

Universidad de las Ciencias Informáticas

Facultad 2



# Módulo de reportes para el Planificador Quirúrgico Ortopédico.

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

Autor: Rudibel Perdigón Llanes

Tutores: Ing. Yoel González Mesa  
Ing. Yania Junco López

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

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 17 días del mes de junio del año 2014

Autor

\_\_\_\_\_  
Rudibel Perdigón Llanes

Tutores:

\_\_\_\_\_  
Yoel González Mesa

\_\_\_\_\_  
Yania Junco López

## DATOS DE CONTACTO

### Tutores:

#### Tutor principal:

**Ing. Yoel González Mesa:** Ing. en Ciencias Informáticas graduado del curso 2008-2009 en la Universidad de Ciencias Informáticas. 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 AlasWebViewer (Visor web) y AlasTraumaView (Planificador Quirúrgico Ortopédico). Se desempeña como Jefe de Proyecto del Planificador Quirúrgico Ortopédico.

Correo electrónico: ygmesa@uci.cu

#### Co tutor

**Ing. Yanía Junco López:** 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). Se desempeña como analista del Proyecto del Planificador Quirúrgico Ortopédico.

Correo electrónico: yjlopez@uci.cu

## DEDICATORIA

*A mis padres, por su amor y apoyo incondicional.*

*A mi familia, por preocuparse tanto por mí y estar pendientes de mi bienestar.*

*A mi novia, por hacer de mí una mejor persona cada día.*

*A mis amigos, por su apoyo, confianza y por los momentos felices que me han regalado.*

## **AGRADECIMIENTOS**

*Agradezco a mis padres por sacrificarse tanto por ver mis aspiraciones realizadas. A mi novia por haber compartido conmigo tantas cosas buenas y por apoyarme en las malas. A mis amigos Fredy, Eddy, Ricardo y Julio por ser mis compañeros de aventuras. A mis compañeros de grupo que me han tolerado durante cinco años de sus vidas. A la universidad por haberme dado la oportunidad de ser uno de sus miembros. A los profesores que me instruyeron y formaron como profesional durante toda la carrera. A mis tutores, al Yoyo, a Yania y a todos aquellos que me brindaron su ayuda durante esta investigación. Muchas Gracias.*

## RESUMEN

Las cirugías ortopédicas se realizan con el objetivo de curar afecciones en el sistema osteomuscular. Estas son consideradas de alto riesgo, pues se realizan en áreas del cuerpo humano que se encargan del sostén y movimiento del mismo. Para garantizar el éxito de la cirugía, el especialista debe ejecutar un proceso de planificación preoperatoria lo más acertado posible. En los centros ortopédicos cubanos la planificación se realiza manualmente, así como los reportes que son emitidos durante la misma, lo que hace que este proceso esté propenso a errores humanos.

Actualmente en el departamento Software Médico Imagenológico, del Centro de Informática Médica, de la Universidad de las Ciencias Informáticas, se desarrolla el sistema nombrado Planificador Quirúrgico Ortopédico; que optimizará en tiempo, recursos y exactitud el proceso de planificación quirúrgica. Este sistema está integrado por varios módulos: cadera, rodilla, columna, pediatría, extremidades y trauma, los cuales poseen diversas herramientas que permiten la realización de mediciones ortopédicas específicas en cada uno de ellos. Estas mediciones constituyen la base del estudio previo a la intervención quirúrgica.

La presente investigación tiene como objetivo desarrollar un módulo de reportes e incorporarlo a dicho sistema de planificación. Este módulo se encargará de plasmar la información preoperatoria y postoperatoria recogida durante la planificación quirúrgica en un documento digital para su posterior utilización y archivo.

La aplicación fue desarrollada sobre la plataforma .NET en su versión 4.0, haciendo uso del lenguaje de programación C# 4.0. Se dispuso además de las tecnologías WPF, MEF y MVVM.

### **Palabras claves:**

Cirugía ortopédica, planificación quirúrgica, preoperatorio, postoperatorio y reporte.

# TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1. TÉRMINOS Y DEFINICIONES .....	6
1.2. EMISIÓN DE REPORTES EN SISTEMAS DE PLANIFICACIÓN QUIRÚRGICA ORTOPÉDICA .....	8
1.3. TECNOLOGÍAS, PATRONES Y LENGUAJES UTILIZADOS .....	17
1.4. MODELO DE CALIDAD, LENGUAJE Y NOTACIÓN DE MODELADO .....	21
1.5. HERRAMIENTAS UTILIZADAS PARA DESARROLLAR LA SOLUCIÓN PROPUESTA.....	24
1.6. CONCLUSIONES.....	25
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>26</b>
2.1. SISTEMA PROPUESTO .....	26
2.2. MODELO DE DOMINIO .....	26
2.3. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	28
2.4. DEFINICIÓN DE LOS ACTORES DEL SISTEMA .....	34
2.5. DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	35
2.6. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.....	35
2.7. CONCLUSIONES.....	37
<b>CAPÍTULO 3. ARQUITECTURA Y DISEÑO DEL SISTEMA</b> .....	<b>38</b>
3.1. DISEÑO.....	38
3.2. DESCRIPCIÓN DE LAS CLASES.....	44
3.3. MODELO ARQUITECTÓNICO .....	45
3.4. CONCLUSIONES.....	46
<b>CAPÍTULO 4. IMPLEMENTACIÓN</b> .....	<b>47</b>
4.1. DIAGRAMA DE COMPONENTES .....	47
4.2. DIAGRAMA DE DESPLIEGUE .....	49
4.3. ESTÁNDARES DE CODIFICACIÓN.....	51
4.4. CONCLUSIONES.....	52
<b>CONCLUSIONES</b> .....	<b>53</b>

<b>RECOMENDACIONES.....</b>	<b>54</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>55</b>
<b>BIBLIOGRAFÍA.....</b>	<b>59</b>
<b>ANEXOS .....</b>	<b>63</b>
ANEXO 1. PREFIJOS DE LOS REQUERIMIENTOS NO FUNCIONALES POR CATEGORÍAS.....	63
ANEXO 2. CASOS DE USO EXPANDIDOS.....	63
ANEXO 3. DESCRIPCIÓN DE CLASES. ....	70
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>76</b>

## **INTRODUCCIÓN**

Con la aparición de las Tecnologías de la Informática y las Comunicaciones (TIC) el ser humano se ha beneficiado en las múltiples esferas de su vida, hoy día prácticamente no existe nada que no pueda hacer un ordenador o en lo que estos no intervengan de alguna manera. Diversos sectores de la sociedad se han beneficiado de este desarrollo vertiginoso, siendo el sector de la salud y la medicina uno de ellos.

Hace más de un siglo que se descubrieron los rayos-x y con ellos surgieron diversas técnicas para la obtención de imágenes médicas, como son: la resonancia magnética, el ultrasonido, la fluoroscopia, entre otras. Estas imágenes constituyen la base del diagnóstico médico desde entonces.

Con la llegada de las TIC a la salud se ha logrado un avance extraordinario en diferentes procesos que se realizan en la misma. El proceso de estudio, interpretación y diagnóstico médico utilizando radiografías e imágenes de los pacientes es uno de estos, que con el apoyo de sistemas informáticos y electrónicos, ha elevado su desarrollo.

La realización de la planificación quirúrgica ortopédica de forma más rápida, exacta y eficiente ha sido posible gracias a herramientas como los sistemas de Cirugía Ortopédica Asistida por Computadora (CAOS, por sus siglas en inglés). Los cuales forman parte de los sistemas de Diagnóstico Asistido por Computadora (CAD, por sus siglas en inglés), estos sistemas desde su surgimiento y desarrollo han revolucionado los procesos de análisis y diagnóstico médico.

Desde el triunfo revolucionario, Cuba ha instaurado un sistema de salud único en su tipo, lo que ha convertido al país, indudablemente, en una potencia médica a nivel mundial; donde su Ministerio de Salud Pública (MINSAP) cuenta con un gran número de profesionales, el cual aumenta cada año gracias a la dedicación del gobierno por mejorar la medicina cubana. El desarrollo del sector de la salud en Cuba ha elevado paralelamente la calidad y expectativa de vida de la población.

La población cubana asciende a un número de 11 167 325 habitantes, cifra obtenida en el censo de población y vivienda realizado en el año 2012; de los cuales un 18,3 por ciento sobrepasa los 60 años y un 64,5 son adultos que avanzan hacia la vejez (1). Estas cifras evidencian de manera irrefutable la gran cantidad de ancianos que existirán en el país dentro de unos pocos años.

Debido a su avanzada edad, muchas son las enfermedades que afectan a este sector social, especialmente las afecciones óseas degenerativas, ejemplo de ellas es la osteoporosis, que afecta a un gran número de

personas a nivel mundial. En algunos sondeos realizados en Cuba, se obtuvo que prácticamente la mayoría de los ancianos que sufren fracturas y lesiones óseas, son víctimas de este padecimiento (2). Aunque cualquier persona independientemente de su edad, puede sufrir una lesión de este tipo.

En Cuba se prestan servicios de atención médica a lesiones osteomusculares en cualquier institución de salud, aunque existen instituciones especializadas en el tratamiento de aquellas afecciones que no puedan ser diagnosticadas en los centros de Atención Primaria de Salud. Ejemplo de ellas son instituciones como el Complejo Científico Ortopédico Internacional (CCOI) Frank País, que constituye el complejo hospitalario más extenso e integral del mundo dedicado a la cirugía ortopédica, traumatológica, reconstructiva y rehabilitadora, del sistema osteomioneuroarticular y el hospital Ortopédico Docente Fructuoso Rodríguez que brinda atención en la especialidad de Ortopedia y Traumatología. (3)

Para llevar a cabo el tratamiento de una afección ósea de un paciente que requiera intervención quirúrgica, es necesario realizar un proceso de planificación preoperatorio. El principal objetivo de este proceso es trazar la estrategia quirúrgica a seguir durante la intervención, la cual puede determinar el éxito de la operación. La planificación preoperatoria permite a los cirujanos ahorrar tiempo en el quirófano, debido a que solamente se llevará a un paciente a la sala de operaciones si se cuenta con todo el material necesario para la cirugía, dicho material se especifica durante el proceso de planificación. Además si se realiza de manera correcta, aumentan las probabilidades de éxito del proceso quirúrgico y garantiza una eficiente recuperación del paciente.

Durante la planificación preoperatoria se emite el reporte preoperatorio, que se utiliza como punto de partida para realizar la operación. Recoge parámetros esenciales de esta como son: la posición en la que se realizará la cirugía, los implementos médicos a utilizar, tipo de anestesia, entre otros.

Por otra parte, luego de ejecutada la cirugía, durante el estudio postoperatorio se emite otro tipo de reporte, en este caso el reporte postoperatorio. Su objetivo es clasificar el tipo de cirugía que se realizó, recoger los datos del cirujano que la ejecutó y sus recomendaciones y comentarios de la misma, así como la imagen del estudio postoperatorio del paciente. Todo esto, en aras de lograr una mejor rehabilitación de la persona operada.

Ambos reportes juegan un papel esencial durante el período quirúrgico y de recuperación al que es sometido el paciente. El primero permite la preparación del quirófano para la cirugía y posibilita que, si el especialista que realizó la planificación no la puede ejecutar, otro lleve a cabo la operación utilizando los

implementos requeridos con anterioridad. Por su parte, el reporte postoperatorio, constituye la base para la recuperación del paciente. Los elementos que contiene son el punto de partida para el proceso de rehabilitación, que será proporcionado por un especialista en esta rama. Si existiera alguna complicación o involución en la cirugía realizada, sería posible consultar al médico que la ejecutó ya que este reporte contiene sus datos.

La Universidad de las Ciencias Informáticas (UCI) tiene como tarea la informatización del país a través de la creación de soluciones informáticas para la gestión de diversos procesos. En el Centro de Informática Medica (CESIM), que es uno de los centros de desarrollo de *software* de esta institución y que se especializa en la creación de soluciones informáticas para la medicina, radica el departamento nombrado Software Médico Imagenológico (SWMI), donde se está desarrollando el sistema denominado Planificador Quirúrgico Ortopédico (PQO). Hasta el momento con el sistema se puede realizar el proceso de planificación en diferentes áreas del cuerpo, referentes a los 6 módulos que lo integran: cadera, rodilla, columna, pediatría, extremidades y trauma. Sin embargo aún no permite realizar la emisión de reportes preoperatorios y postoperatorios.

Hoy día la planificación quirúrgica en los centros hospitalarios cubanos se ejecuta de manera tradicional, o sea manualmente, por lo que puede estar propensa a errores humanos: de medición, de cálculos, entre otros. De igual manera ocurre con los reportes que se emiten como resultado de este proceso. Estos reportes que acompañarán al paciente antes de la operación y después de realizada la misma, para luego formar parte de su historia clínica, cuentan con información e imágenes de vital importancia referentes al proceso quirúrgico en cuestión. La realización manual de los mismos pudiera provocar que contengan errores de formato, sean poco legibles, puedan deteriorarse o extraviarse, pues son documentos y radiografías archivadas en papel. Además necesitan espacio real para su almacenamiento y de tiempo y personal para su posterior búsqueda y consulta. También para su confección se requiere de recursos como: marcadores, papel, radiografías, equipos para la realización y visualización de estas y tiempo del médico especialista.

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

Este problema se enmarca en el **objeto de estudio**: el proceso de emisión de reportes.

El objeto delimita el **campo de acción**: el proceso de emisión de reportes en la planificación quirúrgica ortopédica.

Para dar solución al problema se propone como **objetivo general**: desarrollar el módulo de emisión de reportes para el Planificador Quirúrgico Ortopédico.

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

- Elaboración de los presupuestos teóricos-metodológicos relacionados con el proceso de emisión de reportes.
- Asimilación de las tecnologías empleadas en el desarrollo del PQO.
- Caracterización del proceso de emisión de reportes preoperatorio y postoperatorio.
- Desarrollo del módulo de reportes para el PQO.

Para el desarrollo de la presente investigación se emplearon como métodos científicos los siguientes:

- Histórico Lógico: se empleó para realizar un análisis crítico valorativo de la información contenida en las fuentes bibliográficas consultadas durante la realización de esta investigación, con el fin de conocer el proceso de emisión de reportes durante la planificación quirúrgica ortopédica y el comportamiento del desarrollo de *software* entorno a este. Al inicio de la investigación se realiza un estudio del estado del arte de la problemática planteada, mediante el análisis de los principales sistemas de planificación quirúrgica ortopédica que permiten la emisión de reportes.
- Inductivo Deductivo: teniendo en cuenta la problemática de la investigación y el proceso de emisión de reportes durante la planificación ortopédica, se empleó este método para determinar aspectos específicos de este proceso y desarrollar la propuesta de solución.
- Modelación: este método se empleó con el objetivo de confeccionar modelos y diagramas que contribuyen al diseño y mejor visualización de las funcionalidades de la aplicación, con el fin de lograr una mejor comprensión y entendimiento de los procesos a desarrollar como parte de la investigación.

## **Estructuración del Contenido:**

**Capítulo 1:** Refiere lo concerniente al estudio del estado del arte del proceso de emisión de reportes, por parte de los principales sistemas especializados en la planificación quirúrgica ortopédica existentes a nivel

nacional e internacional. Se mencionan características de las tecnologías y herramientas empleadas en el desarrollo de la solución.

**Capítulo 2:** En este acápite se realiza una descripción de las características del sistema propuesto. Se especifican los requisitos funcionales y no funcionales de la aplicación. Se modela y describe el diagrama de casos de uso del sistema, así como los elementos del dominio, que constituyen el punto de partida para el diseño de la propuesta de solución.

**Capítulo 3:** Este capítulo contiene lo referente a la arquitectura y diseño de la solución. Se modelan los diagramas de clases de diseño de la aplicación y se describen cada una de estas clases. Se muestran además, las interacciones realizadas por cada una de las clases del diseño mediante los diagramas de secuencia. Finalmente, se expone la estructura de la aplicación a través de la arquitectura de la misma.

**Capítulo 4:** Aborda todo lo referente al proceso de implementación de la propuesta de solución. Se describen los componentes que integran la solución y la interacción que se establece entre ellos mediante el diagrama de componentes. Se modelan a través del diagrama de despliegue, los elementos utilizados para la puesta en funcionamiento de la aplicación presentada.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordan los principales conceptos relacionados con la investigación. Se realiza un análisis del estado del arte de los principales sistemas informáticos de planificación quirúrgica ortopédica a nivel nacional e internacional, especialmente de los reportes que estos emiten en dicho proceso.

Se realiza una caracterización de las tecnologías, patrones y lenguajes de programación utilizados para la implementación de la solución propuesta, así como de las herramientas, metodologías, modelo de calidad y notación de modelado empleados en el departamento SWMI.

### 1.1. Términos y Definiciones

Este acápite recoge aspectos significativos referentes a los reportes realizados en el proceso de planificación quirúrgica ortopédica. Se definen los sistemas CAD, CAOS y el estándar de comunicación entre imágenes DICOM, con el fin de brindar una mejor comprensión de los mismos.

#### 1.1.1. Reportes en la Planificación Quirúrgica Ortopédica

La planificación preoperatoria ortopédica consiste en una serie de pasos previos a una cirugía con el objetivo de enumerar todos los procedimientos y herramientas a utilizar en el quirófano. Este proceso permite identificar y clasificar una fractura de manera precisa, así como decidir las diversas acciones a realizar durante la cirugía. (4)

De la correcta realización de la planificación preoperatoria, depende el éxito de la cirugía y el posterior proceso de recuperación del paciente. Durante esta planificación se emite un reporte, denominado reporte preoperatorio y posterior a la intervención quirúrgica se emite otro informe, en este caso el reporte postoperatorio.

Un reporte es un documento digital, audiovisual o impreso, que presenta de manera resumida y clara, datos e información referente a determinado objetivo. Como parte de su contenido puede poseer imágenes, gráficas y tablas. Además puede clasificarse según la información que recoge, en varios tipos: divulgativo, persuasivo, investigativo, entre otros. (5)

El autor define como reporte preoperatorio aquel documento que se emite durante el proceso de planificación preoperatoria, que especifica las herramientas, implementos, objetos y técnicas requeridos para la realización de la cirugía. Este reporte garantiza la preparación previa por parte del personal

correspondiente, del material necesario para realizar la operación, todo esto a solicitud del médico especialista que la llevará a cabo. Permite a su vez, la realización del proceso quirúrgico por parte de otro médico en caso que el especialista que ejecutó la planificación, no esté disponible para realizarlo.

El autor de esta investigación, define como reporte postoperatorio aquel documento que se emite durante el estudio postoperatorio de un caso. Este tipo de reporte contiene la clasificación de la cirugía que se le realizó al paciente, los datos identificativos del especialista que la ejecutó y sus señalamientos y comentarios relacionados con la operación. El reporte postoperatorio, a diferencia del preoperatorio, que se le entrega al personal médico encargado de habilitar los medios que en él se solicitan, formará parte de la historia clínica del paciente y lo acompañará durante todo período de recuperación.

### *1.1.2. Sistemas de Diagnóstico Asistido por Computadora*

Los sistemas CAD, tienen como objetivo principal asistir a los especialistas en la interpretación de imágenes médicas, brindándole a estos una segunda opinión del análisis realizado a las mismas. Estos sistemas han demostrado que son capaces de apoyar en la realización de un diagnóstico preciso a los especialistas; aligeran la carga de trabajo de estos, lo que reduce la posibilidad de realizar un mal diagnóstico, debido al agotamiento de los médicos por el análisis de grandes cantidades de datos. (6)

Se han desarrollado sistemas CAD para la detección y caracterización de diferentes tipos de lesiones en las imágenes médicas, incluyendo radiología convencional, tomografía computarizada, resonancia magnética y ultrasonido. Actualmente su uso se encuentra generalizado en la detección de nódulos cancerígenos. Los órganos en los que se centra su desarrollo son mamas, tórax, cerebro, colon, hígado, riñón, sistema vascular y esquelético; pudiese aplicarse sobre extremidades, corazón o en otros tipo de imagen como la funcional o la angiografía. (7)

Estos sistemas hacen posible resaltar áreas específicas de las imágenes médicas analizadas, facilitando la visualización de estas, mediante la aplicación de algoritmos que aumentan el brillo y contraste, así como con operaciones geométricas y aritméticas que se ejecutan solo sobre imágenes en formato digital. Todo esto permite realizar un mejor análisis y lectura del caso de estudio, en menor período de tiempo.

### *1.1.3. Sistemas de Cirugía Ortopédica Asistida por Computadora*

Los sistemas CAOS, permiten asistir al cirujano en la planificación preoperatoria de cirugías ortopédicas. Son utilizados ampliamente en la planificación de cirugías en fracturas de huesos y posibilitan la realización

de diversos estudios y prácticas clínicas en áreas como: cadera, columna y rodilla. Su utilización no se limita solo a cirugías, también se emplean en traumatología y simulaciones.

Los CAOS surgen debido a que la planificación preoperatoria se realizaba de manera manual, lo que implicaba el consumo considerable de tiempo del especialista y de recursos materiales para su ejecución.

Estos sistemas necesitan de una infraestructura hospitalaria adecuada y de equipos digitales de rayos-x para la adquisición de las imágenes médicas de los pacientes y su posterior manejo. Además, los sistemas CAOS pueden contener un repositorio de los implantes a utilizar para una reducción de fractura y así disminuir su tiempo de búsqueda. (4) Contribuyen a optimizar el proceso de planificación quirúrgica preoperatoria en lo que a tiempo y exactitud en su realización, se refiere.

#### 1.1.4. Estándar DICOM

El estándar Imágenes y Comunicaciones Digitales en Medicina (DICOM, por sus siglas en inglés) es el mecanismo de codificación, almacenamiento y transmisión de imágenes aceptado universalmente por la comunidad médica. La cabecera de este formato, extremadamente rica, permite almacenar información sobre el paciente, las condiciones en las que se tomó la imagen y el formato interno de esta. El estándar DICOM, no es más que la versión 3.0 del estándar ACR-NEMA desarrollado en 1985 por un comité conjunto de especialistas de la *American College of Radiology (ACR)* y a la *National Electrical Manufacturers Association (NEMA)*.

DICOM no es solo un formato de imágenes médicas, es un estándar completo que cubre todas las necesidades de un PACS: almacenamiento, transmisión, comunicaciones en general e impresión. De esta forma, se integran todas las máquinas que forman un PACS, desde los equipos médicos encargados de la obtención de imágenes hasta los clientes, usados por el personal clínico para visualizarlas. (8)

La importancia de este estándar reside en que garantiza la interoperabilidad de los equipos que manejan imágenes médicas, independientemente de su fabricante. Estas imágenes permiten la realización de un diagnóstico más efectivo y acertado de los pacientes a los que les fueron tomadas.

## 1.2. Emisión de reportes en sistemas de planificación quirúrgica ortopédica

Este epígrafe muestra una caracterización de los principales sistemas de planificación quirúrgica ortopédica internacionales que emiten reportes y los obstáculos que imposibilitan su utilización en Cuba. Además

presenta las características fundamentales que posee el sistema cubano PQO, como alternativa para realizar la planificación quirúrgica ortopédica digital, en los centros ortopédicos cubanos.

### 1.2.1. *Sistemas Internacionales*

#### **TraumaCad**

Proporciona a los cirujanos ortopédicos herramientas digitales para realizar la planificación preoperatoria y permite la simulación de los resultados esperados, antes de la cirugía.

La aplicación incluye una extensa biblioteca de plantillas digitales y ofrece un conjunto completo de asistentes y herramientas de medición. Componentes del TraumaCad son los módulos referentes a deformidad de cadera y rodilla, pediatría, extremidades superiores, columna vertebral, piernas y tobillo, así como el módulo de planificación de trauma.

Utilizando imágenes digitales en pantalla, puede realizar mediciones, fijar las prótesis, simular osteotomías, y visualizar la reducción de la fractura. Soporta imágenes DICOM, puede importar imágenes JPG, imágenes escaneadas y de cámaras digitales. Este sistema se puede integrar al servidor PACS o puede ser ejecutado de manera independiente.

En su versión 2.4 posee nuevas herramientas para los módulos de rodilla y de columna. El primero contiene nuevos asistentes para realizar la medición preoperatoria de resección de la línea anteroposterior de manera más rápida y la resección larga de la pierna, así como la evaluación postoperatoria de los resultados. El segundo posee nuevas y mejoradas mediciones, como es el ángulo de clavícula, para una rápida y precisa planificación preoperatoria de la columna utilizando imágenes anteroposteriores y laterales.

Genera de manera automática un reporte preoperatorio que recoge imágenes del paciente, información de los implantes realizados, las mediciones que se tomaron y los comentarios del médico especialista. Este reporte puede ser impreso, salvado localmente o enviado al PACS. (9)

La desventaja principal que posee este sistema CAOS, es que opera solo sobre la plataforma Windows XP, que es un sistema operativo que actualmente está discontinuado. Debido a la aparición de versiones superiores del sistema operativo Windows, esta limitante coloca al TraumaCad a pesar de sus amplias funcionalidades en el área de las cirugías ortopédicas, por detrás del resto de los sistemas CAOS existentes en el mundo.

## TraumaPlan

Es un sistema CAOS, para la planificación preoperatoria de fracturas de miembros inferiores, con el cual el cirujano puede realizar la planificación de forma digital y en un corto período de tiempo. TraumaPlan consiste en una serie de etapas (siete en total) con funcionalidades bien definidas para el desarrollo de la planificación.

Estas etapas son: adquisición, calibración y mejoramiento de la imagen, segmentación y ensamblaje de la fractura (en esta etapa se realizan las mediciones y las anotaciones), colocación y deformación de los implantes y generación del reporte.

El reporte generado por TraumaPlan (ver Fig.1) recoge datos como: el tipo de implante colocado, mediciones de la fractura y anotaciones referentes a la misma, estas mediciones son realizadas por los tres tipos de medida que soporta el sistema: regla, círculo y ángulo. El reporte contiene además los procedimientos quirúrgicos que han de llevarse a cabo, todo esto se recoge en un cuadro de texto que se plasma sobre la imagen del paciente. (4)

En ocasiones la imagen sobre la que se encuentra el texto del reporte recoge un área muy grande de la anatomía del paciente, lo que dificulta la visualización de dicha información, debido a que no se logra el contraste adecuado entre el texto del reporte y el contenido de la imagen. Esta desventaja de los reportes generados por el TraumaPlan, hace que en ocasiones, los especialistas requieran de la utilización de otro sistema CAOS para la emisión de reportes como parte del proceso de planificación preoperatoria, o en el peor de los casos, los confeccionen manualmente.



**Fig.1** Reporte preoperatorio realizado por TraumaPlan.

### OrthoView™

Se utiliza para la creación de planes preoperatorios de manera rápida y fácil a partir de las imágenes de rayos x digitales. OrthoView facilita la planificación digital y plantillas para la artroplastia y revisiones, el trauma, la corrección de la deformidad de miembros, la osteotomía y la evaluación de la columna vertebral. Es elegido por los hospitales de todo el mundo para realizar la radiología libre de película<sup>1</sup>, más de 1850 instituciones lo utilizan.

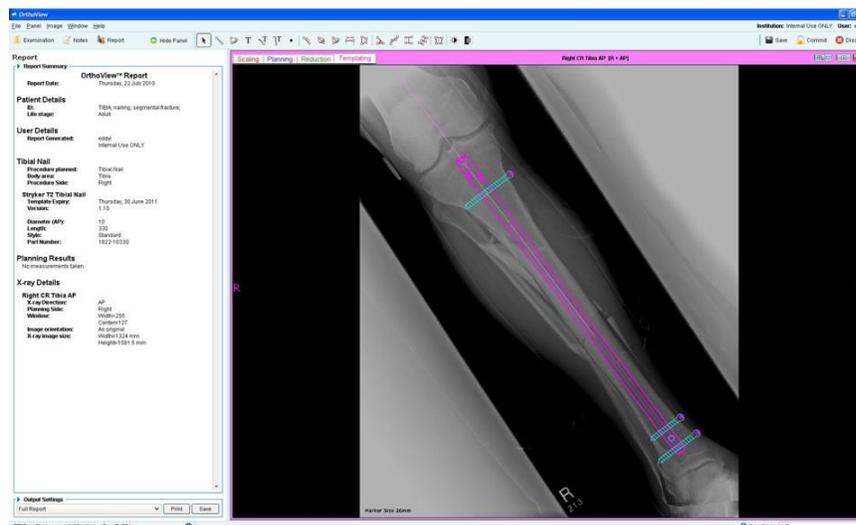
La biblioteca digital de plantillas de OrthoView incluye el reemplazo de articulaciones, tratamiento de fracturas y los implantes de osteotomía. Posee más de 1600 prótesis diferentes de más de 60 fabricantes de ortopedia. Además permite la integración con el PACS y trabajar con las imágenes recogidas en este, reajustar su tamaño para las mediciones de las prótesis y salvarlas nuevamente en el PACS o de manera local.

El sistema contiene cinco módulos referentes a las especialidades clínicas en las que permite realizar el proceso de planificación: Reemplazo de Articulación, Corrección de la Deformidad del Miembro, Evaluación Pediátrica, Evaluación de Médula y Tratamiento de Fractura. Este último genera un reporte (ver Fig.2) que contiene aspectos del plan preoperatorio, notas adicionales e imágenes referentes a la fractura. (10)

El OrthoView solo permite la emisión de reportes preoperatorios en el módulo de Tratamiento de Fractura, el resto de sus módulos no cuentan con dicha funcionalidad. Esta característica provoca que no sea posible conservar la información resultante del proceso de planificación preoperatoria, realizado en cada uno de los módulos restantes que integran el sistema.

---

<sup>1</sup> Conjunto de técnicas para obtener imágenes radiológicas escaneadas en formato digital.



*Fig.2 Reporte preoperatorio del OrthoView.*

## **Merge OrthoCase™**

Es un sistema elaborado por la compañía norteamericana *Merge Healthcare*, que permite a los cirujanos ortopédicos crear planes preoperatorios digitales desde cualquier estación de trabajo de un hospital, basado en la web, con una amplia biblioteca de plantillas digitales y herramientas de la especialidad de ortopedia.

Reemplaza las imágenes de rayos x, en imágenes 2D con plantillas digitales, logrando un proceso de planificación más rápido y eficiente. Accede a una amplia biblioteca de plantillas digitales y herramientas de la especialidad de ortopedia, posibilitando la creación de planes preoperatorios detallados.

Detecta puntos de referencia anatómicos aplicables y realiza mediciones adaptadas especialmente para la planificación de reemplazos totales de cadera, automatizando el proceso de planificación preoperatoria. Posee herramientas básicas para las mediciones y manipulación de imágenes ortopédicas, además de otras avanzadas, tales como el ángulo de giro, ángulo cóncavo, herramienta de esquema y muchos otros.

Permite la creación de reportes preoperatorios que incluyen imágenes, propiedades de implantes y mediciones ortopédicas. Puede integrarse eficientemente con un PACS, en este caso con el servidor ofertado por la misma compañía, el Merge OrthoPACS (11)

La utilización de este sistema requiere para el intercambio de imágenes y reporte médicos con otros equipos, la compra e instalación del servidor PACS desarrollado por la misma compañía, lo que otorga un

valor agregado al empleo de este sistema CAOS, siendo su costo una de las principales desventajas para su utilización. Además, al ser un producto desarrollado por una compañía estadounidense, imposibilita su adquisición para aquellos países que como Cuba, no tienen relaciones económicas con esta potencia.

### **View Pro-X Ortho**

Es un sistema desarrollado por la empresa holandesa *ROGAN Delf*, que se utiliza para la planificación preoperatoria de cirugías de cadera. Permite el manejo de los estudios ortopédicos de manera más fácil, apoyándose en la web; lo que posibilita la realización de la planificación en un departamento del hospital y su posterior visualización en cualquier otra ubicación dentro de este.

Posee herramientas de selección y monitoreo de implantes, de lectura de diferentes tipos de estudios ortopédicos como tomografías computarizadas y resonancias magnéticas. Cuenta también con un asistente que permite encuadrar la zona de la imagen que se desea analizar y realizar las mediciones deseadas, así como con teclas de acceso rápido. Soporta un estándar DICOM conforme a la estructura de informes, en un flujo de trabajo compatible con IHE. Emplea plantillas de implantes elaboradas por distintos productores como son: *Smith & Nephew, Stryker, Biomet, Synthes*, por solo nombrar algunos.

Este sistema permite realizar un reporte de artroplastia (ver Fig.3) con la información referente a las distintas mediciones realizadas y a la diferencia de longitud de las piernas. Esta última compuesta por: la diferencia total, componente acetabular de longitud de la pierna, componente femoral de longitud de la pierna y distancia de la línea de referencia medial. (12) Se integra perfectamente con un PACS.

Este sistema, como se limita únicamente a la planificación preoperatoria del área de cadera, permite solo la emisión de reportes preoperatorios referentes a dicha parte del cuerpo humano. Al ser un sistema que se especializa solo una región de la anatomía humana, y no en la planificación quirúrgica ortopédica de forma general, su utilización no es factible en aquellos centros ortopédicos que deben de atender lesiones óseas en cualquier área del cuerpo.



*Fig.3 Reporte de artroplastia del VPX Ortho.*

## **OrthoWorks ProPlanner**

OrthoWorks ProPlanner es un sistema de cirugía ortopédica asistido por computadora que permite la utilización de plantillas, su archivo y distribución, combinando la manipulación de imágenes y la automatización con una arquitectura diseñada para la integración y personalización. Ofrece una solución digital completa para artroplastia, trauma, corrección de deformidad, rodilla y columna.

Este sistema brinda la posibilidad de realizar diversas mediciones en las imágenes adquiridas, pertenecientes a las diferentes regiones del cuerpo humano mencionadas anteriormente. Incluso permite conectar múltiples imágenes y visualizarlas como si fueran una sola, opción muy eficaz para aquellas áreas del cuerpo donde no es posible capturar su anatomía en una sola imagen.

Mediante la utilización de plantillas es capaz de sugerir las posibles posiciones de los implantes y las mediciones que se pudiesen realizar en la imagen. Los especialistas, quienes tomarán la decisión final, pueden aceptar, modificar o rechazar estas sugerencias. Soporta plantillas de la mayoría de los principales fabricantes y posee una amplia biblioteca de implantes 2D, dedicada a las funcionalidades de sus diferentes módulos.

ProPlanner rellena automáticamente un reporte preoperatorio con imágenes claves del estudio preoperatorio, mediciones, implantes a utilizar y datos demográficos del paciente. Los médicos tienen la

opción de añadir notas técnicas a este reporte. Puede ser perfectamente integrado en las soluciones PACS existentes, utilizando la plataforma de integración C4<sup>2</sup> de Cedara. (13)

El hecho de necesitar una plataforma adicional para su integración con el servidor PACS, aunque mediante su empleo se mejore el flujo de trabajo entre la aplicación y el servidor, hace que este sistema no sea el más idóneo a utilizarse en aquellos centros hospitalarios que carecen de amplios recursos monetarios, ya que el empleo de esta plataforma eleva el costo para la utilización de este sistema. Además si ocurre un mal funcionamiento de esta, se pierde la conexión entre el sistema y la información contenida en el servidor PACS.

### **PreOPlan**

Es un sistema para la planificación quirúrgica ortopédica, utilizado en traumatología para la planificación quirúrgica en el tratamiento de fracturas y para realizar la planificación de osteotomía de rodilla. Surgió como resultado de un proyecto conjunto de *Siemens Healthcare* (Erlangen, Alemania) y *Synthes* (Solothurn, Suiza), líder mundial en implantes óseos.

Permite el análisis de radiografías en 2D, que pueden estar contenidas en dispositivos extraíbles o en un servidor PACS, con el cual se integra perfectamente. Posibilita la realización de mediciones y ensamblaje de la fractura ósea analizada. PreOPlan utiliza una serie de implantes 3D en la reconstrucción ósea durante la planificación, de los cuales brinda sus especificaciones y sugerencias para su uso. Estos implantes solo pueden ser proporcionados por *Synthes*, una de sus empresas creadoras y están contenidos en una base de datos integrada al sistema.

El software genera un reporte que ayuda al personal a preparar el quirófano con los implantes seleccionados. Este reporte contiene las imágenes de la planificación y el cirujano las puede utilizar como orientación durante la cirugía, así como una lista de materiales a ser preparados antes del procedimiento. Este reporte puede ser exportado en formato PDF, ser guardado como un archivo DICOM o ser alojado en el servidor PACS. (14)

---

<sup>2</sup> Esta plataforma C4 (Centro Clínico de Control de Cedara) es un marco de trabajo que permite la integración de aplicaciones clínicas con un impacto mínimo en el host (por ejemplo, PACS) del software producto. Un host habilitado para C4 y la aplicación clínica, proporcionan un escritorio integrado con procedimiento centralizado del flujo de trabajo.

El empleo del PreOPlan garantiza un proceso de planificación quirúrgico ortopédico más ágil y exacto. Sin embargo los centros ortopédicos que lo utilizan, requieren exactamente de los implantes que elabora la compañía *Syntes*, pues es esta compañía, quien provee los modelos de implantes con los que trabaja el sistema. Razón que obstaculiza el uso del PreOPlan a aquellos hospitales que no tienen contratos con dicha compañía y que no pueden adquirir sus implantes óseos.

El análisis y estudio de los principales sistemas de planificación quirúrgica internacionales, principalmente de las funcionalidades referentes a la emisión de reportes, posibilitó una mayor comprensión de las mismas, sentando las bases para el desarrollo del módulo de emisión de reportes para el Planificador Quirúrgico Ortopédico.

Los sistemas analizados anteriormente, responden de manera parcial a los problemas referentes al proceso de planificación quirúrgica ortopédica y la emisión de reportes como parte de este. Sin embargo no constituyen una solución viable para Cuba debido a que son *software* privativos y de altos costos. Además ninguno de ellos permite generar reportes postoperatorios, algunos requieren de tecnologías que elevan el precio del producto o su utilización significaría depender de los implementos para cirugías, ofertados por sus creadores. Otros simplemente, son de muy difícil adquisición debido a que proceden de EEUU, y a la existencia de leyes norteamericana que prohíben la comercialización de estos productos por parte de Cuba.

### 1.2.2. *Sistemas Nacionales*

En Cuba no existe actualmente un sistema que permita realizar la planificación quirúrgica ortopédica de manera digital, ni la emisión de reportes como parte de esta. Sin embargo la UCI se encuentra desarrollando un *software* cuyo objetivo es realizar dicha planificación, lo que facilitará el trabajo de los especialistas ortopédicos.

El PQO está integrado por diferentes módulos, que se han desarrollado de forma independiente: rodilla, cadera, columna, extremidades, pediatría y trauma. Cada uno de ellos cuenta con las herramientas necesarias para la realización de mediciones específicas a cada área del cuerpo humano a la que pertenecen. Este sistema permite el manejo de las imágenes médicas de los pacientes, resaltando en ellas las áreas que el especialista determine para poder realizar una planificación más eficiente. Esto es posible gracias a herramientas como los filtros, las paletas de colores y otras que aumentan el tamaño de la imagen, su brillo y contraste.

El PQO permite la integración inter-hospitalaria utilizando el estándar DICOM y es capaz de obtener información de los servidores PACS.

El sistema, aunque constituye una solución cubana a algunas de las deficiencias existentes en los procesos de planificación quirúrgica en los centros ortopédicos del país, no posee un módulo que permita la emisión de reportes preoperatorio y postoperatorio. El completo desarrollo del sistema y su despliegue supondrán una mejora en la calidad de la realización de las cirugías ortopédicas, así como un ahorro considerable de tiempo y una mayor exactitud en el proceso de planificación y de emisión de reportes.

### 1.3. Tecnologías, patrones y lenguajes utilizados

Este epígrafe abarca las tecnologías, patrones y lenguaje de programación utilizados para el desarrollo de la solución propuesta. Brinda una descripción de las características más representativas de cada uno de ellos, evidenciando las razones de su utilización.

#### 1.3.1. *Plataforma de desarrollo.NET*

Es un entorno de desarrollo que permite la realización de aplicaciones mediante varios lenguajes de programación (C#, C++, Visual Basic) y combina además varias tecnologías en función de lo que se desee lograr: ASP.NET para aplicaciones web, ADO.NET para el acceso a datos, Windows Communication Foundation para aplicaciones orientadas a servicios, Windows Presentation Foundation para mejores interfaces gráficas de usuario y Windows Workflow para aplicaciones habilitadas en el control del flujo de trabajo.

Consta de dos componentes principales: Common Language Runtime, que es el motor de ejecución que controla las aplicaciones en ejecución, y una biblioteca de clases, que proporciona código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones.

Es una tecnología que soporta el desarrollo y ejecución de la próxima generación de aplicaciones y servicios WEB XML. (15) Para el desarrollo de la propuesta de solución se utilizó este entorno de desarrollo, debido a que soporta el resto de las tecnologías, librerías, herramientas y lenguaje de programación empleados.

#### 1.3.2. *Windows Presentation Foundation*

Es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows, tomando características de aplicaciones Windows y de aplicaciones web. Puede crear una amplia gama de aplicaciones independientes u hospedadas en navegador.

El núcleo de Windows Presentation Foundation (WPF) es un motor de representación basado en vectores e independiente de la resolución en que se crea, para sacar partido del *hardware* de gráficos moderno. WPF extiende el núcleo con un conjunto completo de características de desarrollo de aplicaciones que incluye lenguaje XAML, controles, enlace de datos, diseño, gráficos 2D y 3D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en el entorno de desarrollo .NET, de modo que es posible compilar aplicaciones que incorporen otros elementos de la biblioteca de clases de dicho entorno. (16)

Tres aspectos destacan por su importancia:

1. Una plataforma unificada para interfaces de usuario modernas.
2. La posibilidad para desarrolladores y diseñadores de trabajar conjuntamente.
3. Una tecnología común para interfaces de usuario de Windows y explorador web.

El empleo de esta tecnología permitió el desarrollo de las interfaces gráficas del módulo de reportes, mediante la utilización de la librería Fluent y específicamente del recurso Ribbon, se logró alojar las herramientas y funcionalidades del módulo, en la interfaz de usuario. La confección de interfaces visuales más atractivas, la capacidad de utilizar el patrón Model View ViewModel (MVVM) y el hecho de lograr uniformidad con el sistema al que se integrará la propuesta de solución, son las razones fundamentales para la utilización de esta tecnología.

### 1.3.3. *Managed Extensibility Framework*

Managed Extensibility Framework (MEF) es una librería del entorno de desarrollo .NET 4 y de Silverlight 4 que simplifica el diseño de sistemas que se pueden componer y que terceros pueden ampliar después de su implementación. MEF abre las aplicaciones ya existentes, al permitir que desarrolladores de aplicaciones, autores de marcos y extensores de terceros, introduzcan poco a poco nuevas funciones. (17)

Contiene una serie de elementos y conceptos que lo componen y a su vez describen su funcionamiento:

Elementos o partes que se pueden componer: son partes que brindan servicios a otras partes o consumen los servicios proporcionados por alguna otra parte. Estos elementos pueden estar ubicados dentro o fuera de la aplicación, para la biblioteca MEF esto no marca diferencia alguna.

Exportación: servicio que proporciona una parte. Por ejemplo, una parte puede exportar un registrador o, en el caso de Visual Studio, una extensión del editor. Las partes pueden proporcionar varias exportaciones pero generalmente, solo brindan una.

Importación: servicio que una parte utiliza. Las partes pueden importar servicios únicos, como el registrador, o importar varios servicios, como una extensión de editor.

Contratos: un contrato es un identificador para una exportación o importación. Un exportador especifica un contrato que proporciona y un importador especifica el contrato que necesita.

Composición: las partes las compone MEF, el cual les crea una instancia y luego satisface exportadores con importadores.

La utilización de esta librería permitió el desarrollo de manera independiente del módulo de reportes y su posterior integración con el PQO. Garantizando que se pueda realizar cualquier cambio en las funcionalidades de la propuesta de solución sin modificar el funcionamiento del sistema PQO.

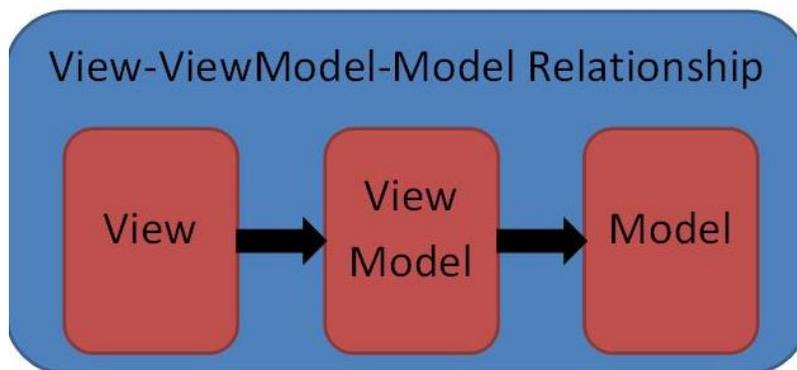
#### *1.3.4. Patrón Model View ViewModel*

Model-View-ViewModel (MVVM) divide el código de la interfaz de usuario en 3 partes conceptuales: Modelo, Vista y Vista del Modelo.

Modelo (Model) es un conjunto de clases que representan los datos procedentes de los servicios o de la base de datos.

La vista (View) es el código correspondiente a la representación visual de los datos, de la forma en que se ve y se interactúa con el usuario.

VistaModelo (ViewModel) sirve como enlace entre la Vista y el Modelo (ver Fig.4). Envuelve los datos del modelo y hace que sea fácil para ser presentados y modificados por la vista. VistaModelo también controla de la vista, las interacciones con el resto de la aplicación incluyendo con cualquier otra vista.



**Fig.4** Relación entre la Vista, VistaModelo y el Modelo.

Este patrón se refiere a las clases del Modelo, pero estas a su vez deben ser independientes del primero. La Vista debe ser consciente de la VistaModelo, la cual debe ser independiente de la otra; ambas se comunican a través de enlace de datos (*bindings*), comandos (*command*) y eventos (*events*). (18)

La posibilidad brindada por este patrón, de separar de manera eficiente el código de la interfaz de usuario de la aplicación, del modelo o lógica del negocio de la misma, permitió el desarrollo de manera paralela e independiente de ambas partes, agilizando el tiempo de confección del módulo de reportes para el PQO.

#### 1.3.5. C# 4.0

C# es un lenguaje diseñado por Microsoft para su entorno de desarrollo .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, este último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Es el lenguaje nativo de la plataforma de desarrollo .NET ya que es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que carece de elementos heredados innecesarios en dicha plataforma. Es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. (19)

Este lenguaje fue lanzado en 2002 y ha venido mejorando conjuntamente con las librerías del entorno de desarrollo .NET, con el objetivo de permitir que los programadores escriban código de una manera más sencilla y cada vez más fácil de mantener. En su versión 4.0 posee una serie de características que permiten la realización de tareas comunes que implican a tipos genéricos, interoperabilidad heredada y el trabajo con modelos de objeto dinámico mucho más sencillos. (20)

Con el objetivo de lograr la compatibilidad entre módulo de reportes y el PQO y por ser el lenguaje natural del entorno de desarrollo .NET, se empleó C# como lenguaje de programación.

### 1.4. Modelo de calidad, lenguaje y notación de modelado

Este epígrafe brinda información referente a las características del modelo de calidad, lenguaje y notación del modelado seguidos en el desarrollo del módulo de reportes. Muestra elementos que argumentan las razones de la utilización de los mismos en la propuesta de solución.

#### 1.4.1. *Modelo de Capacidad y Madurez Integrado.*

El Modelo de Capacidad y Madurez Integrado (CMMI, por sus siglas en inglés) es un modelo de evaluación de los procesos de una organización. Fue desarrollado inicialmente para los procesos relativos al *software* por la Universidad Carnegie -Mellon para el Instituto de Ingeniería de Software. (21)

El CMMI permite aproximarse a la mejora de procesos y a las evaluaciones usando dos representaciones diferentes: continua y por etapas.

La representación continua permite a una organización seleccionar un área de proceso (o un grupo de áreas de proceso) y mejorar los procesos relacionados con ésta. Esta representación utiliza unos niveles de capacidad para caracterizar la mejora concerniente a un área de proceso individual.

La representación por etapas utiliza conjuntos predefinidos de áreas de proceso para definir un camino de mejora para una organización. Este camino de mejora se caracteriza por diversos niveles de madurez. Cada nivel de madurez proporciona un conjunto de áreas de proceso que caracterizan diferentes comportamientos organizativos. (22)

Sus cinco niveles de madurez son: 1. Inicial: donde los resultados son impredecibles y pobremente controlados, 2. Administrado: en este nivel el proceso es reactivo y se caracteriza por su aplicación a proyectos, 3. Definido: es un nivel donde el proceso es conocido y bien entendido, 4. Administrado Cuantitativamente: los resultados son obtenidos a través de un proceso siempre medido y controlado y 5. Optimizado: donde el proceso se enfoca en la mejora constante. Cada uno de estos niveles indica el grado de planificación y gestión en el proceso de desarrollo de un *software* o producto, que posee una organización, asegurando la calidad en la obtención del mismo.

El nivel 2 de la representación por etapas de CMMI, está integrado por 7 áreas de procesos: Gestión de Requisitos (REQM, por sus siglas en inglés), Planificación de Proyectos (PP, por sus siglas en inglés), Seguimiento y Control de Proyectos (PMC, por sus siglas en inglés), Acuerdos con Proveedores (SAM, por sus siglas en inglés), Medición y Análisis (MA, por sus siglas en inglés), Aseguramiento de la Calidad de Procesos y Productos (PPQA, por sus siglas en inglés) y Gestión de la Configuración (CM, por sus siglas en inglés). (23)

La UCI luego de desarrollar un proceso de mejoras, logró que algunos de sus centros de desarrollo certificaran el nivel 2 de CMMI, como es el caso de CESIM. Durante este proceso de mejoras en la universidad se definieron los libros: Libro de Proceso para la Administración de Requisitos (IPP-3510\_2009) donde se describen 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.

Con la certificación de este nivel, CESIM asegura que sus procesos de desarrollo de *software* sean realizados de manera planeada, documentada, monitoreada y controlada a nivel de departamentos. De esta manera, la administración del proyecto puede velar por la calidad y el estado de este mediante hitos de control preestablecidos. Lo que posibilita elevar la calidad del producto, la participación de la organización en el mercado y obtener una mayor satisfacción del cliente.

El empleo de este modelo de calidad permitió que el departamento SWMI, asegurara un mejor monitoreo, control de la documentación y calidad durante el desarrollo del módulo de reportes para el PQO.

### 1.4.2. Lenguaje de Modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje de modelado que sirve para el diseño completo de sistemas complejos, tanto el de los sistemas *software* como para la arquitectura *hardware* donde se ejecutan. Es independiente del lenguaje de implementación, de forma tal que los modelos realizados usando UML se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

Constituye además un método formal de modelado, teniendo como ventajas: un mayor rigor en la especificación, la realización de una verificación y validación del modelo realizado, que se puedan automatizar determinados procesos y que se genere código a partir de los modelos y a la inversa.

Sus principales objetivos son:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo pueda entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: posibilita a partir de modelos especificados, construir sistemas diseñados.
- Documentar: brinda la opción de que sus propios elementos gráficos sirvan como documentación del sistema desarrollado para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de *software*, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de *hardware*. (24)

Por sus potencialidades es utilizado como lenguaje de diseño gráfico con el fin de visualizar, especificar, construir y documentar cada uno de los artefactos generados durante el ciclo de vida de la propuesta de solución.

### 1.4.3. *Proceso Racional Unificado*

El Proceso Racional Unificado (RUP, por sus siglas en inglés) es un proceso de ingeniería de *software* que provee un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización. Su principal objetivo es asegurar la producción de *software* de alta calidad que satisfaga las necesidades de los usuarios finales dentro de un presupuesto y tiempo predecibles. Permite un marco de trabajo amplio, el cual puede ser adaptado tanto a empresas grandes como pequeñas y puede ser modificado para ajustarse a la forma de trabajo de una compañía.

Entre sus principales características se encuentran: es un proceso controlado que permite la gestión de requisitos y control de cambios, se centra en la arquitectura, es soportado por herramientas y soporta técnicas orientadas a objeto así como la utilización del lenguaje UML.

Su desarrollo se basa en cinco principios fundamentales: adaptar el proceso de desarrollo a la empresa u organización, balancear las prioridades de los requerimientos, mostrar los progresos alcanzados en cada iteración y realizar una retroalimentación con el fin de enfrentar los riesgos y redireccionar el proyecto en caso de ser necesario, elevar el nivel de abstracción y por último mantener un control de la calidad, control que será realizado durante todo el ciclo de vida del producto. (25)

RUP se utilizó por sus características para la creación de los artefactos que se obtuvieron durante las etapas: modelado de negocio, especificación de requisitos, análisis, diseño e implementación del proceso de desarrollo de la propuesta de solución. El empleo de esta metodología permitió establecer una correcta distribución de las tareas a realizar en el tiempo requerido, garantizando el cumplimiento de los períodos de entrega para revisión de la aplicación. Además, al ser un proceso dirigido por casos de uso, iterativo e incremental, garantizó velar por la calidad de la aplicación y de sus artefactos, en cada una de las iteraciones que fueron realizadas.

### 1.5. Herramientas utilizadas para desarrollar la solución propuesta

Para lograr la realización de cualquier sistema informático es necesario utilizar herramientas como: un entorno de desarrollo integrado para la confección del mismo, un controlador de versiones para garantizar la seguridad y la preservación del código durante su confección y un sistema de análisis y diseño UML para gestionar los artefactos de la aplicación. En esta sección se presentan las herramientas: Visual Studio 2010, Enterprise Architect 7.5 y Tortoise SVN 1.6.6, que por sus componentes y características son usadas para desarrollar la solución propuesta.

#### 1.5.1. *Visual Studio 2010*

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos este mismo Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), el cual habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes: C#, C++, Visual Basic, F# y JScript. Asimismo, Visual Studio utiliza las funciones del entorno de desarrollo .NET, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. (26)

Soporta además tecnologías como son: Windows Presentation Foundation (WPF), Windows Workflow (WF), Windows Communication Foundation (WCF) y ASP.NET. Lo que posibilita el desarrollo de diferentes tipos de aplicaciones, para diferentes fines y entornos de trabajo.

#### 1.5.2. *Enterprise Architect 7.5*

Enterprise Architect (EA, por sus siglas en inglés) es una plataforma de alto desempeño para el modelado, visualización y diseño, basada en el estándar UML 2.4.1. Ofrece trazabilidad completa desde mapas mentales, pasando por los requerimientos y hasta el diseño y la distribución del *software*, con el nivel de eficiencia, robustez, herramientas de colaboración y seguridad requeridas para sacar adelante proyectos altamente demandantes y de cualquier tamaño. (27)

Es una herramienta que permite: compilar, trazar y transformar modelos desde conceptos y reglas de negocio hasta código ejecutable en Lenguaje de Ejecución de Procesos de Negocio, colaborar y compartir grandes repositorios de empresas sobre múltiples sitios con un alto desempeño e integración con control de versiones. Posibilita también nivelar el uso de nuevas herramientas de scripting para hacer actualizaciones masivas y dinámicas. (28)

Su uso permite modelar, diseñar, trazar y analizar el flujo de desarrollo y funcionamiento de la propuesta de solución, a través de los diagramas, esquemas y reportes pertinentes.

### 1.5.3. Tortoise SVN 1.6.6

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones 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. (29) Garantizando la actualización y preservación permanente de estos ficheros.

## 1.6. Conclusiones

El estudio del proceso de emisión de reportes de los principales sistemas de planificación quirúrgica ortopédica del mundo, así como de sus características; demostró que no satisfacen totalmente las necesidades actuales existentes en este proceso y que son altamente costosos, por lo que el país no tiene acceso a ellos. Son estas las razones fundamentales que evidencian la necesidad de desarrollar un sistema cubano para la planificación quirúrgica ortopédica y que permita la emisión de reportes como parte de dicho proceso. En este capítulo se expusieron las características fundamentales de las tecnologías, lenguajes, metodología y herramientas definidas por el departamento SWMI y utilizadas para el desarrollo del módulo de reportes para el PQO, logrando de esta manera un mejor entendimiento de las razones de su utilización.

### CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se presenta la propuesta de solución a desarrollar. Durante esta sección se muestran a través del modelo de dominio, los procesos que intervienen en la emisión de reportes durante la planificación quirúrgica ortopédica. Se especifican además los requisitos funcionales y no funcionales del módulo de reportes y se exponen también de manera resumida, los casos de usos que han sido identificados durante el proceso de emisión de reportes.

#### 2.1. Sistema Propuesto

Partiendo del análisis realizado al proceso de emisión de reportes durante la planificación quirúrgica ortopédica ejecutado en los centros ortopédicos cubanos, y de la inexistencia de un módulo que permita la realización de estos en el sistema PQO, se propone el desarrollo de un módulo de reportes para dicho sistema. Este módulo tiene como finalidad facilitar y optimizar la emisión de reportes durante el proceso de planificación preoperatoria.

La emisión de reportes de manera automatizada trae como resultado una mejoría en la planificación quirúrgica ortopédica. Garantiza la realización de reportes con una mayor calidad, mucho más legibles y con la información mejor estructurada y detallada. Además, al ser sustituida la manera tradicional de realización de los reportes por la digital, se ahorran gran cantidad de materiales y tiempo de los especialistas en la confección de estos. También garantiza que su almacenamiento sea más seguro, prolongado y no requiera de espacio físico.

El módulo a desarrollar brindará entre sus principales funcionalidades la posibilidad de realizar dos tipos de reportes: preoperatorio y postoperatorio, así como exportar dichos reportes en formato PDF, RTF y HTML. Además tendrá la opción de incluir los comentarios o señalamientos del médico en ambos tipos de reportes.

La integración de este módulo con el sistema PQO se realizará a través del empleo de un mecanismo basado en extensiones, utilizando la librería MEF.

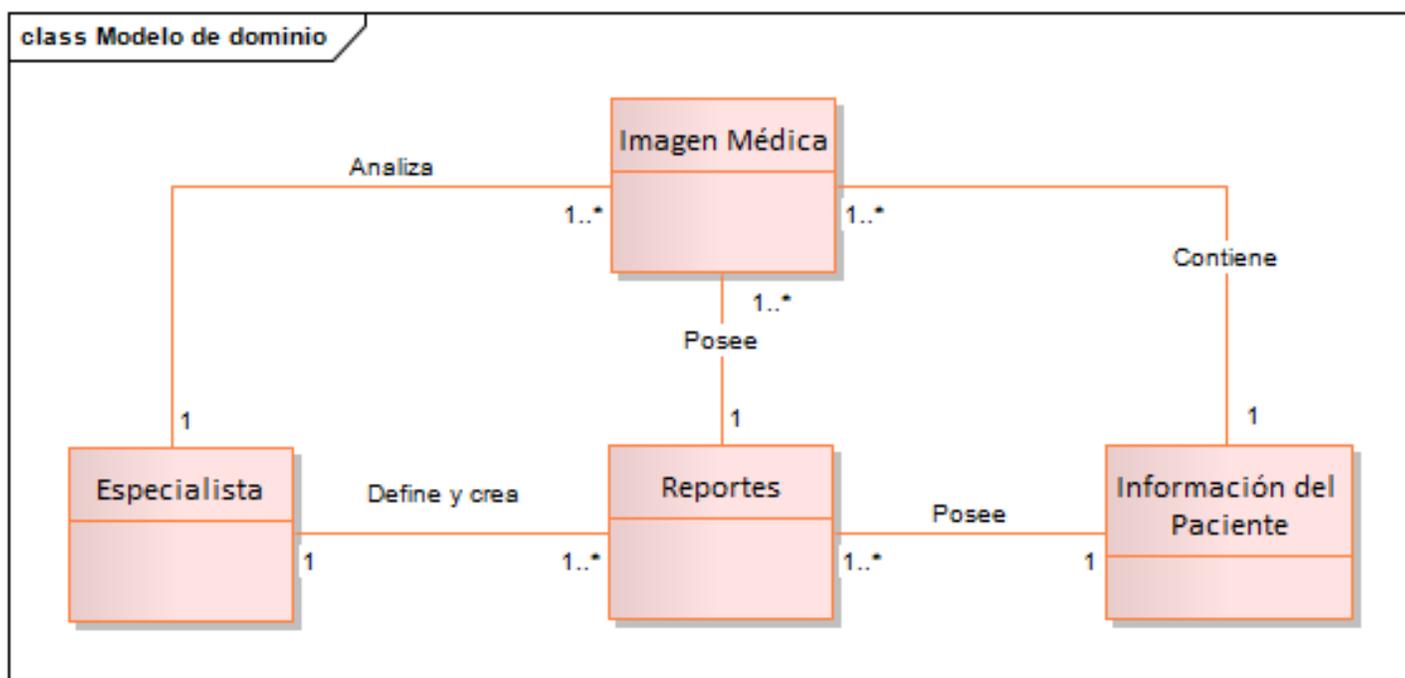
#### 2.2. Modelo de Dominio

El modelo de dominio o conceptual es utilizado por el analista como medio para comprender el sector de negocio al cual el sistema va a servir. Constituye la primera versión del sistema, el punto de partida para el

diseño del mismo, ya que en él se representan conceptos que semejan la realidad y que facilitan la realización de la programación orientada a objetos.

Este modelo permite mostrar de manera visual los principales conceptos que se manejan, ayudando a los usuarios, desarrolladores e interesados; a utilizar un vocabulario común para poder entender el contexto en que se desarrolla el sistema. Además contribuye a identificar personas, eventos, transacciones y objetos involucrados en el mismo.

El modelo de dominio se realiza cuando no se conocen con profundidad los objetivos planteados en el modelo de negocio. (30) Su descripción se ejecuta a través de diagramas de clase utilizando el lenguaje UML.



**Fig.5** Modelo de dominio.

Para la realización de un reporte, el especialista analiza las imágenes médicas del paciente y define el tipo y los datos del reporte a emitir: preoperatorio si es antes de la cirugía, con el objetivo de preparar el quirófano para ejecutar la misma y postoperatorio como resultado de la realización del estudio postquirúrgico. En el reporte se plasman los datos demográficos del paciente, los cuales son contenidos en la cabecera de la

imagen. Por otra parte, el especialista decide qué imagen del paciente plasmar en el reporte, según el estudio médico realizado.

### 2.3. Especificación de los requisitos de software

La Especificación de Requisitos de Software forma parte de la documentación asociada al *software* que se está desarrollando, por tanto debe definir correctamente todos los requerimientos del sistema, pero no más de los necesarios. Esta documentación no deberá describir ningún detalle de diseño, modo de implementación o gestión del proyecto, ya que los requisitos se deben describir de forma que el usuario pueda entenderlos. Al mismo tiempo, se da una mayor flexibilidad a los desarrolladores para la implementación. Si no se realiza una buena especificación de requisitos, los costos de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación. (31)

#### 2.3.1. *Requerimientos funcionales*

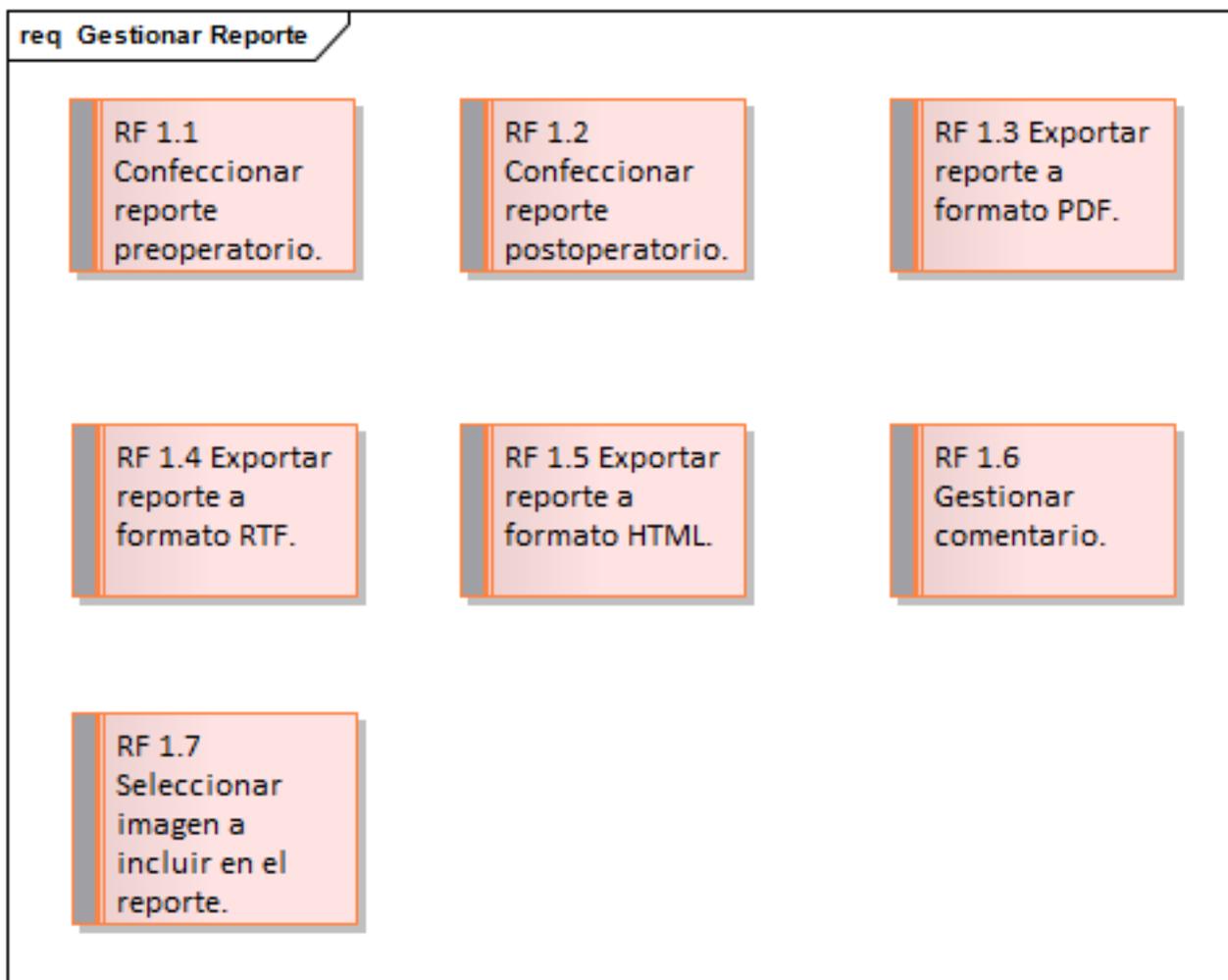
Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe de reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos los requerimientos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer. (32)

Requerimiento	Descripción
RF 1.1 Confeccionar reporte preoperatorio.	Permite especificar toda la información que contiene el reporte de planificación preoperatoria. Dentro de esta información se encuentra la posición del cuerpo durante la operación, el tipo de anestesia a utilizar, seleccionar exámenes complementarios, selecciona tipo de injerto óseo y especificar información adicional del procedimiento.
RF 1.2 Confeccionar reporte postoperatorio.	Esta funcionalidad permite confeccionar el reporte postoperatorio, con la clasificación de la cirugía y el nombre del doctor que emite el reporte.
RF 1.3 Exportar reporte a formato PDF.	Esta funcionalidad permite exportar el reporte como fichero PDF.

RF 1.4 Exportar reporte a formato RTF.	Esta funcionalidad permite exportar el reporte como fichero RTF.
RF 1.5 Exportar reporte a formato HTML.	Esta funcionalidad permite exportar el reporte como fichero HTML.
RF 1.6 Gestionar comentario.	Permite agregar, modificar o eliminar un comentario del reporte.
RF 1.7 Seleccionar imagen a incluir en el reporte.	Permite marcar la imagen que se desea añadir al reporte.

**Tabla1.** Requerimientos funcionales.

En la Fig.6 se muestra el diagrama de requisitos funcionales del módulo de reportes para el sistema de PQO.



*Fig.6 Diagrama de requerimientos funcionales del módulo de reportes para el sistema de PQO.*

### 2.3.2. Requerimientos no funcionales

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares. Estos requerimientos se aplican al sistema en su totalidad, apenas se aplican a características o servicios individuales del sistema. (32)

La siguiente tabla muestra los requisitos no funcionales del módulo de reportes, así como algunos que hereda del sistema PQO, ya que este Módulo será añadido al mismo. Para consultar los prefijos de estos requisitos por categoría, puede remitirse al [Anexo 1](#).

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 rápidamente.
RNU 2. 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.
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 y la MyDICOM.NET para el procesamiento y transmisión de imágenes, la Fluent para los controles visuales y la iTextSharp para generar los reportes en formato PDF.
RNDI 6. Estándar de codificación.	Se utilizará el estándar con notación CamelCase (atributos), PascalCase (métodos) y Notación Húngara (interfaces).

RNDI 7. Uso de Windows Communication Foundation para la comunicación con otros sistemas.	La comunicación con otros sistemas se realizará utilizando la tecnología Windows Communication Foundation (WCF). (RNF heredado del sistema PQO)
RNDI 8. 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 9. Uso de Enterprise Architect como herramienta CASE.	Se utilizará como herramienta CASE Enterprise Architect en su versión 7.5.
RNDI 10. Uso de UML como lenguaje de modelado.	Se utilizará como lenguaje de modelado UML en su versión 2.1.
RNDI 11. Patrones a utilizar para el diseño del sistema.	Los patrones a utilizar para el diseño del sistemas son los siguientes: Command, Singleton, Factory, Model View ViewModel. (RNF heredado del sistema PQO)
RNF 1. Disponibilidad del sistema siempre.	El sistema debe estar disponible siempre.
RNF 2. Exactitud en las salidas del sistema.	El sistema debe brindar salidas precisas.
RNS 1. Implementación de la seguridad de ensamblados mediante firma digital.	Se debe implementar la seguridad de ensamblados mediante firma digital. (RNF heredado del sistema PQO)
RNS 2. Se debe ofuscar el código generado.	El código generado se debe ofuscar.
RNE 1. Tiempo de respuesta al generar un reporte no mayor de 8	Ante la solicitud de generar un reporte de complejidad media o simple, el tiempo de respuesta del sistema no

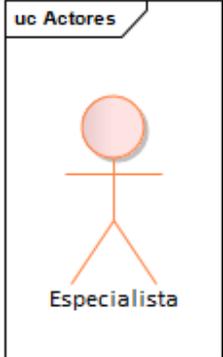
segundos.	debe exceder de 8 segundos. En caso de que sea un reporte masivo, que requiera una gran cantidad de información, el tiempo de respuesta será mayor.
RNFO 1. Memoria RAM de 1Gb.	Para el correcto funcionamiento del sistema, se necesita un CPU Dual Pentium IV 3.0GHz y 1Gb de memoria RAM como mínimo y se recomienda una tarjeta de red Gigabit Ethernet NIC. (RNF heredado del sistema PQO)
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. (RNF heredado del sistema PQO)
RNFO 5. Capacidad de disco duro 20 GB como mínimo.	La capacidad del disco duro debe ser como mínimo de 20 GB. (RNF heredado del sistema PQO)
RNFO 6. Se puede tener en cuenta la necesidad de una tarjeta de aceleración gráfica.	Para mejores resultados se puede tener en cuenta la necesidad de una tarjeta de aceleración gráfica. (RNF heredado del sistema PQO)
RNIU 1. 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.(RNF heredado del sistema PQO)
RNCC 1. Librería MyDICOM.NET	Se compró la librería MyDICOM.NET la cual se debe utilizar en el proceso de conexión con el servidor, las cuales implementan el estándar DICOM y están desarrolladas sobre el lenguaje C#. (RNF heredado del sistema PQO)

<p>RNLI 1. SerialShield SDK para crear las licencias.</p>	<p>Se utiliza SerialShield SDK para crear las licencias del producto. Ya que es una solución de protección de copias y licencias para los editores de software profesionales, que permite proteger proyectos en .NET generando licencias para máquinas específicas, evitando cualquier duplicación o instalaciones no autorizadas.(RNF heredado del sistema PQO)</p>
---	--

**Tabla2.** Requerimientos no funcionales.

#### 2.4. Definición de los actores del sistema

Luego de describir los requerimientos funcionales y no funcionales del módulo de reportes para el sistema de PQO, se define a continuación el actor que interactúa con los casos de uso del sistema. Dicho actor se corresponde con los que ya fueron definidos en los módulos anteriores del PQO, ya que este módulo se integrará de igual forma a dicha aplicación.

Actor	Fundamentación
	<p>Inicia los casos de uso del proceso de gestión de reportes.</p>

**Tabla3.** Actor del sistema.

2.5. Diagrama de casos de uso del sistema

En la siguiente figura se muestra el diagrama de casos de uso del sistema módulo de reportes para el PQO.

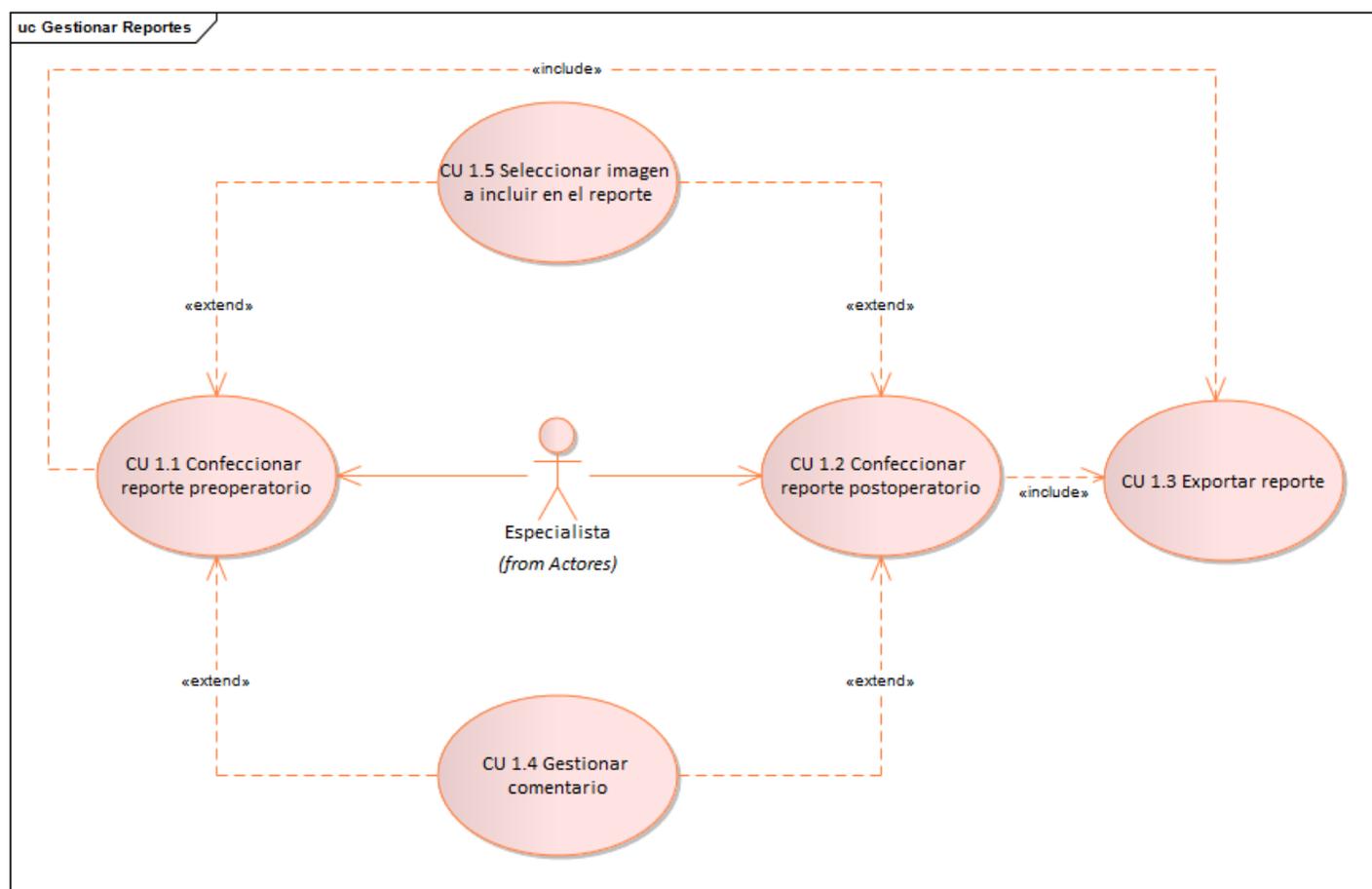


Fig.7 Diagrama de casos de uso del módulo de reportes para el sistema de PQO.

2.6. Descripción de los casos de uso del sistema

A continuación se muestra de manera resumida una descripción de los casos de uso más significativos del sistema. Para consultar las descripciones textuales expandidas de dichos casos de uso ver [Anexo 2](#).

- Caso de uso - Confeccionar reporte preoperatorio.

<b>Objetivo</b>	Confeccionar un reporte preoperatorio que contiene imágenes e información referentes al estudio del caso.
-----------------	---

<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite confeccionar un reporte luego de que se haya terminado el proceso de la planificación, este reporte incluye las imágenes e información que se obtuvieron como resultado del estudio del caso.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.1

**Tabla4.** Resumen del CU Confeccionar reporte preoperatorio.

- Caso de uso - Confeccionar reporte postoperatorio.

<b>Objetivo</b>	Confeccionar un reporte postoperatorio.
<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite confeccionar un reporte luego de que se haya realizado la cirugía.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.2

**Tabla5.** Resumen del CU Confeccionar reporte postoperatorio.

- Caso de uso – Exportar reporte.

<b>Objetivo</b>	Exportar un reporte al formato PDF, RTF o HTML.
<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite exportar en formato RTF, PDF o HTML un reporte que previamente fue confeccionado.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.3, RF 1.4, RF 1.5

**Tabla6.** Resumen del CU Exportar reporte.

- Caso de uso – Gestionar comentario.

<b>Objetivo</b>	Gestionar un comentario para el reporte preoperatorio o postoperatorio.
<b>Actores</b>	Especialista (inicia).
<b>Resumen</b>	El caso de uso permite insertar, modificar o eliminar un comentario. Todas estas operaciones se pueden realizar en una misma ventana.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.6

**Tabla7.** Resumen del CU Gestionar comentario.

- Caso de uso – Seleccionar imagen a incluir en el reporte.

<b>Objetivo</b>	Permite especificar la imagen del estudio que formará parte del reporte.
<b>Actores</b>	Especialista (inicia).
<b>Resumen</b>	El caso de uso brinda la posibilidad de seleccionar la imagen de interés, que debe aparecer en el reporte.
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.7

**Tabla8.** Resumen del CU Seleccionar imágenes a imprimir en el reporte.

## 2.7. Conclusiones

En este capítulo se especificaron los requisitos funcionales y no funcionales de la propuesta de solución. La especificación de los mismos permitió establecer las funcionalidades y las características técnicas y físicas que conforman la aplicación. Se describieron los casos de uso y sus relaciones con los requisitos funcionales identificados, así como los conceptos del dominio que conforman el sistema propuesto, estos elementos facilitaron la definición del diseño de la aplicación.

## CAPÍTULO 3. ARQUITECTURA Y DISEÑO DEL SISTEMA

En este capítulo se presenta la arquitectura del módulo propuesto, así como los diagramas de clases del diseño y de secuencia que tienen relación con los casos de uso desarrollados. También se realiza una descripción de las clases vinculadas a la realización de los casos de uso más significativos desde el punto de vista arquitectónico.

### 3.1. Diseño

El diseño de *software* orientado a objeto transforma el modelo de análisis en un modelo de diseño que sirve como base para la construcción del *software*. Requiere la definición de una arquitectura de *software* multicapa, la especificación de subsistemas que realizan funciones necesarias y proveen soporte de infraestructura, de una descripción de objetos (clases) que son los bloques de construcción del sistema y de una descripción de los mecanismos de comunicación que permiten que los datos fluyan entre las capas, subsistemas y objetos. El diseño de *software* tiene como objetivo mejorar la velocidad del desarrollo y la calidad del producto terminado.

Cuando las características de diseño de un sistema son similares a las de otro, se pueden emplear para su realización patrones de diseño. Estos patrones resuelven problemas específicos de diseño y vuelven al diseño orientado a objeto más flexible, elegante y extremadamente reutilizable. Ayudan a los diseñadores a reutilizar diseños exitosos basando nuevos diseños en experiencias previas. (33) En el diseño del módulo de reportes para el PQO, se empleó el patrón de comportamiento *Command*, el cual permitió encapsular las operaciones referentes a las funcionalidades del sistema en objetos.

Durante el diseño de *software* se establece la arquitectura del sistema. Dicha arquitectura constituye la columna vertebral para su construcción, es en gran medida, responsable de permitir o no ciertos atributos de calidad del sistema, entre los que se destacan la confiabilidad y el rendimiento del *software*. Además establece la estructura, funcionamiento e interacción entre las diferentes partes del mismo. (34)

La realización del diseño debe ser lo más exacta y eficiente posible, para que el sistema pueda ser implementado sin problema alguno. Su objetivo principal es confeccionar los diagramas de clases del diseño, que muestran las clases que participan en la realización de cada caso de uso. Los diagramas de clases del diseño pertenecientes a los casos de uso más relevantes desde el punto de vista arquitectónico se muestran a continuación.

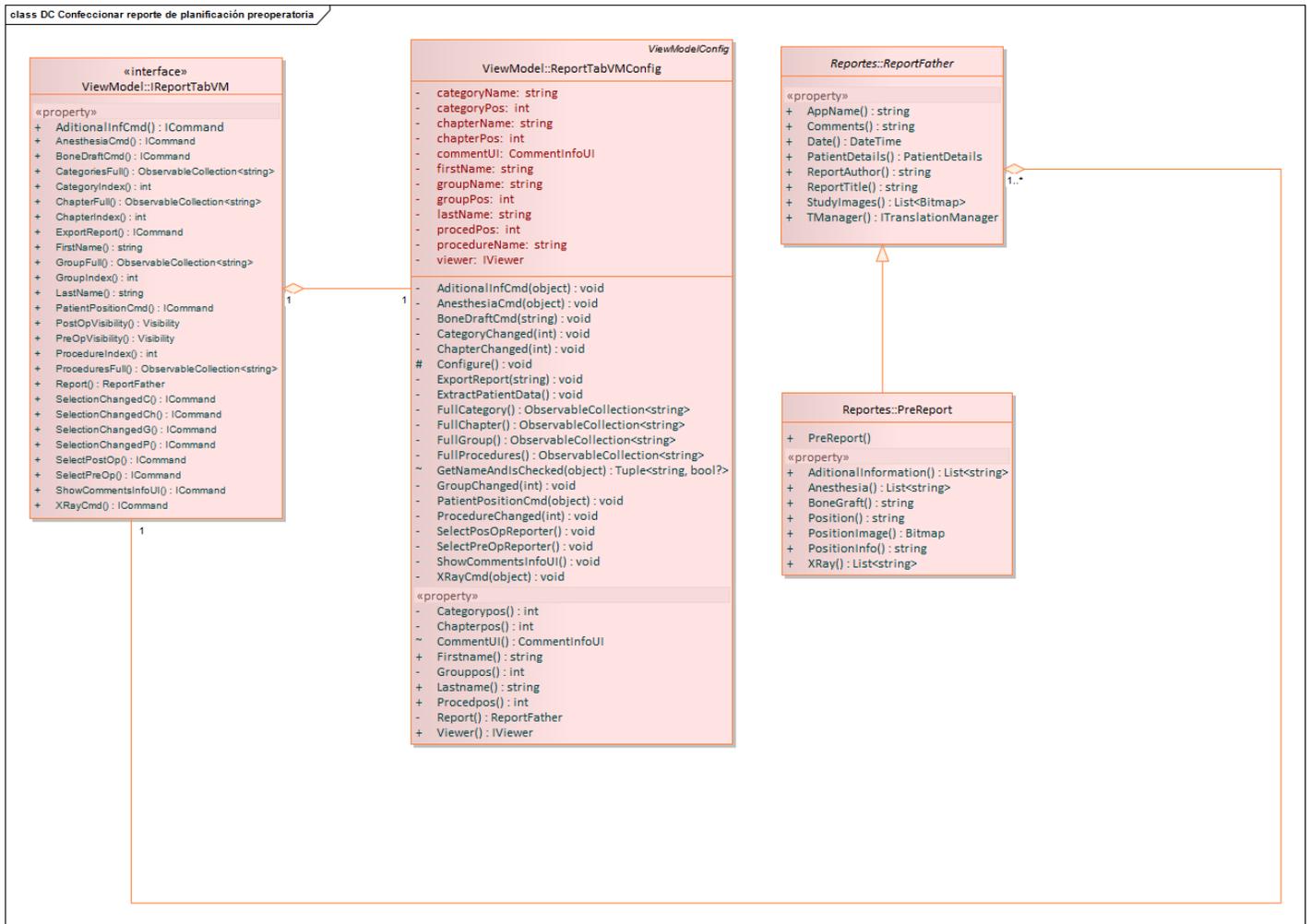


Fig.8 DC-Confeccionar reporte de preoperatorio.

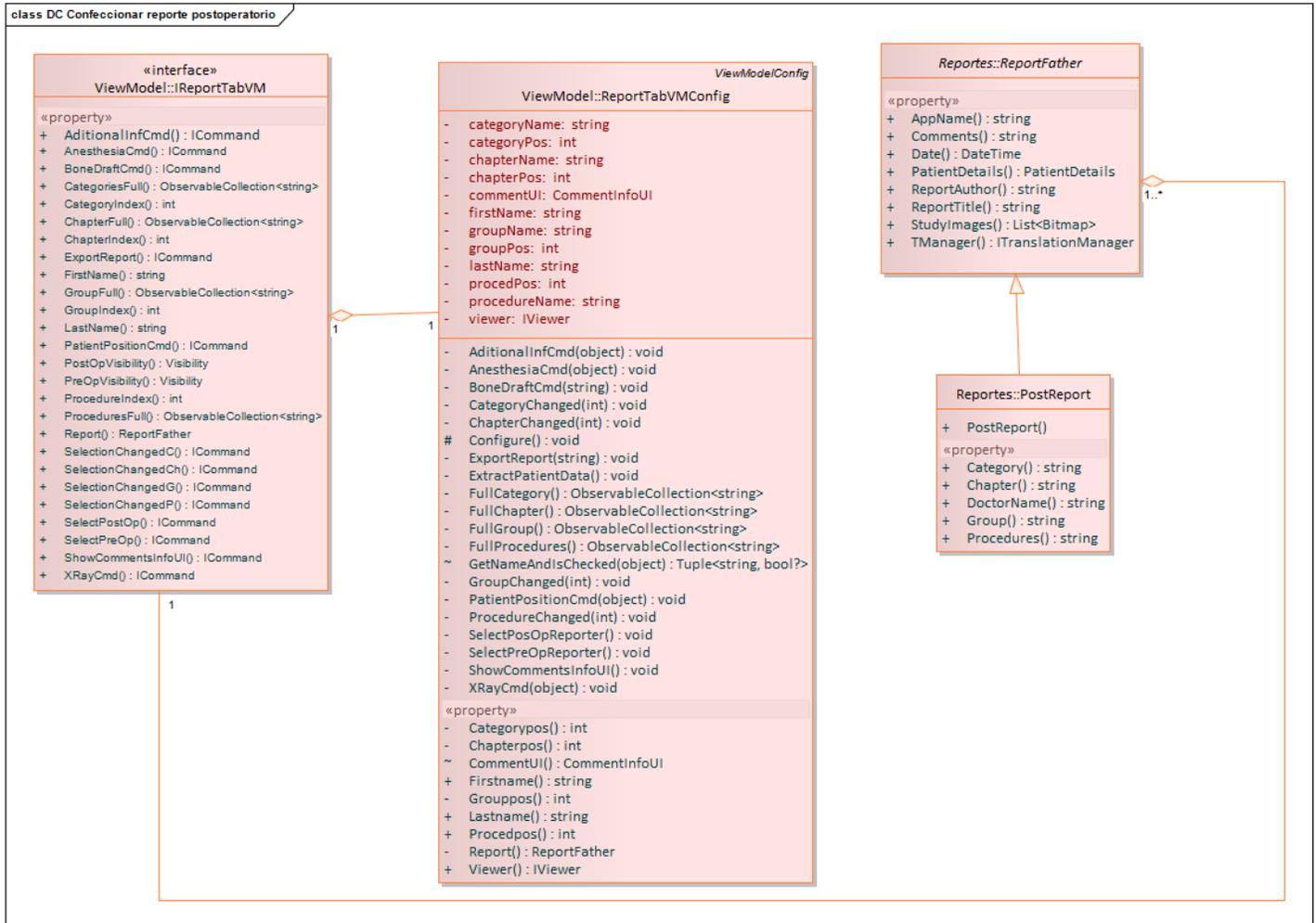


Fig.9 DC Confeccionar reporte postoperatorio.

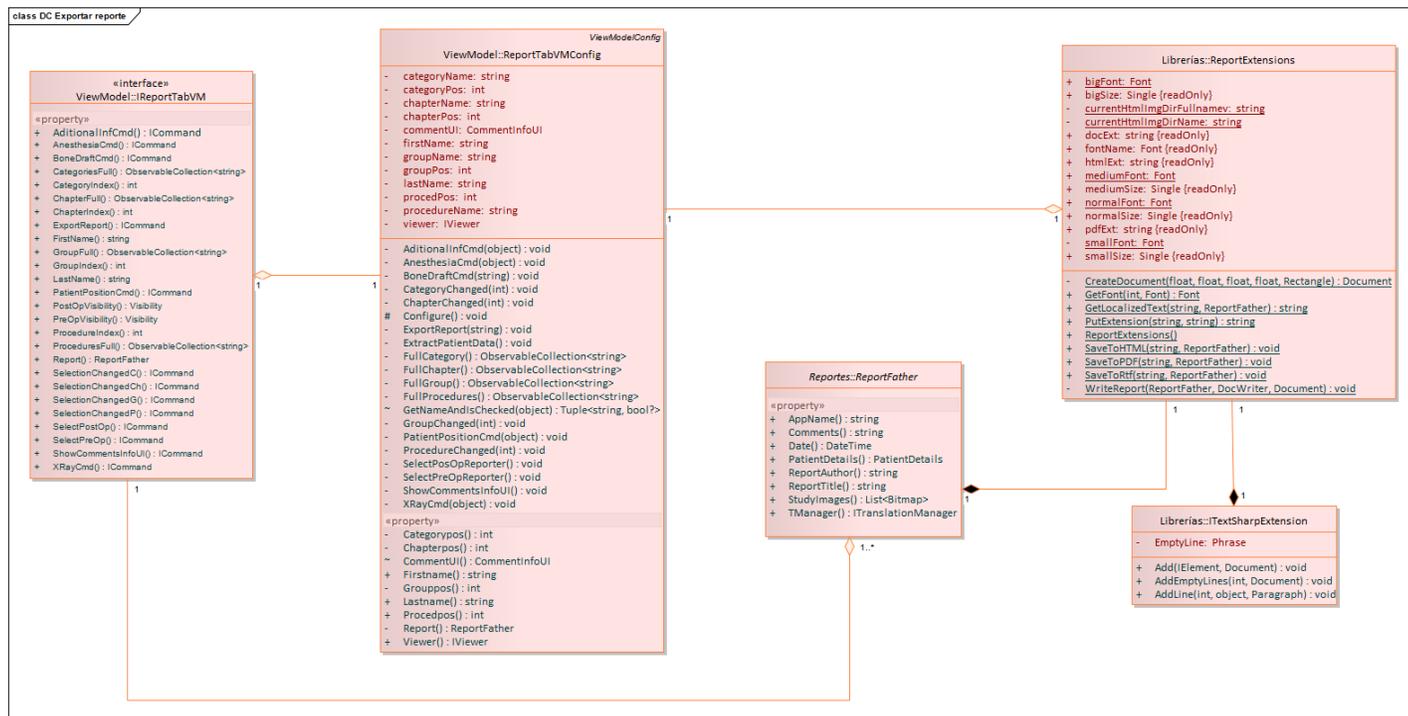


Fig.10 DC Exportar reporte.

Un diagrama de secuencia muestra gráficamente, para un curso de eventos específico en un caso de uso, los actores externos que interactúan directamente con el sistema, así como los eventos del sistema que genera el actor. (35)

A continuación se muestran los diagramas de secuencia de los casos de uso arquitectónicamente más significativos, con el fin de lograr una mejor comprensión de las actividades y eventos que en ellos se llevan a cabo.

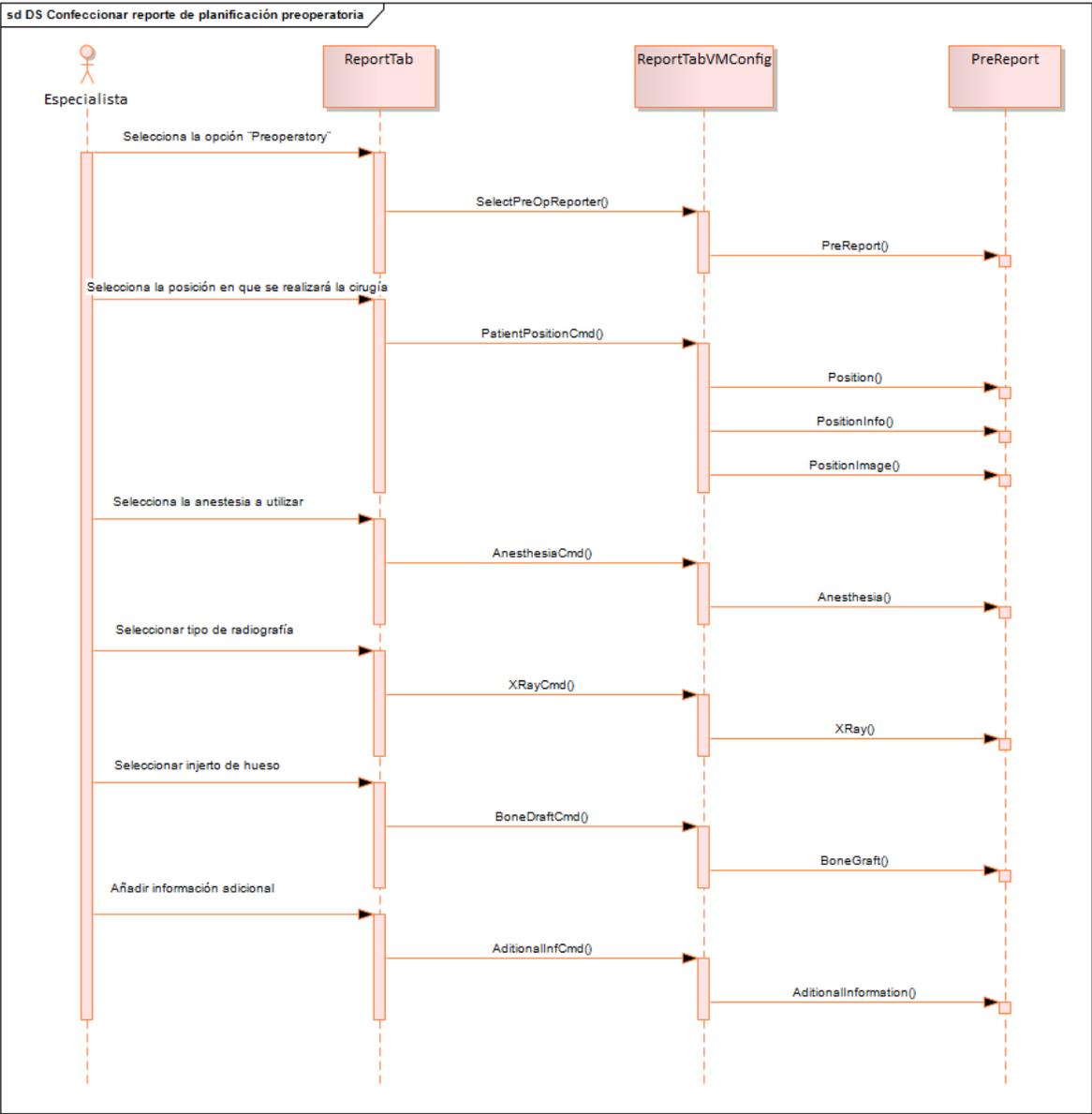


Fig.11 DS Confeccionar reporte de planificación preoperatoria.

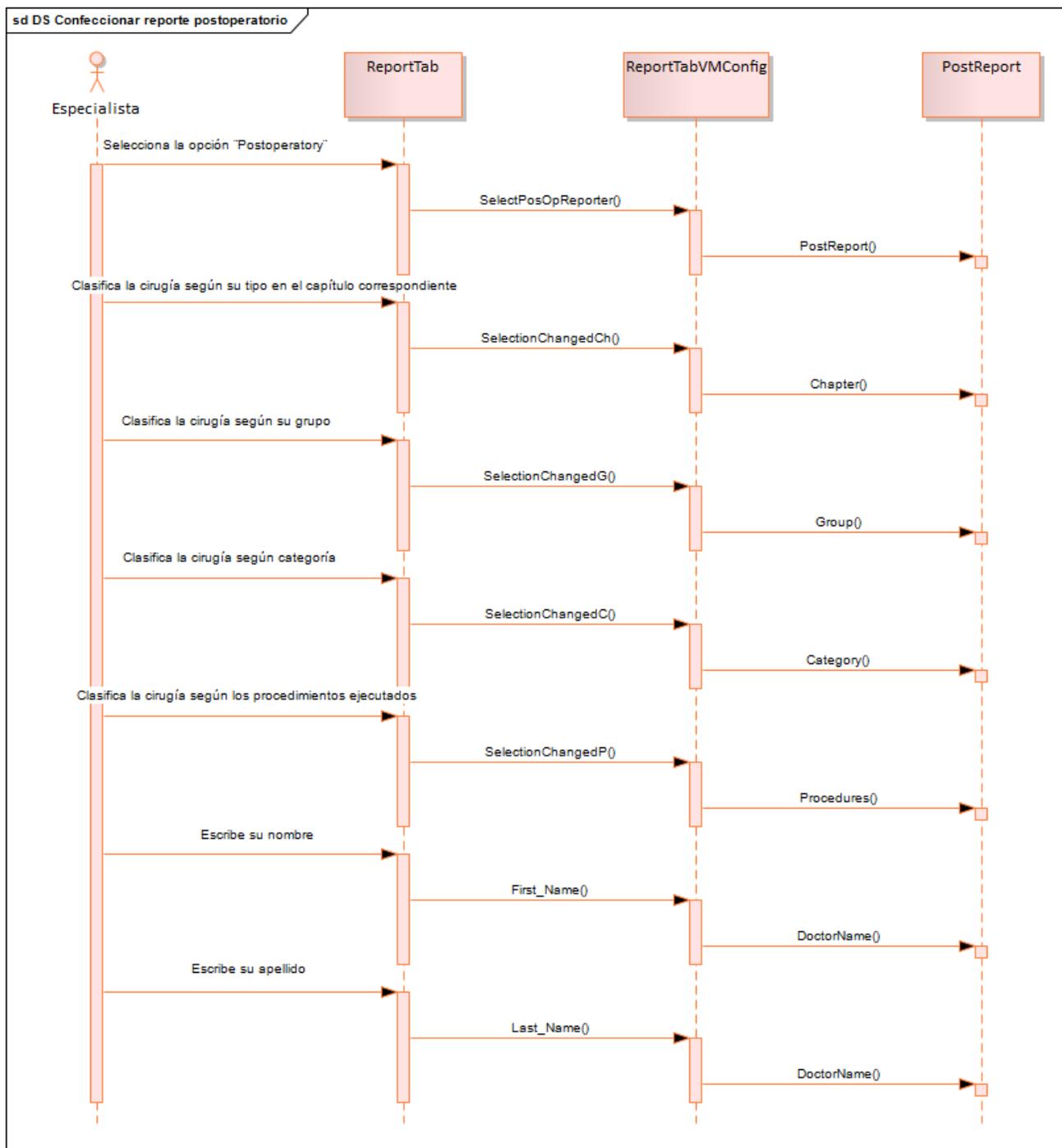


Fig.12 DS Confeccionar reporte postoperatorio.

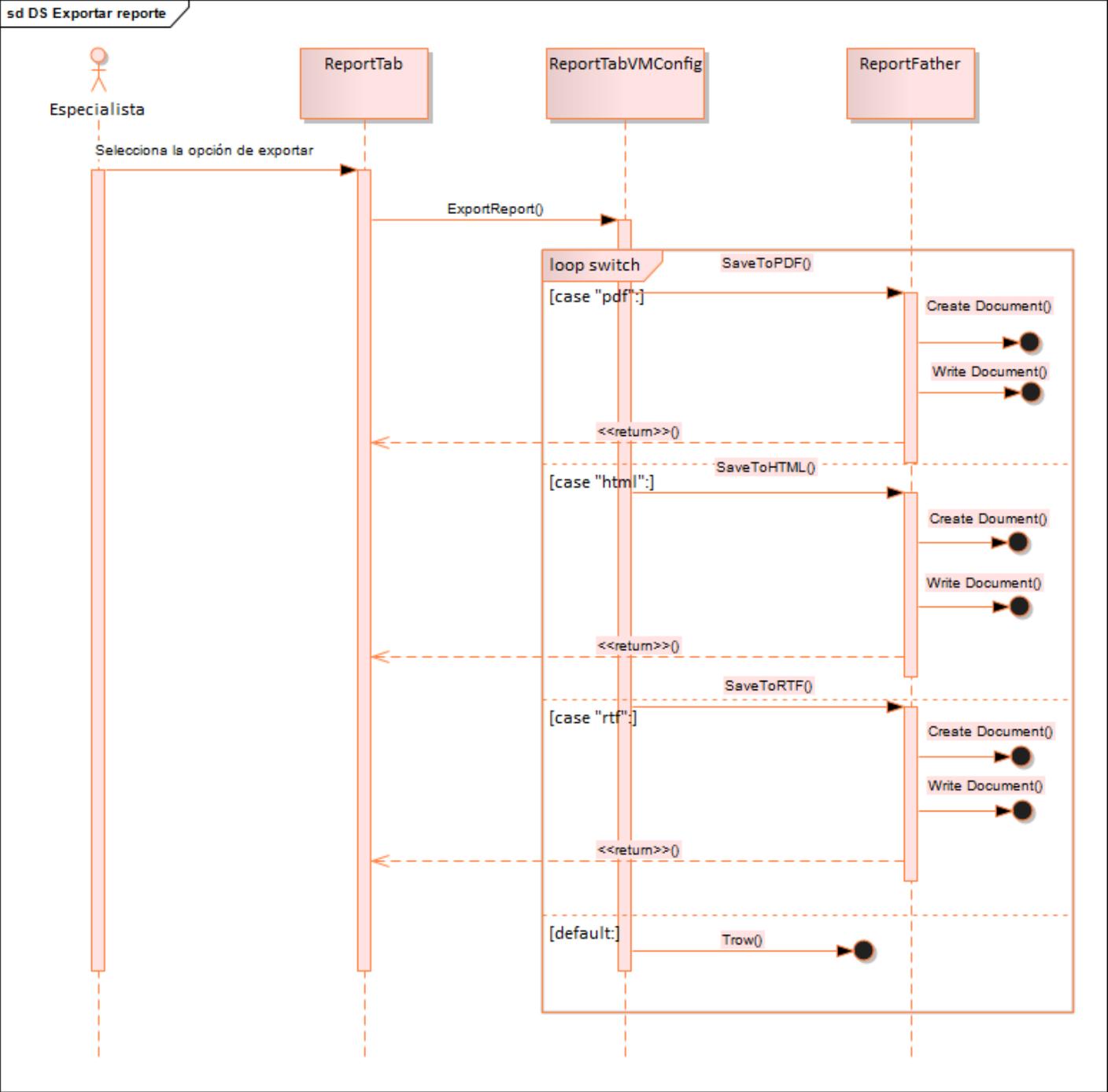


Fig.13 DS Exportar reporte.

3.2. Descripción de las clases

La clase es el concepto fundamental del paradigma orientado a objeto y permite encapsular las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de alguna entidad del mundo real. (33)

El dominio en que se enmarca la solución, pertenece al paradigma orientado a objeto, por lo que son las clases los elementos fundamentales que componen el sistema propuesto y sus relaciones y operaciones las que llevan a cabo las funcionalidades del mismo. La descripción de dichas clases permite una mejor comprensión de los conceptos que representan, principalmente los de aquellas clases que se relacionan con los casos de uso más relevantes arquitectónicamente. Estas descripciones se encuentran en el [Anexo 3](#).

### 3.3. Modelo arquitectónico

El diseño de la arquitectura de un sistema es un proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso determina qué componentes forman el sistema, cómo se relacionan entre ellos y cómo mediante su interacción, llevan a cabo la funcionalidad especificada cumpliendo con los criterios de calidad indicados como: seguridad, disponibilidad, eficiencia o usabilidad. Para diseñar la arquitectura de un sistema es importante tener en cuenta los intereses de los usuarios y los objetivos del negocio al que se enmarca el mismo.

La meta final de la arquitectura es identificar los requisitos que producen un impacto en la estructura del sistema y reducir los riesgos asociados en la construcción del mismo. Debe soportar los cambios futuros del *software*, del *hardware* y de las funcionalidades demandadas por los clientes.

Durante el diseño de la arquitectura se establece el estilo o los estilos arquitectónicos a utilizar, ya que lo normal en una arquitectura es que no se base en uno solo, sino que combine varios de dichos estilos para obtener las ventajas de cada uno. Un estilo arquitectónico es un conjunto de principios que definen a alto nivel un aspecto de la aplicación. Su definición está determinada por un conjunto de componentes, un conjunto de conexiones entre dichos componentes y un conjunto de restricciones sobre cómo se comunican dos componentes cualesquiera conectados. Su organización se realiza en torno al aspecto de la aplicación sobre el cual se centra, estos son: comunicación, despliegue, dominio, interacción y estructura. (36)

El sistema PQO es una aplicación de escritorio que está constituido por un conjunto de componentes que exponen interfaces bien definidas y que colaboran entre sí para brindar las funcionalidades requeridas en el proceso de automatización para el que fue desarrollado. Dichos componentes son independientes de los

demás, por lo que pueden ser actualizados, desplegados, modificados o extendidos para ofrecer nuevos comportamientos sin afectar al resto. Estas características facilitan el despliegue, reducen los costos, la complejidad y garantizan la reusabilidad del sistema.

Otra característica del PQO es que utiliza un mecanismo basado en extensiones haciendo uso de MEF. Esta librería permite a los desarrolladores más flexibilidad y facilita el trabajo en el diseño de aplicaciones que pueden ser compuestas por otras en tiempo de ejecución, ya que posibilita que un sistema sea integrado por una colección de partes. (16)

El módulo de reportes es una solución que será integrada al sistema PQO como un componente o extensión que será cargada en tiempo de ejecución. Para hacer posible esto, la construcción del sistema se basará en una arquitectura híbrida entre algunos elementos de la Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) y un mecanismo basado en extensiones. Las interfaces que utilizan cada servicio y que pueden ser invocadas para llamar al servicio solicitado a través de los contratos, los cuales a su vez, establecen la comunicación entre dichos servicios mediante mensajes con operaciones de aplicación, son algunos elementos de la arquitectura SOA que posibilitarán la integración del módulo de reportes con el PQO. Esta arquitectura garantizará la autonomía de los servicios del sistema, o sea, cada uno se mantiene, desarrolla, despliega y versiona de manera independiente.

El módulo emplea además patrones como Inversión de Control (IOC, por sus siglas en inglés), Inyección de Dependencias (DI, por sus siglas en inglés) y el MVVM. Los primeros garantizan el bajo acoplamiento y reutilización de los componentes del sistema y permiten que este responda a eventos y acciones que hagan los usuarios de la aplicación. El tercero a su vez, constituye el patrón arquitectónico principal de la solución.

### 3.4. Conclusiones

La descripción de las clases de diseño y de las relaciones que existen entre ellas, mediante el uso de los diagramas de clases y de secuencia, así como de la arquitectura que presenta la aplicación, posibilitó sentar las bases para la fase de implementación de la propuesta de solución.

## CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo se describen los elementos relacionados a la implementación del sistema. Se especifican los componentes que lo integran así como las interacciones que se establecen entre ellos, dichos elementos garantizan el correcto funcionamiento de la aplicación. Se muestra además cómo se desplegarán físicamente las distintas partes de la solución y los protocolos mediante los que se comunicarán.

### 4.1. Diagrama de componentes

Un componente es una parte física y reemplazable de un sistema. Posee características similares a una clase: tiene nombre, realiza interfaces, puede participar en relaciones e interacciones y puede tener instancias. Representa un bloque de construcción fundamental sobre el cual se puede diseñar y construir sistemas. (37)

Los diagramas de componentes ilustran las piezas del *software* y los controladores embebidos, que conformarán un sistema. Poseen un nivel de abstracción más elevado que un diagrama de clases ya que un componente puede estar formado por varias clases u objetos en tiempo de ejecución. Los diagramas de componentes ofrecen un mecanismo de agrupamiento rico desde el punto de vista semántico y muestran de manera privada los elementos del modelo. (38)

En la figura 8 se muestran los componentes que integran al módulo de reportes, así como la interacción entre cada uno de ellos.

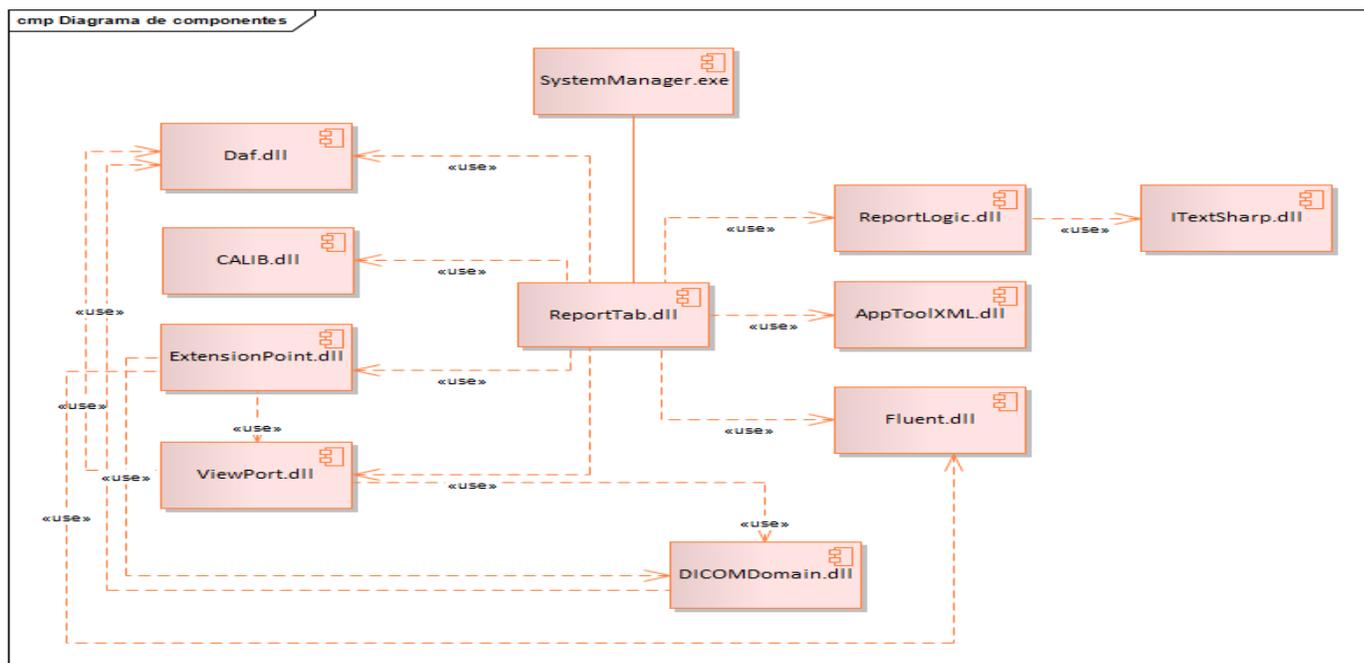


Fig.14 Diagrama de componentes del módulo de reportes.

En la siguiente tabla se muestra la descripción de cada uno de los componentes que conforman el módulo de reportes para el PQO.

Componente	Descripción
CALIB.dll	Librería que permite la lectura y procesamiento de las imágenes DICOM.
DICOMDomain.dll	Esta librería gestiona la información representada por el control de visualización del sistema, referente a las imágenes DICOM.
ViewPort.dll	Componente utilizado para la visualización de las imágenes en el visor del sistema PQO.
ExtensionPoint.dll	Posee los contratos que definen la interacción entre los diversos componentes que integran el sistema PQO.
Daf.dll	A través de sus clases y métodos y mediante un diccionario definido en el sistema, este componente establece el idioma que adoptará cada interfaz de la aplicación.

Fluent.dll	Esta librería se utiliza para garantizar un estilo de la interfaz del sistema capaz de albergar todas las funcionalidades que este requiere y que a su vez sea visualmente más atractiva.
AppToolXML.dll	Brinda información referente a las posibles clasificaciones de las cirugías ortopédicas, en formato .xml a partir de un archivo en formato .xls, estas clasificaciones son recogidas en el reporte postoperatorio.
ReportLogic.dll	Establece los tipos de reporte a generar y define los elementos que estos poseen. Utiliza también el componente ITextSharp para generar los reportes.
ITextSharp.dll	Esta librería permite definir la estructura de los reportes y emitirlos en formato .pdf, .rtf y .html.
ReportTab.dll	Representa la propuesta de solución, la cual emplea los componentes descritos con anterioridad y a su vez constituye un componente del sistema PQO.
SystemManager.exe	Constituye el componente inicial de ejecución del sistema PQO.

**Tabla9.** Descripción de los componentes del módulo de reportes para el PQO.

## 4.2. Diagrama de despliegue

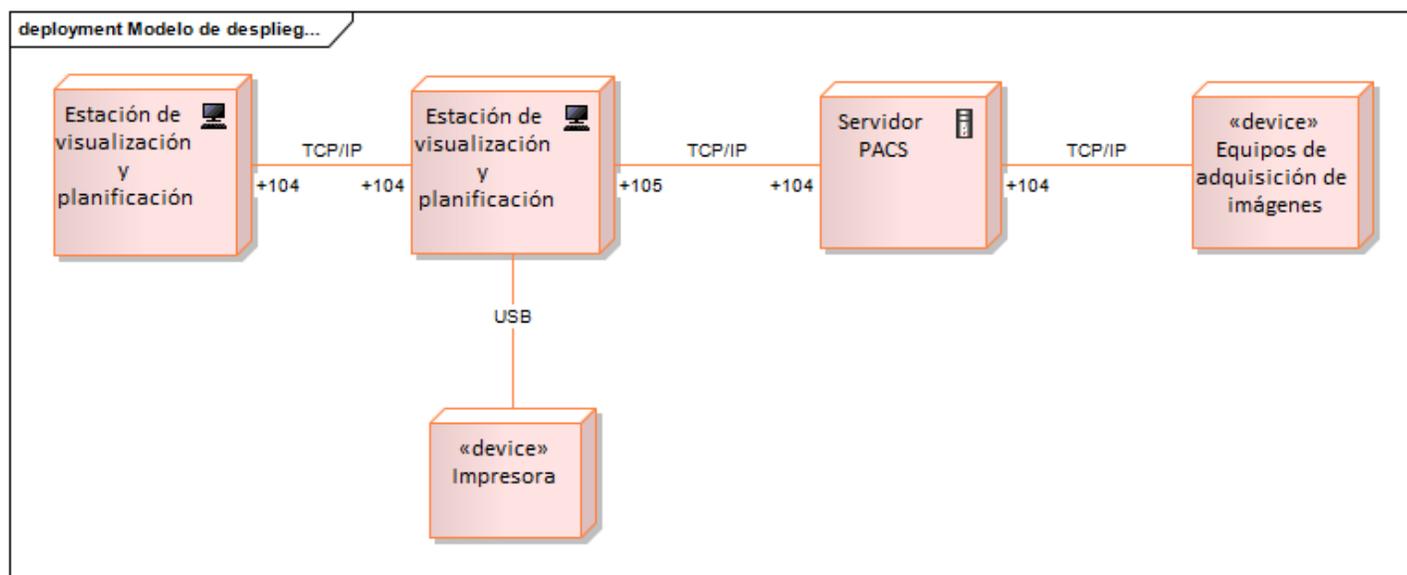
El despliegue es la etapa del desarrollo de *software* que describe la configuración del sistema para su ejecución en un ambiente del mundo real. Para el despliegue se deben de tomar decisiones sobre los parámetros de la configuración del sistema y su funcionamiento, así como la asignación de recursos para su correcta ejecución, distribución y concurrencia.

Los diagramas de despliegue describen la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes del *software*. Presentan de forma detallada la topología del sistema: la estructura de los elementos de *hardware* y el *software* que ejecuta cada uno de ellos. Están compuestos por nodos, los cuales simulan los elementos de *hardware* que emplea el sistema durante su ejecución.

Los nodos son objetos físicos en tiempo de ejecución que representan un recurso computacional el cual generalmente tiene memoria y capacidad de procesamiento. Simbolizan dispositivos sobre los cuales se pueden desplegar los componentes que integran el sistema. (39)

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. (40)

El modelo de despliegue del módulo de reportes es el mismo que el del sistema PQO, puesto que dicho módulo constituye un componente del sistema mencionado. La siguiente figura muestra el modelo de despliegue del Planificador Quirúrgico Ortopédico.



**Fig.15** Modelo de despliegue del PQO.

A continuación se describen cada uno de los nodos que componen el modelo de despliegue del módulo de reportes.

Nodo	Descripción
Equipos de visualización de imágenes.	Son aquellos dispositivos que se utilizan en los centros hospitalarios para la obtención de imágenes y radiografías médicas empleando

	técnicas como la fluoroscopia, resonancia magnética, ultrasonido, tomografía computarizada, etc.
Estaciones de visualización y planificación.	Son aquellos equipos (computadoras) que emplean los especialistas para la visualización de las imágenes tomadas por los dispositivos mencionados con anterioridad y que permiten a su vez realizar el proceso de planificación.
PACS.	Servidor de almacenamiento de imágenes e información médica al que se encuentran conectados las estaciones de trabajo y los equipos de visualización de imágenes.
Impresoras.	Dispositivos que se conectan a las estaciones de trabajo mediante los cuales se pueden imprimir las imágenes y reportes que el médico desee.

**Tabla10.** Descripción de los nodos del modelo de despliegue del PQO.

### 4.3. Estándares de codificación

Los estándares de codificación comprenden todos los aspectos concernientes a la generación del código de una aplicación. Son patrones que garantizan que todo código fuente refleje un estilo armonioso, como si un único programador lo hubiera escrito de una sola vez. Una correcta definición de estos estándares permitirá una mejor comprensión del código por parte de aquellos programadores que no lo escribieron, asegurando que sea posible un mejor mantenimiento del mismo en el futuro. La utilización de un estándar de codificación bien definido, aumentará el rendimiento y la calidad de la aplicación. (41)

#### 4.3.1. Métodos

Se definió para el código empleado en el desarrollo de los métodos de la propuesta de solución, el estándar con notación PascalCase. Esta notación establece que el primer caracter de todas las palabras se escriben en Mayúsculas y el resto de los caracteres en minúsculas. (42)

A continuación se muestra un ejemplo del empleo de esta notación en el método que permite exportar los reportes, en la clase ReporTabVMConfig:

```
private void ExportReport(string reportExt) {...}
```

### 4.3.2. Atributos, variables y parámetros

Para la declaración de los atributos, variables y parámetros de los métodos, se empleó la notación CamelCase. Esta notación establece que el primer caracter de todas las palabras, excepto el de la primera se escriben en Mayúsculas, el resto en minúsculas. (42)

La siguiente línea de código muestra la declaración de un atributo de la clase ReportTabVMConfig ejemplificando el uso de esta notación:

```
private int chapterPos ;
```

### 4.3.3. Interfaz visual

En el código que implementa la interfaz visual del módulo de reportes para el PQO, se definió el uso de la notación Húngara. La cual especifica el tipo de dato de la variable como un prefijo en el nombre de esta. (42)

Este es un ejemplo de un elemento visual de la interfaz de usuario del módulo de reportes para el PQO. En este caso se presenta un atributo de la clase Report, que se enlaza con un textbox de la interfaz visual. Este atributo toma el valor que se inserte en el textbox al que está enlazado.

```
public string sFirstName;
```

## 4.4. Conclusiones

En este capítulo se describieron una serie de elementos que facilitaron la implementación de la propuesta de solución. Mediante los diagramas de componentes del sistema y de despliegue, se modelaron los componentes y nodos físicos que integran la aplicación, así como las interacciones que se establecen entre ellos. La comprensión de las relaciones que existen entre los componentes del sistema permitió una mejor implementación de las funcionalidades que debe de brindar el mismo.

### CONCLUSIONES

Con la culminación de la presente investigación se logró desarrollar, cumpliendo con los requerimientos especificados para su confección, el módulo de reportes para el sistema PQO, el cual permitirá la creación de reportes durante el proceso de planificación quirúrgica ortopédica.

- El estudio de los principales sistemas internacionales que emiten reportes durante la planificación quirúrgica ortopédica, permitió una mejor comprensión e implementación de las funcionalidades que conforman la propuesta de solución. Lo cual contribuyó al desarrollo de una aplicación capaz de solucionar los problemas existentes en el proceso de emisión de reportes. Asegurando de esta manera, con su integración al PQO, un alto nivel competitivo de dicho sistema en el mercado de *software* mundial.
- La arquitectura definida durante el desarrollo del módulo de reportes para el PQO, garantiza que este posea características como: robustez, adaptabilidad y estabilidad ante los cambios del negocio, facilidad para la realización de futuros mantenimientos, flexibilidad y extensibilidad de cualquiera de sus componentes. Además, el empleo de patrones como inversión de control e inyección de dependencias permiten que el sistema sea capaz de responder ante los eventos realizados por los usuarios.
- La integración del módulo de reportes con el PQO, le otorgará a este una serie de funcionalidades que lo convertirán en un sistema CAOS más completo. La utilización de dicho módulo permitirá la creación de reportes preoperatorios y postoperatorios de manera rápida, exacta y eficiente, durante el proceso de planificación quirúrgica. Garantizará que estos reportes al ser generados en formato digital, se almacenen de manera más segura y duradera. Además la aplicación contribuirán a agilizar la preparación del quirófano para efectuar las cirugías, mejorando los servicios médicos brindados a los pacientes.

## **RECOMENDACIONES**

Con el objetivo de perfeccionar la solución propuesta se proponen como recomendaciones para el trabajo presentado las siguientes:

- Implementar la funcionalidad de imprimir los reportes realizados por el sistema durante el proceso de planificación quirúrgica.
- Añadir al reporte preoperatorio los implantes óseos a utilizar, así como la personalización de los mismos, como parte de los implementos a tener en cuenta durante la preparación del quirófano.

## REFERENCIAS BIBLIOGRÁFICAS

1. ONEI. *Censo de Población y Vivienda 2012*. [En línea] [Citado el: 04 de febrero de 2014.] <http://www.one.cu/informacional2012.html>.
2. *Revista Cubana de Ortopedia y Traumatología*. Hernández Falcón, Daylín, Marrero Riverón, Luis Oscar y Ledea Lozano, Oscar E. 2, Ciudad de la Habana : Editorial Ciencias Médicas, 2012, Vol. 26.
3. SERVIMED turismo y salud. *Instituciones de Salud*. [En línea] [Citado el: 04 de febrero de 2014.] [http://www.solymed.it/Turismo%20y%20Salud/institucionsalud\\_6.html](http://www.solymed.it/Turismo%20y%20Salud/institucionsalud_6.html).
4. *TRAUMAPLAN: para la planificación preoperatoria en traumatología*. Ramírez Esmitt, Ernesto Coto. 2, Caracas, Venezuela : s.n., 2011, Enl@ce: Revista Venezolana de Información, tecnología y conocimiento vol8, número 2, págs. 61-78.
5. Definición .de. [En línea] [Citado el: 05 de febrero de 2014.] <http://definicion.de/reporte/>.
6. *Computer-aided diagnosis: The emerging of three CAD system induced by Japanese health care needs*. Hiroshi Fujita, Yoshikazu Uchiyama. 2008, Computer Methods and Programs in Biomedicine, págs. 238–248.
7. Suárez Cuenca, Jorge Juan. *DESARROLLO DE UN SISTEMA DE DIAGNÓSTICO ASISTIDO POR COMPUTADOR PARA DETECCIÓN DE NÓDULOS PULMONARES EN TOMOGRAFÍA COMPUTARIZADA MULTICORE*. Santiago de Compostela : s.n., 2009.
8. Medina, David del Río, Bocanegra Sánchez, Carlos y Santo Orcero, David. *Revista Salud.com. vol4, No16*. [En línea] 2008. [Citado el: 20 de febrero de 2014.] <http://www.revistaesalud.com/index.php/revistaesalud/article/view/266/598>.
9. voyandhealth. *TraumaCad*. [En línea] Voyand Health, 2014. [Citado el: 20 de febrero de 2014.] <http://www.voyanthealth.com/traumacad.jsp>.
10. OrthoView. *orthoview*. [En línea] Meridian Technique Limited. [Citado el: 19 de febrero de 2014.] <http://www.orthoview.com/>.

11. MERGE. *Merge OrthoCase*. [En línea] Merge Healthcare Incorporated. [Citado el: 10 de febrero de 2014.] <http://www.merge.com/Solutions/Orthopedics/OrthoCase.aspx>.
12. Imaging Systems & Services . *View Pro-X Ortho*. [En línea] [Citado el: 2 de marzo de 2014.] [http://www.ncdmedical.com/?content=vpx\\_ortho](http://www.ncdmedical.com/?content=vpx_ortho).
13. P&C X-ray Services. *Cedara ProPlanner*. [En línea] 2009. [Citado el: 2 de marzo de 2014.] [http://www.pcx-ray.com/pdfs/Medilink\\_Soft\\_OrthoWorkProPlanner.pdf](http://www.pcx-ray.com/pdfs/Medilink_Soft_OrthoWorkProPlanner.pdf).
14. Siemens. *healthcare*. [En línea] septiembre de 2010. [Citado el: 4 de marzo de 2014.] [http://www.healthcare.siemens.com/siemens\\_hwem-hwem\\_sssa\\_websites-context-root/wcm/idc/groups/public/@global/@imaging/@c-arms/documents/download/mdaw/mtqx/~edisp/preoplan-product-brochure-00029745.pdf](http://www.healthcare.siemens.com/siemens_hwem-hwem_sssa_websites-context-root/wcm/idc/groups/public/@global/@imaging/@c-arms/documents/download/mdaw/mtqx/~edisp/preoplan-product-brochure-00029745.pdf).
15. MSDN. *Developer Network*. [En línea] Microsoft. [Citado el: 20 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=vs.110%29.aspx>.
16. MacDonald, Matthew. *Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4.0*. New York : APRESS, 2010.
17. MSDN. *Managed extensibility framework*. [En línea] Microsoft. [Citado el: 20 de febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/ee291628.aspx>.
18. Polyack, Nick. CodeProject. *MVVM-Pattern-Made-Simple*. [En línea] 30 de octubre de 2012. [Citado el: 17 de febrero de 2014.] <http://www.codeproject.com/Articles/278901/MVVM-Pattern-Made-Simple>.
19. González Seco, José Antonio. *El lenguaje de programación C#*.
20. MSDN. *C#*. [En línea] 2010. [Citado el: 17 de febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/ff796223.aspx>.
21. Oré Bravo, Alexander. CalidadySoftware. *introduccioncmmi*. [En línea] 13 de abril de 2008. [Citado el: 17 de febrero de 2014.] [http://www.calidadysoftware.com/otros/introduccion\\_cmmi.php](http://www.calidadysoftware.com/otros/introduccion_cmmi.php).
22. Chrissis, Mary Beth, Konrad, Mike y Shrum, Sandy. *CMMI Guía para integración de procesos y la mejora de productos*. s.l. : Pearson Education, 2009.

23. El Modelo CMMI. [En línea] noviembre de 2008. [Citado el: 5 de junio de 2014.] <http://www.allsoft.com.mx/recursos/ElModeloCMMI.pdf> .
24. Hernández Orallo, Enrique. *El Lenguaje Unificado de Modelado (UML)*.
25. Belloso Cicilia, Claudia Ivonne. *MONOGRAFIA SOBRE LA METODOLOGIA DE DESARROLLO DE SOFTWARE*. El Salvador : Universidad Don Bosco, septiembre 2009.
26. MSDN. *VisualStudio*. [En línea] Microsoft. [Citado el: 18 de febrero de 2014.] <http://msdn.microsoft.com/es-es/library/vstudio/6b6b1f4%28v=vs.100%29.aspx>.
27. unitty. [En línea] 03 de octubre de 2012. [Citado el: 20 de febrero de 2014.] [http://www.unitty.com/index.php?option=com\\_k2&view=item&id=139:ea-main&Itemid=88](http://www.unitty.com/index.php?option=com_k2&view=item&id=139:ea-main&Itemid=88).
28. SparxSystems. [En línea] SparxSystems Pty Ltd. [Citado el: 21 de febrero de 2014.] <http://www.sparxsystems.com.ar/products/ea/7.5/index.html#releasenotes>.
29. Küng, Stefan, Onken, Lübbe y Large, Simon. *TortoiseSVN Un cliente de Subversion para Windows*.
30. EcuRed. *Modelo de dominio*. [En línea] 18 de abril de 2012. [Citado el: 26 de marzo de 2014.] [http://www.ecured.cu/index.php/Modelo\\_de\\_dominio](http://www.ecured.cu/index.php/Modelo_de_dominio).
31. Monferrer Agut, Raúl. *Especificación de Requisitos Software según el estándar de IEEE830*. Castellón, España : Universidad de Jaume I, 2001.
32. Sommerville, Ian. *Ingeniería de Requisitos. Ingeniería del Software*. Madrid : PEARSON EDUCACIÓN S.A., 2005.
33. Pressman, Roger S. *Ingeniería del Software: Un enfoque Práctico*. s.l. : McGrawHill. 5ta edición.
34. EcuRed. *Arquitectura de software*. [En línea] 1 de octubre de 2012. [Citado el: 28 de marzo de 2014.] [http://www.ecured.cu/index.php/Arquitectura\\_de\\_software](http://www.ecured.cu/index.php/Arquitectura_de_software).
35. Larman, Craig. *UML y Patrones: una introducción al análisis y diseño orientado a objeto y al proceso unificado*. s.l. : Prentice Hall.
36. Llorente, César de la Torre, y otros. *Guía de Arquitectura N-capas orientada al Dominio con .NET 4.0*. España : Krasis Consulting, 2010.

37. Daniele, Marcela. [En línea] 2007. [Citado el: 30 de abril de 2014.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
38. SparxSystem. *Diagrama de Componentes UML*. [En línea] [Citado el: 30 de abril de 2014.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html).
39. Rubiano, Martha. Scribd. [En línea] 9 de septiembre de 2010. [Citado el: 2 de mayo de 2014.] <http://es.scribd.com/doc/19611312/diagramas-de-despliegue-2222>.
40. Marca Huallpara, Hugo Michel y Quisbert Limachi, Nancy Susana. *Diagrama de Despliegue*.
41. MSDN. *Revisión de código y estándares de codificación*. [En línea] [Citado el: 6 de junio de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
42. Fernández, Alberto. *Estándares de Codificación en C# y Buenas Prácticas de Programación*.

## BIBLIOGRAFÍA

ONEI. *Censo de Población y Vivienda 2012*. [En línea] [Citado el: 04 de febrero de 2014.] <http://www.one.cu/informacional2012.html>.

*Revista Cubana de Ortopedia y Traumatología*. Hernández Falcón, Daylín, Marrero Riverón, Luis Oscar y Ledea Lozano, Oscar E. 2, Ciudad de la Habana : Editorial Ciencias Médicas, 2012, Vol. 26.

SERVIMED turismo y salud. *Instituciones de Salud*. [En línea] [Citado el: 04 de febrero de 2014.] [http://www.solymed.it/Turismo%20y%20Salud/institucionsalud\\_6.html](http://www.solymed.it/Turismo%20y%20Salud/institucionsalud_6.html).

*TRAUMAPLAN: para la planificación preoperatoria en traumatología*. Ramírez Esmitt, Ernesto Coto. 2, Caracas, Venezuela : s.n., 2011, Enl@ce: Revista Venezolana de Información, tecnología y conocimiento vol8, número 2, págs. 61-78.

Definición .de. [En línea] [Citado el: 05 de febrero de 2014.] <http://definicion.de/reporte/>.

*Computer-aided diagnosis: The emerging of three CAD system induced by Japanese health care needs*. Hiroshi Fujita, Yoshikazu Uchiyama. 2008, Computer Methods and Programs in Biomedicine, págs. 238–248.

Suárez Cuenca, Jorge Juan. *DESARROLLO DE UN SISTEMA DE DIAGNÓSTICO ASISTIDO POR COMPUTADOR PARA DETECCIÓN DE NÓDULOS PULMONARES EN TOMOGRAFÍA COMPUTARIZADA MULTICORE*. Santiago de Compostela : s.n., 2009.

Medina, David del Río, Bocanegra Sánchez, Carlos y Santo Orcero, David. *Revista Salud.com*. vol4, No16. [En línea] 2008. [Citado el: 20 de febrero de 2014.] <http://www.revistaesalud.com/index.php/revistaesalud/article/view/266/598>.

voyandhealth. *TraumaCad*. [En línea] Voyand Health, 2014. [Citado el: 20 de febrero de 2014.] <http://www.voyanthealth.com/traumacad.jsp>.

OrthoView. *orthoview*. [En línea] Meridian Technique Limited. [Citado el: 19 de febrero de 2014.] <http://www.orthoview.com/>.

- MERGE. *Merge OrthoCase*. [En línea] Merge Healthcare Incorporated. [Citado el: 10 de febrero de 2014.] <http://www.merge.com/Solutions/Orthopedics/OrthoCase.aspx>.
- Imaging Systems & Services . *View Pro-X Ortho*. [En línea] [Citado el: 2 de marzo de 2014.] [http://www.ncdmedical.com/?content=vpx\\_ortho](http://www.ncdmedical.com/?content=vpx_ortho).
- P&C X-ray Services. *Cedara ProPlanner*. [En línea] 2009. [Citado el: 2 de marzo de 2014.] [http://www.pcx-ray.com/pdfs/Medilink\\_Soft\\_OrthoWorkProPlanner.pdf](http://www.pcx-ray.com/pdfs/Medilink_Soft_OrthoWorkProPlanner.pdf).
- Siemens. *healthcare*. [En línea] septiembre de 2010. [Citado el: 4 de marzo de 2014.] [http://www.healthcare.siemens.com/siemens\\_hwem-hwem\\_sssa\\_websites-context-root/wcm/idc/groups/public/@global/@imaging/@c-arms/documents/download/mdaw/mtqx/~edisp/preoplan-product-brochure-00029745.pdf](http://www.healthcare.siemens.com/siemens_hwem-hwem_sssa_websites-context-root/wcm/idc/groups/public/@global/@imaging/@c-arms/documents/download/mdaw/mtqx/~edisp/preoplan-product-brochure-00029745.pdf).
- MSDN. *Developer Network*. [En línea] Microsoft. [Citado el: 20 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=vs.110%29.aspx>.
- MacDonald, Matthew. *Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4.0*. New York : APRESS, 2010.
- MSDN. *Managed extensibility framework*. [En línea] Microsoft. [Citado el: 20 de febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/ee291628.aspx>.
- Polyack, Nick. CodeProject. *MVVM-Pattern-Made-Simple*. [En línea] 30 de octubre de 2012. [Citado el: 17 de febrero de 2014.] <http://www.codeproject.com/Articles/278901/MVVM-Pattern-Made-Simple>.
- González Seco, José Antonio. *El lenguaje de programación C#*.
- MSDN. *C#*. [En línea] 2010. [Citado el: 17 de febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/ff796223.aspx>.
- Oré Bravo, Alexander. CalidadySoftware. *introduccioncmmi*. [En línea] 13 de abril de 2008. [Citado el: 17 de febrero de 2014.] [http://www.calidadysoftware.com/otros/introduccion\\_cmmi.php](http://www.calidadysoftware.com/otros/introduccion_cmmi.php).
- Chrissis, Mary Beth, Konrad, Mike y Shrum, Sandy. *CMMI Guía para integración de procesos y la mejora de productos*. s.l. : Pearson Education, 2009.

El Modelo CMMI. [En línea] noviembre de 2008. [Citado el: 5 de junio de 2014.] <http://www.allsoft.com.mx/recursos/ElModeloCMMI.pdf> .

Hernández Orallo, Enrique. *El Lenguaje Unificado de Modelado (UML)*.

Belloso Cicilia, Claudia Ivonne. *MONOGRAFIA SOBRE LA METODOLOGIA DE DESARROLLO DE SOFTWARE*. El Salvador : Universidad Don Bosco, septiembre 2009.

MSDN. *VisualStudio*. [En línea] Microsoft. [Citado el: 18 de febrero de 2014.] <http://msdn.microsoft.com/es-es/library/vstudio/6b6b1f4%28v=vs.100%29.aspx>.

unity. [En línea] 03 de octubre de 2012. [Citado el: 20 de febrero de 2014.] [http://www.unity.com/index.php?option=com\\_k2&view=item&id=139:ea-main&Itemid=88](http://www.unity.com/index.php?option=com_k2&view=item&id=139:ea-main&Itemid=88).

SparxSystems. [En línea] SparxSystems Pty Ltd. [Citado el: 21 de febrero de 2014.] <http://www.sparxsystems.com.ar/products/ea/7.5/index.html#releasenotes>.

Küng, Stefan, Onken, Lübbe y Large, Simon. *TortoiseSVN Un cliente de Subversion para Windows*.

EcuRed. *Modelo de dominio*. [En línea] 18 de abril de 2012. [Citado el: 26 de marzo de 2014.] [http://www.ecured.cu/index.php/Modelo\\_de\\_dominio](http://www.ecured.cu/index.php/Modelo_de_dominio).

Monferrer Agut, Raúl. *Especificación de Requisitos Software según el estándar de IEEE830*. Castellón, España : Universidad de Jaume I, 2001.

Sommerville, Ian. *Ingeniería de Requisitos. Ingeniería del Software*. Madrid : PEARSON EDUCACIÓN S.A., 2005.

Pressman, Roger S. *Ingeniería del Software: Un enfoque Práctico*. s.l. : McGrawHill. 5ta edición.

EcuRed. *Arquitectura de software*. [En línea] 1 de octubre de 2012. [Citado el: 28 de marzo de 2014.] [http://www.ecured.cu/index.php/Arquitectura\\_de\\_software](http://www.ecured.cu/index.php/Arquitectura_de_software).

Larman, Craig. *UML y Patrones: una introducción al análisis y diseño orientado a objeto y al proceso unificado*. s.l. : Prentice Hall.

Llorente, César de la Torre, y otros. *Guía de Arquitectura N-capas orientada al Dominio con .NET 4.0*. España : Krasis Consulting, 2010.

- Daniele, Marcela. [En línea] 2007. [Citado el: 30 de abril de 2014.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
- SparxSystem. *Diagrama de Componentes UML*. [En línea] [Citado el: 30 de abril de 2014.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html).
- Rubiano, Martha. Scribd. [En línea] 9 de septiembre de 2010. [Citado el: 2 de mayo de 2014.] <http://es.scribd.com/doc/19611312/diagramas-de-despliegue-2222>.
- Marca Huallpara, Hugo Michel y Quisbert Limachi, Nancy Susana. *Diagrama de Despliegue*.
- MSDN. *Revisión de código y estándares de codificación*. [En línea] [Citado el: 6 de junio de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
- Fernández, Alberto. *Estándares de Codificación en C# y Buenas Prácticas de Programación*.
- Lerman, Julia. *Entity Framework*. s.l. : O'REILLY, 2010.
- Reynoso Billy, Carlos. *Introducción a la Ingeniería de Software*. 2004.
- Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. *Arquitecturas de Software*. 2004.
- Schmuller, Joseph. *Aprendiendo UML*.
- MSDN. *Expresiones lambda (Guía de programación de C#)*. [En línea] [Citado el: 25 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/bb397687.aspx>.
- IBM. *IBM Rational Unified Process (RUP)*. [En línea] [Citado el: 01 de junio de 2014.] <http://www-01.ibm.com/software/rational/rup/>.
- MSDN. *Herencia (Guía de programación de C#)*. [En línea] [Citado el: 10 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/ms173149.aspx>.
- MSDN. *Polimorfismo (Guía de programación de C#)*. [En línea] [Citado el: 11 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/ms173152.aspx>.
- Net-informations.com. *How to create Excel file from XML*. [En línea] [Citado el: 03 de febrero de 2014.] <http://csharp.net-informations.com/xml/excel-from-xml.htm>.

## ANEXOS

Anexo 1. Prefijos de los requerimientos no funcionales por categorías.

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
Legal	RNL	RNL 1. Requisito legal
Fiabilidad	RNF	RNF 1. Requisito de fiabilidad
Seguridad	RNS	RNS 1. Requisito de seguridad
Eficiencia	RNE	RNE 1. Requisito de eficiencia
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
Componentes Comprados	RNCC	RNCC 1. Requisito de componentes comprados
Licencia	RNLI	RNLI 1. Requisito de licencia

Anexo 2. Casos de uso expandidos.

- Caso de uso – Confeccionar reporte preoperatorio.

<b>Objetivo</b>	Confeccionar un reporte preoperatorio que contiene las imágenes e
-----------------	---

	información referentes al estudio del caso.
<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite confeccionar un reporte luego de que se haya terminado el proceso de la planificación, este reporte incluye las imágenes e información que se obtuvieron como resultado del estudio del caso.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.1
<b>Precondiciones</b>	Se ha cargado una imagen en el visor.
<b>Postcondiciones</b>	Se obtuvo el reporte de la planificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Confeccionar reporte de planificación</b>	
<ol style="list-style-type: none"> <li>1. El especialista selecciona la pestaña Reporte.</li> <li>2. El especialista selecciona la opción Preoperatorio.</li> <li>3. El especialista especifica la siguiente información: <ul style="list-style-type: none"> <li>• Posición del cuerpo</li> <li>• Anestesia</li> <li>• Radiografía</li> <li>• Injerto óseo</li> <li>• Información adicional</li> </ul> </li> <li>4. El especialista gestiona los comentarios del informe. (Ver CU Gestionar comentarios).</li> <li>5. El sistema guarda los comentarios realizados.</li> <li>6. El especialista selecciona la imagen a adjuntar al informe. (Ver CU Seleccionar imagen a incluir en el reporte).</li> </ol>	

<p>7. El sistema salva toda la información correspondiente al reporte.</p> <p>8. El especialista exporta el reporte. (Ver CU Exportar reporte)</p> <p>9. El sistema muestra el reporte en el formato especificado.</p> <p>10. Termina el caso de uso.</p>		
<b>Relaciones</b>	<b>CU Incluidos</b>	CU Exportar reporte.
	<b>CU Extendidos</b>	CU Seleccionar imagen a incluir en el reporte. CU Gestionar comentarios.
<b>Requisitos funcionales</b>	<b>no</b>	RNU 1, RNIU 1
<b>Asuntos pendientes</b>		No aplica

- Caso de uso – Confeccionar reporte postoperatorio.

<b>Objetivo</b>	Confeccionar un reporte postoperatorio
<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite confeccionar un reporte luego de que se haya realizado la cirugía.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Referencias</b>	RF 1.2
<b>Precondiciones</b>	Se ha cargado una imagen en el visor.
<b>Postcondiciones</b>	Se obtuvo el reporte postoperatorio
<b>Flujo de eventos</b>	
<b>Flujo básico Confeccionar reporte postoperatorio</b>	

1. El especialista selecciona la pestaña Reporte.
2. El especialista selecciona la opción Postoperatorio.
3. El especialista especifica la siguiente información:
  - Clasificación de la cirugía
  - Nombre del doctor
4. El especialista gestiona los comentarios del informe. (Ver CU Gestionar comentarios).
5. El sistema guarda los comentarios realizados.
6. El especialista selecciona la imagen a adjuntar al informe. (Ver CU Seleccionar imagen a incluir en el reporte).
7. El sistema salva toda la información correspondiente al reporte.
8. El especialista exporta el reporte. (Ver CU Exportar reporte)
9. El sistema muestra el reporte en el formato especificado.
10. Termina el caso de uso.

<b>Relaciones</b>	<b>CU Incluidos</b>	CU Exportar reporte.
	<b>CU Extendidos</b>	CU Seleccionar imagen a incluir en el reporte. CU Gestionar comentarios.
<b>Requisitos no funcionales</b>	no	RNU 1, RNIU 1
<b>Asuntos pendientes</b>		No aplica

- Caso de uso – Exportar reporte.

<b>Objetivo</b>	Exportar un reporte al formato PDF, RTF o HTML.
<b>Actores</b>	Especialista (Inicia).
<b>Resumen</b>	El caso de uso permite exportar en formato RTF, PDF o HTML un

	reporte que previamente fue confeccionado.	
<b>Complejidad</b>	Baja.	
<b>Prioridad</b>	Crítico.	
<b>Referencias</b>	RF 1.3, RF 1.4, RF 1.5	
<b>Precondiciones</b>	Debe de haberse cargado una imagen en el visor.	
<b>Postcondiciones</b>	Se obtuvo un reporte en formato RTF, PDF o HTML.	
<b>Flujo de eventos</b>		
<b>Flujo básico Exportar reporte</b>		
<ol style="list-style-type: none"> <li>El especialista selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>Exportar a PDF.</li> <li>Exportar a RTF.</li> <li>Exportar a HTML.</li> </ul> </li> <li>El sistema muestra una ventana para especificar el nombre del reporte y la ubicación en donde se guardará.</li> <li>El especialista escribe el nombre del reporte, selecciona una ubicación y presiona el botón aceptar.</li> <li>El sistema guarda el reporte en el formato especificado y lo muestra automáticamente.</li> <li>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>	RNDI 5, RNE 1, RNF 2
<b>Asuntos pendientes</b>	No aplica	

- Caso de uso - Gestionar comentario.

<b>Objetivo</b>	Gestionar un comentario para el reporte preoperatorio o postoperatorio.	
<b>Actores</b>	Especialista (inicia).	
<b>Resumen</b>	El caso de uso permite insertar, modificar o eliminar un comentario. Todas estas operaciones se pueden realizar en una misma ventana.	
<b>Complejidad</b>	Baja.	
<b>Prioridad</b>	Crítico.	
<b>Referencias</b>	RF 1.6	
<b>Precondiciones</b>	Debe de haberse cargado una imagen en el visor.	
<b>Postcondiciones</b>	Se gestionó un comentario.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar comentario</b>		
<ol style="list-style-type: none"> <li>1. El especialista selecciona la opción Comentario.</li> <li>2. El sistema muestra una ventana</li> <li>3. El especialista introduce, modifica o cancela el comentario.</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>	RNDI 5	
<b>Asuntos pendientes</b>	No aplica	

- Caso de uso – Seleccionar imagen a incluir en el reporte.

<b>Objetivo</b>	Permite especificar la imagen que formará parte del reporte.	
<b>Actores</b>	Especialista (inicia).	
<b>Resumen</b>	El caso de uso brinda la posibilidad de seleccionar la imagen de interés que debe aparecer en el reporte.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico.	
<b>Referencias</b>	RF 1.7	
<b>Precondiciones</b>	Debe de haberse cargado una imagen en el visor.	
<b>Postcondiciones</b>	Se seleccionaron las imágenes a incluir en el reporte.	
<b>Flujo de eventos</b>		
<b>Flujo básico Seleccionar imágenes a imprimir en el reporte</b>		
<ol style="list-style-type: none"> <li>1. El especialista marca como favorita la imagen.</li> <li>2. El sistema agrega la imagen al reporte que se exporta.</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 funcionales</b>	<b>no</b>	RNU 2, RNDI 5
<b>Asuntos pendientes</b>	No aplica	

## Anexo 3. Descripción de clases.

- Clase ReportTabVMConfig

**Propósito:** Configurar el ViewModel de la pestaña de reportes.

<b>Nombre:</b> ReportTabVMConfig	
<b>Tipo de clase:</b> controladora	
Atributo	Tipo
categoryName	string
chapterName	string
commentUI	CommentInfoUI
firstName	string
groupName	string
lastName	string
procedureName	string
viewer	IViewer
categoryPos	int
chapterPos	int
groupPos	int
procedPos	int
<b>Para cada responsabilidad:</b>	
Nombre:	AditionalInfCmd(item: object)
Descripción:	Recoge los datos que son contemplados como información adicional en el reporte preoperatorio según los elementos seleccionados.
Nombre:	AnesthesiaCmd(item: object)

Descripción:	Inicializa el tipo de anestesia que se utilizará en la cirugía según el elemento seleccionado.
Nombre:	BoneDraftCmd(item: string)
Descripción:	Inicializa el tipo de injerto óseo.
Nombre:	CategoryChanged(index: int)
Descripción:	Actualiza la lista de procedimientos en el TAB de reporte de tipo postoperatorio e inicializa el nombre de la categoría según el índice seleccionado.
Nombre:	ChapterChanged(index: int)
Descripción:	Actualiza la lista de grupos en el TAB de reporte de tipo postoperatorio e inicializa el nombre del capítulo según el índice seleccionado.
Nombre:	Configure ()
Descripción:	Configura la instancia del ViewModel para el contrato actual.
Nombre:	ExportReport (reportExt: string)
Descripción:	Crea el reporte en el formato deseado según la selección realizada.
Nombre:	ExtractPatientData()
Descripción:	Inicializa los datos del paciente en el reporte, estos datos son contenidos originalmente en la cabecera de la imagen DICOM.
Nombre:	FullCategory ()
Descripción:	Inicializa la lista de categorías en el TAB de reporte de tipo postoperatorio.
Nombre:	FullChapter()
Descripción:	Inicializa la lista de categorías en el TAB de reporte de tipo postoperatorio.
Nombre:	FullGroup()
Descripción:	Inicializa la lista de grupos en el TAB de reporte de tipo postoperatorio.
Nombre:	FullProcedures()
Descripción:	Inicializa la lista de procedimientos en el TAB de reporte de tipo postoperatorio.
Nombre:	GetNameAndIsChecked(uiItem: object)
Descripción:	Retorna el nombre del objeto y si está seleccionado o no.
Nombre:	GroupChanged(index: int)

Descripción:	Actualiza la lista de categorías en el TAB de reporte de tipo postoperatorio e inicializa el nombre del grupo según el índice seleccionado.
Nombre:	PatientPositionCmd(item: object)
Descripción:	Inicializa los campos referentes a la posición en que será operado el paciente, la descripción de esta y la imagen que la identifica, todo esto en el reporte preoperatorio y según los valores que posee el objeto seleccionado.
Nombre:	ProcedureChanged(index: int)
Descripción:	Inicializa el nombre del procedimiento en el reporte postoperatorio según el índice seleccionado.
Nombre:	SelectPosOpReporter()
Descripción:	Oculto los elementos referentes al reporte preoperatorio mostrando solo los correspondientes al reporte postoperatorio en el TAB de reportes, e inicializa una instancia de un reporte postoperatorio.
Nombre:	SelectPreOpReporter()
Descripción:	Oculto los elementos referentes al reporte postoperatorio mostrando solo los correspondientes al reporte preoperatorio en el TAB de reportes, e inicializa una instancia de un reporte preoperatorio.
Nombre:	ShowCommentsInfoUI()
Descripción:	Crea una instancia de una clase comentario y muestra una ventana para la confección de este.
Nombre:	XRayCmd(item: object)
Descripción:	Establece el tipo de radiografía a utilizar e inicializa su valor en el reporte preoperatorio.

- Clase PreReport

**Propósito:** Definir la información del reporte preoperatorio.

<b>Nombre:</b> PreReport.
<b>Tipo de clase:</b> Entidad.

Atributo	Tipo
No procede.	No procede.
<b>Para cada responsabilidad:</b>	
Nombre:	PreReport ()
Descripción:	Inicializa una nueva instancia de la clase.

- Clase PostReport

**Propósito:** Definir la información del reporte postoperatorio.

<b>Nombre:</b> PostReport.	
<b>Tipo de clase:</b> Entidad.	
Atributo	Tipo
No procede.	No procede.
<b>Para cada responsabilidad:</b>	
Nombre:	PostReport ()
Descripción:	Inicializa una nueva instancia de la clase.

- Clase ReportExtension

**Propósito:** Crear reportes de diferentes tipos (preoperatorio y postoperatorio) y diferentes formatos (.pdf, .rtf, .html).

<b>Nombre:</b> ReportExtension	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
currentHtmlImgDirFullname	string
currentHtmlImgDirName	string
bigFont	Font

bigSize	Single
docExt	string
fontName	string
htmlExt	string
mediumFont	Font
mediumSize	Single
normalFont	Font
normalSize	Single
pdfExt	string
smallSize	Single
smallFont	Font
<b>Para cada responsabilidad:</b>	
Nombre:	CreateDocument(size: Rectangle, leftMargin: float, rightMargin: float, topMargin: float, bottonMargin: float)
Descripción:	Crea un documento con tamaño y márgenes determinados.
Nombre:	GetFont(other: Font, style: int)
Descripción:	Obtiene una nueva fuente de texto con tamaño y estilo propio.
Nombre:	GetLocalizedText(report: ReportFather, key: string)
Descripción:	Obtiene del diccionario de traducción de la librería Daf la traducción al idioma que se utiliza, de la cadena entrante.
Nombre:	PutExtension(file: string, ext: string)
Descripción:	Obtiene un archivo con el nombre del archivo entrado por parámetro pero con la extensión deseada.
Nombre:	ReportExtensions()
Descripción:	Inicializa una instancia de la clase.
Nombre:	SaveToHTML(report: ReportFather, path: string)

Descripción:	Crea un documento en formato .html con la información del elemento report y lo guarda en la dirección especificada.
Nombre:	SaveToPDF(report: ReportFather, path: string)
Descripción:	Crea un documento en formato .pdf con la información del elemento report y lo guarda en la dirección especificada.
Nombre:	SaveToRtf(report: ReportFather, path: string)
Descripción:	Crea un documento en formato .rtf con la información del elemento report y lo guarda en la dirección especificada.
Nombre:	WriteReport(document: Document, writer: DocWriter, report: ReportFather)
Descripción:	Crea un documento genérico con el contenido del elemento report de manera estructurada.

- Clase ITextSharpExtension

**Propósito:** Añadir líneas y elementos a un párrafo o documento.

<b>Nombre:</b> ITextSharpExtension.	
<b>Tipo de clase:</b> Entidad.	
<b>Atributo</b>	<b>Tipo</b>
emptyLine	Phrase
<b>Para cada responsabilidad:</b>	
Nombre:	AddLine(p: Paragraph, item: object, linesAfter: int)
Descripción:	Añade en una nueva línea a un párrafo, el objeto especificado.
Nombre:	AddEmptyLines(doc: Document, linesCount: int)
Descripción:	Añade al documento especificado una línea en blanco.
Nombre:	Add(doc: Document, items: IElements[])
Descripción:	Añade al documento de manera consecutiva los elementos determinados.

## GLOSARIO DE TÉRMINOS

Angiografía: es un examen de diagnóstico por imagen que estudia los vasos circulatorios que no son visibles mediante la radiología convencional. Su nombre procede de las palabras griegas *angeion*, "vaso", y *graphien*, "descripción".

Componente acetabular: es el implante que se coloca en el acetábulo o porción articular cóncava de la superficie de la pelvis, formada por el ilion, el isquion y el pubis. Su función es acoger la cabeza del fémur y formar la articulación de la cadera.

Componente femoral o vástago: es el implante que se introduce en el interior de la cavidad labrada del fémur con el objetivo de conectarlo con el acetábulo y formar la articulación de la cadera.

Fluoroscopia: es una forma de diagnóstico radiológico que a través de rayos X y con la ayuda de un agente o medio de contraste, permite al médico visualizar el órgano o área de interés del paciente. Es un estudio de las estructuras del cuerpo en movimiento similar a una película de rayos X. Se hace pasar un haz continuo de rayos X a través de la parte del cuerpo que va a examinarse, y se transmite a un monitor parecido a una televisión, de forma que pueda verse en detalle la parte del cuerpo y su movimiento en tiempo real.

Osteomioneuroarticular: sistema formado por el conjunto de órganos que intervienen en el movimiento de los cuerpos vivos, o sea la mecánica animal: huesos, músculos y articulaciones, así como por los nervios y terminaciones nerviosas, que se encargan de generar y modular las órdenes motoras.

Resonancia magnética: Las resonancias magnéticas son un análisis seguro e indoloro en el cual se utiliza un campo magnético y ondas de radio para obtener imágenes detalladas de los órganos y las estructuras del cuerpo humano.

Ultrasonido: es un sonido cuya frecuencia de vibraciones es superior al límite perceptible por el oído humano (20.000 Hz). El ultrasonido tiene múltiples aplicaciones y se utiliza en diversas áreas de la Industria y la Medicina. En la medicina el ultrasonido constituye una técnica de visualización de la anatomía humana.