UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

DISEÑO E IMPLEMENTACIÓN DE EJERCICIOS PARA ENTRENAMIENTO QUIRÚRGICO VIRTUAL EN CIRUGÍA MÍNIMA INVASIVA(*)

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 20 puntos (Mención Honorífica y Mención Publicación).

JURADO EXAMINADOR

Firma:
Nombre: Carlo Magurno

Firma:
Nombre: Assaf Yamin

Firma:
Nombre: Omaira Rodríguez

REALIZADO POR Martínez, Marvi.
Motezuma, José.

PROFESOR GUIA Lic. Rodríguez, Omaira.

FECHA Octubre de 2005.

* Parcialmente financiado por el CDCH, UCV. Proyecto PTP 08.12.5294.2004

SINOPSIS

Este Trabajo Especial de Grado titulado **"Diseño e Implementación de Ejercicios para Entrenamiento Quirúrgico Virtual en Cirugía Mínima Invasiva"**, consiste en diseñar e implementar una serie de ejercicios a ser utilizados en el simulador virtual de cirugía laparoscópica desarrollado en el proyecto LAPAROS. Estos ejercicios tienen como objetivo entrenar a cirujanos en las técnicas básicas de cirugía mínima invasiva las cuales son de suma importancia para el desarrollo profesional de los mismos.

Como metodología a seguir en este trabajo se escogió la del Proceso Unificado, la cual es de amplio uso en proyectos de gran tamaño, desarrollados por grandes grupos de trabajo, por estar basado en la programación orientada a objetos y en un desarrollo incremental lo cual ayuda a disminuir los riesgos que pueden amenazar el ciclo de vida de estos proyectos.

Para la implementación de los ejercicios se utilizó la herramienta Visual C++ 6.0 perteneciente a Microsoft Visual Estudio 6.0 y como motor 3D para la creación y manipulación de escenas 3D se utilizó la librería OpenGL®.

En este trabajo se implementaron 5 ejercicios que desarrollan las destrezas y habilidades básicas necesarias para manipular instrumentos y realizar procedimientos quirúrgicos en Laparoscopia. Esta implementación fue probada por estudiantes de medicina de la Universidad Central de Venezuela y por doctores expertos en cirugía mínima invasiva del Instituto de Cirugía Experimental de la misma casa de estudios.

ÍNDICE GENERAL

DEDICATORIA I
AGRADECIMIENTOS II
SINOPSIS V
ÍNDICE GENERAL VI
ÍNDICE DE TABLAS VIII
ÍNDICE DE FIGURAS VIII
INTRODUCCIÓN 1
CAPÍTULO I. PROBLEMA 2
I.1 PLANTEAMIENTO DEL PROBLEMA 2
I.2 OBJETIVO GENERAL 3
I.3 OBJETIVOS ESPECÍFICOS 3
I.4 JUSTIFICACIÓN 4
I.5 LIMITACIONES Y ALCANCE 5
I.5.1 LIMITACIONES 5
I.5.2 ALCANCE 5
CAPÍTULO II. MARCO TEÓRICO 7
II.1 LAPAROSCOPIA 7
II.2 REALIDAD VIRTUAL 11
II.2.1 CARACTERÍSTICAS DE UN SISTEMA DE REALIDAD VIRTUAL 12
II.2.2 HARDWARE ESPECIALIZADO PARA UN SISTEMA DE REALIDAD VIRTUAL 12
II.2.3 APLICACIONES DE LA REALIDAD VIRTUAL 13
II.3 ENTRENAMIENTO VIRTUAL 14
II.4 EJERCICIOS TRADICIONALES PARA EL ENTRENAMIENTO EN CIRUGIA LAPAROSCÓPICA 15
II.5 OPENGL® 16
II.6 DIRECTX® 19
II.6.1 VENTAJAS DE DIRECTX® 19
II.6.2 COMPONENTES DE DIRECTX® 19
II.7 SIMULADORES LAPAROSCÓPICOS 21
II.7.1 LTS-2000 21
II.7.2 LAP MENTOR 22
II.7.3 PROMIS™ 23
II.7.4 SIMULAP V1.0 24
CAPÍTULO III. MARCO METODOLÓGICO 25
III.1 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE 25
III.1.1 CARACTERÍSTICAS DEL PROCESO UNIFICADO 26
III.1.2 FASES DEL PROCESO UNIFICADO 27
III.1.3 FLUJOS DE TRABAJO DEL PROCESO UNIFICADO 29
III.2 ENTREGABLES GENERADOS 31
CAPÍTULO IV. DESARROLLO 32
IV.1 CAPTURA DE REQUERIMIENTOS 32
VI.1.1 EJERCICIOS PLANTEADOS 35
IV.2 ANÁLISIS 41

IV.2.1 ANÁLISIS DE LA PLATAFORMA DE SOFTWARE 42
IV.2.2 ANÁLISIS DE LA PLATAFORMA DE HARDWARE 44
IV.2.3 ANÁLISIS DE LA ARQUITECTURA DEL SISTEMA 45
IV.3 DISEÑO 46
IV.3.1 DISEÑO DE LAS ESCENAS DEL SISTEMA 48
IV.3.2 DISEÑO DE LOS EJERCICIOS DEL SISTEMA 50
IV.3.3 DISEÑO DE LOS OBJETOS DEL SISTEMA 51
IV.4 IMPLEMENTACIÓN 53
IV.4.1 LECTURA, CARGA Y CONFIGURACIÓN DE PARÁMETROS DE VISUALIZACIÓN DE LA ESCENA 53
IV.4.2 GENERACIÓN DE LAS ESCENAS 57
IV.4.3 CAPTURA Y MANEJO DE LA DATA PROVENIENTE DEL RASTREADOR 58
IV.4.4 DETECCIÓN DE COLISIONES 59
IV.4.5 IMPLEMENTACIÓN DE LOS EJERCICIOS 59
IV.5 PRUEBAS 61
IV.5.1 CASO DE PRUEBA #1: TOCAR LOS COLORES 62
CAPÍTULO V. RESULTADOS 65
CONCLUSIONES 67
RECOMENDACIONES 69
REFERENCIAS BIBLIOGRÁFICAS 70
LIBROS 70
TRABAJOS ESPECIALES DE GRADO 70
DOCUMENTOS ELECTRÓNICOS 70
PAGINAS WEB 71
GLOSARIO 75
APÉNDICE A. CASOS DE USO 77
APÉNDICE B. DIAGRAMAS DE ACTIVIDADES 81
APÉNDICE C. DIAGRAMAS DE SECUENCIA 85
APÉNDICE D. DISEÑO DE LOS EJERCICIOS 89
APÉNDICE E. IMPLEMENTACIóN DE LOS EJERCICIOS 93
APÉNDICE F. GRAFOS DE ESCENA 99
APÉNDICE G. ESTRUCTURAS DE ARCHIVOS UTILIZADOS 101
APENDICE H. CASOS DE PRUEBA 104
APÉNDICE I. ENCUESTAS APLICADAS DURANTE LAS PRUEBAS 107
APÉNDICE J. PAPER ENVIADO A CONGRESO CIMENICS 2006 126
ANEXO A. UNIFIED MODELING LANGUAJE – UML (LENGUAJE UNIFICADO DE MODELADO) 127
ANEXO B. SIMULAP v1.0 131
ANEXO C. QUATERNIONES 133
ANEXO D. DETECCIÓN DE COLISIONES Y ÁRBOLES AABB [3] 136
ANEXO E. MATRICES Y TRANSFORMACIONES EN OPENGL® [19,39] .. 141
ANEXO F. POLHEMUS ISOTRAK II 146

ÍNDICE DE TABLAS

Tabla 1. Clasificación de ejercicios para entrenamiento en Laparoscopia.... 16
Tabla 2. Entregables.... 31
Tabla 3. Funciones del sistema.... 38
Tabla 4. Casos de uso.... 39
Tabla 5. Objetos del sistema.... 52
Tabla 6. Caso de prueba #1: Tocar los Colores.... 62
Tabla 7. Primera medición del sistema con estudiantes.... 63
Tabla 8. Medición del sistema con expertos.... 64
Tabla 9. Caso de prueba #2: Seguir el Camino.... 104
Tabla 10. Caso de prueba #3: Pasar a través de los Aros.... 104
Tabla 11. Caso de prueba #4: Colocar las Figuras.... 105
Tabla 12. Caso de prueba #5: Cortar Cuerdas.... 106
Tabla 13. Segunda medición del sistema con estudiantes.... 107
Tabla 14. Quaterniones.... 133

ÍNDICE DE FIGURAS

Figura 1. Cirugía laparoscópica.... 7
Figura 2. Torre de Laparoscopia producción de imagen.... 9
Figura 3. Instrumentos laparoscópicos.... 10
Figura 4. Modelo del simulador laparoscópico LTS-2000.... 21
Figura 5. Modelo del simulador laparoscópico LAP Mentor.... 22
Figura 6. Modelo del simulador laparoscópico ProMIS.... 23
Figura 8. Fases y flujos del Proceso Unificado.... 27
Figura 9. Diagrama de casos de uso del sistema.... 39
Figura 11. Arquitectura del sistema.... 46
Figura 12. Diagrama de clases del sistema.... 46
Figura 14. Modelo 3D triangulizado.... 51
Figura 15. Posición de la cámara.... 54
Figura 16. Nodo raíz de Grafo de Escena.... 55
Figura 17. Nodo hoja de Grafo de Escena.... 56
Figura 18. Ejercicio Tocar los Colores.... 60
Figura 19. Gráfica 1 de Medición de errores.... 64
Figura 20. Gráfica 2 de Medición de errores.... 64
Figura 21. Diagrama de actividades #2.... 81
Figura 22. Diagrama de actividades #3.... 82
Figura 23. Diagrama de actividades #4.... 83
Figura 24. Diagrama de actividades #5.... 84
Figura 25. Diagrama de secuencia #2.... 85
Figura 26. Diagrama de secuencia #3.... 86
Figura 27. Diagrama de secuencia #4.... 87
Figura 28. Diagrama de secuencia #5.... 88
Figura 29. Ejercicio Seguir el Camino.... 93

Figura 30. Ejercicio Pasar a través de los Aros 95
Figura 31. Ejercicio Colocar las Figuras 96
Figura 32. Ejercicio Cortar las Cuerdas 98
Figura 33. Ejemplo de Grafo de Escena 99
Figura 34. Diagrama de casos de uso 128
Figura 35. Elementos de un diagrama de actividades 129
Figura 36. Simulador SIMULAP v1.0 131
Figura 37. Ejemplo de Quaternion 134
Figura 38. Verificación de intersección entre AABBs 137
Figura 39. Clasificación de triángulos después de la subdivisión de un AABB 138
Figura 40. Uno de los hijos tiene a todos los triángulos, y el otro queda vacío 138
Figura 41. Los AABBs hijos siguen presentando el mismo problema 139
Figura 42. La división a lo largo del segundo eje más largo evita tener AABBs hijos vacíos 139
Figura 43. Sistema de coordenada canónico 141
Figura 44. Secuencia de transformaciones en OpenGL® 145
Figura 45. Isotrak II 146

INTRODUCCIÓN

La Laparoscopia es una técnica de cirugía mínima invasiva que tiene sus inicios a principios del siglo XX, durante muchos años fue utilizada para examinar por medio de un telescopio, los órganos del abdomen. Con el paso del tiempo esta técnica se fue enriqueciendo y perfeccionando con la adición de procedimientos quirúrgicos y de diagnóstico.

Uno de los mayores y más recientes avances fue la introducción de videocámaras, permitiendo la amplificación de la imagen y que el proceso se pudiese llevar a cabo en grupo de cirujanos. Con esto la Laparoscopia se hace cada vez un procedimiento más común, requiriendo de profesionales de alto nivel, ya que el éxito de ésta depende en gran medida del entrenamiento del cirujano.

El avance tecnológico brinda la posibilidad de crear herramientas capaces de asistir en el proceso de entrenamiento de cirujanos para esta técnica, que implica la realización de un procedimiento visto en una pantalla en 2D llevado a cabo en un ambiente 3D.

El objetivo de este Trabajo Especial de Grado es desarrollar un conjunto de ejercicios en un ambiente de realidad virtual, que puedan formar parte del entrenamiento que debe realizar un cirujano para adquirir las destrezas y habilidades necesarias que requiere la cirugía laparoscópica.

La metodología utilizada es la del Proceso Unificado ya que entre algunas de sus características se encuentra la de adaptarse al desarrollo de proyectos de software que incluyen el trabajo de grupos grandes de personas. Esto debido a que el presente trabajo forma parte del proyecto LAPAROS, perteneciente a la Universidad Central de Venezuela que consiste en la construcción de un Simulador Virtual de Entrenamiento en Cirugía Laparoscópica.

CAPÍTULO I. PROBLEMA

I.1 PLANTEAMIENTO DEL PROBLEMA

La cirugía de acceso mínimo (cirugía mínima invasiva) ha revolucionado el mundo de la medicina. Esta técnica permite llevar a cabo operaciones complejas a través de pequeños orificios, donde son insertados los instrumentos a utilizar durante la intervención, logrando minimizar el dolor y los tiempos de recuperación y reposo de los pacientes, así como también disminuyendo los riesgos de infección que se encuentran asociados a una cirugía.

Aunque son muchas las ventajas para el paciente, es importante destacar la dificultad que implica para el cirujano el entrenamiento en esta técnica. La cirugía tradicional es llevada a cabo en un ambiente de quirófano tradicional en tres dimensiones, mientras que en las técnicas de acceso mínimo el ambiente con el que el cirujano se enfrenta es ofrecido por un monitor en tan sólo dos dimensiones y distintas a las reales, obligándolo a entrenarse continuamente en el desarrollo de una adecuada coordinación mano-ojo.

Tradicionalmente el entrenamiento en esta área, se ha llevado a cabo mediante el uso de cadáveres, maniquíes, animales vivos y en cirugías reales supervisadas. Esto se traduce en un número considerable de limitantes, entre ellos la incapacidad de repetir la intervención en cuestión para el aprendizaje óptimo, el riesgo que conlleva la práctica en seres vivos y la poca diversidad de patologías que pueden presentarse en un quirófano.

En la Facultad de Medicina de la Universidad Central de Venezuela, este entrenamiento se ha llevado a cabo de la forma tradicional, realizando las prácticas en conejos, perros, cerdos y ratas los cuales son criados dentro

de las mismas instalaciones de la Universidad, representando esto un inconveniente sanitario.

Con la intención de evitarse los inconvenientes que estas prácticas tradicionales conllevan y la necesidad existente de adaptarse a las nuevas tecnologías, surge el proyecto de investigación **LAPAROS** financiado por el CDCH de la UCV, en el cual participan investigadores del Centro de Bioingeniería de la Facultad de Ingeniería (CEBIO), Centro de Cirugía Experimental de la Facultad de Medicina y el Laboratorio de Computación Gráfica de la Facultad de Ciencias (LCG), donde se plantea una reestructuración de las técnicas de entrenamiento para la Laparoscopia (cirugía mínima invasiva), que incluye el desarrollo de un simulador virtual. La colaboración que se presta en este proyecto es el diseño y desarrollo de ejercicios de entrenamiento quirúrgico que pasarán a formar parte de este simulador.

I.2 OBJETIVO GENERAL

Desarrollar una serie de ejercicios para el entrenamiento virtual quirúrgico que un médico necesita realizar para adquirir destrezas en la utilización de instrumentos y técnicas utilizadas en procedimientos quirúrgicos en Laparoscopia.

I.3 OBJETIVOS ESPECÍFICOS

I.3.1 Estudio y análisis de los conceptos y terminología médica usados en Cirugía Mínima Invasiva.

I.3.2 Análisis del software:

I.3.2.1 Identificación de requerimientos de software y hardware.

I.3.2.2 Enlace de instrumentación médica con hardware y software.

I.3.3 Diseño e Implementación de los ejercicios..

I.4 JUSTIFICACIÓN

En los últimos años, la computación gráfica ha tenido un impacto de grandes proporciones sobre la tecnología en general. Esto ha generado especial interés para muchas disciplinas médicas, ya que su aplicación ha resultado en adelantos muy significativos.

En muchos casos se ha considerado que la medicina en Venezuela está un paso atrás de los adelantos mundiales en la materia. Uno de los motivos es que en el país aún no se ha logrado fusionar estas dos ciencias para la optimización del aprendizaje de los estudiantes de medicina. Con la intención de solucionar parte de este problema, surge el proyecto de investigación **LAPAROS**, el cual incluye este Trabajo Especial de Grado.

El desarrollo y la utilización, por parte de los cirujanos, de esta herramienta permitirán:

- Reducir los costos asociados con la utilización de cadáveres y animales vivos en el entrenamiento en los tipos de cirugía mencionados.

- Proveer de experiencia al cirujano con una mayor variedad de patologías y complicaciones.

- La posibilidad de repetir los ejercicios tantas veces como sea necesario hasta su correcto aprendizaje.

I.5 LIMITACIONES Y ALCANCE

I.5.1 Limitaciones

- El lenguaje de programación a utilizar en la implementación será C++.

- El desarrollo del proyecto debe regirse a los estándares de programación utilizados por los desarrolladores de software del proyecto LAPAROS.

- Todo el desarrollo de software se hará en las instalaciones del LCG de la UCV, debido a que éste es un proyecto confidencial y los encargados del mismo no permiten la salida de información fuera de sus instalaciones.

- Para el desarrollo del proyecto es necesario asistir a reuniones periódicas con ingenieros y expertos en Laparoscopia de la UCV y a reuniones semanales con los encargados del proyecto para analizar los resultados parciales y las metas semanales propuestas.

I.5.2 Alcance

- Los ejercicios a desarrollar contribuirán con el entrenamiento de cirujanos en procedimientos de Laparoscopia, específicamente a la orientación espacial y a la coordinación mano ojo. Facilitando así, el adaptarse al cambio crítico que se presenta entre la cirugía abierta y la Laparoscopia.

- Al culminar la implementación del los ejercicios se prestará entrenamiento para el uso de los mismos a los usuarios primordiales.

CAPÍTULO II. MARCO TEÓRICO

II.1 LAPAROSCOPIA

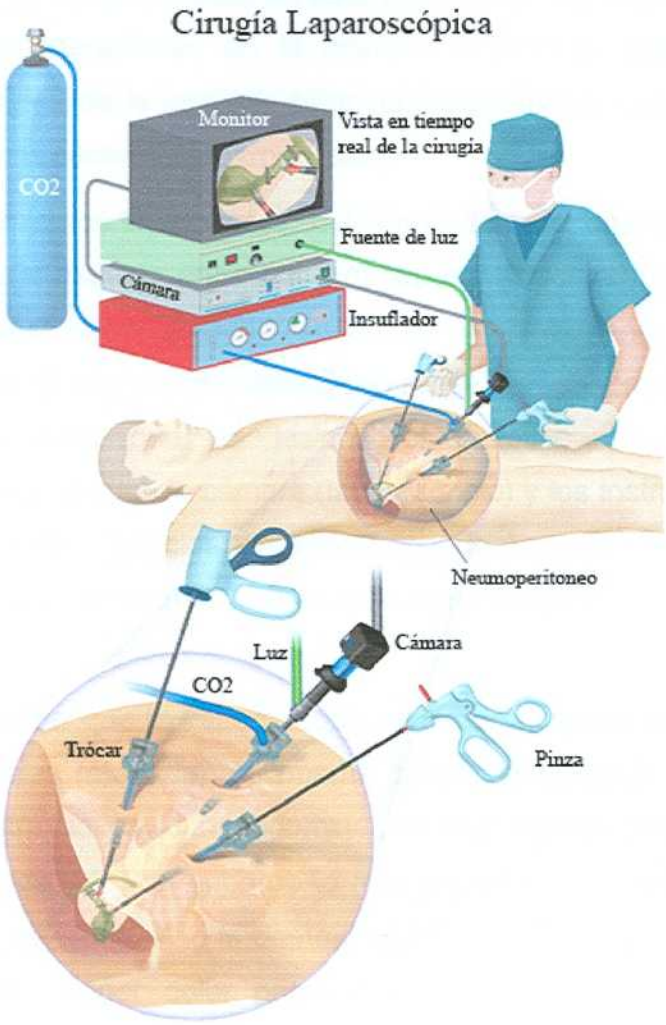

Figura 1. Cirugía laparoscópica
Fuente: Modificada por los autores [35]

El término Laparoscopia se le da a un grupo de procedimientos médicos de mínimo acceso realizados con la ayuda de una cámara, la cual es introducida en el organismo, específicamente en la cavidad abdominal. Originalmente, la Laparoscopia se usó para procedimientos ginecológicos como lo es la ligadura de trompas [20].

En la actualidad existen diversos usos para esta técnica, uno de ellos es su utilización como procedimiento de diagnóstico (biopsias, observación de cáncer en la cavidad, traumatismo del abdomen, etc.), para visualizar traumas o casos de enfermedades en esta zona. Otro uso muy difundido es la realización de operaciones en la cavidad abdominal, tales como, la extirpación quirúrgica de la vesícula biliar (colecistectomía laparoscópica) y del apéndice (apendicectomía laparoscópica).

Los procedimientos laparoscópicos (diagnóstico y operación) se inician con el aumento del espacio entre los órganos del abdomen, creado por la insuflación de gas en la cavidad abdominal (generalmente dióxido de carbono), a través de una aguja especial (aguja de Veress) protegida con un mecanismo que evita que se lesione el intestino u otra víscera. Posteriormente se introduce la cámara (laparoscopio) y los instrumentos que se necesiten, usando orificios llamados "puertos", cuyo número varía dependiendo del procedimiento, 1 si es sólo para diagnóstico y 4 o 5 para otras operaciones [2,6].

Se debe resaltar que la Laparoscopia no es una cirugía diferente a las cirugías tradicionales (abiertas), solamente cambia el abordaje, es decir, la manera de penetrar la cavidad abdominal, lo cual reduce ampliamente el tamaño de la herida [6]. Sin embargo lo más importante de esta técnica no son las ventajas estéticas, sino la demostrada menor agresividad respecto a la cirugía abierta (menor dolor postoperatorio, menores complicaciones, mayor bienestar postoperatorio, menor tiempo de hospitalización, etc.).

Algunos de los elementos que se incluyen en la Laparoscopia son los siguientes [10]:

- Producción de Imagen: gran parte del éxito de una Laparoscopia depende de una adecuada visualización del campo operatorio. El laparoscopio estándar varía en tamaño de 5 a 10 mm con un ángulo

de vista oblicuo de 30° y 45°. Actualmente constan de un tubo que recubre un sistema de fibra de vidrio que transporta la luz procedente de la fuente generadora, y la emiten al campo quirúrgico. La imagen iluminada es interpretada por una cámara colocada al extremo externo del laparoscopio, y de ahí se regenera en un monitor de alta resolución.

Figura 2. Torre de Laparoscopia producción de imagen.
Fuente: [29]

- Accesos Peritoneales: para la cirugía de mínima invasión se logra el acceso intra-abdominal mediante la técnica cerrada con aguja de Veress[1], a través de la cual se insufla el CO2.

- Insuflación de Gas: al lograr el acceso intra-abdominal se crea el campo operatorio, la forma tradicional es mediante la creación del neumoperitoneo que no es más que la insuflación de gas a través de un accesos peritoneal [17], que permite crear un espacio en el abdomen para efectuar los procedimientos quirúrgicos. La cirugía laparoscópica requiere de la insuflación de gas para la visualización, exposición y manipulación de los instrumentos. En la gran mayoría de los casos se utiliza el CO2, no sólo porque es más fácil de adquirir, sino porque tiene un bajo costo y es de fácil manejo.

[1] Aguja de seguridad de punta roma con resorte de retracción que dispone de un dispositivo tipo muelle que cubre el filo de la misma una vez se encuentra en la cavidad abdominal.

- Trócares: luego de la insuflación de gas se introducen los trócares, que son tubos cortos y delgados, a través de los cuales se insertan instrumentos largos y angostos con los cuales el cirujano manipula los órganos. Los trócares se emplazan en forma triangular y con una distancia entre sí no menor de 8 o 10 cm para evitar bloqueos innecesarios entre los instrumentos. En general el número de trócares varía de 3 a 5.

- Instrumentación: los instrumentos quirúrgicos utilizados en una cirugía laparoscópica básicamente son pinzas, tijeras, láser, engrapadoras, ligaduras, porta agujas, todo depende del procedimiento quirúrgico a realizar.

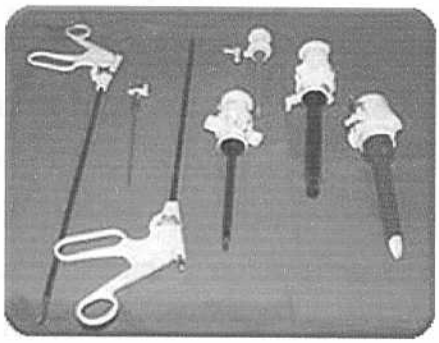

Figura 3. Instrumentos laparoscópicos.
Fuente: [29]

A pesar de los beneficios de la cirugía mínima invasiva, ésta presenta algunos problemas [5,9]:

- Los instrumentos reducen la destreza, eliminan la sensación táctil y reducen la retroalimentación de las fuerzas, ocasionando disminución en la percepción del cirujano.

- En comparación con la cirugía abierta, el cirujano ve e interactúa con la anatomía desde una perspectiva totalmente diferente.

- Los objetos y órganos se encuentran amplificados a la vista.
- Desenvolverse en un espacio tridimensional, observándolo desde un monitor que despliega una imagen en tan sólo 2 dimensiones.
- La imagen se presenta con distorsiones, limitaciones en cuanto a resolución, contraste y color.

II.2 REALIDAD VIRTUAL

La imagen clásica invocada por el término realidad virtual es la de una persona interactuando con una computadora a través de un casco y un guante sensitivo. Son muchas y diversas las definiciones dadas para la realidad virtual, entre ellas podemos mencionar:

Un ambiente gráfico tridimensional (también llamado mundo virtual) creado por computadora, donde el o los usuarios pueden interactuar con el ambiente y los objetos gráficos que contiene el mundo virtual, utilizando técnicas de manipulación directa, y obteniendo información visual, táctil y auditiva principalmente, pudiendo usarse más de un sentido a la vez [32].

La realidad virtual se refiere a las simulaciones en un ordenador del mundo real por medio de imágenes tridimensionales y componentes externos como un casco para permitir que los usuarios interactúen con la simulación. Los usuarios se mueven por una realidad virtual como si estuviesen en un mundo real [34].

Una manera más sencilla de describir la realidad virtual sería como la manipulación de los sentidos humanos (hasta el momento el tacto, la visión y la audición) por medio de ambientes tridimensionales ofrecidos por la computadora en el que uno o varios participantes interactúan de manera rápida e intuitiva de manera tal que el usuario toma como real el entorno generado por la computadora.

La realidad virtual puede tener diferentes grados de "inmersión" (sensación de estar dentro del mundo virtual) dependiendo de los dispositivos de entrada y salida utilizados.

II.2.1 Características de un sistema de realidad virtual [41]

- Interacción: le permite al usuario manipular los datos dentro de un sistema de realidad virtual, permitiendo que éste responda a los estímulos de la persona, creando interdependencia entre ellos.

- Inmersión: es la capacidad del sistema de realidad virtual de enfocar la atención del usuario, convirtiendo una base de datos en experiencias. Lo que se pretende es sustituir la percepción humana, por una percepción generada por la computadora.

- Multisensorialidad: se refiere a la integración de sonidos, imágenes y sensaciones hápticas de forma armónica.

II.2.2 Hardware especializado para un sistema de realidad virtual [41,42]

Los sistemas de realidad virtual, en su propósito de lograr una inmersión total por parte del usuario, hacen uso de hardware especializado, dentro de los más comunes se pueden mencionar:

- Casco Estereoscópico: proyecta escenas, sonidos y determina posición y movimiento de la cabeza del usuario.

- Lentes Estereoscópicos: reproducen imágenes de alta calidad con profundidad y tridimensionalidad.

- Guante Electrónico: utilizado para la manipulación del entorno virtual y para proporcionar la sensibilidad al tacto.

- Computadora con alta capacidad de procesamiento: simula los procesos asociados al entorno virtual.

- Trajes: en la actualidad tienden a ser una prenda completa que otorga las características ofrecidas por los dispositivos ya mencionados.

II.2.3 Aplicaciones de la realidad virtual [40,41]

Debido a todo el potencial de la realidad virtual, son muchas las aplicaciones que a ésta se le han dado, algunas de ellas son:

- Arquitectura: se realizan diseños y recorridos de modelos arquitectónicos, mostrando una representación casi exacta de obras aún antes de su construcción.

- Música: un sonido en particular puede ser representado por algún objeto que lo genere a través de la interacción con el usuario. Esto le permite al usuario componer sus propias melodías a través de la combinación de sonidos y practicar las veces que éste lo considere necesario.

- Psicología: son muchos los casos de éxitos que se conocen por el control de fobias como la claustrofobia o el temor a las alturas, mediante la utilización de sistemas de realidad virtual, donde el paciente toma el control del ambiente que lo rodea, enfrentando sus miedos.

- Química: existen herramientas que permiten diseñar e interactuar con compuestos orgánicos así como también el análisis molecular de nuevos compuestos.

- Medicina: la realidad virtual es ampliamente utilizada en este campo, es de gran valor en prácticas de cirugías donde se requiere un alto grado de destreza, así como también se ha explorado en el mundo de los discapacitados, como por ejemplo el desarrollo de guantes electrónicos que posibilitan la traducción de lenguajes de señas a lenguaje verbal.

- Capacitación y entrenamiento de personal: aquel tipo de actividades que para su realización requieren de coordinación motora pueden beneficiarse de la realidad virtual, ya que con su uso, es posible evaluar si un movimiento se mantiene dentro de la trayectoria correcta, y si la fuerza ejercida es la adecuada.

II.3 ENTRENAMIENTO VIRTUAL [8]

El entrenamiento es un elemento fundamental en cualquier tarea especializada. En el terreno de la cirugía, dicho entrenamiento no sólo es fundamental sino crítico, dado que cualquier fallo ocasionado por la inexperiencia del cirujano puede ser fatal para el paciente. Hasta ahora, el entrenamiento del cirujano se lleva a cabo mediante la utilización de cadáveres, maniquíes, animales vivos o en intervenciones quirúrgicas reales bajo la supervisión de un experto.

En este tipo de entrenamiento tradicional pueden surgir dos problemas obvios: en primer lugar, la falta de realismo en las prácticas realizadas sobre maniquíes y cadáveres, dado que el comportamiento de los tejidos en seres vivos es muy distinto al que presentan los elementos de entrenamiento utilizados; por otro lado, en los entrenamientos realizados sobre animales

vivos o en intervenciones reales de pacientes, está limitada tanto la capacidad de repetir la intervención hasta su correcto aprendizaje, como la diversidad de intervenciones en la que los cirujanos pueden llegar a entrenarse, sin contar los problemas morales que conlleva la utilización de animales vivos en experimentación.

En las últimas décadas la percepción que tenemos de nuestro entorno ha sido transformada por el progreso de la tecnología, además de revolucionar al mundo, no sólo de la informática sino también de una gran diversidad de áreas como la educación, la medicina, la arquitectura, las ingenierías, etc. En efecto, la velocidad con que evolucionan las nuevas tecnologías impone a las instituciones educativas el reto de actualizarse con equipos modernos y programas que favorezcan la creación de nuevos ambientes de aprendizaje y con ello incidir en una mejor preparación de sus estudiantes.

Llegar a simular un procedimiento quirúrgico implica la necesidad de que este proceso sea lo más real posible, no basta con que la manipulación ocurra en tiempo real, sino que además el entorno parezca real. Actualmente, las aplicaciones potenciales de esta nueva tecnología están siendo exploradas en la cirugía mínima invasiva y en la cirugía por telepresencia (telecirugía).

II.4 EJERCICIOS TRADICIONALES PARA EL ENTRENAMIENTO EN CIRUGIA LAPAROSCÓPICA

El entrenamiento en cirugía laparoscópica no es una tarea fácil ya que la complejidad que conlleva obtener la habilidad de realizar este procedimiento que se aleja tanto de la cirugía tradicional, no resulta sencillo.

En las instalaciones de la Escuela de Medicina de la Universidad Central de Venezuela, como ya ha sido mencionado antes, el entrenamiento

en Laparoscopia se lleva a cabo previamente con animales. Actualmente cuentan con un simulador mecánico construido por el Centro de Bioingeniería [4] de la misma institución, denominado por ellos "*Caja Negra*", constituido por una caja que simula la forma del abdomen y una cámara conectada a un monitor, donde practican el manejo y dominio de instrumentos y de la propia cámara, para luego someterse al entrenamiento con seres vivos.

Son muchos los simuladores que se encuentran a la venta en el mercado que cuentan con un conjunto de ejercicios que permiten desarrollar desde las habilidades más básicas, hasta el hecho de realizar procesos quirúrgicos completos que se asemejen bastante a la realidad [4].

En base a toda la información recolectada tanto con el personal de la UCV como el ofrecido por los simuladores comerciales, los ejercicios que forman parte del entrenamiento en Laparoscopia pueden ser clasificados de acuerdo a las destrezas y habilidades que permiten desarrollar:

Clasificación	Habilidades y destrezas a desarrollar
Nivel Básico	Percepción y orientación espacial.
	Coordinación mano - ojo, uso de la mano dominante y no dominante.
	Habilidad eje mano – ojo inverso, donde la persona debe coordinar sus movimientos en un sentido y lo que visualiza va en sentido contrario.
Nivel Medio	Navegación de la cámara.
	Utilización de instrumental laparoscópico
	Realización de suturas y nudos.
Nivel Avanzado	Aprendizaje de técnicas de cirugía mínima invasiva, tales como: separar, cortar, cauterizar y suturar.

Tabla 1. Clasificación de ejercicios para entrenamiento en Laparoscopia
Fuente: Elaboración Propia

II.5 OPENGL®

OpenGL® (*Open Graphics Library*) es una librería multiplataforma para programación de gráficos tridimensionales o bidimensionales. El grupo de

funciones que ésta presenta fue definido originalmente por Silicon Graphics en 1992, basándose en su anterior librería: IRIS GL que hizo tan popular las estaciones de trabajo SGI como plataforma predilecta para desarrollo científico, de ingeniería y de efectos especiales. En la actualidad participan en su estandarización múltiples compañías (Intel, SGI, Evans & Sutherland, nVidia...) mediante el consorcio independiente *OpenGL Architecture Review Board.* [23]

Esta interfaz consiste en alrededor de 250 comandos distintos (unos 200 dentro del núcleo de OpenGL® y otros 50 en la librería de utilidades de OpenGL®) que pueden ser usados para especificar objetos y operaciones necesarias para producir aplicaciones tridimensionales interactivas. [2]

OpenGL® es un lenguaje procedimental más que un lenguaje descriptivo. En lugar de diseñar la escena y su apariencia, el programador describe los pasos a seguir para lograr lo que desea. En esta librería no se incluye ninguna función para la gestión de ventanas, interacción hombre-máquina y manejo de archivos, dejando estos tres aspectos dependientes de la plataforma para lograr así una mayor portabilidad.

Algunas de las características que OpenGL® implementa [23]:

- Primitivas geométricas: permiten construir descripciones matemáticas de objetos. Las actuales primitivas son: puntos, líneas, polígonos, imágenes y bitmaps.

- Codificación del Color: en modos RGBA (Rojo-Verde-Azul-Alfa) o de color indexado.

- Visualización y modelado: permite disponer objetos en una escena tridimensional, mover la cámara por el espacio y seleccionar la

posición ventajosa deseada para visualizar la escena de composición.

- Mapeado de texturas: que ayuda a traer realismo a los modelos por medio del dibujo de superficies realistas en las caras de los modelos poligonales.

- Iluminación de materiales: provee comandos para calcular el color de cualquier punto, dadas las propiedades del material y las fuentes de luz en la habitación.

- Doble buffering: ayuda a eliminar el parpadeo de las animaciones. Cada fotograma consecutivo en una animación se construye en un buffer separado de memoria y mostrado solo cuando está completo.

- Anti-aliasing: reduce los bordes escalonados en las líneas dibujadas sobre una pantalla. El anti-aliasing es una técnica común en gráficos computarizados que modifica el color y la intensidad de los pixels cercanos a la línea para reducir el zig-zag artificial.

- Sombreado Gouraud: es una técnica usada para aplicar sombreados suaves a un objeto 3D y producir una sutil diferencia de color por sus superficies.

- Z-buffering: se usa para registrar la proximidad de un objeto al observador, y también sirve para eliminar superficies ocultas.

- Efectos atmosféricos: como la niebla, el humo y las neblinas hacen que las imágenes producidas por computadora sean más realistas.

- Planos de plantilla: permiten restringir el trazado a ciertas regiones de la pantalla.

- Listas de display: permiten almacenar comandos de dibujo en una lista para un trazado posterior.

- Primitivas de Raster: bitmaps y rectángulos de pixels.

- Transformaciones: rotación, escalado, perspectivas en 3D.

II.6 DIRECTX® [31]

DirectX® es una API (Interfaz de Programación de Aplicaciones) producida por Microsoft, para ayudar a los desarrolladores de juegos y aplicaciones, que manejen gráficos 2D y 3D, a programar eficientemente bajo Windows, lo cual, era improbable antes de la llegada de éste, ya que las tecnologías anteriores no permitían acceder al hardware de video tan directamente como DirectX® lo hace.

II.6.1 Ventajas de DirectX®

- Acceso directo al hardware: muy importante en el caso de la tarjeta de video.

- Rapidez: permite obtener un alto rendimiento del hardware, inclusive, es posible aprovechar mejoras en el hardware que podrían aparecer después de desarrollada la aplicación.

- Facilidad: DirectX® facilita las tareas de configuración del tipo de video, profundidad de colores, reproducción de sonidos, etc.

II.6.2 Componentes de DirectX®

- DirectDraw: es una librería de bajo nivel contenida en su mayoría en el archivo ddraw.dll, que permite el acceso al hardware de video

directamente. Además, a través de DirectDraw se pueden implementar varias técnicas para el manejo de la memoria, intercambio de páginas, sobreimpresiones, etc.

- Direct3D: es una librería que sirve para optimizar la renderización de objetos 3D tomando las ventajas del hardware existente, o mediante software.

- DirectSound: como su nombre lo indica, la librería de DirectSound maneja todo lo relacionado con el sonido. Proporcionando tecnologías de mezcla de sonidos, sonido estéreo y 3D, aprovechando al máximo las capacidades del hardware.

- DirectPlay: se encarga de las opciones de multijugador de los juegos.

- DirectInput: proporciona una interfaz para el manejo de dispositivos de entrada y salida.

- DirectSetup: se utiliza para automatizar la instalación de los controladores DirectX®.

Para el desarrollo de este trabajo se decidió utilizar la librería OpenGL® por su portabilidad, y principalmente debido a que el LCG cuenta con una gran cantidad de expertos en esta librería, quienes han sido de gran ayuda para el desarrollo de este Trabajo Especial de Grado.

II.7 SIMULADORES LAPAROSCÓPICOS

II.7.1 LTS-2000 [11]:

Figura 4. Modelo del simulador laparoscópico LTS-2000
Fuente: [11]

Este simulador permite el entrenamiento de procedimientos laparoscópicos en condiciones similares a las reales. El LTS-2000 simula las dimensiones de un abdomen expandido y está cubierto por un marco removible que contiene una estructura hecha de membranas multicapilares que asemejan la pared abdominal anterior.

El simulador LTS-2000 viene con un reloj cronómetro, para medir el tiempo usado durante los ejercicios, y postes externos de metal para practicar nudos. El simulador puede trabajar con una video cámara que permite una imagen estable y magnificada del interior de la caja (simulador), de esta manera el sistema es transportable y fácil de utilizar ya que no depende de los equipos de sala de operaciones; pero también puede ser usado con un laparoscopio estándar conectado a su correspondiente cámara de video, luz y monitor. En este último caso, el laparoscopio es introducido a *través de una cánula convencional insertada a través de la simulada pared* abdominal.

II.7.3 ProMIS™ [16]:

Figura 6. Modelo del simulador laparoscópico ProMIS
Fuente: [16]

Este simulador presenta 6 módulos: orientación en Laparoscopia, manejo del instrumental, disección, sutura y anudado intracorpóreo, diatermia[2] y ultrasonido.

Cada módulo presenta tareas con tres niveles complejidad progresiva, y se analiza la tasa de errores, velocidad, movimientos en tiempo real.

Los ejercicios utilizan realidad virtual u objetos físicos, ofreciendo la misma libertad de una práctica quirúrgica, los usuarios pueden crear todos los portales que deseen en el abdomen y mover o rotar los instrumentos en la manera en que deseen.

[2] Proceso de calentamiento del cuerpo por vía interna haciéndolo atravesar por corrientes eléctricas de alta frecuencia

II.7.4 SIMULAP v1.0 [4]:

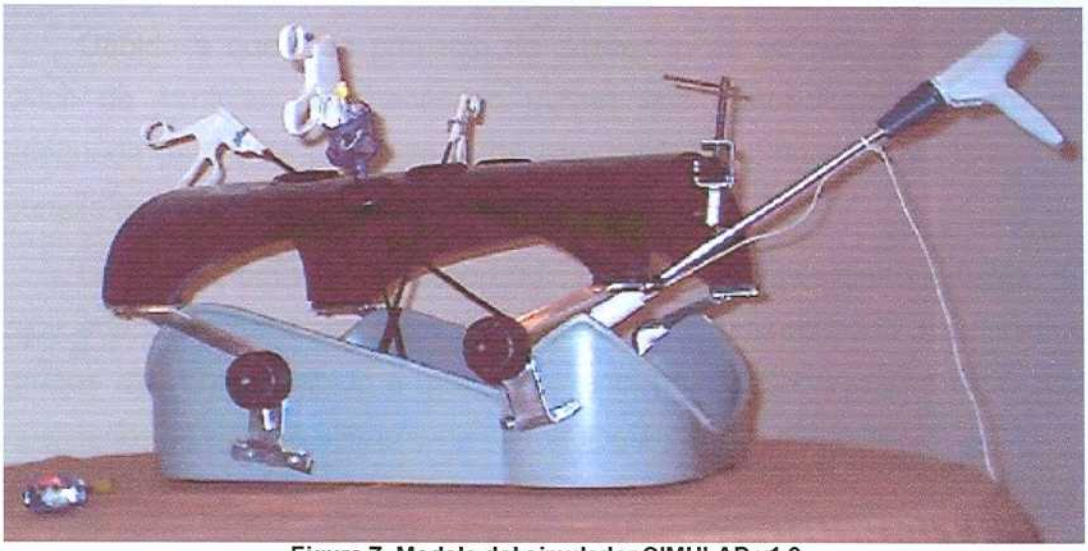

Figura 7. Modelo del simulador SIMULAP v1.0
Fuente: [4]

El simulador mecánico SIMULAP v1.0 fue diseñado y construido por el Centro de Bioingeniería de la Universidad Central de Venezuela. De acuerdo a la realidad latinoamericana el dispositivo construido es versátil y de bajo costo. Integra los componentes mínimos necesarios para que un cirujano o estudiante de medicina reciban un adecuado entrenamiento en técnicas de cirugía laparoscópica. El simulador construido permite la visualización de los procedimientos quirúrgicos en un solo conjunto, sin necesidad de inutilizar algún otro tipo de instrumento no destinado para tal fin y cuyo uso acarrearía el incremento de los costos de las prácticas.

CAPÍTULO III. MARCO METODOLÓGICO

Debido a que todo proyecto de desarrollo de software es riesgoso y difícil de controlar, es necesario utilizar una metodología de software adecuada para este propósito, y de esta manera obtener un producto que cumpla con todas las expectativas.

La metodología del Proceso Unificado (UP por sus siglas en inglés) es muy adaptable para proyectos de largo plazo y desarrollados por distintos grupos de trabajo, debido a que el proyecto LAPAROS presenta estas dos cualidades se decidió utilizar esta metodología para su desarrollo.

III.1 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

El Proceso Unificado es un proceso de desarrollo de software configurable y adaptable a proyectos de cualquier tamaño y complejidad. Se encuentra basado en muchos años de experiencia en el uso de la tecnología Orientada a Objetos para el desarrollo de software de misión crítica en una variedad de industrias. Para lograr el nacimiento de la metodología confluyeron "los tres amigos", como se llaman a sí mismos los tres grandes de la Programación Orientada a Objetos (POO): Grady Booch, James Rumbaugh e Ivar Jacobson [12].

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo mientras se balancean los requerimientos del negocio, el tiempo de producción y los riesgos del proyecto. Éste describe los diversos pasos necesarios para realizar la captura de los requerimientos y el establecimiento de una guía arquitectónica, para diseñar y probar el sistema. El proceso describe qué productos generar y cómo desarrollarlos [27].

III.1.1 Características del Proceso Unificado [2,15]

Según Booch, las características del Proceso Unificado son:

III.1.1.1 Iterativo e Incremental

Debido al gran tamaño que presentan los proyectos de desarrollo de software, el Proceso Unificado propone que éstos sean divididos en miniproyectos. Cada miniproyecto es una iteración, en la cual se desarrolla un grupo de casos de uso. Al realizar cada una de estas iteraciones se crean productos a partir de las iteraciones anteriores lo cual es en definitiva un incremento en el proyecto.

III.1.1.2 Centrado en la Arquitectura [1]

Los casos de uso, aunque muy útiles, por si solos no son suficientes. Se necesita algo más para llegar a un sistema que funcione: la arquitectura.

El papel que juega la arquitectura de software es similar al de la arquitectura en la construcción de edificios. Por lo tanto con la arquitectura de software el diseñador logra observar los distintos puntos de vista del sistema, antes de que sea desarrollado.

La arquitectura es una visión global del sistema pero sin detalles. El diseñador debería tener como objetivos la comprensión del sistema, capacidad de adaptación a cambios futuros y reutilización.

III.1.1.3 Guiado por Casos de Uso

Los Casos de Uso son los requerimientos funcionales del sistema. El modelo de casos de uso, el cual engloba todos los casos de uso del sistema, describe la funcionalidad completa del sistema. Pero éstos no son sólo una herramienta para especificar los requerimientos del sistema, sino que

también guían su diseño, implementación y prueba, es decir, guían el proceso de desarrollo del software.

Los desarrolladores crean modelos de diseño e implementación que realizan los casos de usos, luego los revisan para comprobar si se corresponden con el modelo de casos de uso especificado. Igualmente los responsables de las pruebas del sistema comprueban que se hayan implementado correctamente los casos de uso. En este sentido los casos de uso no sólo inician el proceso de desarrollo del software sino que permanecen ligados a él durante todo el ciclo de vida.

III.1.2 Fases del Proceso Unificado [14]

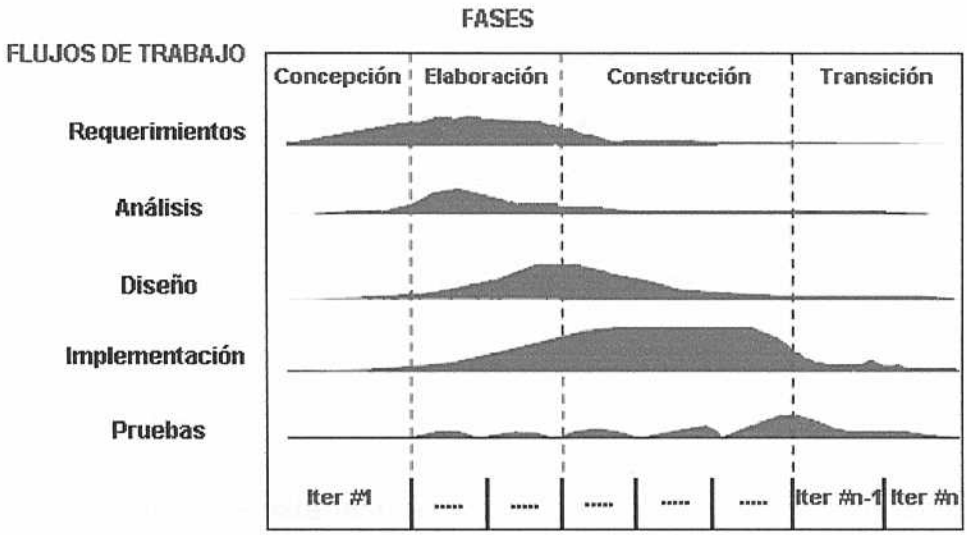

Figura 8. Fases y flujos del Proceso Unificado
Fuente: Modificada por los Autores [13]

III.1.2.1 Concepción

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

III.1.2.2 Elaboración

Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él. Al final de la fase se determina la viabilidad de continuar el proyecto y si se decide proseguir, dado que la mayor parte de los riesgos han sido mitigados, se escriben los planes de trabajo de las etapas de construcción y transición y se detalla el plan de trabajo de la primera iteración de la fase de construcción.

III.1.2.3 Construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto.

III.1.2.4 Transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto al inicio del mismo.

III.1.3 Flujos de Trabajo del Proceso Unificado

III.1.3.1 Requerimientos [7]

Son las capacidades y condiciones con que debe cumplir el sistema, éstos se pueden dividir en:

- Requerimientos Funcionales: definen las funciones que el sistema debe realizar y describen los procesos que el sistema ejecuta sobre las entradas para obtener las salidas.
- Requerimientos no Funcionales: son las características o propiedades que de alguna forma limiten el sistema, como pueden ser, el rendimiento, las interfaces de usuario, la fiabilidad, seguridad, portabilidad, estándares, etc.

III.1.3.2 Análisis [1]

Al finalizar el flujo de captura de requerimientos se inicia el flujo del análisis, en donde se refinan y estructuran todos los requerimientos del sistema, éste facilita la comprensión, preparación y modificación de los requerimientos

En el análisis se describen los requerimientos en el lenguaje del desarrollador utilizando un mayor nivel de formalismo, este flujo puede ser utilizado para razonar sobre el funcionamiento interno del sistema.

III.1.3.3 Diseño [1]

En el diseño se lleva a cabo el modelado del sistema para que soporte todos los requerimientos que fueron capturados y analizados anteriormente.

Los propósitos esenciales del diseño son:

- Obtener una comprensión más profunda de los aspectos relacionados con los requerimientos no funcionales y restricciones con la plataforma tecnológica a utilizar.
- Capturar los requerimientos o subsistemas individuales, las interfaces y las clases a implementar.
- Descomponer en partes manejables los trabajos de implementación, teniendo la posible concurrencia entre ellos.

III.1.3.4 Implementación [1]

Con los resultados del diseño como entrada se comienza el flujo de implementación, en el cual se crea el sistema en términos de componentes: archivos de código, scripts, ficheros de código binario, ejecutables y similares.

Los propósitos de la implementación son:

- Planificar la integración de sistemas a ser creados en cada iteración.
- Implementar las clases y subsistemas diseñados anteriormente.
- Probar cada uno de los componentes por separado, para luego integrarlos en un solo sistema.

III.1.3.5 Pruebas [1]

Luego de finalizar la implementación del sistema es necesario someterlo a un conjunto de pruebas para así verificar los resultados obtenidos en ésta.

Los objetivos principales de la realización de pruebas son:

- Planificar las pruebas en cada iteración, tanto las de integración como las de sistema.
- Diseñar y crear los casos de prueba que definen qué se va a probar.
- Aplicar las pruebas y administrar los resultados de cada prueba sistemáticamente.

III.2 ENTREGABLES GENERADOS

En cada flujo de trabajo se generaron distintos tipos de entregables, algunos de éstos se realizaron utilizando el Lenguaje Unificado de Modelado (Ver Anexo A). En el siguiente cuadro se encuentra los entregables generados durante cada uno de los flujos de trabajo.

Flujo de trabajo	Entregable
Requerimientos	Diagrama de casos de uso(UML)
	Casos de uso
Análisis	Diagramas de actividades(UML)
Diseño	Diagrama de clases(UML)
Implementación	Código Fuente
Pruebas	Pruebas de caja negra
	Pruebas funcionales

Tabla 2. Entregables
Fuente: Elaboración Propia

CAPÍTULO IV. DESARROLLO

IV.1 CAPTURA DE REQUERIMIENTOS

Esta etapa que corresponde con el inicio de este Trabajo Especial de Grado, se hizo en conjunto con el personal del Laboratorio de Computación Gráfica y de la Facultad de Medicina de la Universidad Central de Venezuela, y es debido a esto que se contó con la posibilidad de capturar gran parte de los requerimientos directamente de los profesores del postgrado de cirugía laparoscópica de esta Alma Mater.

En un principio se buscó información en diversas fuentes acerca de la cirugía mínima invasiva, obteniendo con esto un conocimiento básico de todo lo relacionado a ella. Luego se realizaron una serie de reuniones con los expertos en cirugía laparoscópica de la Facultad de Medicina de las cuales se obtuvo información de los siguientes tópicos:

- Las ventajas que obtiene un paciente con una intervención de este tipo, pero a la vez la dificultad que le produce al cirujano enfrentarse con un ambiente mostrado en un monitor que produce imágenes en sólo 2 dimensiones.

- Los métodos empleados por los profesores que dictan el postgrado de Laparoscopia para el entrenamiento de sus alumnos.

- Instrumentos utilizados en esta técnica quirúrgica.

Además de toda esta información, fueron proporcionadas las necesidades y expectativas de los médicos con respecto al simulador virtual a desarrollarse, entre las cuales se pueden mencionar:

- Debe ser una aplicación en la cual el cirujano pueda realizar ejercicios que contribuyan a su entrenamiento en las distintas técnicas laparoscópicas y al mejoramiento de sus habilidades en el uso de los instrumentos utilizados en estas técnicas.

- Los ejercicios desarrollados para éste deben tener diferentes niveles de dificultad, incluyendo el hecho de utilizar la mano no dominante en los ejercicios para poder realizar una evaluación más completa.

- Se deben utilizar 3 niveles para clasificar los ejercicios: básico, medio y avanzado, en los cuales se desarrollen habilidades específicas (Ver Capítulo II).

Luego de conocer los lineamientos a seguir para desarrollar el simulador, se prosiguió a investigar sobre las posibles plataformas en que podía ser realizado un proyecto de este tipo.

Los motores 3D que fueron parte de la investigación son OpenGL® y DirectX®, el primero es multiplataforma, de licencia gratuita para desarrolladores y además es el utilizado por los mismos desarrolladores del LCG, y el segundo trabaja sólo bajo ambientes Microsoft Windows y es de licencia comercial.

En cuanto a la plataforma de hardware, será utilizado el simulador mecánico SIMULAP v1.0 (Ver Capítulo II), incluido como parte esencial del proyecto LAPAROS, el cual actualmente es utilizado para las prácticas de los cirujanos en el postgrado de Laparoscopia.

Con base a toda esta información se procede a determinar, en sucesivas reuniones con el personal del LCG, incluyendo a la tutora del presente trabajo, cuáles serían los requerimientos funcionales a desarrollar. Se propuso desarrollar un conjunto de ejercicios que estuviesen clasificados dentro de los niveles básico y medio (Ver Capítulo II), que pudiesen ser

fácilmente adaptados a la estructura diseñada para el proyecto LAPAROS, tanto a nivel de hardware como de software. Es importante resaltar que estos ejercicios deben ser desarrollados dentro de un ambiente que simule prácticas del tipo denominado "Caja Negra", es decir, el usuario utilizará instrumentos laparoscópicos dentro de un simulador mecánico, y verá reflejado en la pantalla del computador el simulador, los instrumentos y los diversos elementos que formen parte del ejercicio a realizar.

Para conocer de primera mano como es un entrenamiento en Laparoscopia y los diferentes ejercicios que ellos realizan, se ingresó a una serie de clases de Postgrado en Laparoscopia dictadas en la Universidad Central de Venezuela, donde los autores de este trabajo asistieron en calidad de alumnos.

En primer lugar se practicó la correcta utilización de pinzas laparoscópicas. Éstas se utilizaron para agarrar objetos (cuentas) con total libertad para acostumbrarse al uso de las mismas.

Luego en grupos de 2 alumnos se practicó con el simulador mecánico (SIMULAP v1.0), uno que utilizara las pinzas y otro que manejara la cámara conectada a un monitor (como en una cirugía real). Se practicaron 2 tipos de ejercicios, el primero fue apilar cuentas en grupos de a 3 y el segundo consistía en introducir las cuentas en pequeños alambres verticales fijos.

Finalmente los profesores proporcionaron videos de entrenamiento en operaciones reales.

Como complemento se hizo una revisión de distintos simuladores comerciales existentes en el mercado. Cada uno contiene un grupo de ejercicios que fueron analizados con la finalidad de diseñar el conjunto de ejercicios que forman parte de este trabajo.

VI.1.1 Ejercicios Planteados

Los ejercicios que fueron planteados para el desarrollo de este proyecto son los siguientes:

VI.1.1.1 Tocar los colores

Ejercicio ideado por los autores de este trabajo en conjunto con integrantes del proyecto LAPAROS.

Nivel de dificultad: Básico.

Habilidades y destrezas a desarrollar:

- Percepción y orientación espacial.
- Coordinación mano – ojo.
- Uso de la mano dominante y no dominante.

Descripción:

Utilizar la pinza de Laparoscopia para tocar secciones específicas indicadas por colores. El usuario hará repetidamente esta acción.

Requerimientos:

- Debe poseer la capacidad de modificar el tiempo que tiene el usuario para tocar una sección coloreada.
- El número de repeticiones debe ser modificable.
- La secuencia en que el sistema coloree las secciones debe ser aleatoria.

VI.1.1.2 Seguir el Camino

Ejercicio ideado por los autores de este trabajo en conjunto con integrantes del proyecto LAPAROS.

Nivel de dificultad: Básico.

Habilidades y destrezas a desarrollar:

- Percepción y orientación espacial.
- Coordinación mano – ojo.
- Uso de la mano dominante y no dominante.

Descripción:

Utilizar la pinza de Laparoscopia para seguir un camino que será dibujado en la escena del ejercicio. El usuario deberá culminar el recorrido completo del camino.

Requerimientos:

- Debe poseer la capacidad de modificar el largo del camino.
- El camino debe ser aleatorio.

VI.1.1.3 Pasar a través de los Aros

Ejercicio basado en otro perteneciente al simulador LTS 2000 (Ver Capítulo II) y adaptado al sistema virtual por los autores de este trabajo en conjunto con integrantes del proyecto LAPAROS.

Nivel de dificultad: Básico.

Habilidades y destrezas a desarrollar:

- Percepción y orientación espacial.
- Coordinación mano – ojo.
- Uso de la mano dominante y no dominante.

Descripción:

Utilizar la pinza de Laparoscopia para atravesar una sucesión de aros colocados en la escena tratando de no tocar el borde de los mismos.

Requerimientos:

- La disposición de la escena deber ser modificable para permitir la realización del ejercicio tanto con la mano derecha como la izquierda.

VI.1.1.4 Colocar las Figuras

Ejercicio ideado por los autores de este trabajo en conjunto con integrantes del proyecto LAPAROS.

Nivel de dificultad: Básico.

Habilidades y destrezas a desarrollar:

- Percepción y orientación espacial.
- Coordinación mano – ojo.
- Uso de la mano dominante y no dominante.

Descripción:

Tomar con la pinza de Laparoscopia una esfera y colocarla en el sitio que el ejercicio le indique.

Requerimientos:

- Debe existir un tiempo máximo para colocar la figura.
- El sitio para colocar la figura debe ser asignado aleatoriamente.

VI.1.1.5 Cortar las Cuerdas

Ejercicio ideado por los autores de este trabajo en conjunto con integrantes del proyecto LAPAROS.

Nivel de dificultad: Medio.

Habilidades y destrezas a desarrollar:

- Percepción y orientación espacial.
- Coordinación mano – ojo.
- Uso de la mano dominante y no dominante.

Descripción:

Utilizar la tijera de Laparoscopia para cortar cuerdas en la sección indicada por el ejercicio.

Requerimientos:

- El tiempo máximo para cortar la cuerda debe ser modificable.

- La región a cortar de la cuerda debe ser asignada aleatoriamente.

VI.1.2 Funciones del sistema

Las funciones necesarias para realizar estos ejercicios se describen a continuación:

#	Nombre	Descripción
1	Mover la pinza virtual	Capturar los movimientos realizados por el usuario con la pinza que forma parte del simulador mecánico y utilizarlos para mover la pinza virtual de la misma manera y reflejarlo en el monitor.
2	Tocar un objeto	Detectar la colisión ocurrida entre algún objeto de la escena y la pinza virtual mientras ésta se encuentra en movimiento.
3	Abrir/cerrar los dientes de un instrumento (pinza o tijera)	Detectar el momento en que el usuario abre/cierra el instrumento y reflejarlo en la pinza virtual sobre el monitor.
4	Tomar un objeto con la pinza virtual	Detectar el momento en que el usuario cierra la pinza en torno a un objeto tomándolo así y reflejarlo en el monitor
5	Soltar un objeto luego de haber sido tomado con la pinza virtual	Detectar el momento en que el usuario abre la pinza luego de haber tomado un objeto y reflejarlo en el monitor.
6	Cortar una línea con la tijera virtual	Detectar el momento en que el usuario corta una línea con la tijera y reflejarlo en el monitor

Tabla 3. Funciones del sistema
Fuente: Elaboración Propia

Se considera que cada uno de los ejercicios propuestos pueden ser catalogados como casos de uso, y de esta manera definir los requerimientos funcionales del sistema. Para estos casos de uso existe un actor principal, al que se le presentará la opción de realizar los ejercicios.

A continuación se presenta el diagrama de los casos de uso:

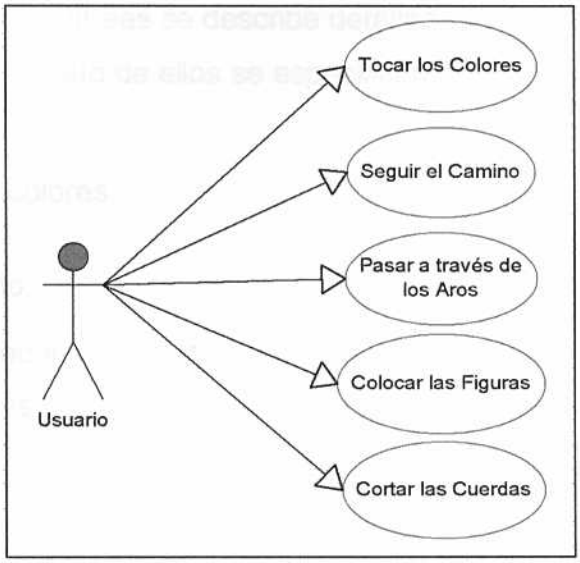

Figura 9. Diagrama de casos de uso del sistema.
Fuente: Elaboración Propia

IV.1.3 Casos de Uso

En la siguiente tabla se observa la lista de los casos de uso del sistema, su descripción y las funciones que utilizan cada uno de ellos:

#	Nombre	Descripción	Funciones requeridas
1	Tocar los Colores (Ejercicio Básico)	En este caso de uso el usuario debe utilizar la pinza de Laparoscopia para tocar secciones que le sean indicadas con colores durante el ejercicio. El usuario hará repetitivamente esta acción, durante una cantidad limitada de intentos.	1, 2,3
2	Seguir el Camino (Ejercicio Básico)	En este caso de uso el usuario debe utilizar la pinza de Laparoscopia para seguir un camino que le será dibujado en la escena del ejercicio. El usuario deberá culminar el recorrido de la línea.	1, 2,3
3	Pasar a través de los Aros (Ejercicio Básico)	En este caso de uso el usuario debe utilizar la pinza de Laparoscopia para atravesar una sucesión de aros colocados en la escena tratando de no tocar los bordes.	1,2,3
4	Colocar las Figuras (Ejercicio Básico)	En este caso de uso el usuario debe tomar con la pinza de Laparoscopia una esfera y colocarla en el sitio que el ejercicio le indique.	1,2,3,4,5
5	Cortar las Cuerdas (Ejercicio Medio)	En este caso de uso el usuario debe utilizar la tijera de Laparoscopia para cortar cuerdas en la sección indicada por el ejercicio.	1,2,3,6

Tabla 4. Casos de uso
Fuente: Elaboración Propia

En las siguientes líneas se describe detalladamente uno de los casos de uso del sistema y el resto de ellos se especifican en el Apéndice A.

#1 Tocar los colores

Actor: Usuario.

Propósito: Tocar con la pinza de Laparoscopia las secciones que le sean indicadas con colores durante el ejercicio.

Precondición: El usuario se encuentra dentro del ejercicio Tocar los Colores y se encuentra coloreada una sección.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. El usuario mueve la pinza en la dirección a la sección coloreada.	
2. El usuario toca con la punta de la pinza la sección coloreada.	3. La sección cambia de color para indicarle al usuario que fue tocada con éxito, e inmediatamente colorea otra sección del ejercicio.

Cursos alternos:

2.a: El usuario no toca la sección coloreada a tiempo.

3.a: El sistema colorea otra sección aleatoriamente y se registra un error cometido por el usuario.

2.b: El usuario toca una sección distinta a la coloreada.

3.b: El sistema colorea otra sección aleatoriamente y se registra un error cometido por el usuario.

Además de los requerimientos funcionales del sistema se pudieron capturar los requerimientos no funcionales, los cuales serán descritos a continuación:

- Portabilidad: por ser éste un sistema en el cual se encuentran involucrados muchas entidades de la Universidad Central de Venezuela, se debe pensar en las posibles migraciones futuras que el mismo pueda sufrir.

- Integración: el sistema debe ser compatible con el software y hardware desarrollado por los otros integrantes del proyecto LAPAROS.

- Mantenimiento: debido a que en este sistema no se cubre por completo el alcance del proyecto LAPAROS, el cual continúa en proceso de desarrollo, se debe pensar en las futuras mejoras, adaptaciones y adiciones que pueden ocurrir sobre el mismo en el futuro.

IV.2 ANÁLISIS

Un producto de esta fase, es la realización de diagramas de actividades correspondientes a cada uno de los casos de uso especificados en la fase anterior. A continuación se muestra el diagrama correspondiente al caso de uso #1 Tocar los Colores y el resto son especificados en el Apéndice B.

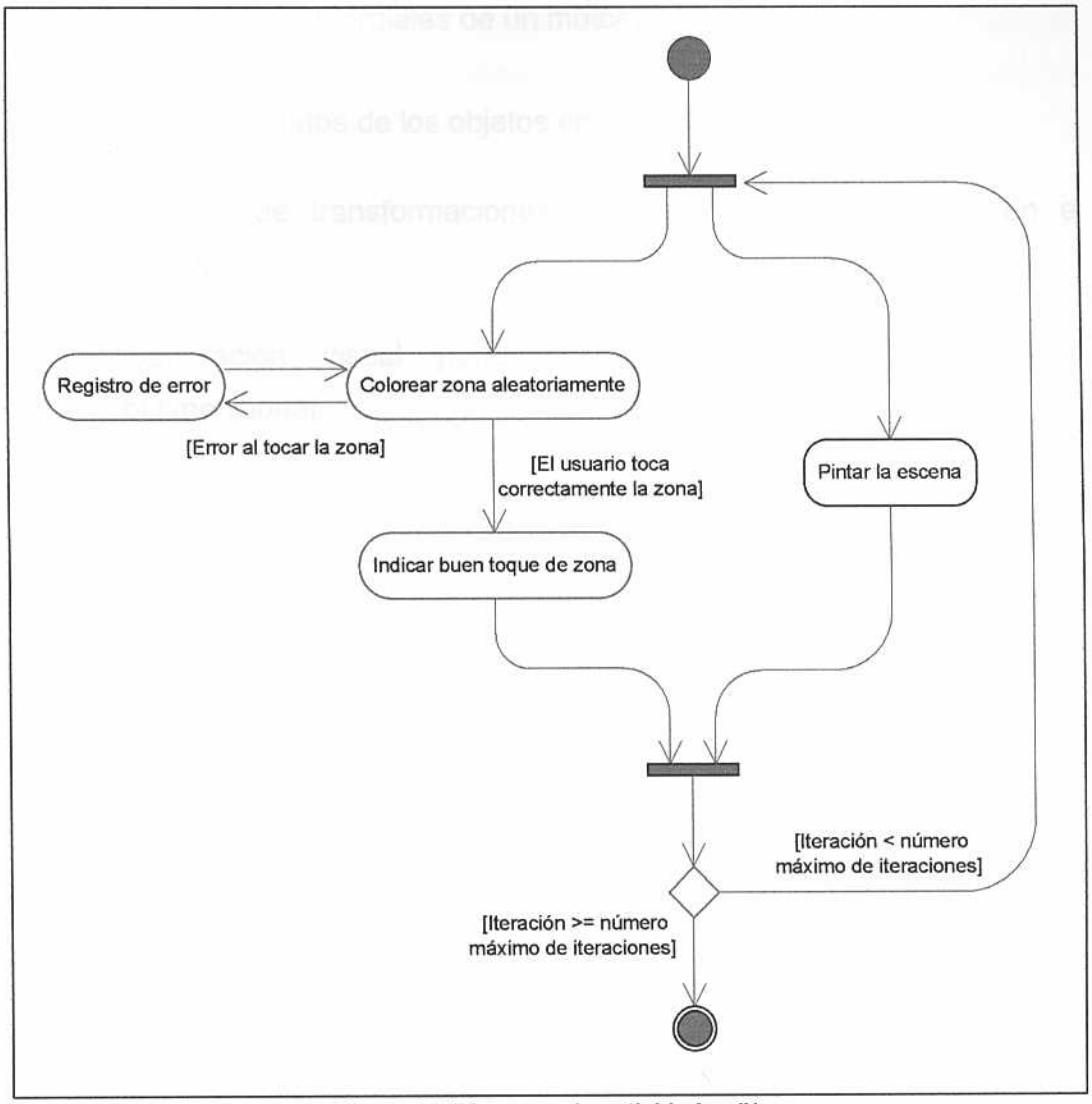

Figura 10. Diagrama de actividades #1.
Fuente: Elaboración Propia

IV.2.1 Análisis de la plataforma de Software

Las aplicaciones gráficas requieren un motor 3D para su desarrollo, éstos son una colección de estructuras, funciones y algoritmos utilizados para visualizar, después de muchos cálculos y transformaciones, objetos tridimensionales en una pantalla bidimensional [25].

Las funciones primordiales de un motor 3D son:

- Ubicar los datos de los objetos en estructuras.
- Proveer de transformaciones para posicionar los objetos en el mundo.
- Generación visual (rendering) de escenas en la pantalla bidimensional.

El motor 3D más indicado para el desarrollo del sistema es OpenGL®, por las siguientes razones:

- Alto nivel de confiabilidad y desempeño avalado por más de 12 años de existencia en el mercado.
- Debido a que los desarrolladores de aplicaciones gráficas no necesitan adquirir licencia para utilizar OpenGL®, existe una muy amplia base de conocimiento a nivel mundial, a la cual se puede recurrir en cualquier momento.
- Gracias a la portabilidad de OpenGL® (Ver Capítulo II) el sistema podrá ejecutarse en una amplia variedad de arquitecturas y de soportes gráficos, garantizando así la consistencia del mismo frente a posibles migraciones.
- Dentro del LCG se encuentra un excelente grupo de expertos en programación con este motor 3D siempre dispuestos a prestar sus conocimientos.

- El LCG cuenta con un grupo de librerías de desarrollo propio las cuales son de gran utilidad para el desarrollo de aplicaciones gráficas.

En cuanto al lenguaje de programación a utilizar para el desarrollo, fue escogido C++, por parte de los dirigentes y creadores del proyecto.

IV.2.2 Análisis de la plataforma de Hardware

El sistema, por ser una aplicación gráfica, requiere de un computador que a nivel de hardware soporte todas las características que un motor 3D requiere para su óptimo funcionamiento, entre las cuales se pueden mencionar:

- Tarjeta gráfica con soporte para OpenGL®.
- Dispositivo de realidad virtual con comunicación igual al del IsotrakII (Ver Anexo F).

Para el proyecto LAPAROS, está planteado el desarrollo y construcción de un hardware de realidad virtual que consta de un rastreador que irá adherido a los instrumentos laparoscópicos que formarán parte del sistema y que a su vez será adaptado a la estructura del simulador mecánico SIMULAP v1.0 (Ver capítulo II).

Este rastreador debe transmitir vía puerto serial, a una velocidad de 9600 Kbps, la información de la posición espacial de los instrumentos, sus rotaciones, el grado de penetración de éstos dentro del simulador mecánico y el ángulo de apertura que pueden tener los instrumentos al ser manipulados (por ejemplo: el ángulo de apertura de los dientes de la pinza laparoscópica).

Para el momento del desarrollo de este trabajo, el hardware propuesto (rastreador) para LAPAROS no se encontraba desarrollado, por lo que se decidió hacer la implementación con la herramienta comercial Isotrak II (Ver Anexo F), propiedad del LCG. El uso de esta herramienta genera un inconveniente, por no estar diseñado a la medida del proyecto LAPAROS, y es que su utilización no permite un aprovechamiento óptimo del sistema. Esto se debe a que el Isotrak II no genera dos de los datos que debe transmitir el rastreador propuesto: el primero es la penetración de los instrumentos en el simulador, impidiendo así conocer el punto de rotación de éstos (pivote) y el segundo es el ángulo de apertura de los instrumentos, ya que sólo es posible transmitir una señal binaria interpretable como apertura total o nula del instrumento.

IV.2.3 Análisis de la Arquitectura del sistema

El sistema tal como está propuesto se basa en la creación de escenas que incluyen la carga de objetos tridimensionales y la manipulación de éstos a través de transformaciones geométricas, que se verán afectadas por la entrada de información ofrecida por el rastreador a través del puerto serial. Esta data debe ser capturada y manipulada permitiendo así, que los movimientos de los instrumentos mecánicos se vean reflejados dentro de cada una de las escenas.

Según los ejercicios a desarrollar, se necesita detectar el choque entre objetos, para conseguirlo es necesario utilizar algún algoritmo de detección de colisiones. Otro de los requerimientos del sistema es la captura de objetos virtuales con los instrumentos laparoscópicos, por lo tanto una vez dada la captura es necesario aplicar las transformaciones de los instrumentos sobre los objetos afectados. Para ello debe ser utilizada una estructura de datos que permita relacionar los objetos dentro de una escena de manera jerárquica.

La siguiente figura muestra la arquitectura del sistema:

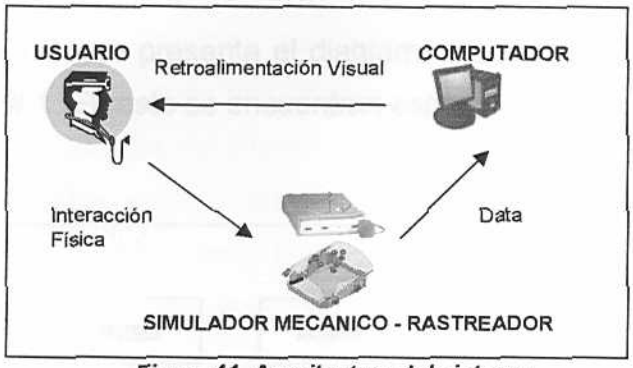

Figura 11. Arquitectura del sistema.
Fuente: Elaboración Propia

IV.3 DISEÑO

A partir de los resultados obtenidos en las etapas previas, se crea el diagrama de clases (Figura 12) correspondiente al sistema.

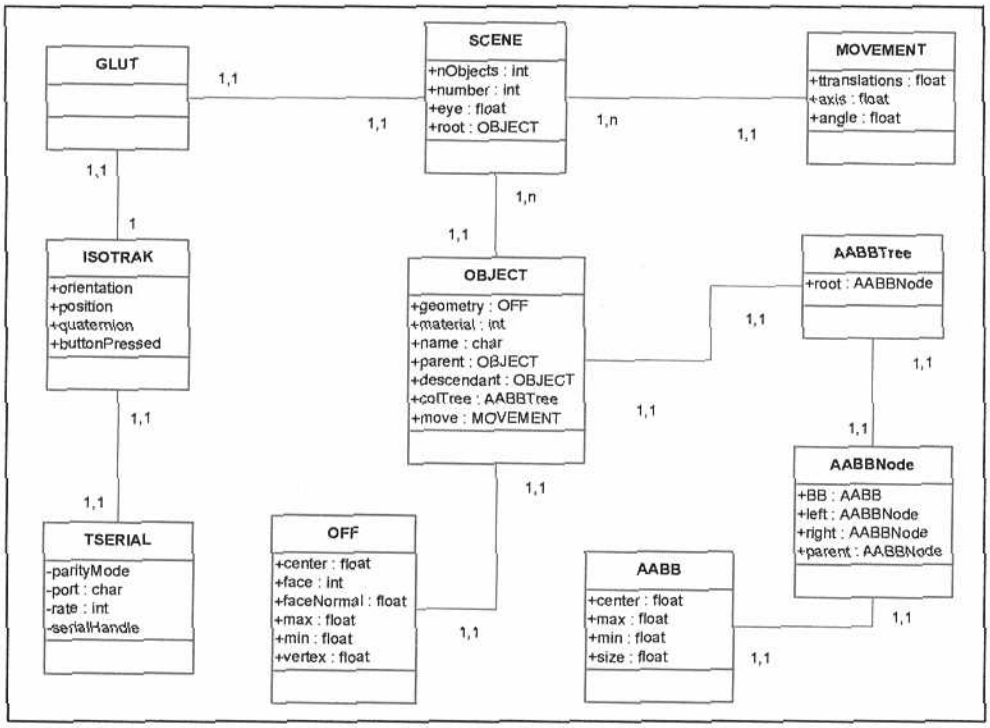

Figura 12. Diagrama de clases del sistema.
Fuente: Elaboración Propia

Para ilustrar la interacción entre los objetos se realizaron diagramas de secuencia para cada uno de los casos de uso.

A continuación se presenta el diagrama de secuencia correspondiente al caso de uso # 1. El resto se encuentran especificados en el Apéndice C.

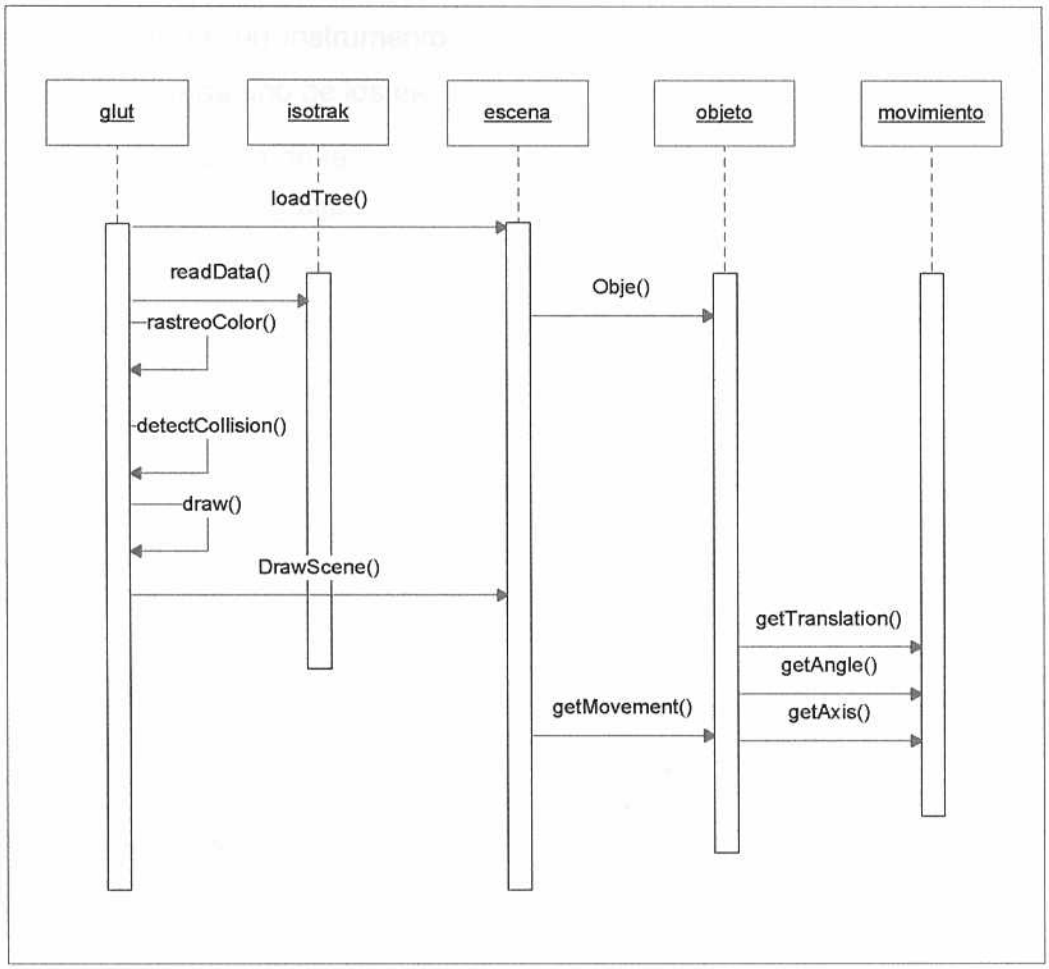

Figura 13. Diagrama de Secuencia #1.
Fuente: Elaboración Propia

IV.3.1 Diseño de las escenas del sistema

Cada uno de los ejercicios que forman parte del sistema, constituyen una escena, las cuales a su vez contienen uno o más objetos. Éstas son representaciones virtuales del interior del simulador mecánico SIMULAP v1.0 donde serán situados objetos virtuales con los que el usuario interactuará con la ayuda de un instrumento laparoscópico virtual, para cumplir con los objetivos de cada uno de los ejercicios.

La información de las escenas y de los objetos debe estar almacenada para poder acceder a ella al momento de ejecutar cada uno de los ejercicios, la que corresponde a las escenas estará contenida en archivos con extensión ".scl", de elaboración propia, y la de los objetos en archivos con extensión ".off", además de esto se creó un archivo de nombre "objects.txt", el cual contiene la ruta en que se encuentran cada uno de los archivos ".off". Las estructuras de estos archivos serán especificadas en el Apéndice G.

Para la generación virtual de las escenas es necesario conocer ciertas propiedades:

- Identificación de la escena: número único que identifica a la escena.
- Posición de la cámara: son las coordenadas de la cámara dentro de la escena.
- Número de objetos: cantidad de objetos que posee la escena.
- Nombre de los objetos: nombre con que se identifican los distintos objetos existentes en la escena.
- Transformaciones iniciales de los objetos: son las propiedades iniciales con las que son cargadas los objetos en las escenas, correspondiente a traslaciones, rotaciones y escala.

Para la carga de los modelos de los objetos 3D dentro de las escenas es necesario conocer las siguientes propiedades:

- Número de vértices: cantidad de vértices que posee el objeto 3D.
- Número de caras: cantidad de caras que componen al objeto 3D.
- Posición de los vértices: coordenadas de los vértices dentro de la escena.
- Vértices de cada cara: número de cada uno de los tres vértices que componen las caras.

Toda esta información luego de ser leída por el sistema, será almacenada en una estructura de datos jerárquica conocida como Grafo de Escena (Ver Apéndice F), la cual es ampliamente utilizada para el desarrollo de juegos 3D. Este grafo permitirá la manipulación de los objetos, las relaciones entre ellos y sus propiedades de una manera eficiente, rápida y sencilla en tiempo de ejecución.

La manera de manipular los objetos tridimensionales es a través de transformaciones geométricas. Existen objetos que dependen de las transformaciones de otros, en este caso, el Grafo de Escena es utilizado para aplicar las transformaciones de los niveles más altos del grafo a los más bajos.

En cuanto a la visualización de las escenas, se tiene que éstas serán desplegadas en pantalla completa, para su iluminación se utilizará un foco de luz colocado en la misma posición que la cámara. El usuario podrá disponer de dos puntos de vista dentro de la escena, uno más cercano a los objetos que el otro.

Debido a que el proyecto LAPAROS no contempla retroalimentación háptica en su hardware se decidió, para evitar pérdida de realismo para el usuario, que los choques entre los objetos serán detectados más no visualizados, es decir, al momento de ocurrir un choque entre un instrumento laparoscópico virtual manipulado por el usuario y un objeto de la escena, éstos no se detendrán al contacto sino que continuarán su trayectoria viéndose penetrados unos a otros.

IV.3.2 Diseño de los ejercicios del sistema

Conociendo ya los objetivos que persigue cada ejercicio del sistema, a continuación se especifica en detalle el diseño de uno de éstos y el resto puede ser encontrado en el Apéndice D.

IV.3.2.1 Diseño del ejercicio Tocar los Colores

Objetos que contiene:

- Pinza laparoscópica.
- Tablero.
- Simulador mecánico.

Descripción:

En este ejercicio se presentará un tablero donde se pintará aleatoriamente, durante un tiempo definido, una sección del mismo y el usuario deberá tocarla con la punta de la pinza, generando así una colisión entre los 2 objetos, la cual será detectada por el sistema. Si el usuario

alcanza a tocar la sección a tiempo, el sistema se lo indicará coloreando ésta de verde durante un corto tiempo, e inmediatamente después, se coloreará una nueva sección, indicándole al usuario su próximo objetivo. Si el tiempo que una sección permanece coloreada finaliza sin que el usuario tenga contacto con ella, el sistema registra un error, procede a colorear otra sección y la anterior la devuelve a su color original. Si el sistema detecta colisión con una sección distinta a la coloreada, se registra un error y se procede a cambiar de sección inmediatamente. El ejercicio finaliza cuando se han coloreado una cantidad de secciones definidas.

IV.3.3 Diseño de los objetos del sistema

Los modelos de los objetos 3D utilizados en el desarrollo de este trabajo, fueron creados en conjunto con otros integrantes del proyecto LAPAROS. Estos modelos están compuestos únicamente por triángulos con la finalidad de obtener una visualización más realista de los mismos.

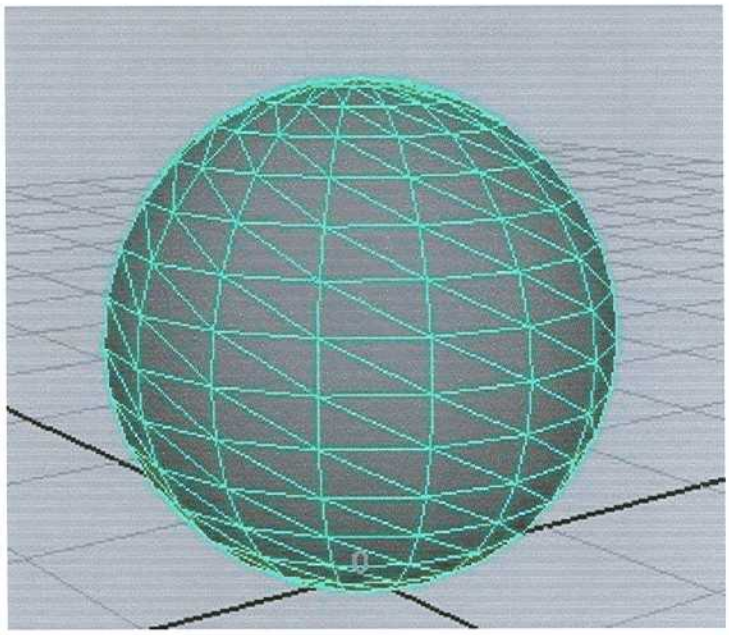

Figura 14. Modelo 3D triangulizado.
Fuente: Elaboración Propia

Según el diseño de cada uno de los ejercicios, deben ser creados los objetos que a continuación se especifican:

Nombre	Descripción	O. C.	M. U.	C. A.
Barra	Cilindro alargado.	No	Sí	No
Diente de pinza	Representación virtual de la sección externa de un diente de una pinza utilizada en la cirugía laparoscópica.	No	Sí	No
Tapa de diente	Representación virtual de la sección interna de un diente de una pinza utilizada en la cirugía laparoscópica.	No	Sí	No
Pinza Laparoscópica	Representación virtual de una pinza utilizada en la cirugía laparoscópica. Compuesta por una barra, un par de dientes de pinza y un par de tapas de diente. El usuario la utiliza para interactuar con los objetos de la escena.	Sí	Sí	No.
Diente de tijera	Representación virtual de un diente de una tijera utilizada en la cirugía laparoscópica.	No	Sí	No
Tijera Laparoscópica	Representación virtual de una tijera utilizada en la cirugía laparoscópica. Compuesta por una barra y un par de dientes de tijera. El usuario la utiliza para interactuar con los objetos de la escena.	Sí	Sí	No.
Simulador Mecánico	Representación virtual del simulador mecánico SIMULAP v1.0.	No	No	No
Cubo	Cubo.	No	No	Sí
Tablero	Tablero compuesto por 64 cubos.	Sí	No	Sí
Aros	Es una barra con un anillo en uno de sus extremos.	No	No	Sí
Línea	Cilindro delgado.	No	No.	Sí
Camino	Representa un camino a seguir, está compuesto por un conjunto de líneas.	Sí	No	Sí
Cuerda	Representa una cuerda a cortar por el usuario. Está compuesta por un conjunto de líneas.	Sí	No	Sí
Esfera	Esfera.	No	Sí	Sí

Leyenda: O.C.: Objeto compuesto. M.U.: Manipulado por el usuario. C.A.: Creado por los autores.

Tabla 5. Objetos del sistema
Fuente: Elaboración Propia

IV.4 IMPLEMENTACIÓN

La implementación del sistema se realizó en ambiente Windows XP con la herramienta Visual C++ 6.0 y con las librerías de OpenGL®, y en computadores con la siguiente configuración de hardware:

- Procesador: Intel Pentium IV de 2.4 Ghz.
- Memoria RAM: 1Ghz.
- Tarjeta Gráfica: Nvidia GeForce Fx5600 de 256 MB de memoria.
- Disco duro: 40GB.

IV.4.1 Lectura, carga y configuración de parámetros de visualización de la escena

Luego de culminar todas las etapas anteriores se inició la construcción del sistema, el primer paso fue cargar los objetos desde sus modelos 3D y almacenarlos en el grafo de escena.

Para poder cargar los objetos que formarán parte de una escena es imprescindible leerlos desde un archivo, para ello se utilizaron y adaptaron diversas librerías, algunas propiedad del LCG. Específicamente las clases Off, Glut, Scene y Obje que se muestran en el diagrama de clases del sistema.

La clase Off posee los métodos que manipulan los archivos ".off", y a partir de ellos genera los objetos en la pantalla. La clase Glut, es una API, encargada del manejo de ventanas, buffers y entradas de teclado, las cuales están fuera del alcance de OpenGL®.

Para la correcta visualización de la escena se hace uso de los siguientes métodos que proveen las librerías Glut y OpenGL®:

- gluLookAt(): este método se encarga de posicionar la cámara y apuntarla correctamente.

Figura 15. Posición de la cámara
Fuente: Elaboración Propia

- glLightfv(): permite especificar las propiedades de un foco de luz, como pueden ser su color, intensidad y posición. Este método se utilizó para colocar el foco de luz que ilumina la escena en la misma posición que la cámara.

- glMaterialfv(): permite especificar las propiedades de un material. Estas propiedades son las mismas que las de los focos de luz. Fue utilizado para especificar las propiedades de los distintos materiales utilizados para pintar los objetos de la escena.

- glutInitDisplayMode(): este método configura el modo inicial del video, para este desarrollo se utilizaron los parámetros RGBA y GLUT_DOUBLE, el primero indica el modelo de color a utilizarse en la ventana y el segundo sirve para utilizar dos buffers de color permitiendo evitar el parpadeo entre una imagen y la siguiente durante la animación.

Para la lectura de los archivos de los objetos y escenas se utilizaron los siguientes métodos:

- load():se encarga de leer las propiedades de cada uno de los objetos (vértices, caras y sus normales).

- loadTree(): se encarga de leer el archivo de cada una de las escenas, hace las llamadas al método load() para cada uno de los objetos pertenecientes a la misma y a partir de esto crea el Grafo de Escena como una instancia de la clase Scene.

La estructura del Grafo de Escena desarrollada para ser utilizada por el sistema es la siguiente:

- Nodo Raíz: es una instancia de la clase Scene que se caracteriza por tener un arreglo de apuntadores a objetos de la clase Obje. Igualmente posee información acerca del identificador de la escena, la cantidad de objetos que ésta posee y la posición de la cámara dentro de la misma.

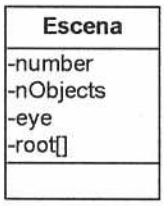

Figura 16. Nodo raíz de Grafo de Escena
Fuente: Elaboración Propia

- Nodos Hoja: son instancias de la clase Obje, por lo que poseen toda la información referente a los objetos que se cargan en la escena. Se conoce el nombre del objeto, ejes y ángulo de rotación, traslaciones y escala iniciales con las cuales debe ser cargado el objeto, así como también el material, y quién es su padre y si tuviese descendientes quiénes son éstos.

Figura 17. Nodo hoja de Grafo de Escena
Fuente: Elaboración Propia

Para la manipulación de este Grafo de Escena se utilizan los siguientes métodos:

- drawScene(): se encarga de llamar a los métodos que aplican las transformaciones a los objetos y los dibuja en la escena constantemente. Las transformaciones pueden ser ocasionadas tanto por el usuario como por el sistema.

- movingChilds(): se encarga de aplicar las transformaciones de un nodo a sus descendientes.

- reorderTree(): se encarga de ordenar el Grafo de Escena, en caso de que un nodo pase a ser descendiente de otro o deje de serlo.

- drawInstruments(): este método es siempre llamado por drawScene() y se encarga de dibujar los instrumentos laparoscópicos, utilizando las transformaciones que se generan a partir de la data recibida del rastreador.

IV.4.2 Generación de las escenas

OpenGL® maneja tres sistemas de coordenadas de tres dimensiones, uno para el mundo, otro para los objetos y el tercero para la cámara, éstos son utilizados para facilitar la conversión de coordenadas en el espacio virtual.

Estos sistemas de coordenadas son representados a través de matrices homogéneas 4x4, en las cuales se reflejan las coordenadas de los ejes del sistema y su desplazamiento con respecto al origen. En el Anexo E se encuentra información detallada sobre estas matrices.

Para aplicar una transformación a un objeto se debe modificar la matriz de objetos, esto es posible con la ayuda de los siguientes métodos pertenecientes a la librería OpenGL®:

- glTranslatef(): este método sirve para trasladar el eje de coordenadas, colocando su centro sobre un punto dado.
- glRotatef(): realiza un rotación al eje de coordenadas, tantos grados como se le indique en torno a un eje también indicado.
- glScalef(): escala el tamaño de los ejes de coordenadas tantas veces como sea indicado.

La manera en que se logra dibujar un objeto en la escena es a través del método render() perteneciente al la clase Off. Éste se encarga de dibujar todas las caras triangulares que forman los objetos con ayuda de las primitivas geométricas de la librería OpenGL®.

IV.4.3 Captura y manejo de la data proveniente del rastreador

El rastreador Isotrak II utiliza las clases Isotrak y Tserial para capturar y manipular la data que se recibe por el puerto serial. De estas clases, la primera fue desarrollada por personal del LCG y la segunda es de licencia pública.

- Tserial: se encarga de leer la información desde el puerto serial que se le indique (COM1, COM2) a la velocidad que la transmisión lo requiera.

- Isotrak: toma la data capturada por Tserial, la almacena, manipula y provee de métodos para manipularla, obteniendo así la capacidad de reflejar los movimientos hechos por el usuario en el simulador.

Con la ayuda de estas clases se logran obtener las posiciones y rotaciones que los instrumentos laparoscópicos poseen en un momento dado, las primeras vienen representadas por vectores de 3 dimensiones (x,y,z) y las segundas por quaterniones de rotación (x,y,z,w).

Debido a que el movimiento de los instrumentos laparoscópicos debe ser reflejado en tiempo real, se decidió hacer la captura de la data a una velocidad que permitiese mostrar las imágenes en la pantalla, de forma tal que se genere una animación continua.

Con la finalidad de reducir el consumo de recursos del sistema ocasionado por lo descrito anteriormente, se decidió utilizar quaterniones para manipular las rotaciones, ya que éstos exigen menos operaciones y menos espacio para almacenarlos que una matriz. En el Anexo C se explica con más detalle todo lo referente a quaterniones.

IV.4.4 Detección de colisiones

En una aplicación de realidad virtual es de suma importancia la detección de colisiones entre objetos, ya que esto permite que la interacción entre los mismos sea posible.

Existen diversos algoritmos de detección de colisiones, entre los cuales se pueden mencionar V-Clip, AABB, OBB y PCS [3]. Para el desarrollo de este sistema se utilizó el método AABB, implementado por el LCG y adaptado a la medida por los autores.

El método AABB detecta la colisión ocurrida entre 2 objetos, para ello utiliza árboles AABB cargados con la información de las cajas contenedoras (bounding boxes) de cada uno de los triángulos de los objetos. Para detectar la colisión se verifica si algunas de las cajas contenedoras de ambos objetos se encuentran solapadas.

Estas estructuras jerárquicas AABB son cargadas con información obtenida de la lectura de archivos de extensión ".aabb", donde cada uno de los modelos de los objetos posee uno de éstos. En el Anexo D se explica en mayor detalle todo lo referente a los árboles AABB, y a la detección de colisiones utilizando este método.

IV.4.5 Implementación de los ejercicios

A continuación se explica la implementación de uno de los ejercicios que forma parte del sistema. La implementación del resto de los ejercicios se encuentra detallada en el Apéndice E.

IV.4.5.1 Tocar los colores

En la Figura 18, se observa la escena que el usuario puede visualizar al realizar este ejercicio.

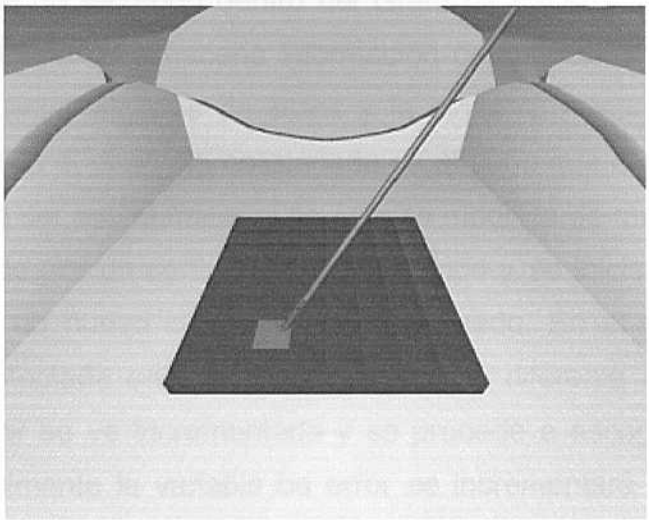

Figura 18. Ejercicio Tocar los Colores
Fuente: Elaboración Propia

Para este ejercicio el usuario interactúa con el tablero a través de la pinza. El sistema se encarga de detectar la colisión ocurrida entre los dientes de la pinza y los cubos que forman parte del tablero. Existen dos elementos fundamentales al momento de la implementación de este ejercicio, el primero es la secuencia que tendrán los cuadros que se colorean, y el segundo es por cuánto tiempo durarán encendidos.

Para generar un número aleatorio se utiliza la función rand() perteneciente a la librería cstdlib.h:

```
painted= ((float)rand()/RAND_MAX)*63;
```

De esta manera se obtiene un número aleatorio del 0 al 63, y se coloreará el cubo al que corresponde este número dentro del Grafo de

Escena. Esto ocurre cada vez que se inicia el tiempo que debe durar encendido un cubo.

Durante la ejecución del ejercicio, cada vez que es llamado el método drawScene(), se hace una llamada al método detectCollision(). En este último método se hace un recorrido dentro del Grafo de Escena, específicamente en los nodos donde se almacena información de los cubos, y se verifica si existe colisión entre algunos de éstos y la pinza. En caso de que exista colisión se verifica si el objeto colisionado es el cubo que está coloreado en ese instante, de ser así, el material de éste se modifica de manera tal que se coloree de verde por un corto período de tiempo y se inicia nuevamente la escogencia de un nuevo cubo para ser coloreado. En el caso de que la colisión sea detectada entre la pinza y un cubo diferente al coloreado, la variable de error se ve incrementada y se procede a escoger nuevamente otro cubo. Igualmente la variable de error se incrementará al momento de que se escoja un nuevo cubo y el anterior no haya sido colisionado.

Cada vez que se coloree un cubo se descuenta la variable repetitions, que almacena inicialmente la cantidad de cubos a ser coloreados durante el ejercicio, y en el momento que valga 0 el ejercicio finaliza mostrándole al usuario la cantidad de errores que cometió.

IV.5 PRUEBAS

Una vez finalizada la implementación del sistema, se realizaron las pruebas correspondientes.

Pruebas del tipo caja negra:

IV.5.1 Caso de prueba #1: Tocar los Colores

Prueba1.1 Curso normal	
Descripción	Prueba que el usuario puede tocar las secciones que le sean indicadas con colores.
Entrada	El usuario se acerca a la sección indicada y la toca con la punta de la pinza.
Salida esperada	La sección cambia de color indicando que la colisión fue exitosa.
Prueba1.2 Curso alternativo: No toca	
Descripción	Prueba que se registra un error y ocurre una retroalimentación visual indicándole al usuario su error, cuando éste no toca la sección indicada
Entrada	El usuario no toca la sección que se le indica en el tiempo previsto.
Salida esperada	La sección vuelve a su color original, se colorea otra aleatoriamente y se registra un error.
Prueba1.3 Curso alternativo: Toque equivocado	
Descripción	Prueba que se registra un error y ocurre una retroalimentación visual indicándole al usuario su error, cuando éste toca una sección equivocada
Entrada	El usuario toca una sección equivocada.
Salida esperada	La sección vuelve a su color original, se colorea otra aleatoriamente y se registra un error.

Tabla 6. Caso de prueba #1: Tocar los Colores
Fuente: Elaboración Propia

Los demás casos de prueba están especificados en el Apéndice H.

Las pruebas de caja negra realizadas sirvieron para demostrar el buen funcionamiento del sistema, pero para demostrar que los objetivos planteados en cuanto al entrenamiento de cirujanos pueden ser cumplidos con el uso del sistema, se procedió a realizar pruebas funcionales con las cuales fuese posible medir las destrezas y habilidades desarrolladas por los usuarios, en técnicas laparoscópicas, con la ayuda del simulador.

Para estas pruebas se trabajó con 2 poblaciones. La primera corresponde a estudiantes de Medicina de la Universidad Central de Venezuela, los cuales no tienen conocimientos de Laparoscopia, y la segunda población está conformada por un cirujano experto en Laparoscopia.

Para comenzar las pruebas con los estudiantes se les registró la cantidad de errores que tenían al realizar cada uno de los ejercicios sin entrenamiento previo. En la siguiente tabla se encuentra reflejada esta medición:

Nombre del Practicante	Ejercicio	Fin	Errores
Olga Gómez	Ej. #1 - Tocar los Colores	Si	7
	Ej. #2 - Seguir el Camino	Si	93
	Ej. #3 - Pasar a través de los Aros	Si	549
	Ej. #4 - Colocar las Figuras	Si	58
	Ej. #5 - Cortar las Cuerdas	Si	23
Yohana Pérez	Ej. #1 - Tocar los Colores	Si	8
	Ej. #2 - Seguir el Camino	Si	128
	Ej. #3 - Pasar a través de los Aros	No	650
	Ej. #4 - Colocar las Figuras	Si	54
	Ej. #5 - Cortar las Cuerdas	Si	23
Mildred Martínez	Ej. #1 - Tocar los Colores	Si	19
	Ej. #2 - Seguir el Camino	Si	47
	Ej. #3 - Pasar a través de los Aros	Si	61
	Ej. #4 - Colocar las Figuras	Si	59
	Ej. #5 - Cortar las Cuerdas	Si	29
Daniel Pereira	Ej. #1 - Tocar los Colores	Si	13
	Ej. #2 - Seguir el Camino	Si	87
	Ej. #3 - Pasar a través de los Aros	Si	20
	Ej. #4 - Colocar las Figuras	Si	57
	Ej. #5 - Cortar las Cuerdas	Si	24
Juan Carlos de Abreu	Ej. #1 - Tocar los Colores	Si	13
	Ej. #2 - Seguir el Camino	Si	59
	Ej. #3 - Pasar a través de los Aros	Si	45
	Ej. #4 - Colocar las Figuras	Si	40
	Ej. #5 - Cortar las Cuerdas	Si	18

Tabla 7. Primera medición del sistema con estudiantes
Fuente: Elaboración Propia

Al terminar cada ejercicio los estudiantes debían llenar una encuesta para capturar sus impresiones sobre el sistema. Éstas se encuentran detalladas en el Apéndice I.

Luego de esto se les permitió a los estudiantes que entrenaran con los 5 ejercicios durante una hora, y se procedió a realizar una nueva medición de los errores que tenían al realizar los ejercicios, esta medición se encuentra en el Apéndice I.

Con la ayuda de estas dos mediciones se logró observar una considerable baja en la cantidad de errores en que los estudiantes incurrían en la realización de los ejercicios (Ver Figura 19). En el ejercicio #3 Pasar a través de los Aros esta tendencia se invierte (Ver Figura 20), lo cual ocurre por los problemas generados por la utilización del hardware Isotrakll ya que este ejercicio necesita un alto nivel de precisión.

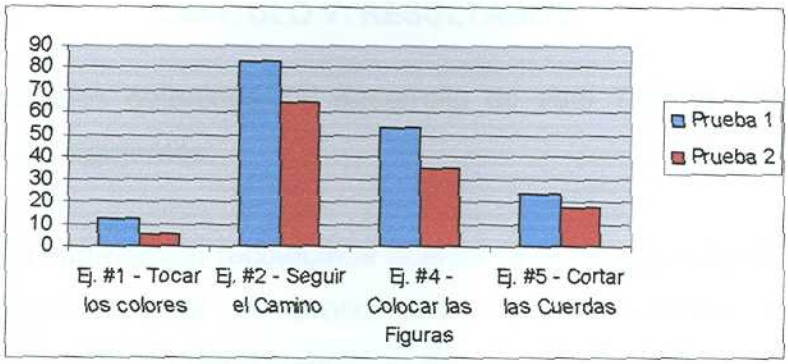

Figura 19. Gráfica 1 de Medición de errores
Fuente: Elaboración Propia

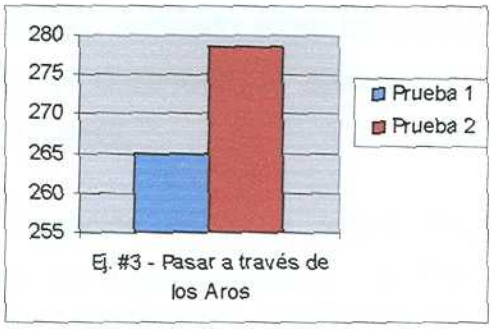

Figura 20. Gráfica 2 de Medición de errores
Fuente: Elaboración Propia

Luego de esto se hizo una demostración completa del sistema a un grupo de profesores expertos del Postgrado de Laparoscopia de la UCV y uno de ellos realizó una prueba de los ejercicios para verificar su buen funcionamiento y realizar las observaciones que tenía sobre el sistema. Estas observaciones se capturaron a través de encuestas similares a las aplicadas al grupo de estudiantes (Apéndice I).

Los resultados obtenidos por el experto son los siguientes:

Nombre del Experto	Ejercicio	Fin	Errores
Dr. Eduardo Troconi	Ej. #1 - Tocar los Colores	Si	6
	Ej. #2 - Seguir el Camino	Si	33
	Ej. #3 - Pasar a través de los Aros	Si	184
	Ej. #4 - Colocar las Figuras	Si	26
	Ej. #5 - Cortar las Cuerdas	Si	12

Tabla 8. Medición del sistema con expertos
Fuente: Elaboración Propia

CAPÍTULO V. RESULTADOS

Los resultados obtenidos del desarrollo de este Trabajo Especial de Grado fueron los siguientes:

- Con la información recolectada acerca de la cirugía mínima invasiva se logró obtener el conocimiento necesario para cubrir las expectativas que debe cumplir un simulador virtual de entrenamiento en Laparoscopia.

- A partir del estudio realizado sobre los motores 3D existentes en el mercado para el desarrollo de aplicaciones gráficas, se escogió como opción OpenGL®.

- Con la investigación realizada sobre diversos simuladores de entrenamiento en Laparoscopia y las técnicas utilizadas en el Postgrado de Laparoscopia dictado en la UCV, se idearon los ejercicios que componen el sistema, los cuales logran el desarrollo de las habilidades y destrezas básicas y medias que deben ser adquiridas para realizar cirugías mínimas invasivas.

- Se realizó la comunicación entre el hardware Isotrak II y los ejercicios desarrollados, para lograr la interacción del usuario con la escena virtual obteniendo así la inmersión necesaria y característica de todo sistema virtual.

- Se crearon algunos de los modelos 3D de los objetos utilizados en las escenas de los ejercicios.

- Se diseñó la estructura de archivos ".scl", utilizada para la almacenar la información requerida para cargar las escenas de cada uno de los ejercicios.

- Se logró que la visualización de las escenas tuviese un alto nivel de realismo ya que se asemeja a la imagen capturada por la cámara que se utiliza en el simulador mecánico.

- Se implementaron 5 ejercicios, 4 de ellos desarrollan habilidades clasificadas en el nivel básico y el otro clasificadas en el nivel medio.

- Se logró adaptar un algoritmo de detección de colisiones desarrollado por el LCG para poder capturar la interacción del usuario con los objetos de la escena.

- Se desarrollaron pruebas de caja negra para el sistema, las cuales comprobaron el funcionamiento óptimo del sistema.

- Se aplicaron pruebas funcionales del sistema para determinar si existe mejoría en la realización de los ejercicios, luego de entrenarse con los mismos. Estas pruebas arrojaron como resultado que luego de ejercitarse con el sistema se obtiene una mejoría significativa en la disminución de errores cometidos.

- Se realizó una demostración del sistema ante profesores expertos en cirugía mínima invasiva de la Universidad Central de Venezuela, los cuales calificaron los ejercicios como muy apropiados para el entrenamiento básico de estudiantes del curso de Postgrado dictado por ellos. Un profesor hizo una prueba funcional del sistema, indicando sus observaciones

- El desarrollo de este sistema contribuye en gran medida a la integración de la tecnología informática con el aprendizaje en el área médica, por ser éste pieza fundamental del proyecto LAPAROS el cual es pionero en la construcción de simuladores virtuales de entrenamiento laparoscópico en Venezuela.

CONCLUSIONES

A partir de los resultados obtenidos durante todo el proceso de desarrollo de software se enuncian las siguientes conclusiones:

- Es determinante para el logro satisfactorio de los objetivos de un proyecto de desarrollo de software involucrar a los clientes durante todas las etapas del mismo, ya que las observaciones dadas por éstos son las que dan base al cumplimiento de sus propias expectativas.

- OpenGL® es una API multiplataforma utilizada para desarrollar aplicaciones gráficas, que permite el manejo y manipulación de gráficos 3D de una manera eficiente, rápida y flexible la cual se adapta fácilmente a la implementación de sistemas de realidad virtual.

- La programación orientada a objetos es en esencia la creación de clases las cuales son módulos completos de código de alta mantenibilidad e independientes del contexto en que se utilicen, lo cual es una solución a los problemas de portabilidad y reutilización de código, características primordiales de los desarrollos de software en grandes grupos de trabajo.

- Debido a las dificultades que presenta la cirugía mínima invasiva para el cirujano, es necesario que los institutos de enseñanza de esta técnica se mantengan en constante actualización en lo referente a sus métodos de instrucción, ya que cada día son numerosos los avances que se realizan en esta área, permitiendo así la formación de mejores profesionales.

- Los simuladores laparoscópicos ofrecen diversos ejercicios de entrenamiento, los cuales a través de su adecuada utilización facilitan la transición entre la cirugía abierta tradicional y la cirugía mínima invasiva, un ejemplo de estos ejercicios son los desarrollados en este trabajo.

RECOMENDACIONES

- Para poner en producción la primera etapa del sistema a desarrollar en el proyecto LAPAROS (que incluye este sistema) es necesario finalizar la construcción del hardware hecho a la medida para este sistema, con lo cual se solucionarán los problemas generados por el uso del hardware IsotrakII.

- Para alcanzar un nivel de realismo más alto del que ofrece el sistema se puede:

  - *Suavizar los bordes de los modelos de los objetos 3D.*
  - Agregar texturas a los objetos 3D.
  - Añadirle al hardware la propiedad de retroalimentación táctil y de ser así no permitir la superposición de los objetos 3D dentro de las escenas.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

[1] Jacobson, I. (1999) *The Unified Software Development Process.* Addison-Wesley Professional.

[2] Woo, M (2003) *The Official Guide to Learning OpenGL.* Addison-Wesley Professional.

TRABAJOS ESPECIALES DE GRADO

[3] Navarro, H. (2004) *Implementación y evaluación comparativa de métodos de detección de colisiones para un ambiente de Realidad Virtual Inmersiva.* Tesis de Maestría, Escuela de Computación, Universidad Central de Venezuela.

[4] Zambrano A. y Piñango A. (2003) *Diseño y Construcción de un Simulador para Cirugía Laparoscópica.* Tesis de Pre-Grado, Facultad de Ingenieria, Universidad Central de Venezuela.

DOCUMENTOS ELECTRÓNICOS

[5]Azorín, J. (2004). *Telecirugía y Simulación quirúrgica*, [en línea]. Disponible en: http://lorca.umh.es/isa/es/doct/rst/Telecirugia.pdf [2005, 18 de Julio].

[6]Bernal F. *Laparoscopia.....¿Laparos que?.....*, [en línea]. Disponible en: http://www.hgm.salud.gob.mx/cu_co_public/laparoscopia.doc [2005, 24 de Mayo].

[7]Herrera, L (2001) *Ingeniería De Requerimientos*, [en línea]. Disponible en: http://www.monografias.com/trabajos6/resof/resof.shtml [2005, 11 de Septiembre].

[8]Monserrat, C. (2002). *Gertiss:Simulador quirúrgico virtual para el entrenamiento en cirugías mínimamente invasivas,* [en línea]. Disponible

en: http://www.seis.es/informed02/INFORMED2002_monserratc.pdf [2005,13 de Mayo].

[9]Tendick, F. (1997) *Comparison of laparoscopic imaging systems and conditions using a knot tying task*, [en línea]. Disponible en: http://www3.interscience.wiley.com/cgi-bin/abstract/56873/ABSTRACT [2005, 2 de Agosto].

[10]Scelza A. (2003). *Laparoscopia en el Abdomen Agudo*, [en línea]. Disponible en: http://www.mednet.org.uy/cq3/emc/monografias/lap-aa-062003.doc [2005, 2 de Agosto].

[11]Dome Medical. (2001). *LTS 2000 Manual For Developing & Assessing Laparoscopic Skills*, [en línea]. Disponible en: http://www.lts2000.com [2004, 28 de Noviembre].

[12]Microsoft y Rational. (1998). *A White Paper on the Benefits of Integrating Microsoft Solutions Framework and The Rational Process.* Disponible en: http://www.rational.com/uml/papers. [2005, 2 de Agosto]

[13]*Proceso Unificado Captura de Requisitos* [en línea]. Disponible en: http://kybele.escet.urjc.es/documentos/IS/Requisitos.pdf [2005, 15 de Julio].

PAGINAS WEB

[14]Castro, R. (2004). *Estructura Básica del Proceso Unificado de Desarrollo del Software* [en línea]. Disponible en: www.icesi.edu.co/es/publicaciones/publicaciones/contenidos/sistemas_telematica/3/rcastro_estructura-bas-puds.pdf [2005, 9 de Agosto].

[15]Fernandez, A (2001). *Introducción a UML.* [en línea]. Disponible en: http://www-gris.det.uvigo.es/~avilas/UML/UML.html [2005, 9 de Agosto].

[16]Haptica. *ProMis*, [en línea]. Disponible en: http://www.haptica.com/ [2004,28 de Noviembre].

[17]Ibáñez, L.(1994)*Colicestectomia Laparoscópica*, [en línea]. Disponible en: http://escuela.med.puc.cl/paginas/publicaciones/Boletin/html/cirugia/2_5.html [2005, 15 de Mayo].

[18]López, A. (2001) *Introducción a OpenGL*, [en línea]. Disponible en: http://www.alobbs.com/revistas/opengl3 [2005, 22 de Septiembre].

[19]López, A. *Introducción a OpenGL (III)*, [en línea]. Disponible en: http://www.alobbs.com/revistas/opengl3 [2005, 25 de Septiembre].

[20]Medline Plus. *Laparoscopia,* [en línea]. Disponible en: http://www.nlm.nih.gov/medlineplus/spanish/ency/article/007016.htm [2005, 15 de Enero].

[21]Microsoft. *Modelado visual de aplicaciones de software,* [en línea]. Disponible en: http://www.microsoft.com/spanish/Msdn/articulos/archivo/230801/voices/modelsoftware.asp [2005, 25 de Septiembre].

[22]Polhemus *Isotrak II*, [en línea]. Disponible en: http://www.inition.co.uk/inition/product_mocaptrack_polhemus_isotrak.php [2005, 30 de Septiembre]

[23]Sepúlveda, M. (1998). *Qué es OpenGL?*, [en línea]. Disponible en: http://www.linuxfocus.org/Castellano/January1998/article15.html [2005,12 de Junio].

[24]Simbionix, LAP Mentor, [en línea]. Disponible en: http://www.simbionix.com/LAP_Mentor.html [2004, 28 de Noviembre].

[25]Vitulli, D. (2000). *The 3D Engine,* [en línea]. Disponible en: http://www.spacesimulator.net/tut1_3dengine.html [2005,14 de Septiembre].

[26]Wikipedia. *Cuaterniones* [en línea]. Disponible en: http://es.wikipedia.org/wiki/Cuaterniones [2005, 22 de Septiembre].

[27]Zavala R. (2002) *Ingeniería de Software*, [en línea]. Disponible en: http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html [2005, 2 de Agosto].

[28]*Basics* [en línea]. Disponible en: http://www.opensg.org/OpenSG-Tutorial-V.1.0/TutorialBasics.html [2005, 22 de Septiembre].

[29]Cirugía Laparoscópica, [en línea]. Disponible en: http://ciberhabitat.gob.mx/hospital/laparoscopia/equipo.htm [2005, 15 de Agosto].

[30]*Diagramas de Clases,* [en línea]. Disponible en: http://www.creangel.com/uml/clases.php [2005, 26 de Septiembre].

[31]*DirectX.* Disponible en: www.lacompu.com.ar/soporte/boletines/directx/index.php3 [2005, 9 de Agosto].

[32]*El mundo virtual con fines educativos*, [en línea]. Disponible en: http://catedra.ucol.mx/virtual.htm [2005, 20 de Enero].

[33]*Game Mathematical Library: Quaterniones*, [en línea]. Disponible en: http://edevi.zonared.com/esp/nums/4/article.php?f=gml_2.php [2005, 22 de Septiembre].

[34]*Glosario*, [en línea]. Disponible en: http://www.learnthenet.com/spanish/glossary/vr.htm [2004,18 de Diciembre].

[35]Laparoscopic Surgery, [en línea]. Disponible en: http://www.levelfive.com/ZINA/HTML/Surgical_HTML/laparoscopy.html [2005, 10 de Septiembre].

[36]*Matrix and Quaternion FAQ*, [en línea]. Disponible en: http://skal.planet-d.net/demo/matrixfaq.htm#Q49 [2005, 22 de Septiembre].

[37]Modelado Dinámico, [en línea]. Disponible en: http://www.clikear.com/manuales/uml/modelodinamico.asp [2005, 25 de Septiembre].

[38]*Object File Format (.off)*, [en línea]. Disponible en: http://shape.cs.princeton.edu/benchmark/documentation/off_format.html [2005, 20 de Septiembre].

[39]*OpenGL: Tutorial 1. Matrices y aspectos matemáticos*, [en línea]. Disponible en: http://worldspace.berlios.de/fase1/tut_1_1/2.html [2005, 22 de Septiembre].

[40]*Potencial de la Realidad Virtual*, [en línea]. Disponible en: http://www.lania.mx/biblioteca/newsletters/1994-otono/art4.html [2004, 13 de Diciembre].

[41]*Realidad Virtual*, [en línea]. Disponible en: http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/virtualoffice.htm [2004, 13 de Diciembre].

[42]*Realidad Virtual*, [en línea]. Disponible en: http://www.inei.gob.pe/biblioineipub/bancopub/inf/lib5047/INDEX.htm [2004, 13 de Diciembre].

[43]*Tutorial de UML*, [en línea]. Disponible en: http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html [2005, 22 de Septiembre].

[44]*UML*, [en línea]. Disponible en: http://usuarios.lycos.es/oopere/uml.html [2005, 22 de Septiembre].

[45]*Un estudio a fondo de UML*, [en línea]. Disponible en: http http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c124.html [2005, 22 de Septiembre].

GLOSARIO

2

2D: Dos dimensiones, por ejemplo, alto y ancho.

3

3D: Tres dimensiones, por ejemplo, alto, ancho y profundo.

A

API: (interfaz de programación de la aplicación) es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación, generalmente entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o íconos en la pantalla.

B

Bounding box: caja que contiene una o varias figuras geométricas.

C

Cirugía mínima invasiva: es un grupo de operaciones que se realizan con la ayuda de una cámara e instrumentos quirúrgicos especiales. Con la ayuda de esta cámara e instrumentos, el cirujano puede practicar la cirugía a través de pequeños orificios sin tener que realizar la tradicional incisión o herida de gran tamaño.

E

Electrocauterio: Un electrocauterio es un instrumento que dirige una corriente eléctrica de alta frecuencia a través de un área de tejido.

L

Laparoscopia: es el nombre que se les da a las cirugías mínimas invasivas realizada en la región abdominal.

M

Modelo 3D: es la representación gráfica en tres dimensiones, asistida por computadora, de objetos reales.

Motor 3D: es una colección de estructuras, funciones y algoritmos utilizados para visualizar, después de muchos cálculos y transformaciones, objetos tridimensionales en una pantalla bidimensional.

P

Portabilidad: La portabilidad es la capacidad que ha de tener el programa para funcionar en distintos sistemas operativos.

R

Rastreador: dispositivo que provee coordenadas numéricas para identificar la posición y/o orientación de un objeto o usuario en espacio real.

Rendering: es el proceso final de generar una imagen 2D o animación a partir de una escena creada.

Retroalimentación háptica: es la capacidad que posee un hardware para comunicar como los ambientes virtuales afectan al usuario, por ejemplo, las paredes deberían de parar a alguien, en vez de permitirle pasar por ella; mientras que una pelota debería golpear al usuario si choca con ella.

RGBA: acrónimo inglés de Red, Gren, Blue, Alpha. Es un modelo de color basado en la síntesis aditiva, en el cual es posible representar un color mediante la mezcla por adición de los tres colores luz primarios: rojo, verde y azul y con un canal alpha también conocido como canal de transparencia, el cual representa el nivel de transparencia del color generado.