

Universidad de las Ciencias Informáticas

Facultad 5



**Propuesta de Interfaz Gráfica de Usuario para el proyecto
VISMEDIC.**

**Trabajo de Diploma para optar por el Título de
Ingenieros en Ciencias Informáticas**

Autores: Yamila L. Verdecia Álvarez

Leyanis Garcell Fundora

Tutor: Ing. Osvaldo Pereira Barzaga

Ciudad de La Habana, Junio de 2010

La conclusión es que sabemos muy poco y sin embargo es asombroso lo mucho que conocemos. Y más asombroso todavía es que un conocimiento tan pequeño pueda dar tanto poder.

Bertrand Arthur William Russell

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yamila Verdecia Álvarez

Autor

Leyanis Garcell Fundora

Autor

Ing. Osvaldo Pereira Barzaga

Tutor

Agradecimientos

De Yamila:

A mi mamá, la persona más importante de mi vida, por sus consejos, su paciencia y apoyo, por su gran esfuerzo por verme un día convertida en ingeniera...!este título es para ti mami!

A mi papá, por tener la paciencia y delicadeza de siempre encontrar las palabras correctas en el momento preciso. Por ayudarme a superarme y guiarme siempre por el buen camino...!te quiero mucho!

A mi hermana, por su apoyo incondicional a lo largo de estos 5 años que hemos recorrido juntas, por todas las horas de estudio y dedicación que pasamos juntas, por soportarme durante el desarrollo de este trabajo.

A mi tía Leonor y mi tío William, que siempre me han apoyado, ayudado y querido como si fuese su hija, a quienes les voy a estar eternamente agradecida por ser como mis segundos padres.

A mi familia en general, en especial a mi hermano Maykel y a mi tía Teresa que estuvieron al tanto de mis estudios y aportaron lo mejor de sí, para que pudiera culminarlos exitosamente.

A mi compañera de tesis, por haber confiado en mí para esta ardua tarea, por no haberse rendido ni un instante hasta no ver materializado este trabajo.

A mi tutor Osvaldo, por el tiempo que nos ha dedicado y la ayuda que nos ha brindado, que sin él no hubiese sido posible este trabajo.

A mis amigos y compañeros de aula, en especial a Maite, Julio, Jorge Carlos y Danay con los cuales he compartido momentos inolvidables, a los que quedaron en el camino y que aún están presentes en mi memoria y los que están para apoyarme en todo momento.

A la UCI por ser para mí más que una escuela, que con sus cosas buenas y malas me ha hecho ser una mejor persona, por brindarnos la oportunidad de formarnos como profesionales y por poner a nuestra disposición los recursos necesarios para lograr nuestros sueños.

A todos muchísimas gracias de todo corazón.

Agradecimientos

De Leyanis:

A mi madre, por sus muestras de amor infinito, sus noches de desvelo y por ser la luz que alumbra mi vida.

A mi padre, por sus sabios consejos.

A mi princesa Stacy, por alegrar mis días.

A mi hermana Eilyn, por su cariño y comprensión.

A mi abuelita Celia, por su infinita ternura.

A mis compañeros de siempre: Natilla, Yandy, Celi y Vity, por estar a mi lado cuando los necesité.

A mi compañera de tesis por su empeño y apoyo.

A mis profesores, por transmitirme sus conocimientos durante estos cinco cursos.

A mi tutor Osvaldo, por su ayuda y confianza en nosotras.

A la UCI, por formarme como profesional.

A la Revolución, por permitirme ser parte de este hermoso proyecto.

A mi novio Vladimir de forma muy especial, por su paciencia, dedicación y apoyo incondicional.

A todos los que de una forma u otra contribuyeron con la realización de esta tesis.

A todos, muchas gracias.

Dedicatoria

De Yamila:

A mis padres y mi hermana que son todo para mí al igual que yo lo soy para ellos. Que han sufrido mi carrera tanto o más que yo.

Por tanto amor, por enseñarme que la perseverancia, la dedicación y el empeño son las armas claves para el logro de nuestros objetivos, por el impulso que me dieron siempre que lo necesité, por las palabras de aliento.

Por guiarme para ser la persona que soy hoy.

De Leyanis:

A mi más preciado tesoro, mi madre.

A la memoria de quien fue un padre para mi, Julio Cruz López, estarás siempre en mi corazón.

Resumen

En los últimos años, el uso de imágenes digitales para la investigación y el diagnóstico médico ha aumentado considerablemente, a raíz de esto, es necesario desarrollar nuevas y mejores aplicaciones para la visualización de imágenes medicas. Debido a que el software que se está desarrollando en el proyecto VISMEDIC en la Facultad 5 de la Universidad de la Ciencias Informática no cuenta con una interfaz gráfica de usuario (GUI), el presente trabajo de diploma propone una GUI fundamentada en el análisis de las principales funcionalidades que presentan algunos de los software de visualización de imágenes médicas existentes en el mercado, en el estudio de las principales bibliotecas gráficas de desarrollo en sistemas GNU/Linux y en los principios básicos para el diseño de las interfaces gráficas.

En este documento se exponen algunos de los conceptos de interfaz gráfica de usuario, las características fundamentales, los principios para el diseño de las mismas, así como el ciclo de desarrollo. Se abordan los conceptos de accesibilidad y usabilidad enmarcado dentro de la Arquitectura de la Información. Se fundamenta la selección de las metodologías, herramientas, tecnologías utilizadas y se presenta la propuesta de solución a través de los artefactos generados en el diseño e implementación. Por último, se expone todo el proceso para la validación de la interfaz propuesta a través de método Delphi.

Palabras claves: Interfaz Gráfica de Usuario (GUI), visualización médica, principios de diseño.

Tabla de Contenidos

Introducción	1
Capítulo 1: Fundamentación Teórica.	5
Introducción.....	5
1.1 Interfaz de Usuario.....	5
1.2 ¿Qué es una Interfaz Gráfica de Usuario?	6
1.2.1 Componentes gráficos.	7
1.2.2 Características de las GUIs.....	8
1.2.3 Ventajas del uso de las GUIs.	8
1.3 Diseño de interfaces gráficas de usuario.	9
1.3.2 Principios para el diseño de interfaces gráficas de usuario.	12
1.4 Arquitectura de la Información.....	15
1.4.2 Usabilidad.	16
1.4.3 Accesibilidad.	18
1.5 Software de Visualización Médica para Imagenología.....	19
1.5.1 Software Internacionales.....	19
1.5.2 Software Nacional.	21
1.5.3 Estudio de Homólogos.	21
1.6 Biblioteca gráfica de desarrollo en sistemas GNU/Linux.	23
1.6.1 Qt.....	23
1.6.2 Gtk+	23
1.7 Lenguaje de programación.....	24
1.7.1 IDE de programación.	25
1.8 Metodología de desarrollo.	26
1.8.1 RUP (Rational Unified Process).	26
1.9 Lenguaje de Modelado.	27
1.9.1 UML (Unified Modeling Language).....	28
1.10 Herramientas CASE.....	28
1.10.1 Visual Paradigm.....	29
1.11 Conclusiones generales del capítulo.....	29

Capítulo 2: Características del Sistema.	31
Introducción	31
2.1 Solución Propuesta	31
2.1.1 Entregable de la Arquitectura de la Información.....	31
2.1.2 Funcionalidades.....	32
2.1.3 Principios de Diseño.	33
2.1.4 Biblioteca gráfica.....	34
2.2 Modelo de dominio	35
2.2.1 Diagrama modelo dominio.	35
2.2.2 Descripción del Modelo de Dominio.	36
2.3 Levantamiento de los requisitos del sistema	36
2.3.1 Requisitos funcionales.	37
2.3.2 Requisitos no funcionales.	38
2.4 Definición de los actores del sistema.	39
2.5 Casos de uso del sistema	40
2.5.1 Diagrama de Casos de usos del sistema.	40
2.5.2 Descripción de los casos de uso del sistema.	41
2.6 Conclusiones generales del capítulo	44
Capítulo 3: Análisis y Diseño del Sistema.	45
Introducción	45
3.1 Análisis del Sistema	45
3.1.1. Diagrama de Clases del Análisis.....	45
Diagrama de Clase del Análisis “Cargar DICOM”	46
Diagrama de Clase del Análisis “Reconstrucción 3D”	46
Diagrama de Clase del Análisis “Segmentación”	47
3.2 Diseño del Sistema	47
3.2.1. Diagramas de Clases.....	47
Diagrama de Clase del Diseño “Cargar DICOM”.....	48
Diagrama de Clase del Diseño “Reconstrucción 3D”	48
Diagrama de Clase del Diseño “Segmentación”.....	49
3.2.2. Diagramas de Interacción.	49
Diagrama de Secuencia “Cargar DICOM”	50
Diagrama de Secuencia “Reconstrucción 3D” (escenario Planar Rendering).....	50
Diagrama de Secuencia “Reconstrucción 3D” (escenario Volume Rendering).....	52
Diagrama de Secuencia “Reconstrucción 3D” (escenario Surface Rendering).....	53

Tabla de Contenidos

Diagrama de Secuencia “Segmentación” (escenario Segmentación manual)	53
Diagrama de Secuencia “Segmentación” (escenario Segmentación automática)	54
2.3 Conclusiones generales del capítulo.....	54
Capítulo 4: Implementación y Validación del Sistema.	55
Introducción.....	55
4.1 Diagrama de Componentes.	55
Diagrama de Componente “Cargar DICOM”	55
Diagrama de Componente “Reconstrucción 3D”	56
Diagrama de Componente “Segmentación”	56
4.2 Validación de la propuesta. Método Delphi	57
4.2.1 Proceso de selección de los expertos	58
4.2.2 Elaboración y aplicación de las encuestas.	59
4.2.3 Desarrollo y procesamiento de la información.....	65
4.3 Análisis de los resultados de la validación del proceso.	72
4.4 Conclusiones generales del capítulo.....	74
Conclusiones	75
Recomendaciones	76
Referencias Bibliográficas	77
Bibliografía Consultada.....	81
Anexos.....	84
Anexo 1 [Caso de Uso Expandidos]	84
Anexo 2 [Encuesta de autovaloración para determinar el coeficiente de competencia]	101
Anexo 3 [Encuesta a expertos para la validación de la Interfaz Propuesta]	103
Glosario de Términos.	105

Índice de Figuras

Figura 1 Estructura de la Interfaz Gráfica de Usuario.....	32
Figura 2 Diagrama Modelo de Dominio.....	35
Figura 3 Diagrama de Casos de Usos del Sistema.....	40
Figura 4 Coeficiente de Competencia de los Expertos.....	72
Figura 5 Nivel de Adecuación de las Interrogantes.....	73
Figura 6 Nivel de Adecuación de las Preguntas de la encuesta.....	73

Índice de Tablas

Tabla 1 Resumen de las actividades de cada una de las etapas del desarrollo.	12
Tabla 2 Estudio comparativo de software para la visualización de imágenes médicas.....	22
Tabla 3 Comparación de funcionalidades software de visualización de imágenes médicas.	22
Tabla 4 Actores del Sistema.....	39
Tabla 5 Descripción del Caso de Uso: Cargar imágenes DICOM.	41
Tabla 6 Descripción del Caso de Uso: Mostrar vistas axial, coronal y sagital.....	41
Tabla 7 Descripción del Caso de Uso: Rotar imagen.	41
Tabla 8 Descripción del Caso de Uso: Escalar imagen.....	42
Tabla 9 Descripción del Caso de Uso: Deshacer acciones.....	42
Tabla 10 Descripción del Caso de Uso: Rehacer acciones.....	42
Tabla 11 Descripción del Caso de Uso: Aplicar Filtros.	43
Tabla 12 Descripción del Caso de Uso: Realizar mediciones sobre la imagen.....	43
Tabla 13 Descripción del Caso de Uso: Calcular Volumen.	43
Tabla 14 Descripción del Caso de Uso: Segmentación.....	44
Tabla 15 Descripción del Caso de Uso: Reconstrucción 3D.....	44
Tabla 16 Grados de influencia en la determinación del coeficiente de argumentación.....	59
Tabla 17 Coeficiente de competencia de los expertos.	61
Tabla 18 Cómputo de evaluación de los expertos por cada pregunta.	63
Tabla 19 Frecuencias absolutas para cada pregunta de la encuesta.	66
Tabla 20 Frecuencias absolutas acumuladas para cada pregunta de la encuesta.....	68
Tabla 21 Frecuencias relativas acumuladas para cada pregunta de la encuesta.....	69
Tabla 22 Puntos de Corte.....	71

Tabla 23 Grados de Adecuación.....	72
---	----

Introducción

En los últimos años la visualización está adquiriendo un carácter cada vez más importante en la medicina, es por ello que los software para la visualización de imágenes médicas han evolucionado a una gran velocidad, incrementado la capacidad de diagnósticos y el conocimiento profundo de las más diversas e insospechadas enfermedades. Las imágenes médicas constituyen una forma de realizar diagnósticos no invasivos a pacientes, por lo que se utilizan mucho en diversas esferas de la medicina. A raíz de todos los avances en el campo de la medicina, han surgido diversas modalidades de adquisición de imágenes especializadas de acuerdo con el tipo de diagnóstico como la Resonancia Magnética, Tomografías, Rayos X, entre otros.

Los equipos de adquisición de imágenes son fabricados por compañías que por lo general proporcionan el software de manipulación de equipo, estos software cumplen con todos los estándares de calidad, sin embargo, las interfaces son muy generales y no están especializadas a las estudios de una estructura anatómica en particular. Los productos de software deben contar con una interfaz amigable, accesible y flexible para todos los usuarios. Una interfaz bien definida es diseñada específicamente según los requisitos de cada usuario final. Pueden existir interfaces con propósitos comunes, apariencias diferentes y resultados exitosos para el cliente [1], diseñada propiamente para la persona o usuario final que la usará. En estos tiempos se hace cada vez más imperativo una mejor interacción hombre-computadora a través de una apropiada interfaz gráfica de usuario (GUI).

La dependencia tecnológica de nuestro país en el área de visualización médica es muy alta. Los software médico para el análisis de imágenes médicas 3D disponibles en el mercado hoy en día necesita de estaciones de trabajo especializadas de alto rendimiento y por ende costosas, así como software propietario para el manejo de estas imágenes. Los sistemas de imagenología médica instalados en los centros hospitalarios están atados a un proveedor específico de hardware y software, con la característica que el médico, radiólogo o técnico se tienen que adaptar a la herramienta. En Cuba anualmente se paga por la renovación de la licencia del software, que por causa del embargo económico impuesto a nuestro país por el gobierno de los Estados Unidos, el precio de estas en ocasiones se duplica.

Por lo anteriormente descrito, se creó un proyecto llamado VISMEDIC en el seno de la Universidad de las Ciencias Informáticas con el objetivo de elaborar un sistema informático de visualización de imágenes 2D y 3D, dicho software no cuenta aún con una interfaz gráfica que cumpla con los estándares de diseño. Esta problemática atenta sobre los niveles de aceptación y usabilidad, elementos indispensables para comercializar o desplegar un software.

Dada la situación problemática expuesta se plantea el siguiente **Problema Científico**: Como facilitar al usuario el acceso e interacción con las funcionalidades que brinda el software de visualización desarrollado por el proyecto VISMEDIC.

El problema planteado está contenido en el **objeto de estudio**: Interfaz gráfica de usuario para aplicaciones médicas. Se define, a partir de la situación problemática, como **objetivo de la investigación**: Desarrollar una interfaz gráfica de usuario que cumpla con los estándares internacionales de diseño para facilitar el acceso e iteración con las funcionalidades del software. Por consiguiente, el **campo de acción** se reduce a: Interfaz gráfica de usuario para aplicaciones médicas de visualización de imágenes 3D.

La **Idea a Defender** que se plantea es: Una interfaz gráfica de usuario funcional para los clientes finales del software es aquella que cumpla con los requerimientos del sistema y los estándares de diseño internacionales.

Para guiar la investigación se plantean las siguientes **Tareas Investigativas**:

- Análisis del marco teórico del desarrollo de GUI para la visualización de imágenes médicas existentes en el ámbito nacional e internacional, para identificar las características básicas que debe presentar una interfaz de este tipo.
- Caracterización y selección de la biblioteca de desarrollo de interfaces gráficas de usuario en GNU/Linux para el desarrollo de la propuesta.
- Caracterización de los estándares internacionales de diseño de las GUIs en software para la visualización de imágenes médicas para lograr que el diseño de la interfaz sea un diseño inteligente, accesible y funcional.

- Captura de requisitos funcionales y no funcionales.
- Diseño e implementación de la solución propuesta.
- Validación de la GUI con especialistas a través de encuestas para conocer el grado de aceptación de la propuesta por parte de los usuarios finales.

Para el desarrollo de estas tareas se utilizan una serie de **métodos científicos**, tanto teóricos como empíricos:

Métodos Teóricos:

- Analítico-Sintético, se aplicará para entender las GUIs a partir del análisis de las características, clasificación y bibliotecas de desarrollo que presentan, y para formular conclusiones a través de la síntesis de los conocimientos y resultados obtenidos.
- Histórico-Lógico, posibilitará conocer los antecedentes y tendencias actuales de las GUIs en la visualización de imágenes médicas a partir del análisis de su evolución y desarrollo actual.
- Modelación, se aplicará para reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

Métodos Empíricos:

- Encuesta a especialistas, permitirá obtener información necesaria para la investigación y su posterior validación, debido a su experiencia; muchas veces no documentada.
- Entrevistas al jefe del proyecto sobre el tema de las GUIs y los estándares de diseños internacionales, que nos permita obtener una guía para el desarrollo del sistema.

El Trabajo de Diploma está estructurado en cuatro capítulos, los cuales se describen a continuación:

Capítulo 1: Fundamentación Teórica.

El contenido de este capítulo está centrado en el estado del arte relacionado con el objeto de estudio. Se presenta un estudio realizado de las principales bibliotecas gráficas de desarrollo en sistemas GNU/Linux. Se abordan diferentes conceptos de la interfaz gráfica de usuario (GUI), características fundamentales y los principios para el diseño de las mismas, así como algunos elementos de la arquitectura de la información para lograr un diseño accesible.

Capítulo 2: Características del Sistema.

En este capítulo se profundiza en el problema a resolver a través de su descripción y se muestra el modelo de dominio generado. Se realiza una propuesta de solución y se identifican los requisitos funcionales y no funcionales que deben tenerse en cuenta. Por último, a partir de los requisitos se realiza una descripción de los casos de uso y se presentan los diagramas relacionados con estos.

Capítulo 3: Análisis y Diseño del Sistema.

Se presentan los diagramas de clases del análisis, los diagramas de clases del diseño, los diagramas de secuencia, se muestran los diagramas de despliegue y de componente.

Capítulo 4: Implementación y Validación.

El contenido de este último capítulo se enfoca en la implementación y validación de la interfaz propuesta, a través del Método Delphi, tomando como base los resultados que se presentan en los capítulos anteriores.

Capítulo 1: Fundamentación Teórica.

Introducción

En este capítulo se realizará un estudio acerca del desarrollo de la interfaz gráfica de usuario, sus características y ventajas, así como un estudio de los diferentes estándares internacionales empleados en el diseño de las mismas. También, se describirán las bibliotecas gráficas de desarrollo más utilizadas tanto en el mundo como en la universidad para la implementación de interfaces gráficas funcionales. Se exponen un conjunto de conceptos y terminologías. Además, se fundamenta la selección de las herramientas y tecnologías que se utilizaran en el desarrollo de la propuesta.

1.1 Interfaz de Usuario.

La interfaz de usuario (IU), es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina. La interfaz de usuario es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario. La IU es responsable de solicitar comandos al usuario, y de desplegar los resultados de la aplicación de una manera comprensible. No es responsable de los cálculos de la aplicación, ni del almacenamiento, recuperación y transmisión de la información.

Clasificación de interfaces de usuario.

Según su construcción, pueden ser de software o de hardware:

- **Interfaces de hardware:** es un conjunto de controles o dispositivos que permiten la interacción hombre-máquina, de modo que permiten introducir o leer datos del equipo, mediante pulsadores, reguladores e instrumentos.
- **Interfaces de software:** son programas o parte de ellos, que permiten expresar nuestros deseos al ordenador o visualizar su respuesta.

Capítulo 1: Fundamentación Teórica

Atendiendo a como el usuario puede interactuar con una interfaz, se conocen varios tipos de interfaces de usuario:

- **Interfaces alfanuméricas:** solamente presentan texto.
- **Interfaces táctiles:** representan gráficamente un "panel de control" en una pantalla sensible que permite interactuar con el dedo de forma similar a si se accionara un control físico.
- **Interfaces gráficas de usuario:** permiten comunicarse con el ordenador de una forma muy rápida e intuitiva representando gráficamente los elementos de control y medida.

1.2 ¿Qué es una Interfaz Gráfica de Usuario?

Las Interfaces Gráficas de Usuarios, en inglés Graphics User Interface (GUI), surgen dada la necesidad de hacer más simple el uso de los ordenadores para todo tipo de usuarios y no solo restringir el uso de estos a usuarios avanzados.

“Las GUI son una forma de representar la información procesada por la computadora de manera visual, concreta y manejable por el usuario final haciéndola más comprensible, más didáctica, más asimilable, más usable.” [2]

También existen otras definiciones como:

“La interfaz gráfica de usuario es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.” [3]

“Una interfaz gráfica de usuario (GUI) es un programa de interfaz que aprovecha las capacidades de despliegue gráfico de la computadora que permite hacer más sencillo el uso de los programas.” [4]

Capítulo 1: Fundamentación Teórica

La interfaz gráfica del usuario, dicho de forma más sencilla, es el conjunto de elementos gráficos (ventanas, menús, etc.) que permiten la interacción entre el usuario y una aplicación informática.

1.2.1 Componentes gráficos.

Algunos de los componentes gráficos comunes en las interfaces gráficas son [5]:

- **Apuntador:** símbolo que aparece en la pantalla y que se desplaza para seleccionar objetos y comandos.
- **Dispositivo apuntador:** dispositivo, como el ratón, que permite seleccionar objetos de la pantalla.
- **Escritorio:** Es el área de la pantalla donde los íconos son agrupados.
- **Íconos:** Pequeñas imágenes que representan comandos, archivos o ventanas.

Los íconos en el contexto de las interfaces gráficas son signos esquemáticos que representan algún tipo de fichero, carpeta, aplicación, o dispositivos de un sistema informático. [6] Los íconos son elementos muy importantes dentro de las GUIs ya que son idóneos para transmitir conceptos en menos espacio del que podría describir una palabra a través del lenguaje verbal.

- **Menús.** Opciones o comandos agrupados en forma de listas.

Los menús son listas de comandos, atributos, o cualquier tipo de elementos, agrupados de forma estructurada normalmente inscrito dentro de una barra de menús o de un área específica en la interfaz. [6] Usualmente los menús esquematizan una organización de elementos de forma jerárquica por niveles, se representan en forma de listas, tanto de forma horizontal como vertical, a través de los mismos se accede a cada uno de los subelementos de cada elemento del menú. Los menús se pueden clasificar en menús contextuales, menús de navegación, menús jerárquicos y menú de inicio.

- **Ventanas.** Son áreas que dividen la pantalla, y en cada una de ellas puede ejecutarse un programa o abrir un archivo diferente.

Las ventanas son recursos interactivos usados para la visualización, jerarquización y navegación de la información en un interfaz gráfico de usuario. Constituye un marco, a través del cual es posible visualizar y manipular información del sistema. [6] A través de las mismas se pueden visualizar un conjunto de imágenes, aplicaciones, íconos, menús, sobre los cuales es posible realizar diversas acciones. Han sido definidas dos tipos generales de ventanas, las ventanas de aplicación y las ventanas de ficheros.

1.2.2 Características de las GUIs.

En la actualidad, una buena interfaz gráfica de usuario debe contar con un conjunto de características [7]:

- Posee un dispositivo apuntador (típicamente un ratón).
- Promueve la consistencia de la interfaz entre programas.
- Sigue el paradigma de la interacción objeto-acción.
- Permite la transferencia de información entre programas.
- Se puede manipular en la pantalla directamente los objetos y la información.
- Provee elementos de interfaz estándar como menús y diálogos.
- Existe una muestra visual de la información y los objetos (íconos y ventanas).
- Proporciona respuesta visual a las acciones del usuario.
- Existe información visual de las acciones y modos del usuario/sistema (menús, paletas).
- Existen controles gráficos (widgets) para la selección e introducción de la información.
- Proporciona flexibilidad en el uso de dispositivos de entrada (teclado/ratón).

1.2.3 Ventajas del uso de las GUIs. [2]

- La asimilación del contenido por parte del usuario es rápida y clara: el usuario al acceder al sistema es capaz de familiarizarse de forma amena con el mismo, le da sensación de seguridad y realismo, permitiéndole una rápida comprensión.

- **Contenidos accesible:** la información que se muestra está disponible para que el usuario acceda a esta en el momento que lo necesite, sin grandes contratiempos.
- **Usuarios sin experiencias pueden aprender el uso del sistema rápidamente:** el usuario no tiene que ser experto en la materia para su interacción con el sistema.
- **Retroalimentación garantizada:** le permite al usuario navegar con más facilidad, siempre es posible volver atrás.
- **Disponibilidad y portabilidad:** se debe poder disponer de la información cuando se necesite, siempre que sea personal autorizado. Así mismo permite ser utilizado en múltiples ordenadores sin que necesiten de modificaciones de importancia.

1.3 Diseño de interfaces gráficas de usuario.

El diseño elegido para un sistema debe ser el apropiado para la comunidad de usuarios a la que va dirigido y las tareas que van a realizar. Un mismo diseño puede ser apropiado para unos usuarios y poco apropiado para otros, eficiente para unas tareas e ineficiente para otras. **[8]** Lo significativo es que los usuarios finales se sientan a gusto y seguros, o sea, que sientan que pueden controlar el sistema. El diseño debe invitar a los usuarios a explorar sus opciones y capacidades, permitiéndoles que puedan aprender fácilmente a manejarlo.

Actualmente, el proceso del desarrollo de una interfaz se concibe como un ciclo que consta de 4 etapas. **[9]** Mientras se disponga de tiempo, se deben de hacer tantos ciclos de mejoramiento como nos sea posible.

- Diseño.
- Implementación.
- Medición.
- Evaluación.

Diseñar la interfaz de usuario.

Normalmente, esta etapa se conoce como análisis de sistemas, ya que en esta fase se definen los objetivos de usabilidad del programa, las tareas del usuario, los objetos y acciones de la interfaz, los íconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas. Al diseñar una interfaz gráfica de usuario, es necesario especificar a qué audiencia va a llegar. ¿Qué es lo que tratan de lograr al utilizar su aplicación? Esto se puede desglosar en tareas o actividades.

En esta fase, el modelado del diseño orientado a objetos y los diagramas de entidad-relación pueden ser utilizados para transmitir y comprender cierta funcionalidad en la interfaz. Algunos de los temas necesarios para realizar el modelo del diseño son los lenguajes de programación que se utilizarán para desarrollar la interfaz y la cantidad de espacio físico de almacenamiento que se necesita.

Implementación.

Es interesante realizar un prototipo previo, una primera versión del programa que se realice rápidamente y permita visualizar el producto para ser probado antes de codificarlo definitivamente. El nivel de sofisticación del prototipo debería incrementarse a lo largo del proceso de diseño de interfaces de usuario. La información recolectada durante las tareas de análisis del sistema y la especificación de los requisitos del usuario constituyen los datos claves para el proceso de prototipación. [10]

Medición y Evaluación.

En esta etapa se prueban las funcionalidades de la interfaz y se tratan de diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. En algunos casos, puede ser beneficioso solicitar a los usuarios reales que prueben el prototipo. A estas pruebas se les conoce como pruebas de aceptación, su objetivo es comprobar si el cliente está satisfecho con el producto desarrollado y si este producto cumple con sus expectativas, en términos de los errores que genera y de la funcionalidad que suministra.

Es muy recomendable que las pruebas de aceptación se realicen en el entorno en que se va a explotar el sistema (incluido el personal que lo maneje), y a medida que el usuario realiza las pruebas se pueden

Capítulo 1: Fundamentación Teórica

documentar los problemas que se van produciendo. Se deben realizar pruebas de usabilidad al producto, en suma es importante, comparar los resultados obtenidos con los esperados.

Preece, define la evaluación del diseño como la “reunión de datos sobre la usabilidad de un diseño o un producto por parte de un grupo de usuarios específicos para una actividad en concreto dentro de un entorno determinado de trabajo” y como “una parte integral del proceso de diseño centrado en el usuario”.

[11]

El proceso de desarrollo de las GUI esta determinado en 4 etapas, las cuales se detallaron a lo largo de este epígrafe. En la siguiente tabla se expone un resumen de las diferentes actividades que se realizan en cada una de las etapas del desarrollo.

Diseño	<ul style="list-style-type: none">• Análisis de requerimientos del producto.• Análisis de las tareas.• Conocimiento del usuario.• Generación de posibles metáforas y análisis de tipo de diálogo.• Revisión de posibilidades para la implementación.
Implementación	<ul style="list-style-type: none">• Generación de prototipos (profundos o amplios, para investigación general o de ajustes).• Desarrollo de la aplicación, sitio o sistema.
Medición (Test de usabilidad)	<ul style="list-style-type: none">• Planificación (desarrollo del plan, definición de las medidas, selección de participantes, formación de observadores, preparación de los materiales).• Test (prueba piloto, tests con usuarios).
Evaluación	<ul style="list-style-type: none">• Conclusión (análisis de los datos, elaboración del informe, resultados y recomendaciones).• Comparación contra estándares (internos y/o externos),

	<p>versiones anteriores del mismo producto y productos competidores.</p> <ul style="list-style-type: none">• Verificación de las diferencias.• Generación de nuevas metas.
--	---

Tabla 1 Resumen de las actividades de cada una de las etapas del desarrollo.

1.3.2 Principios para el diseño de interfaces gráficas de usuario.

Una buena interfaz es aquella que le permite al usuario corregir los errores que pueda cometer, dándole una sensación de control, donde los usuarios ven rápidamente cómo alcanzar sus metas y comprenden cómo realizar su trabajo. Para el diseño de las Interfaces Gráficas de Usuarios existen un conjunto de principios, a tener en cuenta para que dicha interfaz tenga un diseño adecuado y a su vez una buena aceptación por parte de los usuarios.

Anticipación.

Las aplicaciones deberían intentar anticiparse a las necesidades y deseos del usuario. No esperes que el usuario busque o recuerde información o herramientas. Muestra al usuario toda la información y herramientas necesarias para cada etapa en su trabajo. [12]

Autonomía.

No existe autonomía en ausencia de control; y el control no se puede tener sin información suficiente. Comunicar el estado es fundamental para que el usuario responda apropiadamente con la información disponible. [12] El sistema siempre debe mantener informado al usuario acerca de lo que está pasando, a través de una retroalimentación apropiada dentro de un plazo razonable. La interfaz debe ser capaz de posibilitarle al usuario una idea aproximada del estado del sistema, de una forma sutil. Este principio es muy importante, ya que para lograr una interfaz gráfica de usuario apropiada y de buena tolerancia por parte del usuario a quien va dirigida dicha interfaz, el usuario final debe sentirse sumergido en un ambiente flexible, donde sea capaz de aprender rápidamente el uso de la aplicación.

Eficacia del usuario.

Busca la productividad del usuario, no del ordenador. [12] Es importante a lo largo del proceso de desarrollo de la interfaz hacer de la productividad del usuario el objetivo principal. Se deben de escribir mensajes de ayuda concisos, donde un buen texto ayude mucho en la comprensión y resolución de algunos de los problemas que se les puedan presentar a los usuarios cuando estén interactuando con la interfaz.

Percepción del Color.

Básicamente trata la idea de que se deben utilizar además de los colores convencionales en las Interfaces de Usuario, otros mecanismos secundarios para proveer la información a aquellos usuarios con problemas en la visualización de colores. [10] Para nadie es un secreto que el color es uno de los elementos más complejos a la hora de intentar diseñar una interfaz gráfica correcta, además de ser una herramienta potente de comunicación si se usa de forma adecuada.

La simplicidad es importante en el diseño de interfaces a color. Existe una simplicidad inherente en el color, la cual debería ser usada cuando se desarrolla el diseño. Los cuatro colores fisiológicamente primarios son el rojo, el verde y el azul. Estos colores son fáciles de aprender y recordar. Vinculando significados prácticos e intuitivos a estos colores simples cuando se diseña una pantalla, el diseñador de la interfaz enriquece el desarrollo del usuario con un modelo mental efectivo. [13]

La claridad es también una pauta importante para usar color. Experimentos han mostrado que el tiempo de búsqueda para encontrar una pieza de información es disminuido si el color de esta pieza es conocido por anticipado, y si el color sólo se aplica a esa pieza. Los colores de interfaz estandarizados deberán de ser establecidos y usados a través del desarrollo de la GUI. El uso de colores claros y simples puede facilitar a los usuarios la búsqueda de piezas de información de forma rápidamente y eficientemente. El color ha probado ser superior al blanco y negro por la efectividad en el tiempo de proceso de información y por el rendimiento de memoria. La estética y lo atractivo de la interfaz son inherentemente aumentados por el uso del color. [13]

Ley de Fitt.

Esta es la ley más básica y más conocida de las leyes del diseño de interfaces de usuario. Esta ley dice que cuanto más grande y más cercano al puntero del ratón es un objeto, más sencillo es el hacer click sobre él. Esta afirmación es de sentido común, pero muchas veces es ignorado completamente en el diseño de interfaces. [14]

Utiliza objetos grandes para las funciones importantes. [12] Se recomienda que se utilicen objetos grandes para resaltar la ejecución de las funcionalidades más importantes de la aplicación. Este principio exhorta a que los controles más utilizados en la aplicación, deben de ser los más grandes y de mayor distinción. Plantea, además, que nunca se deben colocar los controles a un píxel de distancia del borde de la pantalla.

Interfaces Explorables.

De manera resumida y de una forma más clara, podemos decir que la GUI debe permitirle al usuario realizar la operación inversa de cualquiera acción que pueda efectuar dentro de la interfaz y que pueda implicar algún riesgo en el trabajo que se esté realizando, ya que los usuarios a menudo cometen errores cuando utilizan el sistema, es por ello que la interfaz debe de darle la posibilidad al usuario de explorar el sistema sin temor a cometer errores.

Haz que las acciones sean reversibles. [12] Para dar cumplimiento a este principio es recomendable la anexión de un comando para que el usuario pueda deshacer todas las operaciones que realice, esto sustituiría la necesidad de contar con diálogos de confirmación para cada acción que se realice en el sistema.

Dale al usuario caminos bien señalizados; luego deja que se metan monte a través. [12] No se debe de encerrar al usuario en un único camino, aunque se le debe de guiar por el mejor de los caminos, ya que esto le facilita a los nuevos usuarios o aquellos que solo quieren terminar de desarrollar alguna actividad dentro del sistema, hacerlo de una forma rápida y con el mínimo de esfuerzo, aunque siempre hay que dar posibilidades abiertas a los usuarios que deseen explorar el sistema.

Legibilidad.

En este principio deben tenerse en cuenta algunos aspectos como son: la consistencia y la tipografía. Los comandos y menús deben de tener el mismo formato. La consistencia es un factor importante dentro de la legibilidad ya que las interfaces consistentes reducen el tiempo de aprendizaje del usuario. El tamaño de las fuentes debe ser lo suficientemente grande como para poder ser leído en monitores estándares y es importante hacer clara la presentación visual de objetos, evitando la presentación de información excesiva.

Se deberá usar un número reducido de tipos de letra que deben ser legibles, claros y singulares. Usar colores de gran contraste para el texto y el fondo. [15] El texto que se represente en la interfaz gráfica de usuario debe tener un alto contraste, la utilización de colores como el texto en negro sobre fondo blanco o amarillo claro. No debe mezclarse más de dos o tres tamaños de fuente y estilos (subrayado, negrita, cursiva) en una ventana, debido a que esto se verá poco profesional y distraerá al usuario de la información que se transmite.

Consistencia.

Es tan importante ser visualmente inconsistente con los objetos que se comportan de forma distinta, como ser consistente con los que se comportan de igual manera. Evita la uniformidad. Haz que los objetos que se comportan distintos parezcan distintos. [12]

A menudo, algunos objetos en la interfaz se consideran visibles, porque no parecen controles y es casi imposible que el usuario descubra que se puede interactuar con esos objetos. Por lo tanto, si se decide utilizarlos, deberá aparecer muy claro su significado. La interfaz de usuario se mantiene en concordancia con el ambiente de trabajo. [10] Es necesario mantener su consistencia si no queremos que la gente se pase el día averiguando que se hace con estos objetos.

1.4 Arquitectura de la Información.

La arquitectura de información (IA) tiene como propósito la búsqueda de los mejores diseños para la presentación de la información y su posterior comprensión, así como la usabilidad, que estudia el conjunto

de características del diseño y funcionamiento de una interfaz de usuario, para obtener una correcta operación y comprensión de los contenidos, son disciplinas cuya actividad está dirigida a lograr la máxima satisfacción del usuario durante el proceso de interacción con los productos de información. [16]

Con respecto a la AI, Wurman se refirió a ella como: "... La ocupación profesional emergente del siglo XXI, dedicada a las necesidades de la era, enfocada a la claridad, entendimiento humano y la ciencia de organización de la información...". Edward Tufte, por su parte, la define como "... el diseño de la presentación de la información para facilitar su entendimiento...". [17]

La arquitectura de información se ocupa del diseño estructural de los sistemas de información, su problema central es la organización, recuperación y presentación de información mediante el diseño de ambientes intuitivos. [16]

1.4.2 Usabilidad.

El grado de usabilidad de un sistema interactivo es un aspecto relacionado con la interfaz de usuario y es inversamente proporcional al tiempo que malgastan los usuarios de dicho sistema intentando averiguar el alcance de qué hace o dónde está una determinada funcionalidad. Esta característica hace, por tanto, referencia a la rapidez y facilidad con que las personas llevan a cabo sus tareas a través del uso del producto con el que están trabajando.

Un producto se considera fácil de aprender a usar en términos del tiempo que toma el usuario para llevar a cabo su objetivo, el número de pasos que tiene que realizar para ello y el éxito que tiene en predecir la acción apropiada para llevar a cabo. Para desarrollar productos usables hay que entender los objetivos del usuario, hay que conocer los trabajos y tareas del usuario que el producto automatiza o modifica.

La Organización Internacional para la Estandarización ofrece dos definiciones de usabilidad [19]:

- La usabilidad se refiere a la capacidad de un software de ser comprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. (ISO/IEC 9126).
- La usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios en un contexto de uso específico. (ISO/IEC 9241).

Capítulo 1: Fundamentación Teórica

En la definición podemos observar que la usabilidad se compone de dos tipos de atributos:

- **Atributos cuantificables de forma objetiva:** como son la eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea.
- **Atributos cuantificables de forma subjetiva:** como es la satisfacción de uso, medible a través de la interrogación al usuario, y que tiene una estrecha relación con el concepto de Usabilidad Percibida.

El concepto de usabilidad puede ser definido, además de como atributo de calidad de una aplicación, consecuentemente, como disciplina o enfoque de diseño y evaluación. A partir estas conceptualizaciones, podemos obtener los principios básicos en los que se basa la usabilidad, conocidos como atributos de usabilidad o características orientadas al usuario [20]:

- **Facilidad de aprendizaje:** Cuán fácil es aprender la funcionalidad básica del sistema y ser capaz de realizar correctamente la tarea que desea realizar el usuario. La facilidad de aprendizaje se mide normalmente por el tiempo empleado en el sistema hasta ser capaz de realizar ciertas tareas en menos de un tiempo dado, comparado por el tiempo empleado habitualmente por los usuarios expertos. Este atributo es muy importante para usuarios noveles. Son los usuarios y no los diseñadores o los desarrolladores, los que determinan cuándo un producto es fácil de usar.
- **Eficiencia:** El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Este atributo busca la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez. Nótese que eficiencia del software en cuanto a su velocidad de proceso no implica necesariamente eficiencia del usuario.
- **Recuerdo en el tiempo:** Para usuarios intermitentes (que no utilizan el sistema regularmente) es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez. Este atributo refleja el recuerdo acerca de cómo funciona el sistema que mantiene el usuario, cuando vuelve a utilizarlo tras un período de no utilización.

- **Tasa de errores:** Este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario, y pueden verse como un fracaso en la transmisión al usuario del modo de hacer las cosas con el sistema.
- **Satisfacción:** éste es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.

1.4.3 Accesibilidad.

Un concepto íntimamente ligado al de usabilidad es el de accesibilidad. Este ya no se refiere a la facilidad de uso, sino a la posibilidad de acceso. En concreto a que el diseño, como prerrequisito imprescindible para ser usable, posibilite el acceso a todos sus usuarios, sin excluir a aquellos con limitaciones individuales, discapacidades, dominio del idioma o limitaciones derivadas del contexto de acceso (hardware).

La accesibilidad es la cualidad de accesible, un adjetivo que se refiere a aquello que es de fácil acceso, trato o comprensión. El concepto se utiliza para nombrar al grado en que todas las personas, más allá de sus capacidades físicas o técnicas, pueden utilizar un cierto objeto o acceder a un servicio. [18] Muchas veces cuando se pretende reducir la accesibilidad a una cuestión de corrección del código, olvidamos que muchos de los problemas de accesibilidad que puede presentar un diseño nada tienen que ver con su implementación, sino con su conceptualización arquitectónica.

Los seres humanos son diferentes entre sí y lo ideal sería que todas las interfaces de usuario pudieran acomodarse a esas diferencias, de tal modo que cualquier persona fuera capaz de utilizarlas sin problemas, sin que nadie se vea limitado en el uso de algo por causa de esas diferencias personales. Es necesario evitar diseñar solamente atendiendo a características de grupos de población específicos, imponiendo barreras innecesarias que podrían ser evitadas prestando más atención a las limitaciones de estos. El lenguaje de la interfaz debe ser claro, sencillo y directo, adaptado al nivel comprensivo del usuario, evitando anglicismos y jerga informática.

1.5 Software de Visualización Médica para Imagenología.

El análisis de las estructuras internas del cuerpo sólo ha sido posible durante el último siglo, con el descubrimiento de los rayos-X. Otra revolución llegó no hace más de 30 años con la utilización de las tomografías computarizadas y resonancias magnéticas que permiten el estudio y análisis en tres dimensiones del cuerpo humano.

La Imagenología médica se ha convertido en una de las aplicaciones más importantes del procesamiento y visualización de imágenes en realidad virtual. Este campo de la medicina, cuyo desarrollo empezó hace unos cuantos años se basaba exclusivamente en la lectura de imágenes en 2D y 3D por parte de los especialistas. Esta rama de la medicina se encuentra en una constante evolución hacia el desarrollo de tecnologías que permitan realizar segmentaciones y reconstrucciones tridimensionales.

1.5.1 Software Internacionales.

3D-Doctor, es una avanzada de modelado 3D. [21] Este software proporciona a los médicos, investigadores y a las demás personas que interactúen con el software la posibilidad de crear modelos 3D de malla poligonal a partir de imágenes tomadas de tomografía, resonancia magnética y otras imágenes volumétricas. Una de las ventajas que presenta este software, se pueden cargar diferentes formatos de archivos de imágenes como son: DICOM, TIFF, GIF, JPEG, PNG, BMP, PGM, MRC, RAW.

Este software también crea modelos 3D de superficie y el volumen de representación 2D de imágenes de sección cruzada en tiempo real. Exporta los formatos de malla poligonal STL, DXF, IGES, 3DS, OBJ, VRML, PLY, XYZ y otros formatos de archivo para planificar actividades como quirúrgicas, simulación, análisis de elementos finitos. Da la posibilidad de calcular el volumen 3D, la distancia, la densidad, el histograma de la imagen y realizar otras mediciones 3D para el análisis cuantitativo. Puede fusionar imágenes, corregir espesor desigual, cambiar el tamaño del volumen y rotar las imágenes sobre los diferentes ejes de rotación. 3D-Doctor está siendo utilizado actualmente por los principales hospitales, escuelas médicas y organizaciones de investigación de todo el mundo.

Uno de los principales inconvenientes es que solo se puede ejecutar en computadoras que tengan instalado el Sistema Operativo Windows, incluyendo Windows NT/2000/XP, o versiones más recientes de

Capítulo 1: Fundamentación Teórica

Windows. Este software cuenta con una versión libre, llamada 3D-DOCTOR Viewer, que es para la visualización y efectos de presentación solamente. Si se quiere hacer el procesamiento de imágenes, edición y creación 3D, se tiene que utilizar la versión comercial.

Amira es un sistema de visualización y modelado 3D, permite visualizar conjuntos de datos científicos de varias áreas de aplicación, como por ejemplo, medicina, biología, química, física, o ingeniería. Amira es un software modular orientado a objetos. Los componentes básicos del sistema son módulos y objetos de datos. Se usan los módulos para visualizar los objetos de datos o para realizar algunas operaciones computacionales en ellos. Las conexiones entre los módulos de Amira representan las dependencias entre los objetos de datos y sus métodos. [22]

Amira es la solución de software que satisface incluso las necesidades más exigentes de trabajar con datos de la imagen clínica o preclínica, datos nucleares, las imágenes de microscopía óptica o electrónica, modelos modulares, el vector y el flujo de datos, los datos de simulación en los modelos de elementos finitos, y el tipo de imágenes multidimensionales, vector, tensor, y los datos de la geometría. La arquitectura de Amira está organizada en módulos de software funcional, que consiste en un paquete básico más un complemento adicional, que permiten al usuario optimizar las inversiones mediante la personalización del producto para satisfacer las necesidades exactas.

Algunas de las características de este software es que se encarga de analizar instrumentos de medición, estadísticas de tejidos, análisis molecular y detección molecular. Es capaz de importar y exportar formatos de mapa de bits estándar, imagen de microscopía y de imagen médica. Visualiza volumen rendering, fusión de imágenes, orto y corte oblicuo, asolíneas e isosuperficies. Además, personaliza opción de malla, microscopía y esqueletización.

3DSlicer, es un software para la visualización y análisis de imagen. [23] Posee una gran variedad de usos médicos, entre los que podemos mencionar la esclerosis múltiple, lupus, cáncer de la próstata, esquizofrenia, biomecánica ortopédica, enfermedad cardiovascular y neurocirugía. Entre las diferentes funcionalidades de este software podemos destacar la segmentación, el registro y la visualización tridimensional de las imágenes. Se apoya en los diferentes formatos estándares del archivo de la imagen.

Este software se utiliza para presentar la información que se usa para determinar flujo de la sangre en el cerebro relacionado con la actividad de la médula de los nervios o espinal. 3DSlicer, se basa en VTK, una biblioteca gráfica que proporciona un interfaz de alto nivel. Permite la lectura de imágenes DICOM y de otros formatos, la visualización interactiva de imágenes, de los modelos 3D, y de los rendering del volumen. Admite la segmentación automática de imágenes, así como el análisis y la visualización de los datos de la proyección de imágenes.

1.5.2 Software Nacional.

Entre los software médicos más usados en el ámbito nacional se encuentra el IMAGIS, desarrollado en el Centro de Biofísica Médica para la telemedicina y la transmisión de imágenes médicas multimodales en el Sistema Nacional de Salud Cubano. Este sistema de visualización cubano permite dentro de sus funcionalidades la adquisición, procesamiento, almacenamiento, recuperación e impresión de imágenes médicas obtenidas en equipos de diferentes modalidades y fabricantes como Resonancia Magnética, Ultrasonido, Tomografía Axial Computarizada, Radiología Computarizada, entre otras.

Permite la creación y manipulación de bases de datos de imágenes médicas utilizando para ello el formato DICOM 3.0. Muestra imágenes DICOM obtenidas por diferentes vías utilizando la información de interpretación fotométrica obtenida en la modalidad de origen, así como los parámetros de contraste adecuados. Brinda diversos métodos de procesamiento para el mejoramiento de imágenes, ampliación, compresión, mediciones y anotaciones. [24]

1.5.3 Estudio de Homólogos.

Después de haber visto en el epígrafe anterior diferentes características de algunos de los software para la visualización de imágenes médicas existentes en el ámbito nacional e internacional, presentamos en la tabla 2 un estudio comparativo de estos software basándonos en los criterios de usabilidad, licencia comercial, entre otros.

Capítulo 1: Fundamentación Teórica

Crterios	IMAGIS	3DSlicer	Amira	3D-Doctor
Eficiencia	Alta	Alta	Alta	Alta
Licencia Comercial	Propietario	Libre	Propietario	Propietario
Portabilidad	Windows	Multiplataforma	Windows	Multiplataforma
Mantenibilidad	No modificable	Código Abierto	No modificable	No modificable
Confiabilidad	Alta	Alta	Alta	Alta

Tabla 2 Estudio comparativo de software para la visualización de imágenes médicas.

Como se muestra en la tabla anterior, algunos de los problemas que presentan actualmente los software para imagenología son la portabilidad y la licencia comercial, que trae consigo que en la mayoría de estos software no se pueda modificar el código fuente. Sin embargo, vale destacar que no se está buscando un software para la visualización de imágenes médicas, sino que se están viendo un grupo de estos software para de ellos tomar las principales funcionalidades para desarrollar nuestra propuesta; por consiguiente a continuación se muestra una tabla donde se comparan estos software respecto a las funcionalidades que presentan.

Funcionalidades	Imagis	3D Slicer	Amira	3D-Doctor
Navegación Tridimensional	No	Si	Si	Si
Fusión de Imágenes	No	Si	Si	Si
Segmentación	No	Si	Si	Si
Reconstrucción 3D	No	Si	Si	Si
Filtros	No	Si	Si	Si
Cálculo de Volumen	Si	Si	Si	Si

Tabla 3 Comparación de funcionalidades software de visualización de imágenes médicas.

Como vimos en la tabla anterior estos software presentan una serie de funcionalidades que son comunes para muchos de ellos, las mismas se tendrán en cuenta en la construcción del diseño de la solución propuesta.

1.6 Biblioteca gráfica de desarrollo en sistemas GNU/Linux.

1.6.1 Qt.

Qt es una biblioteca de clases C++ creada para construir interfaces de usuario. Proporciona la mayoría de las entidades gráficas que verá empleadas en una aplicación KDE: menús, botones, regletas, etc. Las aplicaciones basadas en Qt tienen una buena respuesta y un uso aceptable de la memoria. [25] Fue desarrollada por la compañía noruega Trolltech y está totalmente orientada a objetos. La idea de Qt es la misma que la de Java: “Hazlo una vez, ejecútalo donde sea”, o sea, está diseñada con el fin de poder compilar un mismo programa en varias plataformas.

Esta biblioteca es mayormente utilizada en entornos de escritorio KDE, aunque también es perfectamente funcional en otros escritorios como Gnome, Windows y Mac OS. Hace uso del lenguaje C++ de forma nativa, además existen interfaces para C, Python (PyQt), Java (QtJambi), Perl (PerlQt), Gambas (Gb.Qt), Ruby (QtRuby), PHP (PHP-Qt), Mono (Qtoto), entre otros. Posee una doble licencia, una GPL para Aplicaciones de Código Abierto y otra comercial.

1.6.2 Gtk+

Gtk+ o The GIMP Toolkit es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI), principalmente para los entornos gráficos GNOME, XFCE y ROX aunque también se puede usar en el escritorio de Windows, MacOS y otros. [26] Esta biblioteca gráfica usa a C como lenguaje de programación base, aunque sus creadores usan un paradigma orientado a objeto; bajo la licencia LGPL, Gtk+ es software libre y de código abierto.

Dentro del conjunto de ventajas que nos ofrece Gtk podemos mencionar que provee íconos predefinidos y ventanas de diálogos comunes. Esta biblioteca permite definir aceleradores y los widgets soportan rendering personalizados. Hay soporte GTK para muchos otros lenguajes, incluyendo C++, Guile, Perl, Python, TOM, Ada95, Objective C y Eiffel. Gtk+ se basa en varias bibliotecas del equipo de Gtk+ y de GNOME. [27]

Capítulo 1: Fundamentación Teórica

- **Glib:** Biblioteca de bajo nivel estructura básica de GTK+ y GNOME. Proporciona manejo de estructura de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución como ciclos, hilos, carga dinámica o un sistema de objetos.
- **GTK:** Biblioteca la cual realmente contiene los objetos y funciones para crear la interfaz de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.
- **GDK:** Biblioteca que actúa como intermediario entre gráficos de bajo nivel y gráficos de alto nivel.
- **ATK:** Biblioteca para crear interfaces con características de una gran accesibilidad muy importante para personas discapacitadas o minusválidas. Pueden usarse herramientas como lupas de aumento, lectores de pantalla, o entradas de datos alternativas al clásico teclado o mouse.
- **Pango:** Biblioteca para el diseño y renderizado de texto, hace hincapié especialmente en la internacionalización. Es el núcleo para manejar las fuentes y el texto de GTK+2.

1.7 Lenguaje de programación.

Para el desarrollo de interfaces gráficas de usuario en sistemas GNU/Linux existen diferentes lenguajes de programación, tales como: C#, C++, Java. La elección de uno de ellos debe sustentarse principalmente en las particularidades de la aplicación que se quiera desarrollar. A partir de esto se eligió C++, teniendo en cuenta sus características y ventajas. El C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Permite la reutilización del código de una manera más lógica y productiva y para el uso de plantillas o programación genérica (templates). El código escrito en C++ es muy corto en comparación con otros lenguajes, sobretodo porque en este lenguaje es preferible el uso de caracteres especiales que las "palabras clave". El código resultante de una compilación en C++ es muy eficiente, gracias a su capacidad de actuar como lenguaje de alto y bajo nivel.

Ventajas: [28]

- **Difusión:** al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, páginas web, etc. dedicado a él.
- **Versatilidad:** C++ es lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- **Portabilidad:** el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- **Eficiencia:** C++ es uno de los lenguajes más rápidos en cuanto a ejecución.
- **Herramientas:** existe una gran cantidad de compiladores, depuradores, librerías, etc.

1.7.1 IDE de programación.

Las ventajas principales de un IDE radican en la automatización de algunas tareas asociadas al desarrollo de aplicaciones y al soporte para algunas funcionalidades como el auto completamiento de código y la depuración, contribuyendo a evitar pérdidas de tiempo.

Aunque existen varios candidatos entre los que están Eclipse, Visual Studio, Borland C++ Builder 6, Code Block y Qt Creator; se escogió este último porque es bastante completo, moderno, potente, fácil de manejar, eficiente, abierto y gratuito, que permite el desarrollo rápido de aplicaciones en entornos MS Windows, Mac OS y Linux, muestra los errores en línea mientras se escribe, posee integración para control de versiones, además de las siguientes características:

- Utiliza el lenguaje de programación orientado a objetos C++.
- Se basa en Qt, una librería multiplataforma y gratuita para la creación de interfaces gráficos.
- Permite realizar programación visual y programación dirigida por eventos.

Características avanzadas de Qt Creator:

- Qt4 Proyecto de Generación Wizard: Este asistente permite al usuario generar un proyecto para una aplicación de consola, una aplicación GUI, o una librería C++.
- Qt Ayuda de Integración: Se puede acceder a toda la documentación de Qt fácilmente haciendo clic en el botón de Ayuda.
- Qt Integración de Diseño: formas de interfaz de usuario puede ser diseñado en Qt Creator, simplemente con hacer doble clic en la interfaz de usuario de un archivo dentro del explorador de proyectos para poner en marcha la integración.
- Localizador: Una herramienta de navegación muy potente que permite al usuario localizar los archivos y las clases utilizando pulsaciones mínima.

1.8 Metodología de desarrollo.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. [29] En cualquier ámbito de ingeniería hay una fractura entre los responsables de analizar y definir los problemas (necesidades), y los expertos en proveer soluciones (tecnología).

Las metodologías nacen para intentar solucionar este conflicto. Su propósito es establecer un contrato entre todos los integrantes de un proyecto para conseguir la solución más eficaz, sin retrasos y que cumpla con las expectativas de los clientes, con los recursos disponibles.

1.8.1 RUP (Rational Unified Process).

El proceso unificado conocido como RUP, es una metodología que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. [30] Se optó por el uso de esta metodología debido a que se puede ver la evolución del software en cuatro fases (Inicio, elaboración, construcción, transición), al final de las cuales, y tras una serie de iteraciones, establece objetivos a alcanzar bien definidos. Esto permite encausar el esfuerzo de los recursos humanos en términos de habilidades, experiencias, aptitudes y

Capítulo 1: Fundamentación Teórica

capacidades a asumir roles específicos con responsabilidades bien definidas. RUP establece lo que denomina buenas prácticas como forma de trabajo adecuada para la consecución de objetivos que se pueden ir perfeccionando. Documenta los requisitos y los cambios de los requisitos, analizando el impacto de cambios antes de aceptarlos.

Modela visualmente el software empleando el estándar UML. Verifica la calidad de los productos del software asegurando que cumple los estándares. El ciclo de vida de RUP se caracteriza por ser **[31]**:

Dirigido por Casos de Uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de estos.

Centrado en la arquitectura: La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Este concepto incluye los aspectos estáticos y dinámicos más significativos del sistema y se refleja en los casos de uso, pues cada producto tiene tanto una función como una forma, ninguna es suficiente por sí sola.

Iterativo e incremental: Resulta práctico dividir el trabajo en partes más pequeñas o mini-proyectos, los cuales no son más que iteraciones que resultan en un incremento. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software.

1.9 Lenguaje de Modelado.

A lo largo de los años, el desarrollo de los proyectos de software causan bastantes confusiones y malas interpretaciones en los requerimientos de los clientes y usuarios, en parte debido a la abundancia de notaciones, metodologías y conceptos que hace que los desarrolladores de sistemas no se pongan de acuerdo en que es lo que realmente están elaborando.

1.9.1 UML (Unified Modeling Language).

Lenguaje de modelado UML es un lenguaje que permite visualizar, especificar, construir y documentar un sistema de forma gráfica, ofreciendo un plano para describir el sistema mediante un modelo estándar, incluyendo aspectos conceptuales como proceso del negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Algunas de las propiedades de UML como lenguaje de modelado estándar son [32]:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividades actuales y futuras.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

1.10 Herramientas CASE.

Se puede definir a una herramienta CASE (en español, ingeniería de software asistida por computadora, en inglés, Computer Aided Software Engineering) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante los pasos del ciclo de vida de desarrollo de un software. [33]

Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación del código

automáticamente con el diseño dado, compilación automática y documentación o detección de errores. Optimizando la productividad en el desarrollo y mantenimiento del software, aumentando la calidad del software, mejorando la planificación de un proyecto, facilitando el uso de las distintas metodologías propias de la ingeniería del software, ayudando a la reutilización del software, portabilidad y estandarización de la documentación.

1.10.1 Visual Paradigm.

Visual Paradigm es una herramienta que utiliza UML como lenguaje de modelado. Es considerada una herramienta muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Ofrece la capacidad de ingeniería directa (versión profesional) e inversa. Es un producto de probada calidad, soporta varios idiomas, es fácil de instalar y actualizar, y su licencia es gratuita y comercial. La herramienta está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios.

1.11 Conclusiones generales del capítulo.

En el presente capítulo, se presentó un estudio de algunos de los software para la visualización de imágenes médicas existentes en el mercado, llegando a conclusiones objetivas de las funcionalidades comunes en estos software, las cuales se tendrán en cuenta en la construcción del diseño de la solución. Se detalló el proceso de desarrollo de interfaces, el cual consta de 4 etapas: Diseño, Implementación, Medición y Evaluación, a partir del cual se obtiene una guía de las tareas que se van a realizar durante el desarrollo de la propuesta.

Se abordan conceptos de temas referentes a la arquitectura de la información, accesibilidad y usabilidad, los cuales se pretenden incluir en el diseño de la interfaz, atendiendo a la tipografía que se utilizará y a la distribución de la información. Se presentó un estudio de algunos de los principios para el diseño de las interfaces gráficas de usuario, ya que se procura orientar la propuesta al cumplimiento de estos principios, dígame interfaces explorables, ley de Fitt y percepción del color. Se identificó la metodología, IDE de

Capítulo 1: Fundamentación Teórica

desarrollo, el lenguaje de programación y la biblioteca grafica de desarrollo, ofreciendo las definiciones necesarias y precisas para comprender el porqué del uso de las mismas. Todo esto encaminado a obtener los elementos necesarios para dar solución al objetivo de esta investigación.

Capítulo 2: Características del Sistema.

Introducción.

En el presente capítulo se presenta la solución técnica del presente trabajo de diploma, se describe el dominio del problema y conceptos, contiene la descripción de los requerimientos del sistema tanto funcionales como no funcionales, se detallan y justifican los principales actores y se hace una descripción de los casos de usos que incluyen todo lo que el sistema debe hacer.

2.1 Solución Propuesta.

A partir del marco teórico analizado en el capítulo 1, se propone una el desarrollo de una interfaz gráfica de usuario para la visualización de imágenes médicas, basada en los principios básicos de diseño y que incluya dentro de sus requisitos las funcionalidades de algunos de los software para la visualización de imágenes medicas existentes en el mercado.

La interfaz es la parte del sistema con que el usuario interactúa y la que le facilita el acceso a las funcionalidades del sistema, es por ello que su diseño está orientado hacia los usuarios finales, ya que los mismos no están familiarizados con las ciencias de la computación, por lo que se puede decir que no están interesados en la parte interna de la aplicación (el código), sino en cómo se le muestra y cómo usarla.

2.1.1 Entregable de la Arquitectura de la Información.

Al realizar el estudio de las interfaces gráficas de usuario de algunos de los software para la visualización de imágenes médicas vistos en el capítulo anterior se produjo un prototipo de cómo posicionar la información para que los usuarios no se perdieran en la interfaz y así darle solución a los problemas de la accesibilidad y usabilidad, que como se mostró anteriormente son aspectos que se deben de tener en cuenta a la hora de diseñar una interfaz gráfica.

Capítulo 2: Características del Sistema

File	Edit	Image	View	Tool	Windows	Help
Open/Save		Tool Menu			Tool View	
					Control Panel	
Status Message						

Figura 1 Estructura de la Interfaz Gráfica de Usuario.

2.1.2 Funcionalidades.

La interfaz cuenta con un conjunto de funcionalidades, que se utilizan para visualizar, analizar y manipular imágenes. También, se contempla la posibilidad de resaltar algunos aspectos de las imágenes con el fin de facilitar el trabajo de los especialistas que lo requiera. Cada una de estas funcionalidades es controlada por un conjunto de parámetros, que pueden seleccionarse de manera interactiva.

Una de las tareas más importantes en el análisis de imágenes médicas es la segmentación, entendiéndose como tal al proceso de particionarlas según sus componentes estructurales más importantes. Una segmentación precisa es un requisito indispensable en la mayoría de las aplicaciones vistas en el capítulo anterior, y por ende, lo es también para la interfaz que se propone, que cuenta con esta funcionalidad que puede ser automática o manual.

Capítulo 2: Características del Sistema

El 3D Rendering es una de las funcionalidades que está presente en la mayoría de los software para imagenología, y por consiguiente en la solución propuesta, esta funcionalidad permite hacer la representación en superficie (surface rendering) y la representación en volumen (volume rendering). En la representación en superficie, la visualización y manejo de datos multidimensionales se basa únicamente en los límites del objeto y el despliegue de la superficie del objeto de interés que está aislado del resto del volumen. En la representación en volumen, se persigue la representación de múltiples estructuras con diferentes tipos de tejido.

Los filtros, que son otras de las funcionalidades importantes, son para realizar la presentación de imágenes DICOM u otras vistas reconstruidas. La interfaz ofrece una serie de filtros que se pueden aplicar en la ventana de visualización de imágenes, estos procesan la imagen renderizada píxel por píxel, con el objetivo de aumentar estructuras o reducir el ruido o incrementar la degradación de las imágenes.

Se pretende reunir en la interfaz un conjunto de funcionalidades fáciles de usar que permita realizar cálculos morfométricos como volúmenes, distancia y ángulos. Se cuenta con otras funcionalidades básicas en el tratamiento y manipulación de imágenes médicas como son el zoom a diferentes regiones de interés, rotar dichas imágenes en los correspondientes ejes (x, y, z).

2.1.3 Principios de Diseño.

El diseño de la interfaz orienta al usuario respecto al lugar donde se encuentre, mediante la barra de título. Los mensajes de ayuda, así como el del botón cerrar provee al usuario de una respuesta inmediata; el sistema cuenta con un área de mensaje de estado donde se muestra información sobre la tarea que el usuario está realizando en ese momento.

La interfaz propuesta posee una barra de herramienta donde aparecen las funcionalidades más usadas por el usuario; cuenta con un panel de control donde estarán ubicados los diferentes parámetros que se corresponden con la ejecución de las funcionalidades y los elementos incluidos en esta barra de herramientas poseen un tamaño razonable para que no resulte difícil dar clic sobre ellos. Los íconos utilizados para la representación de estas funcionalidades no se utilizan para otra cosa que no sea su finalidad prevista y los colores utilizados en los mismos son de gran contraste entre ellos y respecto al color de fondo. Se le da al usuario la posibilidad de cerrar la aplicación en el momento que así lo desee,

Capítulo 2: Características del Sistema

mostrándole un mensaje de confirmación de la acción en caso que el usuario no esté seguro del procedimiento.

La interfaz le da la posibilidad al usuario de acceder a la ayuda del sistema en caso que este se encuentre desorientado. La información se sitúa con sutileza para facilitar la interacción del usuario con la aplicación, con un tamaño de letra para los íconos y menús (MS Shell Dlg 2), tratando de variar al mínimo. Para el color de las letras se utiliza el negro que se distingue sobre el fondo gris y esto le aporta seriedad a las aplicaciones de este tipo. El diseño proporciona una serie de menús desplegados que solo se muestran cuando el usuario hace clic en uno de ellos; los mismos están visibles en todo momento y son accesibles desde el teclado.

2.1.4 Biblioteca gráfica.

A partir del estudio teórico realizado se optó por desarrollar la interfaz gráfica de usuario con la biblioteca gráfica de desarrollo Qt, debido a que la interfaz de programación de la misma está totalmente orientada a objetos y tienen una buena respuesta y un uso aceptable de la memoria. Además, la biblioteca Qt está concebida como multiplataforma y posee una doble licencia.

Hay que destacar que la selección de esta biblioteca se debe en gran medida a la amplia comunidad de desarrollo que le brinda soporte y porque la misma posee muchas funcionalidades de alto rendimiento. Qt tiene ventajas sobre otros frameworks o toolkits que proveen un recubrimiento para C++, debido a que esta implementada directamente en C++. La calidad en tiempo de ejecución es más alta; no necesita hacer uso de referencias innecesarias y además, no depende de ninguna otra biblioteca, aparte de sí misma [25].

Para desarrollar en Qt solo es necesario aprender una sola API para escribir aplicaciones que correrán en cualquier lado, ventaja que nos garantiza su portabilidad. Qt posee un conjunto completo de widgets estándar, y permite escribir controles personalizados. Se puede mencionar que existen numerosos foros en Internet donde brindan ayuda en línea y en los que los programadores exponen dudas o problemas que son resueltos la mayoría de las veces en muy corto tiempo.

2.2 Modelo de dominio.

Se decidió representar el negocio a través de un Modelo de Dominio (Modelo Conceptual), debido a la relativa simplicidad del entorno donde está enmarcado el sistema y el conocimiento que se posee acerca de su funcionamiento. El mismo es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de componentes de software [34].

2.2.1 Diagrama modelo dominio.

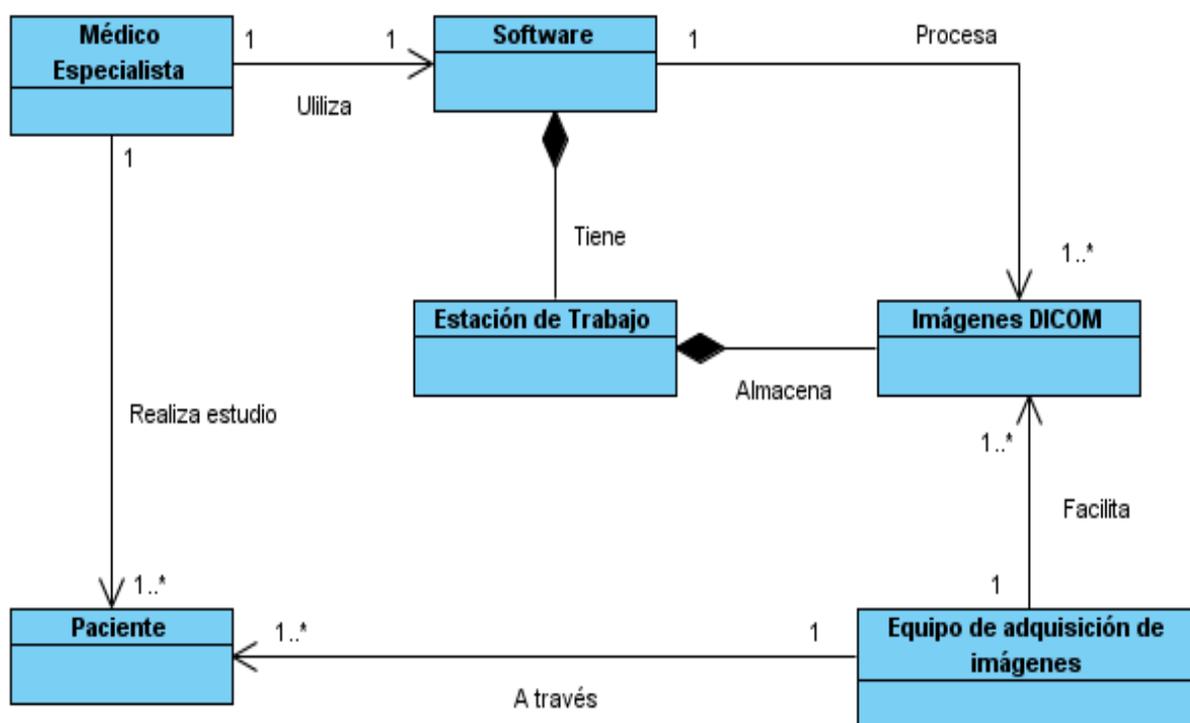


Figura 2 Diagrama Modelo de Dominio.

2.2.2 Descripción del Modelo de Dominio.

- El **Médico Especialista** en Imagenología es aquel que esté adecuadamente capacitado en el manejo y conocimiento de los métodos de Imagenología diagnóstica.
- Un **Software** es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Es una solución para la automatización de ciertas tareas complicadas como puede ser el diagnóstico a través de imágenes.
- Una **estación de trabajo** es un ordenador que facilita a los usuarios el acceso a los servidores y periféricos de la red. Las estaciones de trabajo ofrecen más alto rendimiento de lo que es normalmente encontrado en las computadoras personales, especialmente con lo que respecta al soporte de gráficos, el poder de procesamiento y habilidades multi-tareas.
- Las **Imágenes DICOM**, son un estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas.
- Los **equipos de adquisición de imágenes**, son los conocidos CT (Tomografía Axial Computarizada) y MRI (Resonancia Magnética).
- El **paciente** es aquel que recibe los servicios de un médico u otro profesional de la salud.

Al paciente se le realizan una serie de estudios por parte de los médicos, lo cual puede ser a través de tomografías computarizadas (TAC) o resonancias magnéticas (MRI). Estos estudios adquiridos están formados por un conjunto de imágenes DICOM. El médico especialista analiza las imágenes a través de un software de visualización para realizar un diagnóstico.

2.3 Levantamiento de los requisitos del sistema.

Es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se construya. Los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. Se clasifican en requisitos funcionales y no funcionales.

2.3.1 Requisitos funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Su meta principal es identificar y documentar las acciones que en realidad debe ejecutar el sistema para que cumpla con los objetivos planteados al inicio del presente trabajo de diploma.

De acuerdo con los objetivos propuestos el sistema debe ser capaz de:

RF1. Cargar imágenes DICOM.

RF2. Mostrar vistas axial, coronal y sagital.

RF3. Rotar imagen.

RF3.1 Rotar con cursor.

RF4. Escalar imagen.

RF4.1 Escalar con scroll

RF5. Deshacer acciones.

RF6. Rehacer acciones.

RF7. Aplicar Filtros.

RF8. Realizar mediciones sobre la imagen.

RF9. Calcular volumen.

RF10. Segmentar.

RF10.1 Segmentar automático.

RF10.2 Segmentar manual.

RF11. Reconstruir 3D.

RF11.1 Volumen Rendering.

RF11.2 Surface Rendering.

RF11.3 Planar Rendering.

2.3.2 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Requisitos no funcionales de Hardware.

- Tarjeta gráfica de 512 MB con soporte para textura 3D.
- Procesador Pentium 4 a 3.0 GHz o superior.
- Memoria RAM de 1GB de capacidad.
- Espacio en disco duro de 512 MB.

Requisitos no funcionales de Usabilidad.

- Las personas que interactúan con la interfaz gráfica serán los médicos especialistas, entrenado en el tipo de diagnóstico que se está visualizando en el sistema y deben haber sido capacitado en algún curso básico para la interacción con la aplicación.
- La herramienta deberá poseer una interfaz de navegación acorde y funcional, tanto para usuarios expertos, como para los que no tienen conocimientos profundos de Informática.

Requisitos no funcionales de Restricciones en el Diseño e Implementación.

- La interfaz fue desarrollada en el Sistema Operativo GNU/Linux Debian.

Capítulo 2: Características del Sistema

- Se utilizó Qt Creator como IDE de programación y Qt como biblioteca de desarrollo de Interfaces Gráficas de Usuarios en Sistemas Operativos GNU/Linux.
- El sistema será desarrollado orientado a objeto con la filosofía de software libre.

Requisitos no funcionales de Apariencia o Interfaz Externa.

- El usuario debe tener acceso por varias vías a una determinada funcionalidad.
- El sistema debe contener una interfaz sencilla e intuitiva y sea agradable al usuario.
- La herramienta debe utilizar como idioma principal el español, excepto aquellas palabras técnicas que no puedan ser traducidas.
- Utilizar botones que expresen su función, ya sea a través del ícono o el texto que lo identifica.
- Los colores de la interfaz serán agradables, sin mucho brillo que pueda afectar la vista del usuario que tiene sus ojos concentrados en la lectura.

Requisitos de Rendimiento.

- La herramienta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, al igual que la velocidad de procesamiento de la información.

2.4 Definición del actor del sistema.

Los actores representan entidades que interactúan con el sistema, un actor del sistema es aquel que se beneficia de con los resultados de las funcionalidades del mismo. En el sistema que se describe se identificó el siguiente actor:

Actor del Sistema	Justificación
Médico Especialista	Representa al usuario que va a hacer uso del sistema, teniendo la posibilidad de interactuar con las funcionalidades del mismo.

Tabla 4 Actor del Sistema.

2.5 Casos de uso del sistema.

La forma en que los actores usan el sistema es representada a través de los casos de usos. Estos últimos son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del actor. Los casos de usos identificados en el presente trabajo son enunciados a continuación:

2.5.1 Diagrama de Casos de usos del sistema.

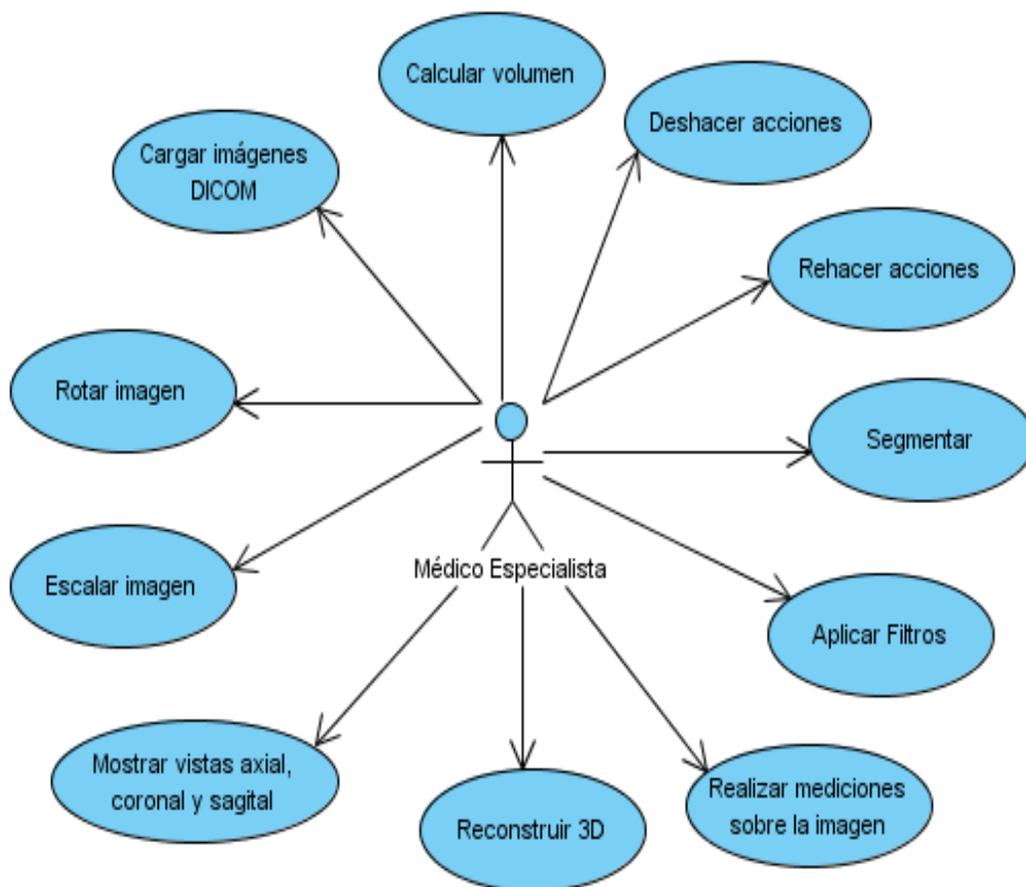


Figura 3 Diagrama de Casos de Usos del Sistema.

Capítulo 2: Características del Sistema

2.5.2 Descripción de los casos de uso del sistema.

Caso de Uso:	Cargar imágenes DICOM
Actores:	Médico Especialista
Resumen:	El caso de uso se inicia cuando el médico desea cargar el fichero DICOM, el sistema muestra el cuadro de diálogo, el médico selecciona la imagen a cargar, el sistema muestra la imagen en la aplicación.
Referencias	RF1

Tabla 5 Descripción del Caso de Uso: Cargar imágenes DICOM.

Caso de Uso:	Mostrar vistas axial, coronal y sagital.
Actores:	Médico Especialista
Resumen:	El caso de inicio se inicia cuando el médico desea visualizar una de las vistas (axial, coronal o sagital), el sistema muestra en la pantalla completa la vista seleccionada.
Referencias	RF2

Tabla 6 Descripción del Caso de Uso: Mostrar vistas axial, coronal y sagital.

Caso de Uso:	Rotar imagen.
Actores:	Médico Especialista
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen que desea transformar, rota la imagen, el sistema muestra la imagen resultante.
Referencias	RF3

Tabla 7 Descripción del Caso de Uso: Rotar imagen.

Caso de Uso:	Escalar imagen
Actores:	Médico Especialista

Capítulo 2: Características del Sistema

Resumen:	El médico puede ampliar la imagen y disminuirla a conveniencia, ya sea toda la imagen o una región específica utilizando una lupa, en esta última forma el cursor toma forma rectangular ampliando la parte de la imagen donde este posicionado el cursor.
Referencias	RF4

Tabla 8 Descripción del Caso de Uso: Escalar imagen.

Caso de Uso:	Deshacer acciones.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona deshacer o, el sistema realiza la acción seleccionada, muestra la imagen luego de haber realizado la acción.
Referencias	RF5

Tabla 9 Descripción del Caso de Uso: Deshacer acciones.

Caso de Uso:	Rehacer acciones.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona deshacer o rehacer acciones, el sistema realiza la acción seleccionada, muestra la imagen luego de haber realizado la acción.
Referencias	RF6

Tabla 10 Descripción del Caso de Uso: Rehacer acciones.

Caso de Uso:	Aplicar Filtros.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona la imagen que desea transformar, selecciona el tipo de filtro, el sistema muestra la imagen que queda como resultado de aplicarle el filtro.

Capítulo 2: Características del Sistema

Referencias	RF7
--------------------	-----

Tabla 11 Descripción del Caso de Uso: Aplicar Filtros.

Caso de Uso:	Realizar mediciones sobre la imagen.
Actores:	Médico Especialista
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen sobre la cual va a realizar la medición, escoge el tipo de medición, procede a ejecutar la medición, el sistema pinta la figura y da el resultado.
Referencias	RF8

Tabla 12 Descripción del Caso de Uso: Realizar mediciones sobre la imagen.

Caso de Uso:	Calcular Volumen
Actores:	Médico Especialista
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen a la que quiere calcular el volumen, selecciona calcular volumen, el sistema muestra el resultado en pantalla.
Referencias	RF9

Tabla 13 Descripción del Caso de Uso: Calcular Volumen.

Caso de Uso:	Segmentación
Actores:	Médico Especialista
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen que desea transformar, el usuario selecciona segmentar, procede a dar click en la parte de la imagen que desea segmentar, el sistema muestra la nueva imagen con los resultados de la segmentación.
Referencias	RF10

Capítulo 2: Características del Sistema

Tabla 14 Descripción del Caso de Uso: Segmentación.

Caso de Uso:	Reconstrucción 3D
Actores:	Médico Especialista
Resumen:	El médico selecciona la imagen que desea transformar, selecciona la opción Reconstrucción 3D, el sistema muestra en pantalla la imagen reconstruida.
Referencias	RF11

Tabla 15 Descripción del Caso de Uso: Reconstrucción 3D.

2.6 Conclusiones generales del capítulo.

En este capítulo se realizó el modelo conceptual o de dominio para comprender la estructura y la dinámica del sistema, se seleccionaron los actores que intervienen. Por último, se agruparon los requisitos funcionales en casos de uso y estos últimos se describieron para un mejor entendimiento de las funcionalidades con que cuenta la interfaz desarrollada. Ya se puede proceder con el análisis y diseño de la aplicación.

Capítulo 3: Análisis y Diseño del Sistema.

Introducción.

En este capítulo se tendrán en cuenta los procesos de análisis y diseño del sistema exponiéndolos a través de los distintos artefactos y diagramas de clases del análisis, los diagramas de clases del diseño, así como las clases que interactúan para llevar a cabo la realización de los distintos casos de uso, dando cumplimiento a los requerimientos funcionales.

3.1 Análisis del Sistema.

Con el objetivo de refinar los requisitos y comprender de una manera más precisa que debe hacer el sistema, se realiza el proceso de análisis. El lenguaje utilizado en el análisis se basa en un modelo de objetos, llamado modelo de análisis. El modelo de análisis contiene clases del análisis incluyendo artefactos asociados. Dicho modelo puede ser un artefacto temporal, razón por la que el mismo evoluciona hacia el modelo de diseño o utilizándose como un modelo conceptual general del sistema.

En resumen, el modelo de análisis ayuda a estructurar y refinar los requisitos, además de brindar una vista interna del sistema. Debido a la utilidad del análisis, se ha realizado el mismo, para la construcción del sistema propuesto, refinando cada uno de los requisitos y modelando la realización de casos de uso mediante las clases de análisis encontradas.

3.1.1. Diagrama de Clases del Análisis.

Diagrama de Clase del Análisis “Cargar DICOM”

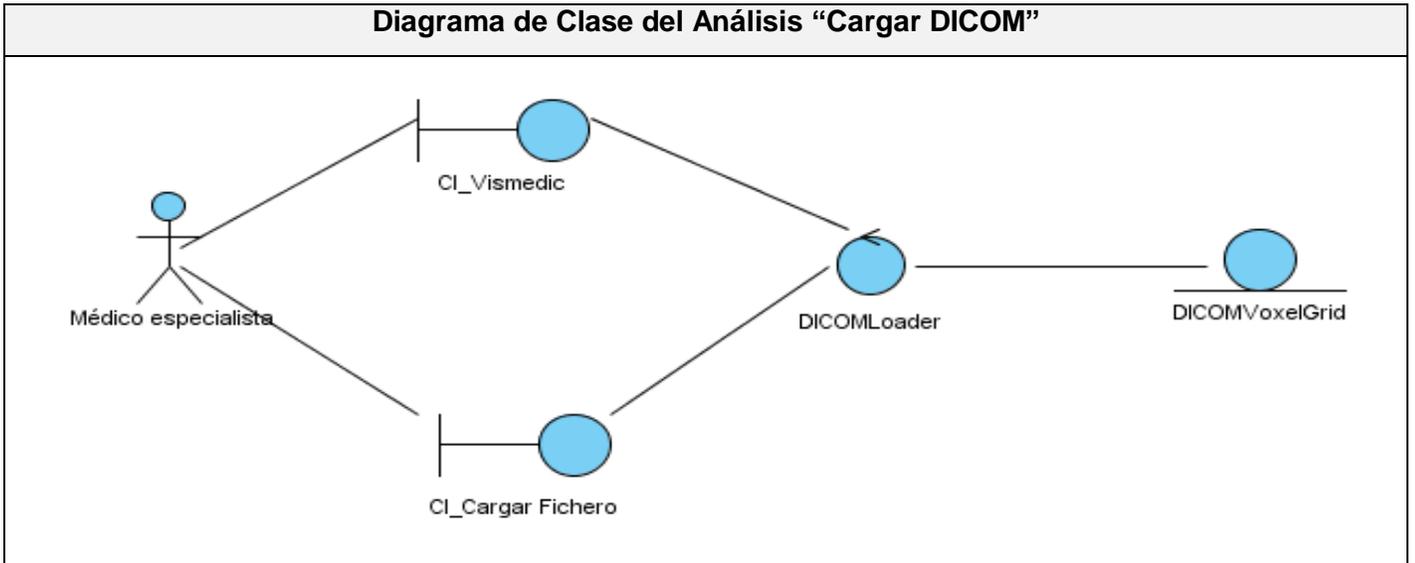
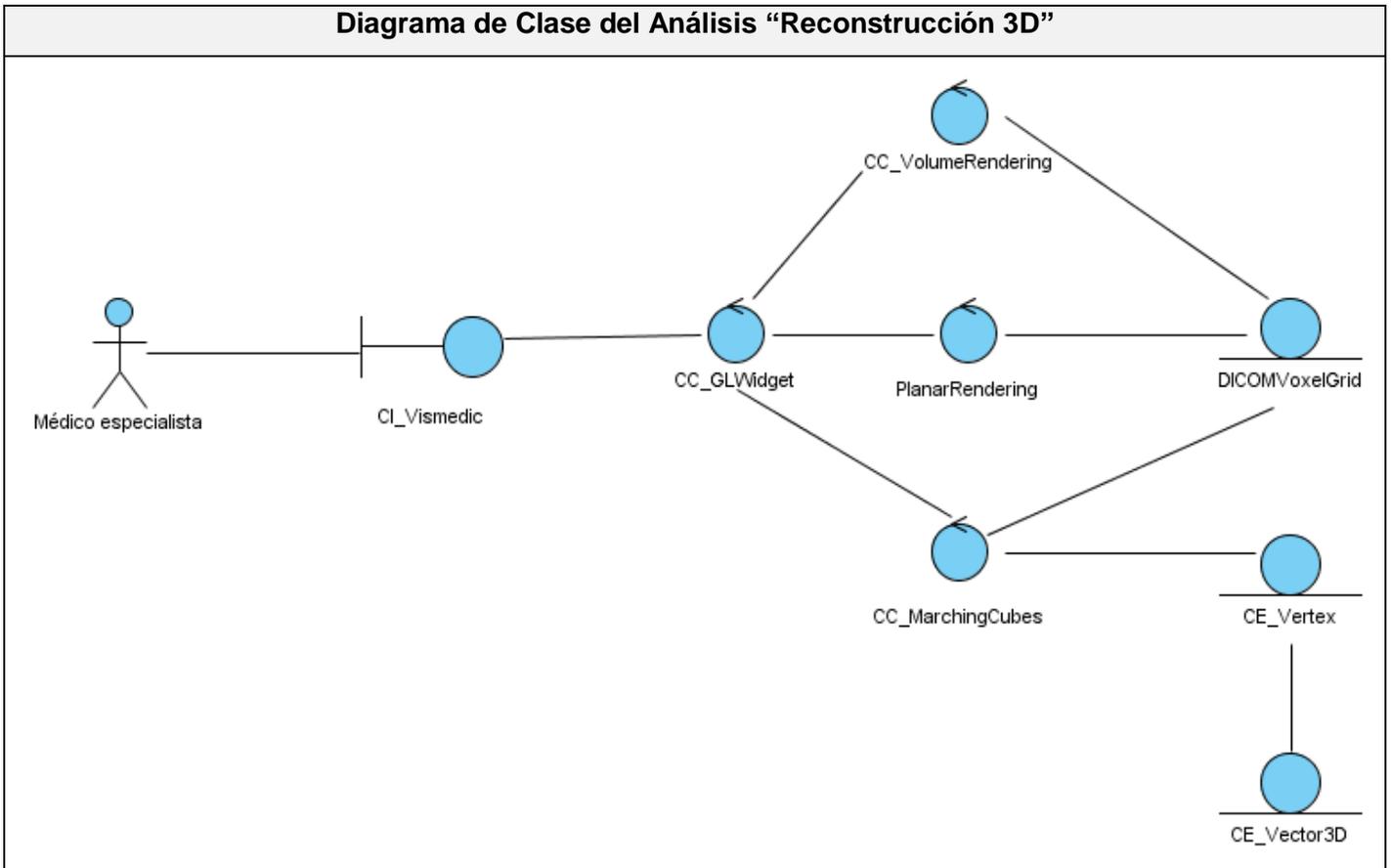
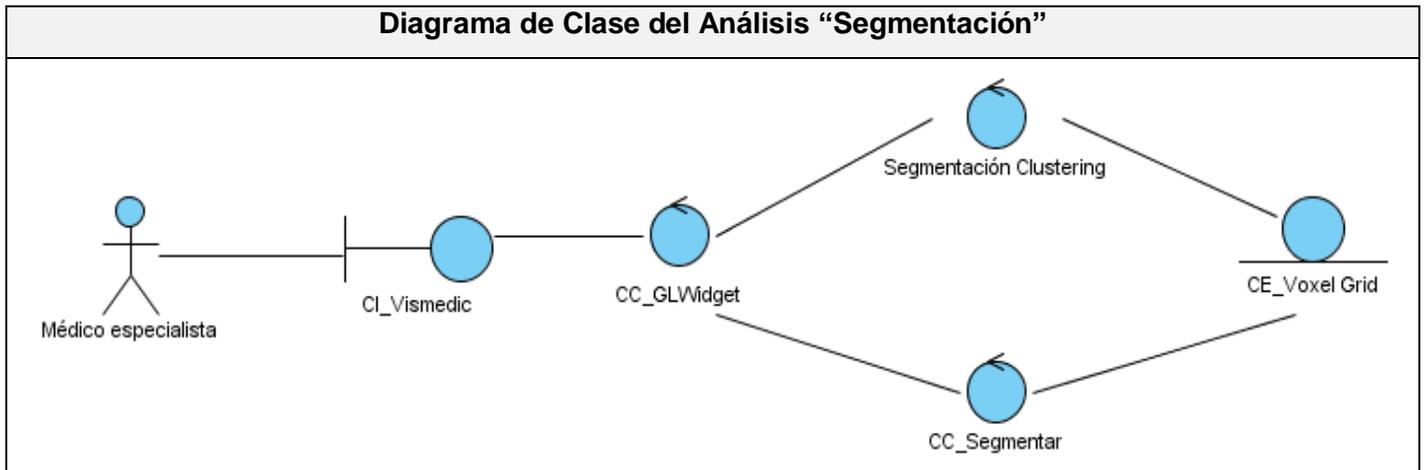


Diagrama de Clase del Análisis “Reconstrucción 3D”





3.2 Diseño del Sistema.

En el flujo de trabajo de Diseño lo más importante es la elaboración de los diagramas de clases de diseño, que muestran las clases finales que dan realización a los casos de uso modelados con anterioridad. Muchas de las clases definidas en el análisis se mantienen presentes o bien aparecen algunas nuevas para cumplir funciones más específicas de las definidas en el análisis.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas. Son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. Por su parte, existen dos tipos de diagramas de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración. En este trabajo se utilizará para la modelación del diseño de la herramienta los diagramas de secuencia, debido al nivel de detalle que describen, proporcionando un mayor flujo de información para el desarrollo de la herramienta.

3.2.1. Diagramas de Clases.

Diagrama de Clase del Diseño “Cargar DICOM”

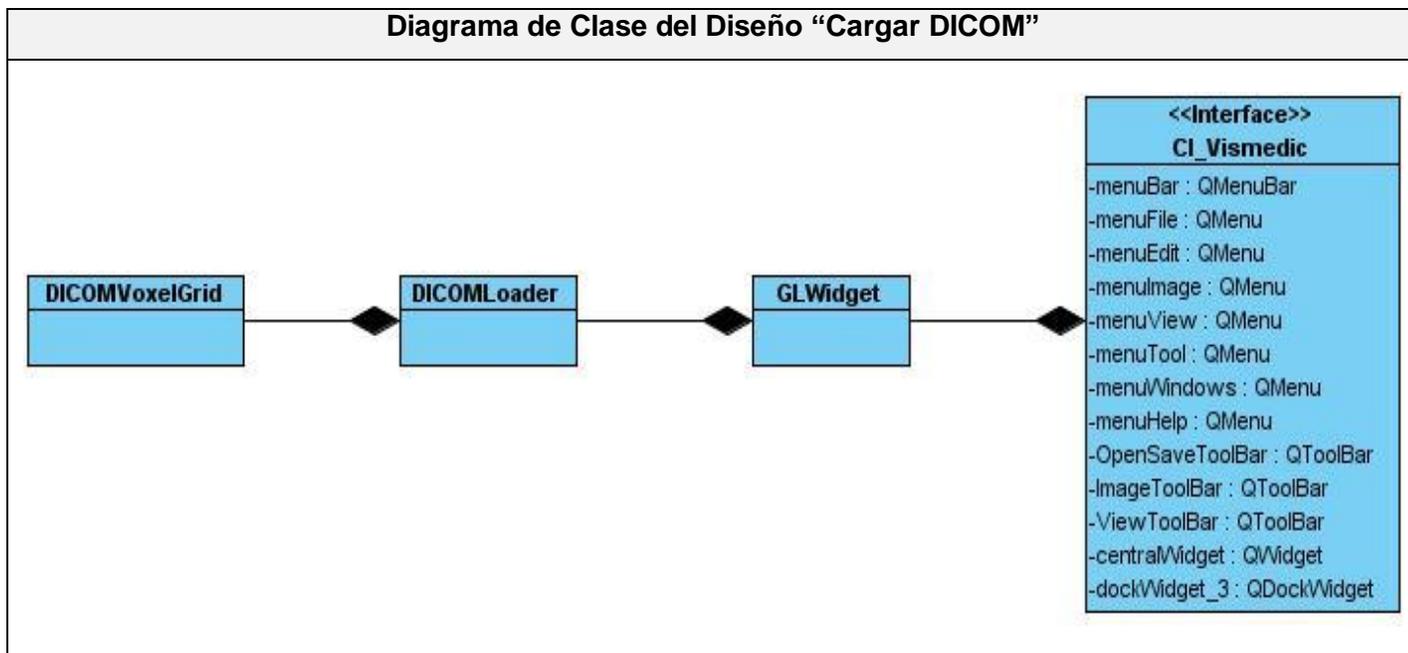
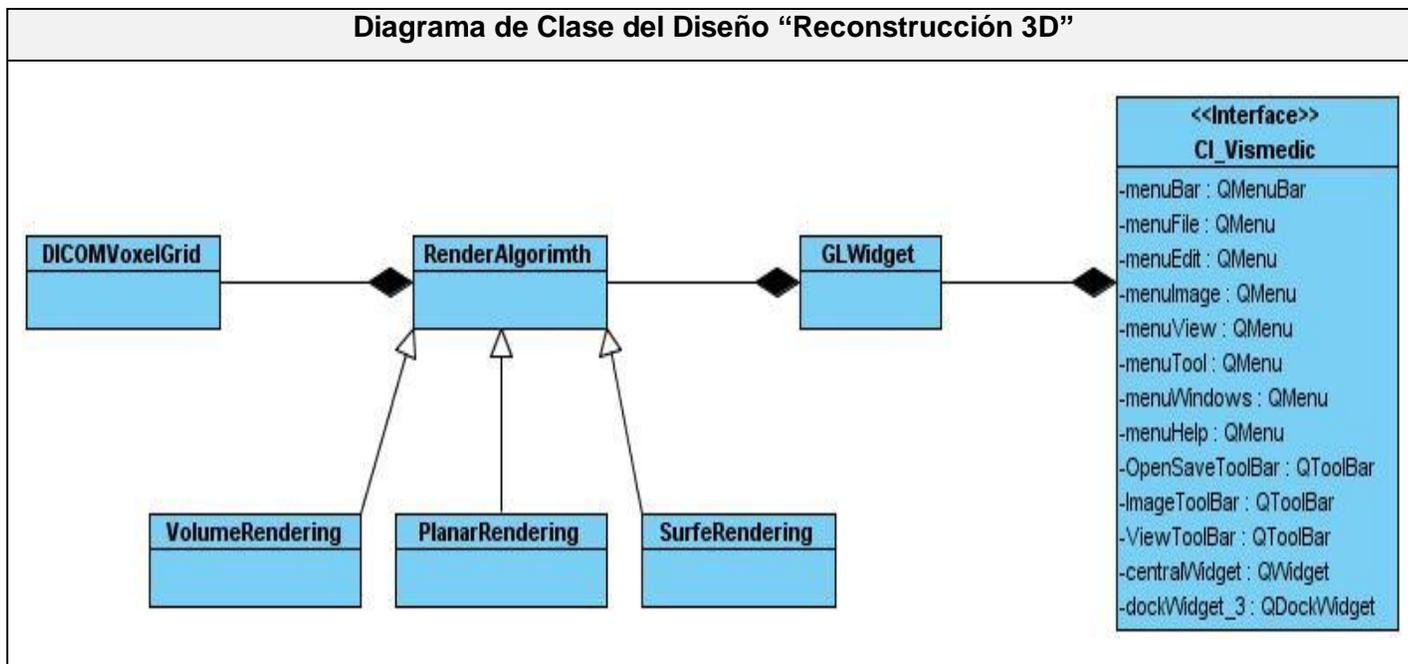
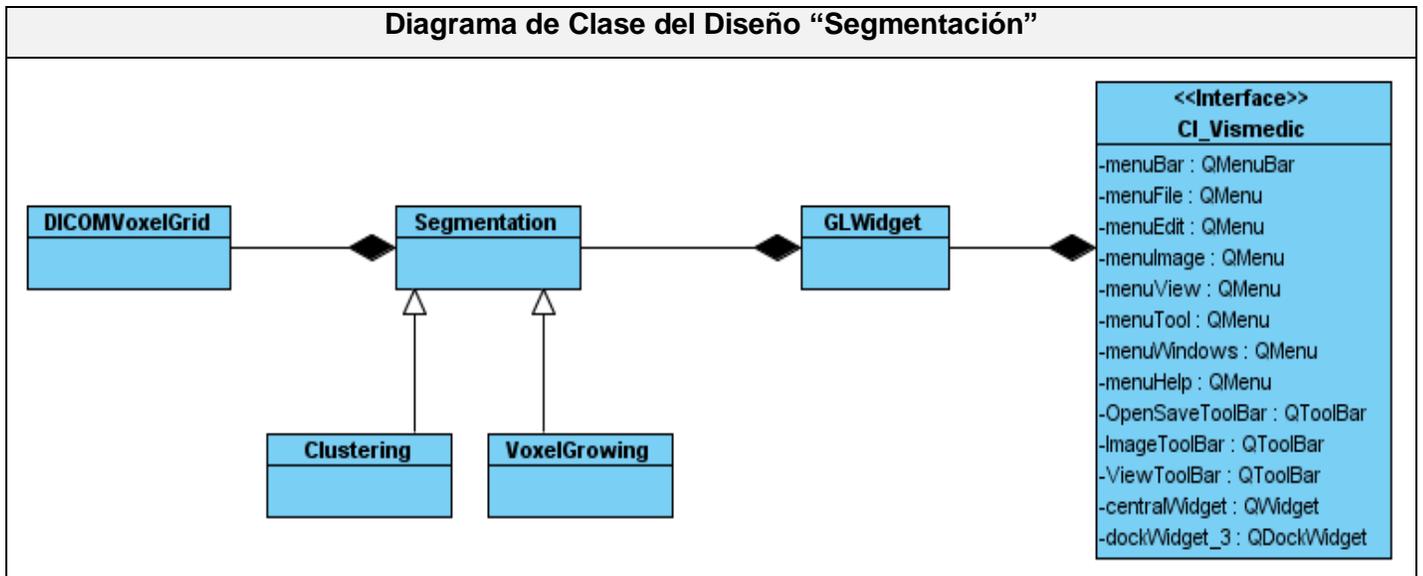


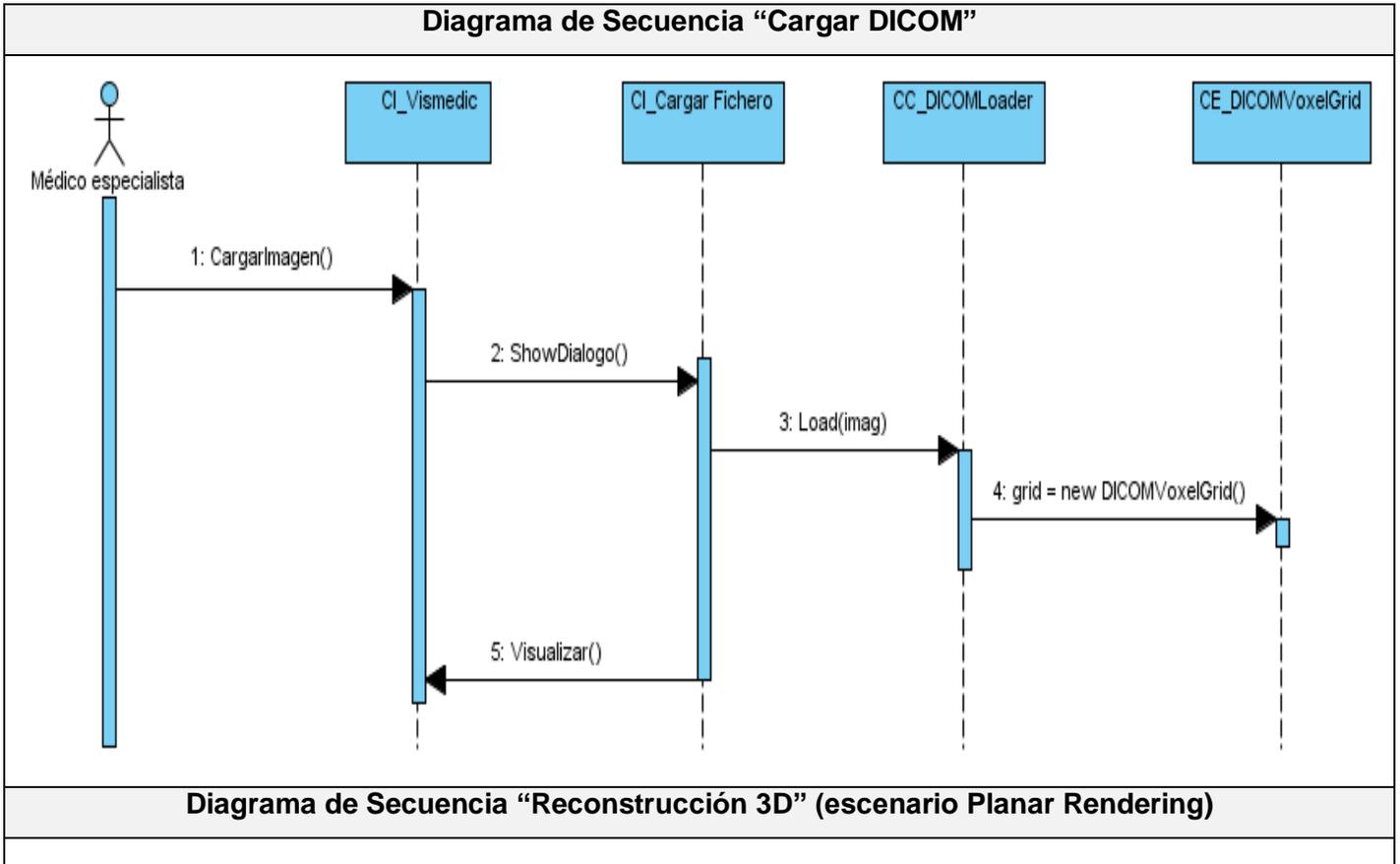
Diagrama de Clase del Diseño “Reconstrucción 3D”





3.2.2. Diagramas de Interacción.

Diagrama de Secuencia "Cargar DICOM"



Capítulo 3: Análisis y Diseño del Sistema

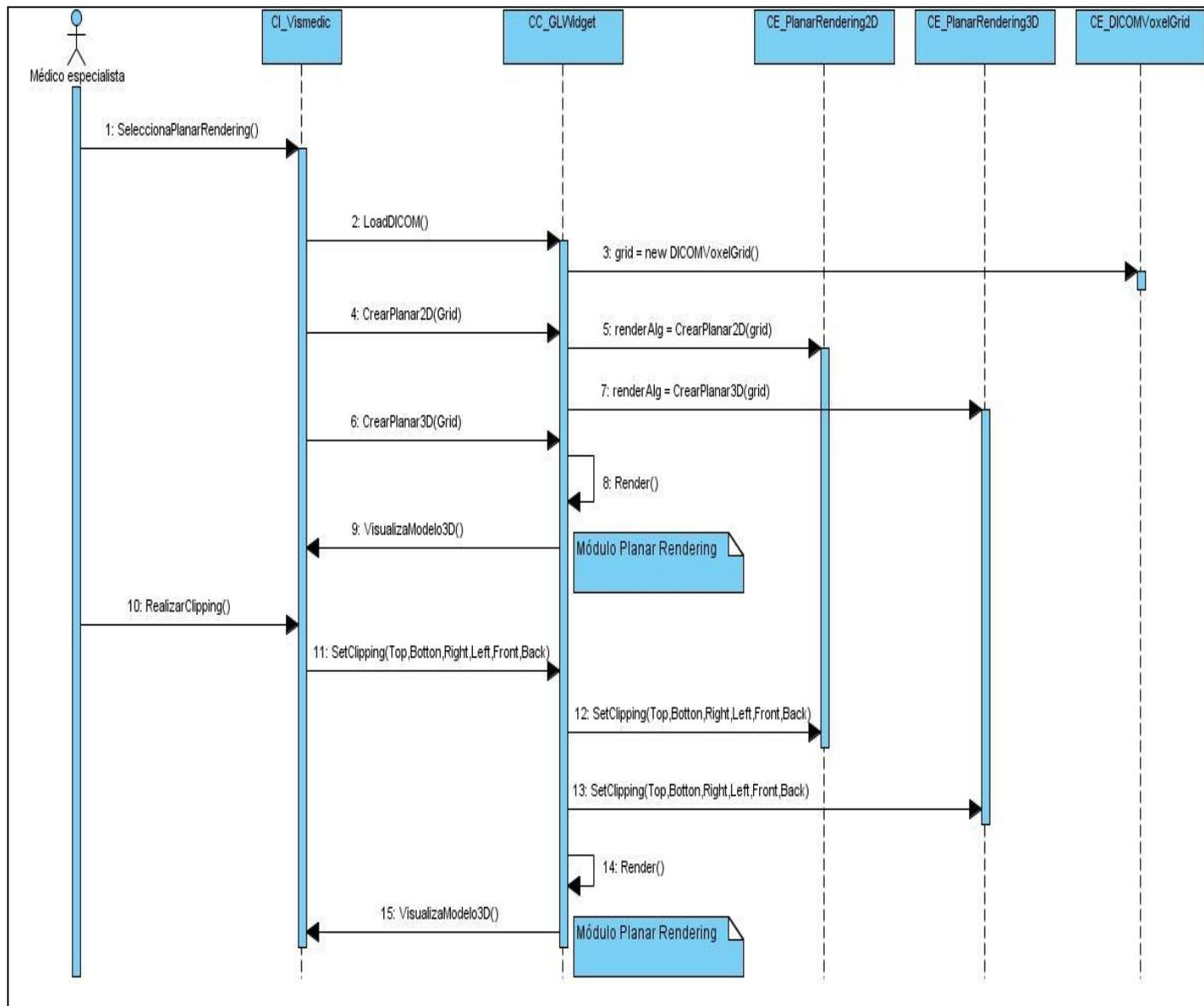


Diagrama de Secuencia "Reconstrucción 3D" (escenario Volume Rendering)

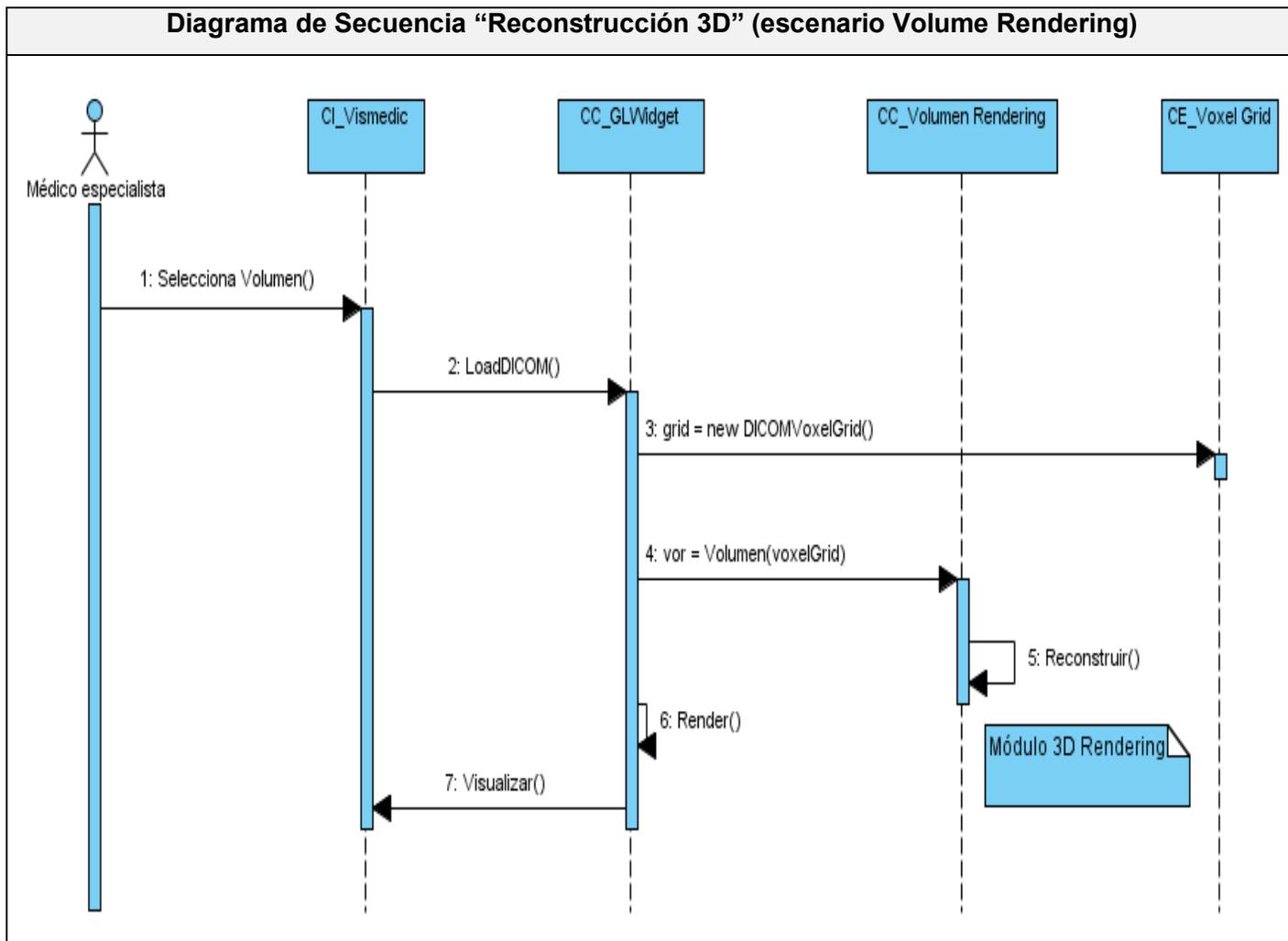


Diagrama de Secuencia "Reconstrucción 3D" (escenario Surface Rendering)

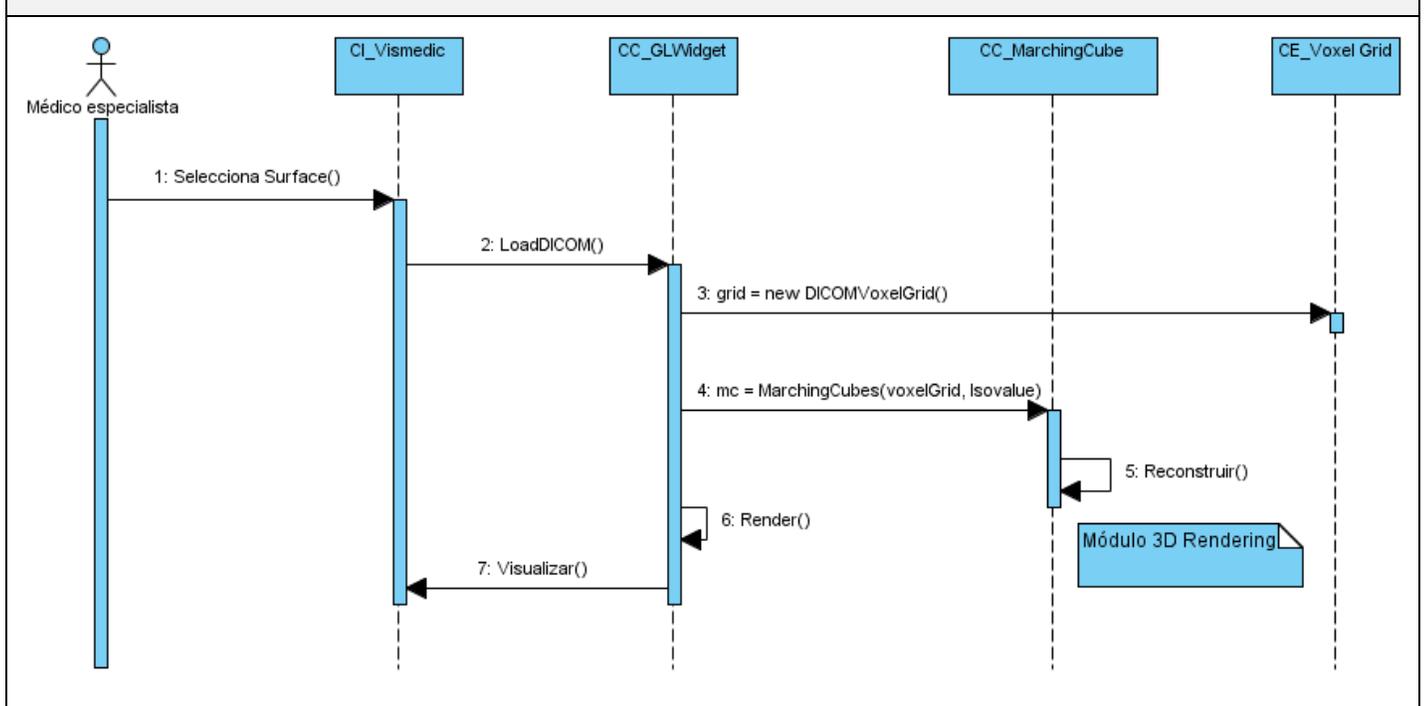
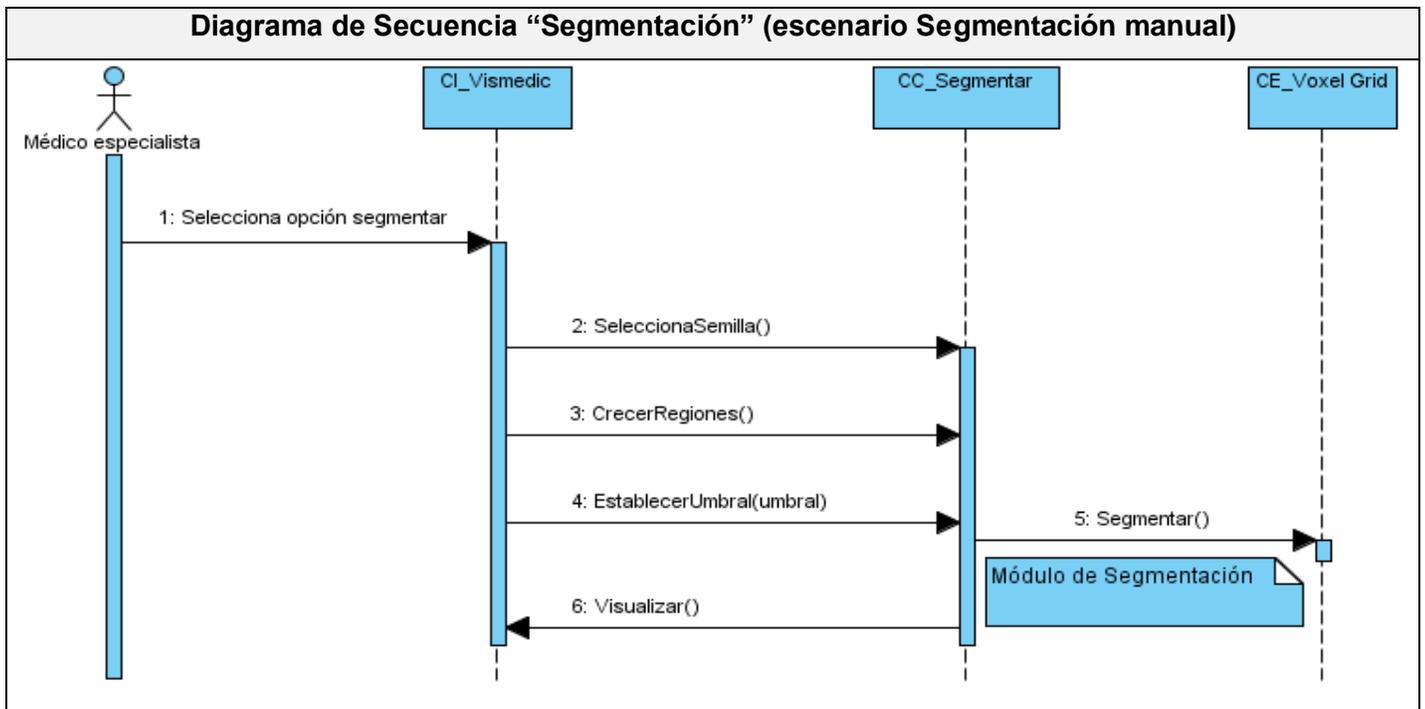
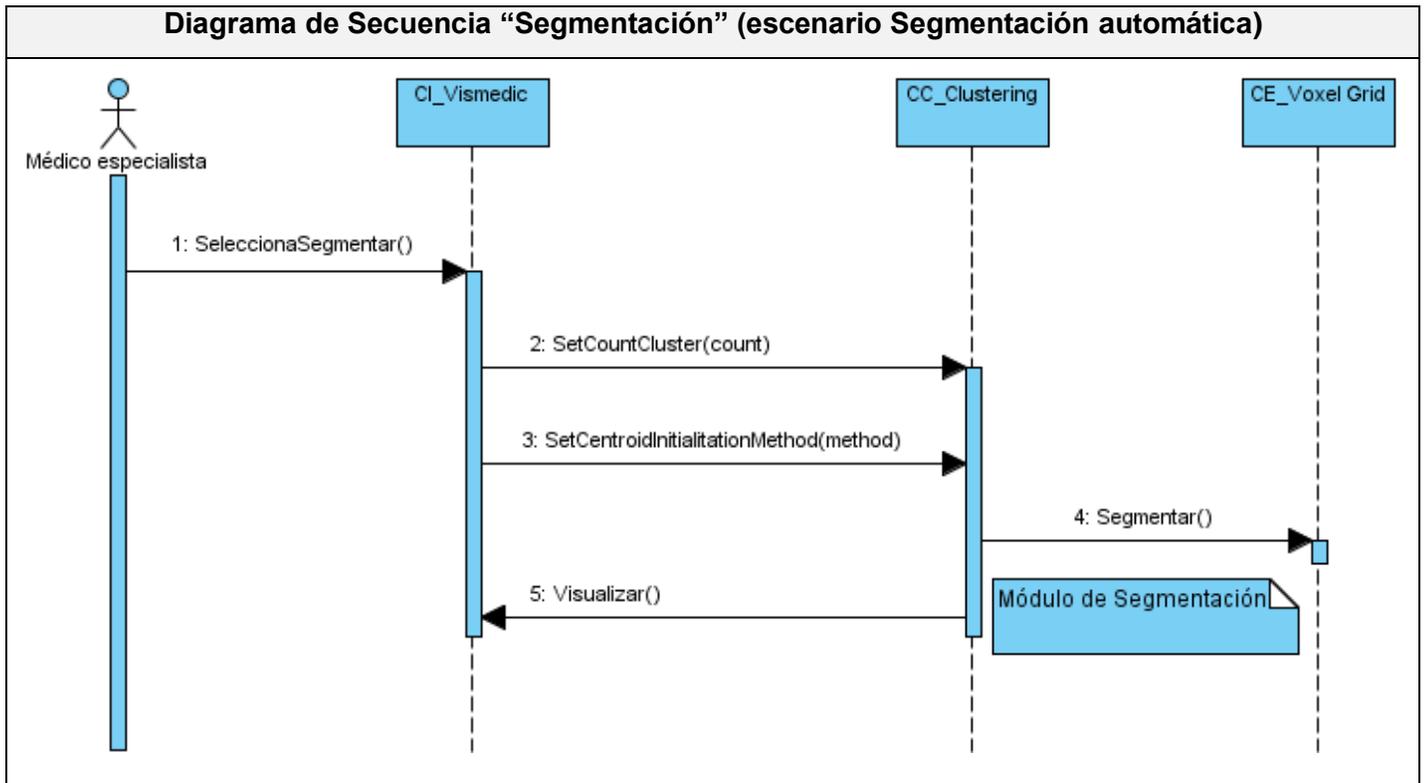


Diagrama de Secuencia "Segmentación" (escenario Segmentación manual)





2.3 Conclusiones generales del capítulo.

En este capítulo se llevó a cabo el desarrollo del flujo de trabajo de análisis y diseño propuesto por el proceso de desarrollo RUP, teniendo en cuenta que el mismo es uno de los flujos más importantes que se llevan a cabo en el ciclo de vida de cualquier producto software.

Se comenzó por el refinamiento de los requerimientos encontrados en el capítulo anterior, con el objetivo de formalizarlos y lograr una primera vista interna del sistema de fácil comprensión por parte de los desarrolladores. Posteriormente se realizaron las actividades de diseño, donde se modelaron las clases que componen el sistema, sus relaciones y su interacción para realizar los casos de uso de la primera iteración.

Capítulo 4: Implementación y Validación del Sistema

Capítulo 4: Implementación y Validación del Sistema.

Introducción.

El presente capítulo cubre el flujo de implementación realizado para la primera iteración del sistema propuesto. El resultado principal de la implementación, es la obtención de componentes, dentro de los que se incluyen ficheros y ejecutables, y sus dependencias. Se presenta todo el proceso de validación del diseño de la interfaz propuesta a través del método Delphi basado en consultas a expertos.

4.1 Diagrama de Componentes.

Un componente representa una parte física del sistema, por ejemplo, una librería, un ejecutable, una tabla, etc., que engloba la implementación de un grupo de clases del diseño. Cada componente define una interfaz que describe su funcionalidad y forma de empleo. El diagrama de componentes permite a los desarrolladores y clientes conocer la estructura física que tiene el sistema y cómo se relacionan sus partes.

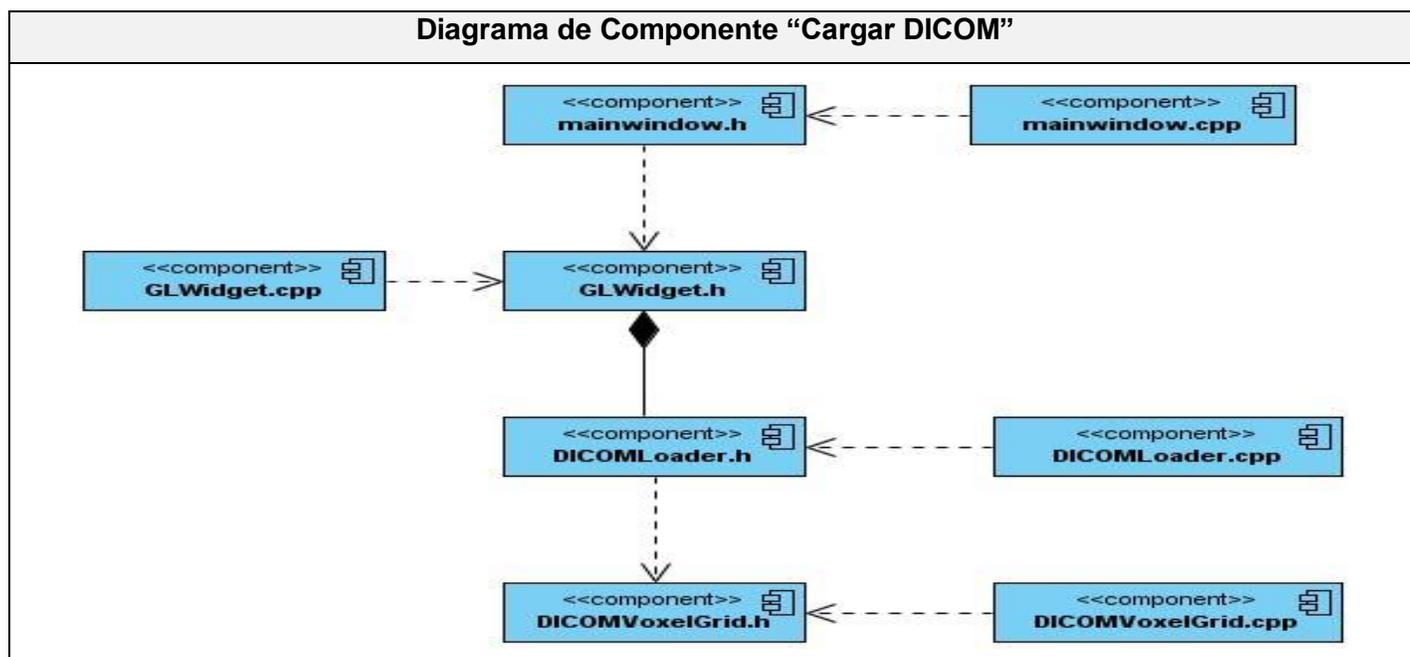


Diagrama de Componente "Reconstrucción 3D"

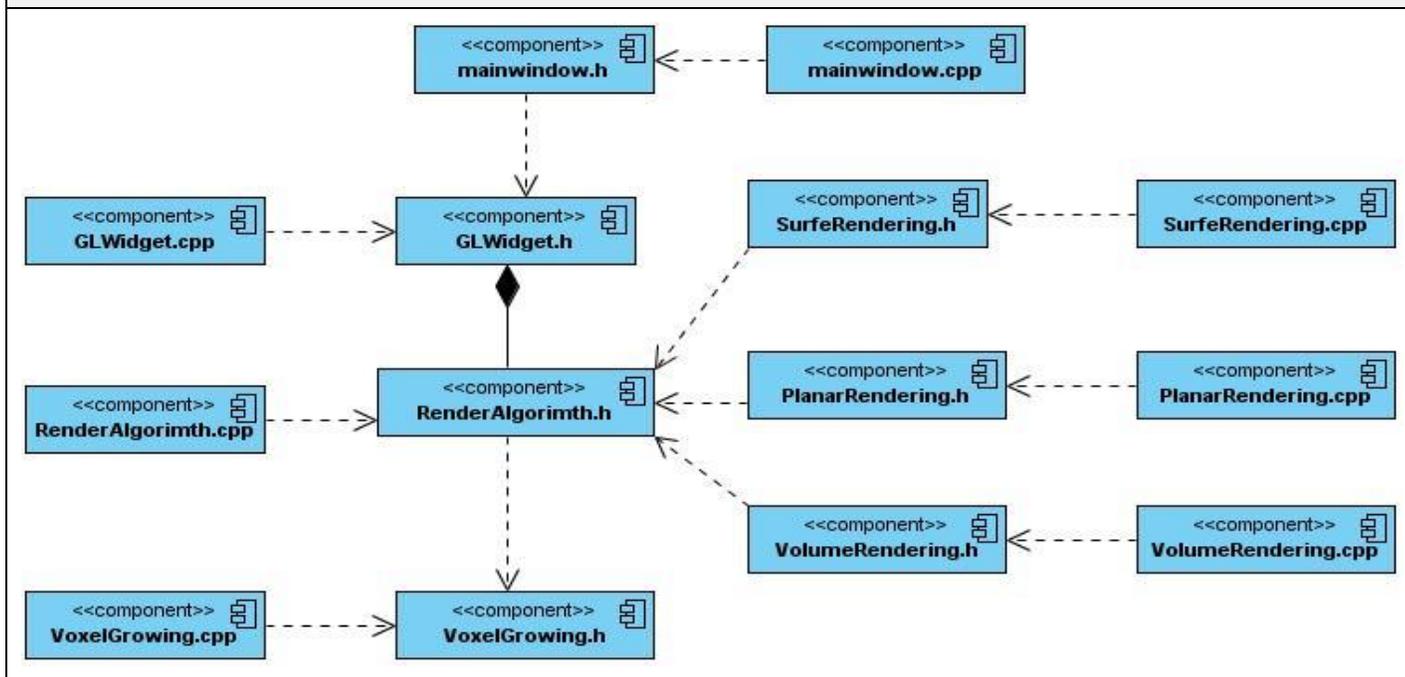
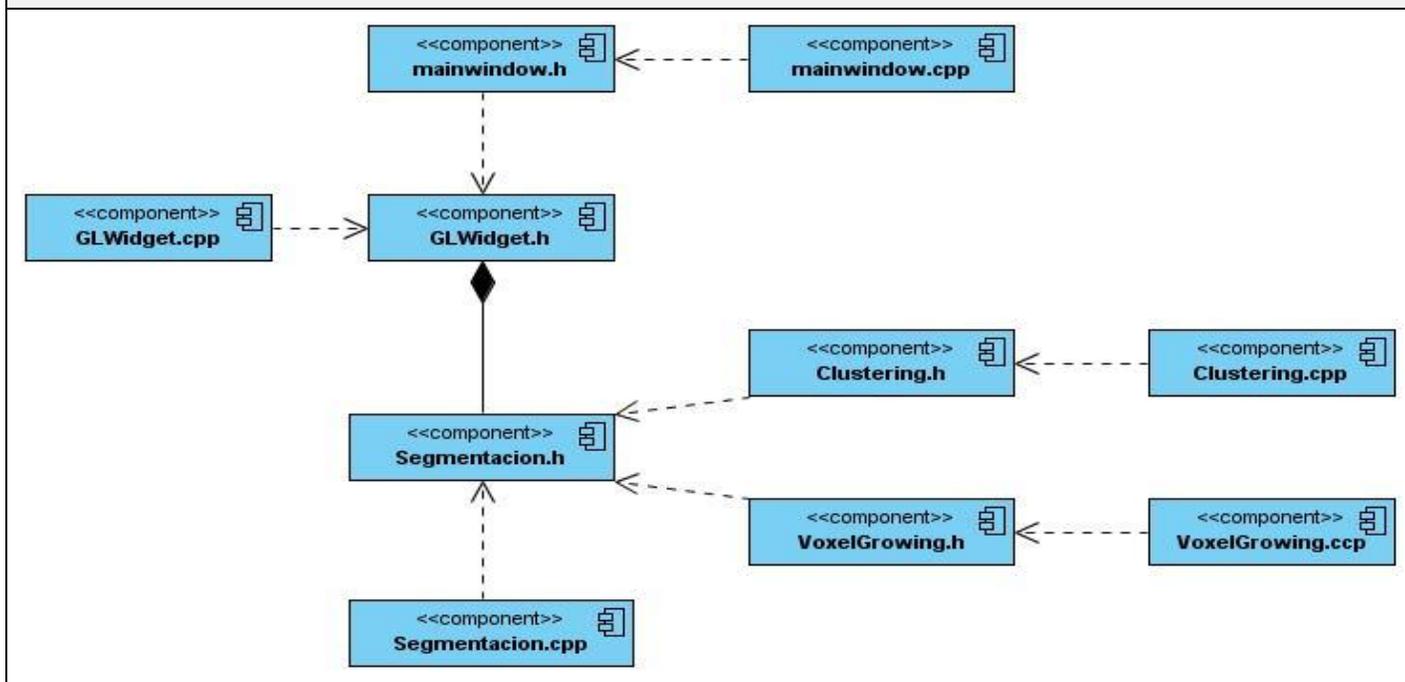


Diagrama de Componente "Segmentación"



Capítulo 4: Implementación y Validación del Sistema

4.2 Validación de la propuesta. Método Delphi.

El método Delphi, creado por la compañía Rand Corporation, es considerado como una de las técnicas subjetivas de pronóstico más confiable. Fue creado con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos que podrían tener lugar en varias ramas de la ciencia, la técnica y la política. De esta manera, se define como la utilización sistémica del juicio intuitivo de un grupo de expertos, para obtener un consenso de opiniones informadas. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro. [35]

El método Delphi es un método de prospectiva basado en consultas a expertos en el que los temas planteados, en este caso en forma de cuestionario, son remitidos a un conjunto de jueces llamado panel de expertos que son los que darán su opinión acerca del asunto que se está tratando y entre los cuales se hará necesario garantizar el anonimato.

Las principales características del método están dadas por el anonimato de los participantes (excepto el investigador), iteración (manejar tantas rondas como sean necesarias), retroalimentación (feedback) controlada, sin presiones para la conformidad, respuesta de grupo en forma estadística (el grado de consenso se procesa por medio de técnicas estadísticas) y justificación de respuestas (discrepancias/consenso). [36]

Suelen distinguirse tres etapas o fases fundamentales en la aplicación del método, según la variada bibliografía consultada al respecto, que son:

- **Fase preliminar:** Se delimita el contexto, los objetivos, el diseño, los elementos básicos del trabajo y la selección de los expertos.
- **Fase exploratoria:** Elaboración y aplicación de los cuestionarios según sucesivas vueltas, de tal forma que con las respuestas más comunes de la primera se confecciona la siguiente.
- **Fase final:** Análisis estadísticos y presentación de la información.

Capítulo 4: Implementación y Validación del Sistema

4.2.1 Proceso de selección de los expertos.

Un experto es una persona o un grupo de personas que poseen un alto grado de conocimientos sobre el tema de estudio, ya sea porque se ha dedicado a la investigación y estudios de aspectos relacionados con el tema, o bien porque en su experiencia profesional, el tema ha constituido parte importante de su trabajo.

De acuerdo con lo anteriormente expresado, se definen como características fundamentales para los expertos que conforman este grupo de evaluación:

- Conocimientos acerca de los contenidos que sustentan la propuesta de solución.
 - Software de Visualización Médica.
 - Diseño.
- Graduado de Nivel Superior.
- Al menos un año de experiencia trabajando los temas que se señalaron anteriormente.
- Prestigio en el colectivo de trabajo.
- Capacidad de análisis y pensamiento lógico.

El conocimiento sobre estos temas permitirá que las opiniones brindadas sean confiables y válidas para el objetivo propuesto. Se tendrán en cuenta otras cualidades como la honestidad, sinceridad y responsabilidad, haciendo más confiables las opiniones brindadas. La búsqueda de los expertos se realizará en el contexto de la Universidad de las Ciencias Informáticas, y en el Instituto Central de Investigación Digital (ICID).

Aunque no existe una norma generalizada que determine un valor óptimo respecto al número de expertos. Los investigadores de Rand Corporation, indican que es necesario, como mínimo 7 expertos y un máximo de 30. Atendiendo las características delimitadoras de este grupo, se decidió contar con un número de 9 expertos.

Capítulo 4: Implementación y Validación del Sistema

4.2.2 Elaboración y aplicación de las encuestas.

Las características de los expertos influyen decisivamente en la confiabilidad de los resultados obtenidos. Estas características son: calificación técnica, capacidad de emitir una decisión al respecto, conocimientos específicos sobre el tema a evaluar, disposición a participar, entre otros. Para determinar la competencia de los expertos se aplicó una encuesta de autovaloración. (Ver Anexo 2).

4.2.2.1 Cálculo del coeficiente de competencia.

La selección de los expertos se hace de acuerdo con la valoración de sus competencias, para esto es necesario calcular el coeficiente de competencia (k) que se basa en los resultados de la encuesta de autovaloración, específicamente en el coeficiente de conocimiento (K_c) que es el resultado de la primera pregunta de la encuesta de autovaloración y el coeficiente de argumentación del conocimiento (K_a) el cual se obtiene luego de analizar los resultados de la tabla que conforma la Pregunta 2 de dicha encuesta.

Para calcular el coeficiente de competencia primeramente se asignan valores a las categorías seleccionadas por los posibles expertos a partir de la siguiente escala.

Fuentes de argumentación	Alto	Medio	Bajo
Análisis realizado por usted.	0.3	0.2	0.1
Experiencia de trabajo en esta actividad.	0.5	0.4	0.2
Trabajos de autores nacionales.	0.05	0.05	0.05
Trabajos de autores internacionales.	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero.	0.05	0.05	0.05
Su intuición.	0.05	0.05	0.05
Total	1.0	0.8	0.5

Tabla 16 Grados de influencia en la determinación del coeficiente de argumentación.

Capítulo 4: Implementación y Validación del Sistema

El coeficiente de argumentación del conocimiento (K_a) será igual a la suma de los valores donde el posible experto haya marcado. A continuación se calcula el coeficiente de competencia según la siguiente fórmula:

$$k = \frac{(k_c + k_a)}{2}$$

Donde k_c es el coeficiente de conocimiento o información que tiene el experto del tema y (K_a) es el coeficiente de argumentación o fundamentación de los criterios del experto. El resultado del coeficiente de competencia de cada experto se analiza de la siguiente forma:

Si $0,8 \leq k \leq 1,0$ el coeficiente de competencia es alto.

Si $0,5 \leq k < 0,8$ el coeficiente de competencia es medio.

Si $k < 0,5$ el coeficiente de competencia es bajo.

Los resultados del análisis de competencia se muestran a continuación:

Experto	Coeficiente de conocimiento. (k_c)	Coeficiente de argumentación. (k_a)	Coeficiente de competencia. (k)	Nivel
Experto 1	0,7	0,9	0,8	Alto
Experto 2	0,8	0,8	0,8	Alto
Experto 3	0,7	0,6	0,65	Medio
Experto 4	0,9	0,9	0,9	Alto
Experto 5	0,6	0,9	0,75	Medio
Experto 6	0,8	0,9	0,85	Alto
Experto 7	0,6	0,9	0,75	Medio
Experto 8	0,7	0,9	0,8	Alto

Capítulo 4: Implementación y Validación del Sistema

Experto 9	0,6	0,8	0,7	Medio
------------------	-----	-----	-----	-------

Tabla 17 Coeficiente de competencia de los expertos.

Para la investigación se seleccionarán aquellos expertos que presentaron un coeficiente de competencia alto y medio.

4.2.2.2 Elaboración del cuestionario de validación.

Una vez seleccionados los expertos, se prosigue con la elaboración de la encuesta de validación (Ver Anexo 3), para lo cual se hace necesario elaborar un cuestionario de forma tal que se adapten a las condiciones de los expertos.

Los encuestados recibieron además del cuestionario con un total de 10 preguntas, la documentación referente a la propuesta de interfaz gráfica de usuario que se evalúa, así como una breve explicación de las condiciones prácticas del desarrollo de la encuesta (garantía de anonimato). El cuestionario fue creado de forma tal que las respuestas fueran categorizadas en Muy adecuado (C1), Bastante adecuado (C2), Adecuado (C3), Poco adecuado (C4) y No adecuado (C5). Se realizó la primera ronda de preguntas y luego se prosiguió a analizar los resultados.

4.2.2.3 Establecimiento de la concordancia entre los expertos.

Un perfecto acuerdo entre los expertos dará mayor validez a la propuesta, por lo que se necesita calcular el Coeficiente de Concordancia de Kendall que ayuda a comprobar el grado de coincidencia de las valoraciones realizadas por los expertos.

Para la aplicación del Coeficiente de Concordancia de Kendall (W), se construye una tabla de preguntas a evaluar contra expertos donde se sitúan los rangos de valoración (en términos numéricos de 1 a 5) de cada pregunta evaluado contra cada uno de los expertos; estos datos son tomados a partir de los resultados arrojados luego de realizar la encuesta de validación.

Capítulo 4: Implementación y Validación del Sistema

No.	Preguntas	Expertos									Rj
		1	2	3	4	5	6	7	8	9	
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?	4	5	4	5	5	4	4	4	5	40
2.	¿Son adecuados los colores utilizados en la interfaz?	5	4	5	3	5	5	5	3	5	40
3	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?	5	4	5	3	5	4	5	3	5	39
4.	¿Considera usted que la tipografía que se utiliza es correcta?	4	5	5	4	5	4	5	4	4	40
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?	4	5	4	4	3	5	3	5	3	36
6.	¿Considera correcta la distribución espacial de los menús?	5	5	5	4	5	5	4	4	4	41
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?	3	4	4	5	5	3	4	4	4	36
8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil	5	5	4	5	4	3	3	4	5	38

Capítulo 4: Implementación y Validación del Sistema

	identificación?										
9.	¿Considera usted que la relación entre menús y submenús es coherente?	4	5	3	5	4	4	5	5	3	38
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?	4	5	5	3	4	5	5	4	5	40

Tabla 18 Cómputo de evaluación de los expertos por cada pregunta.

Después de la elaboración de la tabla se realizan los siguientes pasos:

- Determinar la suma de los valores numéricos asignados a cada aspecto a evaluar, según el criterio dado por cada experto (R_j).
- Determinar el valor medio de las R_j , dado por la sumatoria de los R_j entre N , siendo N el total de aspectos a evaluar (los aspectos serán las preguntas del cuestionario, en este caso $N = 10$).
- Determinación de la desviación media, dada por la diferencia entre cada R_j y el valor de la media.
- Determinación de la suma de los cuadrados de las desviaciones medias, S .
- Determinación del cuadrado del número total de expertos, K . En este caso $K = 9$.
- Determinación del cubo del número total de aspectos a evaluar, N .
- Determinación de la diferencia entre el cubo de N y N y su multiplicación por el cuadrado de K .

\bar{R}_j Es la media de los rangos y se determina mediante la fórmula:

$$\bar{R}_j = \frac{\sum_{j=1}^n R_j}{N}$$

Capítulo 4: Implementación y Validación del Sistema

Obteniendo el valor de:

$$\overline{R_j} = \frac{388}{10} = 38,8$$

S es la suma de los cuadrados de las desviaciones y se calcula de la siguiente forma:

$$S = \sum_{j=1}^n (R_j - \overline{R_j})^2$$

$$S = (40 - 38,8)^2 + (40 - 38,8)^2 + (39 - 38,8)^2 + (40 - 38,8)^2 \\ + (36 - 38,8)^2 + (41 - 38,8)^2 + (36 - 38,8)^2 + (38 - 38,8)^2 \\ + (38 - 38,8)^2 + (40 - 38,8)^2$$

Donde:

$$S = 1,44 + 1,44 + 0,04 + 1,44 + 7,84 + 4,84 + 7,84 + 0,64 + 0,64 \\ + 1,44 = 27,6$$

W es el coeficiente de Kendall y se calcula mediante la fórmula siguiente:

$$W = \frac{12 * S}{K^2(N^3 - N)}$$

Sustituyendo los valores obtenidos en la ecuación:

$$W = \frac{12 * 27,6}{81 (1000 - 10)} = \frac{331,2}{80190} = 0.004$$

Capítulo 4: Implementación y Validación del Sistema

0.004 expresa el grado de acuerdo entre los 9 expertos al dar un orden evaluativo a las preguntas sometidas a valoración. Luego se procede con el cálculo del Chi-Cuadrado para poder ver si existe concordancia entre los expertos, el mismo se obtiene a través de la fórmula siguiente:

$$\chi^2 = K (N - 1)W$$

$$\chi^2 = 9 (10 - 1) * 0.004 = 0.324$$

El Chi cuadrado calculado se compara con el de las tablas estadísticas. Este se compara con el de la tabla inversa de la función de distribución de la variable Chi-Cuadrado, teniendo en cuenta la probabilidad de error de un 10%, según la cantidad de expertos presentes en la evaluación [37].

Si $\chi^2_{real} < \chi^2_{(\alpha, N-1)}$, entonces existe concordancia en el trabajo de los expertos. Luego de la realización de los cálculos pertinentes, estos arrojaron que $\chi^2_{real} = 0,324$ y el $\chi^2_{(0.1, 9)} = 14.68$, lo cual corrobora el cumplimiento de la comparación y por tanto, existe concordancia entre los expertos.

4.2.3 Desarrollo y procesamiento de la información.

A partir de las respuestas de los cuestionarios realizados por los expertos y a fin de llegar a conclusiones objetivas acerca de la propuesta, se procedió al cómputo y análisis de estos. Los resultados se muestran en la siguiente tabla:

Frecuencias Absolutas							
No.	Preguntas	C1	C2	C3	C4	C5	Total
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?	4	5	0	0	0	9
2.	¿Son adecuados los colores utilizados en la interfaz?	6	1	2	0	0	9

Capítulo 4: Implementación y Validación del Sistema

3	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?	5	2	2	0	0	9
4.	¿Considera usted que la tipografía que se utiliza es correcta?	4	5	0	0	0	9
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?	3	3	3	0	0	9
6.	¿Considera correcta la distribución espacial de los menús?	5	4	0	0	0	9
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?	2	5	2	0	0	9
8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?	4	3	2	0	0	9
9.	¿Considera usted que la relación entre menús y submenús es coherente?	4	3	2	0	0	9
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?	5	3	1	0	0	9

Tabla 19 Frecuencias absolutas para cada pregunta de la encuesta.

Después de tener todos los datos computados se procede con el siguiente paso del método para construir la tabla de las frecuencias absolutas acumuladas, este consiste en reformar la tabla anterior pero con la

Capítulo 4: Implementación y Validación del Sistema

peculiaridad de que los valores de las celdas en cada fila se obtienen sumándole el anterior excepto la primera celda, y en la cual además desaparece la última columna.

Frecuencias Absolutas Acumuladas						
No.	Preguntas	C1	C2	C3	C4	C5
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?	4	9	9	9	9
2.	¿Son adecuados los colores utilizados en la interfaz?	6	7	9	9	9
3.	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?	5	7	9	9	9
4.	¿Considera usted que la tipografía que se utiliza es correcta?	4	9	9	9	9
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?	3	6	9	9	9
6.	¿Considera correcta la distribución espacial de los menús?	5	9	9	9	9
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?	2	7	9	9	9
8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?	4	7	9	9	9
9.	¿Considera usted que la relación entre menús y	4	7	9	9	9

Capítulo 4: Implementación y Validación del Sistema

	submenús es coherente?					
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?	5	8	9	9	9

Tabla 20 Frecuencias absolutas acumuladas para cada pregunta de la encuesta.

A continuación, como parte del método se procede con el siguiente paso el cual consiste en construir la tabla de frecuencias relativas acumuladas, la cual se obtiene modificando la tabla anterior, para ello se divide cada valor entre el número de expertos.

Frecuencias relativas acumuladas:						
No.	Aspectos	C1	C2	C3	C4	C5
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?	0,4444	0.9999	0.9999	0.9999	0.9999
2.	¿Son adecuados los colores utilizados en la interfaz?	0,6667	0,7778	0.9999	0.9999	0.9999
3	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?	0,5556	0,7778	0.9999	0.9999	0.9999
4.	¿Considera usted que la tipografía que se utiliza es correcta?	0,4444	0.9999	0.9999	0.9999	0.9999
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?	0,3333	0,6667	0.9999	0.9999	0.9999
6.	¿Considera correcta la distribución espacial de los	0,5556	0.9999	0.9999	0.9999	0.9999

Capítulo 4: Implementación y Validación del Sistema

	menús?					
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?	0,2222	0,7778	0.9999	0.9999	0.9999
8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?	0,4444	0,7778	0.9999	0.9999	0.9999
9.	¿Considera usted que la relación entre menús y submenús es coherente?	0,4444	0,7778	0.9999	0.9999	0.9999
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?	0,5556	0,8889	0.9999	0.9999	0.9999

Tabla 21 Frecuencias relativas acumuladas para cada pregunta de la encuesta.

Como un paso siguiente se buscan las imágenes de los elementos de la tabla anterior por medio de la función (Dist. Normal. Standard Inv.). A la misma tabla se le adicionan tres columnas y una fila para colocar los resultados que se mencionan a continuación.

- Suma de las columnas.
- Suma de filas.
- Promedio de las columnas.
- Los promedios de las filas se obtienen de forma similar, en este caso también se divide por cuatro porque quedan cuatro categorías ya que la última se eliminó.
- Para hallar N, se divide la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas ($N = 1.8$).
- El valor N-P da el valor promedio que otorgan los expertos para cada indicador propuesto.

Capítulo 4: Implementación y Validación del Sistema

La tabla siguiente resume lo dicho en los puntos anteriores:

Puntos de Corte								
No.	Aspectos	C1	C2	C3	C4	Suma	N-P	° Adecuación
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?	-0,14	3,72	3,72	3,72	11,02	-0,96	Muy Adecuado
2.	¿Son adecuados los colores utilizados en la interfaz?	0,43	0,76	3,72	3,72	8,63	-0,36	Muy Adecuado
3	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?	0,14	0,76	3,72	3,72	8,34	-0,29	Muy Adecuado
4.	¿Considera usted que la tipografía que se utiliza es correcta?	-0,14	3,72	3,72	3,72	11,02	-0,96	Muy Adecuado
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?	-0,43	0,43	3,72	3,72	7,44	-0,06	Bastante Adecuado

Capítulo 4: Implementación y Validación del Sistema

6.	¿Considera correcta la distribución espacial de los menús?	0,14	3,72	3,72	3,72	11,3	-1,03	Muy Adecuado
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?	-0,76	0,76	3,72	3,72	7,44	-0,06	Bastante Adecuado
8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?	-0,14	0,76	3,72	3,72	8,06	-0,22	Muy Adecuado
9.	¿Considera usted que la relación entre menús y submenús es coherente?	-0,14	0,76	3,72	3,72	8,06	-0,22	Muy Adecuado
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?	0,14	1,22	3,72	3,72	8,8	-0,4	Muy Adecuado
Suma		-0,9	16,61	37,2	37,2	90,11		
Puntos de Corte		-0,09	1,661	3,72	3,72			

Tabla 22 Puntos de Corte.

Capítulo 4: Implementación y Validación del Sistema

Las sumas obtenidas en las cuatro primeras columnas dan los puntos de corte: Los puntos de corte se utilizan para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados. Con ellos se opera del modo siguiente:

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
-0.09	1.661	3.72	3.72	

Tabla 23 Grados de Adecuación.

4.3 Análisis de los resultados de la validación del proceso.

A partir del análisis de los resultados descritos en los epígrafes anteriores, es importante destacar que todos los expertos seleccionados se encuentran actualmente relacionados con el desarrollo de software y temas relacionados con el propósito de la investigación, con una experiencia que oscila entre uno y cinco años.

Para el desarrollo del método se seleccionaron todos los expertos consultados ya que estos presentaban coeficientes de competencia en el tema de alto y medio. Lo cual se puede constatar a partir de la siguiente gráfica.

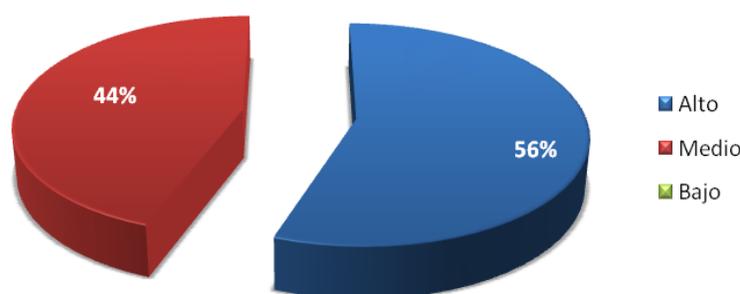


Figura 4 Coeficiente de Competencia de los Expertos.

Como se puede apreciar el 100% de los expertos seleccionados cuentan con el conocimiento necesario para validar la interfaz propuesta, dándole un alto valor a los criterios emitidos.

Capítulo 4: Implementación y Validación del Sistema

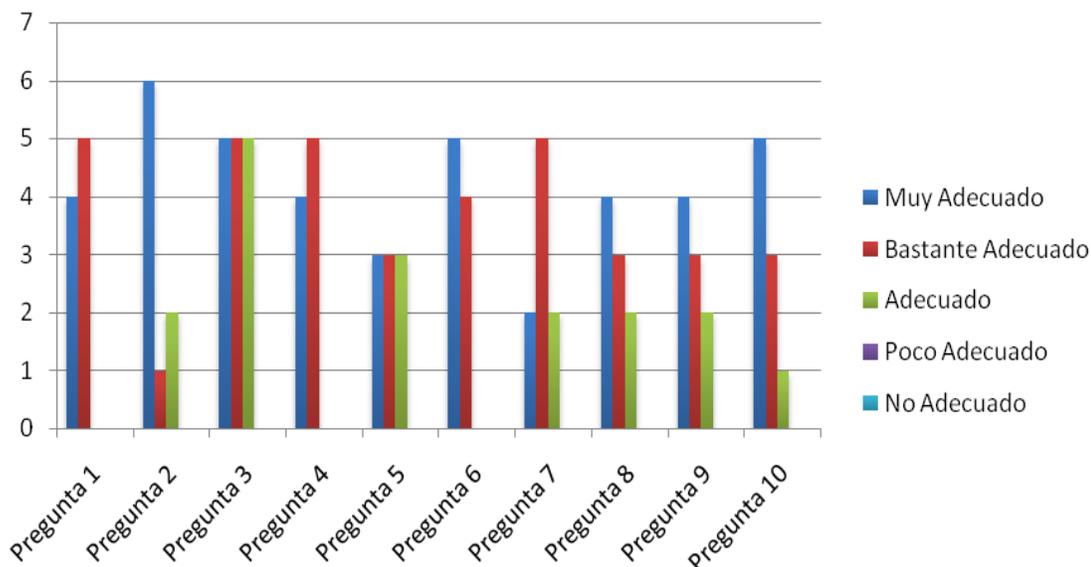


Figura 5 Nivel de Adecuación de las Interrogantes.

En la gráfica anterior se muestra el análisis de las respuestas de los expertos a las interrogantes planteadas en la encuesta, donde se puede apreciar el predominio de las opciones de Muy Adecuado y Bastante Adecuado sobre las demás. Demostrando la valía de la interfaz gráfica de usuario propuesta.

Después de culminado el proceso de validación y computados todos los datos, los resultados obtenidos arrojaron que el 80% de las preguntas fueron catalogadas de Muy Adecuadas mientras que el 20% de estas fueron catalogas de Bastante Adecuadas, como se indica en la siguiente gráfica.

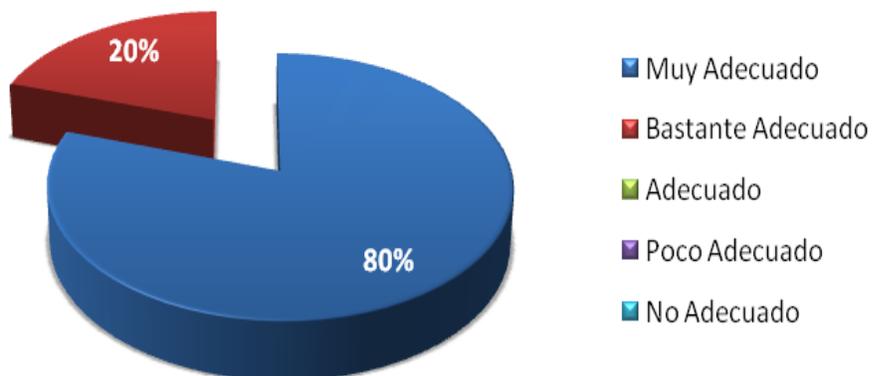


Figura 6 Nivel de Adecuación de las Preguntas de la encuesta.

Capítulo 4: Implementación y Validación del Sistema

4.4 Conclusiones generales del capítulo.

En este capítulo se realizaron los diagramas de componentes correspondientes al flujo de trabajo Implementación. Estos modelos especifican la estructura de los componentes implementados para la aplicación desarrollada. Además, se procedió con la validación de la propuesta a través del método Delphi, aplicando el cuestionario de validación a 9 expertos, arrojando resultados satisfactorios. A partir de las consideraciones emitidas por los expertos se determina no realizar otra iteración del método de validación, puesto que el proceso fue evaluado satisfactoriamente.

Conclusiones

Con la realización de este trabajo de diploma, y en cumplimiento del objetivo inicialmente planteado se obtuvo una interfaz gráfica de usuario, para el software de visualización de imágenes médicas que se está desarrollando en el proyecto VISMEDIC, que incluye dentro de sus requisitos las principales funcionalidades de algunos de los software para la visualización de imágenes médicas existentes en el mercado. Está completamente basada en la política de Open Source y la misma cumple con los principios básicos para el diseño de las interfaces de usuario.

La interfaz será capaz de proporcionar una respuesta visual a las acciones del usuario, brindando un conjunto de funcionalidades que se utilizarán para visualizar, analizar y manipular imágenes DICOM. Además, la misma cuenta con mayor aceptación por parte de los expertos y los médicos especialistas, que la utilizada anteriormente en el software VISMEDIC. Un factor importante lo constituye la naturaleza multiplataforma de la biblioteca gráfica de desarrollo con la que se construyó la interfaz, permitiendo su fácil migración a plataformas libres GNU/Linux, lo cual es un paso de progresivo hacia la emancipación tecnológica que requiere Cuba en estos momentos, por lo que constituye una solución que en un futuro cercano puede tener un impacto positivo a nivel nacional tanto en la economía como en la sociedad.

Recomendaciones

Teniendo en cuenta las conclusiones generales del presente trabajo de diploma, se recomienda:

- La confección de la ayuda y el manual de usuario.
- Darle continuidad a la propuesta desarrollada lográndose incluirles otras funcionalidades a partir del estudio de otro grupo de software para la visualización de imágenes médicas existentes en el mercado.
- La realización de pruebas de calidad a la interfaz.

Referencias Bibliográficas

[1] Yuniel Rodríguez Bello. Diseño e implementación de la interfaz de usuario del Módulo Trabajo Educativo, para el Sistema de gestión de Residencia. Trabajo de Diploma. Universidad de las Ciencias Informática. Ciudad Habana.2008.

[2] Ibáñez, Juan Carlos Pantoja. Arquitectura de la Información: Introducción al proceso de desarrollo en el Diseño de Interfaces de Usuario. [En Línea] [Citado: 30 de septiembre del 2009]

<http://www2.udec.cl/ai/proceso.pdf>

[3] Definición de Interfaz gráfica de usuario. [En Línea] [Citado: 30 de septiembre del 2009]

<http://diccionario.babylon.com/interfaz%20gr%C3%A1fica%20de%20usuario>

[4] GUI. Webopedia. [En Línea] [Citado: 30 de septiembre del 2009]

http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html

[5] Eugenio Jacobo Hernández. Consideraciones para el diseño de interfaces de usuario en sistemas operativos. [En Línea] [Citado: 10 de diciembre del 2009]

<http://www.scribd.com/doc/2522263/Consideraciones-para-el-diseno-de-interfaces-de-usuario-en-sistemas-operativos>

[6] Carlos Marrero Expósito. Interfaz gráfica de usuario: Aproximación semiótica y cognitiva. [En Línea] [Citado: 10 de diciembre del 2009]

http://www.chr5.com/investigacion/investiga_igu/igu_aproximacion_semio-cognitiva_by_chr5.pdf

[7] Aimacaña Toledo, Carlos. Interfaz de Usuario [En Línea]. [Citado: 10 de diciembre del 2009]

<http://www.wikiciencia.org/informatica/programacion/iusuario/index.php>

[8] Marcos Mora, Mari Carmen. Pautas para el diseño y la evaluación de interfaces de usuario. [En Línea] [Citado: 10 de diciembre del 2009] http://www.mcmarcos.com/pdf/2004_pautas-iula.pdf

Referencias Bibliográficas

- [9] Eduardo Mercovich. La intersección entre factores humanos, diseño gráfico, interacción y comunicación. [En Línea] [Citado: 15 de diciembre del 2009]
<http://www.gaiasur.com.ar/infoteca/siggraph99/disenio-de-interfaces-y-usabilidad.html>
- [10] Sebastián Gómez, Leopoldo. Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación [En Línea] [Citado: 15 de diciembre del 2009]
<http://200.14.84.223/apuntesudp/showDoc.php?id=1209&ramo=ICI2423>.
- [11] Preece, J. Human-computer interaction. Harlow England: Adison-Wesley, 1994.
- [12] Bruce Tognazzini. First Principles of Interaction Desing. [En Línea] [Citado: 15 de diciembre del 2009]
<http://www.asktog.com/basics/firstPrinciples.html>
- [13] Psicología del Color. [En Línea] [Citado: 15 de diciembre del 2009]
<http://www.rppnet.com.ar/psicologiadelcolor.htm>
- [14] Benjamin Roe. Diseño de Interfaces de Usuario Usables: Una Guía Rápida para Desarrolladores de Software Libre y de Código Abierto. [En Línea] [Citado: 15 de diciembre del 2009]
<http://mundogeek.net/traducciones/interfaces-usuario-usables/gui.html>
- [15] Jakob Nielsen. Desining Web Usability. [En Línea] [Citado: 17 de enero del 2010]
<http://www.scribd.com/doc/6700249/Jakob-Nielsen-Desining-Web-Usability-Espanhol>
- [16] Ricardo Baeza-Yates, Cuauhtémoc Rivera Loiza y Javier Velasco Martín. Arquitectura de la información y usabilidad en la web [En Línea] [Citado: 17 de enero del 2010]
<http://www.mantruc.com/files/art-epi-2004.pdf>
- [17] Lic. Antonio Montes. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información [En Línea] [Citado: 17 de enero del 2010]
http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm
- [18] Definición de accesibilidad. [En Línea] [Citado: 17 de enero del 2010] <http://definicion.de/accesibilidad/>

Referencias Bibliográficas

- [19] Eduardo Manchón. ¿Qué es la usabilidad? Definición de Usabilidad. [En Línea] [Citado: 17 de enero del 2010] <http://www.ingeneriasimple.com/usabilidad/QueEsUsabilidad.pdf>
- [20] Xavier Ferré Grau. Principios Básicos de Usabilidad para Ingenieros Software [En Línea] [Citado: 17 de enero del 2010] <http://www.willydev.net/Descargas/Articulos/General/usabilidad.pdf>
- [21] 3D-DOCTOR. [En Línea] [Citado: 23 de enero del 2010] <http://www.3d-doctor.com>
- [22] Carlos Pérez Risquet, Juan Carlos Ortega Camacho, Andrés Ortega Camacho, Romel Vázquez Rodríguez. Las Técnicas de visualización científica: Tipos de datos, aéreas de aplicación y sistemas de visualización.
- [23] 3DSlicer. [En Línea] [Citado: 23 de enero del 2010] <http://www.worldlingo.com/ma/enwiki/es/3DSlicer>
- [24] D. Ronda, O. Ferrer, N. A. Álvarez. IMAGIS: Sistema para la transmisión de imágenes médicas multimodales. [En Línea] [Citado: 23 de enero del 2010] <http://www.hab2001.sld.cu/arrepdf/00190.pdf>
- [25] Rocío Gutiérrez González, Raúl Sánchez González. Comparación entre las bibliotecas graficas GTK y Qt. [En Línea] [Citado: 25 de enero del 2010]
<http://yboon.net/~azamudio/linux/Comparacion%20entre%20las%20bibliotecas%20graficas%20GTK%20y%20Qt.pdf>
- [26] Gtk. [En Línea] [Citado: 23 de enero del 2010] <http://es.wikipedia.org/wiki/GTK%2B>.
- [27] Toolkit para Widgets. [En Línea] [Citado: 23 de enero del 2010]
http://eqaula.org/eva/file.php/1011/moddata/forum/301/2184/toolkits_para_Widgets.doc
- [28] Sergio Lujan Mora. C++ paso a paso.
- [29] María A. Mendoza Sánchez. Metodologías De Desarrollo De Software. [En Línea] [Citado: 4 de febrero del 2010]
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

Referencias Bibliográficas

- [30] Mtra. María de Lourdes Santiago Zaragoza. Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado RUP). [En Línea] [Citado: 4 de febrero del 2010]
<http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>
- [31] Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado de Desarrollo de Software.
- [32] James Rumbaugh, Ivar Jacobson, Grady Booch. El Lenguaje Unificado de Modelado.
- [33] Kendall, Kenneth y Kendall, Julie. Análisis y Diseño de Sistemas. S.I.: Prentice-Hall.
- [34] Modelo de Dominio. [En Línea] [Citado: 18 de marzo del 2010]
http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
- [35] María de Lourdes Bravo Estévez, Universidad de Cienfuegos. Cuba. José Joaquín Arrieta Gallastegui, Universidad de Oviedo. España. El método Delphi. Su implementación en una estrategia didáctica para la enseñanza de las demostraciones geométricas. [En Línea] [Citado: 20 de mayo del 2010]
<http://www.rieoei.org/deloslectores/804Bravo.PDF>
- [36] Eneko Astigarraga. EL MÉTODO DELPHI. [En Línea] [Citado: 20 de mayo del 2010]
http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan%20Diego/Plnaifi_Cuencas_Pregrado/Sept_29/Metodo_delphi.pdf
- [37] Zatsiorski, V. N. 1989. Metrología deportiva. Moscú: Editorial Planeta, 1989.

Bibliografía Consultada

1. Benjamin Roe. Diseño de Interfaces de Usuario Usables: Una Guía Rápida para Desarrolladores de Software Libre y de Código Abierto. [En Línea] [Citado: 30 de septiembre del 2009]
<http://mundogeek.net/traducciones/interfaces-usuario-usables/gui.html>
2. Joel Spolsky. Diseño de Interfaz de Usuario para Programadores Capítulo 6: Diseñando para gente que tiene cosas mejores que hacer con su vida. [En Línea] [Citado: 30 de septiembre del 2009]
http://willydev.net/insitecreation/v1.0/descargas/WillyDev_InterfazdeUsuarioparaDesarrolladores6.pdf
3. Carlos Marrero Expósito. Interfaz Gráfica de Usuario. Aproximación semiótica y cognitiva. [En Línea] [Citado: 30 de septiembre del 2009]
http://www.chr5.com/investigacion/investiga_igu/igu_aproximacion_semio-cognitiva_by_chr5.pdf
4. Lizbeth Luna González. El diseño de Interfaz Gráfica de Usuario para publicaciones digitales. [En Línea] [Citado: 30 de septiembre del 2009] http://www.revista.unam.mx/vol.5/num7/art44/ago_art44.pdf
5. Heidi Ruiz Villamil, Pedro Cobo Parra. Interfaz gráfico de usuario para el diseño de absorbentes microperforados múltiples. [En Línea] [Citado: 8 de octubre del 2009]
http://www.sea-acustica.es/Buenos_Aires_2008/a-079.pdf
6. Eugenio Jacobo Hernández Valdelamar. Consideraciones para el diseño de interfaces de usuario en sistemas operativos. [En Línea] [Citado: 8 de octubre del 2009]
<http://www.scribd.com/doc/2522263/Consideraciones-para-el-diseno-de-interfaces-de-usuario-en-sistemas-operativos>
7. Jaime Badia, Marta González, Ramon Salat. Usabilidad en el diseño de una interfaz de usuario para un entorno industrial. [En Línea] [Citado: 15 de octubre del 2009] <http://www.aipo.es/articulos/3/69.pdf>

8. Alejandro Baez, Cristian Castañeda, Diego Castañeda. Metodología para el diseño y desarrollo de interfaces de usuario. [En Línea] [Citado:15 de octubre del 2009]
[http://pegasus.javeriana.edu.co/~fwj2ee/descargas/metodologia\(v0.1\).pdf](http://pegasus.javeriana.edu.co/~fwj2ee/descargas/metodologia(v0.1).pdf)
9. Noe Sierra Romero, Sergio V. Chapa Vergara. Diseño de Interfaces Visuales. [En Línea] [Citado:15 de octubre del 2009] <http://www.cs.cinvestav.mx/CursoVis/prinvisual.html>
10. Yusef Hassan, Francisco J. Martín Fernández, Ghzala Iazza. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. [En Línea] [Citado: 23 de octubre del 2009]
<http://www.hipertext.net/web/pag206.htm#Usabilidad%20y%20accesibilidad>
11. Antonio Montes de Oca. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información. [En Línea] [Citado: 23 de octubre del 2009]
http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm
12. Jesús Tramullas Saz. Arquitectura de la información: más que diseño, hacia la findability. [En Línea] [Citado: 8 de noviembre del 2009] <http://www.sedic.es/clip39.pdf>
13. Marta Gómez Reyes. Arquitectura de la Información: Algo más que un Concepto. [En Línea] [Citado: 8 de noviembre del 2009]
<http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH01fb.dir/doc.pdf>
14. Jesús Bustamante. La arquitectura de la información del siglo XX al XXI. [En Línea] [Citado: 8 de noviembre del 2009] <http://www.elprofesionaldelainformacion.com/contenidos/2003/marzo/4.pdf>
15. Catherine Roy. Accesibilidad. [En Línea] [Citado: 15 de noviembre del 2009]
<http://vecam.org/article611.html>
16. David Moner Cano, Jordi Sabaté Alsina. USABILIDAD. [En Línea] [Citado: 15 de noviembre del 2009]
http://www.unal.edu.co/documentos/servicio_web/usabilidad_para_web.pdf

17. Ana M. Moreno, Maribel Sánchez-Segura. Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico. [En Línea] [Citado: 15 de noviembre del 2009]
<http://www.willydev.net/descargas/prev/PatronesUsa.pdf>
18. Luis González. Programacion avanzada en Java. [En Línea] [Citado: 25 de noviembre del 2009]
<http://eqaula.org/eva/file.php/1011/moddata/forum/301/2144/Widgets.pdf>
19. Alberto Molpeceres. Procesos de desarrollo RUP, XP, FDD. [En Línea] [Citado: 25 de noviembre del 2009] <http://www.willydev.net/descargas/articulos/general/cualxpddrup.PDF>
20. Rational Unified Process. Best Practices for Software Development Teams. [En Línea] [Citado: 25 de noviembre del 2009]
http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
21. Popkin Software and Systems. Modelado de Sistemas com UML. [En Línea] [Citado: 28 de noviembre del 2009] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>
22. Herramientas Case. [En Línea] [Citado: 28 de noviembre del 2009]
<http://jhoel-jp.tripod.com/paginas/herramientascase.pdf>

Anexos

Anexo 1 [Caso de Uso Expandidos]

1. Caso de uso Cargar imágenes DICOM.

Caso de Uso:	Cargar imágenes DICOM	
Actores:	Médico Especialista	
Resumen:	El caso de uso se inicia cuando el médico desea cargar el fichero DICOM, el sistema muestra el cuadro de diálogo, el médico selecciona la imagen a cargar, el sistema muestra la imagen en la aplicación.	
Precondiciones:		
Referencias	RF1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y selecciona la opción cargar archivo DICOM.	1.1 El sistema le pide al usuario la ruta desde donde desea cargar la imagen.	
2. Selecciona el fichero DICOM a cargar.	2.1 El sistema comprueba si el fichero es válido (tamaño y formato). 2.2 El sistema muestra la imagen cargada.	

Prototipo de Interfaz	
Poscondiciones	El sistema guarda en memoria el fichero DICOM cargado

2. Caso de uso Mostrar vistas axial, sagital y coronal.

Caso de Uso:	Mostrar vistas axial, coronal y sagital	
Actores:	Médico Especialista	
Resumen:	El caso de uso se inicia cuando el médico desea visualizar una de las vistas (axial, coronal o sagital), el sistema muestra en la pantalla completa la vista seleccionada.	
Precondiciones:		
Referencias	RF2	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y carga la imagen.		
2. El médico selecciona mostrar vista.	2.1 El sistema muestra al usuario varias opciones de vistas: <ul style="list-style-type: none"> • Axial. • Coronal 	

	<ul style="list-style-type: none"> • Sagital. • 3D.
3. Selecciona la opción deseada.	3.1 El sistema muestra la imagen de acuerdo con la vista seleccionada.
Prototipo de Interfaz	
Poscondiciones	

3. Caso de uso Rotar imagen.

Caso de Uso:	Rotar imagen.	
Actores:	Médico Especialista	
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen que desea transformar, rota la imagen, el sistema muestra la imagen resultante.	
Precondiciones:		
Referencias	RF3	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	

1. El médico accede al sistema y carga la imagen a la que quiere realizar la transformación.	
2. Selecciona la opción Rotar.	<p>2.1 El sistema da la posibilidad de rotar la imagen de dos formas:</p> <p>a) Si desea rotar con el cursor, ir a la sección “Rotar con cursor”.</p> <p>b) Si desea rotar con teclas, ir a la sección “Rotar con tecla”.</p>
Sección “Rotar con Teclas”	
Acción del Actor	Respuesta del Sistema
1. El médico presiona una tecla para rotar la imagen.	<p>1.1 El sistema analiza la tecla presionada.</p> <p>1.2 El sistema muestra la imagen después de ser rotada.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1 Si la tecla presionada no es válida el sistema muestra un mensaje “tecla no válida”.
Prototipo de Interfaz	
Sección “Rotar con Cursor”	
Acción del Actor	Respuesta del Sistema

<p>1. El médico arrastra el cursor en el área donde se encuentra la imagen visualizada.</p>	<p>1.1 El sistema analiza la dirección del movimiento del cursor.</p> <p>1.2 El sistema rota la imagen en la dirección analizada.</p> <p>1.3 El sistema muestra la imagen después de ser rotada.</p>
<p>Prototipo de Interfaz</p>	
<p>Poscondiciones</p>	<p>Se muestra la imagen rotada.</p>

4. Caso de uso Escalar imagen.

<p>Caso de Uso:</p>	<p>Escalar imagen</p>
<p>Actores:</p>	<p>Médico Especialista</p>
<p>Resumen:</p>	<p>El médico puede ampliar la imagen y disminuirla a conveniencia, ya sea toda la imagen o una región específica utilizando una lupa, en esta última forma el cursor toma forma rectangular ampliando la parte de la imagen donde este posicionado el cursor.</p>
<p>Precondiciones:</p>	
<p>Referencias</p>	<p>RF4</p>
<p>Prioridad</p>	<p>Secundario</p>
<p>Flujo Normal de Eventos</p>	
<p>Sección "Principal"</p>	

Acción del Actor	Respuesta del Sistema
1. El médico accede al sistema y carga la imagen a la que quiere realizar la transformación.	
2. Selecciona la opción escalar.	2.1 El sistema da la posibilidad de seleccionar el tipo de zoom. <ul style="list-style-type: none"> • Zoom In. • Zoom Out. • Lupa.
3. Selecciona una de las opciones.	3.1 El sistema da la posibilidad de escalar de dos formas: <p>a) Si desea escalar con el scroll, ir a la sección “Escalar con Cursor”.</p> <p>b) Si desea escalar con teclas, ir a la sección “Escalar con Teclas”.</p>
Sección “Escalar con Teclas”	
Acción del Actor	Respuesta del Sistema
1. El médico presiona una tecla para escalar la imagen.	1.1 El sistema analiza la tecla presionada. 1.2 El sistema muestra la imagen más grande o más pequeña según lo seleccionado.
Flujos Alternos	

Acción del Actor		Respuesta del Sistema
		1.1 Si la tecla presionada no es válida el sistema muestra un mensaje “tecla no válida”.
Prototipo de Interfaz		
Sección “Escarlar con Scroll”		
Acción del Actor		Respuesta del Sistema
1. Selecciona el tipo deseado y procede a arrastrar el cursor en el área donde se encuentra la imagen visualizada.		1.1 El sistema analiza la dirección del movimiento del scroll. 1.2 El sistema muestra la imagen más grande o más pequeña según lo seleccionado.
Prototipo de Interfaz		
Poscondiciones		

5. Caso de uso Deshacer.

Caso de Uso:	Deshacer acciones.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona deshacer o rehacer acciones, el sistema busca en el historial de cambios realizados el último que se efectuó, deshace dicho cambio, muestra la imagen luego de haber realizado la acción.

Precondiciones:	Debe de haberse realizado una acción.	
Referencias	RF5	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico selecciona la opción deshacer o pulsa la combinación de teclas Control + z.	1.1 El sistema busca en el historial de cambios realizados el último cambio que se efectuó y deshace dicho cambio. 1.2 El sistema muestra la imagen en su estado anterior.	
Prototipo de Interfaz		
Poscondiciones		

6. Caso de uso Rehacer acciones.

Caso de Uso:	Rehacer acciones.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona deshacer o rehacer acciones, el sistema busca en el historial de cambios desechos el último cambio desecho, rehace dicho cambio, muestra imagen en su estado posterior.

Precondiciones:	La última acción del médico tiene que haber sido deshacer un cambio.	
Referencias	RF6	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico selecciona la opción rehacer o pulsa la combinación de teclas Control + y.	1.1 El sistema busca en el historial de cambios desechos el último cambio desecho y rehace dicho cambio. 1.2 El sistema muestra la configuración de la imagen en su estado posterior.	
Prototipo de Interfaz		
Poscondiciones		

7. Caso de uso Aplicar Filtros.

Caso de Uso:	Aplicar Filtros.
Actores:	Médico Especialista
Resumen:	El caso de uso inicia cuando el médico selecciona la imagen que desea transformar, selecciona el tipo de filtro, el sistema muestra la imagen que queda como resultado de aplicarle el filtro.

Precondiciones:	
Referencias	RF7
Prioridad	Secundario
Flujo Normal de Eventos	
Sección "Principal"	
Acción del Actor	Respuesta del Sistema
1. El médico accede al sistema y selecciona la imagen a la que quiere realizar las transformaciones.	
2. Selecciona la opción Filtros.	2.1 El sistema da la posibilidad de seleccionar el tipo de filtro a aplicar mediante un menú.
3. El médico selecciona el filtro.	3.1 El sistema muestra la imagen después de ser aplicado el filtro.
Prototipo de Interfaz	
Poscondiciones	El sistema muestra en pantalla la imagen con el filtro aplicado.

8. Caso de uso Realizar mediciones sobre la imagen.

Caso de Uso:	Realizar mediciones sobre la imagen.
Actores:	Médico Especialista

Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen sobre la cual va a realizar la medición, escoge el tipo de medición, procede a ejecutar la medición, el sistema pinta la figura y da el resultado.	
Precondiciones:		
Referencias	RF8	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y selecciona la imagen a la que quiere realizar las transformaciones.		
2. Selecciona la opción realizar mediciones.	2.1 El sistema da la posibilidad de seleccionar el tipo de medición: <ul style="list-style-type: none"> • Distancia. • Ángulo. 	
3. Selecciona el tipo deseado y procede a ejecutar los movimientos del mouse en correspondencia con la selección.	3.1 El sistema dibuja los gráficos sobre la imagen. 3.2 El sistema da el resultado de la medición.	
Prototipo de Interfaz		

Poscondiciones	
-----------------------	--

9. Caso de uso Calcular volumen.

Caso de Uso:	Calcular Volumen	
Actores:	Médico Especialista	
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen a la que quiere calcular el volumen, selecciona calcular volumen, el sistema muestra el resultado en pantalla.	
Precondiciones:		
Referencias	RF9	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y selecciona la imagen a la que desea calcular el volumen.		
2. El médico selecciona la opción calcular volumen.	2.1 El sistema muestra los resultados del cálculo.	
Prototipo de Interfaz		
Poscondiciones		

10. Caso de uso Segmentación.

Caso de Uso:	Segmentación	
Actores:	Médico Especialista	
Resumen:	El caso de uso se inicia cuando el médico selecciona la imagen que desea transformar, el usuario selecciona segmentar, procede a dar click en la parte de la imagen que desea segmentar, el sistema muestra la nueva imagen con los resultados de la segmentación.	
Precondiciones:		
Referencias	RF10	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y selecciona la imagen que desea transformar.		
2. Selecciona la opción segmentar.	2.1 El sistema da la posibilidad de seleccionar el tipo de segmentación: <ul style="list-style-type: none"> • Segmentación automática. • Segmentación manual. 	

3. Selecciona una de las opciones.	3.1 El sistema ejecuta la opción seleccionada: a) Si es automática, ir a la sección "Segmentación Automática". b) Si es manual, ir a la sección "Segmentación manual".
Sección "Segmentación Automática"	
Acción del Actor	Respuesta del Sistema
	1. El sistema pide al médico que introduzca la cantidad de fragmentos en que desea dividir la imagen, así como el algoritmo que va a ejecutar la segmentación.
2. El médico introduce los datos.	2.1 El sistema muestra la imagen segmentada. 2.2 El sistema da la posibilidad de visualizar un fragmento de la imagen segmentada.
3. Selecciona el fragmento a visualizar.	3.1 El sistema muestra el fragmento de la imagen segmentada.
Prototipo de Interfaz	
Sección "Segmentación Manual"	
Acción del Actor	Respuesta del Sistema

1. El médico procede a dar click en parte de la imagen que desea segmentar.	1.1 El sistema visualiza la región de la imagen segmentada.
Prototipo de Interfaz	
Poscondiciones	

11. Caso de uso Reconstrucción 3D.

Caso de Uso:	Reconstrucción 3D	
Actores:	Médico Especialista	
Resumen:	El médico selecciona la imagen que desea transformar, selecciona la opción Reconstrucción 3D, el sistema muestra en pantalla la imagen reconstruida.	
Precondiciones:		
Referencias	RF11	
Prioridad		
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1. El médico accede al sistema y selecciona la imagen que desea transformar.		
2. El médico selecciona Reconstrucción 3D.	2.1 El sistema da la posibilidad de	

	<p>seleccionar:</p> <ul style="list-style-type: none"> • Surface Rendering. • Volume Rendering. • Planar Rendering.
3. Selecciona la opción deseada.	<p>3.1 El sistema ejecuta la opción seleccionada:</p> <p>a) Si es Surface, ir a la sección “Surface Rendering”.</p> <p>b) Si es planar, ir a la sección “Planar Rendering”.</p> <p>c) Si es volumen, ir a la sección “Volume Rendering”.</p>
Sección “Planar Rendering”	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Genera el grid 3D. 2. Genera la textura 3D a partir del grid 3D creado. 3. Aplica la textura 3D a la geometría proxy. 4. Visualiza el modelo 3D con los valores de los cortes por defecto. 5. Da la posibilidad de cambiar los valores de corte.

6. El médico puede o no cambiar los valores de los cortes.	6.1 Visualiza el modelo 3D generado con los valores de los cortes introducidos.
<i>Prototipo de Interfaz</i>	
Sección “Surface Rendering”	
Acción del Actor	Respuesta del Sistema
	1. El sistema pide al usuario que especifique la superficie, el tamaño del cubo y si desea la vista con malla o no.
2. El médico introduce los valores que el sistema le pide.	2.1 El sistema reconstruye el modelo 3D luego de procesar los valores introducidos por el usuario.
<i>Prototipo de Interfaz</i>	

Anexo 2 [Encuesta de autovaloración para determinar el coeficiente de competencia]

Compañero (a):

En la investigación que se lleva a cabo para obtener el título de Ingenieros en Ciencias Informáticas y que tiene como objetivo **“desarrollar una interfaz gráfica de usuario que cumpla con los estándares internacionales de diseño para facilitar al acceso e iteración con las funcionalidades del software”**, se requiere someter la propuesta al criterio de expertos a partir de la aplicación del Método Delphi. Teniendo en cuenta que los autores consideran que Ud. reúne los requisitos que lo avalan como experto, se le solicita tenga a bien responder las preguntas que a continuación se presentan. Su valoración y los criterios que emita serán de gran valor para el perfeccionamiento de la propuesta.

Para ello se necesita conocer el grado de dominio que usted posee acerca del tema de investigación y con ese fin se desea que responda lo que se le pide a continuación.

Nombre(s) y Apellidos: _____

Centro de trabajo: _____

Cargo que ocupa: _____

Años de experiencia: _____

Especialidad: _____

1. Se le solicita que valore su nivel de competencia sobre la problemática que se propone, marcando con una cruz el valor que considere. Tenga en cuenta el valor 1 como no tener ningún conocimiento y 10 el de pleno conocimiento de la problemática tratada.

1	2	3	4	5	6	7	8	9	10

2. Valore el grado de influencia que cada una de las fuentes que se le presenta a continuación ha tenido en su conocimiento y criterios sobre el tema que se presenta.

Fuentes de argumentación	Alto	Medio	Bajo
Análisis realizado por usted.			
Experiencia de trabajo en esta actividad.			
Trabajos de autores nacionales.			
Trabajos de autores internacionales.			
Su propio conocimiento del estado del problema en el extranjero.			
Su intuición.			

Muchas gracias por su valiosa colaboración.

Anexo 3 [Encuesta a expertos para la validación de la Interfaz Propuesta]

Con el objetivo de validar la solución propuesta se han elaborado un conjunto de preguntas que permiten medir la efectividad de la misma. De antemano se le asegura total anonimato y se garantiza que sus opiniones se tendrán en cuenta para la posterior aplicación del proceso.

La propuesta del Interfaz y documentación se encuentra adjunta a esta encuesta.

Para la mejor comprensión de sus criterios se dividieron los resultados en 5 rangos: Muy Adecuado (C1), Bastante Adecuado (C2), Adecuado (C3), Poco Adecuado (C4) y No Adecuado (C5).

No.	Preguntas	Criterio del Experto				
		C1	C2	C3	C4	C5
1.	¿La interfaz muestra al usuario toda la información y las herramientas necesarias para cada etapa en su trabajo?					
2.	¿Son adecuados los colores utilizados en la interfaz?					
3.	¿Cree usted que la propuesta cuenta con las funcionalidades básicas para un software de este tipo?					
4.	¿Considera usted que la tipografía que se utiliza es correcta?					
5.	¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?					
6.	¿Considera correcta la distribución espacial de los menús?					
7.	¿Considera usted que interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?					

8.	¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?					
9.	¿Considera usted que la relación entre menús y submenús es coherente?					
10.	¿Considera usted que el diseño de la interfaz es lo suficientemente sencilla y al mismo tiempo robusta?					

Le agradecemos cualquier sugerencia o recomendación. Por favor, refiéralas a continuación.

Sobre la interfaz: _____

Muchas gracias por su valiosa colaboración.

Glosario de Términos.

3D: Abreviatura de tridimensional, se utiliza para hacer referencia a algo que posea o pueda ser representado en tres dimensiones.

API: Application Programming Interface o Interfaz de Programación de Aplicaciones.

Arquitectura de la Información: Disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos.

C: Lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

C++: Pronunciado "ce más más" o "ce plus plus") es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

CASE: Computer Aided Software Engineering, herramientas de ingeniería de software asistida por computadora.

Framework: En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GPL: Licencia Pública General (General Public License), es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software.

GTK: GTK+ o The GIMP Toolkit, grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI).

GUI: Graphic User Interface o Interfaz Gráfica de Usuario.

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment) es un programa compuesto por un conjunto de herramientas para un programador.

Interfaz de Usuario: Medio que el usuario utiliza para comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

Multiplataforma: Término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

QT: Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

RUP: Proceso Unificado Racional (Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

UML: Unified Modeling Language. Lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

Widget: Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de Widgets o Widget Engine. Entre sus objetivos están: dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.