

Universidad de las Ciencias Informáticas

Facultad 2



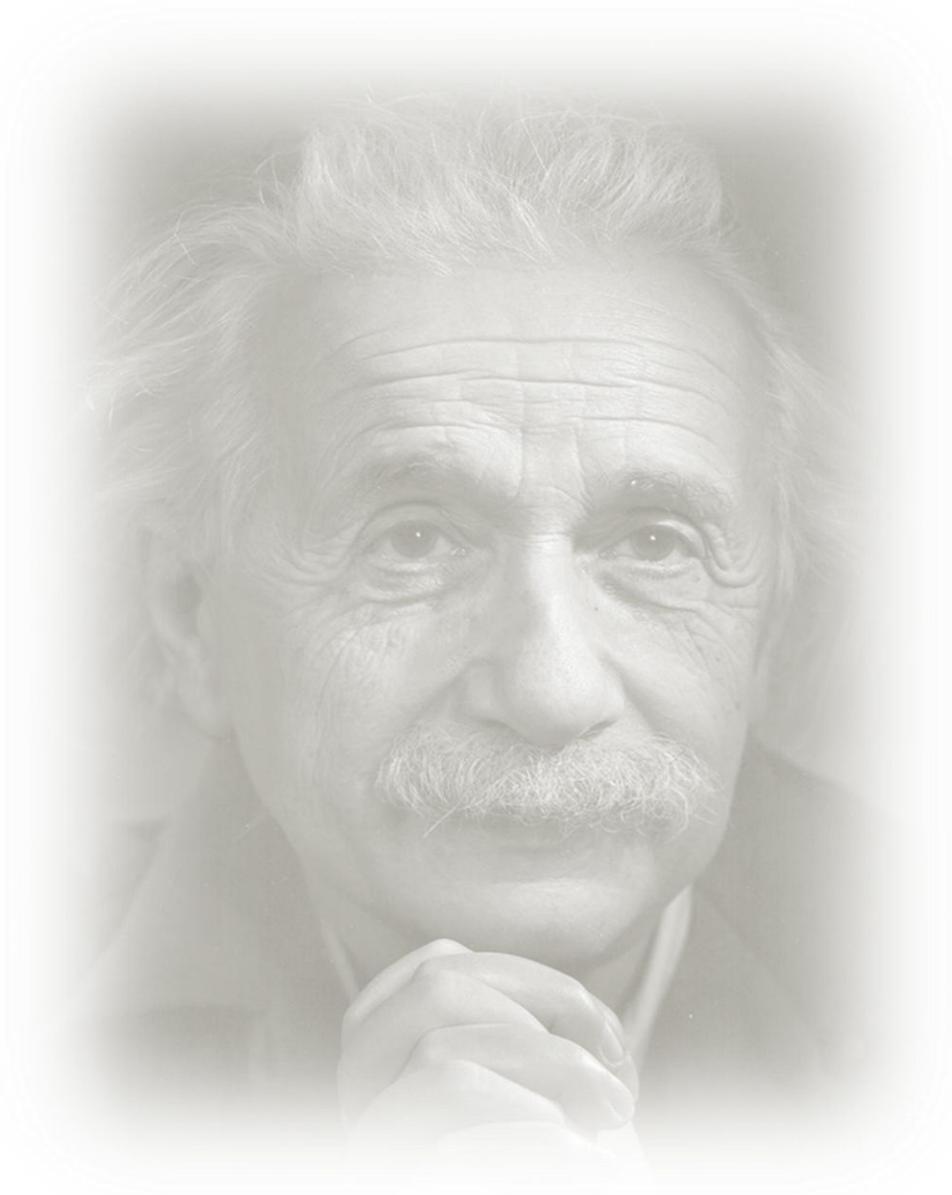
# Desarrollo del módulo de Simulaciones de osteotomía para el sistema de Planificación Quirúrgica Ortopédica

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autores: Alberto Pons Ramírez  
Orlando Pérez Rodríguez

Tutores: Ing. Yoel González Mesa  
Ing. Yoandy González Yera

La Habana, 17 de junio de 2014  
“Año 56 de la Revolución”



*” Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”.*

*Albert Einstein*

## DATOS DE CONTACTO

### Tutores:

**Ing. Yoel González Mesa:** Ing. en Ciencias Informáticas graduado del curso 2008-2009 en la Universidad de Ciencias Informáticas de La Habana. Posee categoría docente de Instructor. Ha impartido las asignaturas de Procesamiento Digital de Imágenes y Señales (2do Perfil), Programación IV, Debate Histórico Contemporáneo y Práctica Profesional. Es Especialista del Departamento de Software Médico Imagenológico (SWMI) del Centro de Informática Médica (CESIM). Ha participado en el desarrollo de sistemas como el Visor web y se desempeña como Jefe de Proyecto del Planificador Quirúrgico Ortopédico.

Correo electrónico: [ygmesa@uci.cu](mailto:ygmesa@uci.cu)

**Ing. Yoandy González Yera:** Ingeniero en Ciencias Informáticas graduado del curso 2012-2013. Recién Graduado en Adiestramiento. Trabajador del Centro de Informática Médica (CESIM), en el Departamento Software Médico Imagenológico (SWMI).

Correo electrónico: [yyera@uci.cu](mailto:yyera@uci.cu)

## DEDICATORIA

### *De Alberto:*

*A mi mamá, por ser mi musa en todo momento, mi faro, mi luz, mi razón de ser y de vivir.*

*A mi papá, por enseñarme a soñar, a ambicionar. Por demostrarme que ser auténtico nos hace único y principalmente por inculcarme la profunda convicción de que la gloria es sólo para aquellos hombres que siguen sus grandes ideas hasta el final.*

*A mis hermanos, por su confianza, su lealtad y por tener la magia de sacarme una sonrisa en todo momento.*

### *De Orlando:*

*Dedico esta tesis a mi mamá, mi papá y hermano por ser las personas más importantes en mi vida, por apoyarme todos estos años, por confiar en mí. Por ustedes estoy aquí, cumpliendo mis sueños de graduarme como ingeniero. Los Amo.*

*Dedico esta tesis también a toda mi familia, mis tíos, mis primos, mis abuelos por confiar en mí y estar siempre a mi lado.*

## **AGRADECIMIENTOS**

### *De Alberto:*

*A mis padres, por su ejemplo, su esfuerzo y su amor desmedido, por nunca haber perdido la confianza en mí y por su completa dedicación a mi formación como un hombre de bien. Los amo.*

*A mi hermano, por ser mi amigo y mi confidente.*

*A mi hermanita, por ser mi amiga y por impulsarme a ser cada día mejor para servirle de ejemplo.*

*A mis abuelos, por su amor, su ternura y por crear la mejor familia del mundo.*

*A mis tíos y primos, por hacer de las páginas de mi vida un libro que jamás dejaré de ojear.*

*A mi novia, por todo su apoyo y por tantos lindos momentos. Te amo princesa.*

*A mis suegros, por recibirme como un hijo más.*

*A mis amigos.*

*A mi compañero de tesis, por estar en los buenos y malos momentos durante todo este trayecto universitario, y a su familia, por hacerme partícipe de su hogar como uno más de ellos.*

*A mis tutores, por su apoyo incondicional.*

*A mis educadores.*

*A Yoandy, por su paciencia, su dedicación, su entrega y por ser además de tutor un amigo.*

*A todos los que de una forma u otra me brindaron su ayuda.*

## **De Orlando:**

*Le agradezco a mi compañero de tesis, por las batallas recorridas juntos en estos 5 años. A todos los amigos que formaron parte de mi recorrido por la universidad.*

*A mis amigos de convivencia en el apartamento, así como a las niñas del grupo, especialmente Heidi y a Digmaris, alias la mía.*

*Le agradezco al tribunal en general, por haber estado disponible para aclarar cualquier duda que pudiera tener en algún tema específico. A mi oponente, la cual no dudó en ayudarnos, siempre que estuviera a su alcance. A los tutores en general, por su apoyo a lo largo de este año.*

*Un agradecimiento especial a Yoandy alias el yoyo, que más que un tutor, fue un amigo y hermano, brindándonos su ayuda en todo cuánto necesitábamos. Siempre estuvo pendiente a nuestros avances durante este año.*

*A mis hermanos allá en Guanabacoa, el Rafi, Carlos, mis hijos al FIFA. Gracias por estar siempre presentes ante cualquier situación en la que haya necesitado su ayuda.*

*Un especial agradecimiento para mis abuelos, estando siempre presentes desde mi infancia hasta hoy, participando en la formación de la persona que soy.*

*A mi tío Juan, por estar siempre aconsejándome y ayudándome a decidir, cuál es el mejor entre dos caminos. A mi tía Laura y su esposo, que aunque hoy no pudieron estar aquí, siempre han estado presentes de alguna manera, brindándome su apoyo.*

*A mis padres y hermano, les debo mi vida. Ellos han sido la principal motivación para llegar a este momento. A ustedes les debo parte de mi crecimiento personal. Les agradezco por su esfuerzo y dedicación, día a día.*

*En general les agradezco a todos por ser las personas que son y por haber estado presentes en alguna oración, párrafo, capítulo, o tal vez en toda la historia de mi vida. Es todo.*

## RESUMEN

Cuba no cuenta con un sistema que apoye la planificación preoperatoria ortopédica. Actualmente la Universidad de las Ciencias Informáticas (UCI) está desarrollando una aplicación que dé solución a esta problemática, llamado Planificador Quirúrgico Ortopédico (PQO). Sin embargo, este software no brinda facilidades para su uso en las osteotomías.

La presente investigación tiene como objetivo desarrollar un módulo de simulaciones de osteotomías e integrarlo al PQO. Para el desarrollo de la aplicación se realizó un estudio de las tendencias actuales en Cuba y en el mundo, relacionadas con el tema.

La aplicación fue desarrollada sobre la plataforma .NET, haciendo uso del lenguaje de programación C# en su versión 4.0 y utilizando el Framework 4.0. Se dispuso además de las tecnologías WPF y MEF, así como MVVM, IoC y DI como patrones arquitectónicos. Para el modelado del sistema se utilizó la herramienta CASE Enterprise Architect en su versión 7.5 y el Lenguaje Unificado de Modelado para la creación de los diagramas correspondientes a las fases de desarrollo del software. El módulo fue integrado al sistema haciendo uso de un mecanismo basado en extensiones.

El desarrollo del módulo de simulaciones representará un avance significativo para la ortopedia cubana. Permitirá que los especialistas tengan una clara visión de las condiciones en las que se llevará a cabo el procedimiento quirúrgico, y por ende, reducirá el tiempo de ejecución del mismo. Posibilitará que el paciente sea sometido a cirugías menos traumáticas y constituirá una vía para la formación de los futuros cirujanos.

### **Palabras claves:**

Cirugías, planificación preoperatoria ortopédica, simulaciones de osteotomías

# TABLA DE CONTENIDOS

INTRODUCCIÓN .....	1
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....</b>	<b>6</b>
1.1. PLANIFICACIÓN QUIRÚRGICA ORTOPÉDICA .....	6
1.2. OSTEOTOMÍA .....	6
1.3. SISTEMAS DE TRANSMISIÓN, ALMACENAMIENTO Y VISUALIZACIÓN DE IMÁGENES MÉDICAS .....	8
1.4. ESTÁNDAR DICOM .....	9
1.5. DIAGNÓSTICO ASISTIDO POR COMPUTADORAS .....	10
1.6. CIRUGÍA ORTOPÉDICA ASISTIDA POR COMPUTADORAS .....	10
1.7. SISTEMAS INFORMÁTICOS EXISTENTES .....	11
1.8. TECNOLOGÍAS, PATRONES Y LENGUAJES DE PROGRAMACIÓN A UTILIZAR .....	14
1.9. MODELO DE CALIDAD, LENGUAJE Y NOTACIÓN DE MODELADO .....	18
1.10. HERRAMIENTAS USADAS PARA LA SOLUCIÓN PROPUESTA .....	20
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>22</b>
2.1. MODELO DE DOMINIO .....	22
2.2. SISTEMA PROPUESTO .....	23
2.3. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE .....	24
2.4. DEFINICIÓN DE LOS ACTORES DEL SISTEMA.....	29
2.5. DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	29
2.6. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA .....	30
2.7. PROPUESTA DE CASOS DE USO POR CICLOS DE DESARROLLO.....	33
<b>CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>34</b>
3.1. DISEÑO.....	34
3.2. DESCRIPCIÓN DE LAS CLASES.....	35
3.3. MODELO ARQUITECTÓNICO .....	35
<b>CAPÍTULO 4. IMPLEMENTACIÓN.....</b>	<b>38</b>
4.1. DIAGRAMA DE COMPONENTES.....	38
4.2. DIAGRAMA DE DESPLIEGUE.....	40

4.3. ESTÁNDARES DE CODIFICACIÓN .....	41
<b>CONCLUSIONES .....</b>	<b>44</b>
<b>RECOMENDACIONES.....</b>	<b>45</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>46</b>
<b>BIBLIOGRAFÍA .....</b>	<b>49</b>
<b>ANEXOS .....</b>	<b>52</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>69</b>

## **INTRODUCCIÓN**

El sistema nacional de salud cubano es gratuito y accesible a todos los ciudadanos. Esto se manifiesta en la red de unidades asistenciales del territorio nacional, en los sistemas de primer, segundo y tercer nivel de atención médica y a través de los diferentes programas priorizados. Como centro de interés sitúa por igual el cuidado de todos los ciudadanos, así como el control de enfermedades que los afectan. Ofrece además, los adelantos de la ciencia y la técnica con los que cuenta la nación, sin distinción de política, raza o religión.

Las imágenes médicas, ejemplos fehacientes de los adelantos, han tenido un gran auge dentro de la medicina, dando origen a modalidades como: ultrasonido (US, por sus siglas en inglés), tomografía computarizada (CT, por sus siglas en inglés), imagen por resonancia magnética (MRI, por sus siglas en inglés), radiografía digital (CR, por sus siglas en inglés), entre otras. Todas estas modalidades constituyen una base fundamental para los diagnósticos médicos dentro de los sistemas de salud modernos a nivel mundial (1).

Con el progreso vertiginoso y el empleo de las modalidades diagnósticas, junto a la gran variedad de equipos de adquisición de imágenes incorporados, unido al desarrollo de la informática médica, las redes de comunicación y los medios de almacenamiento, surgen los Sistemas de Transmisión, Almacenamiento y Visualización de Imágenes Médicas (PACS, por sus siglas en inglés). Estos son los encargados de la adquisición, almacenamiento, visualización y transmisión de imágenes médicas. Su objetivo principal es permitir el funcionamiento de un servicio de imágenes sin la necesidad de archivarlas en documentos de papel o películas (2).

En aras de establecer un formato para el intercambio de información de las imágenes médicas entre los diferentes elementos que componen los sistemas PACS, surge el estándar para la Comunicación y la Imagenología Digital en la Medicina (DICOM, por sus siglas en inglés). Es considerado el estándar industrial para la transferencia y visualización de imágenes médicas digitales y la información asociada a ellas. DICOM describe el formato de archivos y la especificación de los datos primordiales de un paciente en la imagen, describiendo un lenguaje común a distintos sistemas médicos. De esta forma las imágenes pueden venir acompañadas de mediciones, cálculos e información descriptiva relevante para diagnósticos (3).

Para realizar el diagnóstico de las imágenes médicas, a través del procesamiento digital de las mismas, se utilizan los sistemas de Diagnóstico Asistido por Computadora (CAD por sus siglas en inglés). Estos

constituyen sistemas de apoyo a la medicina para la toma de decisiones. Dentro de los sistemas CAD se encuentran los sistemas de Cirugía Ortopédica Asistida por Computadora (CAOS por sus siglas en inglés). Estos tienen como objetivo principal, asistir a los cirujanos ortopédicos durante el proceso de planificación preoperatoria.

La planificación preoperatoria consiste en la creación de modelos físicos de partes internas del organismo, con el fin de incrementar la información que utilizará el cirujano. Reduce la posibilidad de cometer algún error y adapta los dispositivos que serán utilizados a la geometría de cada paciente. Este proceso es el paso más importante antes de la intervención quirúrgica, por lo que su correcta ejecución es determinante en el éxito de la misma.

El Complejo Científico Ortopédico Internacional “Frank País” fundado y dirigido por el Profesor Dr.Sc. Rodrigo Álvarez Cambras, constituye un centro de máximo nivel y una referencia en las especialidades de ortopedia y traumatología en Cuba. Cuenta con las más avanzadas técnicas quirúrgicas y un equipo multidisciplinario de alto nivel científico y ético, que brinda su vasta experiencia a cada paciente. Sin embargo, no se cuenta con un sistema informático que apoye a este equipo y todos los demás especialistas de la rama en el país, durante la planificación quirúrgica ortopédica.

En la Universidad de las Ciencias Informáticas (UCI) existen varios proyectos de desarrollo de software destinados a informatizar la salud cubana. En el departamento de Software Médico Imagenológico (SWMI) del Centro de Informática Médica (CESIM), se encuentra actualmente en desarrollo un sistema para la Planificación Quirúrgica Ortopédica (PQO). Este sistema cuenta con las herramientas necesarias para la planificación quirúrgica en las especialidades de rodilla, cadera, extremidades, columna, trauma y pediatría. El mismo representa un avance significativo para la ortopedia cubana y constituye un punto de partida en el proceso de elevar las probabilidades de éxito de las cirugías. Sin embargo, no posee herramientas que permitan a los especialistas realizar simulaciones de las osteotomías como parte del proceso de planificación preoperatoria.

La osteotomía describe diversos procedimientos de corte de hueso que se usan para corregir una deformidad o reducirla, producir cambios estéticos o aliviar dolor artrítico. La deformidad se origina por artritis, traumatismo, o por anomalías del crecimiento, como por ejemplo, una longitud desigual de las piernas. Prácticamente cualquier hueso afectado se puede reparar mediante osteotomía, por ejemplo, los que componen las manos, las muñecas, los tobillos, los dedos de los pies, las piernas, los brazos, las

caderas y las mandíbulas. Además, cortar el o los huesos, permite reacomodar los extremos para alterar su forma o longitud, remodelar su aspecto, transferir puntos de estrés o componer fracturas.

En la actualidad en los hospitales ortopédicos cubanos, la planificación preoperatoria para las osteotomías se realiza de forma manual. Para ello se requiere de material adicional (plantillas, papel, lápiz, marcadores, regla, goniómetro) y una inversión de tiempo considerable para su puesta en práctica. En mucho de los casos, los especialistas guiados por su experiencia y condicionados por la falta de recursos, realizan la cirugía sin desarrollar una planificación previa. Esto puede acarrear errores significativos, conllevando a que las operaciones quirúrgicas sean imprecisas y poco efectivas.

Por lo antes planteado se identifica como **problema a resolver**: ¿Cómo incorporar los procedimientos de osteotomía al proceso de planificación quirúrgica ortopédica asistido por computadora?

Este problema se enmarca en el **objeto de estudio**: Los procesos de osteotomía.

El objeto delimita el **campo de acción**: Las simulaciones para los procesos de osteotomía durante la planificación quirúrgica ortopédica.

Para la solución del problema se plantea como **objetivo general**: Desarrollar el módulo de simulaciones de osteotomía para el sistema de Planificación Quirúrgica Ortopédica.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la investigación**:

- ✓ Realización de un análisis crítico y valorativo de los sistemas informáticos de planificación preoperatoria ortopédica, que cuentan con herramientas para realizar simulaciones de osteotomía existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
- ✓ Realización de un análisis crítico y valorativo de las tendencias, técnicas, tecnologías, plataformas, librerías, metodologías y herramientas usadas en la actualidad en el Planificador Quirúrgico Ortopédico desarrollado en el CESIM.
- ✓ Análisis de los procesos relacionados con la planificación preoperatoria ortopédica para las simulaciones de osteotomía.
- ✓ Asimilación de la Arquitectura definida en el Planificador Quirúrgico Ortopédico desarrollado en el CESIM.
- ✓ Generación de los artefactos correspondientes a las disciplinas: “Modelado del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación”.

- ✓ Implementación del módulo de simulación aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.

A continuación se presentan los **métodos científicos de la investigación** que fueron usados para confeccionar el diseño metodológico de la investigación en curso:

### Métodos teóricos:

- ✓ **Histórico lógico:** para la determinación de los antecedentes, tendencias y regularidades del objeto de estudio y del campo de acción.
- ✓ **Análisis sintético:** contribuyó al análisis de los documentos, permitiendo sintetizar, clasificar y evaluar la información valiosa y la posibilidad de buscar la esencia de los fenómenos y los rasgos que caracterizan el proceso de planificación quirúrgica ortopédica para las osteotomías.
- ✓ **Inductivo deductivo:** teniendo en cuenta el funcionamiento del proceso de planificación preoperatoria de las osteotomías, se evalúa la problemática existente para determinar aspectos particulares y desarrollar una propuesta de solución.
- ✓ **Modelación:** con la finalidad de confeccionar modelos y diagramas que facilitan la comprensión de los procesos a desarrollar como parte de la investigación.

### Métodos empíricos:

- ✓ **Entrevista:** para la determinación de las particularidades del objeto de estudio y del campo de acción, mediante las entrevistas realizadas a los especialistas del Hospital Militar Dr. Luis Díaz Soto. Ver Anexo 1.
- ✓ **Análisis bibliográfico:** para la apropiación del conocimiento, teniendo en cuenta la experiencia de los autores, así como la vigencia y la confiabilidad de los documentos consultados.

### Estructuración del contenido

**Capítulo 1:** En este capítulo se establecen las bases teóricas de la investigación. Se realiza un estudio del estado del arte de las principales herramientas que existen en el mundo de planificación quirúrgica ortopédica para las simulaciones de osteotomía. Se mencionan características de la metodología y las tecnologías empleadas en la solución del problema planteado.

**Capítulo 2:** Como parte de la propuesta de solución, se describen las características del sistema, se presenta el modelo de dominio, se exponen los principales conceptos asociados al mismo y las relaciones

que se establecen entre ellos. Se especifican los requisitos funcionales y no funcionales del sistema, así como los actores que intervienen en el mismo. Además, se modelan y describen los casos de uso.

**Capítulo 3:** En este capítulo se modelan los diagramas de clases del diseño y los diagramas de secuencia asociados a los casos de uso del sistema. Además, se exponen los patrones de diseño empleados en la solución, con una breve descripción de sus características y el propósito de su uso en el sistema. Se realiza una descripción de la arquitectura del sistema y de los patrones arquitectónicos empleados.

**Capítulo 4:** En este capítulo se representan los diagramas de componentes y de despliegue, así como la descripción de los componentes ejecutables del sistema. Se realiza una especificación sobre cómo van a estar desplegadas físicamente sus distintas partes y mediante qué protocolos se comunicarán. Además, se detallan los estándares de codificación utilizados para un mejor entendimiento del código.

### CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se expone el marco teórico y conceptual asociado a la problemática a resolver, para lograr un mayor entendimiento de la misma. Se muestran además las tendencias actuales del desarrollo de los sistemas de planificación preoperatoria ortopédica que existen en el mundo, que incluyen herramientas para realizar simulaciones de osteotomías. Se caracterizan las tecnologías, los patrones y los lenguajes de programación utilizados. Se describen las herramientas, la metodología, el modelo de calidad y la notación de modelado utilizadas.

#### 1.1. Planificación Quirúrgica Ortopédica

La planificación preoperatoria ortopédica es un procedimiento indispensable que debe realizarse previamente a la intervención quirúrgica y cuyos objetivos son: determinar el resultado final de la cirugía y establecer la táctica quirúrgica a seguir en el procedimiento quirúrgico (4).

Para tener éxito en el resultado final de la cirugía se deben tomar en cuenta los factores iniciales del contacto médico-paciente, realizar una historia clínica completa, un examen físico detallado de la región patológica, incluyendo pruebas especiales como son: radiografías, tomografía, resonancia magnética, ultrasonido, entre otros (4).

#### 1.2. Osteotomía

Las osteotomías siguen los principios de la intervención quirúrgica básica de hueso. Pueden efectuarse para reparar una fractura que se consolidó de manera inapropiada, o durante el reemplazo de la cadera o la rodilla. Típicamente, el cirujano elige el ángulo apropiado, extirpa o inserta una cuña de hueso, y después remodela los huesos o los reacomoda. Los nuevos extremos del hueso después se unen y se fijan en su lugar para permitir que se consoliden.

Existen diversos padecimientos específicos tratados mediante osteotomía. Algunos ejemplos de estos son: reparación de juanetes, piernas arqueadas, artritis debida a mala alineación de huesos y reparación de la longitud desigual de las piernas. Además es asistida la deformidad de barbilla sobre pecho, causada por espondilitis anquilosante, diversas deformidades estéticas y funcionales del maxilar superior e inferior.

### 1.2.1. Técnicas empleadas para la realización de osteotomías

#### ✓ Osteotomía tibial alta:

La artritis degenerativa y el alineamiento defectuoso pueden hacer que los tejidos protectores de la rodilla se desgasten más de un lado que del otro en un círculo de daño repetitivo. Un procedimiento conocido como osteotomía tibial alta abre el hueso superior del tobillo (tibia) en cuña para reorganizar la articulación de la rodilla. La parte de la rodilla que soporta peso pasa de tener un tejido degenerativo o gastado a tener un tejido sano. Una osteotomía tibial alta por lo general se considera un método que prolonga el tiempo antes de sea necesario un reemplazo de rodilla, debido a que los beneficios de aquella suelen desaparecer después de ocho a diez años (5).

Existe una variedad de procedimientos para corregir las afectaciones que son resueltas a través de este tipo de osteotomía, los cuales difieren en la manera y la posición en la cual se corta o da forma al hueso. En lo que se denomina una osteotomía de apertura en cuña, se realiza un corte en la tibia y se utilizan un tapón de hueso e instrumentos para sostener la cuña abierta y realinear la rodilla. En una osteotomía de cierre en cuña, se retira una cuña de hueso de la tibia, se cierra el espacio libre y los bordes se atan entre sí para el realineamiento (5).

#### ✓ Osteotomía femoral distal:

La osteotomía femoral es una intervención conservadora reservada a pacientes jóvenes (menores de 60 años) que presenten una gonartrosis femorotibial externa invalidante con un genu valgo superior a 10°. Su técnica de realización debe ser rigurosa y precisa, porque es difícil y los márgenes de error de la corrección son escasos. El objetivo de corrección es un varo de 0-4°. Si persiste un valgo postoperatorio, el paciente no mejorará y si la corrección es superior a 4° de varo, el resultado satisfactorio en un principio se deteriorará con rapidez, debido a la descompensación del compartimento interno. Se pueden realizar por vía interna o externa con resultados similares. Se debe utilizar un método de osteosíntesis rígida de tipo placa femoral o lámina-placa. Cualquier otro método, como los tornillos o las grapas de forma exclusiva, es insuficiente y da lugar a una pérdida rápida de la corrección, lo que culmina en un mal resultado clínico. Con independencia de la técnica utilizada, la intervención es difícil, porque es muy fácil provocar un callo vicioso en rotación, en flexo o en recurvatum. También es una intervención arriesgada, ya que puede causar complicaciones intraoperatorias graves, como lesiones vasculares o nerviosas (6).

- ✓ Osteotomía de ángulo abierto Smith-Petersen, Osteotomía poli segmentaria Wilson-Turkell y Osteotomía de ángulo cerrado Scudese-Thomassen:

Las deformidades de la columna vertebral pueden provocar cifosis torácica creciente o pérdida de la lordosis lumbar, lo que deriva en desequilibrio en el plano sagital. La deformidad fija puede llegar a ser tan severa que será imposible para el paciente levantar la vista al horizonte, sentarse, o adoptar posiciones erectas y debido al desplazamiento del centro de gravedad hacia delante, la marcha se hará cada vez más dispendiosa. Este tipo de deformidades puede tener efectos debilitantes desde los puntos de vista funcional y psicológico.

Estas osteotomías son utilizadas para corregir la cifosis toracolumbar:

- ✓ La osteotomía de Smith-Petersen (1945) posibilita una corrección de alrededor de 10° en el plano sagital, en cada nivel vertebral en el que se practica. Consiste en llevar la columna en hiperextensión hasta lograr la ruptura del ligamento común vertebral anterior y la apertura del espacio intervertebral. Es beneficiosa para pacientes que presentan un desequilibrio degenerativo en el plano sagital.
- ✓ En 1949 Wilson y Turkell publicaron la primera serie de correcciones con osteotomías poli segmentarias, en el espacio interlaminar y a través de las articulaciones superiores e inferiores de las vértebras adyacentes del espacio intervenido. Este sistema permitió una corrección más gradual y por consiguiente, con menores riesgos vascular y neurológico.
- ✓ En 1963 Scudese ideó la técnica de la osteotomía monosegmental intravertebral en cuña cerrada, resecaando el arco posterior y las facetas para terminar haciendo una osteotomía en cuña cerrada en el mismo cuerpo vertebral. De este modo se logra la corrección mediante extensión pasiva y controlada de la columna lumbar.

### **1.3. Sistemas de Transmisión, Almacenamiento y Visualización de Imágenes Médicas**

La evolución de la tecnología de la informática y de las redes de comunicación ha conducido al surgimiento de los sistemas PACS, como una alternativa al manejo de imágenes digitales en forma eficiente a través de dispositivos conectados en una red. Esto permite proveer servicios de almacenamiento, tratamiento y transferencia de información, para dar soporte a las áreas donde se genera un volumen importante de imágenes (7).

Los componentes básicos de un sistema PACS son: equipos médicos de adquisición de imágenes, red de comunicación, bases de datos, estaciones de diagnóstico y visualización, así como sistemas de almacenamiento a largo y corto plazo, todos ellos integrados a un sistema de gestión de la información (7). Estos componentes interactúan entre sí mediante el protocolo de red TCP/IP y bajo las normativas del estándar DICOM.

La importancia fundamental de los sistemas PACS radica en que permiten lograr la digitalización global e integración de los servicios de los departamentos de imagenología. Son los responsables de conseguir que las imágenes médicas estén disponibles en cualquier momento y en aquel lugar en que se necesiten. Se elimina la necesidad de procesamiento de placas impresas y los costos asociados, permite realizar seguimiento a largo plazo y la probabilidad de pérdida es ínfima (8).

### **1.4. Estándar DICOM**

DICOM es el estándar industrial para la transmisión, tratamiento e impresión de las imágenes médicas digitales. Fue creado por American College of Radiology (ACR por sus siglas en inglés) y National Electrical Manufacturers Association (NEMA por sus siglas en inglés), debido a la necesidad de un protocolo común de trabajo para todos los equipos.

El estándar DICOM define tanto los objetos de información como las clases de servicios, la estructura de datos y su semántica, los diccionarios de datos necesarios, los medios físicos de almacenamiento de información, la estructura lógica de almacenamiento de estudios, el soporte de redes de comunicación y para los dispositivos de visualización de imágenes la estandarización de los niveles de grises que deben soportar los monitores (9).

DICOM hace posible que los archivos médicos puedan viajar de forma segura entre hospitales, centros de investigación y departamentos. Luego esa información puede ser vista remotamente para que los médicos puedan diagnosticar desde su causa y buscar diferentes opiniones de otros expertos de una forma rápida y sencilla (10).

### 1.5. Diagnóstico Asistido por Computadoras

Los sistemas CAD son una combinación de técnicas de procesamiento de imágenes médicas y métodos inteligentes que pueden ser usados para mejorar el proceso de interpretación de estas. Dichos sistemas aportan una segunda opinión al especialista sobre un diagnóstico determinado. Su objetivo fundamental es ayudar al profesional a mejorar su trabajo, dirigiendo la atención del mismo hacia las zonas de la imagen más propensas a irregularidades.

Un modo habitual de evaluar la precisión diagnóstica es medir los valores de sensibilidad y especificidad conseguidos. Cuando se cuenta con la ayuda de un sistema de diagnóstico asistido por computadora, la imagen médica original es transformada en una segunda imagen en la que, mediante algún tipo de marcado, se resaltan las zonas sospechosas de incoherencias. Si el radiólogo trabaja con esta imagen modificada, posiblemente elabore un diagnóstico mejor del que haría con la imagen original. Por tanto la especificidad que se obtendrá en este caso, será mayor a la que se lograría sin el uso del sistema CAD. Lógicamente, el sistema será de utilidad si se consigue mejorar los valores de sensibilidad y especificidad respecto a la imagen (11).

Su uso trae consigo una disminución en el tiempo de lectura del caso de estudio y mayor calidad del diagnóstico. La eficiencia de estos sistemas ha sido probada mediante varios estudios y se ha comprobado que mejoran el rendimiento del radiólogo.

### 1.6. Cirugía Ortopédica Asistida por Computadoras

Los sistemas CAOS están incluidos dentro de los sistemas CAD. En ellos se aplica la tecnología informática con el fin de mejorar los resultados de los procedimientos quirúrgicos ortopédicos. Los objetivos que se traza la cirugía asistida por ordenador son: la mejora de la calidad, la seguridad y la capacidad de previsión de la intervención quirúrgica mediante el empleo de aparatos de alta tecnología (12).

Estos sistemas son empleados en muchos estudios y prácticas clínicas: cadera, pediatría, columna, rodilla, traumatología, cirugía, simulación y planificación preoperatoria. Permiten al cirujano realizar las técnicas quirúrgicas de manera más precisa y menos invasiva, lo que trae consigo una rápida rehabilitación en el paciente, que sufre menos dolor. La tecnología se utiliza, principalmente, para las operaciones en la columna vertebral, rodillas, caderas y áreas de trauma (12). Constituyen un significativo recurso tecnológico que continúa evolucionando día a día, con el fin de mejorar la calidad de vida de incontables pacientes.

### 1.7. Sistemas informáticos existentes

A raíz del análisis de los procesos que se efectúan en los centros ortopédicos y luego de haber identificado los principales problemas que enfrentan los especialistas, se hace un estudio en busca de sistemas informáticos existentes que brinden alguna solución. Además, se desea determinar cuáles de estos sistemas pudieran contribuir y servir de apoyo para el planteamiento de la solución a la cual se pretende arribar al culminar esta investigación.

#### 1.7.1. *Sistemas internacionales*

#### **OSTEOTOMIC**

OSTEOTOMIC es un software de simulación tridimensional de osteotomías creado para facilitar la intervención quirúrgica. A partir de datos anatómicos del paciente, ayuda al cirujano en la fase de planificación quirúrgica, permitiéndole elegir la ubicación y orientación óptima para los cortes. Es una aplicación escalable que dispone de una interfaz de usuario sencilla, con navegación 3D y de fácil aprendizaje para el clínico (13).

Visualiza diferentes formatos 3D y dispone de un descriptor propio en XML de la escena virtual a escenificar. La actual versión ha sido validada clínicamente, pudiendo utilizarse para planificaciones sencillas previas a la intervención, para la formación de residentes clínicos o bien como muestra del proceso quirúrgico a personas interesadas (familiares del paciente, estudiantes, etc) (13). Sin embargo, es una solución que no resuelve los problemas de una planificación preoperatoria para las osteotomías en cualquier región del cuerpo, sino solo para determinadas patologías del fémur.

#### **PreOPlan**

El éxito de una osteotomía pasa por una planificación preoperatoria perfecta. PreOPlan es una solución informática para la planificación preoperatoria. Ayuda al cirujano a analizar la deformidad, a simular la osteotomía y a seleccionar y colocar los implantes de un modo fácil e intuitivo (14).

Las plantillas fotográficas digitales tridimensionales de todas las placas disponibles pueden colocarse sobre la osteotomía simulada. La base de datos inteligente de PreOPlan permite seleccionar de forma intuitiva las especificaciones de las placas y los tornillos directamente sobre la plantilla digital de los implantes.

El programa guía al usuario por una sencilla y clara secuencia de planificación en cuatro pasos: importación de imágenes, análisis de la deformidad y simulación de la osteotomía, selección de los implantes e informe. Esto permite la agilización del proceso de planificación.

Gracias a su interfaz sencilla e intuitiva, la curva de aprendizaje de PreOPlan es muy breve. En cada paso de planificación, muestra únicamente las características pertinentes. Además, las funciones de ayuda interactiva facilitan al usuario el análisis y la planificación de las osteotomías (14). Sin embargo, es una aplicación propietaria de Siemens y Synthes, siendo esta última una empresa norteamericana, lo cual crea imposibilidad de comercio con Cuba.

### **OrthoView™**

OrthoView™ permite a los cirujanos ortopédicos crear planes prequirúrgicos detallados rápida y fácilmente a partir de imágenes de radiografías digitales. Facilita la planificación y aplicación de plantillas digitales para la artroplastia de articulaciones y revisiones, trauma, corrección de deformación de miembros, osteotomía y evaluación de columna. Es elegido por los hospitales en todo el mundo para completar su objetivo de radiología sin películas (15).

Cuenta con un módulo para la corrección de deformaciones de miembros que permite la planificación precisa de osteotomía en pantalla para un corte único o para múltiples cortes. Además brinda una gran variedad de herramientas para la planificación de todos los aspectos de la cirugía correctora del miembro.

El módulo posee tres capacidades principales:

- ✓ Evaluar la anatomía del paciente previamente a la cirugía: se evalúa la orientación de la articulación de la cadera, rodilla y tobillo desde un plano frontal y uno sagital.
- ✓ Planificar cortes de osteotomía: se dibujan una amplia variedad de ejes en base a las líneas media diafisaria y de las articulaciones. Se agregan cortes y se demuestra la animación del segmento del hueso para visualizar el resultado post quirúrgico.
- ✓ Personalizar el Marco espacial de Taylor: se mide (mediante el Asistente de OV) la posición del marco con respecto a la anatomía.

No obstante, es un sistema que tiene limitantes para su ejecución en versiones actuales del sistema operativo Windows. Además, solo permite realizar simulaciones de osteotomía en las extremidades.

### **TraumaCad**

TraumaCad es un sistema desarrollado por la compañía VoyantHealth. Permite a los cirujanos evaluar y manipular imágenes médicas, así como realizar varias planificaciones preoperatorias de cirugía y evaluación de imágenes (16).

Preciso y exacto, permite realizar mediciones, prótesis de revisión, simular osteotomías y visualizar la reducción de las fracturas. Posee herramientas de medición de procedimientos específicos y asistentes semiautomáticos para guiar los procesos de medición paso a paso.

Después de completar las medidas de alineación de la extremidad, pueden hacerse las simulaciones de osteotomías y la selección de implantes. Múltiples osteotomías pueden hacerse con el mismo método. Este admite entre otras cosas escoger el lugar exacto de la osteotomía, realizar cortes en la parte del hueso deseada y extraer los segmentos resultantes, así como colocar un eje de rotación del fragmento y manipularlo manualmente.

TraumaCad permite incrementar la productividad y brinda seguridad al paciente. La aplicación tiene una integración completa con el PACS y una extensa librería de plantillas digitales de grandes fabricantes que es actualizada regularmente. Brinda sencillas soluciones para diversas subespecialidades ortopédicas tales como: cadera, rodilla, pediatría, trauma, columna, pie, tobillo, extremidades superiores y cuenta con una suite 3D opcional (16).

Sin embargo, es un software muy costoso, perteneciente a Múnich y Telaviv, de Israel. Además, solo es posible ejecutarlo en el sistema operativo Windows XP.

#### *1.7.2. Sistemas nacionales*

Cuba actualmente no cuenta con un sistema informático que les permita realizar una planificación preoperatoria digital. La universidad de las ciencias informáticas se encuentra hoy desarrollando un software que brinde esta posibilidad.

El Planificador Quirúrgico Ortopédico (PQO) permite la apertura de imágenes en cualquier modalidad del formato DICOM, las cuales pueden estar almacenadas localmente o en un servidor de imágenes DICOM.

Para el procesamiento general de las mismas brinda una serie de herramientas, tales como: brillo, contraste, filtros, escala de grises, paleta de colores, transformar imagen, entre otras.

Permite la generación de reportes, en los cuales se almacenan datos concernientes al paciente y detalles de la intervención quirúrgica. Estos pueden ser impresos y exportados como PDF o HTML. Permite además la integración con un entorno inter-hospitalario a través del estándar DICOM 3.0 y la obtención de estudios imagenológicos desde servidores PACS.

Este sistema cuenta con diferentes módulos: rodilla, cadera, columna, pediatría, extremidades y trauma, cada uno de los cuales permite al especialista en ortopedia el uso de herramientas especializadas para realizar mediciones específicas en cada una de las áreas. Sin embargo, no cuenta con un módulo para la planificación quirúrgica ortopédica en las simulaciones de osteotomía. No obstante, a pesar de tener esta limitante es objeto de estudio en la presente investigación, debido a que conocer su estructura y diseño es fundamental y determinante para el resultado de la misma.

Como resultado del análisis de las soluciones existentes presentadas, se puede determinar el alto grado de acabado, prestación y el eficiente manejo de la información médica para mejorar las planificaciones preoperatorias. El estudio del TraumaCad fue de gran ayuda para determinar las principales funcionalidades utilizadas para realizar las simulaciones de osteotomía, como la selección de los fragmentos y la rotación y el movimiento de los mismos.

No obstante, estos sistemas son privativos debido a que requieren la adquisición de licencias para su uso y/o la autorización de sus propietarios. Al mismo tiempo, una vez adquiridos, no implica que se tenga acceso a su código fuente, por lo que se hace casi imposible incorporar nuevas funcionalidades al sistema sin contar con la empresa que la confecciona. Conjuntamente, poseen limitantes en cuanto a funcionalidades para desarrollar las simulaciones de osteotomía en determinadas regiones del cuerpo, así como su ejecución en versiones actuales del sistema operativo Windows. Además, requieren de una inversión de capital considerable para poner a punto la infraestructura tecnológica necesaria.

## **1.8. Tecnologías, patrones y lenguajes de programación a utilizar**

### *1.8.1. Plataforma de desarrollo .NET*

La plataforma .NET es un conjunto de tecnologías para desarrollar y manipular componentes que permitan crear aplicaciones Windows. Apuesta por un futuro en el que las aplicaciones se ejecutan de manera

distribuida en Internet. Así, una aplicación se ejecuta en un solo servidor y no existen múltiples copias de la misma. Además, una misma aplicación puede "adornarse" con distintas interfaces para que, desde diferentes dispositivos (teléfonos móviles, PDAs, portátiles, etc.) pueda accederse a la misma (17).

Esta plataforma define un flujo de trabajo sencillo que posibilita la rápida, eficiente y eficaz salida de productos de gran calidad y seguridad. Marca nuevos paradigmas para desarrollar aplicaciones de escritorio, para móviles, componentes reutilizables para SO y otras aplicaciones, brindando soluciones predefinidas a gran cantidad de problemas cotidianos que se encuentran en el desarrollo de software en la actualidad (17). Proporciona un entorno coherente de Programación Orientada a Objetos (POO por sus siglas en inglés), brindando niveles de extensibilidad y reusabilidad excelentes.

La plataforma .NET contiene dos componentes principales: el Entorno Común de Ejecución para Lenguajes (CLR por sus siglas en inglés) y la Biblioteca de Clases (BCL por sus siglas en inglés). Estos proporcionan un entorno de ejecución administrado, de implementación simplificada y la integración con una amplia variedad de lenguajes de programación (17).

Entre las aplicaciones que pueden desarrollarse haciendo uso de esta plataforma se tienen:

- ✓ Aplicaciones de línea de comandos.
- ✓ Servicios de Windows.
- ✓ Aplicaciones de escritorio con Windows Forms o WPF.
- ✓ Aplicaciones web con el framework ASP.NET, o Silverlight.
- ✓ Aplicaciones para dispositivos móviles con Windows Mobile.

### 1.8.2. *Windows Presentation Foundation*

Windows Presentation Foundation (WPF por sus siglas en inglés), es un subsistema de Windows (librerías integradas en el sistema operativo), orientado a unificar los mecanismos de creación y gestión de interfaces de usuario. Permite crear una amplia gama de aplicaciones independientes y hospedadas en el navegador. Presenta aspectos interesantes como: la separación de apariencia y lógica, soporte del patrón "orden" (command), fácil conexión a fuentes de datos vía ligaduras (bindings), simplificación de trabajo con objetos observables mediante propiedades de dependencia, herencia de valores para propiedades por relación

jerárquica entre componentes, acceso directo a hardware gráfico, animaciones, personalización completa de componentes mediante plantillas, entre otros (18).

Está compuesto por un motor que saca ventaja del hardware con gráficos modernos y por un conjunto de clases administradas que los desarrolladores tienen la posibilidad de usar para crear aplicaciones ricas visualmente. Incluye además gráficos, animaciones y multimedia de gran calidad; además es compatible con el desarrollo de aplicaciones de uso internacional. Proporciona herramientas que permiten la optimización del rendimiento, la conformidad con la especificación XPS (XML Paper Specification), las pruebas de la interfaz de usuario y la edición de XAML (Extensible Application Markup Language) (19).

### 1.8.3. *Managed Extensibility Framework*

Managed Extensibility Framework (MEF por sus siglas en inglés), es biblioteca para crear aplicaciones ligeras y extensibles. Permite a los desarrolladores de aplicaciones detectar y utilizar extensiones sin requisitos de configuración. También permite a los desarrolladores de extensiones encapsular código con facilidad y evitar dependencias lógicas frágiles. MEF no solo permite reutilizar extensiones dentro de las aplicaciones sino también entre ellas (20).

MEF es una parte integral de .NET Framework 4 y está disponible siempre que se use .NET Framework. Puede utilizar MEF en aplicaciones cliente que usen WPF (20).

Los elementos MEF especifican sus capacidades mediante declaración y son reconocibles en tiempo de ejecución, lo que significa que una aplicación puede utilizar elementos sin referencias incluidas en el código o archivos de configuración frágiles. MEF permite a las aplicaciones detectar y examinar elementos por sus metadatos, sin crear instancias ni cargar sus ensamblados. Por consiguiente, no hay necesidad de especificar meticulosamente qué extensiones se deben cargar y cuándo (20).

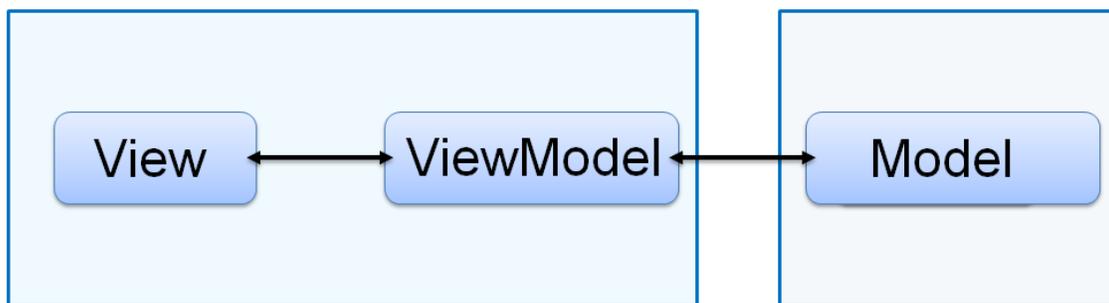
### 1.8.4. *Patrón Model View ViewModel*

Model View ViewModel (MVVM por sus siglas en inglés), es un patrón arquitectónico basado en el patrón Modelo Vista Controlador (MVC). Su enfoque es hacia plataformas de desarrollo de interfaz de usuario como Windows Presentation Foundation.

El patrón MVVM organiza el código de tal forma, que es posible cambiar partes individuales sin que los cambios afecten a las demás partes. Esto presenta numerosas ventajas, tales como:

- ✓ Permite un estilo de codificación exploratorio e iterativo.
- ✓ Simplifica las pruebas unitarias.
- ✓ Permite aprovechar mejor herramientas de diseño como Expression Blend.
- ✓ Admite la colaboración en equipo.

Tiene como principal objetivo separar el modelo y la vista (interfaces de usuario), introduciendo una capa entre ellos que sirve de enlace, llamada modelo de vista (ViewModel).



*Figura 1. Esquema general del patrón Model-View-ViewModel.*

La principal ventaja de una arquitectura desacoplada (separación por capas) es que permite aislar el impacto de los cambios. Esto implica que sea mucho menos arriesgado probar nuevas características, corregir errores e incorporar las contribuciones de los colaboradores.

### 1.8.5. C# 4.0

C# 4.0 es el lenguaje de programación en que se desarrolla el sistema. Fue diseñado para crear una amplia gama de aplicaciones que se ejecutan en la plataforma .NET. Es simple, eficaz, con seguridad de tipos y orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto a otros lenguajes como C++, es que no admite funciones ni variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código (21).

Otras de sus principales características es la portabilidad del código, consta de un editor de código completo, plantillas de proyecto, asistentes para código, así como un depurador muy eficiente y fácil de usar. Además maneja mecanismos que aseguran que los accesos a tipos de datos siempre se realicen

correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto (22).

El uso de las tecnologías y herramientas descritas está condicionado por la arquitectura del módulo que se desarrollará. La plataforma de desarrollo .Net posee una biblioteca de clases que brinda un entorno de implementación simplificada e integración con múltiples lenguajes de programación, dentro de los cuales se seleccionó C# 4.0 por su portabilidad de código. Además, se escoge la tecnología WPF para separar la lógica de la apariencia y desarrollar una aplicación rica visualmente. MEF es usada como herramienta para apoyar la arquitectura basada en extensiones y el patrón MVVM, es utilizado para lograr un desacoplamiento entre las capas, apoyando el principio de la arquitectura orientada a dominio.

### **1.9. Modelo de calidad, lenguaje y notación de modelado**

#### *1.9.1. Modelo de Capacidad y Madurez*

El Modelo de Capacidad y Madurez (CMMI por sus siglas en inglés) es un modelo creado por el Instituto de Ingeniería del Software (SEI por sus siglas en inglés), destinado a evaluar y mejorar los procesos de desarrollo, implementación y mantenimiento del software (23).

Representa un camino de mejoramiento, y permite determinar la madurez y evaluar las capacidades de las organizaciones que desarrollan software. Es una colección estructurada de elementos, que describe características de procesos que han demostrado, por experiencia, ser exitosos. Es recomendado para organizaciones que quieren incrementar la capacidad de su proceso de desarrollo y desarrollar software con calidad (24).

CMMI soporta dos tipos de representaciones que son totalmente equivalentes, la continua y la escalonada. La primera se centra en la mejora de un proceso o un conjunto de ellos, relacionados a un área de proceso en que una organización desea mejorar. Esta representación utiliza unos niveles de capacidad para caracterizar la mejora concerniente a un área de proceso individual (25).

La representación escalonada ofrece un método sistemático y estructurado de mejoramiento de procesos que implica mejorar por etapas o niveles. Una organización que alcanza un nivel determinado, asegura que cuenta con una infraestructura robusta de procesos para optar por alcanzar el nivel siguiente, pudiendo ser certificada bajo un nivel de madurez (25).

El centro CESIM, como resultado de un proceso de mejoras que se llevó a cabo en la UCI, certificó el nivel 2 correspondiente a la representación escalonada de CMMI, la cual define 5 niveles de madurez. Una organización CMMI Nivel 2 asegura que sus procesos son planeados, documentados, realizados, monitoreados y controlados a nivel de proyectos. Los requerimientos, estándares y objetivos para los procesos, sus productos de trabajo y sus servicios son definidos y documentados (26).

Este nivel está compuesto por 7 áreas de proceso: Gestión de Requisitos (REQM), Planificación de Proyectos (PP), Seguimiento y Control de Proyectos (PMC), Acuerdos con Proveedores (SAM), Medición y Análisis (MA), Aseguramiento de la Calidad de Procesos y Productos (PPQA) y Gestión de la Configuración (CM).

Se especificaron durante el proceso de mejoras en la UCI los libros: Libro de Proceso para la Administración de Requisitos (IPP-3510\_2009) donde se detallan las actividades, los roles y los artefactos para cumplir con las prácticas y subprácticas de REQM y el libro Ciclo de Vida de Proyectos que especifica las fases y las disciplinas por las que deben transitar todos los proyectos. Ambos materiales tuvieron como base lo que define CMMI en su nivel 2 para el área de REQM y el criterio de expertos.

### *1.9.2. Lenguaje Unificado de Modelado*

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es utilizado como lenguaje gráfico para visualizar, especificar, construir y documentar cada uno de los artefactos generados durante el ciclo de vida del software. Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, compuestos reciclados y esquemas de bases de datos (27).

UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente ayuda a visualizar el diseño y a hacerlo más accesible para otros. Está diseñado para su uso con software orientado a objetos y tiene un uso limitado en otro tipo de cuestiones de programación. Es controlado por el grupo de administración de objetos (OMG) y constituye el estándar de descripción de esquemas de software (28).

### *1.9.3. Proceso Unificado de Rational*

El Proceso Unificado de Rational (RUP, por sus siglas en inglés) es la metodología utilizada para la creación de los artefactos que se obtienen durante el desarrollo de las disciplinas: "Modelado del Negocio",

“Requisitos”, “Análisis y Diseño” e “Implementación. Utiliza UML como lenguaje de modelado para representar todos los esquemas y diagramas del sistema. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y ser guiado por los casos de uso (29).

### **1.10. Herramientas usadas para la solución propuesta**

#### *1.10.1. Enterprise Architect 7.5*

Para realizar el modelado es definida por el departamento SWMI la herramienta Enterprise Architect (EA por sus siglas en inglés) en su versión 7.5. Es una herramienta que abarca integralmente el ciclo de vida, cubriendo el desarrollo de software desde el levantamiento de los requerimientos, a través de las etapas de análisis, modelos de diseño, prueba y mantenimiento. Está diseñada para ayudar a construir sistemas robustos y fáciles de mantener (30).

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C#, que es el lenguaje de programación utilizado para desarrollar la solución propuesta. Además, es la herramienta empleada para modelar los artefactos que se generan durante el desarrollo del software.

#### *1.10.2. Microsoft Visual Studio 2010*

Como entorno de desarrollo integrado (IDE por sus siglas en inglés) se ha utilizado Microsoft Visual Studio. Este permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. Además, brinda funcionalidades para la implementación y administración de cambios que garantizan que la base de datos y la aplicación estén siempre sincronizadas (31). Al mismo tiempo soporta el uso del lenguaje de programación C # y de tecnologías que facilitan el desacoplamiento entre capas como WPF.

#### *1.10.3. TortoiseSVN 1.6.6*

Como cliente de Subversion se ha designado TortoiseSVN en su versión 1.6.6. Es un cliente gratuito de código abierto para el sistema de control de versiones Apache™ Subversion®. TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se

hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio (32).

En este capítulo se asienta la base teórica de la investigación a partir de la fundamentación de los principales aspectos que la componen. El estudio de los sistemas existentes en el mundo para la planificación quirúrgica ortopédica y la realización de simulaciones de osteotomías, demostró que estos son sistemas privativos y altamente costosos, a los cuales Cuba no tiene acceso, además de presentar limitantes de compatibilidad con las versiones actuales del sistema operativo Windows, de ahí la necesidad de desarrollar uno propio. Finalmente se describen las tecnologías, patrones, herramientas y lenguajes a utilizar para el modelado, diseño e implementación de la solución propuesta, así como la definición del modelo de calidad y la notación de modelado, facilitando el uso de los mismos durante el trabajo.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se expone la propuesta del sistema a desarrollar. Se presenta el modelo de dominio detallándose los principales elementos conceptuales que lo componen. Además se definen los requisitos funcionales y no funcionales, así como los casos de uso del sistema identificados, mostrándose una breve descripción de cada uno de ellos.

2.1. Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales o entidades del mundo real. Esto ayuda a los usuarios, clientes y desarrolladores a utilizar un lenguaje común para poder entender el contexto en que se desarrolla el sistema. Captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno en el que trabaja el sistema y no incluyen las responsabilidades de las personas que ejecutan las actividades (33).

A continuación se presenta el modelo de dominio realizado, donde se evidencian los principales conceptos identificados en la solución del problema planteado. Además se hace una descripción textual de estos conceptos y sus relaciones.

2.1.1. Modelo de dominio

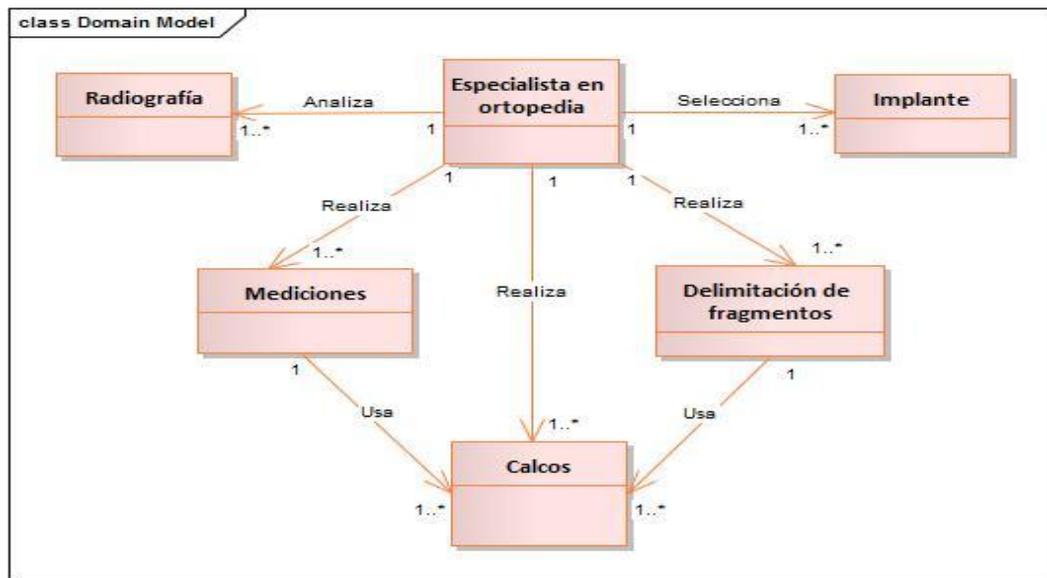


Figura 2. Modelo de dominio.

### 2.1.2. Descripción de las clases conceptuales del dominio

**Especialista en ortopedia:** Persona con los conocimientos necesarios en el área, encargada de realizar la planificación preoperatoria de las osteotomías.

**Radiografía:** Imagen médica registrada en una placa o película fotográfica, destinada al diagnóstico de las zonas óseas del organismo.

**Implante:** Dispositivo médico fabricado para reemplazar una estructura biológica que falta, suplantar una dañada o mejorar una existente.

**Calcos:** Copias de radiografías obtenidas a través de trazos o escrituras haciendo uso de papeles semitransparentes y marcadores por medio de un negatoscopio.

**Mediciones:** Conjunto de operaciones (cálculos, trazos, medidas) realizadas sobre los calcos, encaminadas a corregir las afectaciones óseas del paciente.

**Delimitación de fragmentos:** Permite al especialista delimitar la zona que sufrirá cambios en su posición, una vez realizada las mediciones y los calcos correspondientes.

## 2.2. Sistema propuesto

A partir del estudio realizado sobre los disímiles trastornos y enfermedades que afectan a los huesos, los diferentes procedimientos quirúrgicos y la descripción del proceso de planificación quirúrgica llevado a cabo en los hospitales ortopédicos del país, se propone desarrollar el módulo de simulaciones de osteotomía para el sistema PQO.

El módulo brindará diversas funcionalidades comunes entre las distintas técnicas utilizadas para la realización de las osteotomías. Permitirá aplicar la herramienta lazo en aras de seleccionar el fragmento con el que se desea maniobrar y además, posibilitará limpiar la imagen en caso de que se quiera repetir el proceso para buscar mejoras en los detalles del corte. Posteriormente ofrecerá la opción de mover, rotar y/o eliminar dicho fragmento en dependencia de la técnica osteotómica empleada. Por último, garantizará dicomizar la imagen resultante, posibilitando que los cambios realizados sobre la imagen puedan ser guardados. Otros módulos serán los encargados de realizar las mediciones especializadas y/o agregar los implantes dependiendo de la táctica quirúrgica empleada.

La digitalización del proceso de planificación preoperatoria para las osteotomías mediante este sistema, contribuirá a que las cirugías efectuadas en las distintas áreas anatómicas sean realizadas en el menor tiempo posible. El especialista tendrá una visión más clara de las condiciones en las que se realizará la operación, aumentando la efectividad de los procedimientos quirúrgicos y garantizando mejores resultados. La aplicación podrá utilizarse además para la formación de estudiantes y para el entrenamiento de los especialistas. Todo esto se traducirá en un aumento de los indicadores de eficiencia en los servicios a los pacientes.

La integración del módulo con el PGO se realizará a través de un complemento (plugin) que hará uso de la interfaz del framework .Net IServiceProvider. Además, se utilizará la arquitectura basada en extensiones para permitir la posterior ampliación de la aplicación sin necesidad de redistribuir un nuevo ejecutable, aportando extensibilidad e independencia entre los módulos.

### 2.3. Especificación de los requisitos de software

#### 2.3.1. Requerimientos funcionales

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas (34).

La Tabla 1 muestra los requisitos funcionales del módulo de simulaciones de osteotomía para el sistema PGO.

Requerimiento	Descripción
RF 1.1 Aplicar herramienta lazo.	Permite seleccionar un fragmento de la imagen de forma variable con el cursor y el mismo queda cortado listo para mover.
RF 1.2 Mover fragmento seleccionado.	Luego de cortado un fragmento de la imagen, esta funcionalidad permite mover el mismo.
RF 1.3 Rotar libremente fragmento.	Luego de cortado un fragmento de la imagen, esta funcionalidad permite rotar el fragmento teniendo como centro de rotación un punto de anclaje.
RF 1.4 Cambiar posición de	Permite cambiar la posición del punto de anclaje que es el centro de

punto de anclaje de rotación.	rotación del fragmento.
RF 1.5 Restaurar rotación de fragmento.	Permite restaurar el fragmento a su forma original, luego de haber rotado el mismo.
RF 1.6 Eliminar fragmento.	Permite eliminar el fragmento seleccionado o todos los fragmentos.
RF 1.7 Dicomizar.	Permite crear un archivo DICOM a partir de distintos tipos de imágenes, ya sea JPG o Bitmap.
RF 1.8 Limpiar imagen.	Permite restaurar la imagen a su forma original, manteniendo los fragmentos extraídos como componentes separados de la imagen.

Tabla 1. Requisitos funcionales del módulo de simulaciones de osteotomía para el sistema de PQO.

En la Figura 3 se muestra el diagrama de requisitos funcionales del módulo de simulaciones de osteotomía para el sistema PQO.

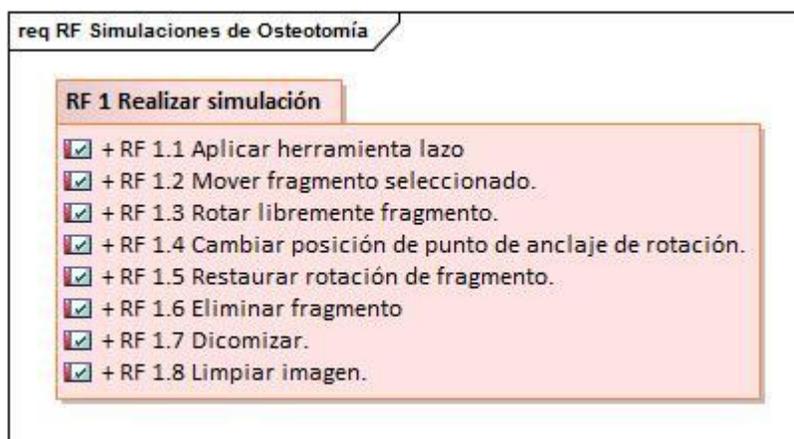


Figura 3. Diagrama de requisitos funcionales del módulo de simulaciones de osteotomía para el sistema PQO.

### 2.3.2. Requerimientos no funcionales

Los requerimientos no funcionales definen cualidades o propiedades que el producto debe tener. Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento

(en tiempo y espacio), fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares.

Los requisitos no funcionales fueron nombrados asignándoles un prefijo, de acuerdo a las Pautas para la Fase de Requerimientos del Departamento SWMI. En el Anexo 2 se muestra la tabla de los prefijos según la categoría del requisito.

La Tabla 2 muestra los requisitos no funcionales del módulo de simulaciones de osteotomía para el sistema PQO.

Requerimiento	Descripción
RNU 1. Facilidad de empleo para usuarios sin experiencia.	El sistema debe tener una interfaz de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse con él.
RNU 2. Todas las operaciones se deben realizar sobre la grilla activa.	Todas las operaciones se deben realizar sobre la grilla que se encuentre activa en ese momento.
RNU 3. Cuando se seleccione la opción de grillas debe mantenerse activa la opción anterior.	Cuando se seleccione la opción de grillas se debe mantener activa la opción que estaba seleccionada anteriormente.
RNU 4. Doble clic sobre la imagen para visualizarla en pantalla completa y viceversa.	Se debe poder visualizar la imagen en pantalla completa al dar doble clic sobre la misma y viceversa.
RNU 5. Tanto el idioma como el estilo de las líneas y los puntos deben ser configurables.	El sistema debe permitir la configuración del idioma y de los estilos de las líneas y los puntos.
RNU 6. Se utilizará el atajo de teclas Shift + Clic para el marcado múltiple de fragmentos.	Para marcar varios fragmentos se usará el atajo de teclas Shift + Clic.
RNU 7. Se utilizará la tecla Del para eliminar uno o varios fragmentos	Para eliminar uno o varios fragmentos seleccionados se usará la tecla Del.

seleccionados.	
RNDI 1. Uso de Framework.Net 4.0.	Se especifica el uso de Microsoft Framework.Net 4.0 que ofrece mejoras en cuanto a administración y rendimiento. El lenguaje de programación C# depende de este Framework.
RNDI 2. C# como lenguaje de programación.	Se deberá utilizar C# como lenguaje de programación pues está diseñado y optimizado para la plataforma .NET.
RNDI 3. Entorno Integrado de Desarrollo Microsoft Visual Studio 2010.	Se utilizará Microsoft Visual Studio 2010 como entorno integrado de desarrollo. Este IDE utiliza como marco de trabajo el Framework.Net 4.0 y soporta C# como lenguaje de programación.
RNDI 4. TortoiseSVN versión 1.6.6 como herramienta para el control de versiones.	Se utilizará la herramienta TortoiseSVN para el control de versiones, la cual puede integrarse al entorno integrado de desarrollo seleccionado.
RNDI 5. Librerías a utilizar.	Se utiliza la librería CALIB para el procesamiento de imágenes y la Fluent para los controles visuales.
RNDI 6. Estándar de codificación.	Se utilizará el estándar con notación CamelCase (atributos), PascalCase (métodos) y Notación Húngara (interfaces).
RNDI 7. Uso de Windows Presentation Foundation para el diseño de la interfaz de usuario.	La interfaz de usuario será diseñada utilizando la tecnología Windows Presentation Foundation (WPF).
RNDI 8. Uso de Enterprise Architect como herramienta CASE.	Se utilizará como herramienta CASE Enterprise Architect en su versión 7.5.

RNDI 9. Uso de UML como lenguaje de modelado.	Se utilizará como lenguaje de modelado UML en su versión 2.1.
RNDI 10. Patrones a utilizar para el diseño del sistema.	Los patrones a utilizar para el diseño del sistemas son los siguientes: Command y Singleton.
RNF 1. Exactitud en las salidas del sistema.	El sistema debe brindar salidas precisas.
RNFO 1. Memoria RAM de 512 MB.	Para el correcto funcionamiento del sistema, se necesita un CPU Dual Pentium IV 3.0GHz y 512 Mb de memoria RAM como mínimo y se recomienda una tarjeta de red Gigabit Ethernet NIC.
RNFO 2. CPU Dual Pentium IV 3.0GHz.	
RNFO 3. Gigabit Ethernet NIC.	
RNFO 4. Sistema operativo Windows XP Service Pack 3 o superior.	El software debe instalarse sobre el sistema operativo Windows XP Service Pack 3 o superior.
RNFO 5. Capacidad de disco duro 20 GB como mínimo.	La capacidad del disco duro debe ser como mínimo de 20 GB.
RNSO 1. Ayuda y documentación.	Se debe brindar un manual de usuario y documentación del sistema para lograr un correcto entendimiento de las funcionalidades del mismo y una mayor facilidad y experiencia de uso.
RNIU 1. Fácil acceso a herramientas más usadas.	El sistema debe mostrar las herramientas más usadas en un área de fácil acceso al usuario y visible para el mismo.
RNIU 2. Uso de Ribbon como contenedor de herramientas.	Se utilizará el Ribbon de Microsoft Office como contenedor de herramientas.

RNI 1. Conexión al servidor de imágenes mediante el protocolo TCP/IP y por el puerto 104.	La conexión al servidor de imágenes debe realizarse a través del protocolo TCP/IP y por el puerto 104.
---	--

Tabla 2. Requisitos no funcionales del sistema PQO.

### 2.4. Definición de los actores del sistema

Luego de describir los requisitos funcionales y no funcionales del módulo de simulaciones de osteotomía para el sistema PQO, se define el actor que interactúa con los casos de uso del sistema, el mismo se muestra en la tabla siguiente.

Actor	Justificación
 <p>The diagram shows a use case box labeled 'uc Actores' containing a stick figure actor labeled 'Especialista'.</p>	Realizar las operaciones iniciales del proceso de simulaciones de osteotomía.

Tabla 3. Actor del sistema.

### 2.5. Diagrama de casos de uso del sistema

En la Figura 4 se muestra el diagrama de casos de uso del módulo de simulaciones de osteotomía para el sistema PQO.

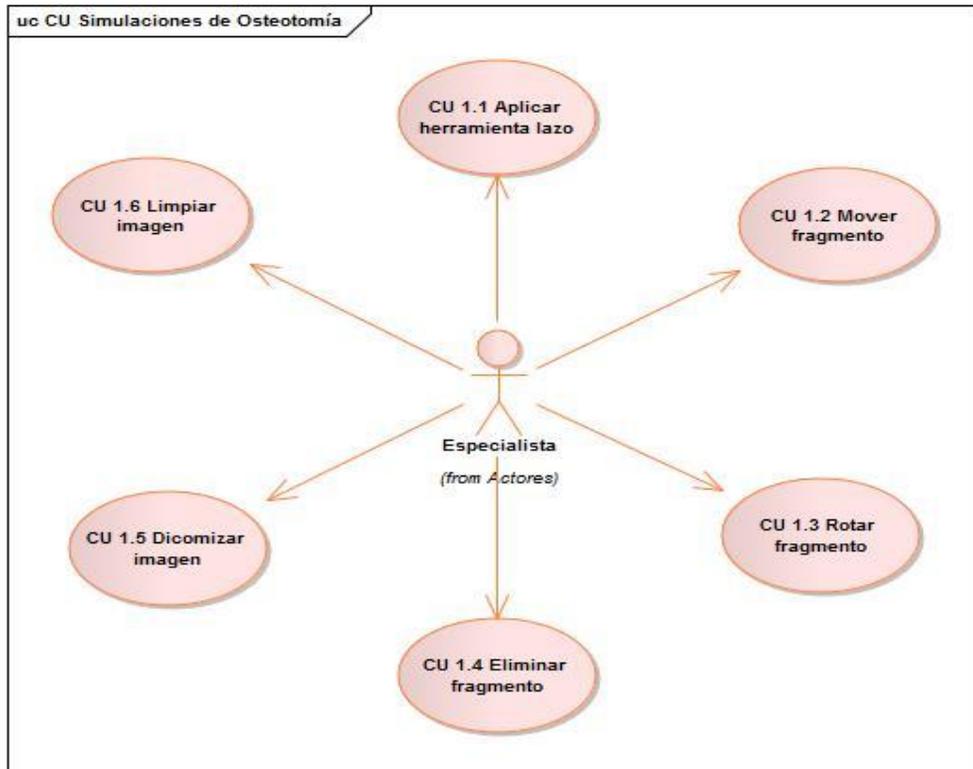


Figura 4. Diagrama de casos de uso del sistema del módulo de simulaciones de osteotomía para el sistema PQQ.

### 2.6. Descripción de los casos de uso del sistema

Las tablas que se muestran a continuación contienen un resumen de los casos de uso del sistema. En el Anexo 3 se muestran las descripciones textuales expandidas de los mismos.

- Caso de uso del sistema - Aplicar herramienta lazo

Objetivo	Seleccionar un fragmento de la imagen, quedando cortado y listo para mover.
Actores	Especialista (inicia).
Resumen	El especialista selecciona un fragmento de la imagen de forma variable con el cursor, quedando cortado y listo para mover.
Complejidad	Media.

Prioridad	Crítico.
Referencias	RF 1.1

*Tabla 4. Resumen del CU Aplicar herramienta lazo.*

- Caso de uso del sistema – Mover fragmento

Objetivo	Mover el fragmento cortado con la herramienta lazo.
Actores	Especialista (Inicia).
Resumen	El médico selecciona la herramienta Mover de la sección de simulaciones y mueve el fragmento cortado por el área de trabajo.
Complejidad	Media.
Prioridad	Secundario.
Referencias	RF 1.2

*Tabla 5. Resumen del CU Mover fragmento.*

- Caso de uso del sistema – Rotar fragmento

Objetivo	Rotar el fragmento en la dirección deseada, teniendo como centro de rotación un punto de anclaje.
Actores	Especialista (Inicia).
Resumen	El especialista selecciona la herramienta Rotar y rota el fragmento en la dirección deseada.
Complejidad	Media.
Prioridad	Secundario.
Referencias	RF 1.3, RF 1.4, RF 1.5

*Tabla 6. Resumen del CU Rotar fragmento.*

- Caso de uso del sistema – Eliminar fragmento

Objetivo	Eliminar el fragmento seleccionado.
----------	-------------------------------------

Actores	Especialista (Inicia).
Resumen	El especialista selecciona la opción Eliminar o Eliminar todas, para eliminar un fragmento o todos los fragmentos seleccionados.
Complejidad	Media.
Prioridad	Secundario.
Referencias	RF 1.6

*Tabla 7. Resumen del CU Eliminar fragmento.*

- Caso de uso del sistema – Dicomizar imagen

Objetivo	Dicomizar una imagen o un conjunto de ellas.
Actores	Especialista (Inicia).
Resumen	El especialista realiza una simulación y desea dicomizar la imagen con los cambios realizados.
Complejidad	Alta.
Prioridad	Secundario.
Referencias	RF 1.7

*Tabla 8. Resumen del CU Dicomizar imagen.*

- Caso de uso del sistema – Limpiar imagen

Objetivo	Permite restaurar la imagen luego de haberse realizado algún corte de fragmento.
Actores	Especialista (Inicia).
Resumen	El especialista selecciona la opción “Limpiar imagen” para restaurar la imagen a su forma original. Cuando se limpia la imagen, el fragmento que ha sido construido se mantiene como un componente separado de la imagen.
Complejidad	Alta.
Prioridad	Secundario.

Referencias	RF 1.8
-------------	--------

*Tabla 9. Resumen del CU Limpiar imagen.*

### 2.7. Propuesta de casos de uso por ciclos de desarrollo

Todos los casos de uso del módulo permiten brindar servicios significativos, por lo que para el desarrollo del sistema, sin tener en cuenta un orden de prioridad, se proponen los ciclos de desarrollo siguientes:

Nombre del caso de uso	Justificación
Aplicar herramienta lazo	Estos casos de usos son primarios, pues son los más importantes para el módulo.
Mover fragmento	
Rotar fragmento	

*Tabla 10. Casos de uso propuestos para el primer ciclo de desarrollo.*

Nombre del caso de uso	Justificación
Eliminar fragmento	Estas funcionalidades son significativas para el módulo.
Dicomizar imagen	
Limpiar imagen	

*Tabla 11. Casos de uso propuestos para el segundo ciclo de desarrollo.*

En este capítulo se establecieron las funcionalidades que debe tener el sistema a partir de la especificación de los requisitos funcionales. La definición de los requerimientos no funcionales brindó un mejor enfoque de las características físicas y técnicas que debe cumplir el módulo de simulación para su correcto funcionamiento. La descripción de los principales conceptos y sus relaciones en el modelo de dominio, permitió un mayor entendimiento del contexto en que se desarrolla el módulo. Se identificaron además los casos de uso del sistema y su distribución por ciclos de desarrollo, logrando así una mejor organización de la implementación.

### CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se define la arquitectura del sistema. Se presentan los diagramas de clases de diseño y de secuencia que participan en la realización de los casos de uso. Además, se describen las clases relacionadas con los casos de uso arquitectónicamente significativos.

#### 3.1. Diseño

La fase de diseño es técnicamente la parte central de la ingeniería del software durante la que se desarrollan, revisan y documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño es la única forma de traducir con precisión los requisitos del cliente en un producto o sistema acabado. Proporciona una imagen completa del software, desde una perspectiva de implementación, de forma que el sistema cumpla con parámetros como extensibilidad, reusabilidad, compatibilidad, portabilidad y robustez, lo que permite que todo el software alcance un alto nivel de calidad. Es la base de todas las partes posteriores del desarrollo y de la fase de prueba (35).

El diseño debe estar bien especificado para que el sistema pueda ser implementado sin imprecisiones, y su principal objetivo es la elaboración de los diagramas de clases de diseño, que muestran las clases del sistema, sus atributos y las relaciones entre ellos. Los diagramas de clases del diseño de los casos de usos arquitectónicamente significativos se muestran en el Anexo 4.

Para describir gráficamente un caso de uso o escenario son utilizados los diagramas de secuencia, mediante los cuales se detalla el flujo de acciones entre objetos, haciendo uso de mensajes. En el Anexo 5 se muestran los diagramas de secuencia vinculados a los casos de uso arquitectónicamente significativos.

##### 3.1.1. Patrones de diseño

Un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico y en medio de “fuerzas” que pueden tener un impacto en la manera en que se aplica y utiliza el patrón (36). Con el conocimiento de estos patrones, los programadores son capaces de identificar las situaciones en las que éstos tienen aplicación y utilizarlos sin tener que detenerse para analizar el problema y vislumbrar diferentes estrategias de resolución. La aplicación de los patrones de

diseño permite ahorrar tiempo y mejorar el sistema, haciéndolo más eficiente, dinámico y seguro. Los patrones utilizados en el diseño del módulo de simulación son los siguientes:

**Singleton:** Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a esta instancia. De esta forma, todos los objetos que hagan uso de esta clase utilizarán la misma instancia. Es clasificado como un patrón creacional.

**Command:** También llamado Action o Transaction, tiene el objetivo de aportar una interfaz abstracta de gestión de operaciones sobre un cierto receptor, permitiendo la ejecución de operaciones sin tener que conocer exactamente su tipo ni quien la implementa. Es clasificado como un patrón de comportamiento a nivel de objeto.

### 3.2. Descripción de las clases

Una clase es la representación de un conjunto de objetos de la vida real con características y comportamientos similares. Agrupa propiedades (atributos) y métodos (funciones) que le permiten la interacción consigo misma o con el mundo exterior. Constituye el componente más importante en el paradigma de la programación orientada a objetos (37).

La descripción de las clases ofrece información sobre los atributos y métodos que en ella se encuentran, así como una mejor comprensión de su estructura. En el Anexo 6 aparecen descritas las clases involucradas en la realización de los casos de uso arquitectónicamente significativos.

### 3.3. Modelo arquitectónico

La arquitectura es un nivel de diseño que consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software, cumpliendo los objetivos del sistema. Tiene como misión principal la identificación de requerimientos críticos en la estructura del sistema y la reducción de riesgos asociados a la construcción del mismo. Debe orientarse a las necesidades del cliente y ser flexible en cuanto a futuras modificaciones, tanto de software como de hardware, evitando que efectos inesperados puedan tener repercusión en el sistema (38).

Los requerimientos pertenecientes al módulo de simulaciones de osteotomía serán incluidos al sistema PQO haciendo uso de una arquitectura híbrida, capaz de integrar elementos de diferentes arquitecturas. La arquitectura basada en extensiones permite la posterior ampliación de la aplicación sin necesidad de

redistribuir un nuevo ejecutable. Además brinda facilidades en cuanto a la adaptación a cambios en los requisitos, añadiendo o modificando una funcionalidad mediante la creación o modificación de la extensión adecuada. Microsoft Extensibility Framework es la herramienta utilizada para trabajo con esta arquitectura, aportando extensibilidad automática y ligereza al sistema (39).

Al ser el PQO una aplicación compleja, el comportamiento de las reglas de negocio (lógica del Dominio) está sujeto a muchos cambios y es muy importante poder modificar, construir y realizar pruebas sobre dichas capas de lógica del dominio de una forma fácil e independiente. Debido a esto, un objetivo importante es tener el mínimo acoplamiento entre el Modelo del Dominio (lógica y reglas de negocio) y el resto de capas del sistema (Capas de presentación, Capas de Infraestructura, persistencia de datos, entre otras.). Para esto es utilizada la arquitectura orientada a dominio (DDD por sus siglas en inglés), que permite aislar tanto como sea posible el código del Modelo del Dominio, disminuyendo así las posibilidades de tener que realizar cambios en otras áreas de la aplicación.

### 3.3.1. Patrones arquitectónicos

Se utilizan además patrones arquitectónicos con el fin de habilitar el desacoplamiento entre capas, garantizando robustez, adaptabilidad, estabilidad, facilidad de mantenimiento, flexibilidad y extensibilidad. El patrón de Inversión de Control (IoC por sus siglas en inglés) describe técnicas para soportar una arquitectura tipo “plug-in”, donde los objetos pueden buscar instancias de otros objetos que requieren y de los cuales dependen. El patrón de Inyección de Dependencias (Dependency Injection, DI, por sus siglas en inglés) es un caso especial de IoC en el que se suplen objetos/dependencias a una clase, en lugar de ser la propia clase quien cree los objetos/dependencias que necesita (40).

El patrón Model View ViewModel permite presentar y gestionar de manera más sencilla los datos que son trasladados a la vista dentro de la aplicación. El concepto fundamental de este patrón es separar el Modelo (Model) y la Vista (View), introduciendo una capa abstracta de enlace entre ellos denominada modelo de vista (ViewModel), que expone los datos desde el modelo y gestiona en el sistema la lógica de visualización de la vista (41).

Durante este capítulo se sentaron las bases para el inicio de la implementación del sistema. La descripción de las clases más significativas brindó una mayor comprensión de sus atributos y métodos. Los diagramas de clases del diseño mostraron las relaciones entre las distintas clases, logrando así una estructura clara de la aplicación. Los diagramas de secuencia expusieron el flujo de eventos, logrando de esta manera describir

la interacción entre el usuario y el componente que se desea desarrollar. Además, se dieron a conocer los detalles de la arquitectura empleada, permitiendo de esta manera presentar un modelo arquitectónico bien definido.

### CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo se describen los temas asociados al flujo de trabajo de implementación. Se detallan los componentes ejecutables del sistema, así como la forma en la que interactúan para posibilitar su correcto funcionamiento. Se realiza una especificación sobre cómo van a estar desplegadas físicamente sus distintas partes y mediante qué protocolos se comunicarán. Se describen brevemente los estándares de codificación utilizados.

#### 4.1. Diagrama de componentes

Un componente es la parte física, desplegable y reemplazable de un sistema. Básicamente encapsula implementación, interfaces y proporciona la realización de los mismos. Típicamente contiene clases y puede ser implementado por uno o más artefactos como ficheros ejecutables, binarios, entre otros. En un componente se puede implementar uno o varios elementos, ejemplo varias clases, sin embargo la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando (42).

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Permiten visualizar el modelado de un sistema o subsistema además de especificar un componente con interfaces bien definidas. Para la realización de un diagrama de componentes se tienen en cuenta los requisitos asociados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

En la figura 5 se muestra el diagrama de componentes del módulo de simulaciones de osteotomía para el sistema PQO.

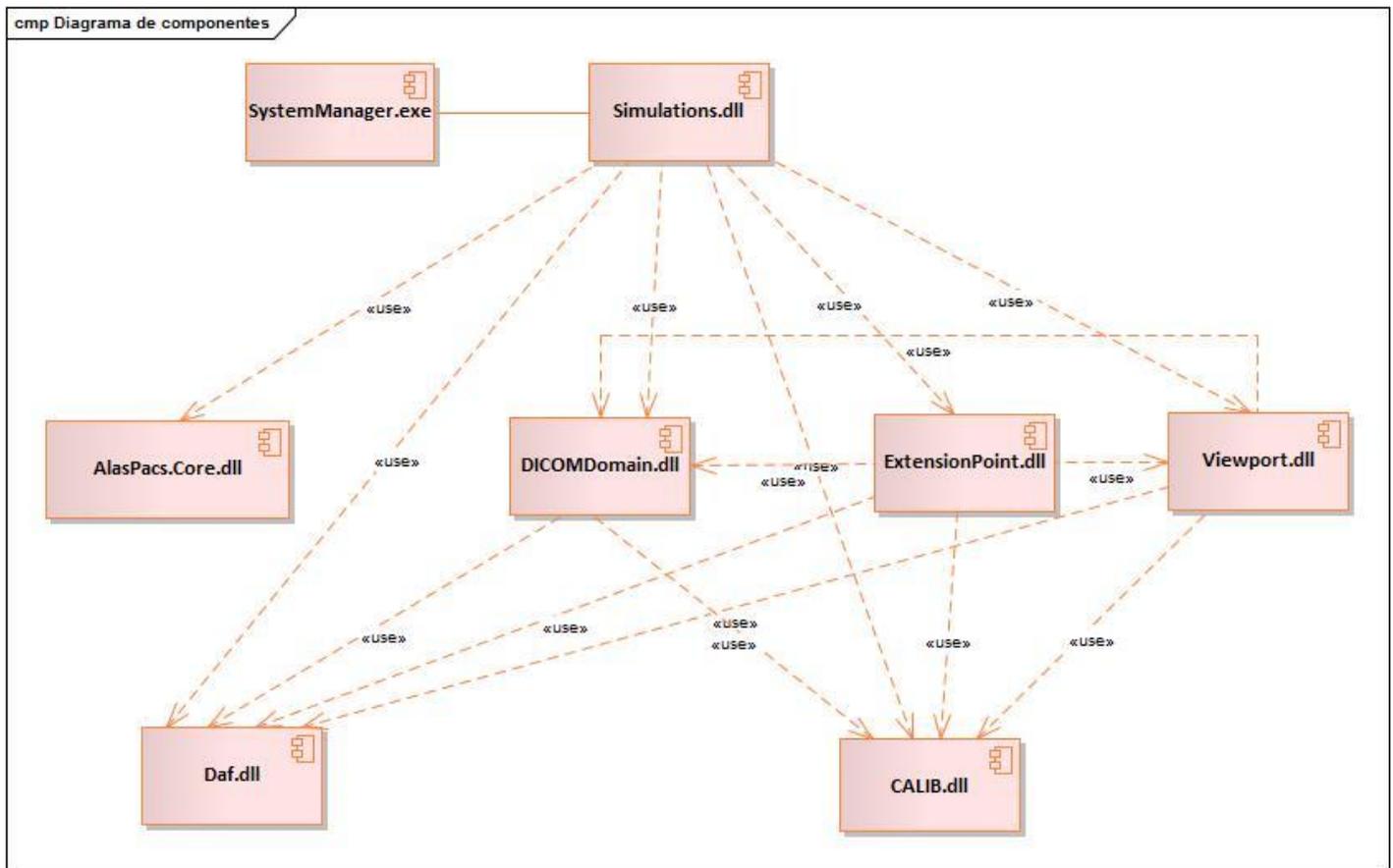


Figura 5. Diagrama de componentes del módulo de simulaciones de osteotomía para el sistema PQO.

#### 4.1.1. Descripción de los componentes

**CALIB.dll:** Librería que se encarga de la manipulación de archivos DICOM. Permite la lectura y descompresión de ficheros en este formato. Además, permite obtener la imagen del archivo DICOM, así como la información asociada a ella.

**AlasPacs.Core.dll:** Contiene el catálogo de servicios que el sistema brinda a sus extensiones y maneja la localización del mismo. Se emplea para integrar el módulo al sistema PQO.

**DICOMDomain.dll:** Esta biblioteca contiene un conjunto de clases que permiten la fácil interacción con los datos que brinda un fichero DICOM. Es utilizada en el módulo para obtener datos referentes a la calibración de la imagen cargada.

**Daf.dll:** Contiene paquetes de clases para la configuración de los ViewModel del sistema, la localización de servicios y soporte para la integración y el uso de comandos.

**ExtensionPoint.dll:** Contiene los contratos que definen la interacción entre el sistema y las partes extensibles del mismo. Se empleó para subscribirse a los cambios de idiomas que el sistema sufra, de tal forma que los textos que se muestran en la interfaz visual del módulo cambien en correspondencia con el lenguaje del sistema.

**Viewport.dll:** Componente que contiene un conjunto de clases para la creación de herramientas a utilizar sobre la imagen y su comportamiento. Se empleó para la delimitación de fragmentos siguiendo la lógica de la herramienta “Medición libre “.

**Simulations.dll:** Componente que se integra al sistema PQO en forma de plugin, brindando el servicio de simulaciones de osteotomía.

**SystemManager.exe:** Ejecutable que se encarga de cargar todos los componentes y librerías al iniciarse el sistema PQO.

## 4.2. Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Se utiliza tanto en el diseño como en la implementación y está compuesto por nodos, componentes que residen en los nodos y las relaciones entre estos. Un nodo es un elemento físico que representa un recurso computacional utilizado para modelar la topología del hardware sobre el que se ejecuta el sistema. Estos son conectados por asociaciones de comunicación, tales como enlaces de red, conexiones TCP/IP, entre otros.

El software se ejecuta sobre una red de computadoras o nodos de procesamiento. Por tal motivo, para la realización de los diagramas de despliegue se tienen en cuenta los requisitos no funcionales del sistema, tales como la disponibilidad, confiabilidad, rendimiento y escalabilidad.

En la figura 6 se muestra el modelo de despliegue del sistema de PQO.

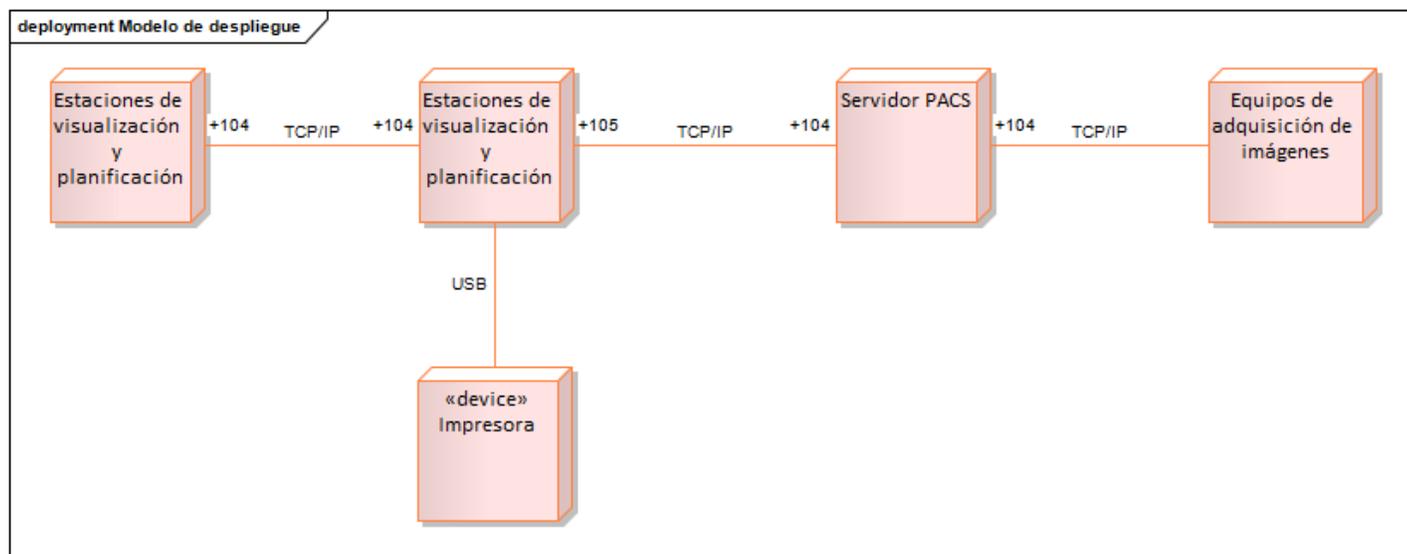


Figura 6. Modelo de despliegue del sistema PQO.

### 4.2.1. Descripción de los nodos que componen el diagrama de despliegue

**Equipos de adquisición de imágenes:** Equipos médicos que realizan los estudios a pacientes y envían las imágenes al servidor PACS a través de una conexión TCP/IP por el puerto 104.

**Servidor PACS:** Servidor que permitirá la gestión del archivo imagenológico de un hospital, así como la disponibilidad del mismo para las distintas estaciones clientes. El servidor PACS se comunica con las estaciones a través de una conexión TCP/IP por el puerto 105.

**Estaciones de visualización y planificación:** Ordenadores utilizados por los especialistas para realizar la visualización y el procesamiento de las imágenes médicas, así como la planificación quirúrgica ortopédica de las cirugías. Las estaciones pueden comunicarse a través de una conexión TCP/IP por el puerto 104.

**Impresora:** Permite imprimir imágenes médicas (impresión en placas) o los informes generados en las estaciones de visualización y planificación.

### 4.3. Estándares de codificación

Los estándares de codificación son una forma de normalizar la programación de forma tal que al trabajar en un proyecto, cualquiera de las personas involucradas en el mismo tenga acceso y comprenda el código.

Permiten definir la escritura y organización del código fuente de un programa para reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código legible, es de gran importancia para la calidad del software y para obtener un buen rendimiento (43). A continuación se muestran algunos de los aspectos para los que se establecieron estándares durante la implementación del sistema:

**Atributos:** Los nombres de los atributos se escribieron utilizando la notación *lowerCamelCase*, que consiste en utilizar las mayúsculas como separadores de palabras, comenzando la primera con minúscula.

En la clase controladora “SimulationBehavior” se declararon los siguientes atributos donde se evidencia la utilización del estilo aplicado:

```
private FreeMeasure _freeMeasure;  
private DrawingGroup _drawingGroup;
```

**Métodos:** En el caso de los métodos el estándar aplicado fue *PascalCase*, que cumple con los mismos requerimientos descritos por *UpperCamelCase* (utilizar las mayúsculas como separadores de palabras, comenzando la primera con mayúscula). *PascalCase* puede ajustarse y ser un tipo de *CamelCase* si se trata de conceptualizar, pero no sucede lo mismo a la inversa.

En la clase controladora “RotateBehavior”, en el método “RestaurarForma()” se muestra un ejemplo del estilo utilizado en cada uno de los métodos de las clases pertenecientes al módulo desarrollado:

```
public void RestaurarForma()
```

**Interfaces:** Para las interfaces el estándar empleado fue la Notación Húngara, que consiste en añadir prefijos o identificadores a los nombres y que indican su tipo. El prefijo utilizado para las interfaces en la implementación del sistema fue la letra “I”. A continuación se muestra un ejemplo de la utilización de este estilo:

```
public interface IRgBoxSimulationVm
```

**Llaves:** Las llaves de apertura se colocarán solitarias en la línea siguiente e indentadas al nivel de la línea cabecera del bloque. Las llaves de cierre se colocarán solitarias en la línea que sigue a la última línea del bloque. En el caso de cuerpos de bloque con una sola sentencia se podrá o no usar las llaves a gusto del

programador. Este estilo agrega más líneas de código al programa al ubicar las llaves solitarias en una línea, pero a su vez se gana en legibilidad del código.

El siguiente ejemplo muestra la utilización del estándar definido anteriormente en el desarrollo del módulo de simulaciones de osteotomía:

```
private void MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    if (_isRotating)
    {
        CommandManager.InvalidateRequerySuggested();
        _isRotating = false;
    }
    if (_isMovingCenter)
    {
        CommandManager.InvalidateRequerySuggested();
        _isMovingCenter = false;
    }
}
```

En este capítulo fueron expuestos elementos significativos relacionados con la implementación del módulo de simulación mediante una descripción detallada del diagrama de componentes. Se mostró además la disposición física de los nodos que componen el sistema a través del diagrama de despliegue. Se describieron los estándares de codificación empleados en la solución, los cuales permitieron obtener un sistema fácil de mantener en el transcurso del tiempo.

---

## **CONCLUSIONES**

Al concluir la presente investigación se logró desarrollar el módulo de simulaciones de osteotomía para el sistema de Planificación Quirúrgica Ortopédica.

- ✓ El análisis de los sistemas existentes a nivel mundial que realizan simulaciones de osteotomía y las entrevistas realizadas a los especialistas ortopédicos en el Hospital Militar Dr. Luis Díaz Soto, permitieron identificar las funcionalidades que componen el módulo desarrollado.
- ✓ El análisis de los sistemas homólogos permitió determinar que los mismos no constituyen una solución viable a la problemática actual, ya que son privativos, poseen altos costos y tienen limitantes para su ejecución en determinados sistemas operativos.
- ✓ Se utilizaron los estilos arquitectónicos basado en extensiones y orientado a dominio, permitiendo facilidad de ampliación y mantenimiento de la aplicación en un futuro.
- ✓ Se emplean como patrones arquitectónicos el MVVM, el IoC y el DI y algunas tecnologías novedosas como WPF y MEF, garantizando adaptabilidad, estabilidad, facilidad de mantenimiento, flexibilidad e interfaces más atractivas.
- ✓ El módulo desarrollado permitirá realizar planificaciones preoperatorias de osteotomías antes de que los pacientes sean sometidos a las intervenciones quirúrgicas y reducir el tiempo de ejecución de los procedimientos quirúrgicos.

---

## **RECOMENDACIONES**

En aras de enriquecer la solución propuesta se propone la siguiente recomendación:

- ✓ Incorporar al módulo desarrollado, plantillas digitales de implantes, para ser usadas en las simulaciones de osteotomía.

### REFERENCIAS BIBLIOGRÁFICAS

1. Preoperative planning simulator for spinal deformity surgeries. *PubMed*. [En línea] Septiembre de 2008. [Citado el: 18 de Febrero de 2014.] <http://www.ncbi.nlm.nih.gov/pubmed/18794755>.
2. **Ordóñez, y otros, y otros.** *Sistemas de Comunicación y Gestión de Imágenes Médicas. PACS y Estándar DICOM*. España: Ediciones de la Universidad de Castilla-La Mancha : s.n., 2007.
3. **PAS, Grupo.** *Estándar y Protocolo de Imágenes Médicas DICOM*. Universidad de Deusto : s.n., 2005.
4. **Ramírez, Esmitt y Coto, Ernesto.** *TRAUMAPLAN: para la planificación preoperatoria en traumatología*. 2011. ISSN: 1690-7515.
5. **Ripoll y De Prado.** Servicio de Traumatología y Cirugía Ortopédica. [En línea] 2013. [Citado el: 24 de Febrero de 2014.] <http://www.ripollydeprado.com/ver/osteotomia-tibial-alta/>.
6. **Masson, Elsevier.** Técnicas Quirúrgicas en Ortopedia y Traumatología. *EMC*. [En línea] 2005. [Citado el: 25 de Febrero de 2014.] <http://www.em-consulte.com/es/article/770421/osteotomias-del-femur-distal>.
7. **Martínez Martínez, Alfonso y Cervantes Maceda, Humberto.** *Diseño y construcción de una arquitectura de línea de producto para sistemas PACS. Maestría en Ciencias y Tecnologías de la Información*. 2010.
8. **Rincón, Marcela y Rodríguez, Alejandra.** *Sistemas PACS y el formato DICOM. Grupo de Bioinstrumentación y procesamiento de señales "Kirón"*. 2009.
9. **García Mateos.** Proyectos de Fin de Carrera. [En línea] [Citado el: 27 de Febrero de 2014.] <http://dis.um.es/~ginesgm/varios.html>.
10. **Ballesteros Herranz, Fernando.** *Desarrollo de aplicaciones DICOM para la gestión de imágenes*. 2003.
11. **K.Huang, H.** PACS and Imaging Informatics basic principles and applications. [En línea] 2010. [Citado el: 28 de Febrero de 2014.] <http://www.amazon.es/PACS-Imaging-Informatics-Principles-Applications/dp/0470373725>.
12. PRmob. [En línea] 2012. [Citado el: 28 de Febrero de 2014.] <http://es.prmob.net/cirug%C3%ADa/bangkok-hospital/cirug%C3%ADa-ortop%C3%A9dica-1371488.html>.
13. **Luengo Orol, Francisca.** ITC. Instituto Tecnológico de Canarias. [En línea] 2011. [Citado el: 2 de Marzo de 2014.] <http://www.itccanarias.org/web/tecnologias/mecanica/humic.jsp?lang=es>.

14. **SYNTHESES.** *Sistema de osteotomía TomoFix para la rodilla. Soluciones clínicas para el restablecimiento funcional de la articulación de la rodilla.* 2012.
15. **OrthoView LLC.** Orthopaedic Digital Pre-operative Planning. *OrthoView.* [En línea] 2012. [Citado el: 2 de Marzo de 2014.] <http://www.orthoview.com/es/>.
16. **Orthocrat.** *TraumaCad User's Guide.* 2008.
17. **Microsoft.** Información general y conceptual sobre .NET Framework. [En línea] [Citado el: 3 de Marzo de 2014.] <http://msdn.microsoft.com/library/zw4w595w.aspx>.
18. **Marzal, Andrés.** *Desarrollo de aplicaciones con .NET y WPF.* 2010.
19. **Sells, Chris y Griffiths, Ian.** *Programming WPF.* s.l. : O'Reilly Media, 2007.
20. Información general sobre Managed Extensibility Framework. [En línea] [Citado el: 5 de Marzo de 2014.] [http://msdn.microsoft.com/es-es/library/dd460648.aspx#what\\_is\\_mef](http://msdn.microsoft.com/es-es/library/dd460648.aspx#what_is_mef).
21. **Cortijo Bon, Francisco.** Curso de C#. [En línea] [Citado el: 5 de Marzo de 2014.] <http://elvex.ugr.es/decsai/csharp/language/intro.xml>.
22. **Sharp, John.** *Microsoft Visual C#. Paso a paso.* 2010.
23. ¿Qué es el modelo CMM? *SoftQaNetwork.* [En línea] 13 de Junio de 2006. [Citado el: 8 de Marzo de 2014.] <http://www.softqanetwork.com/%C2%BFque-es-el-modelo-cmm>.
24. VATES ingeniería de software. [En línea] 2013. [Citado el: 8 de Marzo de 2014.] <http://www.vates.com/cmml/cmml.html>.
25. **Chirino Delgado, Jenis E.** *Evaluación de organizaciones software mediante el modelo CMMI .* 2010.
26. VATES ingeniería de software. *VATES ingeniería de software.* [En línea] 2013. [Citado el: 5 de abril de 2014.] <http://www.vates.com/cmml/cmml.html>.
27. SCHOLARIUM. *UML.* [En línea] 2011. [Citado el: 10 de Marzo de 2014.] <http://scholarium.info/uml/>.
28. KDE Documentation. [En línea] [Citado el: 10 de Marzo de 2014.] <http://docs.kde.org/stable/es/kdesdk/umbrello/umlbasics.html>.
29. **Jacobson, y otros, y otros.** *El proceso unificado de desarrollo de software.* 2000.

30. **Sparx Systems.** *Enterprise Architect User Guide.* 2009.
31. Microsoft Partner. *ITTalent.* [En línea] 2010. [Citado el: 12 de Marzo de 2014.] <http://www.ittalent.com.co:26619/Certificaciones/VISUALST/Paginas/default.aspx>.
32. EtnasSoft. *OpenLibra.* [En línea] 2011. [Citado el: 13 de Marzo de 2014.] <http://www.etnassoft.com/biblioteca/tortoisesvn-uncliente-de-subversion-para-windows/>.
33. **Larman, Craig y Prentice, Hall.** *UML y Patrones. 2ª Edición. Modelado del Dominio.* 2003.
34. **Kontoya, G y Sommerville.** *I.Requirements Engineering: Processes and Techniques.* 2010.
35. **Gil Torres, Manuel.** *Fundamentos del diseño de software. Ingeniería del software y diseño del software.* 2010.
36. **Pressman, Roger S.** *INGENIERÍA DEL SOFTWARE. Sexta edición.* 2008.
37. *Idem 18.*
38. **Shaw, Mary y Garlan, David.** *An introduction to software architecture.* School of Computer Science, Carnegie Mellon University : s.n., 1994.
39. **Córdoba, Jorge.** The Cold Sun. Programming articles about Delphi, .NET, Concurrency and more. *Sistema de plugins con C#. Parte 1.* [En línea] 2009. [Citado el: 17 de Marzo de 2014.] <http://www.thecoldsun.com/es/content/01-2009/sistema-de-plugins-con-c-parte-i-conceptos>.
40. **Microsoft|Architecture.** *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0.* 2010.
41. **Mora, David.** Patrón MVVM explicado. *Maromas Digitales.* [En línea] [Citado el: 18 de Marzo de 2014.] <http://maromasdigitales.net/2010/05/patron-mvvm-explicado/>.
42. Teoría 1: El arte de modelar. [En línea] 2010. [Citado el: 23 de Marzo de 2014.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
43. **Microsoft.** Developer Network. Revisiones de código y estándares de codificación. [En línea] [Citado el: 28 de Mayo de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

## BIBLIOGRAFÍA

- Ballesteros Herranz, Fernando.** *Desarrollo de aplicaciones DICOM para la gestión de imágenes* . 2003.
- Córdoba, Jorge.** 2009. The Cold Sun. Programming articles about Delphi, .NET, Concurrency and more. *Sistema de plugins con C#. Parte 1*. [En línea] 2009. [Citado el: 17 de Marzo de 2014.] <http://www.thecoldsun.com/es/content/01-2009/sistema-de-plugins-con-c-parte-i-conceptos>.
- Cortijo Bon, Francisco.** Curso de C#. [En línea] [Citado el: 5 de Marzo de 2014.] <http://elvex.ugr.es/decsai/csharp/language/intro.xml>.
- Elsevier Masson.** EMC Técnicas Quirúrgicas en Ortopedia y Traumatología. [En línea] 2005. [Citado el: 26 de febrero de 2014.] <http://www.em-consulte.com/es/article/770421/osteotomias-del-femur-distal>.
- EtnasSoft. *OpenLibra*. [En línea] 2011. [Citado el: 13 de Marzo de 2014.] <http://www.etnassoft.com/biblioteca/tortoisesvn-uncliente-de-subversion-para-windows/>.
- Fernández Ferrer, Arlettys y González Vega, Reynier.** *Desarrollo del módulo de Trauma para el sistema de Planificación Quirúrgica Ortopédica*. Ciudad de la Habana : s.n., 2012.
- Fisterra.com. Atención Primaria en la Red. *Radiografía*. [En línea] [Citado el: 21 de Febrero de 2014.] <http://www.fisterra.com/salud/3proceDT/radiografia.asp>.
- García Mateos.** Proyectos de Fin de Carrera. [En línea] [Citado el: 28 de Febrero de 2014.] <http://dis.um.es/~ginesgm/varios.html>.
- Gil Torres, Manuel.** *Fundamentos del diseño de software. Ingeniería del software y diseño del software*.
- González Mesa, Yoel, Cartelle Cruz, María de Jesús y Junco López, Yanía.** *Sistema para la Planificación Quirúrgica Ortopédica en el área de Pediatría*. Ciudad de la Habana : s.n., 2013.
- Gracia, Joaquín.** IngenieroSoftware. Calidad. *CMM-CMMI Nivel 2*. [En línea] [Citado el: 9 de Marzo de 2014.] <http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>.
- Hernández León, Rolando Alfredo y Coello González, Zayda.** 2002. *El paradigma cuantitativo de la investigación científica*. Ciudad de La Habana : Eduniv, 2002. ISBN: 959-16-0343-6.

- H.K.Huang.** PACS and Imaging Informatics basic principles and applications. [En línea] 2010. [Citado el: 1 de Marzo de 2014.] <http://www.amazon.es/PACS-Imaging-Informatics-Principles-Applications/dp/0470373725>.
- Información general sobre Managed Extensibility Framework. [En línea] [Citado el: 5 de Marzo de 2014.] [http://msdn.microsoft.com/es-es/library/dd460648.aspx#what\\_is\\_mef](http://msdn.microsoft.com/es-es/library/dd460648.aspx#what_is_mef).
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. 2000.
- Junco López, Yania y Camacho Díaz, Ariel Felipe.** 2012. *Desarrollo del módulo de columna para el sistema de Planificación Quirúrgica Ortopédica*. La Habana : s.n., 2012.
- KDE Documentation. [En línea] [Citado el: 10 de Marzo de 2014.] <http://docs.kde.org/stable/es/kdesdk/umbrello/umlbasics.html>.
- Kontoya, G y Sommerville.** *I.Requirements Engineering: Processes and Techniques*. 2010.
- Larman, Craig.** 1999. *UML y patrones. Introducción al análisis y diseño orientado a objetos*. 1999.
- Luengo Orol, Francisca .** 2011. ITC. Instituto Tecnológico de Canarias. [En línea] 2011. [Citado el: 2 de Marzo de 2014.] <http://www.itccanarias.org/web/tecnologias/mecanica/humic.jsp?lang=es>.
- Marzal, Andrés.** *Desarrollo de aplicaciones con .NET y WPF*. 2010.
- Microsoft Partner. *ITTalent*. [En línea] [Citado el: 12 de Marzo de 2014.] <http://www.ittalent.com.co:26619/Certificaciones/VISUALST/Paginas/default.aspx>.
- Microsoft.** Información general y conceptual sobre .NET Framework. [En línea] [Citado el: 3 de Marzo de 2014.] <http://msdn.microsoft.com/library/zw4w595w.aspx>.
- Microsoft| Architecture.** *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0*. 2010.
- Mora, David.** Patrón MVVM explicado. *Maromas Digitales*. [En línea] [Citado el: 18 de Marzo de 2014.] <http://maromasdigitales.net/2010/05/patron-mvvm-explicado/>.
- Orthocrat.** *TraumaCad User's Guide*. 2008.
- OrthoView LLC.** OrthoView. Orthopaedic Digital Pre-operative Planning. [En línea] [Citado el: 2 de Marzo de 2014.] <http://www.orthoview.com/es/>.

- Pello, Javier.** SoftQaNetwork. *¿Que es el Modelo CMM?* [En línea] 13 de Junio de 2006. [Citado el: 8 de Marzo de 2014.] <http://www.softqanetwork.com/%C2%BFque-es-el-modelo-cmm>.
- Preoperative planning simulator for spinal deformity surgeries. *PubMed*. [En línea] Septiembre de 2008. [Citado el: 18 de Febrero de 2014.] <http://www.ncbi.nlm.nih.gov/pubmed/18794755>.
- PRmob. [En línea] 2012. [Citado el: 2 de Marzo de 2014.] <http://es.prmob.net/cirug%C3%ADa/bangkok-hospital/cirug%C3%ADa-ortop%C3%A9dica-1371488.html>.
- Ramírez, Esmitt y Coto, Ernesto. 2011.** *TRAUMAPLAN: para la planificación preoperatoria en traumatología*. 2011. ISSN: 1690-7515.
- Ripoll y De Prado.** Servicio de Traumatología y Cirugía Ortopédica. [En línea] 2013. [Citado el: 25 de Febrero de 2014.] <http://www.ripollydeprado.com/ver/osteotomia-tibial-alta/>.
- Rivera Suárez, Yicel y Noruega López, Pedro Jorge. 2012.** *Desarrollo del módulo de extremidades para el sistema de Planificación Quirúrgica Ortopédica*. La Habana : s.n., 2012.
- Sells, Chris y Griffiths, Ian.** *Programming WPF. s.l. : O'Reilly Media. 2007.*
- Sharp, John.** *Microsoft Press Microsoft Visual C Sharp. Step by Step.* 2010.
- Shaw, Mary y Garlan, David .** *An introduction to software architecture*. School of Computer Science, Carnegie Mellon University : s.n., 1994.
- Sotolongo Pack, Laura Migdalia y González Yera, Yoandy.** *Implementación e integración de los módulos en el Planificador Quirúrgico Ortopédico*. Ciudad de La Habana : s.n., 2013.
- Sparx Systems.** *Enterprise Architect User Guide.* 2009.
- S. Pressman, Roger. 2008.** *Ingeniería del software. Un enfoque práctico.* 2008.
- Teoría 1: El arte de modelar. [En línea] 2010. [Citado el: 23 de Marzo de 2014.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
- UML y Patrones. 2ª Edición. Craig Larman. Prentice Hall. *Modelado del Dominio*. [En línea] 2003. [Citado el: 15 de Marzo de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
- VATES ingeniería de software. [En línea] 2013. [Citado el: 8 de Marzo de 2014.] <http://www.vates.com/cmimi/cmimi.html>.

### ANEXOS

**Anexo 1.** Entrevista realizada a los especialistas del Hospital Militar Dr. Luis Díaz Soto.

1. ¿Cuáles son los pasos para realizar la planificación preoperatoria de las osteotomías?
2. ¿Cuáles son los instrumentos y materiales usados durante la planificación preoperatoria de las osteotomías?
3. ¿Es complejo el procedimiento que se sigue para realizar la planificación preoperatoria?
4. ¿Cuáles son los objetivos de realizar una planificación preoperatoria de las osteotomías?
5. ¿Cómo se realizan los calcos a las radiografías?
6. ¿Siempre están disponibles los materiales necesarios para realizar una planificación preoperatoria?

**Anexo 2.** Prefijos de los requerimientos no funcionales por categoría.

Categoría	Prefijo	Ejemplo
Usabilidad	RNU	RNU 1. Requisito de usabilidad A RNU 2. Requisito de usabilidad B
Diseño e Implementación	RNDI	RNDI 1. Requisito de diseño e implementación
Fiabilidad	RNF	RNF 1. Requisito de fiabilidad
Funcionamiento	RNFO	RNFO 1. Requisito de funcionamiento
Soporte	RNSO	RNSO 1. Requisito de soporte
Interfaz de usuario	RNIU	RNIU 1. Requisito de interfaz de usuario
Interconexión	RNI	RNI 1. Requisito de interconexión

**Anexo 3.** Casos de uso expandidos.

- Caso de uso - Aplicar herramienta lazo

<b>Objetivo</b>	Seleccionar un fragmento de la imagen, quedando cortado y listo para mover.
-----------------	---

<b>Actores</b>	Especialista (inicia).	
<b>Resumen</b>	El especialista selecciona un fragmento de la imagen de forma variable con el cursor, quedando cortado y listo para mover.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico.	
<b>Referencias</b>	RF 1.1	
<b>Precondiciones</b>	La imagen ha sido cargada previamente.	
<b>Postcondiciones</b>	Se seleccionó el fragmento deseado y se corta el mismo.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Aplicar herramienta lazo&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción "Lazo".</li> <li>2. El especialista selecciona un fragmento de la imagen de forma variable con el cursor.</li> <li>3. El fragmento seleccionado queda cortado y listo para mover.</li> <li>4. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos no funcionales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

- Caso de uso - Mover fragmento

<b>Objetivo</b>	Mover el fragmento cortado con la herramienta lazo.	
<b>Actores</b>	Especialista (Inicia)	
<b>Resumen</b>	El médico selecciona la herramienta mover de la sección de simulaciones y mueve el fragmento cortado por el área de trabajo.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario.	
<b>Referencias</b>	RF 1.2	
<b>Precondiciones</b>	Se ha cortado un fragmento.	
<b>Postcondiciones</b>	Se movió el fragmento	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Mover fragmento&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción "Mover".</li> <li>2. El especialista mueve el fragmento.</li> <li>3. El sistema muestra el fragmento en las posiciones en los que va colocando el especialista el fragmento.</li> <li>4. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	No aplica
<b>Asuntos</b>	No aplica	

<b>pendientes</b>	
-------------------	--

- Caso de uso - Rotar fragmento

<b>Objetivo</b>	Rotar el fragmento en la dirección deseada, teniendo como centro de rotación un punto de anclaje.
<b>Actores</b>	Especialista (Inicia)
<b>Resumen</b>	El especialista selecciona la herramienta “Rotar”, desplaza el punto de anclaje hasta donde necesite y rota el fragmento en la dirección deseada.  El especialista selecciona la opción “Restaurar” y el fragmento retorna a su posición original antes de haber sido rotado.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Secundario.
<b>Referencias</b>	RF 1.3, RF 1.4, RF 1.5
<b>Precondiciones</b>	Se ha seleccionado un fragmento de la imagen.  Para cambiar la posición del punto de anclaje que representa el centro de rotación del fragmento, este no debe haberse movido de su lugar original.
<b>Postcondiciones</b>	Se rotó el fragmento en la dirección deseada.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt;Rotar fragmento&gt;</b>	
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Rotar”.</li> <li>2. El especialista mantiene el clic izquierdo presionado sobre el fragmento y mueve el mouse en la dirección deseada.</li> </ol>	

<ol style="list-style-type: none"> <li>3. El sistema rota el fragmento en la posición deseada.</li> <li>4. Termina el caso de uso</li> </ol>		
<b>Flujo básico &lt;Cambiar posición de punto de anclaje de rotación&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Rotar”.</li> <li>2. El especialista da clic izquierdo sobre el fragmento.</li> <li>3. El especialista mantiene presionado el clic izquierdo del mouse sobre el punto de anclaje y lo mueve en la dirección deseada dentro del área del fragmento.</li> <li>4. El sistema cambia el centro de rotación hacia la posición donde fue colocado el punto de anclaje.</li> <li>5. Termina el caso de uso</li> </ol>		
<b>Flujo básico &lt;Restaurar rotación del fragmento&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Restaurar posición”.</li> <li>2. El sistema muestra el fragmento en su posición original.</li> <li>3. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos no funcionales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

- Caso de uso - Eliminar fragmento

<b>Objetivo</b>	Eliminar el fragmento seleccionado.
<b>Actores</b>	Especialista (Inicia)
<b>Resumen</b>	El especialista selecciona la opción Eliminar o Eliminar todas, para

	eliminar un fragmento o todos los fragmentos seleccionados.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario.	
<b>Referencias</b>	RF 1.6	
<b>Precondiciones</b>	Se ha seleccionado un fragmento.	
<b>Postcondiciones</b>	Se eliminó el fragmento o todos los fragmentos.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Eliminar fragmento&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Eliminar”.</li> <li>2. El sistema elimina el fragmento seleccionado.</li> <li>3. Termina el caso de uso.</li> </ol>		
<b>Flujo alterno</b>		
<b>2a. Eliminar todos los fragmentos</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Eliminar Todas”.</li> <li>2. El sistema elimina todos los fragmentos que han sido construidos.</li> <li>3. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos no funcionales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

- Caso de uso - Dicomizar imagen

<b>Objetivo</b>	Dicomizar una imagen o un conjunto de ellas.	
<b>Actores</b>	Especialista (Inicia)	
<b>Resumen</b>	El especialista realiza una simulación y desea dicomizar la imagen con los cambios realizados.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundario	
<b>Referencias</b>	RF 1.7	
<b>Precondiciones</b>	No aplica	
<b>Postcondiciones</b>	Se dicomizó la imagen.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Dicomizar imagen&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista realiza una simulación de un procedimiento quirúrgico.</li> <li>2. El especialista selecciona la opción "Dicomizar".</li> <li>3. El sistema dicomiza la o las imágenes.</li> <li>4. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales no</b>	No aplica	
<b>Asuntos</b>	No aplica	

<b>pendientes</b>	
-------------------	--

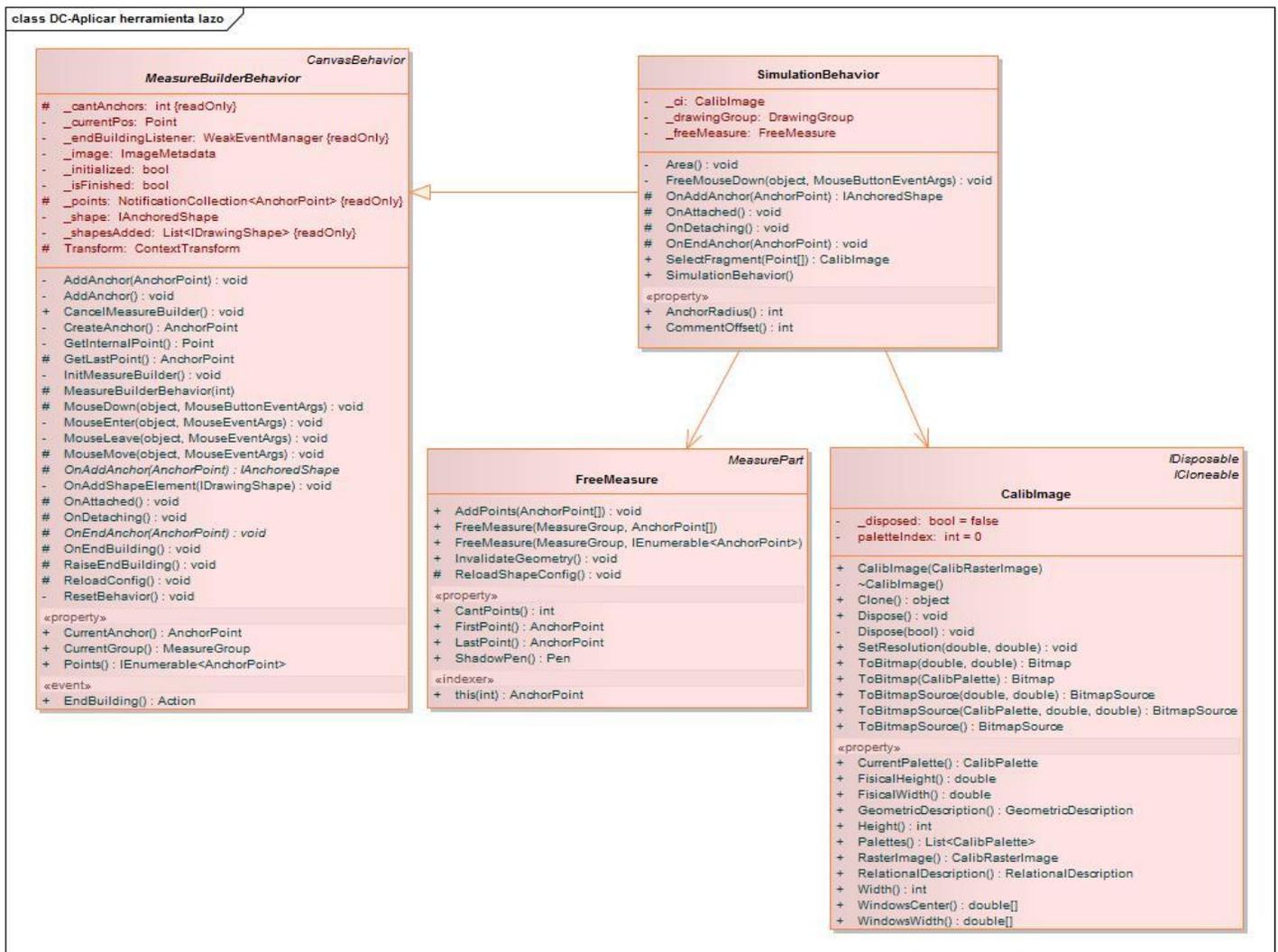
- Caso de uso - Limpiar imagen

<b>Objetivo</b>	Permite restaurar la imagen luego de haberse realizado algún corte de fragmento.	
<b>Actores</b>	Especialista (Inicia)	
<b>Resumen</b>	El especialista selecciona la opción “Limpiar imagen” para restaurar la imagen a su forma original. Cuando se limpia la imagen, el fragmento que ha sido construido se mantiene como un componente separado de la imagen.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundario	
<b>Referencias</b>	RF 1.8	
<b>Precondiciones</b>	Se ha cortado al menos un fragmento de la imagen.	
<b>Postcondiciones</b>	Se limpió la imagen.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Limpiar imagen&gt;</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción “Limpiar Imagen” de la sección Simulaciones.</li> <li>2. El sistema restaura la imagen a su forma inicial dejando separado el fragmento que ha sido extraído.</li> <li>3. Termina el caso de uso.</li> </ol>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos no funcionales</b>	No aplica	

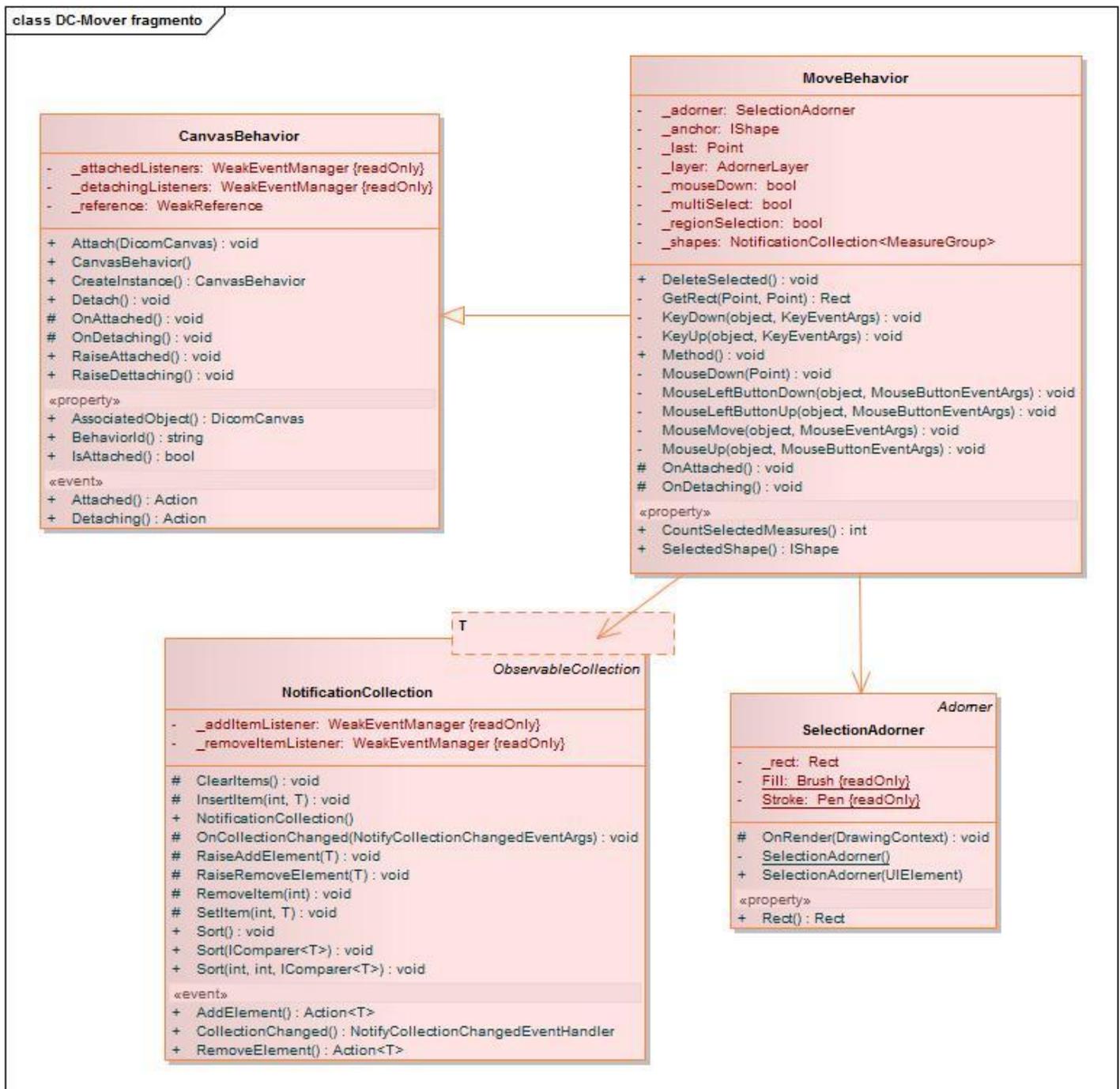
Asuntos pendientes	No aplica
--------------------	-----------

### Anexo 4. Diagramas de clases del diseño de los casos de uso.

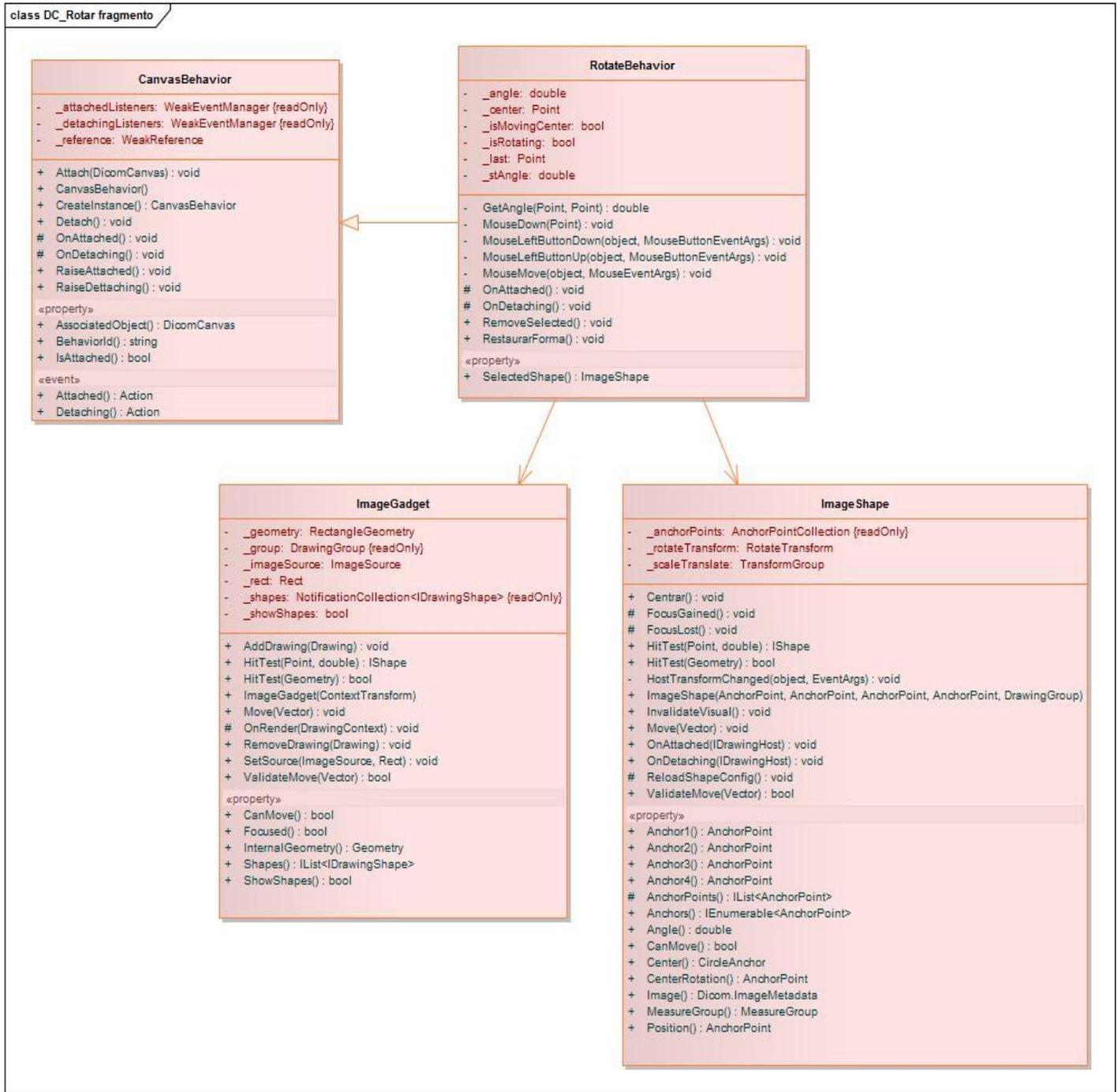
- Diagrama de clase del CUS - Aplicar herramienta lazo



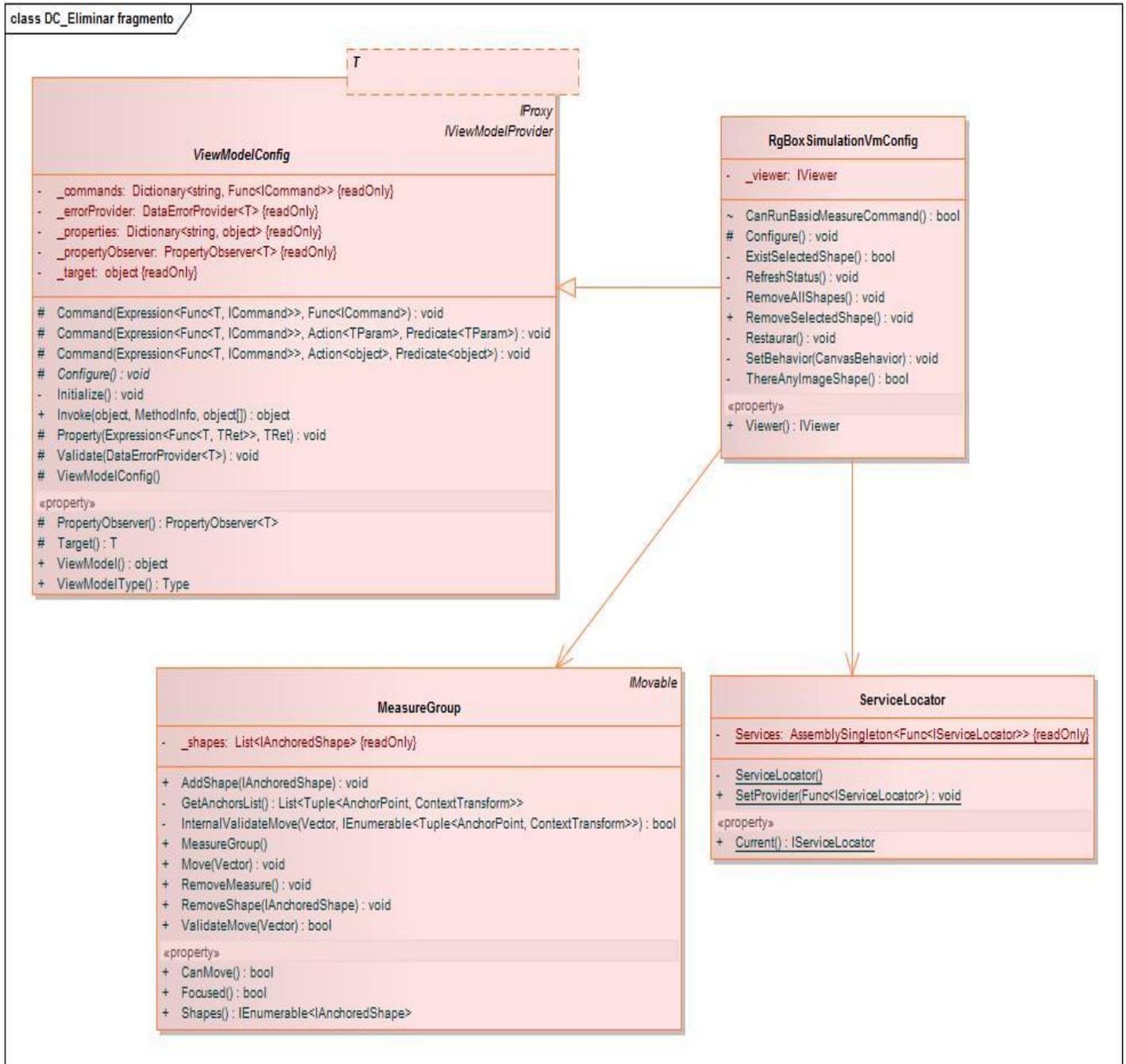
- Diagrama de clase del CUS – Mover fragmento



- Diagrama de clase del CUS – Rotar fragmento



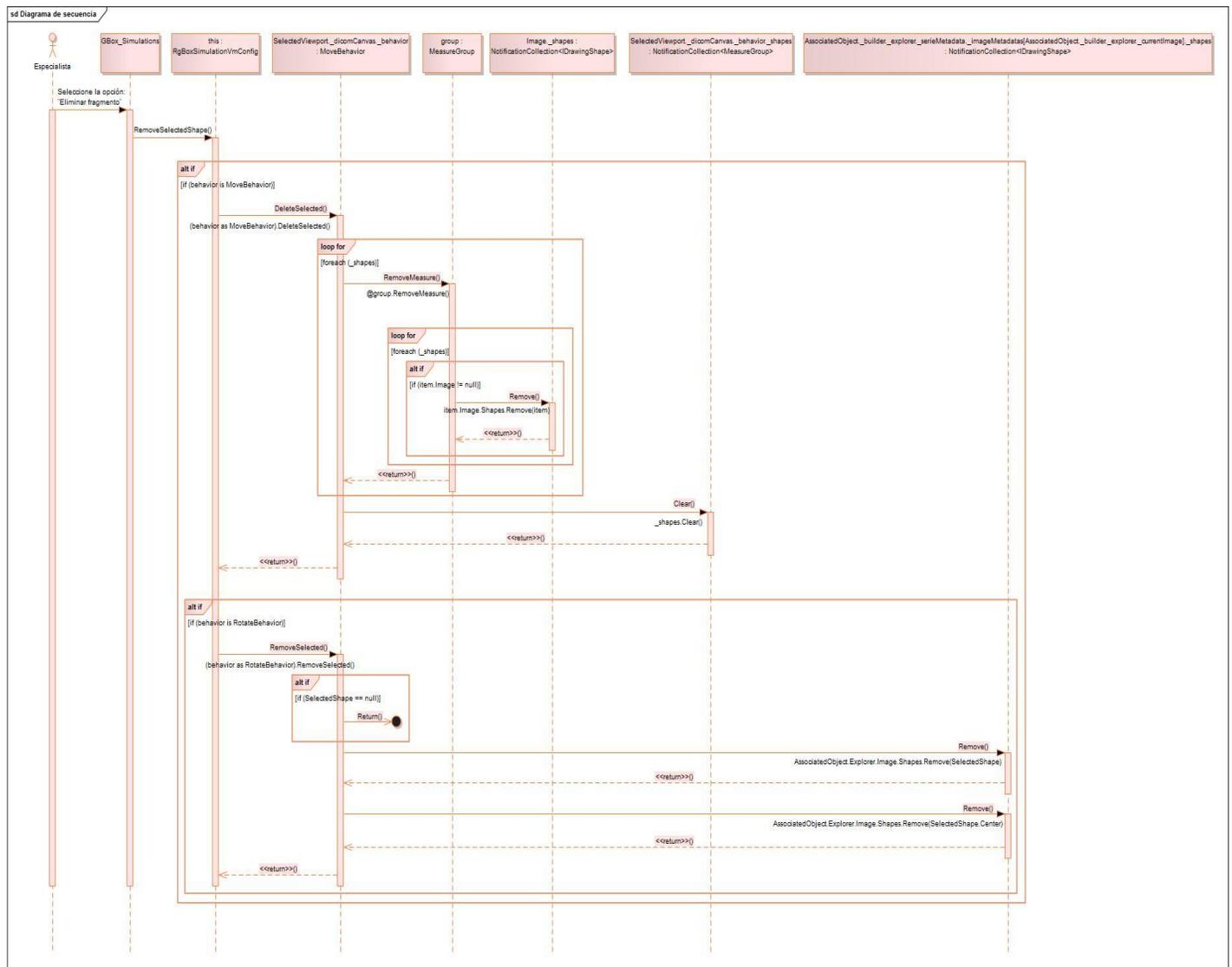
- Diagrama de clase del CUS – Eliminar fragmento



## Anexo 5. Diagramas de secuencia de los casos de uso.

- Diagrama de secuencia del CUS – Eliminar fragmento

A continuación se muestra el diagrama de secuencia del caso de uso del sistema “Eliminar fragmento”. El resto de los diagramas de secuencia, por su dimensión, se encuentran en el Expediente de Proyecto del Planificador Quirúrgico Ortopédico en el archivo del Enterprise Architect que contiene todos los diagramas referentes al sistema.



### Anexo 6. Descripciones de clases.

- Clase SimulationBehavior.

**Propósito:** Definir el comportamiento a la hora de seleccionar un fragmento sobre la imagen.

#### Descripción:

<b>Nombre:</b> SimulationBehavior.	
<b>Tipo de clase:</b> controladora	
<b>Atributo</b>	<b>Tipo</b>
_freeMeasure	FreeMeasure
_drawingGroup	DrawingGroup
<b>Para cada responsabilidad:</b>	
Nombre:	OnAttached ()
Descripción:	Subscribe el panel de visualización al evento capturador de las pulsaciones del mouse.
Nombre:	OnDetaching ()
Descripción:	Elimina la subscripción del panel de visualización al evento capturador de las pulsaciones del mouse.
Nombre:	FreeMouseDown (sender: object, e: MouseButtonEventArgs)
Descripción:	Maneja el evento referente a las pulsaciones del mouse.
Nombre:	OnAddAnchor (point: AnchorPoint)
Descripción:	Agrega un nuevo punto de anclaje para la creación de fragmentos.
Nombre:	SelectFragment (points: params Point[])
Descripción:	Selecciona el fragmento determinado por los puntos ubicados por el usuario.

- Clase RotateBehavior.

**Propósito:** Rotar el fragmento seleccionado sobre el área de visualización.

**Descripción:**

<b>Nombre:</b> RotateBehavior.	
<b>Tipo de clase:</b> controladora	
<b>Atributo</b>	<b>Tipo</b>
_isRotating	bool
_isMovingCenter	bool
_last	Point
_center	Point
_angle	double
_stAngle	double
<b>Para cada responsabilidad:</b>	
Nombre:	OnAttached ()
Descripción:	Subscribe el panel de visualización a los eventos correspondientes al mouse.
Nombre:	OnDetaching ()
Descripción:	Elimina la subscripción del panel de visualización a los eventos correspondientes al mouse.
Nombre:	MouseDown (sender: object, e: MouseButtonEventArgs)
Descripción:	Maneja el evento referente a las pulsaciones del botón izquierdo del mouse.
Nombre:	MouseMove (sender: object, e: MouseEventArgs)
Descripción:	Maneja el evento referente al movimiento del mouse.
Nombre:	MouseUp (sender: object, e: MouseButtonEventArgs)

Descripción:	Maneja el evento referente a la liberación del botón izquierdo del mouse.
Nombre:	GetAngle (center: Point, pos: Point)
Descripción:	Calcula el ángulo formado por la línea que une dos puntos respecto a la horizontal.
Nombre:	RemoveSelected ()
Descripción:	Elimina el fragmento de imagen seleccionado.
Nombre:	RestaurarForma ()
Descripción:	Restaura la rotación del fragmento de imagen seleccionado.

- Clase ImageShape.

**Propósito:** Describe un fragmento de imagen.

**Descripción:**

<b>Nombre:</b> ImageShape.	
<b>Tipo de clase:</b> entidad	
<b>Atributo</b>	<b>Tipo</b>
_scaleTranslate	TransformGroup
_rotateTransform	RotateTransform
_anchorPoints	AnchorPointCollection
<b>Para cada responsabilidad:</b>	
Nombre:	InvalidateVisual ()
Descripción:	Actualiza el visual de un fragmento de imagen después de haber sido trasladado o rotado.
Nombre:	AnchorPoints ()
Descripción:	Devuelve una lista con los puntos que delimitan el fragmento de imagen.

Nombre:	Move (vector: Vector)
Descripción:	Describe la forma en que se mueve el fragmento de imagen en el mismo sentido de un vector determinado.
Nombre:	CanMove ()
Descripción:	Determina si el fragmento de imagen se puede mover.
Nombre:	FocusGained ()
Descripción:	Describe la forma en que el fragmento de imagen obtiene el foco.
Nombre:	FocusLost ()
Descripción:	Describe la forma en que el fragmento de imagen pierde el foco.

### GLOSARIO DE TÉRMINOS

**Artritis:** La artritis es una enfermedad de las articulaciones (áreas donde se unen dos huesos, tales como el codo y la rodilla), en la que los signos más comunes son: dolor, rigidez, inflamación, calor y enrojecimiento.

**Artroplastias:** Son operaciones con las que se trata de reconstruir la articulación y la mecánica articular. Esta técnica varía de acuerdo con la anatomía y la función de cada articulación.

**Cifosis:** Es la curvatura de la columna que produce un arqueamiento o redondeo de la espalda, llevando a que se presente una postura jorobada o agachada. También conocida como enfermedad de Scheuermann, es provocada por el apretamiento de varios huesos de la columna (vértebras) consecutivos.

**DICOM** (Digital Imaging and Communications in Medicine): Estándar para el tratamiento de imágenes digitales y comunicaciones para el campo de la medicina, que facilita el manejo de la información médica entre hospitales y centros de investigación.

**Estándar:** Sirve como tipo, modelo, norma, patrón o referencia.

**Faceta articular:** Pequeña superficie plana de un hueso en el punto donde se articula con otra estructura.

**Fractura:** Es la pérdida de continuidad normal de la sustancia ósea. La fractura es una discontinuidad en los huesos, a consecuencia de golpes, fuerzas o tracciones cuyas intensidades superen la elasticidad del hueso. El término es extensivo para todo tipo de roturas de los huesos, desde aquellas en que el hueso se destruye amplia y evidentemente, hasta aquellas lesiones muy pequeñas e incluso microscópicas.

**Genu varo/valgo:** Ángulo formado entre el muslo y la pierna en el plano frontal. Cuando el segmento distal se aleja de la línea media (rodilla junta) se habla de genu valgo y cuando sucede lo contrario (rodillas separadas) se denomina genu varo.

**Gonartrosis:** La gonartrosis es la alteración crónica de la articulación de la rodilla. Puede localizarse en varias zonas: entre el fémur y la tibia (artrosis femorotibial interna o externa), entre el fémur y la patella o rótula (artrosis femoro patelar) o entre el fémur, la tibia y la rótula.

**Goniómetro:** Un goniómetro es un instrumento de medición con forma de semicírculo o círculo graduado en 180° o 360°, utilizado para medir o construir ángulos.

**Hiperextensión:** Consiste en la extensión de cualquier articulación más allá del punto final del límite de su intervalo de recorrido. En el caso de la espina lumbar, esto significa arquear forzosamente la espalda por encima de su curvatura normal, trayendo consigo posibles hernias y lesiones.

**Juanete:** Un juanete (Hallux valgus) es una desviación de la articulación de la base del dedo gordo del pie, que puede provocar inflamación y dolor, ya que es una zona del cuerpo que soporta gran parte de su peso.

**Lordosis:** Es la curvatura fisiológica de la columna en la región cervical (a nivel de la nuca, entre la cabeza y la parte superior de la espalda) o lumbar (parte inferior de la espalda, aquella cercana al cóccix).

**Modalidad:** Se denomina modalidad a toda técnica de exploración realizada con un equipo de adquisición de datos susceptibles de ser utilizados para realizar un diagnóstico.

**Negatoscopio:** Dispositivo diseñado para la observación directa de los estudios de imagenología impresos en placas radiográficas, el cual consta de una fuente de luz que produce una intensidad de iluminación uniforme.

**Ortopedia:** Especialidad médica que estudia las deformaciones del cuerpo humano y su corrección por medios fisioterapéuticos, quirúrgicos o protésicos.

**Osteosíntesis:** La osteosíntesis es un tratamiento quirúrgico de fracturas, en el que éstas son reducidas y fijadas en forma estable. Existe la osteosíntesis cerrada donde la fijación es externa y la osteosíntesis abierta donde la fijación es interna. En esta última se utiliza la implantación de diferentes dispositivos tales como placas, clavos, tornillos, alambre, agujas y pines, entre otros.

**Osteotomía:** Intervención quirúrgica en la que se practican cortes en un hueso, de forma que el cirujano puede efectuar cambios en su posición.

**PACS** (Picture Archiving and Communication System): Sistemas de Transmisión, Almacenamiento y Visualización de Imágenes Médicas.

**Patrón arquitectónico:** Un patrón arquitectónico expresa un esquema estructural fundamental de la organización para un sistema de software, que consiste en subsistemas, sus responsabilidades e interrelaciones.

**Planificación Quirúrgica Ortopédica:** Serie de procedimientos previos a una operación que el cirujano ortopeda debe realizar para garantizar la eficacia de la misma.

**Plataforma:** Determinado software y/o hardware con el cual una aplicación puede ejecutarse o desarrollarse.

**Requisitos:** Capacidades, condiciones o cualidades que el sistema debe cumplir y tener.

**Servidor:** Computadora que forma parte de una red, brindando servicios a otras computadoras que reciben el nombre de clientes.

**Simulación:** Experimentación con un modelo que imita ciertos aspectos de la realidad. Esto permite trabajar en condiciones similares a las reales, pero con variables controladas y en un entorno que se asemeja al real pero que está creado o acondicionado artificialmente. Así, es posible corregir fallos antes de que la experiencia, efectivamente, se concrete en el plano de lo real.

**TCP/IP:** La sigla TCP/IP significa "Protocolo de control de transmisión/Protocolo de Internet". Proviene de los nombres de dos protocolos importantes del conjunto de protocolos, es decir, del protocolo TCP y del protocolo IP.

**Traumatología:** Especialidad médica dedicada al diagnóstico y tratamiento de las lesiones traumáticas que afectan al aparato locomotor.