

**Universidad de las Ciencias Informáticas**

**Facultad 7**



**Título: Casandra Web. Aplicación para la visualización y  
gestión de imágenes médicas.**

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

**Autor:** Alejandro Arias Naranjo

**Tutor:** Msc. Martha Yurina Ambruster Crespo

Ciudad de la Habana, Junio 2007

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Procesamiento de Imágenes y Señales (GPI) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 20 días del mes de junio del año 2007.

---

Diplomante: Alejandro Arias Naranjo

---

Tutor: Martha Yurina Ambruster Crespo

## DATOS DE CONTACTO

TUTOR: Msc. Martha Yurina Ambruster Crespo ([mambruster@uci.cu](mailto:mambruster@uci.cu))

Profesora graduada de Ingeniería Informática en el Instituto Superior Politécnico José Antonio Echeverría. Posee categoría docente de Instructor. Ha impartido las asignaturas de Programación 2, Programación 3 y Gráfico por Computadoras. Es profesora de la Facultad 7 y en la actualidad pertenece al Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI), en el que se desempeña como Jefe de Proyecto.

DEDICATORIA

A mi familia, que siempre ha estado presente en todo momento.

Gracias a mi mamá, mi papá, a Gilbe y a Abue.

A mi prima Soraya, que la quiero como si fuera mi tía.

A mis amistades que han ayudado de una forma u otra en el trabajo.

## RESUMEN

En el presente trabajo se propone como objetivo, desarrollar una aplicación web para la visualización y gestión de imágenes médicas, que aprovechando la arquitectura cliente/servidor permita a los usuarios abstraerse de las dificultades existentes en las estaciones de visualización de las soluciones PACS (Sistemas para la comunicación y almacenamiento de imágenes médicas) en Cuba.

Para la realización del sistema se hizo un estudio de las nuevas tecnologías que son utilizadas para el desarrollo de aplicaciones Web. Como resultado de esta investigación, se decidió utilizar la tecnología ASP.NET que brinda la Plataforma .NET de Microsoft con el lenguaje C# y PostgreSQL como gestor de base de datos.

La aplicación presenta las características siguientes: permite visualizar imágenes médicas, aplicar diferentes tipos de filtros, brillo y contraste, paletas de colores, realizar mediciones (ángulos, perímetros, distancias, áreas, densidades) sobre la imagen, reproducir imágenes multiframe. Visualizar el histograma de las imágenes tanto RGB como en tonos de grises así como datos estadísticos. Así como realizar la búsqueda, obtención y almacenamiento de imágenes en un servidor compatible. El especialista puede emitir un reporte con el diagnóstico basado en el estudio de las imágenes, que es enviado a un repositorio o impreso en documento Word.

## PALABRAS CLAVES

**DICOM, PACS, WEB**

## TABLA DE CONTENIDOS

<b>DEDICATORIA</b> .....	<b>I</b>
<b>RESUMEN</b> .....	<b>II</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
1.1. Términos y definiciones .....	4
1.1.1. PACS.....	4
1.1.2. DICOM .....	5
1.2. Tecnologías de desarrollo.....	5
1.2.1. Arquitectura Cliente/Servidor .....	5
1.2.2. Servicios Web.....	7
1.2.3. Plataformas de desarrollo .....	7
1.2.3.1. Microsoft .NET .....	7
1.2.3.2. Java 2 Enterprise Edition (J2EE) .....	9
1.2.3.3. Comparación entre Microsoft .NET y J2EE .....	9
1.2.3.3.1. Lenguajes de programación .....	10
1.2.3.3.2. Herramientas de desarrollo .....	10
1.2.3.3.3. Portabilidad .....	10
1.2.3.3.4. Comunicación .....	10
1.2.3.3.5. Servicios Web .....	11
1.2.3.3.6. Páginas Web con contenido activo.....	11
1.2.4. ASP.NET .....	12
1.2.5. C#.....	13
1.3. Gestores de bases de datos .....	13
1.3.1. Oracle.....	13
1.3.2. Microsoft SQL Server.....	14
1.3.3. MySQL .....	14
1.3.4. PostgreSQL .....	15

1.3.5.	Fundamentación de la selección.....	17
1.4.	Estado del arte sobre sistemas similares.....	17
1.4.1.	En el mundo .....	17
1.4.2.	En Cuba.....	19
1.4.3.	Cassandra Web.....	19
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>		<b>20</b>
2.1.	Problema científico y situación problemática .....	20
2.2.	Objeto de automatización .....	21
2.3.	Sistema propuesto.....	21
2.4.	Modelo de negocio .....	22
2.4.1.	Representación de los casos de uso del negocio .....	22
2.4.1.1.	Actores .....	22
2.4.1.2.	Trabajadores .....	22
2.4.1.3.	Descripción de los casos de uso.....	23
2.4.2.	Diagramas del modelo del negocio .....	24
2.4.2.1.	Diagrama de casos de uso del negocio .....	24
2.4.2.2.	Diagrama de actividad del negocio .....	25
2.4.2.3.	Diagrama de clases del modelo de objeto .....	26
2.5.	Especificación de los requisitos de software .....	26
2.5.1.	Dependencias y relaciones.....	26
2.5.2.	Requerimientos Funcionales.....	27
2.5.3.	Requisitos no funcionales .....	28
2.6.	Definición de los casos de uso.....	29
2.6.1.	Definición de los actores.....	30
2.6.2.	Listado de los casos de uso.....	30
2.6.3.	Descripción de los casos de uso.....	31
2.6.4.	Diagrama de casos de uso .....	35
2.6.5.	Casos de uso por ciclo.....	38
2.6.5.1.	Ciclo 1 de desarrollo .....	38

2.6.5.2. Ciclo 2 de desarrollo .....	38
2.6.5.3. Ciclo 3 de desarrollo .....	39
2.6.6. Casos de uso expandidos.....	39
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>40</b>
3.1. Análisis .....	40
3.1.1. Definición del modelo de análisis .....	40
3.1.2. Diagrama de clases de análisis .....	41
3.2. Diseño .....	41
3.2.1. Diagramas de interacción .....	41
3.2.2. Diagramas de clases de diseño .....	41
3.2.2.1. Descripción de las clases .....	41
3.2.3. Diseño de la base de datos.....	42
3.2.3.1. Diagrama entidad de relación de la base de datos .....	42
3.2.3.2. Descripción de las tablas .....	43
<b>CAPÍTULO 4: IMPLEMENTACIÓN .....</b>	<b>44</b>
4.1. Diagrama de despliegue .....	44
4.2. Diagrama de componentes.....	45
<b>CONCLUSIONES .....</b>	<b>47</b>
<b>RECOMENDACIONES.....</b>	<b>48</b>
<b>REFERENCIA BIBLIOGRÁFICA .....</b>	<b>49</b>
<b>BIBLIOGRAFÍA.....</b>	<b>50</b>
<b>ANEXOS .....</b>	<b>52</b>
Anexo 1 Casos de Uso Expandidos .....	52
Anexo 2 Modelo de Clases del Análisis .....	59
Anexo 3 Diagramas de Interacción.....	60
3.2.1.1. Diagrama de interacción – Visualizar imágenes .....	60
3.2.1.2. Diagrama de interacción – Generar reporte .....	61



3.2.1.3.	Diagrama de interacción – Buscar imágenes en servidor DICOM.....	62
3.2.1.4.	Diagrama de interacción – Obtener imágenes en servidor DICOM.....	63
3.2.1.5.	Diagrama de interacción – Almacenar imágenes en servidor DICOM.....	64
3.2.1.6.	Diagrama de interacción – Aplicar filtros.....	65
Anexo 4	Diagramas de Clases .....	66
3.2.2.1.	Diagrama de clases del diseño – Visualizar imágenes .....	66
3.2.2.2.	Diagrama de clases del diseño – Generar reporte .....	67
3.2.2.3.	Diagrama de clases del diseño – Buscar imágenes en servidor DICOM.....	68
3.2.2.4.	Diagrama de clases del diseño – Obtener imágenes en servidor DICOM.....	69
3.2.2.5.	Diagrama de clases del diseño – Almacenar imágenes en servidor DICOM.....	70
3.2.2.6.	Diagrama de clases del diseño – Aplicar filtros.....	71
Anexo 5	Descripción de las Clases .....	72
Anexo 6	Descripción de las Tablas .....	80
<b>GLOSARIO</b> .....		<b>83</b>

## INTRODUCCIÓN

Con el surgimiento y evolución de las tecnologías para equipos de adquisición de imágenes médicas, modalidades como la Tomografía Computarizada, Resonancia Magnética, Ultrasonido entre otras. Los especialistas fueron beneficiándose cada vez más, teniendo la posibilidad de ver y estudiar con más detalles los casos. Se le suman otras tecnologías como la red de comunicación, las computadoras personales y medios de almacenamiento. Haciendo factible la transmisión de imágenes médicas a distancia.

Surgen así, los Sistemas de Transmisión y Almacenamiento de Imágenes (PACS), que están formados fundamentalmente por equipos de adquisición, la infraestructura de comunicación, servidor y estaciones de visualización. Para favorecer el desarrollo y la expansión de los PACS la Asociación de Radiólogos Americanos (ACR); en conjunto con la Asociación Nacional de Empresas Eléctricas de EEUU (NEMA); define el estándar DICOM que establece como realizar la transferencia de imágenes digitales e información médica entre computadoras, la comunicación digital entre equipos de diagnóstico terapéutico y entre sistemas de diferentes fabricantes.

Nuestro país no ha quedado al margen de este desarrollo, desde hace un tiempo atrás, ha adquirido e instalado equipos de última generación en diferentes hospitales; se han desarrollado dos software que cumplen con el papel de PACS. Pero esos sistemas presentan problemas de configuración y mantenimiento en el momento de instalación o cuando dejan de prestar el servicio requerido. Por tal razón, el personal especializado tiene que ir a cada uno de las estaciones de visualización para poner el sistema a funcionar. Esto requiere de un gran esfuerzo en la medida que aumentan las estaciones y su dispersión geográfica es mayor.

Teniendo en cuenta, las razones mencionadas anteriormente se plantea el problema científico: ¿Cómo resolver las insuficiencias de configuración y mantenimiento de los sistemas para la búsqueda, obtención y visualización de imágenes médicas, existentes en nuestro país?

Para resolver este problema se decide diseñar una aplicación web para visualizar, almacenar y transmitir las imágenes que son generadas por estos equipos. El especialista contaría con herramientas para poder estudiar las imágenes con mayor detenimiento, se buscarían en un servidor DICOM

compatible, para luego, pueda emitir un diagnóstico, que sería enviado a un repositorio o ser impreso en formato Word.

### **Objeto de Estudio**

Proceso de visualización y gestión de imágenes médicas.

### **Campo de Acción**

Proceso de visualización y gestión de imágenes médicas a través de un PACS en hospitales con áreas de imagenología.

### **Objetivo General**

Desarrollar una aplicación web para la visualización y la gestión de imágenes médicas.

### **Tareas**

- Realizar un estudio de la situación actual en los hospitales, en cuanto al software con que cuentan hoy en día para la visualización y gestión de imágenes médicas.
- Analizar los problemas que se presentan en los sistemas similares de tipo escritorio, con el objetivo de optimizar el entorno de trabajo en la parte cliente.
- Realizar una captura de requisitos de acuerdo a las necesidades de los especialistas.
- Proponer la arquitectura de la aplicación.
- Hacer el análisis y diseño de la aplicación web manteniendo la compatibilidad con el estándar DICOM.
- Implementar una aplicación web para la visualización y gestión de imágenes médicas.

Si se desarrolla este sistema sería el primero de su tipo en nuestro país, los especialistas aprovechando las ventajas de la web, tendrían la posibilidad de analizar las imágenes cada vez que lo necesiten, sin importar momento ni lugar, realizándole todas la operaciones que brinda el sistema para lograr un mejor diagnostico del caso.

Para la realización del sistema se hizo un estudio de las nuevas tecnologías que son utilizadas para el desarrollo de aplicaciones Web. Como resultado de esta investigación, se decidió utilizar la tecnología ASP.NET que brinda la Plataforma .NET de Microsoft con el lenguaje C# y PostgreSQL como gestor de base de datos.

El documento está conformado por cuatro capítulos, a través de los cuales se ofrece una información más pormenorizada del objeto de la investigación:

En el Capítulo 1: Fundamentación Teórica, se realiza un estudio acerca de las principales tendencias de la visualización y gestión de imágenes médicas a nivel internacional y nacional. Así como las técnicas, tecnologías y software más usados en la actualidad y en las que se apoya para la solución del problema y sobre los que es necesario profundizar.

El Capítulo 2: Características del Sistema, se describen los procesos estudiados, especificando en el objeto de automatización, así como la propuesta del sistema. También incluye los requerimientos, tanto funcionales como no funcionales, los modelos del negocio y del sistema.

El Capítulo 3: Análisis y diseño del sistema, plantea los aspectos internos del sistema y se representan los diagramas de clases de interacción y de entidad – relación de la base de datos. Además, se describen las clases utilizadas en el diseño y de las tablas que pertenecen a la base de datos.

El Capítulo 4: Implementación, muestra los diagramas de despliegue y los de componentes, artefactos que son generados durante este flujo de trabajo.

A continuación, aparecen las Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y Glosario de Términos.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La evolución de tecnologías para equipos de adquisición de imágenes médicas ha beneficiado a los especialistas que tienen la posibilidad de visualizar y gestionar las imágenes médicas con más detalles. Surgen así, los Sistemas de Transmisión y Almacenamiento de Imágenes (PACS), que basados por el Estándar DICOM permite la transferencia de imágenes digitales e información médica entre computadoras y la comunicación digital entre equipos de diagnóstico terapéutico y entre sistemas de diferentes fabricantes.

En este capítulo se definen los términos fundamentales que son utilizados en el trabajo. Se exponen las características de las tecnologías y herramientas utilizadas para el desarrollo de la aplicación. Se realiza un estudio del estado del arte de sistemas similares que existen dentro y fuera del país, de tal forma que se pueda fundamentar la realización de este proyecto.

### **1.1. Términos y definiciones**

#### **1.1.1. PACS**

La función del PACS es la comunicación con todas las modalidades de imágenes (tomografía, ecografía, mamografía, etc.) y estaciones de trabajo de la red de imagen, almacenando las imágenes enviadas por las modalidades, y enviando los estudios solicitados a las estaciones de visualización.

La infraestructura de los PACS está compuesta por un esqueleto básico de hardware (interfaz de dispositivos de imágenes, dispositivos de almacenamiento, computadoras, redes de comunicación digital y sistemas de visualización). Estos componentes se integran a través de subsistemas de software estandarizados y flexibles para la comunicación, manejo de bases de datos, almacenamiento, planificación de tareas, comunicación entre procesadores, manejo de errores y monitoreo de red. Los módulos de software de la infraestructura engloban suficiente comprensión y cooperación a nivel de sistema para posibilitar que los componentes trabajen de forma integrada como un sistema y no de forma aislada.

### **1.1.2. DICOM**

Para favorecer el desarrollo y la expansión de los PACS la Asociación de Radiólogos Americanos (ACR); en conjunto con la Asociación Nacional de Empresas Eléctricas de EEUU (NEMA); se reúnen en 1983, en 1985 definen la primera versión del estándar DICOM que establece como realizar la transferencia de imágenes digitales e información médica entre computadoras, la comunicación digital entre equipos de diagnóstico terapéutico y entre sistemas de diferentes fabricantes.

DICOM (Digital Imaging and Communication in Medicine) es el estándar que rige todo lo relacionado con el manejo, almacenamiento, impresión y transmisión de las imágenes médicas, estableciendo un formato de fichero y un protocolo de comunicación por la red. De esta forma DICOM permite la integración de los equipos médicos, estaciones de visualización y servidores de múltiples proveedores que participan en el sistema de almacenamiento y comunicación de imágenes médicas. [ACR, 2004]

## **1.2. Tecnologías de desarrollo**

### **1.2.1. Arquitectura Cliente/Servidor**

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos los recursos de máquina, de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspassa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de las computadoras personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### 1.2.2. Servicios Web

Los Servicios Web XML permiten que las aplicaciones compartan información, que además invoquen funciones de otras aplicaciones, independientemente de cómo se hayan creado, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles sean los dispositivos utilizados para obtener acceso a ellas. Aunque los Servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Estos son invocados desde Internet por medio de protocolos estándar tales como SOAP (Protocolo de acceso a objetos simple, Simple Object Access Protocol); XML (Lenguaje Extensible de Marcas, Extensible Markup Language); UDDI (Descripción, descubrimiento e Integración Universales, Universal Description, Discovery, and Integration) que están definidas por organizaciones de estándares públicos como el consorcio W3C.

### 1.2.3. Plataformas de desarrollo

#### 1.2.3.1. Microsoft .NET

Esta plataforma provee los cimientos para la nueva generación de software. Utiliza los Servicios Web como un medio para poder operar entre distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

El Framework de .Net es una infraestructura que reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son: [RECIO, 2004]

- Lenguajes de compilación
- BCL (Biblioteca de Clases Base, Base Class Library)



- CLR (Entorno Común de Ejecución, Common Language Runtime)

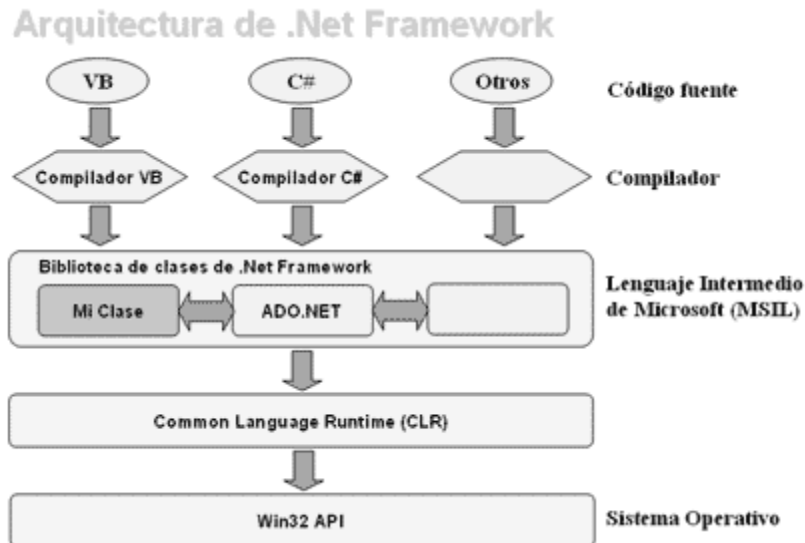


Fig 1 – Arquitectura de .NET Framework.

.NET Framework permite desarrollar en más de 30 lenguajes de programación y a pesar de que cada uno tiene características propias, es posible implementar cualquier tipo de aplicación basados en esta plataforma. Dentro de los lenguajes que ofrece el marco de trabajo de .NET son C#, C++, Visual Basic, Turbo Delphi, J#, Perl, Python, Fortran, Cobol, entre otros más[RECIO, 2004].

La biblioteca de clases base proporciona un conjunto unificado, orientado a objetos, jerárquico y extensible de las bibliotecas de clases (API) para que los desarrolladores la utilicen. La BCL está compuesta por cientos de clases que manejan la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones, algunas como: interacción con los dispositivos periféricos, manipulación de datos de distintos orígenes (ADO.NET), parcelas de seguridad, administración de memoria, cifrado de datos, operaciones aritméticas, manejo y administración de excepciones, transmisión y recepción de datos por distintos medios (XML, TCP/IP), administración de componentes Web que corren en el servidor como en el cliente (ASP.NET), interacción con otras aplicaciones entre otras.

El Entorno Común de Ejecución tiene un papel tanto en el tiempo de desarrollo como en el de ejecución de un componente. Mientras el componente se está ejecutando, el CLR es responsable de la administración de la asignación de memoria, arranque y terminación de hilos y procesos, aplicación de la

política de seguridad, así como de la satisfacción de cualquier dependencia que el componente pueda tener de otros componentes. En tiempo de desarrollo el papel del CLR cambia ligeramente: puesto que automatiza tantas tareas (como la gestión de memoria, por ejemplo). El CLR hace la experiencia del desarrollador muy sencilla.

La estructura de .NET en resumen ofrece distintas ventajas que pueden ser utilizadas por los programadores:

- Un modelo de programación consistente.
- Compatibilidad directa con cuestiones de seguridad.
- Operaciones simplificadas de desarrollo.
- Sencilla implementación y mantenimiento de las aplicaciones robustas y escalables.

#### 1.2.3.2. *Java 2 Enterprise Edition (J2EE)*

Java 2 Enterprise Edition (J2EE) es una arquitectura multicapa para implementar aplicaciones de tipo empresarial y aplicaciones basadas en la Web. Esta tecnología soporta una gran variedad de tipos de aplicaciones desde aplicaciones Web de gran escala a pequeñas aplicaciones cliente-servidor. El objetivo principal de la tecnología J2EE es crear un simple modelo de desarrollo para aplicaciones empresariales utilizando componentes basados en el modelo de aplicación. En este modelo dichos componentes utilizan servicios proporcionados por el contenedor, que de otro modo tendrían que estar incorporados en el código de la aplicación[MCCALLUM, 2004].

#### 1.2.3.3. *Comparación entre Microsoft .NET y J2EE*

En este epígrafe se realizará una comparación entre estas dos plataformas, teniendo en cuenta que son las dos más utilizadas por las empresas, para desarrollar sus aplicaciones y de esta forma dar argumentos para fundamentar una selección entre ellas. Para realizarla se determinan algunos elementos que son imprescindibles tener en cuenta para esta comparación.

#### 1.2.3.3.1. *Lenguajes de programación*

En el entorno .Net da la posibilidad de emplear múltiples lenguajes de programación, mientras que J2EE sólo trabaja con uno: Java. Para algunos desarrolladores tener un solo lenguaje para programar puede ser lo ideal, pero la realidad es que esta alta diversidad de lenguajes es obligatoria por la misma variedad de las necesidades de los programadores. Por otro lado, .Net posibilita así que programadores de terceros lenguajes pasen a esta plataforma reduciendo el tiempo de aprendizaje y entrenamiento. Por todo lo anterior se considera por muchos especialistas que la variedad de los lenguajes de programación es una ventaja muy importante de .Net frente a J2EE.[ABIÁN, 2002]

#### 1.2.3.3.2. *Herramientas de desarrollo*

Las herramientas de desarrollo incluidas por Microsoft en su Visual Studio .Net son simples, intuitivas y sencillas de manejar, mientras que herramientas equivalentes en J2EE son más complejas y difíciles de utilizar. Cualquier programador medio/avanzado se adaptará rápidamente a la forma de programar aplicaciones en Visual Studio .Net, al igual que sucedía con versiones anteriores.[ABIÁN, 2002]

#### 1.2.3.3.3. *Portabilidad*

En el caso de J2EE gracias al entorno de ejecución (JRE) puede correr sobre diferentes plataformas y sistemas operativos. Por su parte .NET funciona en Windows. Aunque con el proyecto Mono es posible la implementación de .NET en Linux y FreeBSD, aunque se prevé dar soporte a más plataformas.

#### 1.2.3.3.4. *Comunicación*

.NET utiliza SOAP para comunicar aplicaciones. Esto permite comunicar aplicaciones hechas en cualquier tecnología o plataforma con otras. Sin embargo J2EE no permite la intercomunicación entre sus aplicaciones de una forma tan sencilla como .NET.

#### 1.2.3.3.5. *Servicios Web*

Microsoft ha impulsado con gran energía los servicios web y resaltado su importancia entre toda la comunidad de desarrolladores. La plataforma .Net se ha diseñado considerando los servicios web. Siendo estos servicios propios de la plataforma y ofrece una nueva versión de ASP, ASP .Net, que puede considerarse un entorno de programación "de verdad" en lugar de un entorno basado en scripts.

Mientras que J2EE no tuvo en cuenta estas ventajas en sus aplicaciones, aunque respondió con el lanzamiento del Java Web Services Developer Pack. De todos modos, la facilidad, rapidez y sencillez con la que se pueden construir servicios Web con el Asistente de Servicios Web de Visual Studio .Net son muy superiores a las de las herramientas para construir Servicios Web dentro del entorno de J2EE.

#### 1.2.3.3.6. *Páginas Web con contenido activo*

Con .NET se pueden desarrollar controles ActiveX, que se pueden mostrar en las páginas web, de la misma forma que los applet de Java. Pero los ActiveX, ofrecen algunas ventajas en las cuales supera a los applet. Por ejemplo, los ActiveX corren aproximadamente 10 veces más rápido, tiene todo el poder de una aplicación nativa, que le permite tomar ventaja de las APIs, como DirectX, OpenGL, y QuickDraw3D que se encuentran en el sistema operativo.

En cuanto al contenido activo de páginas web .Net ofrece ventajas en la rapidez y compatibilidad con los recursos del sistema.

Después de haber realizado el análisis comparativo entre estas dos plataformas, se decide tomar la Plataforma Microsoft .NET para el desarrollo de esta aplicación. Dentro de las razones fundamentales en que se basaron esta selección se encuentran: la rapidez y facilidad con que se pueden desarrollar las aplicaciones, siendo estas más eficientes y escalables debido a su arquitectura, que es basa fundamentalmente en servicios web.

Por otra parte, como se pueden crear ActiveX, que son visualizados en las páginas web y teniendo en cuenta de que utiliza los recursos del cliente; se puede obtener el mismo rendimiento que las estaciones de visualización es escritorio durante el trabajo con las imágenes.

#### 1.2.4. ASP.NET

ASP.NET es un marco de trabajo de programación generado en el CLR que puede utilizarse en un servidor para generar aplicaciones Web. Aunque no descarta totalmente la idea de los lenguajes de scripting, introduce soporte para lenguajes completamente compilados, ofreciendo al desarrollador escribir el código en Visual Basic, C++ o C#.

A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Supone un incremento en el rendimiento cuando se solicita una página.

Se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

Debido a que ASP.NET se basa en el CLR, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad del CLR garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic. Además, el CLR simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.

### 1.2.5. C#

C# es un lenguaje de propósito general orientado a objetos diseñado por Microsoft para su plataforma .NET. Toma las mejores características de varios lenguajes preexistentes como Visual Basic, Java o C++ y las combinan en uno solo, eliminando aquellos elementos que son innecesarios. Soporta todas las características propias del paradigma de programación orientada a objetos (encapsulación, herencia y polimorfismo). Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

Proporciona una funcionalidad óptima para racionalizar los procesos empresariales, incluidos: compatibilidad con el diseño, el desarrollo y la implementación de servicios web XML con rapidez; diseñadores de formularios y controles visuales para crear aplicaciones basadas en Windows muy completas; herramientas y servicios de diseño para crear eficaces soluciones de Microsoft .NET basadas en servidor.

Es un lenguaje elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se pueden crear aplicaciones cliente para Windows tradicionales, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más.[MICROSOFT, 2007]

## 1.3. Gestores de bases de datos

A continuación se analizan las principales características de los gestores de bases de datos más utilizados.

### 1.3.1. Oracle

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y por ser multiplataforma.

Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web, aunque se reconoce su calidad, al ser un sistema muy caro no está tan extendido como otras bases de datos.

### 1.3.2. Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje Transact-SQL. Diseñado para procesamiento de transacciones en línea de alto rendimiento, almacenamiento de datos y aplicaciones de comercio electrónico.

Aunque es un potente gestor de base de datos, tiene el inconveniente de no ser multiplataforma, por ser Microsoft su propietario; solamente corre sobre la plataforma de Windows, además que se tiene que pagar altos precios para su adquisición.

### 1.3.3. MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de APIs en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

MySQL es un gestor de base de datos que fue creado con el objetivo de buscar rapidez de respuesta; por tal razón carecía de muchas de las características que tiene un gestor potente, como es la integridad referencial. Con el tiempo se le han adicionado algunas de las funcionalidades que antes no permitía, pero todavía no han alcanzado la robustez necesaria para que sea confiable.

### 1.3.4. PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS: Object-Relational Database Management System) basado en POSTGRES, Versión 4.2, desarrollado en la Universidad de California. Proporciona un gran número de características que normalmente se encontraban en las bases de datos comerciales tales como DB2 y Oracle[WORSLEY, 2001] [GROUP, TPGD, 2006]. Algunas de las características con las que cuenta son:

- **Sistemas Operativos:** PostgreSQL corre sobre la mayoría de los sistemas operativos existente, por ejemplo Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.

Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays. Altamente Extensible: PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.

- **Soporte SQL Comprensivo:** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **API Flexible:** La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- **Lenguajes Procedurales:** PostgreSQL incluye un lenguaje nativo denominado PL/pgSQL, que es comparable al PL/SQL de Oracle. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.



- MVCC: MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios.
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- Write Ahead Logging (WAL): Incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

Actualmente PostgreSQL se encuentra en su versión 8.2 el cual trae más de 200 nuevas prestaciones y mejoras contribuidas por docenas de desarrolladores, entre ellas se encuentran [GROUP, PGD, 2007]:

- Bloqueos consultivos («advisory locks»): permiten el control de objetos de bases de datos a nivel de aplicación usando el eficiente motor de bloqueos de PostgreSQL.
- pg\_dump selectivo: permite extraer vuelcos transaccionalmente consistentes de relaciones, seleccionando la inclusión y exclusión usando expresiones regulares.
- Criptografía: el módulo pgcrypto, soportando criptografía dentro de la base de datos, fue actualizado con los últimos algoritmos.
- Mejoras al SQL: nueva sintaxis, incluyendo UPDATE RETURNING, DROP IF EXISTS, ON COMMIT y nuevos comandos de propiedad («ownership») y permisos, para hacer más fácil el manejo de objetos de la base de datos en la línea de órdenes.

- Extracción de registros por lotes en psql: permite devolver filas a la consola en lotes en lugar de todas a la vez.

### 1.3.5. Fundamentación de la selección

Después de haber analizado los gestores de bases de datos más utilizadas, viendo sus características fundamentales, se decidió seleccionar PostgreSQL para la realización de este trabajo. Esto se debe a que es un potente gestor que puede compararse con sus competidores como son Oracle y SQL Server. Es libre y corre sobre varias plataformas, haciéndolo multiplataforma. Permite transacciones, es decir, múltiples operaciones de tabla o registros de manera segura.

Al proporcionar la funcionalidad del Control de Concurrencia Multi-Versión, este gestor mantiene la consistencia de los datos. Evitando que alguna transacción vea datos inconsistentes que puedan ser causados por la actualización de otra transacción concurrente en la misma fila de datos. Es un excelente optimizador de consultas, permite escribir funciones (procedimientos almacenados) en varios lenguajes (pl/pgsql, pl/perl, pl/python, pl/r), triggers, vistas, índices funcionales y parciales, tipos de datos extensibles. Es reconocido por su gran estabilidad y confiabilidad, es muy común que algunas compañías hayan reportado que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

## 1.4. Estado del arte sobre sistemas similares

### 1.4.1. En el mundo

En la actualidad existen varias compañías que dan soluciones PACS, están centradas fundamentalmente en soluciones a nivel de hospital, con altos costos y una gran dependencia posterior. Entre las principales empresas que se dedican a desarrollar y a distribuir este tipo de sistema se encuentran: Philips Medical Systems, Siemens Medical Solutions, AGFA Corp., Cedara Software Corp., IMAGE Information Systems Ltd y Dynamic Imaging.

Estas empresas para entregar información referente al precio que poseen sus productos, piden ser contactados primero, en algunos casos antes de realizar la compra exigen hacer una evaluación de las

condiciones del lugar para la instalación de la solución PACS, por otra parte las ventas en el mercado están destinadas hacia el continente europeo, asiático o en América del norte principalmente.

Si se realiza una búsqueda en Internet de soluciones PACS basados en web, se pueden encontrar varios, incluso gratis, como por ejemplo:

DIOWave Visual Storage, corre en servidores W2K y utiliza ASP.NET con C# o C++. El mismo ofrece reducidas posibilidades de procesamiento como por ejemplo realizar zoom, cambiar de ancho y largo. Por otra parte utiliza Microsoft Access para almacenar la información necesaria de los estudios, lo que implica que con el aumento de los estudios con el tiempo, se hará más difícil la búsqueda de información. Las últimas versiones utilizan el idioma japonés, lo cual sería un problema para los especialistas de habla hispana o inglesa. <http://diowave-vs.sourceforge.net/>

MyFreePACS, se requiere de Windows 2000 Professional ó Server, XP Professional ó Windows 2003 Server, utiliza PHP y como gestor de base de datos SQL Server. Para la visualización de las imágenes y la manipulación en el navegador se utiliza un control ActiveX. El usuario cuenta con las fundamentales herramientas que puede tener una estación de visualización, incluye localización 3D, rotación, inversión, zoom, entre otros. A pesar que ofrece grandes facilidades, el servidor presenta problemas para el almacenamiento de las imágenes, no soporta imágenes multiframe, las imágenes tiene que estar obligatoriamente en el servidor para poderlas utilizar en el visor. <http://www.myfreepacs.com/index.php>

Además de las dificultades que presentan estos sistemas, son creados para clínicas pequeñas y en algunos casos para clínicas específicas. También se puede decir, que en muchos casos son desarrollados por programadores independientes que no pertenecen a ninguna compañía en particular, a la cual se le pueda recurrir en caso de asistencia técnica o de actualizaciones, en otras palabras no hay ninguna entidad que sea responsable del sistema.

### **1.4.2. En Cuba**

Nuestro país no cuenta con una solución PACS basada en web. Las que existen son aplicaciones de escritorio. En la actualidad se encuentra distribuido de forma limitada el software IMAGIS, desarrollado hace varios años por el Centro de Biofísica Médica (CBM) de la Universidad de Oriente y el Software Savin desarrollado por una empresa española a la que se incorporó el desarrollador principal del IMAGIS. Estos sistemas serán sustituidos paulatinamente por el Sistema Cassandra PACS.

### **1.4.3. Cassandra Web**

A partir del estudio realizado sobre el estado del arte de los sistemas existentes, dentro y fuera del país; teniendo en cuenta los altos costos para la adquisición de estos sistemas. Además, las soluciones que se pueden encontrar gratis en Internet presentan problemas y en muchos casos no ofrecen garantías de su estabilidad. Por todo lo anterior se decide desarrollar la aplicación Cassandra Web.

Esta será compatible con imágenes que son generadas por equipos de última generación, incluyendo multiframe. Una vez que sean visualizadas, se les puede realizar varias transformaciones, tales como: aplicar brillo y contraste, paletas de colores, rotaciones. Será posible reproducir las imágenes multiframe a la velocidad especificada en el fichero DICOM. También, se podrán realizar mediciones de distancias, ángulos, áreas y volúmenes sobre la imagen. Una característica fundamental que presenta esta aplicación es que utiliza los recursos del cliente para realizar procesos tan complejos como es el procesamiento de las imágenes.

En este capítulo, se abordaron los términos fundamentales que son utilizados en el trabajo. Se caracterizaron las principales tecnologías y herramientas que son utilizadas mundialmente para el desarrollo de aplicaciones web. Se realizó una comparación entre las dos plataformas de desarrollo más utilizadas para el desarrollo de aplicaciones. Se estudiaron los sistemas similares que existen dentro y fuera del país, lo que permitió fundamentar la realización de este proyecto.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realiza una descripción del problema y su ubicación; así como el entorno en que se desarrollan los procesos. También se hace un estudio detallado del modelo del negocio, reflejándose la descripción de los actores, trabajadores y casos de usos, se muestra además, los diagramas de caso de uso, de actividad y el de modelo de objetos. Se describen los requerimientos que tiene el sistema, tanto los funcionales como los no funcionales.

### **2.1. Problema científico y situación problemática**

Cuando un paciente se le orienta un estudio imagenológico, es enviado a la sala donde se encuentra el equipo. Una vez que se realiza el estudio, dichas imágenes que son generadas por el equipo, son almacenadas en un servidor DICOM, que tiene la función fundamental de almacenar estas imágenes, para luego, desde estaciones de visualización puedan ser accedidas por el o los especialistas que estén atendiendo el caso. Estas estaciones de visualización le ofrecen herramientas para poder procesar estas imágenes y de esta forma poder emitir un reporte.

Estas soluciones PACS que se encuentran funcionando en nuestro país, ofrecen una arquitectura cliente – servidor, garantizando que los especialistas puedan acceder a la información y realizar el estudio desde lugares geográficamente remotas. Pero estas estaciones de visualización al ser aplicaciones de escritorio traen consigo problemas de configuración y mantenimiento en el momento de instalación o cuando dejan de prestar el servicio requerido, algo que está fuera del alcance del conocimiento de los médicos que interactúan con la aplicación.

Es decir, que el personal informático encargado de realizar estas labores tiene que ir a cada una de las estaciones que presenten problemas o desean realizar una nueva instalación. Como es de suponerse, esto requiere de tiempo y esfuerzo, que se va incrementando a medida que aumentan las estaciones de visualización y con ello la dispersión geográfica.

Por tanto, con el sistema propuesto, con una arquitectura cliente –servidor pero sobre la web, los problemas de configuración y mantenimiento durante la instalación se centraran en la parte del servidor. De esta forma el personal informático encargado solamente tiene que trabajar sobre el servidor. Los

especialistas pueden acceder a la información y realizar las operaciones que comúnmente hacían mediante una interfaz web, abstrayéndolos de todos estos problemas antes expuestos.

## **2.2. Objeto de automatización**

El especialista para poder realizar el estudio de las imágenes pertenecientes al paciente que esté analizando en ese momento, cuenta con una serie de recursos para el procesamiento de dichas imágenes, tales como: aplicar diferentes tipos de filtros (paso bajo o alto, ecualizar, invertir, brillo y contraste, paletas de colores), realizar mediciones sobre la imagen de figuras geométricas conocidas o de forma variable, en caso de que la imagen sea multiframe puede realizar la reproducción de la misma, entre otras.

Una vez que el especialista haya analizado las imágenes puede emitir un reporte, donde se pone información del paciente y sus resultados después de haber realizado el estudio.

## **2.3. Sistema propuesto**

El sistema propuesto consiste en una aplicación web, al cual el especialista tiene que autenticarse para poder entrar. Una vez dentro puede acceder a las imágenes que tiene almacenadas en una base de datos temporal que hay en el servidor web, de no tener la imagen o imágenes deseadas, puede hacer la búsqueda y la obtención de ella(s) mediante la bandeja de casos, la cual se conecta al servidor DICOM mediante un servicio web.

Una vez visualizada la imagen, el especialista puede realizar el procesamiento de la misma, recurriendo a los recursos que le brinda la aplicación, para luego emitir el reporte, que este puede ser impreso en un documento Word o ser enviado a un repositorio de reportes (Cassandra X-Web).

Este producto formará parte del Cassandra PACS, la cual está compuesta por una serie de aplicaciones (DMail, X-Web, Cassandra Clinic, el Viewer y el Servidor). El Cassandra Web Viewer brindará las mismas posibilidades de visualización y manipulación de imágenes DICOM que el Cassandra Viewer para escritorio, pero a través del uso de la web, trayendo consigo la facilidad de acceso y uso para puestos remotos.

Los especialistas tendrán la posibilidad de visualizar los estudios almacenados en el PACS desde cualquier lugar donde halla una computadora conectada a la red del hospital, mediante un navegador de Internet. Esto permitirá extender el ámbito del PACS más allá del servicio de radiología fácilmente. El hecho que la interfaz de la distribución web sea semejante a las estaciones de visualización disminuye el tiempo de formación de los usuarios, que se familiarizarían rápidamente con ella.

## 2.4. Modelo de negocio

### 2.4.1. Representación de los casos de uso del negocio

#### 2.4.1.1. Actores

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso juegan al interactuar con éstos. Normalmente, un actor representa un rol que es jugado por una persona, un dispositivo hardware o incluso otro sistema.[BOOCH, 2000]

Actores del negocio	Justificación
Paciente	Es quien se realiza el estudio y el beneficiado una vez terminado el proceso.

#### 2.4.1.2. Trabajadores

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajadores del negocio	Justificación
Especialista	Es quien analiza las imágenes del paciente para luego emitir un reporte de su estado.

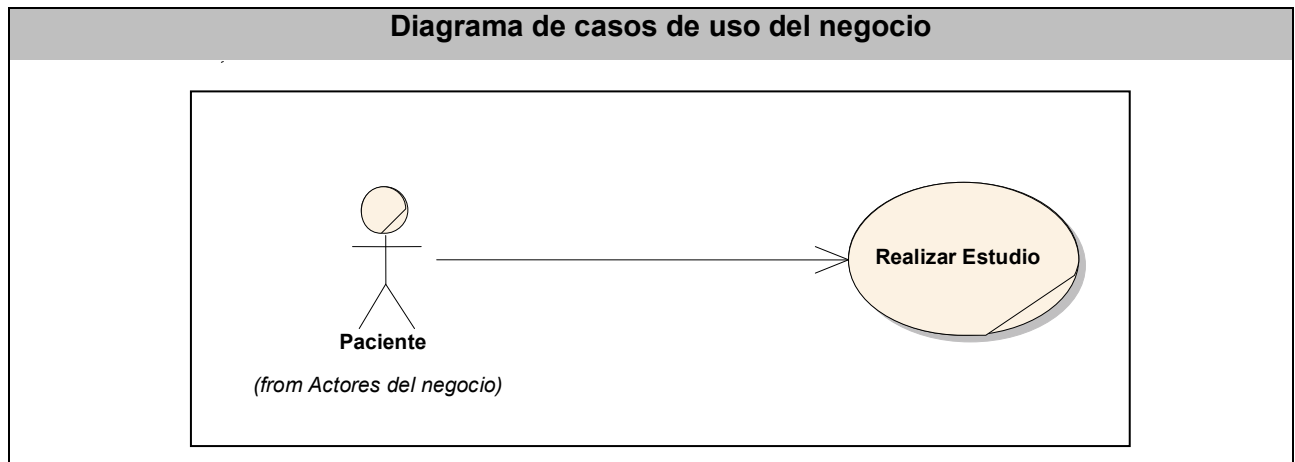
## 2.4.1.3. Descripción de los casos de uso

<b>Nombre del caso de uso del negocio:</b>	Realizar estudio.	
<b>Actores del negocio:</b>	Paciente	
<b>Propósito:</b>	Obtener un diagnostico de su estado de salud.	
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el paciente llega al Departamento de imagenología, entrega la orden del estudio al técnico de laboratorio o al radiólogo. El caso de uso finaliza una vez obtenida las imágenes del equipo, el radiólogo las analiza y emite un diagnostico que se le entrega al paciente.</p>	
<b>Casos de uso asociados:</b>	-	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
<ol style="list-style-type: none"> <li>1. El paciente entrega la orden de estudio al especialista.</li> <li>5. El paciente recibe el diagnostico y se retira, termina el caso de uso.</li> </ol>	<ol style="list-style-type: none"> <li>2. El especialista recibe la orden de estudio.</li> <li>3. Se le realiza el estudio al paciente.</li> <li>4. El especialista analiza las imágenes y realiza un diagnostico que se lo entrega al paciente.</li> </ol>	
<b>Prioridad:</b>	-	
<b>Mejoras:</b>	Se le brindará facilidades para que el especialista pueda realizar el diagnostico desde cualquier lugar, se abstrae de todo problema de instalación o actualización del sistema.	
<b>Cursos alternos:</b>	-	

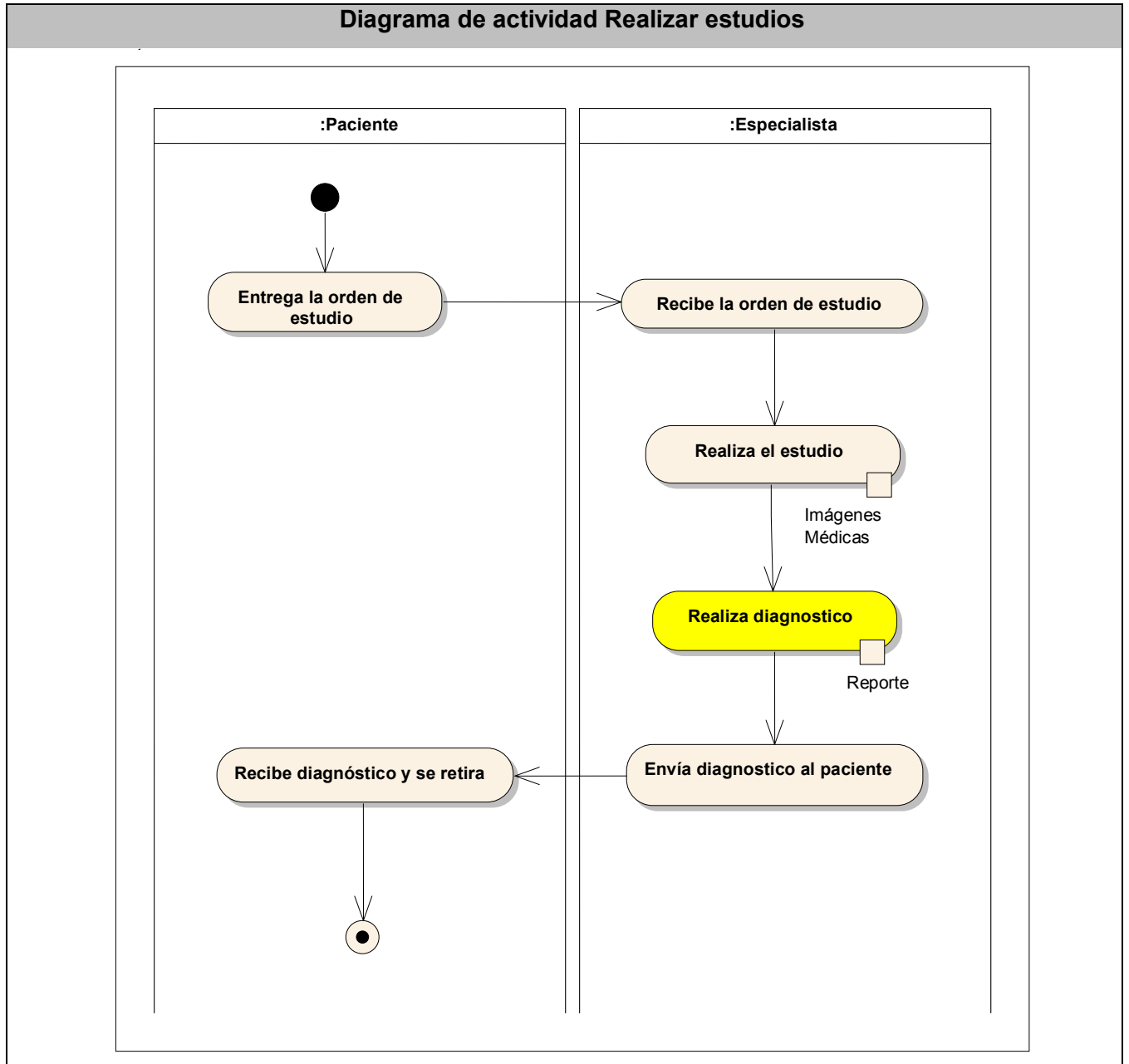


### 2.4.2. Diagramas del modelo del negocio

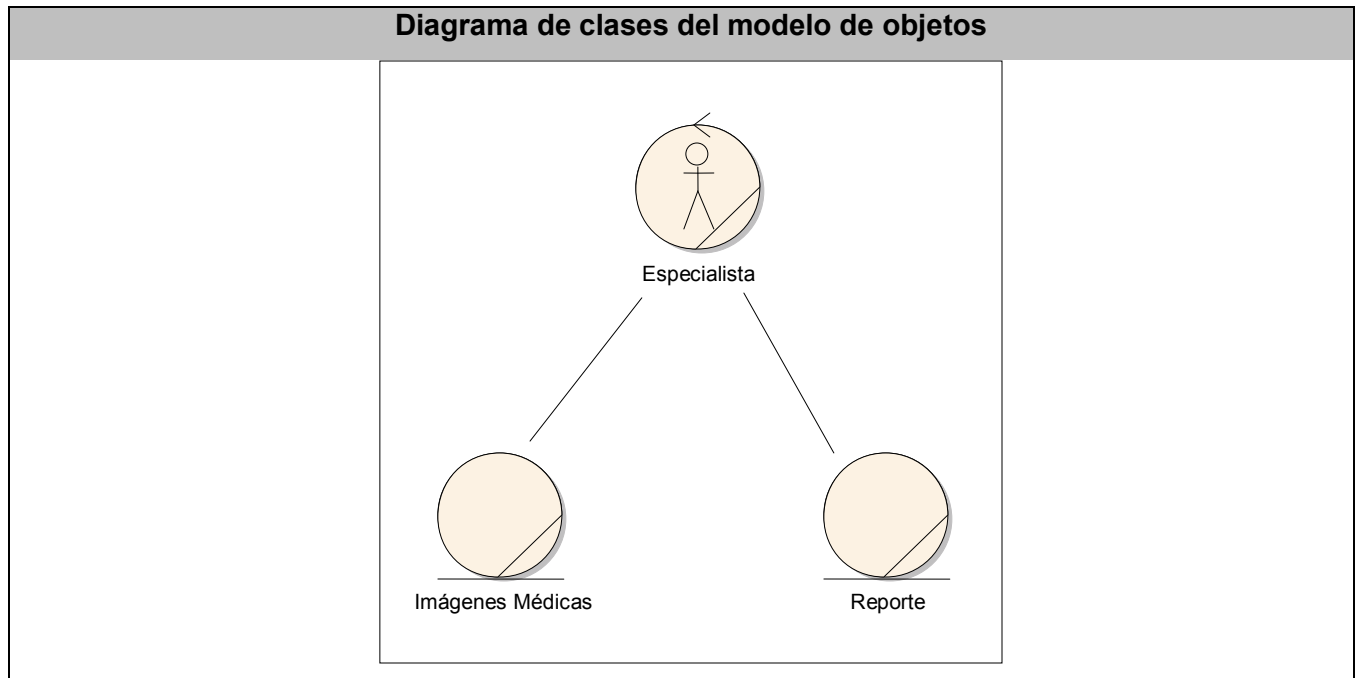
#### 2.4.2.1. Diagrama de casos de uso del negocio



2.4.2.2. Diagrama de actividad del negocio



### 2.4.2.3. Diagrama de clases del modelo de objeto



## 2.5. Especificación de los requisitos de software

### 2.5.1. Dependencias y relaciones

Cassandra Web se relaciona con un servidor DICOM compatible que tiene la función de almacenar todas las imágenes que son generadas por los equipos médicos que se encuentren conectados a él. Mediante esta relación el especialista puede adquirir las imágenes de los pacientes que está analizando. Otra relación que tiene el sistema es con un repositorio de reportes, donde el especialista una vez que haya terminado de elaborar el reporte pueda enviarlo al mismo. El repositorio ofrece una interfaz donde se puede recurrir a los reportes de un paciente determinado para estudios posteriores.

### 2.5.2. Requerimientos Funcionales

1. Visualizar imágenes
  - 1.1. Visualizar imágenes DICOM simples y multiframe ya sean monochrome1, monochrome2 o RGB.
  - 1.2. Visualizar de una vez varias imágenes DICOM de una misma serie.
  - 1.3. Visualizar mosaicos de imágenes.
  - 1.4. Ubicar referencias de imágenes de diferentes vistas.
  - 1.5. Mostrar toda la información del metadata adjunta en el fichero DICOM.
  - 1.6. Navegar por las imágenes que componen el fichero DICOM multiframe.
  - 1.7. Reproducir imágenes multiframe a la velocidad especificada en el fichero DICOM (frame\_time o vector\_frame\_time).
  - 1.8. Visualizar el histograma de las imágenes tanto RGB como en tonos de grises así como datos estadísticos.
2. Generar reportes
  - 2.1. Generar reportes médicos con el codificador de enfermedades establecido por la ACR.
  - 2.2. Imprimir los reportes en formato .doc.
  - 2.3. Enviar el reporte de un estudio hacia un repositorio central de reportes.
3. Interactuar con servidor DICOM compatible.
  - 3.1. Buscar imágenes o series de imágenes a partir del nivel de pacientes en un servidor DICOM.
  - 3.2. Buscar imágenes o series de imágenes a partir del nivel de estudio en un servidor DICOM.
  - 3.3. Obtención de imágenes o series de imágenes a partir del nivel de pacientes en un servidor DICOM.
  - 3.4. Obtención de imágenes o series de imágenes a partir del nivel de estudio en un servidor DICOM.
  - 3.5. Almacenar imágenes o series de imágenes médicas en un servidor DICOM compatible.
4. Transformaciones espaciales.
  - 4.1. Rotar imágenes a ángulos fijos (90, 180, 270).
  - 4.2. Realizar zoom sobre la imagen.
    - 4.2.1. Realizar zoom sobre toda la imagen.
    - 4.2.2. Realizar zoom sobre una región de interés en la imagen.

- 4.3. Realizar espejos horizontales y verticales sobre la imagen.
5. Procesar imágenes
  - 5.1. Invertir una imagen.
  - 5.2. Aplicar brillo y contraste a la imagen.
    - 5.2.1. A partir de cambiar ancho y centro de ventanas.
    - 5.2.2. A partir de valores etiquetados de ancho y centro de ventana.
  - 5.3. Filtrar imágenes RGB a niveles de grises.
  - 5.4. Filtrar imágenes por ecualización.
  - 5.5. Filtrar imágenes por filtros paso bajo y paso alto con algunas variantes.
  - 5.6. Filtrar imágenes por filtros de suavizados.
  - 5.7. Aplicar paletas de colores a las imágenes a partir de ficheros lut estandarizados.
6. Realizar mediciones.
  - 6.1. Medir ángulos sobre las imágenes.
  - 6.2. Medir perímetro sobre las imágenes.
  - 6.3. Medir distancias sobre las imágenes.
  - 6.4. Medir área para regiones geoméricamente conocidas o dibujadas de forma irregular.
7. Exportar imágenes
  - 7.1. Exportar imágenes DICOM a otros formatos conocidos (jpg, bmp, tif).
  - 7.2. Exportar imágenes DICOM multiframe al formato de video avi.
8. Anonimizar un fichero DICOM.

### 2.5.3. Requisitos no funcionales

- **Portabilidad**
  - La base de datos debe ser multiplataforma.
- **Seguridad**
  - El acceso a la aplicación será restringido.
  - La información de los pacientes que atiende un especialista solo será visible por él.
- **Políticos – Culturales**
  - El sistema deberá dar la posibilidad de cambiar el idioma en correspondencia al país donde se instale.

- **Legales**
  - La aplicación, junto con toda la información generada durante el desarrollo de la misma pertenecen a la Universidad de la Ciencias Informáticas.
- **Confiabilidad**
  - El sistema debe estar disponible las 24 horas del día.
- **Interfaz**
  - La interfaz de la aplicación debe ser similar al Cassandra Viewer, para que los especialistas mantengan la misma experiencia de usuario y no necesiten tiempo en prepararse.
- **Ayuda y documentación en línea**
  - La aplicación debe ofrecer una ayuda.
- **Software**
  - En el Cliente
    - .NET Framework v2.0.
    - Microsoft Windows XP;
    - Internet Explorer v6 o superior.
  - En el Servidor
    - .NET Framework v2.0.
    - Internet Information Service (IIS) v5 o superior.
    - PostgreSQL v8.2.
- **Hardware**
  - Pentium IV 2.6 GHz
  - Memoria RAM: 512 MB.

## 2.6. Definición de los casos de uso

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema.

Además ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Por lo general el nombre de un caso de uso comienza con un verbo en infinitivo. Un caso de uso describe un proceso de principio a fin, es decir, una secuencia de eventos, las acciones y las transacciones que se requieren para realizarlo.

### 2.6.1. Definición de los actores

<b>.Actores</b>	<b>Justificación</b>
Especialista	Es la persona que va a interactuar directamente con el sistema. Realiza un análisis de las imágenes de los pacientes que está atendiendo para después hacer un emitir un reporte.
Servidor DICOM	Es un subsistema externo con el cual se interactúa para buscar y obtener las imágenes que contiene.
Repositorio de reporte	Es un subsistema externo que es donde se almacenan los reportes que son emitidos por los especialistas.

### 2.6.2. Listado de los casos de uso

1. Visualizar imágenes.
2. Generar reporte.
3. Buscar imágenes en servidor DICOM.
4. Obtener imágenes en servidor DICOM.
5. Almacenar imágenes en servidor DICOM.
6. Rotar imagen.
7. Aplicar espejos sobre la imagen.
8. Aplicar zoom en la imagen.
9. Aplicar filtros sobre la imagen.
10. Realizar mediciones sobre la imagen.

11. Exportar imágenes a otros formatos.
12. Guardar DICOM anonimizado.

### 2.6.3. Descripción de los casos de uso

CU1	Visualizar imágenes.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista busca la imagen que desea ver, el sistema la carga y la visualiza en pantalla, una vez mostrada la imagen el especialista puede trabajar sobre ella.
<b>Referencia</b>	RF1.1, RF1.2, RF1.3, RF1.4, RF1.5, RF1.6, RF1.7, RF1.8

CU2	Generar reporte.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista después de haber realizado el diagnostico del paciente una vez analizado las imágenes del estudio correspondiente, puede emitir un reporte, el cual puede ser impreso en un documento Word o enviarlo a un repositorio de reportes.
<b>Referencia</b>	RF2.1, RF2.2, RF2.3

CU3	Buscar imágenes en servidor DICOM.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede realizar búsquedas de las imágenes en un servidor DICOM a partir de dos niveles, por paciente o por estudio.
<b>Referencia</b>	RF3.1, RF3.2



CU4	Obtener imágenes en servidor DICOM.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	Después que el especialista realiza la búsqueda de las imágenes, puede realizar la obtención de dichas imágenes para su estudio.
<b>Referencia</b>	RF3.3, RF3.4

CU5	Almacenar imágenes en servidor DICOM.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista tiene la posibilidad de que imágenes que no se encuentren en el servidor pueda subirlo al mismo.
<b>Referencia</b>	RF3.5

CU6	Rotar imagen.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede rotar la imagen a 90, 180 o 270 grados.
<b>Referencia</b>	RF4.1

CU7	Aplicar espejos sobre la imagen.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede aplicarle dos tipos de espejo a la imagen, con respecto a la horizontal y con respecto a la vertical.
<b>Referencia</b>	RF4.2

CU8	Aplicar zoom en la imagen.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede ampliar la imagen y disminuirla a conveniencia, ya sea a toda la imagen o una región específica utilizando una lupa, en esta última forma el cursor toma forma rectangular ampliando la parte imagen por donde esté pasando el cursor.
<b>Referencia</b>	RF4.3.1, RF4.3.2

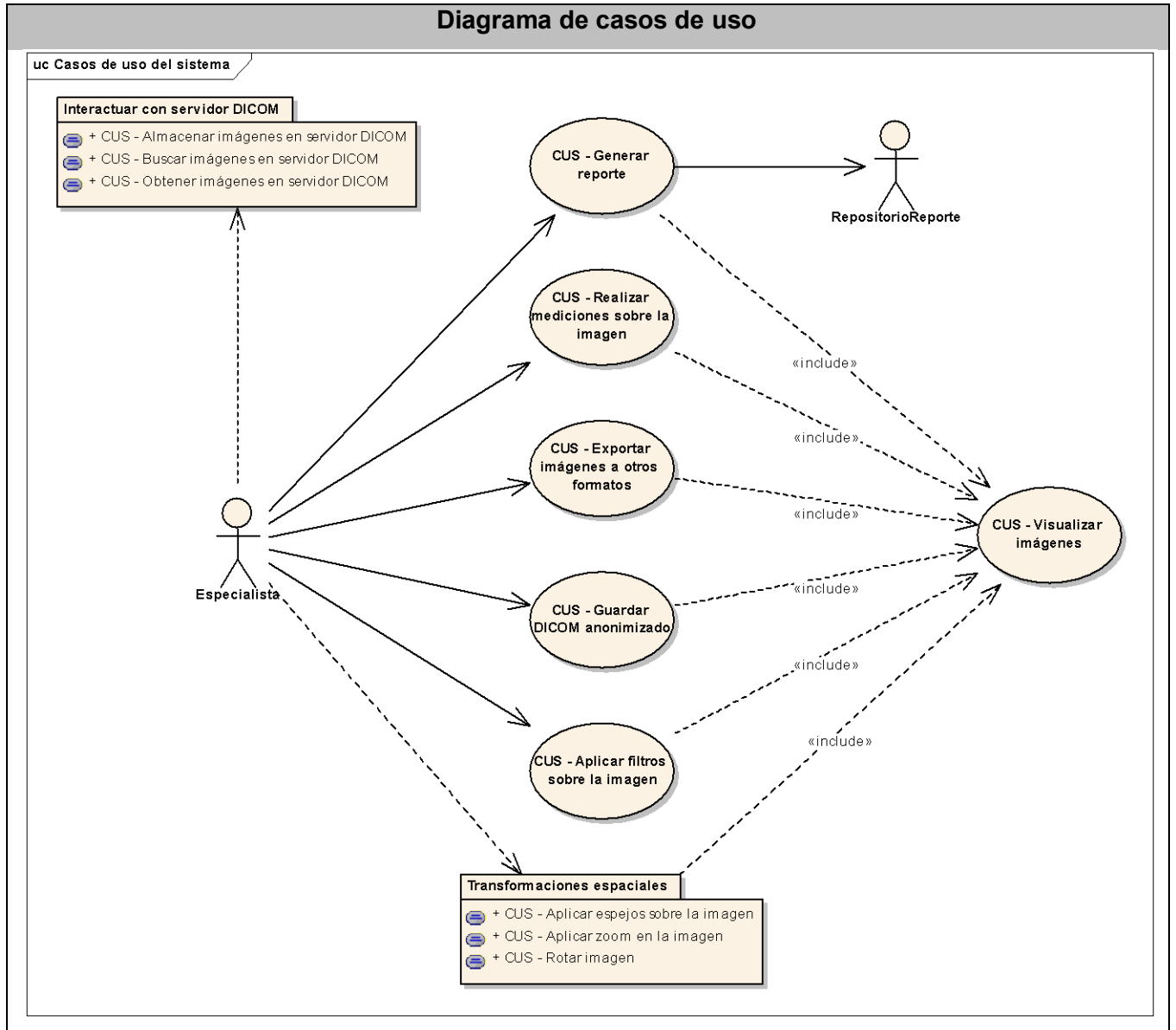
CU9	Aplicar filtros sobre la imagen.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede aplicarle una serie de filtros a la imagen con el objetivo de resaltar partes en la imagen, de la cual originalmente no se ven. Dentro de los filtros que se pueden aplicar están: invertir, paso alto o paso bajo, por ecualización, suavizados etc.
<b>Referencia</b>	RF5.1, RF5.2.1, RF5.2.2, RF5.3, RF5.4, RF5.5, RF5.6, RF5.7

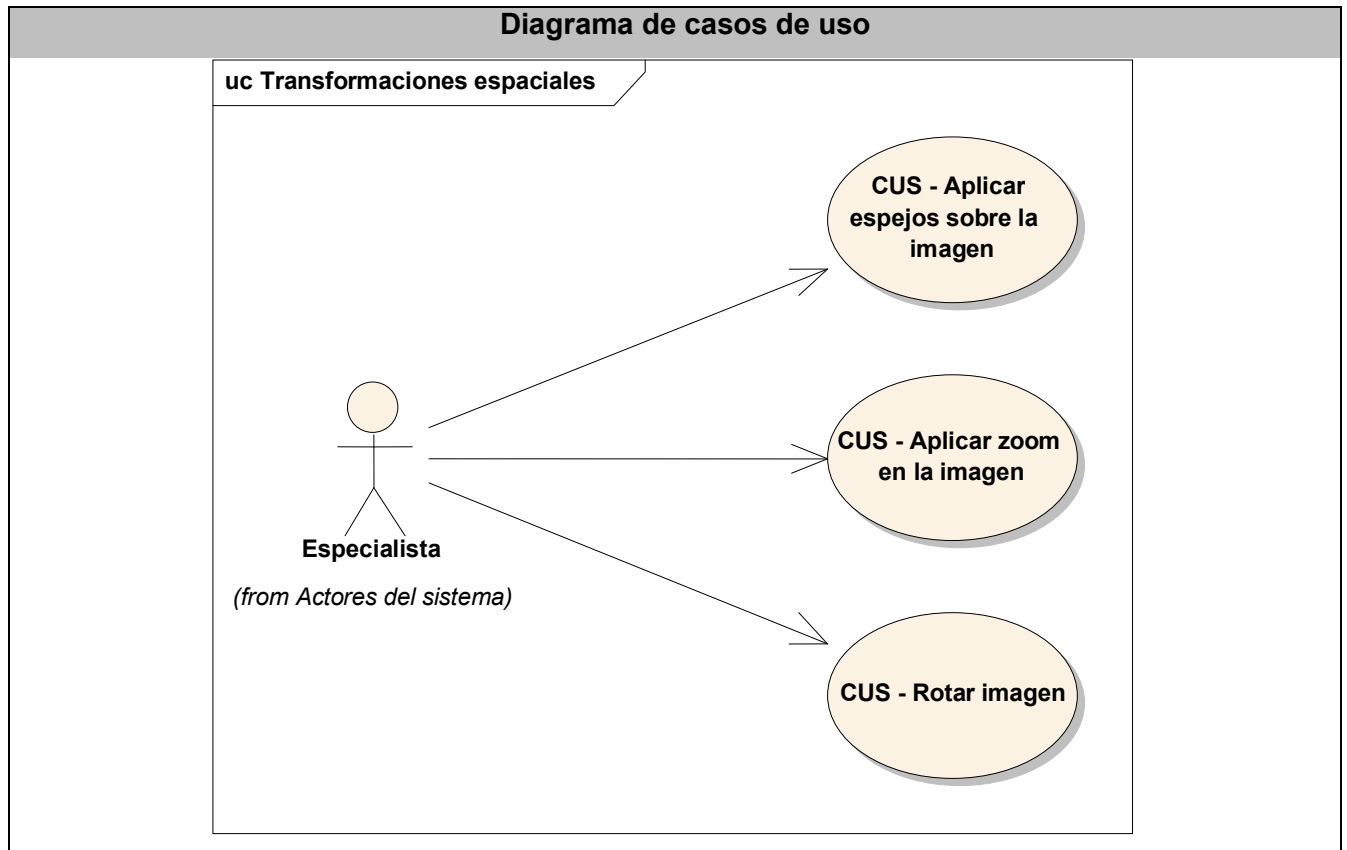
CU10	Realizar mediciones sobre la imagen.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede realizar diferentes tipos mediciones sobre la imagen, de esta forma puede saber la distancia que hay entre dos puntos cuales quiera o el área y el perímetro de una región.
<b>Referencia</b>	RF6.1, RF6.2, RF6.3, RF6.4

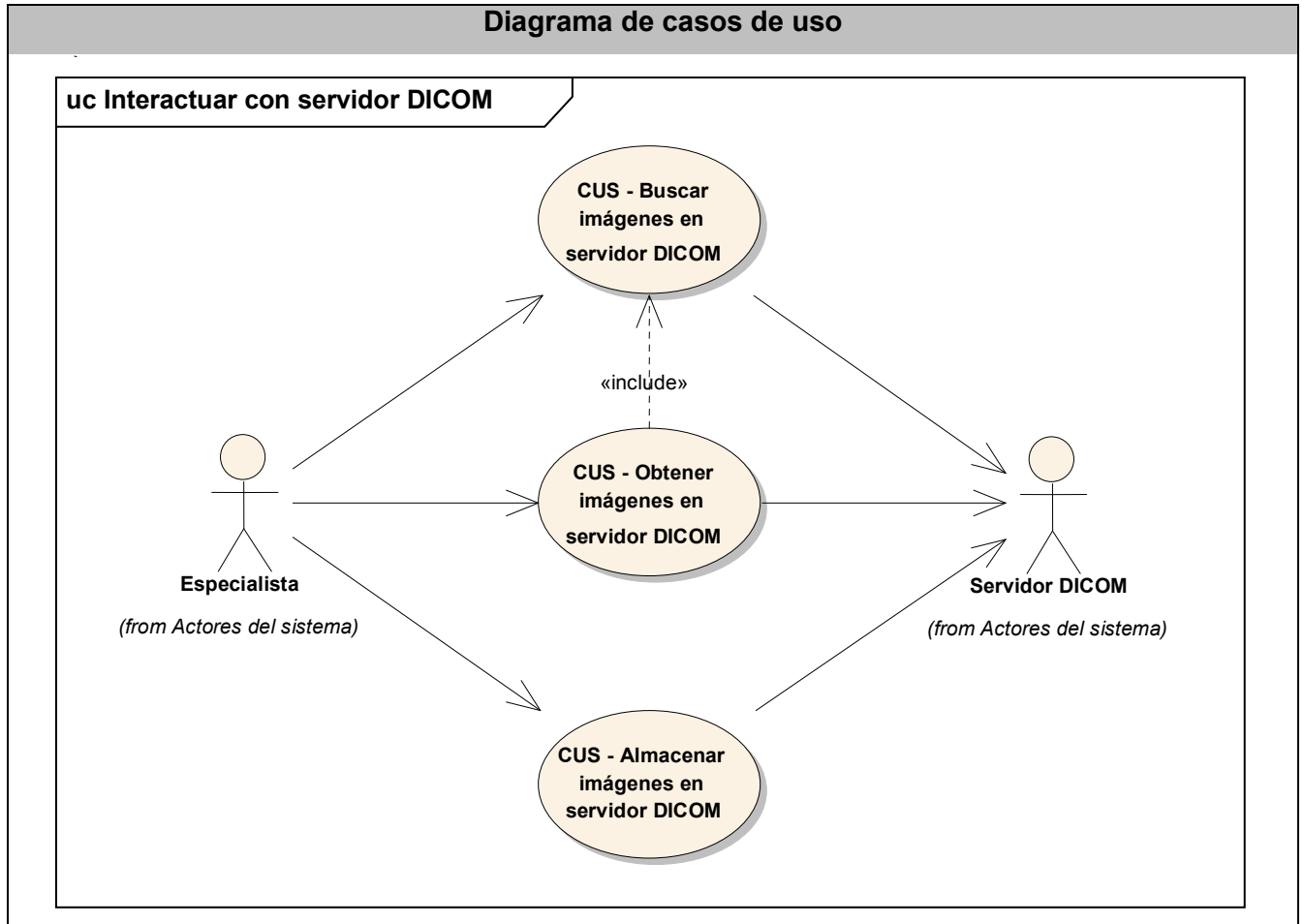
CU11	Exportar imágenes a otros formatos.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede exportar las imágenes a formatos de jpg, bmp o tif, en caso de que las imágenes sean multiframe puede exportar a formato avi.
<b>Referencia</b>	RF7.1, RF7.2

CU12	Guardar DICOM anonimizado.
<b>Actor</b>	Especialista (inicia).
<b>Descripción</b>	El especialista puede guardar el fichero DICOM de forma anonimizada, es decir que los datos sensibles, como los datos del paciente son retirados del mismo.
<b>Referencia</b>	RF8

### 2.6.4. Diagrama de casos de uso







### 2.6.5. Casos de uso por ciclo

A continuación se muestran los casos de uso separados por 3 ciclos de desarrollo del proyecto.

#### 2.6.5.1. Ciclo 1 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU1	Visualizar imágenes.	Estos casos de usos son los más importantes, debido a que cumplen con las necesidades básicas que pueden tener un especialista a la hora de realizar un estudio de las imágenes que son generadas por los equipos.
CU2	Generar reporte.	
CU3	Buscar imágenes en servidor DICOM.	
CU4	Obtener imágenes en servidor DICOM.	
CU5	Almacenar imágenes en servidor DICOM.	
CU9	Aplicar filtros sobre la imagen.	

#### 2.6.5.2. Ciclo 2 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU10	Exportar imágenes a otros formatos.	Estos casos de uso dependen de la visualización de las imágenes y como no son tan críticos, se dejan para este ciclo.
CU12	Guardar DICOM anonimizado.	

### 2.6.5.3. Ciclo 3 de desarrollo

Cód	Nombre de caso de uso	Justificación de la selección.
CU6	Rotar imagen.	Estos casos de usos poseen menos complejidad de implementación que los anteriores, por lo tanto se deja para este ciclo.
CU7	Aplicar espejos sobre la imagen.	
CU8	Aplicar zoom en la imagen.	
CU10	Realizar mediciones sobre la imagen.	

### 2.6.6. Casos de uso expandidos

Los casos de uso expandidos son muy útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos. Constituyen un documento narrativo que describe la secuencia de eventos de un actor que utiliza el sistema para completar un proceso.[LARMAN, 2002]

Ver casos de uso expandidos en [ANEXO 1](#).

En este capítulo se quedaron definidos los procesos del negocio. Se obtuvieron los requerimientos funcionales y no funcionales de la aplicación a partir del modelo del negocio. Posteriormente, y partiendo de los requisitos del sistema fueron presentados los casos de uso y sus relaciones con los actores. Los mismos fueron separados por ciclos de desarrollo en dependencia del nivel de prioridad de los mismos.



## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Una vez que se realiza la captura de requisitos se obtiene una vista externa del sistema expresado en un lenguaje que el cliente puede entender. A partir de aquí se comienza a especificar los casos de uso y extraer las clases necesarias para poder realizar las distintas funcionalidades que requiere el sistema, de esta forma se permite reflejar una vista interna del sistema utilizando un lenguaje que es entendible por los desarrolladores. Estos procesos se realizan dentro del flujo de trabajo de análisis y diseño, los cuales se abordan en este capítulo.

### 3.1. Análisis

“El análisis pone énfasis en una investigación del problema y los requisitos, en vez de ponerlo en una solución”.[LARMAN, 2002]

#### 3.1.1. Definición del modelo de análisis

El modelo de análisis ofrece una especificación más precisa de los requisitos que la se tiene como resultado de la captura de requisitos, los mismos son estructurados de un modo que facilita su comprensión, su preparación, su modificación, puede considerarse como la primera aproximación al modelo de diseño y es por tanto una entrada fundamental cuando se le da forma al sistema en el diseño y en la implementación.[JACOBSON, 2000]

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos.

RUP propone clasificar a las clases en: entidad, son aquellas que modelan la información que posee larga vida y que es a menudo persistente; interfaz, modelan la interacción entre los sistemas y sus actores; controladora, coordinan la realización de los casos de uso coordinando las actividades de los objetos que implementen la funcionalidad del caso de uso.

### 3.1.2. Diagrama de clases de análisis

En el diagrama de clases de análisis se representan las clases participantes en la realización de los casos de usos así como sus relaciones entre ellas y con los actores.

Ver diagrama de clases del análisis en [ANEXO 2](#).

## 3.2. Diseño

“El diseño pone énfasis en una solución conceptual que satisface los requisitos, en vez de ponerlo en la implementación”. [LARMAN, 2002]

### 3.2.1. Diagramas de interacción

Mediante el diagrama de interacción se describe la secuencia de intercambios de mensajes entre los objetos para realizar las tareas de una manera gráfica.

Ver diagramas de interacción en [ANEXO 3](#).

### 3.2.2. Diagramas de clases de diseño

El diagrama de clases de diseño ofrece una vista estática de las definiciones de las clases participantes en el software.

Ver diagramas de clases en [ANEXO 4](#).

#### 3.2.2.1. Descripción de las clases

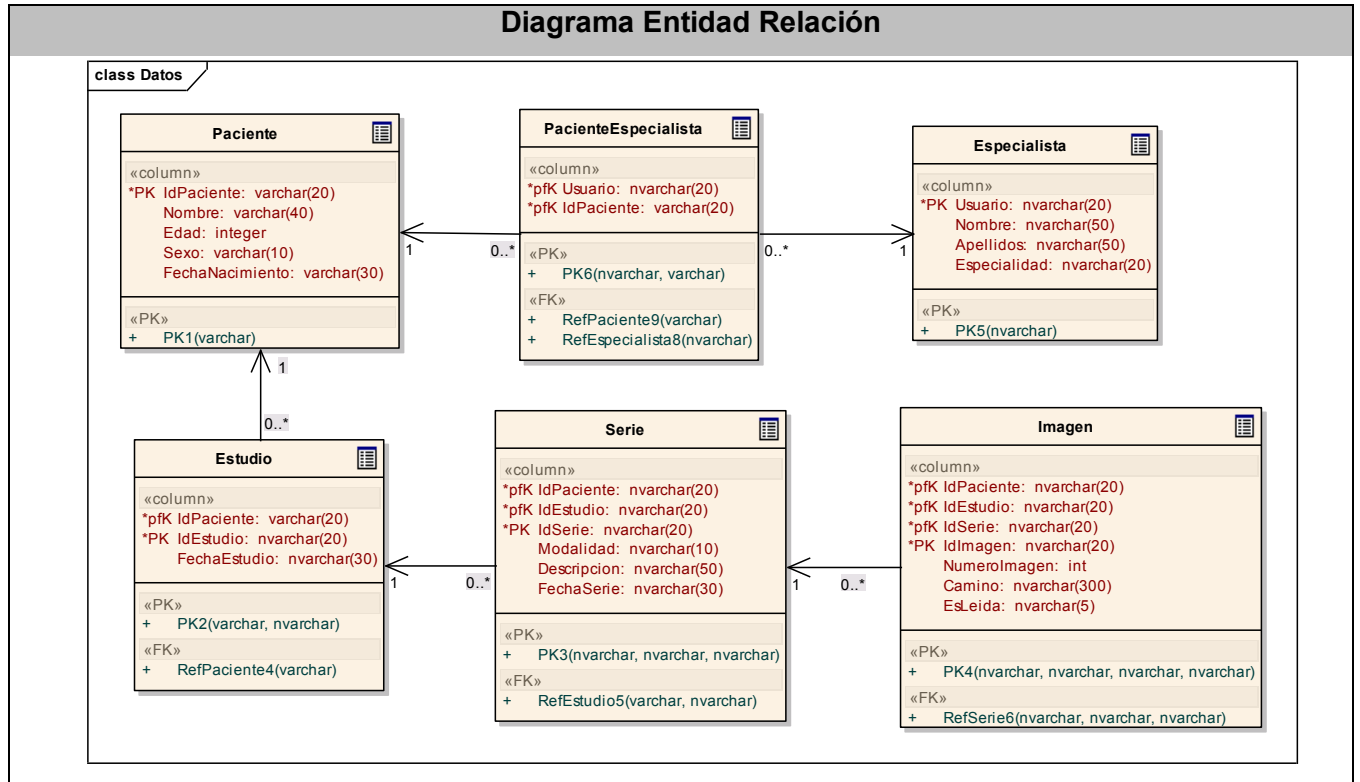
Con la descripción de las clases se obtiene una información más detallada sobre las clases que están implicadas, pues se conoce sus atributos de tener, así como las operaciones que las hacen únicas de acuerdo a su responsabilidad dentro del sistema.

Ver descripción de las clases en [ANEXO 5](#).

### 3.2.3. Diseño de la base de datos

#### 3.2.3.1. Diagrama entidad de relación de la base de datos

El diagrama entidad relación o DER es una notación gráfica utilizado para el modelado de datos de alto nivel, representando las relaciones que hay entre las entidades, así como sus propiedades.



### 3.2.3.2. *Descripción de las tablas*

La descripción de las tablas contiene la estructura de la información que se almacena en ellas.

Las clases Especialista, Paciente, Estudio, Serie, Imagen por ser clases persistentes pasan a ser tablas en la base de datos.

Ver descripción de las tablas en el [ANEXO 6](#).

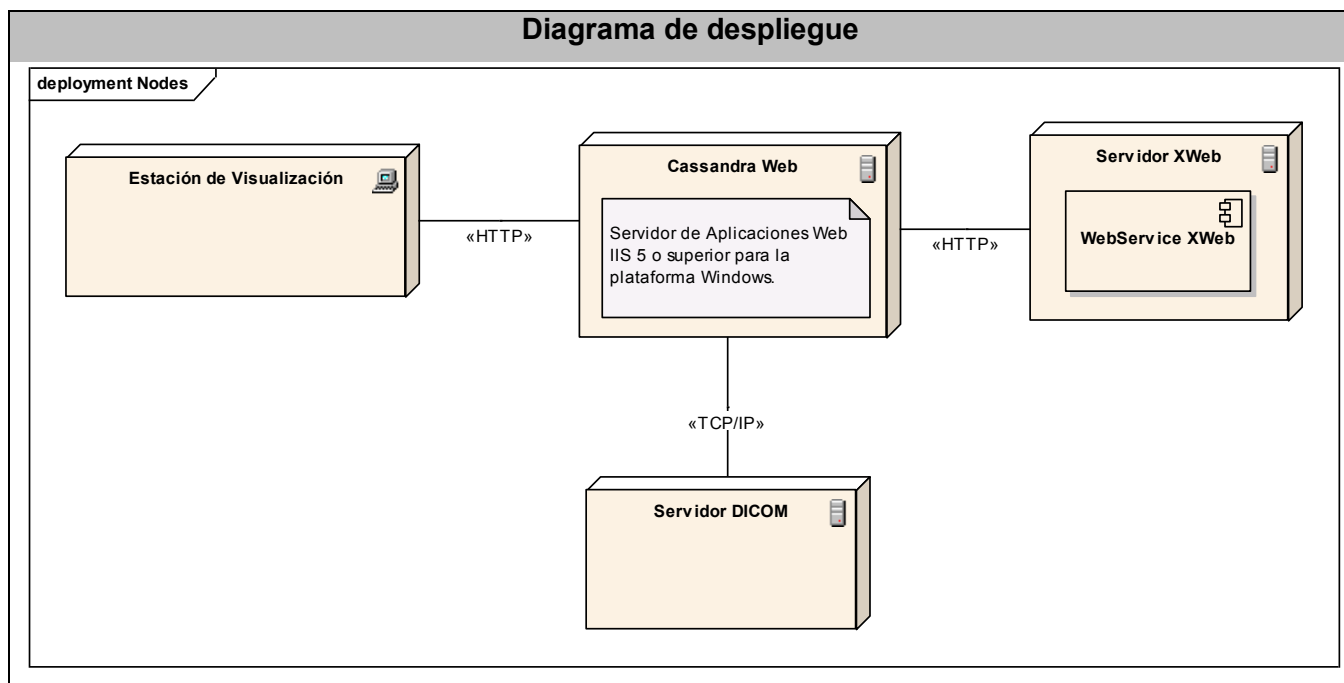
En este capítulo se describieron los diagramas de clases, de interacción y la descripción de las clases que participan en el sistema. Por otra parte se vio la estructura de la base de datos, mostrando el diagrama entidad de relación y la descripción las tablas.

## CAPÍTULO 4: IMPLEMENTACIÓN

La mayor parte de la arquitectura del sistema es capturada en el diseño, a partir de los resultados que se obtienen en esta etapa es que comienza la fase de implementación, que es donde se realiza la implementación como tal del sistema, es decir, es donde se crean los ficheros de código fuente, script, ejecutables, etc. En este capítulo se verán los artefactos fundamentales en esta fase.

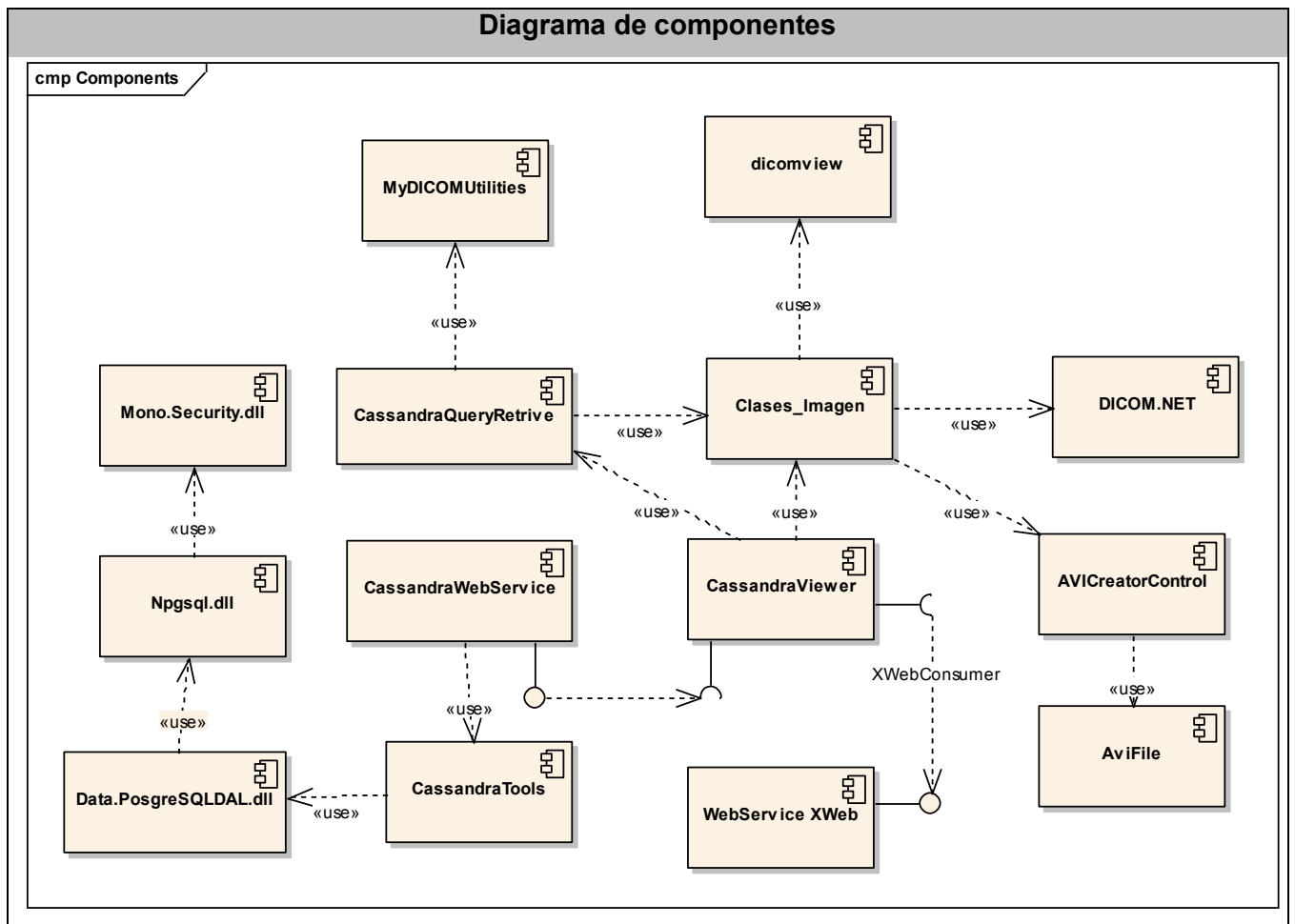
### 4.1. Diagrama de despliegue

Los diagramas de despliegue muestran la disposición física 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. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.



### 4.2. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.



En este capítulo se mostró el diagrama de despliegue, donde se ve la relación que hay entre los componentes que componen el sistema. También se vio el diagrama de componente donde se mostró la relación que hay entre los elementos que conforman el software.

## CONCLUSIONES

Con la culminación de este trabajo se ha llegado a las siguientes conclusiones:

- Se realizó un estudio de las aplicaciones existentes instaladas en los hospitales de Cuba, donde se detectó los problemas que presentan en el momento de instalación o de configuración.
- Se decide desarrollar una solución que resuelva estos problemas, proponiendo una aplicación web, aprovechando las facilidades que presentan este tipo de aplicación. Al ofrecer una arquitectura cliente-servidor, estos problemas de configuración se encapsulan en la parte servidora, los usuarios solamente accederían a la aplicación mediante un navegador.
- Las necesidades de los especialistas fueron identificadas mediante la captura de los requisitos.
- Se realizó el análisis y diseño del primer ciclo de desarrollo, mostrando a través de los diagramas de clases del análisis y diseño, y los de interacción; los conceptos más significativos en el dominio del problema así como el comportamiento interno del sistema. Durante el desarrollo de este proceso se tuvo en cuenta mantener la compatibilidad con el estándar DICOM.
- Por último se implementó una aplicación web para la visualización y gestión de imágenes médicas. En esta, los especialistas tendrían otra variante para poder visualizar y gestionar las imágenes que pertenecen a los casos que está atendiendo. La misma se integraría a la solución Cassandra PACS, logrando así una mayor extensión del mismo.



## RECOMENDACIONES

- Adicionarle las siguientes funcionalidades al sistema:
  - Reconstruir y visualizar estructuras 3D a partir de ficheros DICOM o de series de ficheros DICOM.
  - Visualizar y navegar por planos 3D a partir de imágenes DICOM o series de imágenes DICOM.
  - Imprimir imágenes DICOM en impresoras DICOM o no DICOM.
- El estudio de la posibilidad de migración hacia la plataforma libre.
- Integración con sistemas para la gestión de la información hospitalaria (Cassandra Clinic).
- Integración con el servidor de referencias desarrollado por el proyecto.
- Someter a la aplicación a las pruebas de calidad.

## REFERENCIA BIBLIOGRÁFICA

- [ABIÁN, 2002] ABIÁN, M. Á. *J2EE Y .NET: La rivalidad permanente*, 2002. [Disponible en: <http://www.javahispano.org/articles.article.action?id=55>]
- [ACR, 2004] ACR and NEMA. *Estándar DICOM*. National Electrical Manufacturers Association, 2004. p.
- [BOOCH, 2000] BOOCH, G.; J. RUMBAUGH, *et al. El Lenguaje Unificado de Modelado*. 2000. p.
- [GROUP, 2007] GROUP, P. G. D. *Sitio de PostgreSQL*, 2007. [Disponible en: <http://www.postgresql.org/>]
- [GROUP, 2006] GROUP, T. P. G. D. *PostgreSQL 8.2.3 Documentation*, 2006.
- [JACOBSON, 2000] JACOBSON, I.; G. BOOCH, *et al. El proceso unificado de desarrollo de software*. 2000. p.
- [LARMAN, 2002] LARMAN, C. *UML y Patrones. 2*. Addison Wesley Professional, 2002. p.
- [MCCALLUM, 2004] MCCALLUM, G. *Construir Aplicaciones EJB con JBoss, Lomboz y Eclipse*, 2004. [Disponible en: [http://www.programacion.com/java/tutorial/jap\\_aplic\\_jboss/3/](http://www.programacion.com/java/tutorial/jap_aplic_jboss/3/)]
- [MICROSOFT, 2007] MICROSOFT.COM. *MDSN - Visual C# Developer Center*, 2007. [Disponible en: [http://msdn2.microsoft.com/es-es/library/kx37x362\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/kx37x362(VS.80).aspx)]
- [RECIO, 2004] RECIO, F. and D. PROVENCIO. *Manual sobre la plataforma .Net*, 2004. [Disponible en: <http://www.desarrolloweb.com/manuales/48/>]
- [WORSLEY, 2001] WORSLEY, J. and J. DRAKE. *Practical PostgreSQL*. 2001. p.

## BIBLIOGRAFÍA

AG, S. Sitio de Siemens Medical Solutions, 2007.

ANÓNIMO. *NHS Connecting for Health*, 2007 [Disponible en: <http://www.connectionforhealth.nhs.uk>]

ARORA, G.; B. AIASWAMY, *et al.* *C#*. 2002. p.

AUTORES, C. D. *Programación Java Server con J2EE Edición 1.3*. 2002. p.

CERAMI, E. *Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*. O'Reilly, 2002. 304 p. 0-596-00224-6

CORPORATION, M. *Cambio de aplicaciones a .NET*, 2002. [Disponible en: [http://www.microsoft.com/spanish/msdn/articulos/archivo/130902/voices/dotnet\\_MovingJavaApps.asp](http://www.microsoft.com/spanish/msdn/articulos/archivo/130902/voices/dotnet_MovingJavaApps.asp)]

MSDN. *La seguridad en los servicios Web XML*, 2002. [Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/030502/voices/xmlwssec.asp>]

ECHARTE, P. *Introducción a la plataforma .NET y Mono*, 2005

FERGUSON, J.; B. PATTERSON, *et al.* *La biblia de C#*. 2003. p.

FONSECA, R. A. *Programación de funciones en PL/pgSQL para PostgreSQL*, 2002.

GROUP, A.-G. *Sitio de AGFA Corp.*, 2007. [Disponible en: <http://www.agfa.com/en/co/index.jsp>]

IMAGING, D. *Sitio de Dynamic Imaging*, 2007. [Disponible en: <http://www.dynamic-imaging.com/>]

JAVAHISPANO. *Sitio de JavaHispano*, 2005. [Disponible en: [www.javaHispano.org](http://www.javaHispano.org)]

LTD, I. I. S. *Sitio de IMAGE Information Systems Ltd*, 2007. [Disponible en: [http://www.image-systems.biz/index.php?option=com\\_content&task=view&id=56&Itemid=61](http://www.image-systems.biz/index.php?option=com_content&task=view&id=56&Itemid=61)]

MACDONALD, M. *ASP.NET Manual de Referencia*, 2002.

MICROSOFT.COM. *MSDN*, 2007. [Disponible en: <http://msdn2.microsoft.com/en-us/default.aspx>]

MSDN. *Información general acerca de Windows .NET Framework 1.1*, 2003.

PHILIPS. *Sitio de Philips Medical Systems*, 2007. [Disponible en: <http://www.medical.philips.com/main/>]

POSTGRESQL, E. E. D. D. *Tutorial de PostgreSQL*, 2002.

PRADO, T. J. R. y R. T. FERNÁNDEZ. *Introducción a PostgreSQL: Un ejemplo sencillo*, 2002.

SECO, J. A. G. *Excepciones*, 2006.

SOFTWARE, C. *Sitio de Cedara Software Corp.*, 2007. [Disponible en: <http://www.cedara.com/products/pacs/index.htm>]

TOURIS, A. M. y M. P. MARIÑÁN. *Arquitectura empresarial y software libre, J2EE*, 2002.

WALTHER, S. *Nuevas características de seguridad en ASP.NET 2.0*, 2004.

WOLTER, R. *Fundamentos de los servicios Web XML* 2001. [Disponible en: [http://www.microsoft.com/spanish/msdn/articulos/archivo/151102/voices/fundamentos\\_xml.asp](http://www.microsoft.com/spanish/msdn/articulos/archivo/151102/voices/fundamentos_xml.asp)]

## ANEXOS

**Anexo 1 Casos de Uso Expandidos**

<b>Caso de uso</b>	
CU1	Visualizar imágenes.
<b>Propósito</b>	Visualizar una o varias imágenes compatibles con el estándar DICOM v3.0, así como la información en el metadatos asociada a esa o esas imágenes DICOM.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista escoge la(s) imagen(es) que desea visualizar, y a continuación se muestra(n), además de la información del metadatos de cada imagen. Por otra parte el sistema le da la posibilidad de navegar por las imágenes en caso de que el DICOM sea multiframe.	
<b>Referencias</b>	RF1.1, RF1.2, RF1.3, RF1.4, RF1.5, RF1.6, RF1.7, RF1.8
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Este caso de uso comienza cuando el especialista escoge la(s) imagen(es) que desea visualizar. Estas imágenes se pueden encontrar en un servidor DICOM compatible o en la PC.	2. El sistema visualiza la(s) imagen(es) y en la parte derecha de la pantalla principal las miniaturas correspondientes a esa imagen(es).  3. El sistema le da la facilidad de mostrar la información de la metadatos, de navegar por las imágenes y mostrarlas en forma de mosaico. En caso de que las imágenes sean multiframe se puede reproducir utilizando los botones del reproductor que aparece en la parte inferior. Termina el caso de uso.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU2	Generar reporte.
<b>Propósito</b>	Generar un reporte con los resultados del diagnostico del paciente.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista edita todos sus campos para el reporte, los del paciente, los del estudio y los datos generales, y/o escoge la(s) imagen(es) que necesita para imprimir y utiliza el codificar internacional para la enfermedad y la modalidad. Una vez generado el reporte puede ser impreso en un documento o enviado a un repositorio central de reportes.	
<b>Referencias</b>	RF2.1, RF2.2, RF2.3
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<ol style="list-style-type: none"> <li>Este caso de uso comienza cuando el especialista decide emitir un reporte.</li> <li>El especialista selecciona las imágenes que desea reportar, edita los campos necesarios para generar el reporte así como los datos del paciente y del estudio. Escoge dentro de las opciones que ofrece el sistema para el reporte.</li> </ol>	<ol style="list-style-type: none"> <li>El sistema muestra la interfaz para editar el reporte. La misma ofrece la posibilidad de: <ul style="list-style-type: none"> <li>Imprimir el reporte.</li> <li>Imprimir las imágenes.</li> <li>Enviar a un repositorio central de reporte.</li> </ul> </li> <li>Si el especialista selecciona "Imprimir el reporte" o "Imprimir las imágenes" se genera un documento, en el primer caso con los datos del reporte incluyendo las imágenes seleccionadas y en el segundo caso solamente las imágenes.</li> <li>En el caso de enviar reporte a un repositorio central, se crea un documento xml con la información del reporte, se verifica si el servicio web está disponible.</li> <li>Si el servicio web está disponible se envía el</li> </ol>

	<p>reporte, de lo contrario el documento xml se queda almacenado para un próximo intento.</p> <p>Termina el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU3	Buscar imágenes en servidor DICOM
<b>Propósito</b>	Realizar búsquedas en un servidor DICOM compatible.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista mediante la interface de búsqueda y obtención de la bandeja de casos del sistema puede realizar búsquedas en el servidor DICOM. Una vez que tenga la ventana visualizada, escoge uno de los criterios de búsqueda (por paciente o por estudio) que se muestran y llena los datos necesarios y efectúa la búsqueda, el sistema muestra el resultado de la búsqueda.	
<b>Referencias</b>	RF3.1, RF3.2
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>4. Este caso de uso comienza cuando el especialista escoge un nivel de búsqueda (por paciente o por estudio).</p> <p>3. El especialista completa la información necesaria para efectuar la búsqueda y presiona el botón Buscar.</p>	<p>2. El sistema muestra los datos a llenar en dependencia del nivel seleccionado.</p> <p>4. En caso de que los datos estén correctamente se realiza la búsqueda en el servidor DICOM, luego muestra los resultados. De haber algún error en la entrada de los datos el sistema lanza una excepción y pasa al paso 3. Termina el caso de uso.</p>

<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU4	Obtener imágenes en servidor DICOM
<b>Propósito</b>	Realizar la obtención de imágenes desde un servidor DICOM compatible para su estudio.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista mediante la interface de búsqueda y obtención de la bandeja de casos del sistema, después de haber realizado la búsqueda de las imágenes, el sistema le da la posibilidad de obtener dichas imágenes ya sean independientes o formen una serie.	
<b>Referencias</b>	RF3.3, RF3.4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3. Este caso de uso comienza cuando el especialista obtiene resultados en la búsqueda.  3. El especialista selecciona la o las imágenes que desea adquirir y da el botón para obtener.	2. El sistema muestra el botón para obtener.  4. El sistema comienza el proceso de obtención de las imágenes desde el servidor DICOM. Una vez terminado el proceso, se le informa al especialista si fue satisfactorio o no. Termina el caso de uso.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Puntos de extensión.</b>	

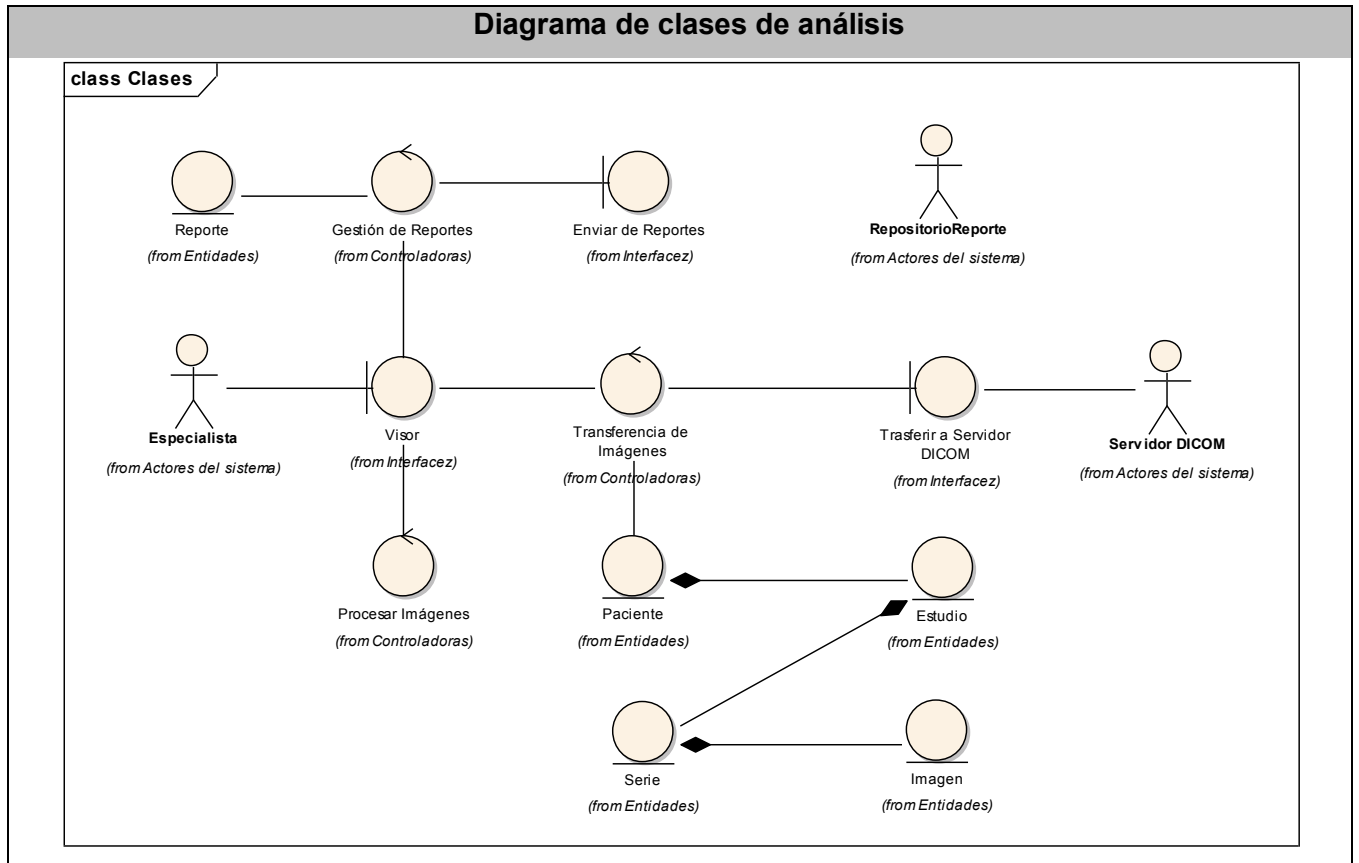


<b>Caso de uso</b>	
CU5	Almacenar imágenes en servidor DICOM
<b>Propósito</b>	Poder almacenar imágenes en el servidor DICOM.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista mediante la interface de la bandeja de casos del sistema puede subir imágenes al servidor. La bandeja de caso visualiza las imágenes que contiene el especialista, selecciona las imágenes que desea subir y presiona botón para almacenarlas en el servidor.	
<b>Referencias</b>	RF3.5
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Este caso de uso comienza cuando el especialista abre la bandeja de casos.	2. El sistema la bandeja de casos con la lista de las imágenes que tiene el especialista.
5. El especialista selecciona las imágenes que desea subir y presiona el botón Almacenar en Servidor DICOM.	4. El sistema comienza a subir las imágenes la servidor DICOM, una vez que termine, el sistema le informará el resultado del proceso. Termina el caso de uso.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU9	Aplicar filtros sobre la imagen.
<b>Propósito</b>	Realizar transformaciones a las imágenes con el objetivo de buscar información que a simple vista no es perceptible.
<b>Actores:</b> Especialista (Inicia).	
<b>Resumen:</b> El especialista selecciona la imagen que desea transformar, selecciona el tipo de filtro, el sistema muestra la imagen que queda como resultado de aplicarle el filtro.	
<b>Referencias</b>	RF5.1, RF5.2.1, RF5.2.2, RF5.3, RF5.4, RF5.5, RF5.6, RF5.7
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1. Este caso de uso comienza cuando el especialista escoge la imagen a la que se quiere realizar la transformación.</p> <p>3. Selecciona el tipo de filtro, de seleccionar Brillo/Contraste se ejecuta 5</p>	<p>2. El sistema muestra la imagen seleccionada. Da la posibilidad de seleccionar el tipo de filtro a aplicar mediante un menú con los siguientes filtros:</p> <ul style="list-style-type: none"> <li>– Brillo/Contraste</li> <li>– Invertir</li> <li>– Escala de gris</li> <li>– Suavizado</li> <li>– Paletas</li> <li>– Ecuilizar</li> <li>– Por paso alto (suave o fuerte) y paso bajo.</li> </ul> <p>4. El sistema muestra la imagen después de ser transformada por el filtro. Termina el caso de uso.</p>
<b>Flujo alternativo: Aplicar brillo y contraste</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
6. El especialista selecciona una de las variantes y ajusta el ancho y centro a su	5. El sistema ofrece dos formas de aplicar brillo y contraste que son las siguientes:

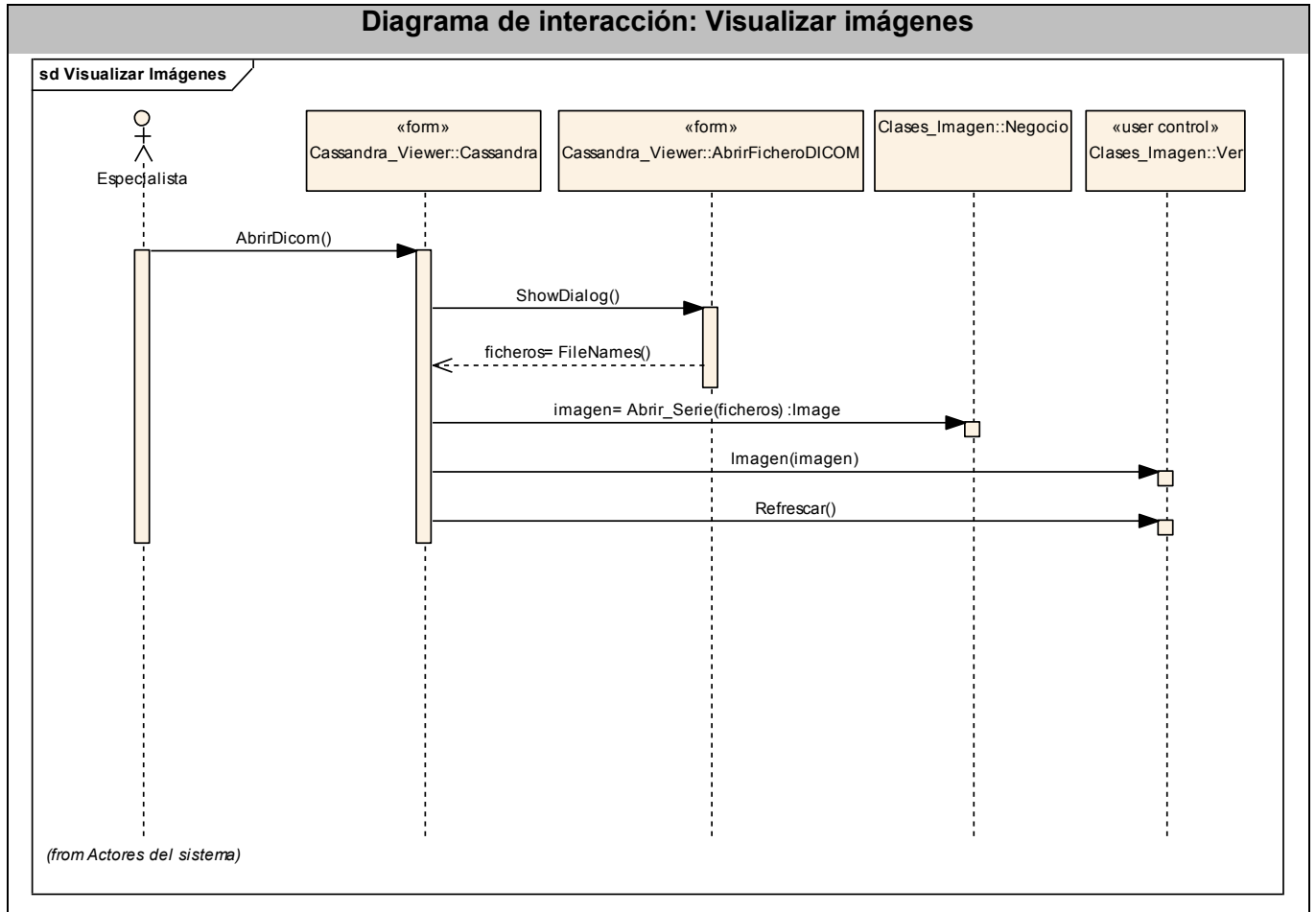
preferencia. Pasa al evento 4.	<ul style="list-style-type: none"><li>- Mediante una ventana donde se define el ancho y centro.</li><li>- Realizando movimientos horizontales (ancho) y verticales (centro) sobre la imagen con el clic derecho presionando.</li></ul>
<b>Puntos de extensión.</b>	

### Anexo 2 Modelo de Clases del Análisis

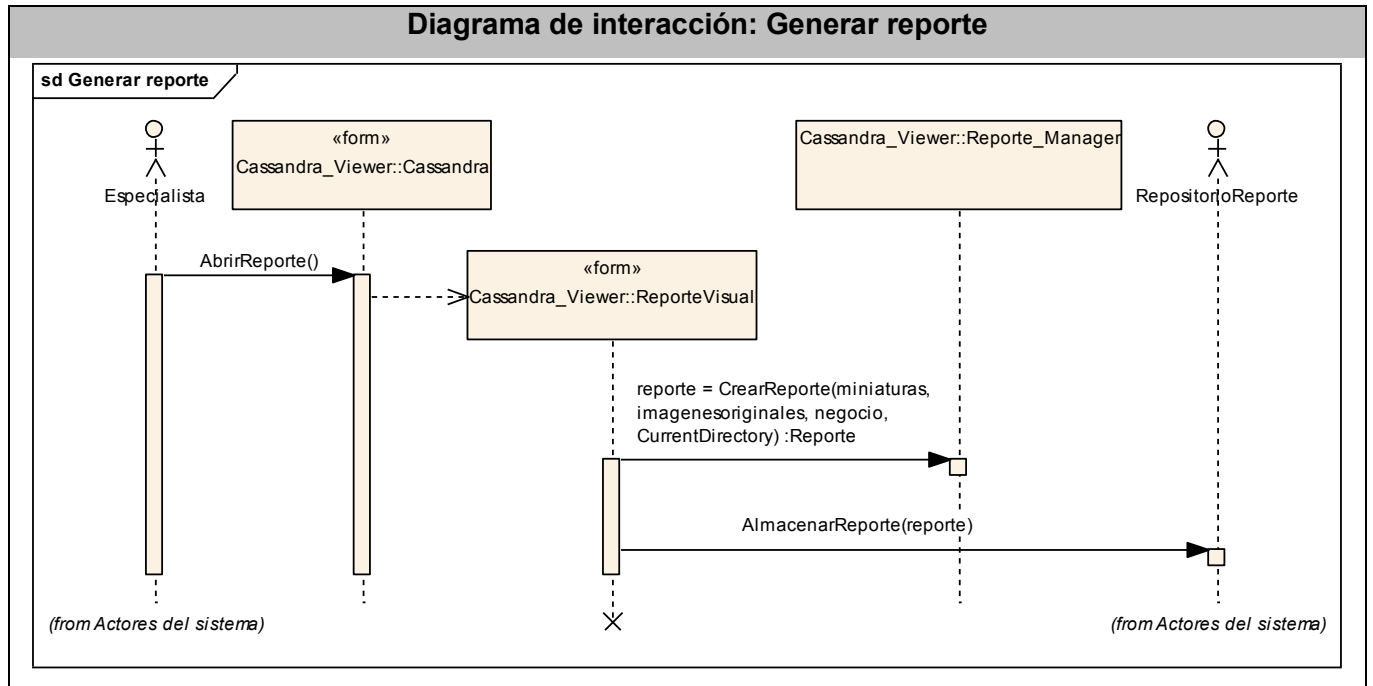


### Anexo 3 Diagramas de Interacción

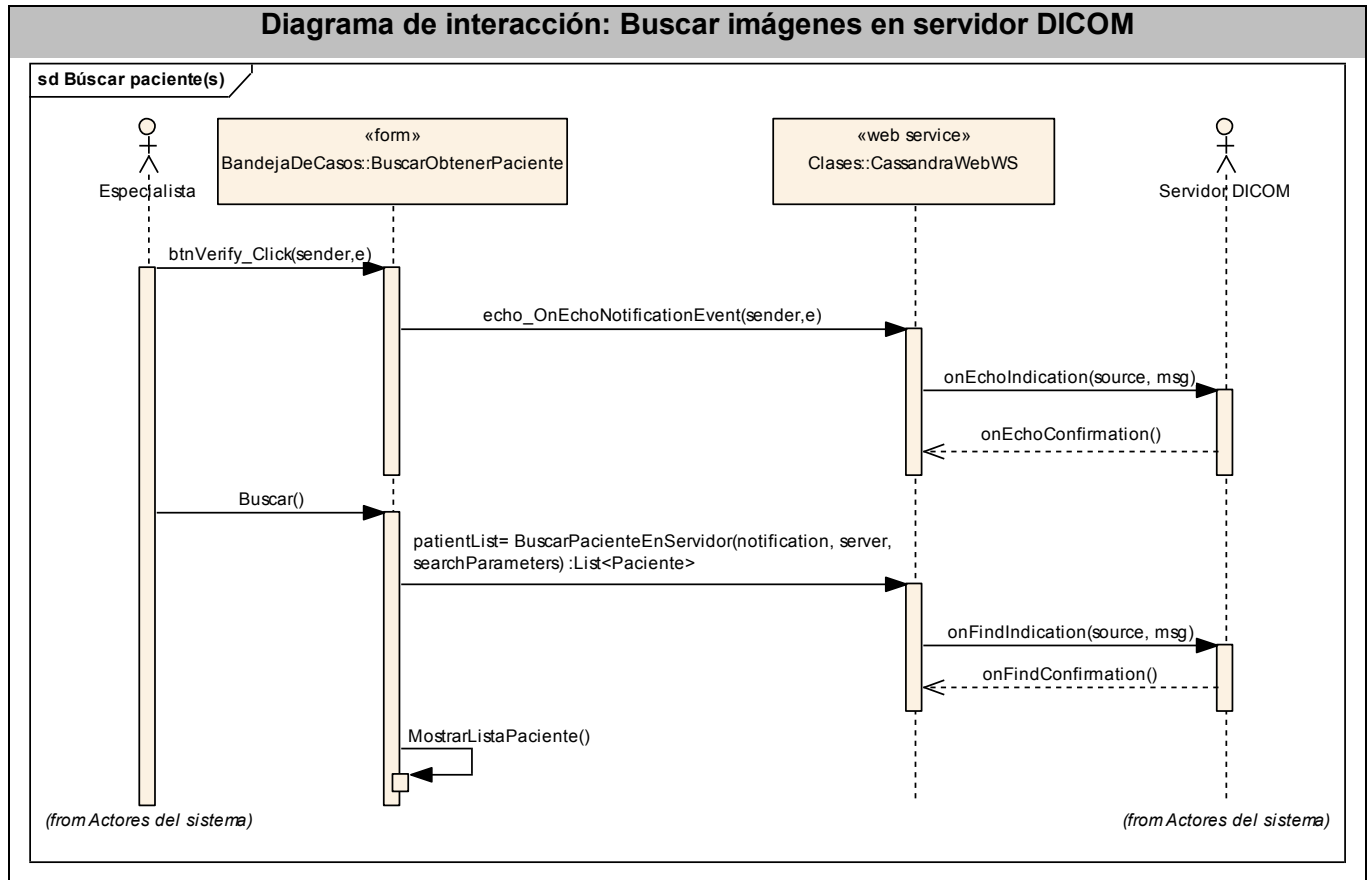
#### 3.2.1.1. Diagrama de interacción – Visualizar imágenes



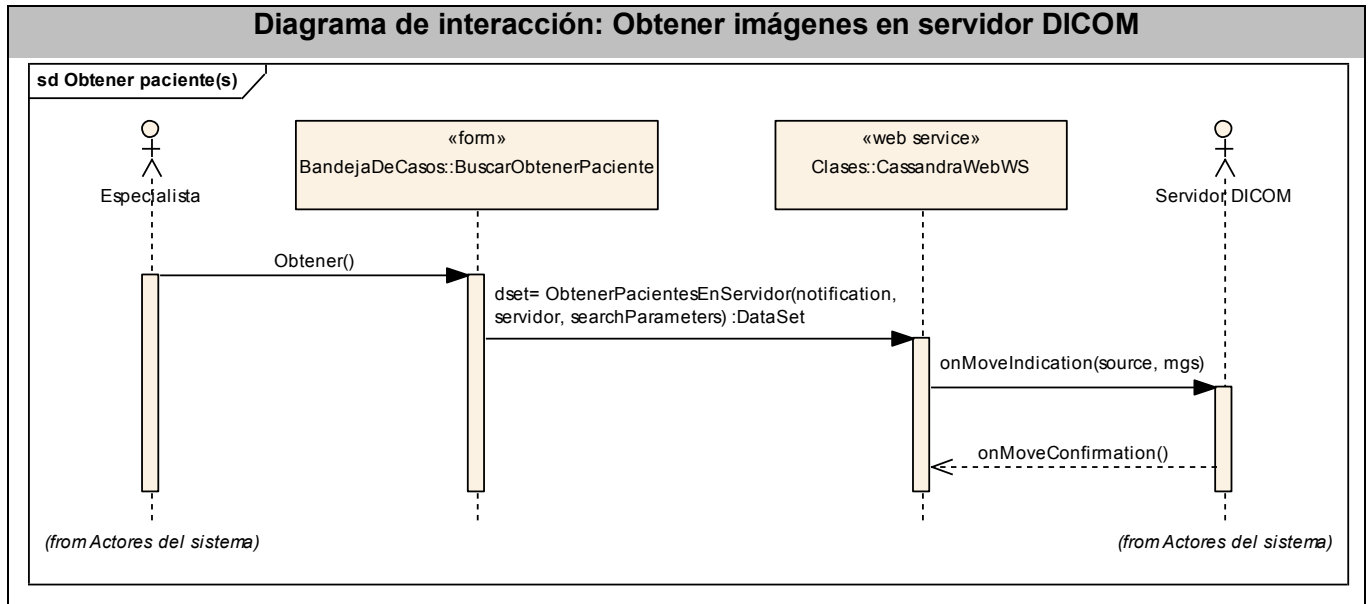
3.2.1.2. Diagrama de interacción – Generar reporte



3.2.1.3. Diagrama de interacción – Buscar imágenes en servidor DICOM

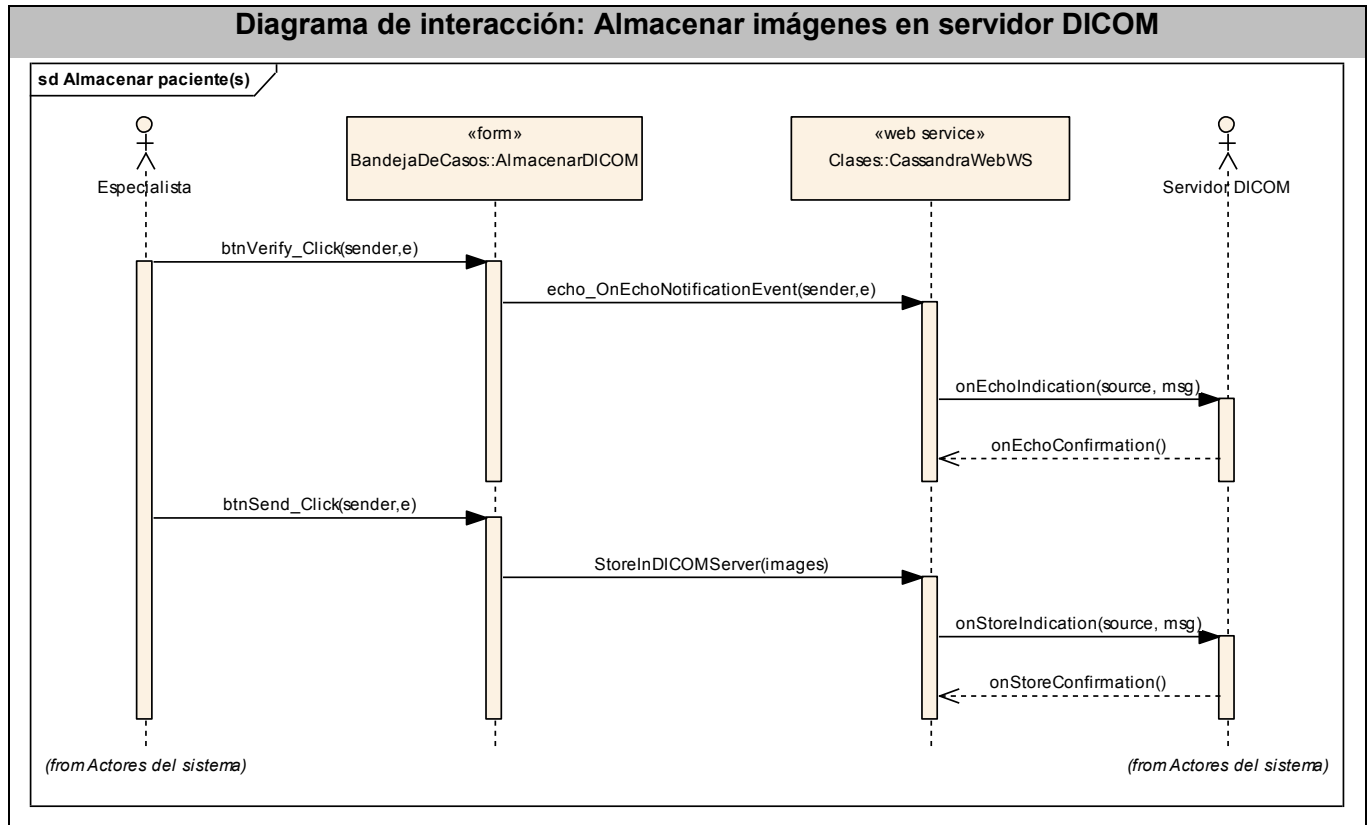


3.2.1.4. Diagrama de interacción – Obtener imágenes en servidor DICOM

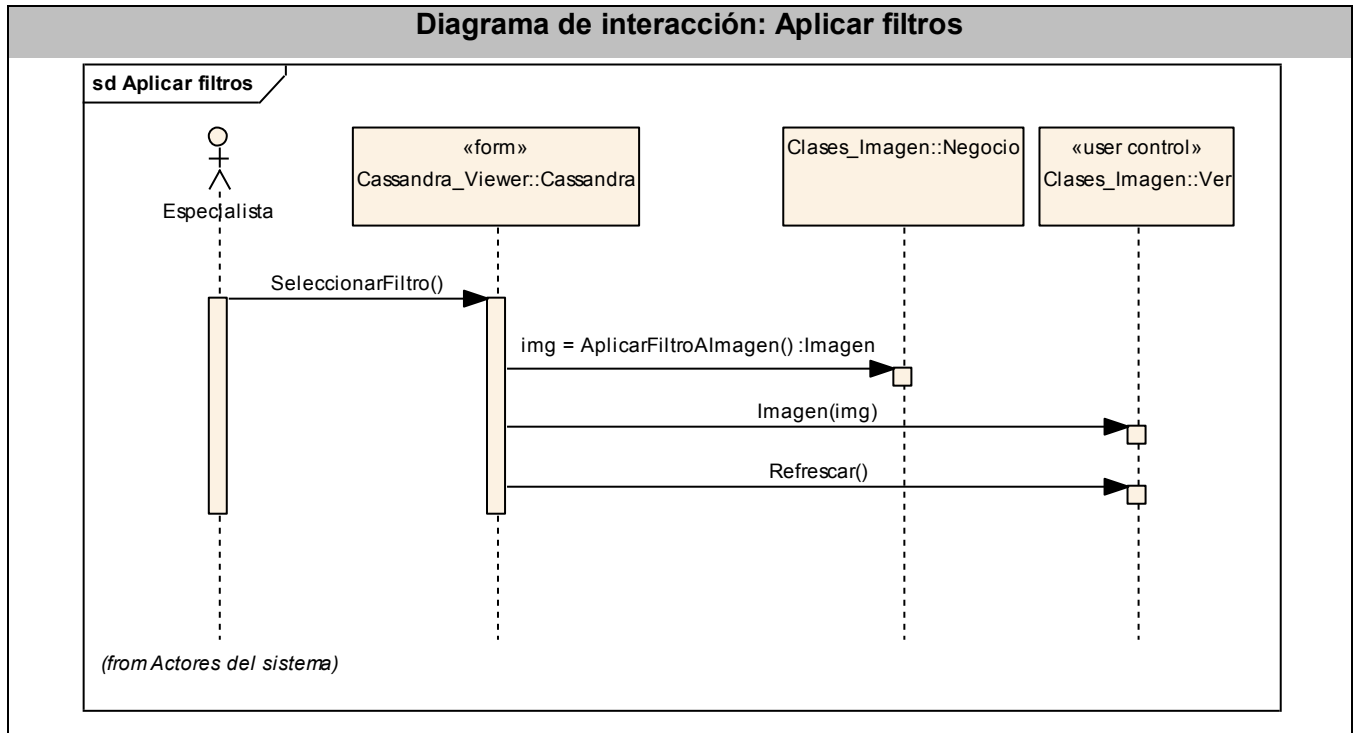




3.2.1.5. Diagrama de interacción – Almacenar imágenes en servidor DICOM

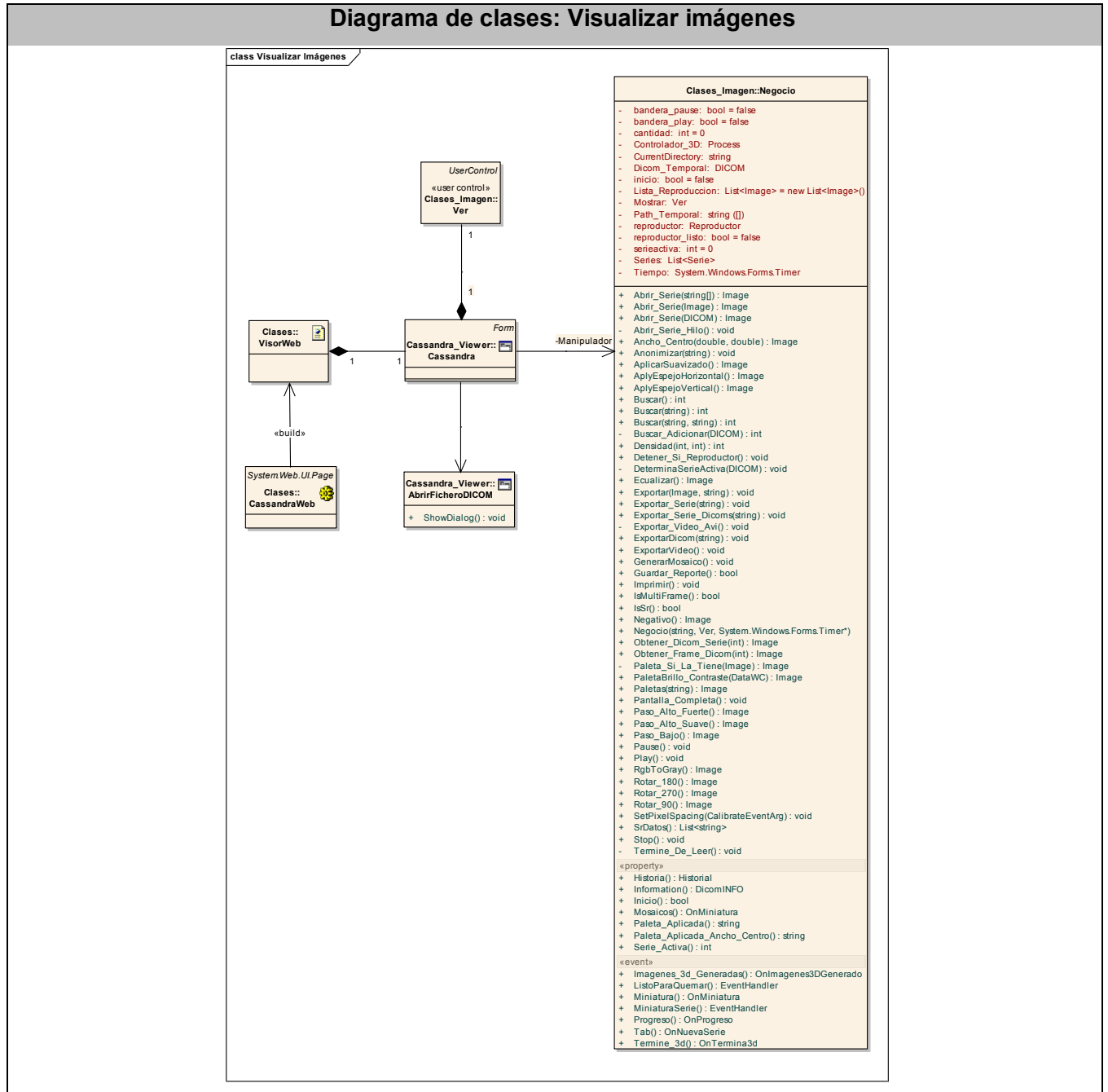


3.2.1.6. Diagrama de interacción – Aplicar filtros

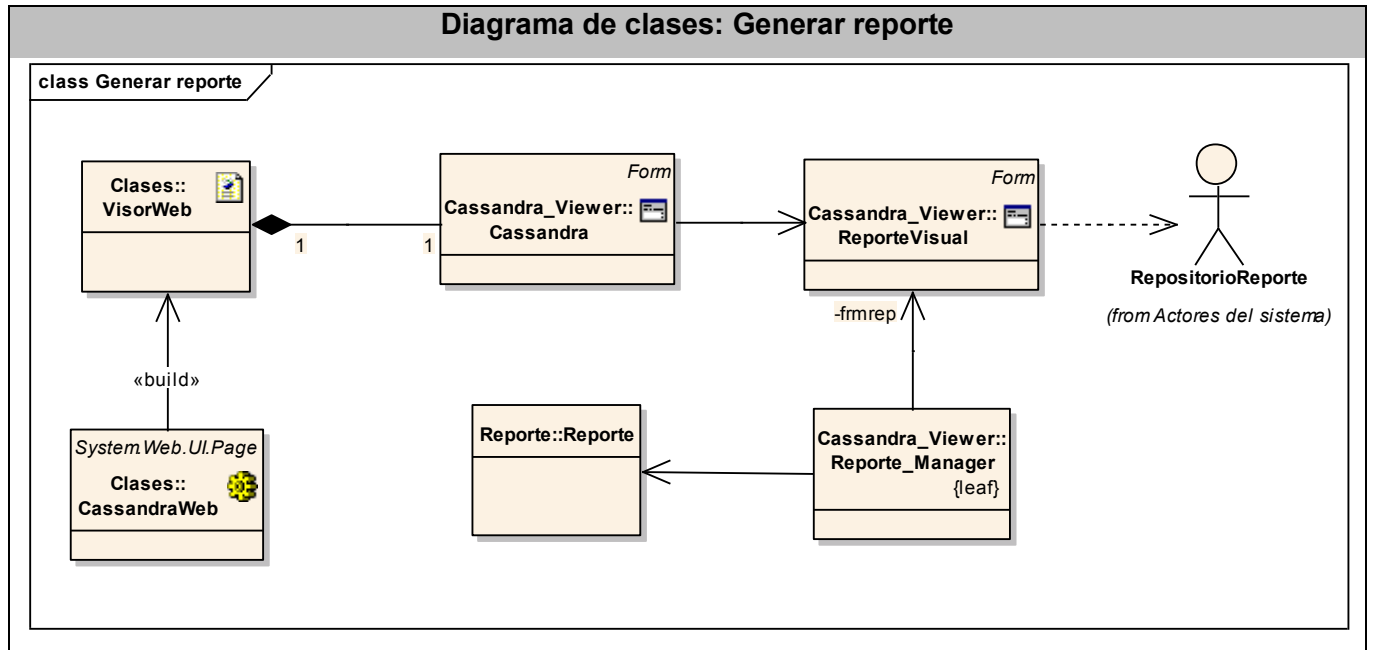


Anexo 4 Diagramas de Clases

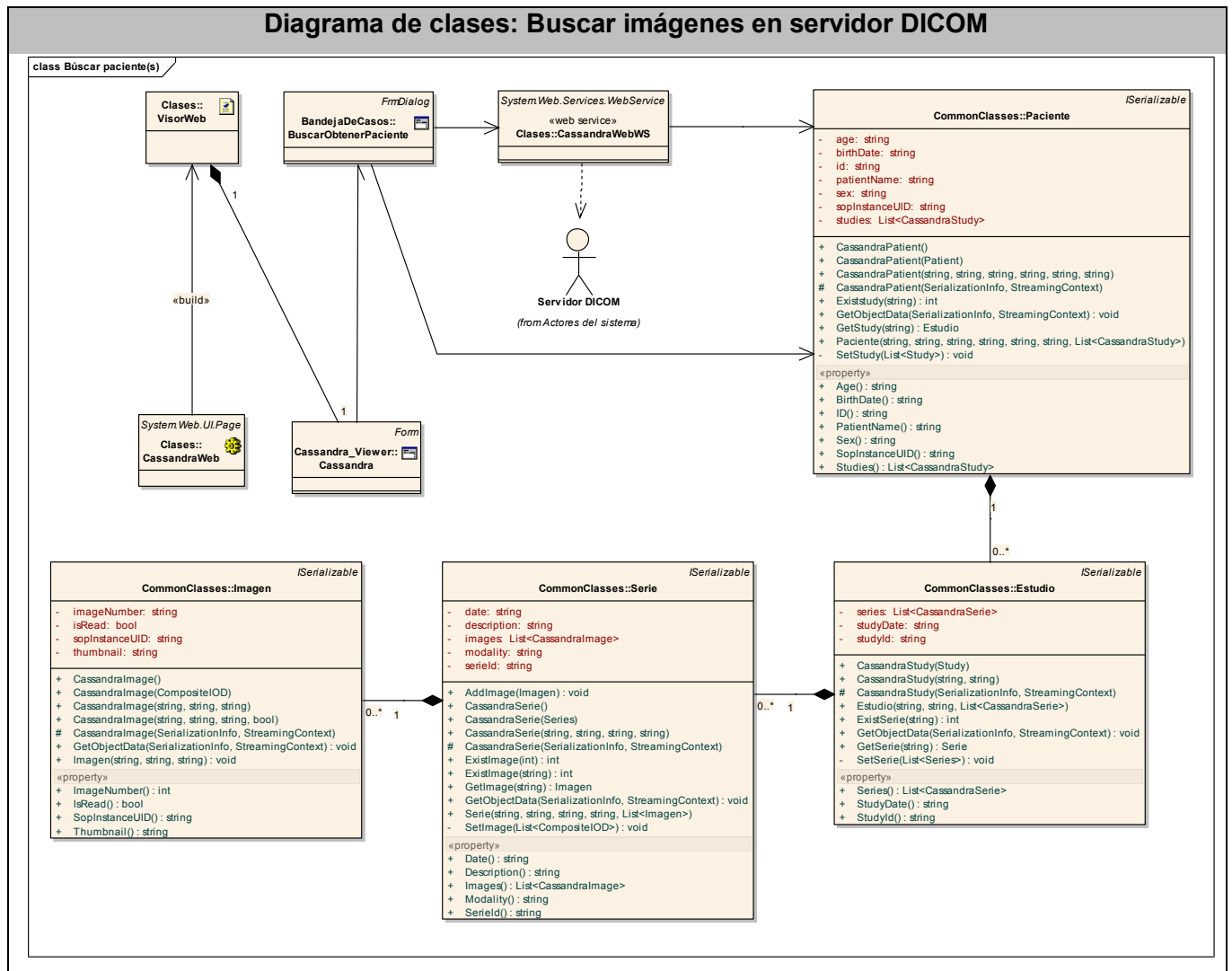
3.2.2.1. Diagrama de clases del diseño – Visualizar imágenes



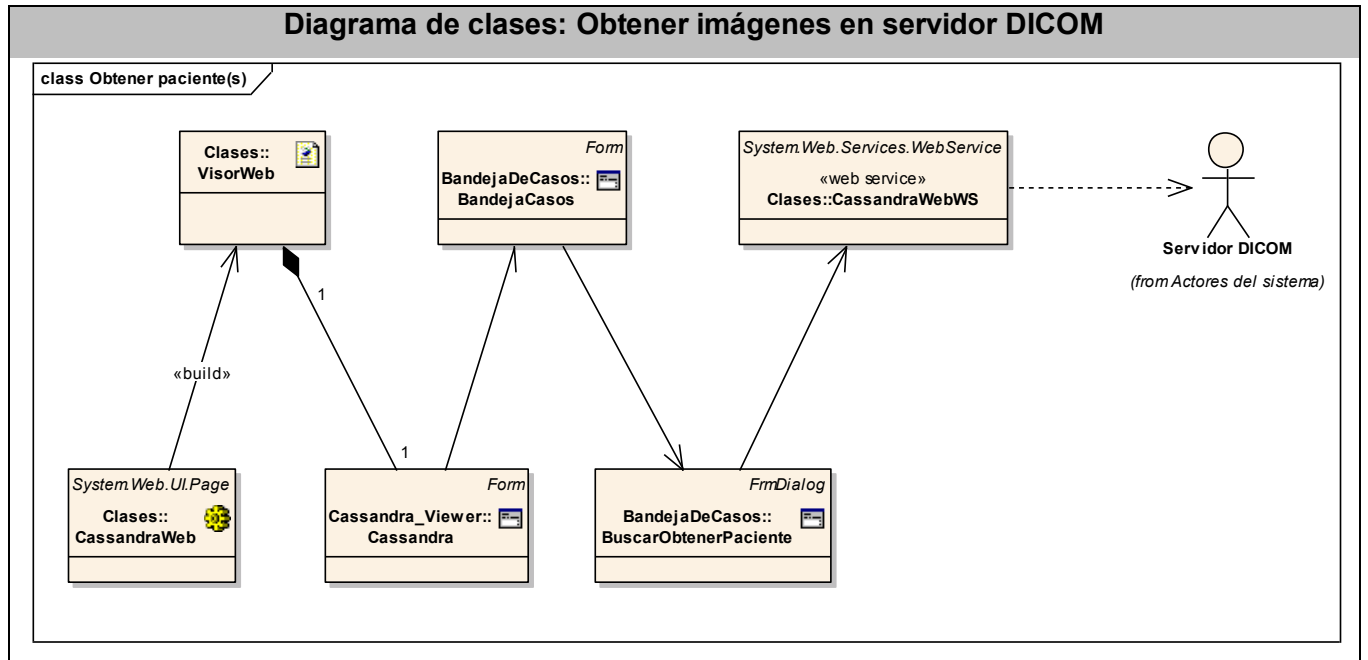
3.2.2.2. Diagrama de clases del diseño – Generar reporte



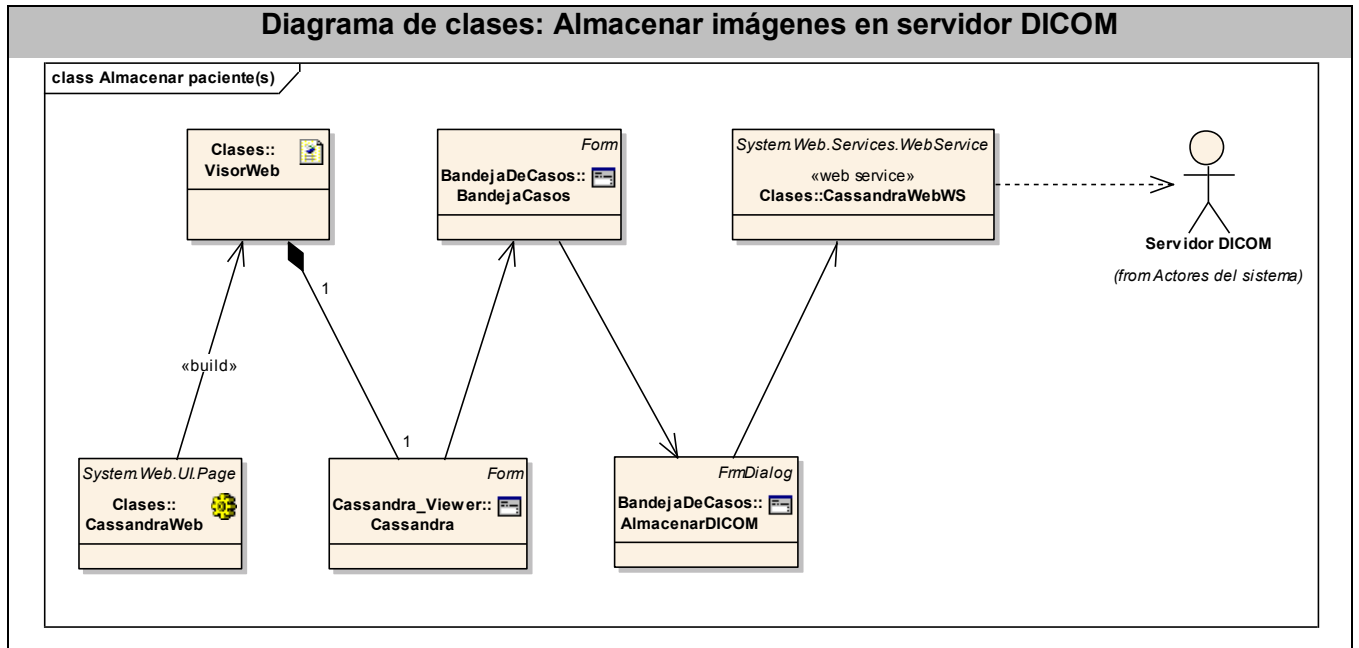
3.2.2.3. Diagrama de clases del diseño – Buscar imágenes en servidor DICOM



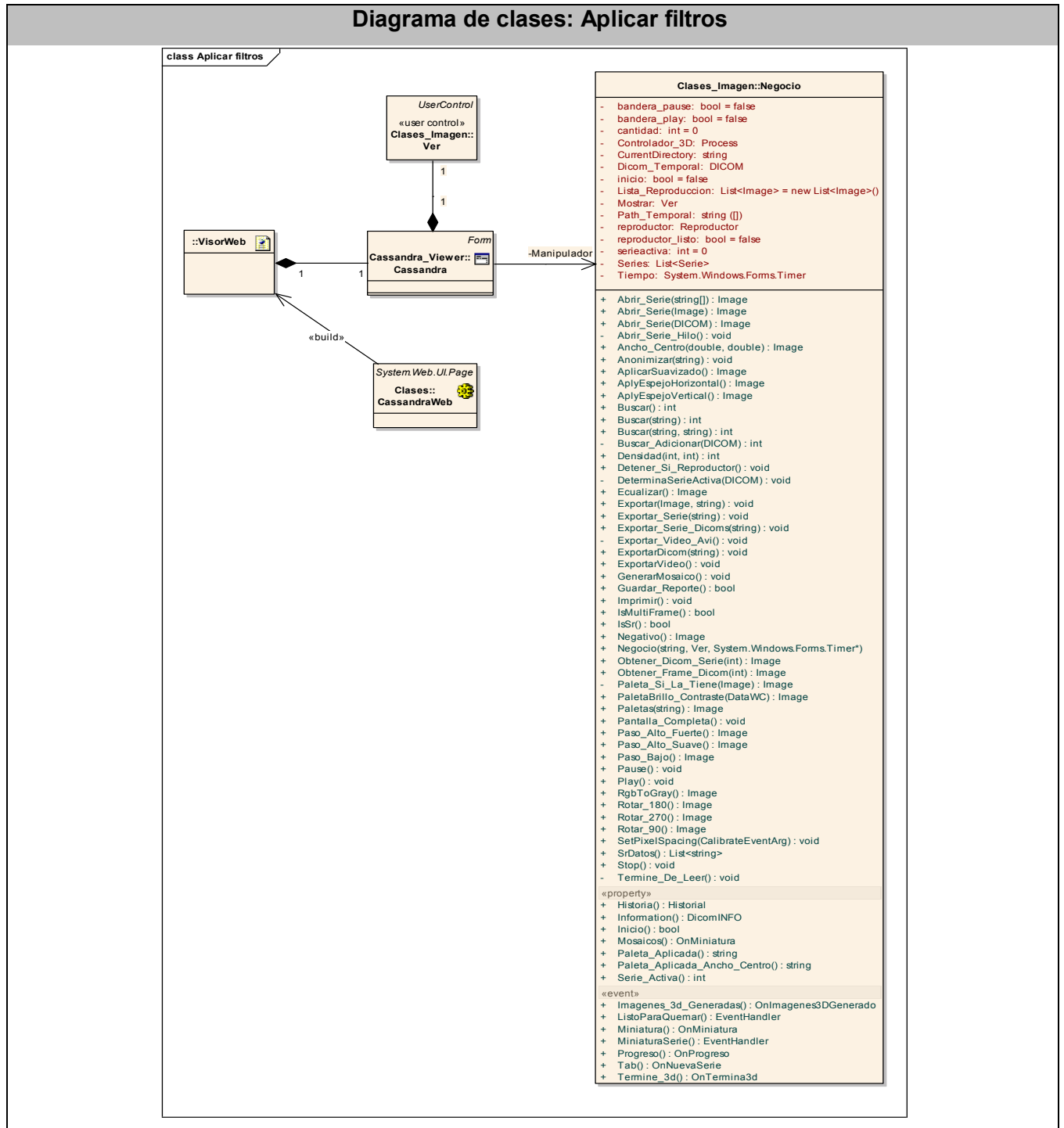
3.2.2.4. Diagrama de clases del diseño – Obtener imágenes en servidor DICOM



3.2.2.5. Diagrama de clases del diseño – Almacenar imágenes en servidor DICOM



3.2.2.6. Diagrama de clases del diseño – Aplicar filtros





**Anexo 5 Descripción de las Clases**

<b>Nombre:</b> Paciente	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	string
edad	string
fechaNacimiento	string
nombrePaciente	string
sexo	string
sopInstanceUID	string
estudios	List<Estudio>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
ID	Retorna el ID del paciente.
Edad	Retorna la edad del paciente.
AñoNacimiento	Retorna el año de nacimiento del paciente
NombrePaciente	Retorna el nombre del paciente.
Sexo	Retorna el sexo del paciente.
SopInstanceUID	Retorna el SopInstanceUID.
Estudios	Retorna todos los estudios que tiene el paciente.
ExisteEstudio	Verifica si un estudio dado su ID pertenece al paciente.
GetEstudio	Retorna el estudio dado su ID si pertenece al paciente.

<b>Nombre:</b> Estudio	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idEstudio	string
fechaEstudio	string
series	List<Serie>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
IDEstudio	Retorna el id del estudio.
FechaEstudio	Retorna la fecha de realización del estudio.
Series	Retorna todas las series correspondientes al estudio.
ExisteSerie	Verifica si una serie dado su ID la contiene el estudio.
GetSerie	Retorna la serie dado su ID si se encuentra dentro del estudio.

<b>Nombre:</b> Serie	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idSerie	string
fecha	string
descripción	string
imágenes	List<Imagen>
modalidad	string
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
IDSerie	Retorna el ID que le corresponde a la serie.
Fecha	Retorna la fecha de realización de la serie.
Descripción	Retorna la descripción de la serie.
Imágenes	Retorna las imágenes que le corresponden a la serie.
Modalidad	Retorna la modalidad a la que pertenece la serie.
AdicionarImagen	Adicionar una nueva imagen a la serie.
ExistelImage	Verifica si una imagen dado su ID pertenece a la serie.
GetImage	Retorna la imagen dado un ID si se encuentra en dentro de las serie.

<b>Nombre:</b> Imagen	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
númerolimagen	string
esLeida	bool
sopInstanceUID	string
ruta	string
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
Númerolimagen	Número consecutivo que identifica a la imagen dentro de la serie.
EsLeida	Informa sobre el estado de la imagen con respecto a si ha sido analizada o no por el especialista.
SopInstanceUID	Retorna el Sop Instance UID
Ruta	Retorna la dirección física de donde se encuentra la imagen en el servidor.

<b>Nombre:</b> Reporte	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
comentario	string
tratamiento	string
recomendación	string
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
Comentario	Retorna el comentario del especialista.
Tratamiento	Retorna el tratamiento emitido por el especialista.
Recomendación	Retorna la recomendación que sugiere el especialista.

<b>Nombre:</b> Negocio	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
AplicarFiltroAlmagen	Transforma la imagen en dependencia del filtro que se haya seleccionado.
ExportarVideo	A partir de las imágenes multiframe es esportada al formato avi.
ExportarSerie	Exporta todas las imágenes de una serie a formato jpg, bmp o tif.
ExportarImagen	Exporta una imagen a formato jpg, bmp o tif.

<b>Nombre:</b> Reporte_Manager	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
CrearReporte	Crea el reporte que es emitido por el especialista, para después enviarlo al repositorio de reporte.

<b>Nombre:</b> Cassandra	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
SeleccionarFiltro	Brinda la posibilidad al especialista de seleccionar el tipo de filtro que desea aplicar a la imagen.
SeleccionarTipoDeMedición	Brinda la posibilidad al especialista de seleccionar el tipo de medición que desea realizar sobre la imagen.
ExportarVideo	Brinda la posibilidad al especialista de poder exportar las imágenes multiframe a video.
ExportarImagen	Brinda la posibilidad al especialista de poder exportar un fichero de imagen.
ExportorSerie	Brinda la posibilidad al especialista de poder exportar todas las imágenes de una serie a fichero de imagen.
AbrirReporte	Brinda la posibilidad al especialista de poder crear un reporte para luego enviarlo a un repositorio de reportes.

<b>Nombre:</b> Ver	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
imagen	Image
mediciones	List<Medición>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>
RealizarMedición	Realiza la medición que haya seleccionado el especialista sobre la imagen.
NuevaMedición	Adiciona una nueva medición a la lista de mediciones.
Refrescar	Refresca los controles visuales, modificando y ajustando la imagen.

<b>Nombre:</b> ReporteVisual	
<b>Tipo de clase (interfaz, controladora, entidad):</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
<b>Método</b>	
<b>Nombre</b>	<b>Descripción</b>



**Anexo 6 Descripción de las Tablas**

<b>Nombre:</b> Especialista		
<b>Descripción:</b> En esta tabla se guardan los datos de los especialistas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Usuario	nvarchar	Usuario del especialista. Es la llave primaria.
Nombre	nvarchar	Nombre del especialista.
Apellidos	nvarchar	Apellidos del especialista.
Especialidad	nvarchar	Especialidad a la cual pertenece el especialista.

<b>Nombre:</b> Estudio		
<b>Descripción:</b> En esta tabla se guardan los datos de los estudios relacionados con cada paciente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdPaciente	varchar	Identificador del paciente. Es una llave extranjera.
IdEstudio	nvarchar	Identificador del estudio. Es la llave primaria.
FechaEstudio	nvarchar	Fecha cuando el paciente se realiza el estudio.

<b>Nombre:</b> Imagen		
<b>Descripción:</b> En esta tabla se guardan los datos de las imágenes que pertenecen a cada serie.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdPaciente	nvarchar	Identificador del paciente. Es una llave extranjera.
IdEstudio	nvarchar	Identificador del estudio. Es una llave extranjera.
IdSerie	nvarchar	Identificador de la serie. Es una llave extranjera.
IdImagen	nvarchar	Identificador de la imagen. Es una llave extranjera.
NumeroImagen	int	Número de orden que tiene la imagen dentro de la serie a la cual pertenece.
Camino	nvarchar	Ruta donde está almacenada la imagen en el servidor.
EsLeida	nvarchar	Campo que indica se la imagen a sido leído o no por el especialista.

<b>Nombre:</b> Paciente		
<b>Descripción:</b> En esta tabla se guardan los datos de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdPaciente	varchar	Identificador del paciente. Es la llave primaria.
Nombre	varchar	Nombre del paciente.
Edad	integer	Edad del paciente
Sexo	varchar	Sexo del paciente
FechaNacimiento	varchar	Fecha de nacimiento del paciente

<b>Nombre:</b> PacienteEspecialista		
<b>Descripción:</b> En esta tabla se guarda la relación que hay entre los especialistas y los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Usuario	nvarchar	Usuario del especialista. Es una llave extranjera.
IdPaciente	varchar	Identificador del paciente. Es una llave extranjera.

<b>Nombre:</b> Serie		
<b>Descripción:</b> En esta tabla se guardan los datos de las series que pertenecen a cada estudio.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdPaciente	nvarchar	Identificador del paciente. Es una llave extranjera.
IdEstudio	nvarchar	Identificador del estudio. Es una llave extranjera.
IdSerie	nvarchar	Identificador de la serie. Es la llave primaria.
Modalidad	nvarchar	Modalidad de la serie.
Descripcion	nvarchar	Un comentario.
FechaSerie	nvarchar	Fecha cuando se realiza la serie.

## GLOSARIO

**ActiveX:** Un software desarrollado por Microsoft y lanzado al mercado en 1997, que permite que programas o contenido sea llevado a computadoras con Windows por medio del World Wide Web. Por ejemplo, poder abrir archivos Word o Excel directamente en el navegador. Los controles Active X tienen muchas libertades dentro del sistema lo cual puede ser inseguro; expertos en seguridad informática recomiendan que no sea usado.

**Anonimizar las imágenes:** Eliminar del fichero DICOM la información relacionada con el paciente y/o el especialista.

**API (Application Program Interface):** Interfaz de Programación de Aplicaciones. Un juego de rutinas usados por una aplicación para gestionar generalmente servicios de bajo nivel, realizados por el sistema operativo de la computadora. Uno de los principales propósitos de un API consiste en proporcionar un conjunto de funciones de uso general, de esta forma los programadores se benefician de las ventajas del API, ahorrándose el trabajo de programar todo de nuevo.

**Apache:** Un servidor de código libre para las plataformas Windows y Linux. Es capaz de apoyar Perl, TCL(Lenguaje de Ordenes para Herramientas) y otros lenguajes para escribir guiones basados en el servidor. Es uno de los servidores de Web más comunes en Internet.

**Aplicación Web:** Aplicación a la que se puede acceder a través de la red.

**Applet:** Pequeña aplicación escrita en Java la cual se difunde a través de la red en orden de ejecutarse en el navegador cliente.

**BMP (bit map):** Extensión BMP, extensión de un fichero gráfico del tipo bit maps.

**Cassandra:** Nombre de la solución (paquete) PACS, para la visualización, el almacenamiento y la transmisión de imágenes médicas.

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

**DirectX:** es un conjunto de APIs creada por Microsoft para mejorar el funcionamiento de aplicaciones multimedia y juegos en Windows. Las funciones DirectX pueden ser utilizadas libremente por los programadores para mejorar sus productos.

**Ecuación:** Redistribución del histograma de una forma equilibrada.

**Estaciones de visualización (Estaciones de diagnóstico):** PC donde el médico tiene la aplicación Cassandra Web Viewer y realiza la visualización de las imágenes médicas, las modifica.

**Ficheros lut:** Ficheros estandarizados para resaltar valores de frecuencias en una imagen, al aplicarlos sobre la misma obtenemos una imagen en colores.

**GNU:** El Proyecto GNU fue creado en 1984 con el fin de desarrollar un sistema operativo tipo Unix según la filosofía del "software libre".

**GPL:** Acrónimo en inglés de General Public License (Licencia Publica General). Regula los derechos de autor de los programas de software libre (free software) promovido por el Free Software Foundation (FSF) en el marco de la iniciativa GNU. Permite la distribución de copias de programas (e incluso cobrar por ello), así como modificar el código fuente de los mismos o utilizarlo en otros programas.

**Histograma:** Frecuencia con que se repite un valor de intensidad en la imagen.

**HTML (HyperText Markup Language):** Es el lenguaje para la representación de la información en la web.

**Imágenes Multiframe:** Secuencia de una imagen en movimiento.

**JPG:** Extensión JPG, extensión de los ficheros de imágenes comprimidas.

**Mainframes:** A los Mainframes también se les conoce con el nombre de grandes ordenadores. Se dedican principalmente a la gestión, pudiendo realizar muchos trabajos a la vez. Una de sus aplicaciones puede ser controlar la red de cajeros automáticos de un Banco. El mainframe será capaz de gestionar la información de todos los cajeros conectados a él.

**Metadata:** Ubicación de toda la información que viene adjunta a la imagen en el fichero DICOM.

**Mono:** es el nombre de un proyecto de código abierto impulsado por Novell para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET según lo especificado por el ECMA.

**Monochrome 1 y Monochrome 2:** Modos de visualización monocromáticos.

**PACS (Picture Archiving and Communication Systems):** Sistema para el almacenamiento y comunicación de imágenes médicas.

**Paso bajo:** Filtro que permite pasar las bajas frecuencias.

**Paso alto:** Filtro que permite pasar las altas frecuencias.

**PDF (Portable Document Format):** Formato portátil de un documento, formato de archivo que conserva el formateo de documentos y facilita a los usuarios ver un archivo en su forma original desde cualquier computadora a través del programa Acrobat Reader (desarrollado por Adobe Systems).

**Repositorio de reportes:** Almacén de datos, donde se guardan los reportes y brinda la posibilidad de que sean accedidos.

**RGB:** Matrices de colores rojo, verde, azul (*Red, Green, Blue*) que especifica los niveles de intensidad de colores en una imagen.

**SOAP:** Protocolo Simple para Acceso a Objetos, estándar de la industria para intercambio de datos con Servicios Web XML.

**SQL:** Structured Query Language. Es un lenguaje especializado de programación que permite realizar consultas (queries) a bases de datos. Los orígenes del SQL están ligados a los de las bases de datos

relacionales. En 1970 Dr. E.F. Codd, investigador de IBM, propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUery Language) que más tarde sería ampliamente implementado por el SGBD experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial. El SEQUEL terminaría siendo el predecesor de SQL. La mayoría de las aplicaciones de bases de datos complejas y muchas otras más pequeñas pueden ser manejadas usando SQL. Es un lenguaje de programación interactivo y estandarizado para extraer información y actualizar una base de datos.

**TCP/IP:** El nombre TCP/IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). En español es Protocolo de Control de Transmisión y Protocolo de Internet. Forma de comunicación básica que usa el Internet, la cual hace posible que cualquier tipo de información (mensajes, gráficos o audio) viaje en forma de paquetes sin que estos se pierdan y siguiendo cualquier ruta posible.

**TIF:** Fichero TIF, fichero gráfico comprimido.

**UDDI (Universal Description, Discovery and Integration):** Es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros

**WEB:** La World Wide Web, la Web o WWW, es un sistema de navegador web para extraer elementos de información llamados "documentos" o "páginas web". Puede referirse a "una web" como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a "la Web", que es la enorme e interconectada red disponible prácticamente en todos los sitios de Internet.

**W3C:** W3C o World Wide Web Consortium. Es un consorcio internacional de compañías, involucradas con Internet y la Web. Su objetivo fundamental es el desarrollo de estándares abiertos para asegurar la evolución de la web en una sola dirección.

**XML:** Lenguaje de marcas utilizado para dar formato a documentos o información en general.