* ya implementan ésta tecnología.

El reconocimiento de voz tiene como limitantes expandir su diccionario de instrucciones y poder reconocer idiomas o palabras de un mismo idioma, que son muy parecidos y dependen de su clara pronunciación.

- **Instrucciones del pensamiento**

El ser humano, por naturaleza, realiza sus movimientos y acciones a través de señales nerviosas generadas en el cerebro, que viajan por el sistema nervioso hasta cumplir con su tarea. Una interfaz de 3D pretende realizar de igual forma estos movimientos; sin embargo, aún no existe una tecnología madura para realizar este control de interfaz.

2.2 Aplicación de una metáfora

La aplicación de las metáforas debe realizarse durante el análisis del *software*, e integrarse a la metodología que se esté utilizando, con el fin de mantener los estándares que se apliquen durante todo el desarrollo del *software*.

Para crear una metáfora, es esencial no confundir la comparación o semejanza de los objetos de interfaz de usuario con los reales y la comparación de objetos reales con objetos internos funcionales del *software*, que algunas metodologías proponen para facilitar su construcción, ya que son procedimientos completamente diferentes.

Para la creación de una metáfora de una interfaz en 3D, se recomienda responder las siguientes preguntas que facilitarán su correcta formulación:

- ¿Cuál es el objetivo principal?
Permitirá asegurar que ésta cumpla con los requerimientos necesarios para el *software* en el que será utilizada.

- ¿Cómo funcionará la metáfora en distintos escenarios?
Se describirá específicamente la función de la metáfora, cuáles serán sus límites y alcances, y la forma de trabajo en escenarios de acciones que el usuario debe realizar.
- ¿Es necesario aplicar sub-metáforas?
Según la complejidad de una metáfora, puede llegar a contener sub-metáforas, las cuales se recomienda identificar para facilitar los cambios cuando sean necesarios.

Estas preguntas permiten establecer un marco de trabajo claro para describir todas las metáforas que se consideren necesarias para aplicarse a una interfaz de usuario en 3D.

Una de las primeras metáforas conocida como “La Galería de Tareas”, en la cual se utiliza el estilo de interfaz de aplicación tipo ventana, donde se pueden acomodar tareas de forma tridimensional, como el papel dentro del escritorio, tenía como objetivo permitir al usuario tener una visión de todas sus tareas a través de una sola vista o perspectiva, dándole al usuario la capacidad de acomodar sus tareas como cuadros de pintura y poder seleccionarlos según lo deseara.

Figura 6. **Ejemplo de la metáfora de galería de tareas**



Fuente: *An Interaction Model for Designing Post-WIMP User Interfaces*. Michel Beaudouin-Lafon

Ficha Descriptiva de Metáfora

Cada metáfora creada debe ser documentada, para ello se sugiere la siguiente ficha descriptiva y el significado de cada dato:

- ID de identificación: permite identificar de manera abreviada y sencilla una metáfora.
- Nombre: título con el cual se identificará.
- Clasificación: especifica la funcionalidad del *software* que cubrirá.
- Objetivo: describe el fundamento de su creación.
- Descripción: describe detalladamente a la metáfora.
- Límites y alcances: permitirá establecer qué aspectos o acciones realizará y cuáles no.
- ID metáfora padre: si fuera una sub-metáfora identificará a su metáfora padre.

- ID sub-metáforas: en el caso de ser una metáfora padre, se especificarán todas sus sub-metáforas.
- Objetos: listará los objetos de la realidad o interacción, que forman parte de la metáfora.

Figura 7. Ejemplo de plantilla de ficha descriptiva

Ficha Descriptiva de Metáfora		<ID Metáfora>
Nombre:	<Nombre de la Metáfora>	
Clasificación	<Seleccionar Funcionalidad>	
Objetivo	<Objetivo>	
Descripción:	<Descripción detallada>	
Límites y Alcances:	<ul style="list-style-type: none"> • <Lista de Límites y Alcances> • <Lista de Límites y Alcances> • <Lista de Límites y Alcances> 	
ID Metáfora Padre:	<ID Metáfora Padre>	
ID Sub-metáforas:	<ol style="list-style-type: none"> 1. <Lista de Sub-metáforas> 2. <Lista de Sub-metáforas> 3. <Lista de Sub-metáforas> 	
Objetos:	<ol style="list-style-type: none"> 1. <Lista de Objetos> 2. <Lista de Objetos> 3. <Lista de Objetos> 	

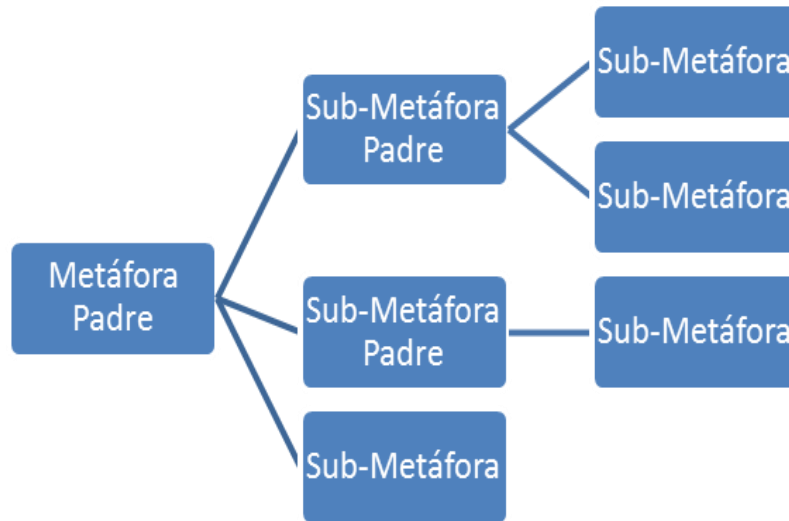
2.3 Diagramas de metáforas

A continuación se recomienda la elaboración de diagramas que permiten obtener una visión clara y práctica de las metáforas creadas:

a. Diagramas de relación de metáforas

Permite desplegar la relación entre las metáforas padre y sub-metáforas, validando la correlación e integración entre ellas.

Figura 8. **Ejemplo de diagrama de relación de metáforas**



b. Árbol de metáforas según clasificación

Permite desplegar qué metáforas serán utilizadas para cumplir con cada funcionalidad específica del *software*, permitiendo evitar la duplicidad de acciones y la integración de las metáforas.

Figura 9. **Ejemplo de árbol de metáforas según clasificación**



2.4 Evaluación de metáforas

La evaluación de las metáforas permite establecer un mecanismo de control que valide su correcta creación e integración. Cuando una o varias metáforas no superen la evaluación, éstas deben ser incluidas en un ciclo de revisión, el cual permita concluir si vale la pena modificarlas o si bien es necesario desecharla. Este ciclo se conforma por el grupo de evaluaciones aplicables, que se detallarán en los siguientes ítems.

Es tal el impacto de una metáfora mal creada, que al aplicarse a una interfaz de usuario se dice que es peor que no haber diseñado una. Puede causar un aumento en la complejidad de la comprensión del usuario y por lo tanto dificulta su manejo.

Un ejemplo práctico de una mala aplicación fue el Maletín en Microsoft Windows 95, consistía en que todos los archivos colocados ahí se almacenaban para luego llevarse a casa u otro lugar; sin embargo, ésta requería de una segunda acción, que era copiar todo el contenido del maletín al dispositivo de almacenamiento portátil. Esto generó la siguiente cuestión: ¿Para qué utilizar el Maletín si los puedo almacenar directamente al dispositivo de almacenamiento?, y por ende la desaparición del concepto en las posteriores versiones de Microsoft Windows.

Para evaluar las metáforas creadas, y determinar si éstas son aplicables para el desarrollo de *software*, es necesario realizar como mínimo las siguientes evaluaciones:

- Evaluación visual de diagramas
- Evaluación de objetos
- Evaluación externa

Es importante mencionar que estas evaluaciones deben integrarse como parte de la batería de pruebas del análisis y diseño del *software*, además de realizarse la cantidad de veces que sea necesaria para obtener los resultados esperados.

a. Evaluación visual de diagramas

A través del estudio visual de los diagramas de relación de metáforas y del árbol de metáforas, se debe determinar que las metáforas están correctamente integradas, de tal forma, que asemejen el concepto de un sistema, el cual integra a todos sus elementos y la cooperación entre sí, para cumplir su objetivo.

b. Evaluación de objetos

Cada metáfora contiene uno o varios objetos, los cuales pueden considerarse como controles de navegación u objetos de la realidad en una interfaz en 3D. Es necesario aplicar una revisión general de los requerimientos fundamentales a cada objeto y garantizar que estos se cumplan. Para realizar dicha evaluación se sugiere utilizar el instrumento de evaluación de requerimientos, el cual permitirá validar el cumplimiento de los mismos.

Figura 10. Ejemplo de evaluación de requerimientos fundamentales

Evaluación de Requerimientos Fundamentales de Comportamiento y Manejo															
	1. Invitación	6. Predicción	11. Objetos de interfaz real												
	2. Proximidad	7. Simplicidad	12. Reconocimiento de movimientos												
	3. Navegación	8. Consistencia	13. Reconocimiento de la presión del tacto												
	4. Selección y Manipulación	9. Contraste	14. Reconocimiento de voz												
	5. Sistema de Control	10. Retroalimentación	15. Instrucciones del pensamiento												
Objeto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Objeto 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Objeto 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Objeto 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

c. Evaluación Externa

Para evaluar las metáforas creadas y determinar si éstas son aplicables para el diseño en tercera dimensión, es necesario validar la representación de estas metáforas, en una muestra del segmento de usuarios objetivo, para el cual se esté desarrollando el *software*.

Como primera fase, deben elaborarse mapas conceptuales y una caricaturización básica de las metáforas y su funcionamiento. Las cuales deben entregarse a cada usuario de la muestra que apoyará la evaluación externa de forma individual; se les pedirá que documenten con una descripción detallada su razonamiento, tanto de los mapas como la caricaturización. Posteriormente, se realizará el mismo ejercicio en forma grupal.

Se recomienda que los grupos sean mayores de 5 y menores de 11 integrantes, esto completará la segunda fase de la evaluación.

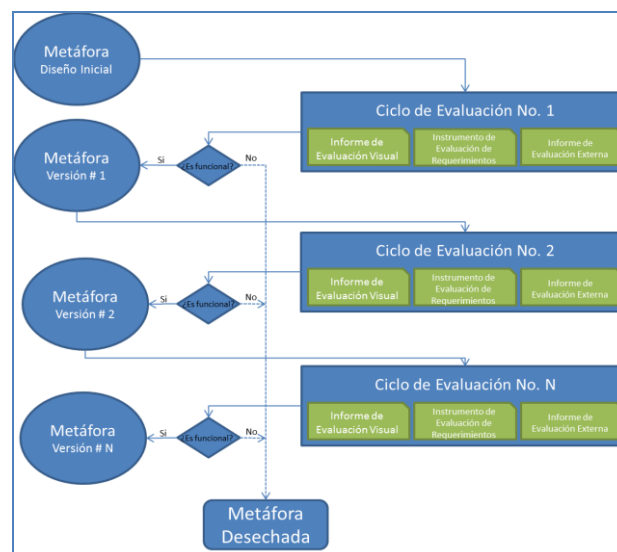
La fase final de evaluación implicará la recopilación de las descripciones para cada una de las metáforas, permitiendo realizar un análisis que calificará cada metáfora y permitirá realizar las mejoras o correcciones que sean determinadas.

2.5 Manejo de metáforas

Desde su creación cada metáfora debe clasificarse y organizarse adecuadamente, ya que dependiendo del objetivo del *software*, la cantidad de metáforas puede convertirse en un problema durante el análisis y desarrollo. Es necesario incluir toda la información o documentos de las metáforas, como parte de la administración de configuración de la metodología de desarrollo de *software* que se esté utilizando.

Para el manejo de metáforas se recomienda utilizar como mínimo el siguiente árbol de versiones, el cual comprende las evaluaciones visuales de diagramas, de objetos y externas explicadas anteriormente.

Figura 11. Ejemplo de diagrama de evaluación de metáforas



El diagrama se describe que para cada metáfora, en su diseño inicial, se le deben realizar N ciclos de evaluaciones visuales de diagramas, de objetos y externas explicadas, como se explicaron anteriormente. El resultado de cada ciclo debe mostrar si la metáfora es funcional o no. De concluir que una metáfora no es funcional ésta debe desecharse.

Una vez que se considera que la metáfora a llegado al nivel deseado de madurez, el diseño se encontrará listo para pasar al proceso de construcción o desarrollo de la misma.

3 FACTORES TECNOLÓGICOS Y REQUERIMIENTOS DE FUNCIONAMIENTO INTERNO DE UNA INTERFAZ EN 3D

Se han creado muchas tecnologías que permiten diseñar y crear entornos en tercera dimensión, la mayoría para las industrias de juegos, cine y televisión, estas tecnologías requieren alto rendimiento en procesamiento gráfico y de datos, para poder desarrollarse y ejecutarse con un buen desempeño. Su objetivo ha sido dar el máximo realismo posible a los entornos que crean y permitir a la personas simular el sentimiento del realismo.

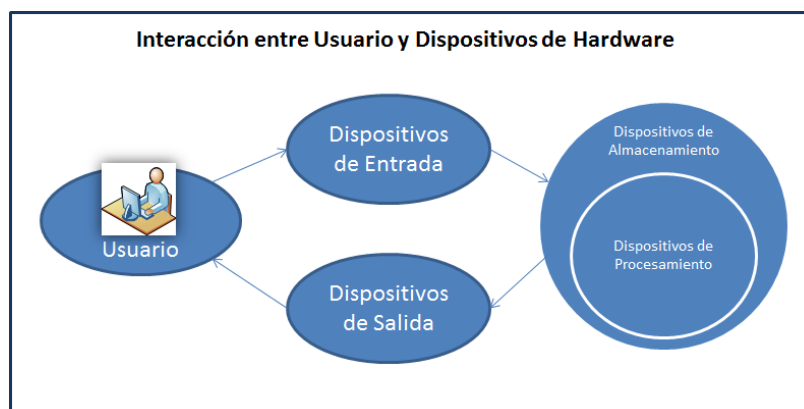
Sin embargo, como toda tecnología popular éstas son cada vez más accesibles a los desarrolladores y usuarios, con algunas limitantes de procesamiento de datos y recursos gráficos. Para compensar estas limitantes se han creado requerimientos de funcionamiento interno que permiten crear controles durante el desarrollo y la etapa de pruebas del *software* específicamente, las cuales permiten optimizar el uso de los recursos limitados.

Este capítulo resumirá las tecnologías de *hardware* y *software* para el desarrollo de las interfaces en 3D, considerando las ya disponibles y las que se encuentran en construcción, además considerará los requerimientos de funcionamiento interno que debe poseer el *software* con interfaz en 3D para tener un desempeño y funcionamiento adecuado adaptado a las capacidades del *hardware*.

3.1 Tecnologías de *hardware*

Para detallar el *hardware* que se requiere en una interfaz en 3D, se clasificará según la comunicación que éste ejerce con el usuario, como se muestra a continuación:

Figura 12. **Diagrama de interacción de usuario y dispositivos de *hardware***



En cada clasificación se resaltarán los atributos necesarios para que el *hardware* pueda ser utilizado dentro de las interfaces en tercera dimensión, así como una descripción de sus características. Es importante resaltar que existen varios proyectos en universidades de Estados Unidos y Asia específicamente aún en construcción, cuya información aún no es pública dirigidos a la integración de *software* para uso de la tercera dimensión.

3.1.1 Dispositivos de entrada

Un dispositivo de entrada, es un elemento físico que transforma la información que llega del usuario en datos y órdenes que el *software* procesa a través de la computadora.

A continuación se listan las características esenciales que debe poseer un dispositivo de entrada para que sea útil:

- Adaptarse a las características físicas y psicológicas de los usuarios
- Incentivar la capacidad de aprender de un usuario
- Acoplarse con las características del trabajo y del entorno donde funcione

Cuando un dispositivo entrada cumple con todas sus características de utilidad, se clasifica según el envío de datos. En el siguiente cuadro se describen los dos tipos de clasificación según el envío de datos que realizan:

Tabla VII. **Clasificación de dispositivos de entrada según el envío de datos**

Tipo	Descripción	Ejemplos
Discreto	Envían datos a través de una o varias posiciones, considerando bloques de datos asilados en distintos puntos del tiempo.	Teclados, botones de pulso de gamepads, mouse , etc.
Continuo	Envían datos de forma continua durante un período de tiempo.	Palancas de movimiento de <i>joysticks</i> , mouse de <i>trackball</i> o infrarrojo, etc.

La mayoría de los dispositivos de entrada utilizados en los últimos años, poseen un problema para utilizarse dentro de una interfaz en 3D, ya que únicamente consideran dos dimensiones, formando rectas y caminos de punto a punto para movilizarse dentro de la interfaz. En una interfaz en 3D es necesario que los dispositivos de entrada incluyan la variable de profundidad que consiste en generar la percepción del individuo.

A continuación se describen los tipos de dispositivos de entrada, que según la interacción que realizan con el usuario y que cumplen con su aplicación dentro de la interfaz de tercera dimensión son:

- Pantalla de sensor de tacto –*touch screen*–
- Sensor de movimiento
- Reconocimiento gráfico
- Reconocimiento biométrico y de señales nerviosas del cerebro

a. Pantalla de sensor de tacto

Los dispositivos con pantalla de sensor de tacto, son conocidos comúnmente como “*Touch Screen*”, es una pantalla sensible la cual permite a través del toque directo o tacto en su superficie, el ingreso de datos y envío de órdenes, son clasificados como continuos y discretos al mismo tiempo, por poseer ambas capacidades de envío de datos.

Este tipo de dispositivos se ha creado a través de varias generaciones de implementación tecnológica desde su creación en 1971 por el Dr. Samuel C. Hurst, con la interfaz electrónica táctil. En la siguiente tabla se describen las distintas generaciones y cuáles de ellas son capaces de utilizar los 3 ejes para una implementación de una interfaz en tercera dimensión.

Tabla VIII. Implementaciones tecnológicas de sensores de tacto

Nombre	Descripción	Capacidad 3D “Profundidad”
Resistiva	Formada a través de varias capas, las cuales al entrar en contacto pueden determinar la posición y la presión que se realizó sobre ellas.	Si

Continúa Tabla VIII.

Nombre	Descripción	Capacidad 3D "Profundidad"
Onda acústica superficial	Genera ondas de ultrasonido que son absorbidas al momento de tocarse. El sensor puede calcular la onda de rebote para calcular la presión que se ejerció.	Si
Capacitivas	Creada a través de una cubierta de un conductor eléctrico, la cual al ser tocada por otro conductor permite encontrar la referencia exacta de la posición y el movimiento.	Si
Infrarrojos	Utiliza una red de sensores infrarrojos la cual, al interrumpirse la señal de los mismos, da la ubicación del contacto y del movimiento que realiza.	No
Galga extensiométrica	Formada de una estructura elástica, es capaz de detectar posición, movimiento y presión realizada sobre la misma.	Si
Imagen óptica	Utiliza la triangulación de imagen a través de sensores infrarrojos ubicados en las esquinas de la pantalla.	No
Tecnología de señal dispersiva	Utiliza la energía mecánica ejercida en cristal para la detección del movimiento y contacto; sin embargo, no permite detectar la presión ejercida.	No
Reconocimiento de pulso acústico	Utiliza la energía mecánica convertida en pulsos acústicos, al ponerse en contacto con cristal, que permite la ubicación de la posición.	No

El eje "Z" se define como la variable de profundidad, y aplicada en la práctica, en este tipo de dispositivos, es la cantidad presión que se ejerce durante el contacto con la superficie de los dispositivos, permitiendo realizar movimientos continuos en cualquier dirección. No obstante en una implementación de una interfaz en 3D con este tipo de dispositivos, posee un problema, ya que si se toma el movimiento natural se avanza al frente por medio de la presión y al mismo tiempo se moviliza a cualquier lado, pero si se requiere retroceder, es necesaria una acción o instrucción adicional para invertir el movimiento que genera la presión en la superficie del dispositivo.

b. Sensor de movimiento

Los sensores de movimiento tienen una aplicación madura en cine, seguridad, investigación y recientemente en consolas de juegos como Wii de Nintendo, PlayStation III de Sony, Kinect XBOX 360 de Microsoft, etc. Actualmente no existe un dispositivo a la venta exclusivamente para aplicarse en un interfaz de 3D; sin embargo, los controles de las consolas de juegos mencionadas son adaptables para realizar una conexión por computadora y poseen las características necesarias para su aplicación en un interfaz de 3D. Dentro de los sensores de movimiento en construcción para las interfaces en 3D están:

- Mouse 3D, que agrega el concepto de presión para simular el eje Z.

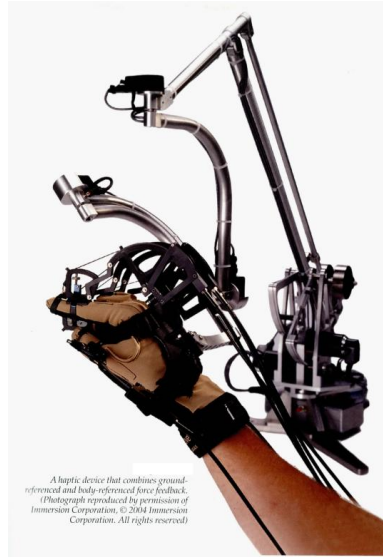
Figura 13. Ejemplo de *mouse* 3D



Fuente: <http://www.3dconnexion.com/products/spacepilot-pro.html>

- Guante de movimientos, se coloca en la mano y permite obtener información del movimiento de la mano a través de una serie de transistores.

Figura 14. Ejemplo de guante de movimientos



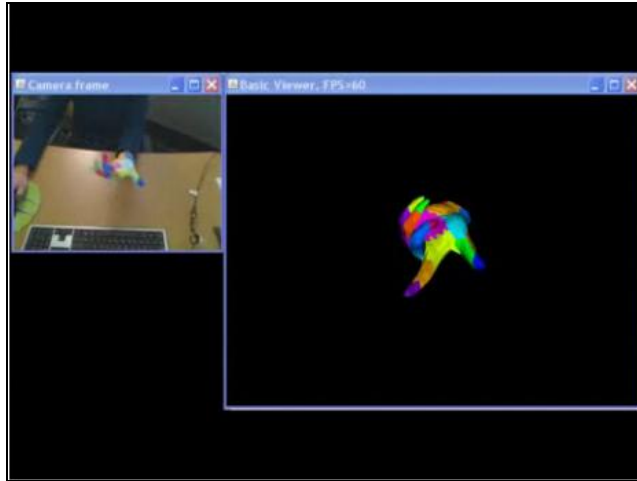
Fuente: Doug A. Browman, Ernest Kruijff, Joseph J. LaViola, Jr. Ivan Poupyrev, 3D User Interfaces: Theory and Practice, Pág. 76

c. Reconocimiento gráfico

Este tipo de dispositivos aún se encuentra en desarrollo, su función principal será la de reconocer gráficamente objetos, o elementos que permitan realizar movimientos e interacción a través del reconocimiento gráfico. Un dispositivo de reconocimiento gráfico posee las características de un sensor de movimiento.

Recientemente fue publicado el proyecto de “*Minoru 3D WebCam*”, el cual permite mover objetos utilizando una cámara web y un guante diseñado específicamente para ser reconocido mientras es enfocado, la interfaz en la cual fue probado permite generar un objeto en forma de una mano, la cual puede moverse libremente sobre una superficie y realizar una interacción con objetos en tercera dimensión.

Figura 15. **Ejemplo de reconocimiento gráfico**



Fuente: http://www.newlaunches.com/entry_images/0510/21/psychedelic-3D-gloves.php

d. Reconocimiento biométrico y de señales nerviosas del cerebro

Los dispositivos biométricos han sido diseñados en su mayoría para sistemas de seguridad complejos, los cuales aprovechan la identidad única que cada ser humano tiene según sus características digitales que lo distinguen. En el caso de los dispositivos para la detección de señales nerviosas la mayoría son utilizados con fines médicos.

Los dispositivos con la combinación de estas tecnologías pretenden dar accesibilidad a personas con problemas de paraplejía completa, capaces de utilizarse en el desarrollo de interfaces de tercera dimensión aún son propuestas de varios proyectos, los cuales incluyen los siguientes objetivos:

- Reconocimiento del iris humano, dilatación y movimiento del ojo para dar instrucciones de movimiento
- Capacidad de dar instrucciones y movimiento a través de señales provenientes del cerebro

3.1.2 Dispositivos de procesamiento

El procesamiento central se realiza a través del microprocesador que está compuesto básicamente por registros, unidades de control, aritmético-lógica, y dependiendo del tipo una unidad flotante. Ejecuta las instrucciones almacenadas secuencialmente en la memoria principal de la computadora realizando las siguientes fases para cada instrucción.

- Pre-Fetch, pre-lectura de la instrucción desde la memoria principal
- Fetch, envía la instrucción al decodificador
- Decodificación de la instrucción
- Lectura de operando, si la instrucción lo requiere
- Ejecución, que lleva a cabo el procesamiento
- Respuesta, escritura de los resultados en la memoria principal o en los registros;

Cada una de estas fases se realiza en uno o varios ciclos de CPU, para las interfaces en tercera dimensión, se recomienda que puedan ejecutar más de 10 instrucciones por ciclo.

La frecuencia de los procesadores se ha mantenido en el rango de los 1,5 GHz a 4 GHz, y trabajan con la tendencia de agregar núcleos encapsulados para aumentar el rendimiento por medio del procesamiento en paralelo.

Para medir el rendimiento entre procesadores para las interfaces en tercera dimensión, es únicamente posible si poseen una arquitectura similar, y además dependen de los componentes restantes del equipo de cómputo, en especial de la memoria RAM. Sin embargo, considerando una medición aproximada algunos expertos utilizan como referencia la cantidad de millones de instrucciones por segundo MIPS.

Para las interfaces en tercera dimensión, se debe contar con procesadores capaces de ejecutar al menos 25 000 MIPS, debido a la cantidad de instrucciones que deben ejecutar por la interacción del usuario.

a. Procesador(es) gráfico(s)

El procesamiento gráfico se basa en la capacidad de la tarjeta de video y sus procesadores integrados, los cuales realizarán ejecución gráfica de aplicaciones en tercera dimensión y permitiendo que él o los procesadores centrales realicen otras operaciones optimizando el funcionamiento del *software*.

Actualmente, existen varias marcas de tarjetas de video que ofrecen todas estas especificaciones y características técnicas para una interfaz en tercera dimensión; sin embargo, entre las más recomendadas por los diseñadores de sistemas operativos y juegos en tercera dimensión son:

- *NVIDIA GRAPHICS* (www.nvidia.com)
- ATI actualmente AMD (www.ati.com)
- INTEL (www.intel.com)

Las siguientes características o especificaciones técnicas, para las interfaces de tercera dimensión, se obtuvieron de la tecnología disponible de *NVidia Graphics*, una de las tecnologías que más avance ha tendido en la última década:

- *GPU*
- *GeForce 3D Vision Ready*
- *GeForce 3D Vision Surround Ready*

- *NVIDIA SLI® Ready*
- *NVIDIA Tegra*
- *Soporte para OpenGL 4.0*
- *Soporte DirectX 11.0 Hardware*
- *Soporte para el lenguaje de programación CUDA*

Es notable destacar que dentro estas características ahora es posible tener dos tarjetas de videos conectadas en formato “Dual”, para lograr aún más desempeño gráfico. Cuentan además con las siguientes características que optimizan el rendimiento y calidad:

- *Resoluciones de hasta 2,560x1,600 (digital) y 2,048x1,536 analógico*
- *Tecnología PureVideo® HD*
- *Tecnología PhysX™*
- *Compatibilidad con PCI-E 2.0 x16*
- *Soporta Shader Model 5.0*

3.1.3 Dispositivos de almacenamiento

a. Memoria principal

Es la memoria de trabajo del sistema operativo y programas, en ella se cargan todas las instrucciones que ejecutan el procesador y otros dispositivos. Se denomina por sus siglas en inglés “*Random Access Memory*”, -memoria de acceso aleatorio-, porque permite leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición dentro de ella.

Los sistemas operativos actuales requieren para un buen funcionamiento de 2 a 4 GB de memoria RAM. Para el desarrollo de interfaces en tercera dimensión, y considerando que ya se cuenta con memoria propia dentro de la tarjeta de video debe tenerse como mínimo 4GB.

Debe considerarse que éste es un recurso costoso y debe administrarse correctamente para que el *software* se ejecute correctamente y realice todas las tareas requeridas por el usuario eficientemente. Políticas de planificación para la memoria existen muchas, y en general las que actualmente se utilizan no tienen una documentación pública; sin embargo, están basadas en conceptos base como:

- *Entrelazado de direcciones*
- *Traducción de direcciones*
- *Algoritmos de asignación de memoria*
- *Memoria variable*
- *Memoria estática*
- *Fragmentación*
- *Compactación*

Para un mejor detalle de cada uno de estos conceptos se recomienda tratar el tema específico de gestión de memoria, considerando las herramientas de desarrollo y los lenguajes de programación seleccionados para el desarrollo de la interfaz en 3D, ya que internamente ofrecen ciertos lineamientos y recomendaciones a seguir en la optimización de memoria.

b. Espacio en disco duro requerido

El espacio requerido en disco duro para que una interfaz en tercera dimensión se ejecute, aún no tiene un rango de dimensión establecida, ya que depende en general de la calidad de imagen de detalle que se quiera integrar al *software*.

Proporciones estimadas del espacio requerido para que una aplicación en tercera dimensión completa se ejecute correctamente es entre el 20% y 60% del total de espacio requerido de la instalación de la aplicación.

c. Memoria Virtual

La memoria virtual es un punto clave en el rendimiento de la interfaz en tercera dimensión, ya que independientemente de la plataforma, esta debe ser aplicada con todas las políticas de asignación y creación para darle a la interfaz vida propia, es decir, tener un bloque reservado específicamente para el procesamiento de datos de la interfaz.

En el caso de Microsoft Windows, el sistema operativo se encarga de definirla automáticamente de acuerdo a un análisis de características realizado durante la instalación, y en la familia de sistemas operativos UNIX, se utiliza la memoria SWAP, la cual como recomendación se establece como el doble de la memoria real (RAM).

3.1.4 Dispositivos de salida

Retroalimentan al usuario con la sensación de tercera dimensión, estos dispositivos vienen en dos clases:

Tabla IX. **Clases de dispositivos de salida**

Clase	Descripción
Monoscópico	Permiten que ambos ojos vean exactamente lo mismo que despliega el dispositivo.
Estereoscópico	Permiten a los ojos ver dos vistas separadas, dando la sensación de estereoscopia o profundidad, que permite la simulación de la tercera dimensión.

Durante este contexto se debe aclarar que la tercera dimensión puede alcanzarse con uno o ambas clases de dispositivos; sin embargo, en el siguiente capítulo se analizará el impacto de ambas tecnologías y cómo puede impactar el uso de las mismas en el ser humano.

Es importante recordar que existen dispositivos que son tanto de entrada como de salida, tal es el caso de los monitores tipo *Touch Screen*, que además de poder enviar las ordenes, despliegan el resultado de las mismas.

3.2 Tecnologías de *software*

Las tecnologías de *software* permiten la codificación y construcción del *software*, proveyendo las herramientas que facilitan la integración de componentes, objetos y codificación a través de los lenguajes de programación.

En una nueva tecnología de *software* como las interfaces en tercera dimensión, es muy común encontrar aún herramientas tipo beta o demo, que no proveen funcionalidades completas, porque aún se encuentran en construcción.

A continuación se detallarán algunas de las tecnologías capaces de desarrollar y ejecutar interfaces en tercera dimensión, en el orden siguiente.

- Lenguajes de programación
- Tecnologías en desarrollo 3D
- Herramientas de desarrollo

a. Lenguajes de programación

Los lenguajes de programación que se mencionan a continuación actualmente cuentan con la capacidad de integrar librerías y componentes que facilitan y permiten el desarrollo de interfaces en tercera dimensión:

- Java
- C++
- C#
- Perl
- Leguaje ensamblador cuando sea requerido.

Considerados como los lenguajes de alto desempeño y capacidad, permiten además de la integración gráfica, acoplarse a través de estándares de servicios de comunicación entre cualquier tecnología que maneje dichos estándares, o bien integrarlos a través de tecnologías de desarrollo.

En la actualidad existen varios componentes o bibliotecas de objetos gráficos que permiten la creación de los objetos en tercera dimensión, denominados tecnologías de desarrollo 3D.

b. Tecnologías de desarrollo

Dentro de las tecnologías que deben considerarse para la implementación de una interfaz, se recomienda aplicar el estándar abierto X3D, que define la interacción en tiempo real para tercera dimensión. Éste se encuentra definido dentro de los estándares de la norma ISO/IEC 19775/19776/19777 (<http://www.web3d.org>).

Existen además librerías y componentes que pueden ser utilizados para facilitar la creación de la interfaz en tercera dimensión:

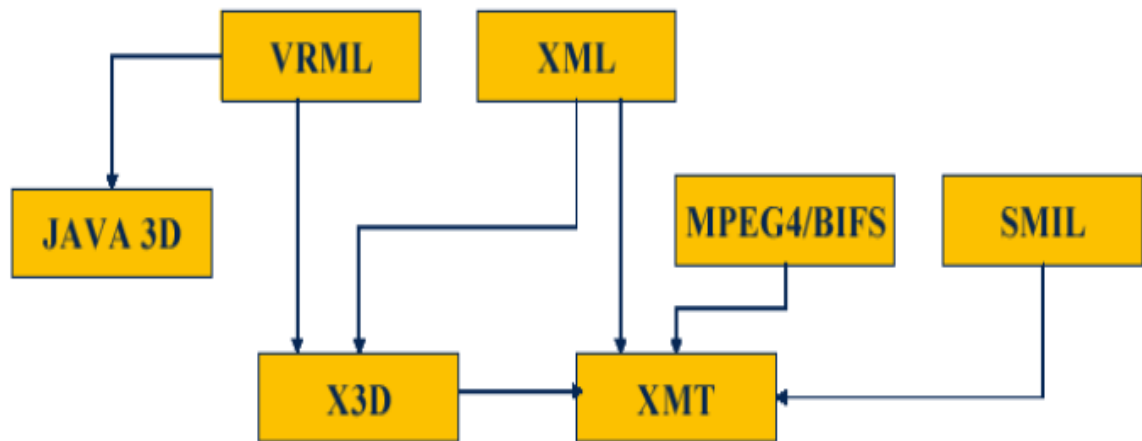
- OpenGL
- Java 3D API (<http://www.java3d.org>)
- Microsoft SDK DirectX 3D

Estos componentes utilizan algunos lenguajes de programación orientados a generar mundos virtuales, entre los cuales se puede destacar por su relación.

- VRML, trabaja bien como un formato de intercambio para modelos 3D en tiempo real y como herramienta de modelado.
- X3D, facilita a los diseñadores web la inclusión de imágenes 3D en sus páginas.
- Java3D, aporta un sistema gráfico de control de bajo nivel.

- MPEG-4, distribuye modelos 3D entre una gran variedad de plataformas (web, televisión, etc.).
- XMT, aporta una descripción textual para la representación de MPEG-4.

Figura 16. Relación de lenguajes de programación para 3DUI

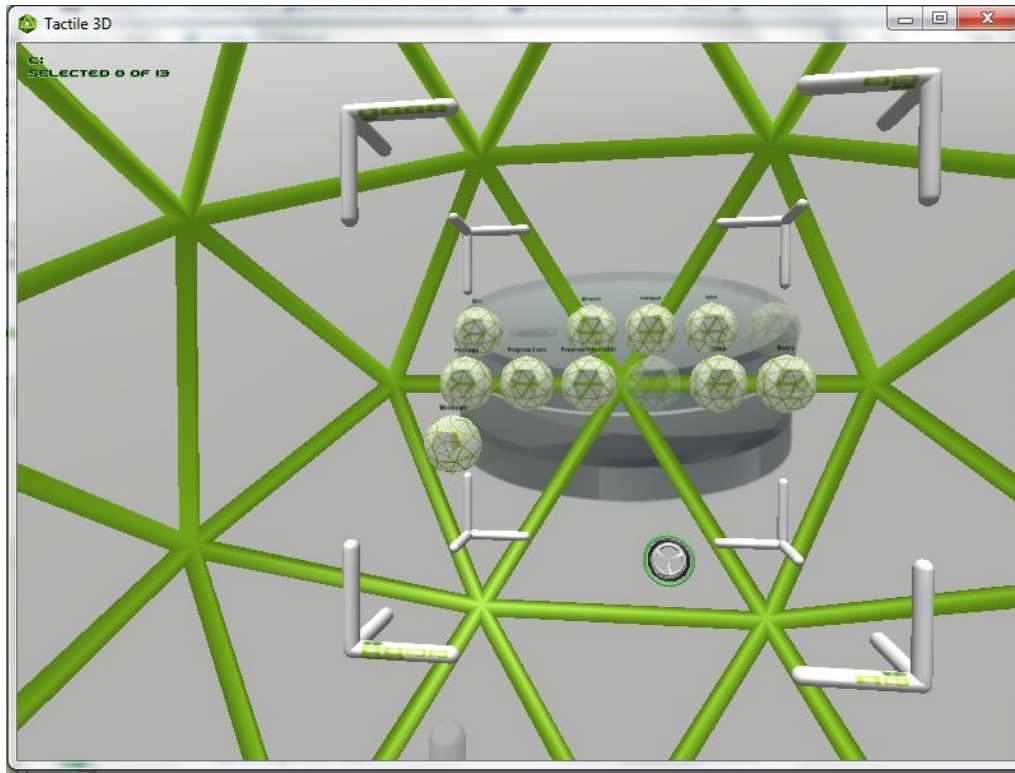


Fuente: http://www.newlaunches.com/entry_images/0510/21/psychedelic-3D-gloves.php

“Existen varias iniciativas que están en desarrollo, propuestas de interfaces con estas tecnologías en 3D, una de ellas es Upper Bounds Interactive Inc., que desarrolla Tactile 3D”⁴, la cual muestra un explorador de archivos desarrollado con OpenGL, en el cual se puede navegar simulando el espacio y con planetas cada unidad y/o directorio del sistema de archivos de una computadora.

⁴ Tactile 3D Interface. Upper Bounds Interactive Incorporated, [ref. 13 de septiembre de 2010]. Disponible en Web: <<http://www.tactile3d.com/>>

Figura 17. Ejemplo de funcionamiento de *Tactile 3D*



Fuente: Copyright © 2005-2010 Upper Bounds Interactive Inc. All Rights Reserved. Space Navigator is a Registered Trademark of 3Dconnexion, a Logitech Company.

c. Herramientas de desarrollo

Actualmente no existe un entorno de desarrollo diseñado para la creación de interfaces en 3D para *software*, básicamente la creación de los objetos pueden realizarse a través de diseño gráfico y programación en herramientas de desarrollo actuales, y para lograr el comportamiento deseado ir realizando los procesos conocidos como "*Debug*" o compilación, para observar el avance durante la codificación de la interfaz.

Algunas herramientas diseñadas para la construcción de juegos en tercera persona están incursionando en esta tecnología, agregando componentes que permiten la manejabilidad de objetos en diseño a través de árboles y canales, como Quest3D u Ogre 3D, estos permiten la ejecución de componentes diseñados gráficamente, creados con herramientas CAD, como AutoCAD, 3DMax, Maya, etc.; sin embargo, aún no se encuentran completamente orientadas a interfaces de usuario en tercera dimensión.

Quest3D, es una mezcla de un entorno de desarrollo y un motor de juegos, utilizado en su mayoría para la creación de realidad virtual, que posteriormente son integrados en aplicaciones interactivas en 3D. Dentro de sus características principales es que es un 90% visual, y por su diseño interno se puede desarrollar en tiempo real sin la necesidad de la compilación; sin embargo, su costo se encuentra aproximadamente por los € 10 000.00.

OGRE 3D es acrónimo de *Object-Oriented Graphics Rendering Engine*, es un motor de renderizado en tercera dimensión, basado en el lenguaje de programación C++.

Sus componentes o bibliotecas evitan la codificación a bajo nivel en *OpenGL* o *Direct3D*, proveyendo de una interfaz gráfica para este tipo de componentes. Es necesario mencionar que éste es un *software* libre, licenciado por el Instituto Tecnológico de Massachusetts -MIT-.

3.3 Requerimientos de funcionamiento interno

Consiste en adoptar objetivos y acciones para llevarlos a cabo y alcanzarlos (Autor-Informe, Año); así mismo, se debe tomar las decisiones apropiadas.

Un objeto bien diseñado es aquel cuyo funcionamiento es óptimo, dando lugar a que los procesos asignados a ese objeto en particular se lleven a cabo de forma rápida, utilizando la menor cantidad de recursos.

Un objeto en tercera dimensión debe capturar un porcentaje de los recursos disponibles para su creación y vida dentro de la interfaz; sin embargo, hay que destacar que la optimización para el funcionamiento de los objetos debe estar dada por el sistema operativo. A continuación se consideran varios requerimientos del funcionamiento interno:

a. Entorno gráfico

Es el elemento esencial de la interfaz de usuario donde se devuelve el funcionamiento completo, con la cual el usuario interactúa directamente para ejecutar las aplicaciones, ésta debe ser capaz de representar la realidad y la tercera dimensión, para que cumpla con el objetivo de desarrollar *software* con 3DUI y mejorar el rendimiento de los usuarios, con la comodidad de tratar con un mundo virtual para trabajar.

Para el desarrollo del entorno gráfico se requiere de un lenguaje de programación de alto nivel de los ya mencionados, para lograr capacidades de la tercera dimensión con la mayor parte de sus elementos, como se mencionó en el capítulo 2.

Considerar la creación de las siguientes capas para un funcionamiento correcto:

- Capa de monitor gráfico, para monitorear los procesos y recursos usados por el entorno gráfico.

- Capa renderizada, para transformar las imágenes según el detalle de personalización del usuario, en distintas resoluciones.
- Capa de comunicación, para el envío y recepción de mensajes para facilitar la detección de problemas.
- Capa de recuperación, permitir recuperar el entorno gráfico automáticamente, si la capa de monitor reportara algún problema.

b. Planificación de recursos

La planificación de recursos es el punto de inflexión en el desarrollo de las 3DUI, ya que de ésta depende el rendimiento que se va obtener del *software*, y el cual no debe ser mermado por la utilización de gráficas en tercera dimensión, procesamiento de datos, ejecución de aplicaciones, etc., tanto en paralelo como concurrentemente.

Para realizarlo adecuadamente, deben crearse las especificaciones técnicas mínimas para la ejecución de la interfaz y generar un plan de pruebas según la metodología de *software* establecida que permita evaluar a través de monitores de recursos los umbrales permitidos para su ejecución.

c. Manejo de procesos de sistema operativo para el funcionamiento

El funcionamiento de un sistema operativo se encuentra basado en proceso e hilos, los procesos deben mantenerse en ejecución para que éste funcione. Se recomienda la separación de los procesos que se utilizarán para entorno grafico de los utilizados para la ejecución de tareas alternas, esto se logra a través de la arquitectura con la cual es creado el *software*, y debería considerarse 2 premisas:

- Permitir la ejecución autónoma de procesos para la interfaz gráfica
- Permitir compartir recursos para procesos más complejos o voluminosos

d. Manejo de procesos de aplicaciones

Los procesos externos del sistema operativo o procesos de aplicación, son generalmente de un proceso por aplicación en ejecución. Es decir, si se está ejecutando Microsoft Word, en el monitor de procesos, únicamente se muestra un proceso en ejecución para esa aplicación. Considerando el ejemplo anterior, la idea esencial para manejar los procesos de los entornos de la interfaz en tercera dimensión, requiere crear un árbol de procesos, y permitir que un subproceso o proceso hijo realice la ejecución dentro de la interfaz en tercera dimensión.

e. Prevención de errores

Una prevención adecuada de errores debe planificarse de acuerdo al objetivo del *software*, durante el diseño, tomando en cuenta todas las acciones que el usuario puede generar, y los posibles errores más comunes, tomando en cuenta los siguientes criterios:

- El usuario es inexperto en el uso de computadoras
- El usuario es inexperto en el uso de *software* semejante
- El usuario no posee el conocimiento de las tareas que puede ejecutar con el *software*
- La aplicación debe ser robusta
- El sistema puede colapsar si no se realizan como deben ser las operaciones a ejecutar

- El sistema operativo no soporta la tecnología en tercera dimensión
- El *hardware* no soporta la tecnología en tercera dimensión

f. Reconocimiento de errores, diagnósticos y recuperación

El reconocimiento de errores estará a cargo de dos componentes importantes e independientes; cuando el error no sea un error de la interfaz y no constituya un error grave al sistema operativo, este puede ser controlado y recuperado por la capa de recuperación gráfica, de lo contrario éste será enviado a la capa de enlace contextual que lo enviará directamente al kernel del sistema operativo, y éste se encargue de diagnosticar si puede ser recuperado o definitivamente se provocó un error con el cual colapso el sistema y requiere de un reinicio completo.

Si el *software* desarrollado se distribuye en varios países, es recomendable contar con un servicio de internet, el cual pueda recopilar toda la información de los problemas de *software* y favorecer la prevención o corrección de los mismos. Esto contribuirá en generar una base de conocimiento que permite resolver de forma práctica errores.

g. Reconocimiento de audio

El reconocimiento de audio es una tecnología programada por *software*, la cual tiene como fin integrarse para aquellas personas que posean discapacidad como paraplejía, a través de un micrófono como dispositivo de entrada. En una interfaz en tercera dimensión, esta tecnología implica agregar una capa de instrucciones de voz muy específica, ya que a diferencia de las actuales el movimiento se realiza bidimensional y no tridimensional.

h. Optimización

Para lograr un funcionamiento correcto de una interfaz en tercera dimensión, hay que tomar en cuenta las políticas de optimización desde el momento es que se diseña.

Aquí se mencionan algunas políticas que deben mantenerse a nivel de sistema operativo y aplicaciones que no deben faltar en un *software* con interfaz en tercera dimensión que se esté construyendo, para un adecuado funcionamiento, según la siguiente clasificación:

- Sistemas operativos
 - Aplicaciones dirigidas o no Multiplataforma
 - Aplicaciones multiplataforma
-
- **Sistemas operativos**

Dentro de un sistema operativo para un funcionamiento eficiente existen políticas que debe mantener:

- Crear un “Garbage Collector” o gusano recolector, el cual, en informática, se encarga de darle mantenimiento a la información alojada tanto de la RAM como en la virtual.
- Al monitor de procesos, integrar recolector de procesos hijo sin funcionamiento que pueda detener o matar procesos que únicamente consuman recursos y el usuario no realice ninguna acción con ellos.

- **Aplicaciones dirigidas o no multiplataforma**

Una aplicación, en general, desde su desarrollo debe cumplir con varios requerimientos, los cuales se describen a continuación:

- No solicitar el uso de recursos apropiativos si no son necesarios.
- Crear un “Garbage Collector” o gusano recolector, que se encargue de darle mantenimiento a la memoria utilizada por la aplicación.
- No incluir objetos en la interfaz que no se utilicen dentro de la aplicación.
- No utilizar más del 40% de los recursos gráficos del equipo de cómputo, a menos que sean estrictamente necesario.

- **Aplicación multiplataforma**

Existen algunos programas que se ejecutan sobre diferentes arquitecturas de sistemas operativos, en el caso de la interfaces en tercera dimensión, el proyecto *Looking Glass* es el único proyecto multiplataforma hasta ahora conocido. Los factores de una aplicación multiplataforma más importantes para evaluar son:

- Portabilidad
- Transparencia
- Carga de recursos

La portabilidad para la generación de interfaces en 3D está muy restringida al lenguaje de desarrollo, en la actualidad el lenguaje multiplataforma más famoso por todas ventajas que presenta es Java.

Como se menciona anteriormente se tiene conocimiento de solo una aplicación con entorno de interfaz 3D multiplataforma. Existen librerías OpenGL que se han desarrollado específicamente para ser multiplataforma y en algunos casos han sido integradas a distintos lenguajes de alto nivel, tal es el caso de Java, C, C++, C#.

Sin embargo, las herramientas de desarrollo para aplicar el uso multiplataforma y de interfaz en 3D aún no se encuentran creadas.

La transparencia de una aplicación implica un punto importante a mencionar porque al tener una interfaz completa en tercera dimensión, en el caso de los sistemas operativos actuales, es tomar los entornos gráficos actuales de trabajo y proveerles la interacción en 3D completa, para que las herramientas se desenvuelvan de la misma forma.

Considerar que la creación de una interfaz en tercera dimensión, si ésta no se crea con los conocimientos adecuados de multiprocesamiento que se mencionó, el consumo de recursos puede ver afectado significativamente.

4 IMPACTO TECNOLÓGICO DEL DESARROLLO DE SOFTWARE CON INTERFAZ EN 3D

Como toda nueva tecnología las interfaces en tercera dimensión, deben pasar por varias etapas de crecimiento, que en tecnología duran varios años hasta llegar a un punto de madurez, en el cual se encontrará lista para ser utilizada por empresas y personas que desarrollan *software* para su distribución mundial.

Este trabajo de investigación es parte de este crecimiento, en la cual se inicia la construcción formal de los lineamientos, tanto en metodología de desarrollo como técnicos, para realizar una implementación de *software* con interfaces en tercera dimensión. Por ello, para describir el impacto tecnológico de las interfaces en tercera dimensión, este capítulo se enfocará en responder las siguientes interrogantes:

- ¿Qué hace posible el desarrollo de interfaces en 3D?
- ¿Qué tecnologías quedarán obsoletas?
- ¿Qué problemas se minimizan o resuelven?
- ¿Qué ventajas posee el *software* con interfaz en 3D?
- ¿Qué obstáculos posee frente a otras tecnologías?
- ¿Cuál es el futuro de las interfaces en 3D?

Estas permitirán ampliar la visión en la aplicación de las interfaces, tomando en cuenta factores económicos, sociales, humanos, etc., que permitan entender el impacto tecnológico.

4.1 ¿Qué hace posible el desarrollo de interfaces en 3D?

A partir de la década de los 90, los lenguajes de programación se convirtieron en una de las tecnologías que más mejoras presentaban, se integraban instrucciones de alto nivel, nuevos paradigmas de programación, y se creaban IDE's (Entornos de Programación Integrados –*Integrated Development Enviroment*-), inteligentes y ergonómicos, que facilitaban la codificación durante el desarrollo del *software*, esto permitió que los lenguajes de programación tuvieran un ritmo de evolución acelerado. Sin embargo, para que este crecimiento fuera sostenible y en aumento hasta la actualidad, se necesitó del crecimiento descontrolado del internet. Éste se convirtió en la herramienta de colaboración, comunicación y de aprendizaje más grande del mundo, permitiendo que la tecnología sufra mejoras aceleradamente.

Con las mejoras tecnológicas en los lenguajes de programación y *hardware*, se iniciaron las creaciones de las fantasías del mundo ideal del hombre, en la cual las personas crean su mundo ideal y lo hacían realidad a través de gráficos diseñados por computadora, permitiendo crear escenarios en tercera dimensión. La mayoría de estos entornos fueron posibles gracias al trabajo de diseñadores gráficos e ingenieros, que crearon las herramientas necesarias tanto en *software* como en *hardware*.

Dentro de las herramientas más importantes, se encuentra el diseño de escáner de tercera dimensión, el cual es una herramienta que permite obtener modelos tridimensionales por medio de laser que analizan la forma de los objetos en formato digital y que en sus modelos más avanzados, inclusive, puede obtener información del color y superficie de los objetos.

Con este paso se dio origen a la realidad virtual en tercera dimensión, y se abrió el campo para el desarrollo de juegos con escenarios en tercera dimensión y en la cinematografía a una cantidad infinita de efectos especiales, incluyendo la creación de películas completamente en tercera dimensión. Permitiendo que el mercado de empresas se interesaran en la inversión de las tecnologías en tercera dimensión, el avance en diseño gráfico y modelos de construcción creados en herramientas especializadas fue creciendo al punto que actualmente, incluso en Guatemala, existen estudios especializados para la edición de video y gráficos en tercera dimensión, que participan en la creación de películas.

Este auge permitió que las tecnologías de *software* para interfaces de usuario en tercera dimensión fuera creciendo, al grado que en el año 2006, se creara el primer simposio exclusivamente sobre las interfaces de usuario en 3D, organizado por la IEEE. A la fecha el simposio continúa realizándose anualmente, en él se presentan proyectos, investigaciones y se evalúan prototipos y documentos a través de concursos para incentivar la continuidad en el crecimiento de ésta tecnología.

4.2 ¿Qué tecnologías quedarán obsoletas?

El objetivo principal de reemplazar o mejorar la tecnología actual, es facilitar las tareas que se realizan y permitir que más personas tengan acceso a ella. Las interfaces de usuario de 2 dimensiones o GUI, son utilizadas actualmente por la mayoría de usuarios que utilizan una computadora para realizar sus tareas diarias, y las CUI o interfaces de una dimensión, son preferidas por usuarios expertos, en su mayoría personas que tienen un grado alto de conocimiento informático.

Las interfaces en tercera dimensión están destinadas a reemplazar a las GUI; sin embargo, no en un 100% ya que no todos los usuarios poseen las mismas preferencias y conocimientos. Durante la etapa de transición entre ambas tecnologías, básicamente será una necesidad conservar ambas, mientras la mayoría de los usuarios se acoplan al cambio, debido a la resistencia al cambio que pueda existir.

Un ejemplo interesante de esta práctica es el caso de Microsoft con sus sistemas operativos Windows Vista y 7, los cuales poseen por defecto la tecnología Aero, que permite una experiencia visual más completa y elegante de su sistema operativo y el despliegue clásico de ventanas que tiende a verse sencillo.

Esta transición permite continuar con el crecimiento de las interfaces en tercera dimensión y en la cual, no se debe olvidar considerar los requerimientos de *hardware* que requiere para funcionar; debido a que la mayoría del utilizado para la ejecución de *software* en dos dimensiones, no posee las características establecidas en el capítulo 3, de esta investigación, para tener un funcionamiento adecuado de esta nueva tecnología.

Dispositivos que no sean capaces de funcionar dentro de una interfaz en 3D quedarían prácticamente como pérdidas para las empresas o lugares de venta, porque dejarían de ser rentables, podrían venderse a precios muy bajos considerando que pueda dárseles otra utilidad o bien donarse para que tenga una vida útil apropiada.

Dentro de las implicaciones de esta mejora tecnológica, se consideran los costos en *hardware* que el usuario deberá sufragar para tener esta tecnología, es preciso tomar en cuenta que no todo el *hardware* su vuelve inservible, según la configuración cada equipo de cómputo puede estar en capacidad de actualizarse con los dispositivos adecuados que le permitirán ejecutar correctamente las interfaces en tercera dimensión, sin necesidad de comprar equipos de cómputo nuevos.

Al igual que el *hardware*, el *software* desarrollado podría seguirse ejecutando mientras los sistemas operativos sean diseñados para ejecutar *software* con interfaces 3D, GUI y CUI. Sin embargo, la mayoría del *software* requeriría actualizaciones o parches para que funcionen correctamente y no todas las empresas los crearán, por ello, aquellas que decidan no continuar con las actualizaciones para su *software*, quedarán obsoletas.

Al igual que todo cambio, éste requiere actualizaciones en todo sentido, y es indispensable mencionar que existirá una gran cantidad de información técnica, práctica y de capacitaciones, tanto para *hardware* y *software* que dejará de utilizarse, para dar paso a una nueva generación de información.

4.3 ¿Qué problemas se minimizan o resuelven?

Las interfaces en tercera dimensión, como se ha mencionado en ésta investigación, deben poseer todos los requerimientos internos y externos para que cumplan con su principal objetivo, “ser una semejanza de la realidad e incentivar a los usuarios a utilizar la tecnología”.

Para delimitar de forma práctica los problemas que las interfaces en tercera dimensión ayudarán a resolver o minimizar, se tratarán bajo los siguientes dos puntos de vista:

- Perspectiva para el usuario
- Perspectiva para la sociedad

Ambas darán un escenario, considerando la interacción que tendrán con el *software* de forma individual y el que tendrán de grupal o social.

a. Perspectiva para el usuario

Una gran condicionante para personas que no manejan un equipo de cómputo, por falta de conocimiento o miedo a la tecnología, puede reducirse drásticamente, ya que una interfaz en tercera dimensión posibilita el aumento de la comodidad y adaptabilidad al uso de la computadora por el ambiente gráfico que desplegará, permitiendo al usuario manejar intuitivamente o inconscientemente el *software*.

La consciencia del ser humano se despierta cuando la realidad se hace presente en la tecnología, ya que la misma lógica que se utiliza para pintar un cuadro con un pincel, pintura y un lienzo en la realidad, puede reflejarse en una interfaz en tercera dimensión, permitiendo al usuario pintar un cuadro en su lienzo blanco, con uno o varios tipos de pinceles y todos los colores de pintura que la combinación de colores que desee, simplemente enseñándole a perder el miedo a utilizar los dispositivos adecuadamente. Aunque no se pretende reemplazar el hecho de pintar un cuadro de forma clásica.

Bajo este punto de vista los usuarios podrán minimizar:

- Resistencia al uso de la tecnología a través de equipos de cómputo
- Complejidad para aquellos que posean alguna discapacidad
- Dificultad para recordar cómo realizar una tarea específica dentro del software
- Dificultad para aprender rápidamente a través de la conciencia en el uso de objetos reales

b. Perspectiva para la sociedad

Como toda nueva tecnología que se espera tenga un impacto en todo el globo, por ello es lógico que deba analizarse también a nivel social. Sin embargo, cada sociedad distribuye de modo diferente la ejecución de sus funciones y la utilidad de sus beneficios, convirtiéndose en un tema de investigación muy extenso, al cual tratará de aclararse de forma resumida con un ejemplo real como *Facebook*.

Facebook es una herramienta social, la cual ha modificado y reemplazado algunas funciones humanas, que se han generalizado y modificado las relaciones humanas, generando nuevos escenarios sociales.

Existe un impacto cultural de una tecnología directamente proporcional a su éxito, integrando con ella un sistema socio-técnico inseparable, que condicionan su forma de organización, así como la visión macro de una cultura condiciona las tecnologías que está dispuesta a utilizar.

Las condiciones que se esperan minimizar con esta tecnología son:

- Que los grupos sociales permitan aprender a usar la tecnología a un ritmo más balanceado, sin que las condiciones económicas sean tan distantes por el realismo que brinda la tecnología.
- Con una tecnología madura los costos se estabilizarán y podrán estandarizarse al grado, que le permitan a la mayoría de las sociedades tener el alcance de ésta tecnología.

4.4 ¿Qué ventajas posee el *software* con interfaz en 3D?

Algunos expertos en tecnología de la información coinciden en que la necesidad de las nuevas tecnologías se expliquen mejor a las empresas y personas individuales, debido a que el problema con la introducción de éstas no es cuestión de costos o infraestructura directamente, sino de la falta de conocimiento de sus ventajas o beneficios a corto, mediano y largo plazo.

Este es un punto controversial que en realidad desde la perspectiva en que se vea tiene mucho que evaluar, considerando que no todas las empresas o personas pueden darse el lujo de aventurarse en inversiones si no tienen los recursos para hacerlo. Sin embargo, a continuación se muestran las ventajas que posee esta tecnología, considerando algunos estudios especializados para este tema, considerando las ventajas según la memoria espacial del hombre, su curva de aprendizaje, y cómo una ventaja competitiva de marketing para las empresas.

Según el funcionamiento de la memoria espacial del hombre

La utilización de la tercera dimensión en el *software*, permite que el hombre ponga en práctica su memoria espacial, permitiendo que toda persona pueda discernir en ambientes físicos en cómo dirigirse o movilizarse.

Un ejemplo práctico es considerar a una persona que en su niñez conoció una ciudad y luego de varios años vuelve a visitarla, su memoria espacial le indica como movilizarse de nuevo dentro de la ciudad, porque la persona almacenó dicha información en su primera visita en su memoria espacial.

“Un experimento realizado por Tavanti y Lind, divulgados en InfoVis 2001, proporciona un resultado completo en favor de la tercera dimensión, en el cual sus participantes recordaron la localización de las letras del alfabeto con más eficacia al usar un interfaz en 3D que al usar una de dos dimensiones”⁵. El experimento publicado se basa en un modelo perceptivo de las dimensiones con la modificación de controlar algunos factores previamente incontrolados. Los resultados mostraron que los usuarios mejoran su eficiencia claramente y presentan una mayor facilidad en la interacción con la interfaz.

Otras investigaciones en sistemas recientes han proporcionado las visualizaciones tridimensionales interactivas para apoyar el trabajo diario como los documentos de administración de sistemas. ¿Pero qué mejoras presentan las interfaces 3D sobre tradicionales de dos dimensiones? Andy Cockburn and Bruce McKenzie, presentaron uno de los primeros análisis sobre las mejoras, el

⁵ 2D vs 3D, Implications on Spatial Memory. Proceedings of IEEE InfoVis 2001. Tavanti, M. and M. Lind. Symposium on Information Visualization, San Diego, 2001 [ref. 20 de noviembre de 2010]. Disponible en Web: < <http://portal.acm.org/citation.cfm?id=976314>>

cual mostró una gran preferencia por la interacción con una tercera dimensión. Concluyendo con las siguientes ventajas:

- El porcentaje de finalización de una tarea genérica, se completó en 4,8 segundos más rápido en una interfaz de tercera dimensión.
- La organización de las ventanas en forma de perspectiva de papel, proporcionó una diferencia de 6 ventanas, más y mejor acopladas.
- La memoria espacial en un ambiente en 3D, provocó que los usuarios localizaran con más facilidad los objetos con los cuales podían realizar sus tareas.

Según la tendencia de la curva de aprendizaje del hombre

En el estudio de mejoras mencionado anteriormente, las interfaces en tercera dimensión demostraron que la práctica es la forma de aprendizaje del hombre que facilitará su uso. Debido a que los usuarios que no experimentaron la tercera dimensión, es decir, que utilizaron una interfaz tradicional de dos dimensiones, ya están acostumbrados a este tipo de interfaz y presentaron un mejor rendimiento en la ubicación de los objetos.

Determinar una referencia del nivel de curva de aprendizaje bajo práctica para interfaces en tres dimensiones, requerirá para cada *software* considerar que el aprendizaje está compuesto en 4 fases, la incompetencia inconsciente que es cuando no se conoce cómo se realiza alguna tarea, la incompetencia consciente cuando se toma conciencia que no se conocía como realizar una tarea, la competencia consciente se da al entender cómo realizar una tarea, pero aún se experimenta y la competencia inconsciente, en la cual ya es tan práctico realizar la tarea que ya no es necesaria la concentración.

Para realizar un estudio basado en fases y realizar las mediciones para el avance de cada una de las fases.

Según las ventajas competitivas del mercado

Cuando deja de ser indispensable el costo o la funcionalidad técnica de un producto tecnológico por cumplir con esas expectativas, se consideran otros factores que permiten seleccionarlo entre los demás. Un ejemplo sencillo de este caso es al seleccionar ropa, la belleza puede sobreponerse sobre las consideraciones prácticas, ya que compramos ropa bonita aunque sepamos que sus detalles de confección o acabados no son óptimos, o que tendrá poca duración debido a los materiales utilizados.

Cuando la función principal de los objetos tecnológicos se convierte en una representación de la persona, al igual que la analogía de la ropa, se inicia la satisfacción para poder tener la tecnología de punta y de moda, para establecer un estatus social, permitiendo obtener grandes ganancias a las empresas u organizaciones que vendan, en este caso de investigación, el *software* con interfaces en tercera dimensión. Obteniendo en el mercado clientes satisfechos, las marcas generan su propia estabilidad, cuando inicia la depreciación de precios y por ende la globalización de los productos.

4.5 ¿Qué obstáculos posee frente a otras tecnologías?

En la actualidad los principales obstáculos para la implementación del *software* con interfaces en tercera dimensión se encuentran en:

- Las tecnologías que le dan vida a las interfaces en tercera dimensión, ya que aún están en construcción;

- La complejidad para determinar el tiempo de la creación del *software* desde su concepción hasta su implementación;
- El tiempo requerido para que los componentes de desarrollo de las interfaces se encuentren ya maduros.

Es importante delimitar cómo estos obstáculos impactan en las etapas de un proyecto de desarrollo de *software*, considerando los riesgos en los que se incurre y sus costos, son directamente proporcionales a la inversión o capital del proyecto cuando se inicia, como se describe a continuación.

a. Análisis y diseño

El tiempo de creación del *software* con interfaz 3D en comparación con el actual en dos dimensiones, no posee un punto de comparación, por lo que se dificulta más la generación de una planificación aproximada en el desarrollo, al igual que toda nueva tecnología estará guiada por la experiencia y la capacidad administrativa para guiar el proyecto.

El diseño del *software*, como cualquier producto, depende del tamaño y complejidad de los requerimientos del mismo. Para las interfaces en tercera dimensión debe considerarse agregar a la metodología de *software* a utilizar las siguientes etapas:

- Concretizar el objetivo del software de forma adecuada para la aplicación de los requerimientos de comportamiento;
- Aplicación de las metáforas necesarias para cada ambiente o ambientes que vayan a integrar la aplicación;
- Evaluación y manejo de metáforas.

Las etapas anteriores son panificables; sin embargo, poseen una alta probabilidad de cambio, por lo que se sugiere que durante las primeras 3 revisiones sea considerada la re-planificación, sin perder de vista que de no aplicarse las metáforas correctamente, la ventajas de la tercera dimensión se volverán desventajas y la construcción sería un producto fallido.

b. Construcción

Con el diseño listo, comienza el trabajo de desarrollo, donde se plasmará la tercera dimensión y la funcionalidad de la interfaz. Es muy aventurado indicar que no se presentarán problemas técnicos y deben considerarse las holguras necesarias para salir adelante. Es importante tener claridad que la programación del *software* se fundamentará en la capacidad de los programadores considerando su experiencia, el lenguaje de programación y cómo, lo más elemental, el conocimiento que se tenga para diseñar la tercera dimensión.

c. Evaluación de costos

Cuando se habla de costos hay que tomar en cuenta las variables más esenciales en lo que ha desarrollo de *software* se refiere. Por ello a continuación se mencionan los costos que deben considerar como mínimo, debido que al ser una nueva tecnología se pueden incurrir en pérdidas considerables.

- Costo de aprendizaje, el cual cubre todo lo relacionado con capacitaciones, cursos o seminarios, en los cuales se estudie cómo utilizar las herramientas seleccionadas para el desarrollo y para el diseño espacial o de escenarios en tercera dimensión, en todos los miembros

del proyecto de desarrollo para que puedan crear y obtener los beneficios de la tercera dimensión.

- Costo de hardware, enmarca todo el hardware necesario para la construcción del software, actualmente y considerando la cantidad de personas para un proyecto, tiende a ser costoso como toda nueva tecnología.
- Costo de software, considera todo el licenciamiento, mantenimiento y soporte necesario de lenguajes de programación y herramientas especializadas, para diseño de las interfaces en tercera dimensión, vistas en el capítulo 3.
- Costo del recurso humano, refiriéndose directamente al salario del personal que participa en el proyecto de desarrollo del software en su totalidad.
- Costo de sostenimiento de servicios, son los costos relacionados con el lugar de desarrollo (alquiler si es necesario), teléfono, luz, agua, internet y todo lo necesario, para mantener el desarrollo durante el tiempo planificado.
- Costo para contingencias, un costo al que nadie quisiera llegar, pero que es provocado por problemas que se presentan de improviso y para los cuales es necesario estar preparado.

4.6 ¿Cuál es el futuro de las interfaces en 3D?

En la actualidad con la colaboración de miles de personas, empresas y organizaciones, se creó y desarrolló el concepto de tecnologías de la información, éste permite abstraer muchas áreas de tecnología y de los sistemas de información, que han permitido crear metodologías, estándares, recomendaciones de mejores prácticas, y lineamientos para la implementación de la tecnología.

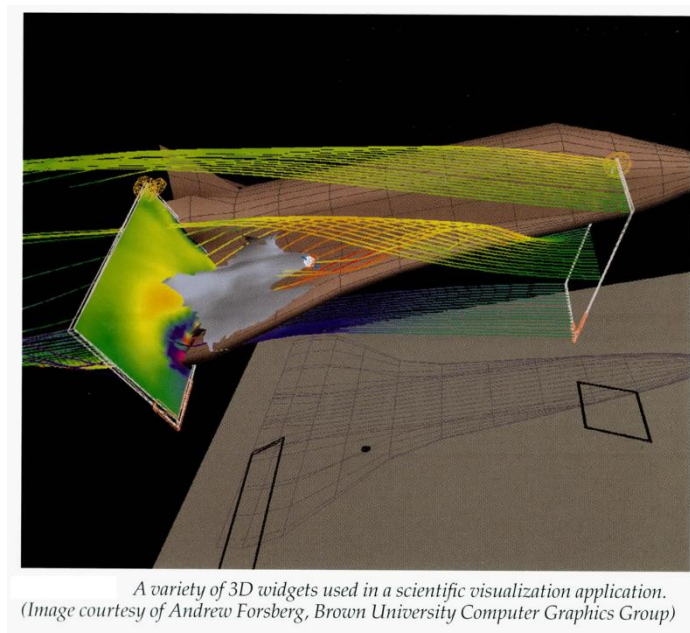
Las interfaces en tercera dimensión deberán integrarse a las tecnologías de la información, para que puedan mantener un orden y garantizar su correcto funcionamiento y aplicación. Según algunos investigadores, incluyéndome, ésta tecnología necesita al menos 3 años más para madurar, y al menos 5 años más para utilizarse a nivel mundial.

Una de sus grandes ventajas es que durante este tiempo de crecimiento ésta puede tener varias áreas de aplicación que se verían beneficiadas, tal es el caso de:

- Educación: para crear programas diseñados a la interacción que muestre el funcionamiento real de las cosas, a través de una interacción real.
- Diseño y modelado arquitectónico con interacción simple: para facilitar el diseño de modelos del mundo real, sobre un contexto real de tercera dimensión.
- Comercio electrónico: creando tiendas interactivas con productos reales para comprarse por internet o inclusive automatizar auto servicios.
- Turismo: creando servicios interactivos para turistas desde la reservación de viajes, hospedajes, programación de lugares, etc.
- Sistemas de control o administración: se aplicará un cambio en la forma de operar las instrucciones en estos sistemas, convirtiéndolas de instrucciones generadas por clics, en botones, a interacciones gráficas con los objetos, permitiendo eliminar la barrera que hay entre los idiomas.

- Tratamiento psiquiátrico: creando escenarios reales para personas que poseen fuertes miedos hacia cierto tipo de objetos o situaciones, para su tratamiento son generalmente sometidos a “terapias de exposición”, donde gradualmente se les van presentando en niveles para ir generando los estímulos que les permitan perder el miedo. Estos tratamientos pueden realizarse a través de interfaces de tercera dimensión.
- Observaciones científicas: ejecución, experimentos y simulaciones que producen una gran cantidad de datos, que luego pueden ser observados y analizados, para su mejor comprensión, o bien utilizados para generar diversos puntos de visualización.

Figura 18. **Ejemplo de aplicación científica**



Fuente: Doug A. Browman, Ernest Kruijff, Joseph J. LaViola, Jr. Ivan Poupyrev, 3D User Interfaces: Theory and Practice, Pág. 264

- Historia: capacidad para generar fuentes de historia en 3D, para civilizaciones antiguas o actuales, a través de un conocimiento rico en audio, texto e imagen.

El hombre por naturaleza ha tratado de facilitar sus tareas, y crear inventos tecnológicos, que mejoren su forma de vida automatizando tareas complejas. Con el tiempo se realizarán adelantos que permitan al *software*, con interfaz en tercera dimensión, construir los efectos especiales creados por computadora que han mostrado la interacción de personas con este sistema, a través de funciones prediseñadas, que simulan la comunicación, manejo y despliegue de información en tercera dimensión, inclusive teniendo interacción con hologramas que reconocen la acción que se ejerce sobre los objetos.

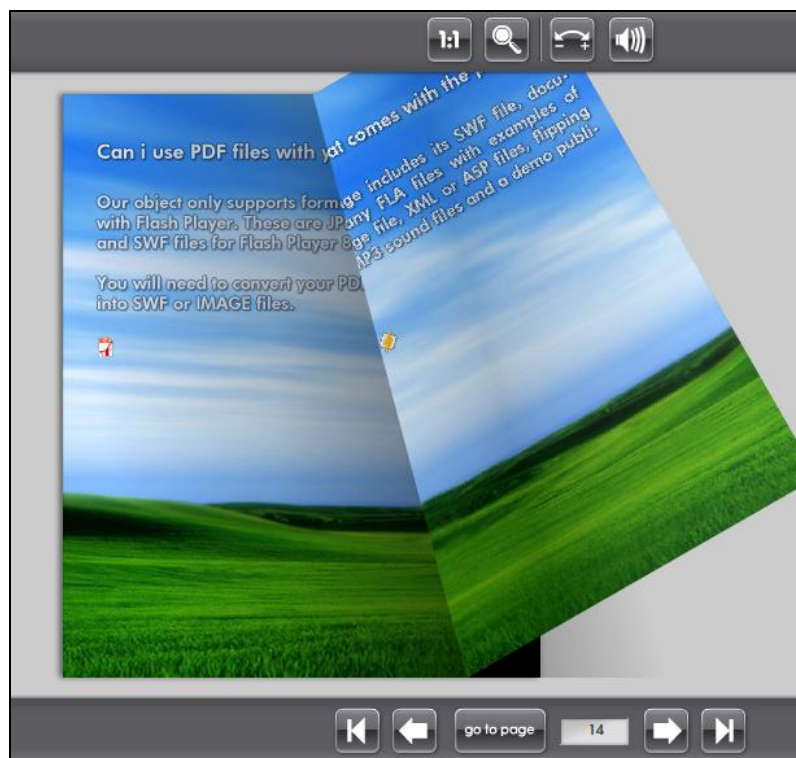
En el caso de la historia y turismo, se puede ver su aplicación con los efectos especiales de interacción en tercera dimensión en la película llamada “Máquina del Tiempo”, en la cual, en una de sus escenas muestra a una persona en versión holograma, desplazándose para explicar la historia del mundo con interacción directa.

Sin embargo, el hecho más importante es que el hombre siempre ha tratado de volver realidad sus sueños e imaginación, con la tecnología actual ya se inicia este proceso y en el ámbito de innovación tecnológica, este tema dará mucho de qué hablar, una vez que las herramientas de construcción o desarrollo estén más maduras, permitirán crear una ilimitada gama de aplicaciones.

Como aporte durante de esta investigación, se encontró que esta tecnología llama mucho la atención de las personas, solamente con el hecho de interactuar con objetos reales desarrollados por computadora, con las misma esencia de funcionamiento que les caracteriza.

Esta conclusión se obtuvo de la presentación del efecto Libro, a cuatro de cinco usuarios, que interactuaron con esta sencilla aplicación que permite la lectura de un libro con movimientos de página reales, opinando que era muy atractiva y práctica de utilizar. Por ello, se propone como la siguiente etapa del diseño de *software*.

Figura 19. **Ejemplo simple del efecto libro**



Fuente: imagen de FlashPageFlip

Esta atención se ve reflejada también con la aparición del 3D en televisiones, reproductor de video y cámaras digitales, que tienen la posibilidad de generar solo el efecto visual, siendo necesario el uso de gafas en tercera dimensión, tal y como podemos verlo en el cine 3D.

El mercado que recientemente inició a utilizar ésta tecnología y que está teniendo una rápida y buena aceptación, es en aplicaciones de teléfonos celulares, los cuales le dan al usuario atracción visual e interacción con elementos de tercera dimensión, sencillos y de cómodo manejo, que han generado gran gusto por la adquisición de éstos aparatos. Por lo que se espera que este mismo fenómeno se presente una vez que estén en el mercado las aplicaciones con interfaces en tercera dimensión para equipos de cómputo.

Los sistemas operativos más comerciales funcionarán completamente bajo una interfaz de tercera dimensión, permitiendo la expansión de éstos a dispositivos que utilicen cualquier equipo digital con despliegue información visual, incluyendo electrodomésticos, equipo de trabajo especializado, etc.

En el caso de los electrodomésticos, se tienen identificadas dos propuestas de implantación de las interfaces en tercera dimensión, una para el uso interno, es decir la interacción directa con el aparato y otra a través de *software* con aplicación domótica para su control.

La domótica en este punto se define como el conjunto de sistemas que permiten la automatización de un ambiente cerrado que comúnmente es una vivienda, para ella se amplía este campo ya que el control total del ambiente se podría reproducir directamente a una interfaz en tercera dimensión propia de una aplicación.

Inclusive se podrá contar con esta misma tecnología dentro de las páginas web, para adquirir o buscar información, con una versión mejorada del lenguaje de modelado para realidad virtual, adhiriendo una mejora más a la herramienta de colaboración más grande del mundo.

Además, hay que decir que las interfaces en tercera dimensión son un paso para llegar a una simulación completa de la realidad, ya que también se están creando aparatos capaces de reproducir olores según la interacción que se vaya generando con objetos o ambientes dentro de una interfaz de usuario.

Todos estos adelantos provocarán un impacto socio-económico a nivel mundial, se beneficiarán las empresas que logren ganar la carrera de la innovación de un producto final completo, aquellos que tengan el poder adquisitivo para nuevas tecnologías y paulatinamente se globalizará, para convertirse en la tecnología de uso cotidiano, a la espera de ser actualizada.

5 LINEAMIENTOS DE EVALUACIÓN PARA APLICACIÓN DE MÉTRICAS PARA EL ANÁLISIS Y DESARROLLO DE APLICACIONES CON TERCERA DIMENSIÓN

Las interfaces en tercera dimensión deben poseer varios requerimientos de comportamiento y de funcionamiento interno como se trató en capítulos anteriores, así como acoplarse a metodologías de desarrollo existentes.

Las metodologías de desarrollo de *software* más utilizadas tienen como común denominador la elaboración de ciclos, etapas, fases o iteraciones de revisión tanto a nivel de análisis y diseño, como del proceso de desarrollar el *software*. En cada uno se generan versiones del diseño o prototipos del producto, que permiten documentar el proceso y asegurar que el desarrollo se lleve a cabo con altos estándares de calidad, cumpliendo con los requerimientos funcionales y no funcionales del *software*.

Para el *software* con interfaz en tercera dimensión, se recomienda a la metodología seleccionada agregar dos conjuntos de lineamientos para aplicación de métricas, un conjunto práctico para revisión de los requerimientos que pertenecen esencialmente a las interfaces en tercera dimensión, y otro que apoya la evaluación del funcionamiento de los prototipos. En este ámbito métrica se define como criterios o variables que son medibles de forma objetiva.

En este capítulo se detallarán ambos conjuntos. Además, es preciso indicar que en ambos no se hará mención sobre el uso de estándares de codificación, ya que éstos deberían existir por defecto dentro de la metodología seleccionada, según se defina durante todo el ciclo de vida del *software*.

5.1 Lineamientos para requerimientos de interfaces en 3D

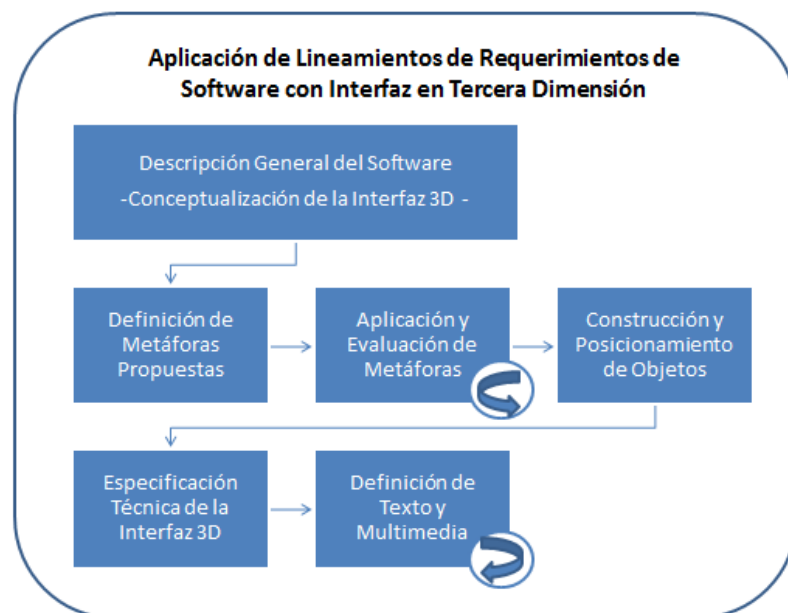
Este conjunto se basa en acciones sencillas, las cuales permiten mantener alineados los requerimientos de *software* con interfaces en tercera dimensión.

Su objetivo principal es mantener delimitado el diseño del *software* con interfaz en tercera dimensión, a través de instrumentos que además puedan ser utilizados en las etapas de desarrollo y evaluación del *software*. Es necesario aclarar las siguientes consideraciones.

- Este grupo o conjunto no reemplaza a ningún instrumento de la metodología seleccionada;
- Debe crearse durante la etapa de análisis y diseño, y se recomienda que sea validado por todos los equipos participantes en el proyecto de *software*;
- Este conjunto es una propuesta, la cual puede adherirse a cualquier metodología y estar sujeta a cambios para mejorar su objetivo. Es recomendable adaptarla según la madurez de la aplicación de la tecnología;
- Aplicado el conjunto de lineamientos correctamente, permitirá al equipo de desarrollo tener el escenario completo del proyecto, permitiendo dar continuidad durante toda esta etapa y minimizar fallas por interacciones y objetos mal contruidos.

Se deben cumplir los siguientes 6 pasos que recopilarán información básica para la creación de una 3DUI y generarán la documentación necesaria para su desarrollo, para ello a continuación se muestra el diagrama del modelo de requerimientos para interfaz en 3D y posteriormente una breve descripción para cada paso.

Figura 20. **Aplicación de lineamientos de requerimientos de *software* con interfaz en 3D**



a) Elaboración de una descripción general del *software*, que permitirá resumir la información que se utilizará para la creación de la interfaz en 3D, la cual debe contener:

- Objetivo general
- Objetivos específicos
- Población objetivo del *software*

- Funciones generales a ejecutar dentro del *software*
- Especificación de requerimientos funcionales
- Especificación de requerimientos no funcionales
- Descripción visual de la aplicación

Figura 21. **Estructura de la descripción general del *software* con interfaz en 3D**

DESCRIPCIÓN GENERAL DEL SOFTWARE CON INTERFAZ EN 3D

Objetivo General:
<Objetivo General>

Objetivos Específicos:

- Obj. 1
- Obj. 2

Población Objetivo:
<Población Objetivo del Software>

Requerimientos Funcionales:

1. Func. 1
2. Func. 2

Requerimientos No Funcionales:

1. No. Func. 1
2. No. Func. 2

Descripción Visual del Software:
<Descripción Visual del Software>

A continuación, se muestra un ejemplo práctico para una aplicación de lectura de libros electrónicos, que durante la explicación se irá explicando:

Figura 22. Ejemplo de descripción general del software con interfaz en 3D

DESCRIPCIÓN GENERAL DEL SOFTWARE CON INTERFAZ EN 3D
<i>Objetivo General:</i> Permitir la lectura de un libro electrónico.
<i>Objetivos Específicos:</i> <ul style="list-style-type: none">• Brindar al usuario una herramienta con la accesibilidad necesaria para la lectura.
<i>Población Objetivo:</i> Diseñado para cualquier usuario que requiera leer un libro electrónico.
<i>Requerimientos Funcionales:</i> <ol style="list-style-type: none">1. Tener la capacidad de abrir libros en formato PDF, como documento predeterminado de lectura.2. Utilizar un separador de página para almacenar la última página leída, para continuar su lectura más adelante.3. Poseer la opción de impresión del libro, considerando únicamente la impresión de todo el libro o bien de rango de páginas.
<i>Requerimientos No Funcionales:</i> <ol style="list-style-type: none">1. Enviar electrónicamente el libro a una dirección de correo electrónica.
<i>Descripción Visual del Software:</i> A partir de una biblioteca de documentos se seleccionará un libro electrónico en PDF, el cual se abrirá automáticamente la aplicación en forma de libro la cual brindará una interacción completa tal y como se realiza con un libro normal. El movimiento de las páginas y el agregar un separador como marcador, para que al cerrar se almacene dentro de la aplicación, y se reutilice al abrir nuevamente el mismo libro indicando la hoja en que fue hecho el marcador.

Establecido el objetivo general y los específicos del *software*, se aclara la población objetivo a la cual se dirige el *software* para tomar las consideraciones necesarias en las siguientes fases. Posteriormente se recopila la información sobre las funciones que ejecutará el *software*, para luego tener una perspectiva clara de qué elementos se vuelven funcionales y cuáles no. Con esta información se puede generar una descripción general de cómo se desea la experiencia visual e iniciar el proceso de creación de la interfaz en tercera dimensión.

- b) Definición de las posibles metáforas y objetos del mundo real, es una lista general de las metáforas y objetos de la realidad, propuestos para ser mapeados y evaluados según la descripción visual de la descripción general del *software*. Con este documento deberá iniciarse la aplicación y evaluación de metáforas.

Figura 23. Estructura de definición de metáforas y objetos

Definición Inicial de Metáforas y Objetos		
Metáforas		
No.	Título	Descripción
1	<Título de metáfora 1>	<Descripción de la metáfora 1>
2	<Título de metáfora 2>	<Descripción de la metáfora 2>
Objetos		
No.	Nombre	Metáforas
1	<Nombre Objeto 1>	<1, 2>
2	<Nombre Objeto 2>	<1, 2>

Figura 24. Ejemplo de definición de metáforas y objetos

Definición Inicial de Metáforas y Objetos		
Metáforas		
No.	Título	Descripción
1	Lectura de Libro	Permite el despliegue de un libro para su lectura, considerando que además del cambio de páginas, este podrá tener separador de páginas para continuar su lectura.
Objetos		
No.	Nombre	Metáforas
1	Libro	1
2	Separador de Página	1

- c) Aplicación y evaluación de metáforas: se debe aplicar y generar toda la documentación tal y como se describió en el capítulo 2, con el fin de integrarse como validadores durante el diseño de la interfaz. Es importante realizar todos los análisis y documentarlos, ya que serán el punto de partida para el siguiente paso.
- d) Construcción y posicionamiento de objetos en el ambiente tridimensional, considerando el resultado de la evaluación externa de las metáforas, se debe trasladar ya a un diseño gráfico la ubicación inicial de los objetos, para que los desarrolladores inicien con la construcción y codificación.
- e) Especificación de detalle gráfico, contendrá los detalles técnicos que se desean alcanzar, tales como:
- Lenguaje de programación
 - Bibliotecas o componentes a utilizar
 - Plataforma de sistemas operativos a los que va dirigida
 - Arquitectura gráfica para el desarrollo
 - Requerimientos mínimos a los que va dirigida
 - Variables gráficas globales para personalización
 - Configuración mínima y máxima alcanzable por las variables gráficas
- f) Definición del texto y multimedia a utilizar, éste es un documento que generaliza la configuración los idiomas a utilizar si aplican, cómo se definirá la ubicación y el diseño de imágenes o video necesarios para la ejecución de la interfaz.

Este documento se encuentra separado de la especificación de detalle gráfico debido a que puede extenderse bastante, y debe ser parte fundamental del manual técnico del software.

La medición dentro de estos lineamientos se lleva a cabo, considerando como métrica el cumplimiento de todos los puntos. Éste debe ser revisado al menos en dos ocasiones por el equipo de desarrollo en su totalidad.

5.2 Lineamientos para evaluación de prototipos de interfaces en 3D

El objetivo de estos lineamientos, es permitir agregar un conjunto de pruebas diseñadas a la medida por tarea a ejecutar, dentro de las interfaces en tercera dimensión. La aplicación de éste conjunto debe agregarse a la batería de pruebas de los prototipos de *software*, durante las etapas de desarrollo y pruebas. Generalmente, el equipo de pruebas es el encargado de realizar dichas pruebas; sin embargo, este conjunto recomienda más adelante incluir a los desarrolladores, con el fin de entender claramente cuándo son necesarios cambios en los prototipos.

Los lineamientos para evaluación de prototipos propuesta para interfaces en tercera dimensión, se basa en dos tipos de evaluaciones:

- Evaluación de interacción
- Evaluación de usuario

Cada una con métricas que permiten el análisis de resultados para considerar qué modificaciones son necesarias en el funcionamiento de los prototipos de interfaz en tercera dimensión.

Para que el diseño desarrollado cumpla con una serie de aspectos esenciales, que exploten todas sus ventajas, considerando las opiniones del equipo de trabajo y de los usuarios para adecuar el *software* a sus necesidades.

Cuando este conjunto aplicado provea la información necesaria para generar uno o varios cambios en los prototipos, es necesario considerar lo siguiente:

- Determinar si el cambio pertenece estrictamente a un error de desarrollo;
- Si el error pertenece al diseño, deben realizarle las correcciones necesarias aplicando modificaciones en los instrumentos del conjunto de lineamientos de requerimientos y en el prototipo, de lo contrario, únicamente se corregirá el prototipo.

5.2.1 Evaluación de interacción

Este conjunto de lineamientos de evaluación se basa en métricas que permiten realizar un análisis cuantitativo de los resultados, para luego ser interpretados por los analistas y diseñadores del *software*, determinando qué cambios son necesarios realizar en el prototipo evaluado.

Para crear esta evaluación debe prepararse al menos un grupo de 20 usuarios que la realicen, para que la evaluación de una muestra sea lo suficientemente confiable. Previo a iniciarla se deben preparar los siguientes requisitos:

- a. Definir las tareas o funciones que los usuarios van a ejecutar. Éstas deben ser cuidadosamente seleccionadas y deben corresponderse con aquellos procesos que se estén evaluando. Es muy común que se

inicien con las funciones más complejas que requieran más tiempo y dificultad para completarlas.

- b. Establecer el objetivo de cada función para poder ponderar la ejecución de la función como éxito o fracaso.
- c. Definir cuestionarios para validar la satisfacción del usuario en el uso de la herramienta considerando las funciones que ejecutarán según el primer inciso de estos prerequisites.
- d. Preparar una herramienta que permite recolectar los datos y consultarlos de forma inmediata de ser posible. De no contar con alguna herramienta se deberá hacerlo de forma manual, cronometrando los tiempos y recogiendo los datos en alguna plantilla prediseñada.

Existen estándares de calidad que recomiendan evaluaciones de interfaces agrupadas, para realizar evaluaciones ordenadas y sencillas de analizar. La agrupación propuesta se muestra a continuación:

Figura 25. **Ejemplo de mediciones de evaluación de interacción**



Las variables propuestas que se evaluarán, según su agrupación, son las siguientes:

Medición de efectividad

Permiten medir la exactitud con la que se alcanzan los objetivos de una tarea o función específica. Se genera el listado total de tareas a evaluar y se miden las variables siguientes para cada tarea.

Tabla X. **Métricas de medición de efectividad**

Métrica	Ponderación y fórmula de medición	Meta (Recomendada)
Porcentaje total de tareas o funciones completadas.	0% a 100% - Promedio de usuarios que completaron una tarea.	90%
Porcentaje de tareas o función completadas en el primer intento.	0% a 100% - Promedio de usuarios que completaron una tarea en el primer intento.	55%
Porcentaje de tareas o función completadas en el segundo intento.	0% a 100% - Promedio de usuarios que completaron una tarea en el segundo intento.	30%
Porcentaje de tareas o función completadas en más dos intentos.	0% a 100% - Promedio de usuarios que completaron una tarea en tres o más intentos.	15%
Porcentaje de usuarios que completaron todas las tareas o funciones.	0% a 100% - Promedio de usuarios que completaron todas las tareas.	85%
Porcentaje de éxitos y fracasos por tarea o función.	0% a 100% - Promedio de tareas ejecutadas exitosamente por usuarios.	Éxitos 90% - Fracasos 10%
Porcentaje de veces que los usuarios solicitan ayuda para realizar una tarea.	0% a 100% - Promedio de usuarios que solicitaron ayuda por cada tarea.	0% - 15%
Porcentajes de objetos que no fueron utilizados.	0% a 100% - Promedio de objetos que no fueron utilizados por los usuarios.	5% - 10%

Medición de eficiencia

Mide el esfuerzo que un usuario tiene que hacer para conseguir un objetivo. Las variables propuestas se presentan a continuación:

Tabla XI. **Métricas de medición de eficiencia**

Métrica	Ponderación y fórmula de medición	Meta (Recomendada)
Tiempo transcurrido para completar cada tarea o función.	Minutos y segundos - Medición de tiempo para completar una tarea.	Establecer un tiempo promedio por tarea.
Porcentaje o número de errores cometidos por tarea o función.	0% a 100% - Promedio de errores desplegados por tarea.	0% - 10%
Porcentaje de errores o problemas según su severidad.	0% a 100% - Promedio de errores según severidad por tarea.	0% - 5%
Tiempo transcurrido para recuperación de errores.	Minutos y segundos - Medición de tiempo para que el <i>software</i> se recupere de un error por tarea.	Establecer un tiempo promedio por tarea.
Número de acciones realizadas para completar una tarea o función.	0% a 100% - Promedio de acciones realizadas por usuario, para completar una tarea.	Establecer el número de acciones promedio para completar cada tarea.
Tiempo empleado en escenarios de la interfaz de tercera dimensión para localizar una función predefinida.	Minutos y segundos - Medición de tiempo para localizar un escenario por usuario y tarea.	Establecer un tiempo promedio por localización de escenario.
Porcentaje o número de veces que se acude a ayudas, o documentación de ayuda.	0% a 100% - Promedio de usuarios que solicitaron ayuda para realizar una tarea.	%0 – 20%

Medición de satisfacción

Esta medición permite evaluar la principal ventaja de las interfaces en tercera dimensión, ya que evalúan la parte emocional o subjetiva con la reacción de un usuario. Las siguientes variables pueden utilizarse para este fin:

Tabla XII. **Métricas de medición de satisfacción**

Métrica	Ponderación y fórmula de medición	Meta (Recomendada)
Porcentaje de usuarios que después de utilizar el producto lo recomendaría a un amigo.	Si y No - Promedio de la cantidad de respuestas con valor "Si".	75% - 100%
Proporción de adjetivos positivos o negativos que cada usuario dé al producto.	0% a 100% - Proporción de la cantidad de adjetivos distinguiendo positivos y negativos.	Positivos: 70% - Negativos 30%
Porcentaje de usuarios que califican el producto más fácil de usar.	Si y No - Promedio de la cantidad de respuestas con valor "Si".	85%
Proporción de veces que el usuario expresa satisfacción o insatisfacción del uso del <i>software</i> .	0% a 100% - Proporción de veces que el usuario expresa satisfacción.	Positivos: 70% Negativos 30%
Porcentaje de tareas que considera adecuadas dentro del <i>software</i> .	0% a 100% - Promedio de usuarios que completaron una tarea específica.	90%
Porcentaje de escenarios que preferiría cambiar.	0% a 100% - Cantidad de apariciones por escenario.	Definido por los escenarios con más apariciones.
Qué funciones o tareas cree innecesarias.	0% a 100% - Cantidad de función por aparición.	Definido por las tareas con más apariciones.

Se pueden encontrar otras métricas o mediciones para ergonomía del *software* multimedia, como parte de los estándares de la IEEE, y otros dentro del estándar internacional ISO 14000.

Interpretación de los resultados

La forma más sencilla de desplegar los resultados de la medición de métricas es a través de gráficas. Éstas permitirán mostrar, claramente, los puntos críticos que se deben mejorar de la interfaz de usuario en tercera dimensión. Se considerará como ejemplo la primera variable de la medición de la eficacia, con los siguientes datos.

Tabla XIII. **Usuarios que completaron una tarea (0% ó 100%)**

Usuario(s)	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
Usuario 1	100	100	100	100	100
Usuario 2	0	0	0	0	0
Usuario 3	100	0	0	100	100
Usuario 4	0	100	0	100	100
Usuario 5	0	0	0	100	0
Usuario 6	0	0	100	0	0
Usuario 7	100	0	0	100	100
Usuario 8	100	0	0	100	0
Usuario 9	0	0	0	100	0
Usuario 10	100	0	0	100	100
Usuario 11	0	0	0	100	0
Usuario 12	0	0	0	100	0
Usuario 13	100	0	0	100	100
Usuario 14	100	0	0	100	0
Usuario 15	100	0	0	0	0
Usuario 16	100	0	0	100	100
Usuario 17	0	0	0	100	0
Usuario 18	0	0	0	0	100
Usuario 19	100	0	0	0	0
Usuario 20	0	100	0	0	100

Tabla XIV. **Ejemplo usuarios que completaron una tarea (0% ó 100%) para la lectura del libro**

Usuario(s)	Abrir Documento	Aplicar Separador	Impresión del Libro	Envío de Correo Electrónico
Usuario 1	100	100	100	100
Usuario 2	100	0	100	0
Usuario 3	100	0	100	0
Usuario 4	100	100	0	100
Usuario 5	100	100	0	0

Posteriormente se obtiene un resumen que muestra la recolección de información de la siguiente forma:

Tabla XV. **Promedio de usuarios que completaron una tarea (0% ó 100%)**

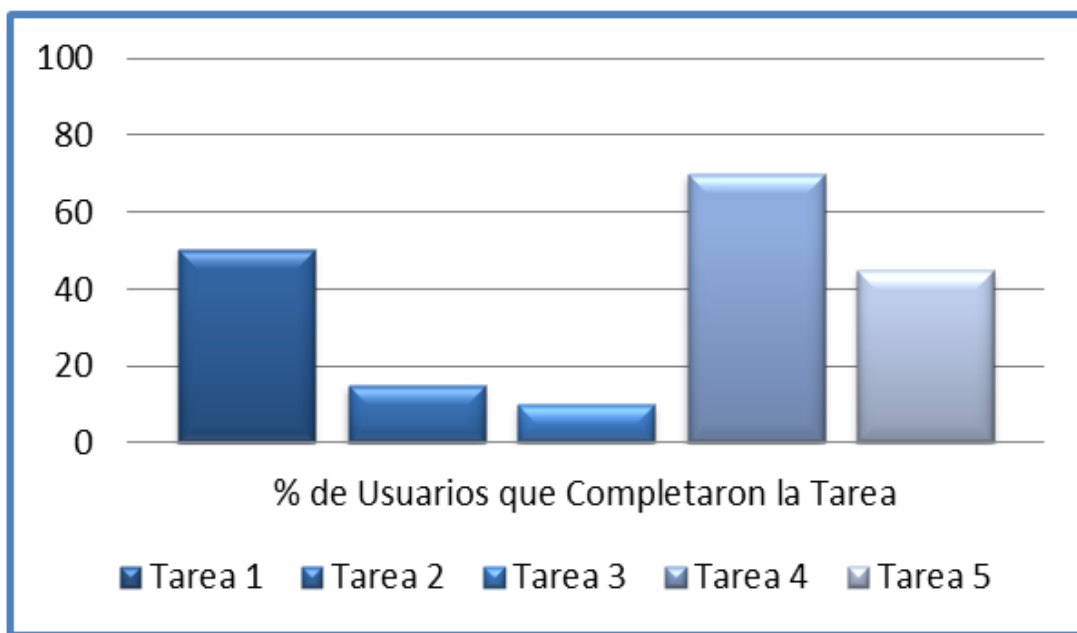
Usuario(s)	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
% de usuarios que completaron la tarea	50	15	10	70	45

Tabla XVI. **Ejemplo del promedio de usuarios que completaron una tarea (0% ó 100%), para la lectura del libro**

Usuario(s)	Abrir Documento	Aplicar Separador	Impresión del Libro	Envío de Correo Electrónico
% de usuarios que completaron la tarea	100	60	60	40

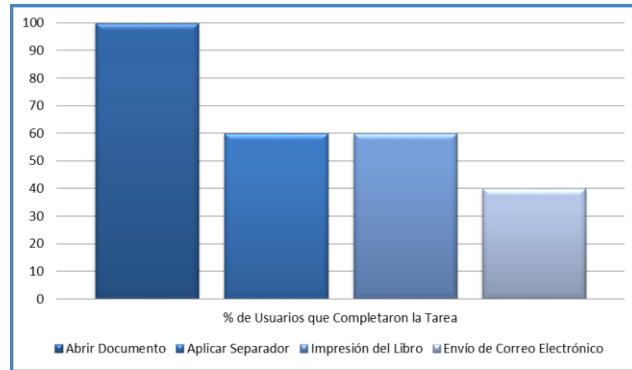
Con lo cual se genera la siguiente gráfica, donde se muestra el porcentaje de usuarios que fueron capaces de completar cada una de las 5 tareas que se ejemplifican para una muestra de 20 usuarios:

Figura 26. **Ejemplo de gráfica de promedio de usuarios que completaron una tarea**



Para cada métrica es necesaria la revisión y la interpretación de los datos, para que los analistas determinen cuáles son los problemas que deben corregirse, esta tarea debe tener básicamente a un equipo que realice el análisis e interpretación.

Figura 27. **Ejemplo de gráfica de promedio de usuarios que completaron una tarea para la lectura del libro**



Considerando los instrumentos definidos en el modelo de requerimientos, esta gráfica nos muestra cómo de dos de los tres requerimientos funcionales, solo el 60% de los usuarios logró completar las tareas con éxito. Con una aplicación tan sencilla las 3 tareas que relacionan directamente a los requerimientos funcionales deberían estar por encima del 90%, para que consideramos que está bien construida.

En el caso del requerimiento no funcional, se obtuvo un 40% por lo que debe considerarse si beneficiará a la herramienta y si aún se debe invertir en su corrección.

Comparación de resultados de una evaluación de interacción y su versión anterior

Como parte de la evaluación de interacción, una vez que se completa una fase de la evaluación de interacción, se corrigen los errores y se implementan las mejoras, se generará un nuevo prototipo, al cual deberá aplicarse nuevamente la evaluación, pero esta vez se agregará la comparación

del resultado actual frente al resultado anterior, para determinar qué avances se lograron y qué errores restan por corregir. Solamente se debe tener especial cuidado que esta comparación se puede realizar únicamente entre el universo de tareas o funciones que estén contempladas en ambas versiones, anterior y actual.

La siguiente tabla muestra un ejemplo de una evaluación de interacción que muestra la comparación.

Tabla XVII. **Tiempo transcurrido para completar cada tarea (en minutos)**

Tarea	Versión Anterior	Versión Rediseño
Tarea 1	16,7	5,1
Tarea 2	9,8	9,6
Tarea 3	5	5
Tarea 4	3,1	3
Tarea 5	5,5	5,4

En este cuadro se puede observar claramente dónde se aplicaron las mejoras y el impacto que tuvieron entre ambas versiones del *software*. Aunque esto complique un poco la documentación del desarrollo de *software*, es necesario para que esta tecnología pueda ser integrada y considerada como una opción. Se ve cómo todos los tiempos para realizar las diferentes tareas, se redujeron considerablemente.

5.2.2 Evaluación de usuario

Como parte de los lineamientos para evaluación de las interfaces en tercera dimensión y apoyado en el desarrollo de *software* de calidad, se propone una evaluación de usuario, la cual permite validar tanto a nivel de

equipo de pruebas como de usuarios de la aplicación, el garantizar que la interfaz cumpla con sus funciones y objetivo. Para este tipo de pruebas es recomendable que el equipo de pruebas cuente con una persona capacitada para evaluar el cumplimiento de los requerimientos de comportamiento, internos, y de desempeño del *software* con el usuario final y con los analistas programadores.

Ésta es una evaluación cualitativa y cuantitativa, que permitirá evaluar qué tan atractiva y funcional es la idea de utilizar el *software* con interfaces en tercera dimensión, y si los objetivos del uso del *software* están bien dirigidos.

La primera etapa de la evaluación, requiere que se consoliden grupos entre 3 y 5 usuarios, según las características de las posibles poblaciones objetivo que utilizarán el *software*, y dentro de cada grupo se encontrará un analista programador o bien un probador de *software*.

A continuación se deben determinar las funciones más utilizadas para cada perfil de usuario seleccionado con lo cual llevarán a cabo la evaluación.

Las siguientes variables se deberán considerar para cada perfil, personalizando la evaluación para los usuarios:

- Determinar si le llama la atención investigar cómo funciona el *software* sin ayuda, o bien se desea una inducción práctica de su uso;
- Determinar si la calidad de las imágenes son apropiadas; se recomienda que la respuesta sea tipo falso y verdadero;
- Determinar si el tamaño y la navegación entre los objetos es la adecuada, se recomienda que la respuesta sea falso y verdadero;

- Especificar qué objetos no son de su agrado y el motivo de por qué le gustaría removerlos;
- Qué funciones no cumplen con sus expectativas, según el objetivo general del *software*;
- Determinar si el usuario recomendaría el uso del *software* con interfaz de tercera dimensión a un familiar o amigo;
- Es esencial que los participantes de la evaluación intenten realizar las tareas que realizará un usuario de ese producto. Y que todos, incluyendo a los analistas y probadores, apliquen las evaluaciones de tipo usuario que permitirán la mejora de este *software*.

La participación del analista programador o probador en cada grupo, tiene un fin específico, el podrá realizar un informe interno sobre los comentarios de los usuarios para garantizar la veracidad de la evaluación.

Para todas las variables que quedarán como métricas, es indispensable aplicar una interpretación, como se explicó anteriormente en la evaluación de interacción.

CONCLUSIONES

1. La tendencia de crecimiento tecnológico, indica que las interfaces en tercera dimensión son la próxima generación en la clasificación de interfaces de usuario, y seguramente del *software*, permitiendo al hombre brindar realismo en la ejecución de sus tareas ejecutadas por computadora, generándole más confianza, seguridad y facilidad en el uso del *software*.
2. Los requerimientos de comportamiento aplicados a través de las metáforas, son instrumentos prácticos para el diseño de *software* con interfaz en tercera dimensión; sin embargo, éstos son inservibles de no aplicarse su correcta evaluación y manejo, como fundamento del proyecto de desarrollo de *software*.
3. Existe suficiente tecnología de *hardware* con capacidad de ejecutar *software* en tercera dimensión; sin embargo, a nivel de *software* el universo de herramientas de desarrollo para interfaces en tercera dimensión es limitado, considerando además que se requiere de un total dominio de las bibliotecas y componentes propios de los lenguajes de programación para lograr el desarrollo de *software*.
4. Mientras la etapa de madurez de esta tecnología no llegue, el impacto más grande se encontrará en el costo de desarrollo de la tecnología, ya que implica la adquisición de *hardware*, herramientas y componentes de desarrollo; el capital humano y capacitación para el dominio de las

herramientas de desarrollo y del sentido espacial que deben tener los desarrolladores para generar el *software*.

5. Únicamente el *software* que cumpla con todos los requerimientos de comportamiento y funcionamiento interno, podrá obtener el impacto esperado, es decir, estar diseñado, modelado y evaluado a la medida, de tal forma que el usuario pueda interpretar el manejo sin ayuda, saber para qué sirve y cómo puede utilizarlo de forma eficaz, eficiente y cumpliendo con el objetivo claro del *software*.

RECOMENDACIONES

1. Es importante hacer énfasis en investigaciones más detalladas o profundas para poner en práctica esta tecnología, ya que la facilidad y practicidad que darán las interfaces de tercera dimensión a los usuarios en el uso de los equipos de cómputo será de gran beneficio como la siguiente generación del *software*.
2. Esta investigación brinda lineamientos para iniciar un proyecto de *software* con interfaz en tercera dimensión. Sin embargo, es necesario realizar un análisis para la optimización de éstos y sus instrumentos, a través de proyectos prácticos de implementación de *software* con interfaz en tercera dimensión.
3. La construcción de *software* con interfaces en tercera dimensión, actualmente, ya ha iniciado en distintos proyectos en el mundo. Dadas las características que poseerá será muy importante explotarla en las áreas que requieren más importancia en nuestro país, tal es el caso de educación y salud, además de determinar en qué otras áreas de especialización pueden realizarse estudios de factibilidad para su aplicación.
4. Considerar incluir en el pensum de la carrera de ingeniería de ciencias y sistemas este tema, a fin de enseñar a los estudiantes, la importancia de generar sistemas con interfaz de usuario apropiadas para explotar todas las capacidades de los usuarios. Despertando su iniciativa de investigación en torno a esta tecnología, e incluso incentivar ejercicios

profesionales supervisados, por ser parte de la actualidad tecnológica, ya que aunque no se cuente con los instrumentos formales de desarrollo, sí es importante para los estudiantes el fomentar conceptos a este nivel para su futuro profesional.

5. Promover la creación de un grupo de personas que implemente esta tecnología a través de proyectos especializados, que apoyen el crecimiento tecnológico con apoyo del CONCYT, y la presenten en los congresos internacionales de 3DUI de la IEEE. Ésto permitirá crear en Guatemala departamentos de investigación tecnológica que abrirían las puertas para la expansión de nuevos mercados y ser parte de proyectos de tecnología de punta a nivel mundial.

REFERENCIAS

1. 3DUI (*3D User Interface*). Organización dedicada al diseño y evaluación de interfaces en tercera dimensión. [ref. 18 de agosto de 2010]. Disponible en Web: <<http://www.3dui.org>>
2. *IEEE VGTC (Visualization & Graphics Technical Committee)*. [ref. 30 de septiembre de 2010]. Disponible en Web: <<http://www.cs.sfu.ca/~vis/Track/3DUI.html>>
3. HEISS, Janice J. *Going 3D Project Looking Glass*. Oracle. Octubre 2004. [ref. 10 de agosto de 2010]. Disponible en Web: <<http://java.sun.com/developer/technicalArticles/J2SE/Desktop/lookingglass/>>
4. *Tactile 3D Interface*. *Upper Bounds Interactive Incorporated*, [ref. 13 de septiembre de 2010]. Disponible en Web: <<http://www.tactile3d.com/>>
5. *2D vs 3D, Implications on Spatial Memory*. *Proceedings of IEEE InfoVis 2001*. Tavanti, M. and M. Lind. *Symposium on Information Visualization, San Diego, 2001* [ref. 20 de noviembre de 2010]. Disponible en Web: <<http://portal.acm.org/citation.cfm?id=976314>>

BIBLIOGRAFÍA

1. BOWMAN, Doug A.; TECH, Virginia.; KRUIJFF, Ernst.; IMK, Fraunhofer.; LAVIOLA, Joseph J.; POUPYREV, Jr. Ivan. *3D User Interfaces: Theory and Practice*. Estados Unidos: Addison-Wesley/Pearson Education, 2004, 512 p. ISBN-13: 978-0201758672.
2. CONNER, D. B.; SNIBBE, S. S.; HERNDON, K. P.; ROBBINS, D. C.; ZELEZNIK, R. C.; VAN DAM, A. *Three-Dimensional Widgets.: Proceedings of the 1992 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics*. 26 p.
3. COCKBURN Andy,; MCKENZIE, Bruce. *Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments*. University of Caterbury, Christchurch, New Zeland, 5-8 p.
4. SPOLSKY, Joel. *User Interface Design for Programmers*. [ref. 30 de septiembre de 2010]. Disponible en Web: <<http://www.joelonsoftware.com/uibook/chapters/fog0000000060.html>>
5. JIMÉNEZ, Ramón Villahermosa; SÁNCHEZ DE LA BLANCA, Jesús José. *Hacia Nuevos Modos de Interacción: Interfaces Post-WIMP. 2002*. [ref. 15 de agosto de 2010]. Disponible en Web: <<http://www.dsi.uclm.es/asignaturas/300210/Doctorado%2009-10-PostWimp.pdf>>

ANEXOS

Contenido primer seminario de interfaces de usuario en 3D

3DUI SYMPOSIUM 2006 CALL FOR PAPERS

3DUI 2006: THE 1ST IEEE SYMPOSIUM ON 3D USER INTERFACES 2006

March 25 and 26, Alexandria, Virginia USA
(In conjunction with IEEE Virtual Reality 2006)
<http://www-human.ist.osaka-u.ac.jp/3dui2006>

The IEEE Symposium on 3D User Interfaces 2006 is the first international symposium focused on the topic of 3D UIs and will be held to trigger discussions among participants and to provide an intensive exchange between industrial and academic researchers working in various 3DUI research areas. The Symposium builds on and expands the scope of the workshop "New Directions in 3D User Interfaces" in March 2005 and the workshop "Beyond Wand and Glove Based Interaction" in March 2004. The symposium will last for a day and a half, and will be followed immediately by the IEEE VR conference. Ravin Balakrishnan (University of Toronto) will give a keynote address.

The theme of the Symposium will again be expanded and the scope will include all types of 3D UI research. The Symposium themes include but are not limited to the following topics:

- 3D input devices
- 3D display devices
- 3D interaction techniques
- 3D user interface metaphors
- Collaborative 3D interaction
- Evaluation methods for 3D UIs
- Applications of 3D UI techniques

The Symposium solicits high quality papers and technotes. Papers (up to 8 pages, typically 6-8 pages) should describe original and mature research results and will typically include some evaluation of the presented results. Technotes (up to 4 pages, typically 2-4 pages) should contain preliminary results of research or design work within the scope of 3D UIs. Both should be prepared in IEEE VGTC format (<http://www.cs.sfu.ca/~vis/Tasks/camera.html>) and submitted through the submission web site in PDF format. All submissions will be reviewed by the members of the program committee. The accepted papers and technotes will be published in the official Symposium proceedings.

For paper submission, registration, and hotel details, please see the Symposium website:
<http://www-human.ist.osaka-u.ac.jp/3dui2006/>

Important Dates


- Papers and Technotes submission: November 30, 2005
- Review decisions: January 10, 2006
- Camera-ready copy due: January 20, 2006

Symposium Chairs (3dui2006spc@hi-mail.ise.eng.osaka-u.ac.jp)

Yoshifumi Kitamura (Osaka university, Japan) Doug Bowman (Virginia Tech, USA)
Bernd Froehlich (Bauhaus University Weimar, Germany) Wolfgang Stuerzlinger (York University, Canada)

Contenido sexto seminario de interfaces de usuario en 3D

IEEE Symposium on 3D User Interfaces 2011



In conjunction with IEEE Virtual Reality 2011 March 19-20, 2011 SUNTEC International Convention Center, Singapore

Overview & Important Dates

We invite you to participate in the [IEEE Symposium on 3D User Interfaces 2011](#) in Singapore, March 19-20, 2011. Singapore is one of the most advanced and modernized cities in Asia. It is home to two of the world's top-ranking institutes of higher learning; National University of Singapore (NUS) and Nanyang Technological University (NTU). A large presence of leading hardware, software, IT services and internet companies in Singapore creates significant synergies and provides fertile ground for collaboration.

The IEEE Symposium on 3D User Interfaces 2011 is the sixth international symposium focused on the topic of 3D User Interfaces and provide an opportunity for industrial and academic researchers to exchange ideas about the state of art 3DUI research. It builds on successful 3DUI workshops in [2004](#) and [2005](#), as well as the IEEE 3DUI Symposiums in [2006](#), [2007](#), [2008](#), [2009](#), and [2010](#).

The symposium will last for two days, and will be followed immediately by the [IEEE Virtual Reality conference](#). The theme of the Symposium will cover all areas of 3D UI research, including the following topics:

- 3D interaction techniques
- 3D user interface metaphors
- 3D UIs for games and entertainment
- 3D input devices
- Desktop 3D UIs
- 3D UIs for VR and AR
- Evaluation methods for 3D UIs
- 3D display devices
- Applications of 3D UI techniques
- 3D UIs for CAD
- Biologically inspired 3D UIs
- Collaborative 3D interaction