

Universidad de las Ciencias Informáticas

Facultad 7



Servidor de adquisición y ruteo de imágenes médicas

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

Autores: Shirly Lafuente Marcos

Maikel Blanco Dieguez

Tutores: Ing. Lázaro González Rodríguez

Ing. Alejandro Hernández Noguera

La Habana, junio de 2011
“Año 53 de la Revolución”

DATOS DE CONTACTO

Tutores:

Ing. Lázaro González Rodríguez (lgonzalezr@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas, egresado de la Universidad de las Ciencias Informáticas. Ha impartido la asignatura de Práctica Profesional. Es profesor de la Facultad 7 y desempeña el rol de desarrollador en el Departamento de Producción de Software Médico Imagenológico, en la UCI.

Ing. Alejandro Hernández Noguera (anoguera@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el 2009. Ha impartido la asignatura de Introducción a la Programación como profesor de la Facultad 7. Se desempeña actualmente como Implementador en el Departamento de Producción de Software Médico Imagenológico, en la Universidad de las Ciencias Informáticas.

DEDICATORIA

A mis padres, por su amor, ejemplo y confiar siempre en mí.

A mi hermana, porque eres única y maravillosa.

A mis abuelas, por su cariño y ternura.

A Ale, por ser muy especial en mi vida, por ser hacerme sonreír, soñar y amar.

Shirly.

A mi mamá, que todo lo que soy o espero ser se lo debo a ella.

A mi abuela, que fue quien me hizo crecer y siempre ha estado a mi lado.

A mi manito, por ser mi inspiración y para quien construyo mi ejemplo.

Maikel.

AGRADECIMIENTOS

A la Revolución, por habernos dado la oportunidad de desarrollarnos profesionalmente.

A la Universidad de las Ciencias Informáticas.

A André, por ser amigo, por su apoyo incondicional, su preocupación y dedicación.

A nuestro tutor, profesor Lázaro, por transmitirnos su experiencia y ayudarnos.

A los profesores que nos han formado como profesionales.

A nuestros amigos y a los que aportaron al desarrollo de este trabajo.

AGRADECIMIENTOS

A mis padres, por ofrecerme su conocimiento, enseñanza y apoyo incondicional para llegar hasta aquí, porque siempre están presentes. Gracias por ser los mejores padres que cualquiera puede desear.

A mi compañero de tesis, por compartir juntos la lejanía, la tristeza, la alegría y el trabajo de hacer realidad este gran sueño.

A mis amigos, por haber estado presente durante estos años. A Mary, Yisi, Hildín, Dayi y Yisel por ser estupendas, por estar siempre juntas en los buenos y malos momentos, las quiero mucho. A Denys, Alexito, José y Leyva, por su ayuda, preocupación y todo el aliento que me han ofrecido para seguir adelante.

A mi novio Ale, por quererme, comprenderme, brindarme siempre su amor y por ser la razón de mi felicidad.

Shirly.

Son muchas las personas a las que me gustaría agradecer por su amistad, apoyo, ánimo y compañía, quiero darle las gracias por formar parte de mí y por todo lo que me han brindado.

A mi familia que no me equivoco si digo que es la familia más hermosa del mundo, gracias por su apoyo, esfuerzo y toda la confianza que depositaron en mí, gracias porque aunque lejos, siempre han estado a mi lado.

A mi compañera de tesis, Shirlispín, que es una compañera maravillosa y si no fuera por ella este sueño no hubiera sido posible.

A mis amigos que me apoyaron en todo momento, en especial a Denys, por ayudarme y darme siempre una idea mejor.

Maikel.

RESUMEN

La presente investigación tiene como objetivo el desarrollo de un servidor de adquisición y ruteo de imágenes médicas. Es un sistema orientado al área de radiología de una clínica hospitalaria, con el fin de distribuir las imágenes generadas por los equipos de adquisición.

Se ha elaborado según lo estipulado en el estándar DICOM 3.0 para el almacenamiento y transmisión de las imágenes médicas. Fue desarrollado sobre plataforma .NET con lenguaje de programación C#, utilizando el Framework 2.0, lo cual facilitará la migración hacia una plataforma libre. Incluye tareas de pre-procesamiento a las imágenes. Posee una interfaz amigable que posibilita que la aplicación sea fácil de configurar.

La implantación del sistema proporciona la disminución de las conexiones al servidor de almacenamiento de imágenes, un equilibrio en la carga del servidor de almacenamiento y la red y permite que los paquetes enviados sean de menor tamaño y capacidad, así como el procesamiento de la información con un costo bajo en cuanto a almacenamiento.

Palabras claves:

DICOM, Distribución de imágenes médicas, Gateway, PACS.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Sistemas PACS.....	5
1.2. Estándar DICOM.....	6
1.3. Normas IHE	6
1.4. Servidor de imágenes	7
1.5. Compresión de imágenes	8
1.6. Puerta de enlace de adquisición de imágenes (Acquisition Gateway)	8
1.7. Sistemas basados en reglas	9
1.8. Estado del arte de los sistemas de distribución de imágenes médicas.....	9
1.9. Herramientas y tecnologías utilizadas	13
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	18
2.1. Adquisición y ruteo de imágenes médicas	18
2.2. Propuesta del sistema.....	18
2.3. Modelo de domino.....	19
2.4. Especificación de los Requerimientos del Software.....	20
2.5. Definición de los actores del sistema.....	26
2.6. Diagrama de casos de uso del sistema.....	27
2.7. Descripción de los casos de uso del sistema.....	29
CAPÍTULO 3. ARQUITECTURA Y DISEÑO	34
3.1. Arquitectura	34
3.2. Diseño.....	36
CAPÍTULO 4. IMPLEMENTACIÓN	38
4.1. Diagrama de componentes	38
4.2. Modelo de despliegue	40
4.3. Pruebas preliminares realizadas al sistema.....	40
CONCLUSIONES	43

RECOMENDACIONES.....	44
REFERENCIAS BIBLIOGRÁFICAS	45
BIBLIOGRAFÍA	48
GLOSARIO DE TÉRMINOS.....	50
ANEXOS	52

INTRODUCCIÓN

El concepto de radiología digital fue introducido a finales de 1970 y principios de 1980, con el objetivo de sustituir la placa fotográfica por detector electrónico de alta resolución. A través del desarrollo de los diferentes equipos de adquisición y de las imágenes digitales fue posible mejorar su valor diagnóstico.

A partir de estas y otras tecnologías se hizo necesaria la transmisión, almacenamiento y gestión de imágenes médicas, encargándose de estas tareas los sistemas PACS (Sistema para el Almacenamiento, Transmisión y Visualización de Imágenes Médicas). Estos sistemas están compuestos por los equipos de adquisición de imágenes, la puerta de enlace de adquisición de imágenes (acquisition gateway), la infraestructura de comunicaciones, servidores de bases de datos y de imágenes, y las estaciones de visualización y procesamiento de la información, integrados por las redes digitales y software de aplicación.

Por la necesidad de comunicación de estos sistemas con los equipos médicos de adquisición de imágenes de diferentes fabricantes y/o desarrolladores de software surge el estándar DICOM (Estándar para el intercambio, almacenamiento, visualización e impresión de imágenes médicas), para favorecer el desarrollo y expansión de los PACS, proporcionando un formato común para las imágenes.

La estandarización de las imágenes médicas hace posible que, mediante una transmisión segura, los datos de los pacientes puedan viajar de departamento en departamento y de hospital en hospital, garantizando que toda la información pueda ser vista fuera del equipo de adquisición que le dio origen sin costo adicional al paciente. (1)

La transmisión de imágenes e información textual entre los sistemas de información de salud siempre ha sido difícil por dos razones. En primer lugar, el uso de diferentes plataformas por los sistemas de información, y en segundo lugar, las imágenes y los datos son generados por diferentes fabricantes de diferentes modalidades de imágenes. Con los estándares de la industria emergente de la salud, se ha hecho posible la integración de todas las imágenes médicas heterogéneas y datos de texto en un sistema organizado. Con todos los buenos estándares disponibles, se necesitaba una iniciativa para promover la utilización de los mismos, así surge IHE (Integrating the Healthcare Enterprise), un modelo de información de alto nivel para impulsar la adopción de estándares. (2)

Como consecuencia de los beneficios que traen consigo estas nuevas tecnologías introducidas en la radiología digital para el diagnóstico médico, Cuba comenzó a adquirir modernos equipos médicos que garantizan diagnósticos rápidos y confiables; de los cuales la mayoría genera la información en forma de imágenes digitales.

La Universidad de las Ciencias Informáticas (UCI), se dio a la tarea, dentro del sector de la salud, de lograr que el flujo de la información del paciente se realizara eficazmente y con la calidad necesaria, a través del desarrollo de soluciones tecnológicas que garanticen el procesamiento de la información que se genera en todos los niveles de atención a la salud en Cuba.

El Centro de Informática Médica (CESIM) de la UCI desarrolló la solución alas PACS para el almacenamiento, transmisión y visualización de imágenes médicas. Esta solución no cuenta con un procedimiento capaz de realizar una distribución de las imágenes provenientes de los equipos de adquisición.

En la actualidad, el proceso de adquisición y almacenamiento de imágenes se realiza directamente desde los equipos de adquisición y hacia al servidor de almacenamiento. En este proceso los equipos realizan múltiples envíos y a su vez pueden realizar búsquedas y obtenciones desde el servidor. Simultáneamente, las estaciones de visualización efectúan envíos, búsquedas y obtenciones de imágenes. En un ambiente de un departamento radiológico de este tipo, este proceso de transmisión puede ralentizar el flujo de trabajo, aún más, si el servidor almacena grandes volúmenes de información. Todo esto trae consigo una sobrecarga en el servidor, un tráfico excesivo en la red, muchas conexiones concurrentes, un comportamiento inestable del servidor en horarios picos y no se logra la distribución de imágenes provenientes de los equipos de adquisición.

Por lo antes planteado se identifica como **problema a resolver**: ¿Cómo distribuir las imágenes provenientes de los equipos de adquisición?

Este problema se enmarca en el **objeto de estudio**: el proceso de distribución de imágenes médicas. El objeto delimita el **campo de acción** en: el proceso de distribución de las imágenes provenientes de los equipos de adquisición en los departamentos radiológicos.

Para la solución del problema se plantea como **objetivo general**: Desarrollar un sistema capaz de realizar el enrutamiento de las imágenes provenientes de los equipos de adquisición.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas investigativas**:

- Realizar un análisis crítico y valorativo de los sistemas informáticos de distribución de imágenes médicas existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
- Analizar la metodología, plataforma, estándares internacionales relacionados, tecnologías, librerías y herramientas.
- Analizar el flujo de trabajo asociado a la distribución de imágenes médicas, logrando un modelo único como guía para la implementación del sistema.
- Generar los artefactos correspondientes a los Flujos de Trabajo: “Modelamiento del Negocio”, “Requerimientos”, “Análisis y Diseño”, “Implementación” y “Pruebas”.
- Implementar el sistema informático aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.

El trabajo está estructurado en cuatro capítulos, cuyo contenido se describe a continuación:

En el Capítulo 1 se presenta un estudio de las tendencias actuales de desarrollo de los sistemas de adquisición y distribución de imágenes médicas así como los aspectos relacionados a los estándares internacionales de los sistemas para la salud y un análisis de las diferentes tecnologías que se emplean en la solución.

El Capítulo 2 aborda detalladamente la propuesta de solución. Se describen las características del sistema y se presenta el modelo de dominio. Posteriormente, se agregará la especificación de los requisitos funcionales y no funcionales, definiendo y describiendo los casos de uso y los actores.

En el Capítulo 3 se presentará los resultados del flujo de análisis y diseño del sistema, así como los artefactos más importantes generados en este flujo. En el mismo se expondrán los patrones de arquitectura empleados en la solución, con una breve descripción de sus características y el propósito de su uso en el sistema.

En el Capítulo 4 se mostrarán los diagramas de componentes y de despliegue que regirán la fase de elaboración del sistema, así como la descripción de las pruebas preliminares realizadas luego de su implementación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

A continuación se ofrece un estudio de los conceptos fundamentales y estándares internacionales relacionados con la investigación así como las tendencias actuales de desarrollo de los sistemas de adquisición y distribución de imágenes médicas y una exposición de las principales características de las tecnologías, herramientas y metodologías utilizadas para la implementación de la solución propuesta.

1.1. Sistemas PACS

Un PACS es un sistema que brinda la posibilidad de transmitir, procesar y almacenar imágenes médicas en formato digital reduciendo los costos y los riesgos de pérdidas o daño de la información.

La primera referencia a los PACS que se conoce data de 1979 cuando Heinz Lemke, profesor de ciencias de la computación de la Universidad Técnica de Berlín, publicó un documento sobre el procesamiento de imágenes aplicadas y los métodos gráficos por ordenador en un estudio de tomografía computarizada de cabeza. (3)

Un PACS se compone por los equipos de adquisición de imágenes, la puerta de enlace de adquisición de imágenes, la infraestructura de comunicaciones, servidores de bases de datos y de imágenes, y las estaciones de visualización y procesamiento de la información, integrados por las redes digitales y software de aplicación.

Los sistemas PACS desarrollados en la década de 1980 fueron diseñados principalmente sobre una base donde cada módulo de un departamento de radiología funcionaba como una isla aislada sin poder comunicarse con otros módulos. El enfoque era débil porque no se refirió a la conectividad y la cooperación entre los módulos. (4)

El diseño de un PACS, que ahora entendemos, se centra en la conectividad del sistema y la eficiencia del flujo de trabajo. (5) Su infraestructura consiste en uno o varios servidores centrales de bases de datos y de imágenes médicas, que reciben las imágenes desde la puerta de enlace de adquisición vía LAN para su almacenamiento. Las estaciones de trabajo clientes están conectadas al o los servidores, lo cual permite visualizar y manipular de forma interactiva las imágenes médicas.

Este diseño proporciona el marco necesario para la integración de dispositivos de imágenes distribuidos y heterogéneos y hace posible la gestión inteligente de bases de datos de toda la información relacionada con el paciente. Además, ofrece un medio eficiente de ver, analizar y documentar los resultados del estudio. (6)

1.2. Estándar DICOM

El estándar DICOM se inicia en 1985 a partir de la investigación por parte del ACR-NEMA, Colegio Americano de Radiología y Asociación Nacional de Productores de Equipos Eléctricos. (7)

ACR-NEMA, creó un comité para desarrollar un conjunto de normas que sirvieran de base común para varios proveedores de equipos de imágenes médicas con el objetivo de que los equipos recientemente desarrollados pudieran comunicarse y participar en el intercambio de información de imágenes médicas, en particular, en el entorno PACS.

La primera versión surgió en 1985, especificó estándares en la transmisión de mensajes punto a punto, el formato de datos, y presentación e incluyó un conjunto preliminar de los comandos de comunicación y un formato de diccionario de datos. La segunda versión, la ACR-NEMA 2.0, publicada en 1988, fue una mejora a la primera versión, incluyendo tanto las definiciones de hardware y protocolos de software, así como un diccionario de datos estándar. (8)

Una nueva versión fue lanzada en 1992, con el objetivo de incluir protocolos de red, recibiendo un nuevo nombre: DICOM 3.0. En 1996 fue liberada otra versión contando con 13 partes publicadas que constituyen la base de las futuras versiones de DICOM. Los fabricantes rápidamente adoptaron esta versión a sus productos de imagen. (9)

El estándar permite que los sistemas PACS puedan archivar y visualizar imágenes sin importar la modalidad o el fabricante al cual pertenezcan, siempre y cuando estos equipos estén en conformidad con DICOM.

1.3. Normas IHE

Incluso con el DICOM y estándares de salud disponibles, todavía hay una necesidad de un consenso común sobre la manera de utilizar estos estándares para la integración de los sistemas de información heterogéneos de salud sin problemas. IHE es una iniciativa conjunta de la RSNA (Sociedad de Radiología

de Norteamérica) y HIMSS (Healthcare Information and Management Systems Society) que se inició en 1998. La misión era definir y estimular a los fabricantes a utilizar DICOM, HL7 (HealthLevelSeven) y los sistemas de información para facilitar la operación clínica diaria. (10)

IHE define perfiles de integración que utilizan estándares ya existentes para la integración de sistemas de manera que proporcionen una interoperabilidad efectiva y un flujo de trabajo eficiente. Los perfiles de integración IHE comenzaron por primera vez en el dominio de la radiología, creciendo rápidamente a diferentes dominios.

Cada perfil de integración IHE describe una necesidad clínica de integrar sistemas y la solución para llevarla a cabo. Define también los componentes funcionales, a los que se denominan actores IHE, y especifica con el mayor grado de detalle posible las transacciones que cada actor deberá llevar a cabo, basadas siempre en estándares como el de DICOM y HL7. (11)

1.4. Servidor de imágenes

Un servidor de imágenes es un componente importante de un sistema PACS, cuyo objetivo principal es el almacenamiento de imágenes médicas generadas en los equipos de adquisición de imágenes, así como atender las peticiones de las estaciones de visualización.

El almacenamiento tradicional obstaculiza el rápido acceso a las imágenes y no permite gestionar el acceso no autorizado, así como es propenso a la pérdida o deterioro de los exámenes. Sin embargo, un servidor de imágenes brinda la posibilidad de centralizar el almacenamiento de las imágenes en formato digital en un medio de fácil y rápido acceso. Un servidor consta de un servidor de base de datos y de un sistema de archivos.

Las principales funciones de un servidor de imágenes en un sistema PACS son:

- Recibe las imágenes de los exámenes de la puerta de enlace de adquisición.
- Realiza la compresión de imágenes si es necesario.
- Realiza la verificación de la integridad de los datos de las imágenes si es necesario.
- Determina las estaciones de trabajo de destino a las que se enviarán los exámenes.
- Provee a las estaciones de trabajo de servicios de consulta y recuperación de exámenes.

- Actualiza el sistema de gestión de base de datos.
- Garantiza la protección de la información.

1.5. Compresión de imágenes

En la actualidad, la imagen original aún es utilizada para el diagnóstico primario. Después de que el resultado del diagnóstico se ha documentado y adjuntado con el archivo de imagen, la imagen puede ser comprimida con la calidad razonable para su almacenamiento. (12)

Técnicamente, la mayoría de los métodos de compresión de datos de imagen se pueden clasificar en dos tipos: compresión reversible o sin pérdida y compresión irreversible o con pérdida.

Un esquema reversible o sin pérdida alcanza proporciones modestas de compresión pero permite la recuperación exacta de la imagen original del archivo comprimido. Uno de los métodos de compresión de imágenes libre de errores es reducir el tamaño del archivo de imagen al descartar el fondo de la imagen. Se incluye dentro de un rectángulo solo los límites externos de la imagen. Otro de los métodos existentes está basado en la eliminación de píxeles adyacentes repetidos y puede ser utilizado para comprimir filas o columnas de las imágenes, conocido como RLE (Run-Length Encoding). (13)

Un esquema irreversible o con pérdida no permitiría la recuperación exacta después de la compresión, pero podría lograr ratios mucho más altos de compresión. Generalmente, una mayor compresión se obtiene a expensas de la degradación de la calidad de la imagen.

Otro tipo de compresión en los sistemas de imágenes médicas es la selección de imágenes clínicas, se almacenan sólo unas pocas imágenes relevantes, según lo determinado por los médicos, de una serie o varias series de muchas imágenes obtenidas en tiempo real, reduciendo así el número total de imágenes en un archivo de examen. (14)

1.6. Puerta de enlace de adquisición de imágenes (Acquisition Gateway)

Una tarea importante en un PACS es adquirir las imágenes de cada equipo de adquisición. (15) Esta tarea es realizada por el componente puerta de enlace de adquisición de imágenes, el cual suele ser colocado entre las modalidades de imágenes y el resto de la red del PACS.

El funcionamiento de este componente es transparente para los usuarios y totalmente o altamente automático. Adquiere los datos desde los equipos de adquisición radiológicos, convierte los datos al formato estándar de un PACS (Formato de encabezado, orden de bytes, el tamaño de la matriz) que es compatible con formato de datos DICOM, en caso que sea necesario, y remite el estudio al servidor del PACS y/o estaciones de trabajo.

También realiza tareas adicionales como funciones de pre-procesamiento de imágenes, antes que estas sean enviadas al servidor o estaciones de trabajo del PACS. Cada equipo de adquisición tiene un conjunto específico de requisitos para las funciones de pre-procesamiento. (16)

1.7. Sistemas basados en reglas

Los sistemas basados en reglas son uno de los modelos de representación del conocimiento más ampliamente utilizados. Esto es debido a que resultan muy apropiados en situaciones en las que el conocimiento que se desea representar surge de forma natural con estructura de reglas. (17)

Los componentes fundamentales de un sistema basado en reglas son los datos o base de hechos, la base de conocimientos y el motor de inferencia.

Los datos están formados por la evidencia o los hechos conocidos en una situación particular. Este elemento es dinámico, es decir, puede cambiar de una aplicación a otra. Por esta razón, no es de naturaleza permanente y se almacena en la memoria de trabajo.

En situaciones deterministas, las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas. El conocimiento se almacena en la base de conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos. La información almacenada en la base de conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje. (18)

1.8. Estado del arte de los sistemas de distribución de imágenes médicas

Los sistemas de distribución de imágenes médicas son comercializados por lo general como parte de un PACS, pudiendo no ser así en algunos casos. A continuación se abordan los sistemas utilizados como referencia.

1.8.1. Sistemas de distribución de imágenes médicas en el mundo

A nivel internacional la comercialización de sistemas PACS se encuentra fundamentalmente en manos de las grandes compañías como General Electric, Siemens, Philips, iQSystem, Image Information Systems, Digital Imaging, Fujifilm y algunas otras.

A continuación se mencionan sistemas de distribución de imágenes médicas que comercializan algunas de estas compañías.

1.8.1.1. Centricity PACS 3.0 DAS

Centricity PACS 3.0 DAS (DICOM Acquisition Services) es un subsistema de la solución Centricity PACS, desarrollado por la empresa General Electric, una de las principales proveedoras de PACS del mercado. El subsistema es la interfaz de entrada para todos los equipos de adquisición de imágenes en un sistema PACS. Soporta todas las clases de servicios del estándar DICOM, como almacenamiento, búsqueda y obtención, proveedor de clases de almacenamiento (SCP) y proveedor de clases de usuario (SCU). Permite el control de acceso al sistema a través de permisos configurables. (19)

DAS recibe los exámenes desde los equipos de adquisición de imágenes, los organiza, hace las anotaciones pertinentes en la base de datos del PACS, realiza compresión Wavelet a los exámenes Multi-Frame y mueve al servidor de imágenes para su almacenamiento.

Una vez que el examen es archivado, el DAS envía un mensaje de respuesta al equipo que lo envió. Centricity PACS 3.0 DAS trabaja sobre el sistema operativo Linux. (20)

1.8.1.2. Gateway Router

Gateway Router es un producto de la empresa RamSoft, la cual tiene una larga experiencia en el desarrollo de soluciones informáticas. Este sistema organiza la distribución de datos dentro de una red de todo tipo de

imágenes en formato DICOM que viajan desde un dispositivo DICOM que las genera, hacia el servidor y estaciones de trabajo. Comprime estas imágenes DICOM utilizando algoritmos RLE, JPEG Lossy, JPEG Lossless. (21)

Se comunica con otros dispositivos DICOM como son, otros PACS, modalidades de impresoras DICOM, equipos médicos así como servidores DICOM y estaciones de trabajo; para lo cual posee una lista de dichos dispositivos por cuestiones de seguridad. Realiza copias de seguridad a la base de datos. Contiene una opción DICOM conocida como bombardeo que realiza un seguimiento automático de la conectividad a las estaciones de trabajo. (22)

El sistema admite la configuración de las distintas tareas de servicio, con respecto a la frecuencia (en días) y hora de realización, así como el estado de cumplimiento de las mismas. Luego de la configuración es totalmente autosuficiente. Posee una sección de Control de la Calidad que proporciona instrucciones sobre cómo realizar una serie de tareas que son normalmente necesarias en el entorno de imágenes médicas. (23)

1.8.1.3. Sistema iQ-Router

El sistema iQ-Router es un producto desarrollado por la empresa Image Information Systems que se especializa en la oferta de soluciones en el campo de la imagen médica, poseedora de un gran conocimiento en el campo de la compresión de imágenes e integración de las soluciones digitales en los actuales procesos médicos. (24)

Este producto permite la transmisión de imágenes DICOM a través del envío en un formato comprimido, los cuales pueden ser JPEG, JPEG 2000 y Wavelet. Controla el número de conexiones paralelas que serán aceptadas y procesadas. Si se alcanza el máximo, todas las conexiones entrantes adicionales serán rechazadas. No proporciona ninguna interfaz de usuario. (25)

Tiene los datos de las estaciones y su configuración de red para comunicarse con ellas como son el AE Title (Application Entity Title - representación utilizada para identificar los nodos DICOM para comunicarse) y el puerto de red por donde el sistema escuchará las conexiones entrantes. (26)

La cabecera DICOM de las imágenes se puede completar, modificar o borrar con la versión PRO del software. La versión PRO también ofrece la posibilidad de agregar reglas de distribución automática de datos, lo cual permite enviar un tipo de exámenes a una estación de visualización mientras que otros tipos de exámenes de una modalidad específica se envían automáticamente a la estación de otro médico, con la consideración simultánea de horarios. (27)

1.8.1.4. DICOM Gateway Model XG1

El sistema DICOM Gateway Model XG1 fue desarrollado por la empresa Images On Call (IOC), que ofrece una línea de productos de teleradiología y componentes de un PACS que superan las normas establecidas por las principales organizaciones profesionales médicas.

Este modelo recibe imágenes a partir de equipos de adquisición de imágenes para su transmisión a una estación de trabajo o para su almacenamiento en un servidor local. Las imágenes pueden ser recibidas simultáneamente desde múltiples equipos, siendo estos de cualquier modalidad y/o fabricante. (28)

Puede ser configurado para enviar imágenes a cualquier número de estaciones de trabajo en la red. Logra transmisiones de imágenes de 8, 12 y 16 bits por modalidad. Reduce drásticamente la cantidad de datos que deben transmitirse con el uso de avanzados algoritmos de compresión Wavelet, diseñados específicamente para las imágenes médicas. (29)

En todos los casos, las soluciones mencionadas se caracterizan por ser comerciales y de altos costos, lo cual provoca que su adquisición implicaría un elevado gasto para Cuba. Estas soluciones atan al cliente por la configuración y el soporte de las empresas patronas. Presentan dependencias de otros sistemas mayormente propietarios y/o plataformas específicas.

En algunos casos funcionan sobre sistema operativo Windows y utilizan gestores de bases de datos comerciales como SQL Server y Oracle, cuyas licencias son sumamente costosas. En ciertos sistemas ocurre que el número de conexiones aceptadas y procesadas es limitado rechazando las conexiones entrantes que sobrepasen el número aceptado.

1.8.2. Sistemas de distribución de imágenes médicas en Cuba

En Cuba, el único centro que había comenzado a trabajar con el estándar DICOM hasta el año 2005, era el Centro de Biofísica Médica de la Universidad de Oriente. Este centro desarrolló una solución denominada iMagis en el año 1998 que posibilitaba buscar y recuperar imágenes con acceso a una red pero no soportaba múltiples conexiones ni funcionaba como servicio.

En el año 2005, la UCI se dio a la tarea de desarrollar una nueva solución PACS, nombrada alas PACS, que actualmente se utiliza en algunos hospitales de Cuba como el Centro de Investigaciones Médico Quirúrgicas (CIMEQ) y Hermanos Ameijeiras y algunos Centros de Alta Tecnología (CAT) de Venezuela, entre ellos, el Complejo de Salud Integral “Dr. Salvador Allende”.

Esta solución PACS realiza la adquisición de imágenes médicas desde los equipos de adquisición de imágenes a través del sistema alas PACS Server.

Este sistema acepta peticiones, permitiendo la búsqueda y obtención de imágenes desde las estaciones de visualización y los equipos de adquisición, pero no cuenta con un sistema que permita la distribución de los estudios imagenológicos en la red. Razones por las cuales se reafirma la elaboración de un sistema propio que se encargue de este proceso.

1.9. Herramientas y tecnologías utilizadas

A continuación se presenta un estudio realizado de cada una de las herramientas y tecnologías que se decidieron utilizar en la presente investigación.

1.9.1. *Microsoft Visual Studio Team System 2008*

Como entorno de desarrollo integrado (IDE) se utilizará Microsoft Visual Studio Team System 2008. Este IDE provee amplias facilidades como la inclusión de un editor de código que soporta resaltado de sintaxis y completamiento de código, manejo eficiente de errores y características de depuración que brindan información de forma intuitiva.

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, servicios web y windows, aplicaciones de escritorio y aplicaciones móviles. Utiliza como marco de trabajo el .NET Framework 2.0, el cual brinda mejoras como el manejo de datos genéricos y conjuntos de datos, la sincronización de hilos y acceso a datos.

Soporta un diverso grupo de lenguajes de programación tales como Visual C++ .NET, Visual J# .NET, Visual Basic .NET y Visual C# .NET.

1.9.2. Lenguaje de programación C#

Como lenguaje de programación se utilizará C# 2.0. Este es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en el framework .NET. La biblioteca de clases del framework .NET ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles, adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

C# es simple, eficaz, orientado a objetos. Cuenta con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas.

1.9.3. Enterprise Architect 7.5

Como herramienta de modelado se utilizará Enterprise Architect en su versión 7.5. Enterprise Architect es una herramienta CASE (Ingeniería de Software Asistida por Computadora) que combina la potencia de la última especificación UML (Unified Modeling Language) 2.1 y BPMN (Business Process Modeling Notation) 1.1, con un alto rendimiento e intuitiva interfaz de usuario.

Entre sus características fundamentales se destacan la capacidad de Importación/Exportación XMI 2.1, motor de Reporte HTML, transformaciones MDA, perfiles y soporte de tecnologías, pruebas y rastreo de recursos y mantenimiento. (30)

Es una herramienta multi-usuario, basada en Windows, aunque compatible o emulable en sistemas GNU/Linux. Las principales características por las cuales se seleccionó esta herramienta es que posee generación de código en C# e integración con Visual Studio .NET, lenguaje y entorno integrado de desarrollo que se utilizarán para el desarrollo de este sistema.

1.9.4. MyDICOM.NET .SDK.5.2.3

MyDICOM.NET es un conjunto de librerías que implementan el estándar DICOM. Las mismas están desarrolladas sobre el lenguaje C#. Entre sus servicios más importantes están los de transmisión, visualización, impresión y mensajería DICOM. (31)

Tienen un código bien diseñado y portable a otras plataformas como UNIX, con la utilización de MONO. Son fáciles de usar, de ampliar, muy estables y con un gran rendimiento. (32)

Aunque en la actualidad existen varias librerías para el manejo de los servicios del estándar DICOM, entre las que se encuentran Java DICOM Toolkit, CTN, DCMTK, OpenDICOM, se seleccionó MyDICOM debido a que están implementadas en lenguaje C# y su facilidad de uso en las tareas fundamentales.

1.9.5. Subversion 1.6

Para el control de versiones se utilizará Subversion, el cual se puede integrar al entorno de desarrollo Visual Studio.

Gestiona eficientemente archivos binarios a diferencia de otros sistemas de control de versiones. Posee un historial de los archivos y directorios a través de copias y renombrados. Las modificaciones, incluyendo cambios a varios archivos, son atómicas.

Los directorios son versionados así como las operaciones de copiar, eliminar, renombrar. Soporta, pero no requiere, el bloqueo de archivos para que los usuarios pueden recibir avisos cuando varias de ellos intentan editar el mismo archivo. (33)

Como cliente de Subversion se utilizará TortoiseSVN. Es una herramienta desarrollada bajo la licencia GPL. No es una integración de un IDE específico como Visual Studio, Eclipse u otros, se puede usar con cualquier herramienta de desarrollo.

TortoiseSVN es fácil de usar, permite configuración por proyecto e integración con sistemas de seguimiento de errores, posee diálogo de confirmación y corrector ortográfico integrado. (34)

1.9.6. RUP

Rational Unified Process (RUP) es una metodología de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. RUP mantiene al equipo de desarrollo enfocado en producir incrementalmente software operativo a tiempo, con las características y calidad requeridas. (35)

Es un proceso iterativo e incremental. Organiza el desarrollo para poder hacer planificaciones y definir hitos que posibiliten controlar el progreso del proceso de desarrollo. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Una de las mejores prácticas centrales de RUP es la noción de desarrollar iterativamente. Rational Unified Process organiza los proyectos en términos de disciplinas y fases, consistiendo cada una, en una o más iteraciones. Con esta aproximación iterativa, el énfasis de cada flujo de trabajo variará a través del ciclo de vida. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes releases ejecutables. (36)

1.9.7. UML 2.1

Como lenguaje de modelado se utilizará UML (Unified Modeling Language) 2.1 que permite especificar, visualizar, construir y documentar los artefactos que son generados a lo largo del ciclo de vida de un software.

UML está construido sobre los principales conceptos de la orientación a objetos y es por naturaleza el más utilizado para lenguajes orientados a objetos como C++, Java y C#. UML es independiente de la metodología que se desee utilizar, lo que ha provocado sin dudas el amplio uso que tiene en la industria de desarrollo de software. (37)

Su importancia radica en que posee características visuales que ayudan a los integrantes de un equipo de desarrollo (analistas, diseñadores, especialistas de área y programadores) a comunicarse fácilmente, ya sea para establecer colaboración o para facilitar posteriores desarrollos. (38)

1.9.8. MoMa 1.2.4

MoMA (Analizador de Migración a Mono), es la herramienta que se utiliza para analizar los ensamblados y chequear el estado de las operaciones invocadas desde el código. De esta manera se pueden identificar problemas cuando se porte una aplicación desde .NET hacia Mono. (39) La versión de MoMA a utilizar sería la 1.2.4.

En este capítulo se abordaron los conceptos relacionados con la adquisición y distribución de imágenes médicas. Se realizó un análisis del estado del arte y de las tendencias actuales en el desarrollo de sistemas de este tipo. Se expusieron las herramientas y tecnologías a utilizar para el desarrollo del sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

A continuación se describen los procesos que se van a automatizar dentro del flujo de trabajo de la distribución de imágenes médicas en los departamentos radiológicos de las instituciones hospitalarias. Se presentará la propuesta del sistema a desarrollar. Posteriormente, se especificarán los requerimientos funcionales y no funcionales y se detallarán los casos de uso identificados a través de una breve explicación de cada uno.

2.1. Adquisición y ruteo de imágenes médicas

El desarrollo de un sistema de adquisición y ruteo de imágenes médicas tiene como objetivo la distribución de las imágenes provenientes de los equipos de adquisición.

El flujo de procesos se identifica en la distribución de imágenes médicas, en los departamentos radiológicos de las instituciones hospitalarias. Comienza una vez que un equipo de adquisición genera un estudio imagenológico de un paciente y se decide enviar el estudio a un servidor de imágenes y/o estaciones de trabajo que lo necesiten.

Dentro del flujo descrito anteriormente, los procesos a automatizar están representados por el enrutamiento de las imágenes provenientes de los equipos de adquisición. La automatización de este proceso facilita la verificación de la integridad y compresión de las imágenes antes de llegar a su destino, logrando así su normalización.

2.2. Propuesta del sistema

Con el estudio de las funcionalidades existentes en otros sistemas estudiados, lo estipulado en el estándar DICOM 3.0 y las normativas de IHE para la adquisición de imágenes médicas; se plantea crear un sistema que logre adquirir imágenes generadas por equipos de diferentes fabricantes y modalidades, siempre que estos sean compatibles con DICOM 3.0, para distribuirlas a través de la infraestructura de red a las estaciones clientes que las requieran.

Se propone pasar las imágenes adquiridas por un proceso de normalización en el cual se verificaría su integridad y se comprimirían, antes de distribuirlas hacia su destino. La creación de este sistema

proporciona la disminución de las conexiones al servidor, un equilibrio en la carga del servidor y la red y permite que los paquetes enviados sean de menor tamaño y capacidad, así como el procesamiento de la información con un costo bajo en cuanto a almacenamiento.

El sistema propuesto se desarrollará bajo la plataforma .NET utilizando el Framework 2.0, lo cual facilitará en un futuro la migración hacia una plataforma libre.

2.3. Modelo de domino

Debido a que los procesos del negocio no se encuentran bien definidos se propone construir un Modelo de Dominio, el cual proporciona una vista estructural del sistema donde se describen las entidades implicadas en la distribución de imágenes médicas y sus relaciones.

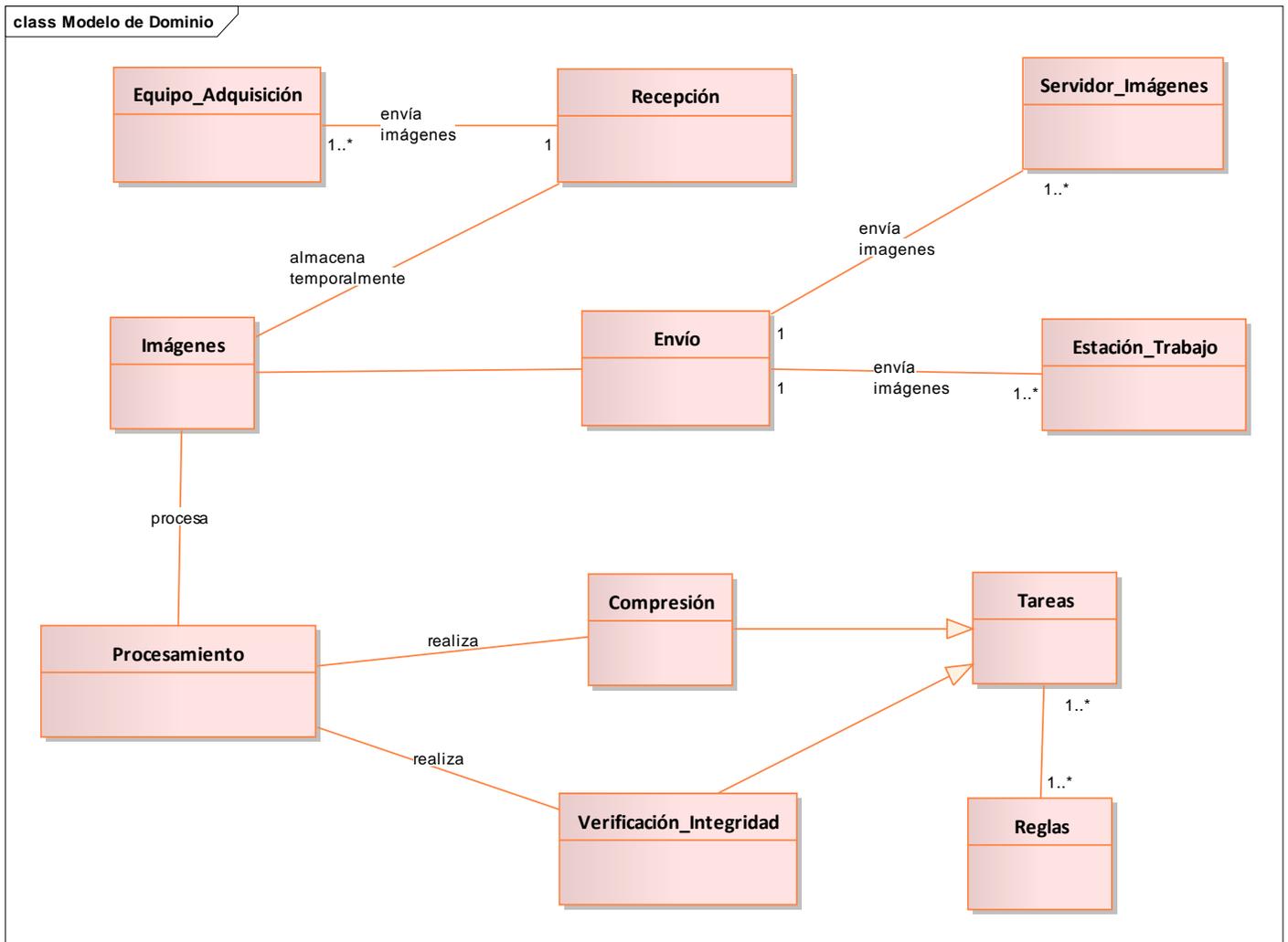


Fig. 2.1. Modelo de Dominio.

2.4. Especificación de los Requerimientos del Software

La Especificación de Requerimientos de Software (SRS) define de forma precisa el producto de software que se va a construir. Los requerimientos para un sistema determinan lo que este debe hacer y definen las restricciones de su operación e implementación. La definición y descripción de los requerimientos logra un acuerdo entre los clientes y los desarrolladores del software.

2.4.1. Requerimientos funcionales

Una vez modelado y relacionado los conceptos del dominio, el modelo se convierte en una base estable para la definición de las actividades que se deben automatizar, las cuales constituyen la base para identificar los requerimientos funcionales del sistema.

La Tabla 2.1 muestra los requerimientos funcionales del sistema.

Requerimiento	Descripción
RF 1 Gestionar fuente.	Permite al administrador del sistema insertar, modificar, eliminar y listar las fuentes.
RF 1.1 Insertar fuente.	Permite al administrador del sistema insertar los datos de los equipos de adquisición (fuentes) que podrán enviar estudios imagenológicos al sistema.
RF 1.2 Modificar fuente.	El administrador del sistema podrá modificar los datos de una fuente existente.
RF1.3 Eliminar fuente.	El administrador del sistema podrá eliminar los datos de una fuente existente.
RF 1.4 Listar fuentes.	Permite al administrador del sistema observar las fuentes existentes en el sistema.
RF 2 Gestionar destino.	Permite al administrador del sistema insertar, modificar, eliminar y listar destinos.
RF 2.1 Insertar destino.	Permite al administrador del sistema insertar los datos de las estaciones clientes (destinos), a las cuales el sistema podrá enviar estudios imagenológicos.
RF 2.2 Modificar destino.	El administrador del sistema podrá modificar los datos de un destino existente.
RF 2.3 Eliminar destino.	El administrador del sistema podrá eliminar los datos de un destino existente.
RF 2.4 Listar destinos.	Permite al administrador del sistema observar los destinos existentes en el sistema.
RF 3 Gestionar regla.	Permite al administrador del sistema insertar, modificar y eliminar las reglas.

RF 3.1 Insertar regla.	El administrador podrá configurar el sistema, insertando los datos de una regla. Una regla especifica las tareas de procesamiento que se pueden realizar a un estudio imagenológico proveniente de una fuente existente y conocer el destino hacia donde se debe enviar.
RF 3.2 Modificar regla.	El administrador del sistema podrá modificar los datos de una regla existente.
RF 3.3 Eliminar regla.	El administrador del sistema podrá eliminar los datos de una regla existente.
RF 3.4 Listar reglas.	Permite al administrador del sistema observar las reglas existentes en el sistema.
RF 4 Recibir imagen.	El sistema debe ser capaz de recibir las imágenes provenientes de las fuentes configuradas anteriormente, para lo cual debe mantenerse en constante escucha.
RF 4.1 Almacenar imagen local.	Una vez recibido una imagen, el sistema debe almacenarla localmente.
RF 5 Procesar imagen.	El sistema debe ser capaz de realizar tareas de procesamiento a las imágenes almacenadas, dichas tareas son compresión y verificación de integridad de los datos.
RF 5.1 Comprimir imagen.	El sistema debe ser capaz de comprimir las imágenes sin degradar su calidad, utilizando el algoritmo especificado en la regla asociada a dicha imagen. La compresión es una de las tareas de procesamiento.
RF 5.2 Verificar integridad.	Los valores de los campos de una imagen serán verificados y/o modificados de acuerdo a lo especificado en la regla asociada a dicha imagen. La verificación de la integridad de los campos es una de las tareas de procesamiento.
RF 6 Enviar imagen.	El sistema debe enviar las imágenes a su destino, de acuerdo a lo especificado en la regla asociada.
RF 7 Enviar solicitud de conexión.	El sistema debe realizar una solicitud de conexión a un destino para conocer si puede enviar las imágenes.

Tabla 2.1. Requerimientos funcionales del sistema

En la **Fig. 2.2** se muestra el diagrama de requerimientos funcionales, agrupados por paquetes lógicos.

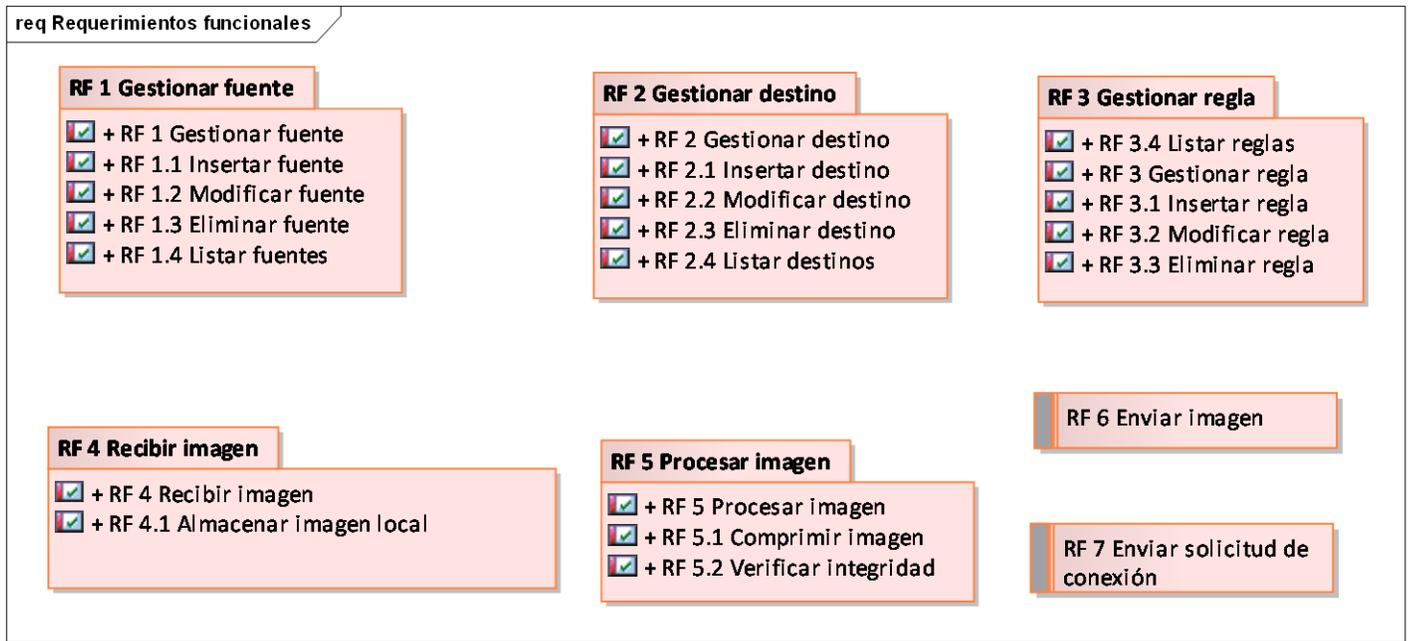


Fig. 2.2. Requerimientos funcionales

2.4.2. *Requerimientos no funcionales*

Los requerimientos no funcionales forman una parte significativa de la especificación de requerimientos. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, como por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, robustez del sistema, disponibilidad, mantenimiento, seguridad, portabilidad, estándares, entre otras.

A continuación se presentarán los requerimientos no funcionales definidos para el sistema. Estos están divididos por categorías y cada una tiene asociado un prefijo con el que será identificada: Funcionamiento: RNFO, Diseño e Implementación: RNDI, Interconexión: RNI, Seguridad: RNS, Interfaz de usuario: RNIU, Eficiencia: RNE, Fiabilidad: RNF.

La Tabla 2.2 muestra los requerimientos no funcionales del sistema.

Requerimiento	Descripción
---------------	-------------

RNFO 1. Memoria RAM de 1GB.	Para el correcto funcionamiento del sistema, se necesita un CPU Dual Pentium IV 3.0GHz, 1 GB de memoria RAM y 250 GB de disco duro como mínimo y se recomienda una tarjeta de red Gigabit Ethernet NIC.
RNFO 2. CPU Dual Pentium IV 3.0GHz.	
RNFO 3. 250 GB de disco duro.	
RNFO 4. Gigabit Ethernet NIC.	
RNFO 5. Sistema operativo Windows Server 2003 o superior.	El sistema debe instalarse sobre el sistema operativo Windows Server 2003 o superior.
RNDI 1. Uso de Framework.Net 2.0.	Se especifica el uso de Microsoft Framework.Net 2.0 que ofrece mejoras en cuanto a administración y rendimiento. El lenguaje de programación C# depende de este Framework.
RNDI 2. Librería MyDicom.NET SDK que implementa el estándar DICOM 3.0.	Utilización de la librería MyDicom.NET SDK que implementa los servicios del estándar DICOM.
RNDI 3. El sistema debe escuchar por alguno de los puertos definidos en el estándar DICOM para realizar la adquisición de imágenes.	El sistema debe escuchar por alguno de los puertos definidos en el estándar DICOM para recibir las imágenes enviadas por los equipos de adquisición.
RNDI 4. El sistema debe realizar el envío de imágenes a los destinos por los puertos especificados por estos de acuerdo al estándar DICOM.	El sistema debe enviar por alguno de los puertos definidos en el estándar DICOM para realizar el envío de las imágenes.
RNI 1. El sistema debe soportar interfaces digitales basadas en el estándar DICOM.	La aplicación debe soportar interfaces digitales basadas en el estándar DICOM, lo que hace más simple y menos costosa su conexión a un PACS.
RNS 1. El sistema debe soportar múltiples solicitudes simultáneamente.	La aplicación debe permitir la recepción de solicitudes de múltiples clientes simultáneamente.
RNS 2. Debe existir un registro de	Cada evento del sistema debe quedar registrado.

todos los sucesos.	
RNIU 1. Las reglas de ruteo deben ser configurables por el usuario.	El sistema debe permitir la configuración de las reglas por un usuario (administrador del sistema).
RNIU 2. Facilidad de configuración del sistema.	El sistema debe tener una interfaz de fácil aprendizaje para que los usuarios puedan familiarizarse rápidamente.
RNE 1. El tiempo de respuesta a las solicitudes debe ser igual o superior a dos imágenes por segundo en imágenes de tamaño medio de 512 Kb.	El tiempo de respuesta a las solicitudes debe ser rápido, igual o superior a dos imágenes por segundo en imágenes de tamaño medio de 512 Kb.
RNF 1. El sistema debe estar disponible siempre.	El sistema es un componente importante en el desarrollo del flujo de trabajo de un PACS por lo cual debe estar disponible siempre.

Tabla 2.2. Requerimientos no funcionales del sistema

En la **Fig. 2.3** se muestra el diagrama de requerimientos no funcionales, agrupados por paquetes lógicos.

pkg Requerimientos No Funcionales1

Funcionamiento

- + RNFO 1. Memoria RAM de 1Gb
- + RNFO 2. CPU Dual Pentium IV 3.0GHz.
- + RNFO 3. Gigabit Ethernet NIC.
- + RNFO 4. Sistema operativo Windows Server 2003 o superior.

Fiabilidad

- + RNF 1. El sistema debe estar disponible siempre.

Interfaz de Usuario

- + RNIU 1. Las reglas de ruteo deben ser configurables por el usuario.
- + RNIU 2. Facilidad de configuración del sistema.

Seguridad

- + RNS 1. El sistema debe soportar múltiples solicitudes simultáneamente.
- + RNS 2. Debe existir un registro de todos los sucesos.

Interconexión

- + RNI 1. El sistema debe soportar interfaces digitales basadas en el estándar DICOM.

Eficiencia

- + RNE 1. El tiempo de respuesta a las solicitudes debe ser igual o superior a dos imágenes por segundo en imágenes de tamaño medio de 512Kb.

Diseño e Implementación

- + RNDI 1. Uso de Framework.Net 2.0.
- + RNDI 2. Librería MyDicom.NET SDK que implementa el estándar DICOM 3.0.
- + RNDI 3. El sistema debe escuchar por alguno de los puertos definidos en el estándar DICOM para realizar la adquisición de imágenes.
- + RNDI 4. El sistema debe realizar el envío de imágenes a los destinos por los puertos especificados por estos de acuerdo al estándar DICOM.

Fig. 2.3. Requerimientos no funcionales

2.5. Definición de los actores del sistema

Descrito el modelo de casos de uso del negocio y los requerimientos funcionales y no funcionales, se da paso a la definición de los actores que interactuarán con el sistema.

Actor	Descripción
-------	-------------

<p>uc Actores</p>  <p>Administrador</p>	<p>Realiza la configuración de las reglas del sistema.</p>
<p>uc Actores</p>  <p>Equipo de adquisición</p>	<p>Sistema externo que realiza las solicitudes de almacenamiento al sistema.</p>
<p>uc Actores</p>  <p>Reloj</p>	<p>Inicia los procesos automáticos de procesamiento y ruteo de imágenes.</p>

Tabla 2.3. Actores del sistema

2.6. Diagrama de casos de uso del sistema

En la Fig. 2.4 se muestra el diagrama de casos de uso del sistema, que representa el comportamiento y la interacción del sistema con los usuarios u otros sistemas.

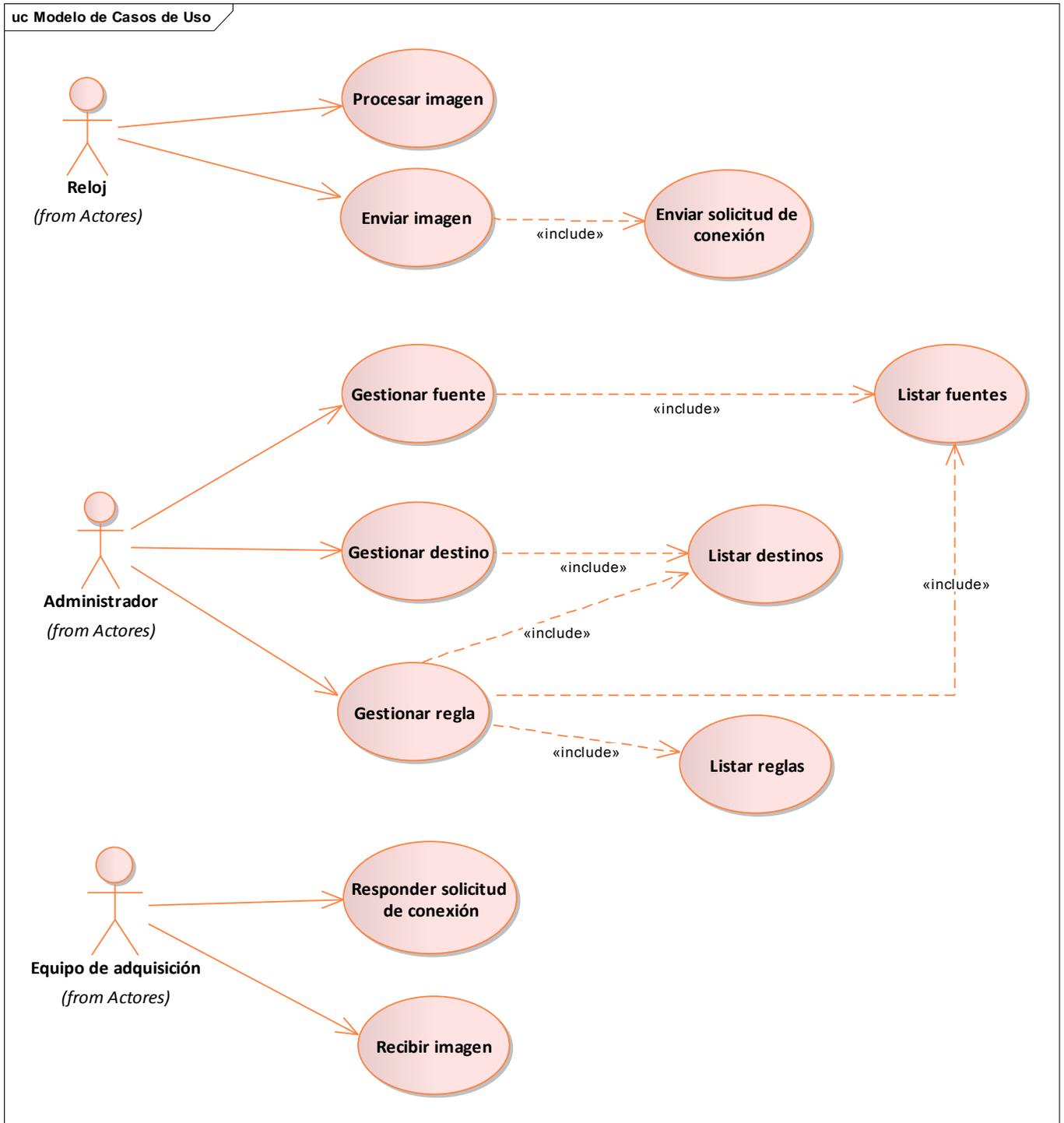


Fig. 2.4. Diagrama de casos de uso del sistema

2.7. Descripción de los casos de uso del sistema

Las tablas que se muestran a continuación muestran un resumen de los casos de uso del sistema. En el [Anexo 2](#) se muestran las descripciones textuales ampliadas de los casos de uso más críticos.

CU # 1. Gestionar fuente	
Objetivo	Insertar, modificar y eliminar fuente.
Actores	Administrador del sistema (inicia).
Resumen	El administrador inserta los datos de los equipos de adquisición (fuentes) que pueden enviar imágenes al sistema. También podrá modificar y eliminar una fuente existente.
Complejidad	Fácil
Prioridad	Crítico
Referencias	RF 1.1, RF 1.2, RF 1.3

Tabla 2.4. Resumen del CU Gestionar fuente

CU # 2. Gestionar destino	
Objetivo	Insertar, modificar y eliminar destino.
Actores	Administrador del sistema (inicia).
Resumen	El administrador inserta los datos de las estaciones clientes (destinos), a las cuales el sistema podrá enviar imágenes. También podrá modificar y eliminar un destino existente.
Complejidad	Fácil

Prioridad	Crítico
Referencias	RF 2.1, RF 2.2, RF 2.3

Tabla 2.5. Resumen del CU Gestionar destino

CU # 3. Gestionar regla	
Objetivo	Insertar, modificar y eliminar regla.
Actores	Administrador del sistema (inicia).
Resumen	El administrador inserta los datos de una regla. Una regla especifica las tareas de procesamiento que se pueden realizar a una imagen y el destino hacia donde se debe enviar. También podrá modificar y eliminar una regla existente.
Complejidad	Fácil
Prioridad	Crítico
Referencias	RF 3.1, RF 3.2, RF 3.3

Tabla 2.6. Resumen del CU Gestionar regla

CU # 4. Listar fuentes	
Objetivo	Mostrar las fuentes existentes en el sistema.
Actores	CU Gestionar fuente, CU Gestionar regla.
Resumen	Es un caso de uso incluido. El sistema muestra las fuentes existentes.
Complejidad	Fácil
Prioridad	Crítico

Referencias	RF 1.4
--------------------	--------

Tabla 2.7. Resumen del CU Listar fuentes

CU # 5. Listar destinos	
Objetivo	Mostrar los destinos existentes en el sistema.
Actores	CU Gestionar destino, CU Gestionar regla.
Resumen	Es un caso de uso incluido. El sistema muestra los destinos existentes.
Complejidad	Fácil
Prioridad	Crítico
Referencias	RF 2.4

Tabla 2.8. Resumen del CU Listar destinos

CU # 6. Listar reglas	
Objetivo	Mostrar las reglas existentes en el sistema.
Actores	CU Gestionar regla.
Resumen	Es un caso de uso incluido. El sistema muestra las reglas existentes.
Complejidad	Fácil
Prioridad	Crítico
Referencias	RF 3.4

Tabla 2.9. Resumen del CU Listar reglas

CU # 7. Recibir imagen	
Objetivo	El sistema debe almacenar las imágenes que son enviadas por un equipo de adquisición.
Actores	Equipo de adquisición (inicia).
Resumen	Un equipo de adquisición envía imágenes al sistema, el cual debe almacenarlas localmente.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF 4.1

Tabla 2.10. Resumen del CU Recibir imagen

CU # 8. Procesar imagen	
Objetivo	El sistema debe procesar todas las imágenes que están almacenadas.
Actores	Reloj (inicia).
Resumen	El sistema debe realizar las tareas de procesamiento que están configuradas a las imágenes almacenadas. Se registra el evento sucedido.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF 5.1, RF 5.1

Tabla 2.11. Resumen del CU Procesar imagen

CU # 9. Enviar imagen	
Objetivo	El sistema debe enviar todas las imágenes procesadas hacia sus destinos, configurados en las reglas.
Actores	Reloj (inicia).
Resumen	El sistema envía las imágenes procesadas hacia su destino y luego las elimina. Se registra el evento sucedido.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF 6

Tabla 2.12. Resumen del CU Enviar imagen

En el capítulo se describió el negocio identificado en la distribución de imágenes médicas en los departamentos radiológicos y las ventajas que traen consigo la automatización de los procesos representados por el enrutamiento de las imágenes provenientes de los equipos de adquisición. Se especificaron los requisitos funcionales y no funcionales. Se identificaron y describieron los casos de uso del sistema así como el modelado del diagrama de casos de uso que muestra las principales funcionalidades del sistema.

CAPÍTULO 3. ARQUITECTURA Y DISEÑO

En el capítulo se define y explica la arquitectura del sistema. Se presentan los diagramas de clases de diseño que intervienen en la realización de los casos de uso. También se modelan los distintos diagramas de secuencia de dichos casos de usos y se brinda una breve descripción de las clases del sistema.

3.1. Arquitectura

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (40) Define el estilo o combinación de estilos para una solución y es esencial para el éxito o fracaso de un proyecto. (41)

El estilo arquitectónico utilizado para la construcción del sistema se incluye dentro de las arquitecturas de flujo de datos o estilos de flujos de datos, se denomina secuencial en lote (*batch sequential*).

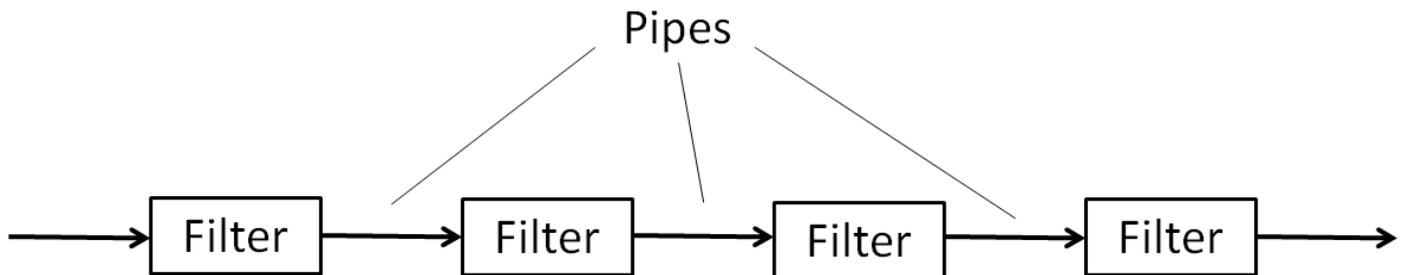


Fig. 3.1. Estilo arquitectónico Batch sequential

Este estilo se basa en el patrón *pipe and filter* (tuberías y filtros), consta de un conjunto de componentes denominados *filtros*, conectados entre si por *tuberías*, que transmiten datos desde un componente hacia el siguiente. Cada filtro trabaja de manera independiente de los componentes que se encuentran situados antes o después de él. Se diseñan de tal modo que esperan un conjunto de datos en un determinado formato y obtiene como resultado otros datos de salida en un formato específico. Cuando el flujo degenera en una única línea de transformación, se denomina secuencial en lote. (42)

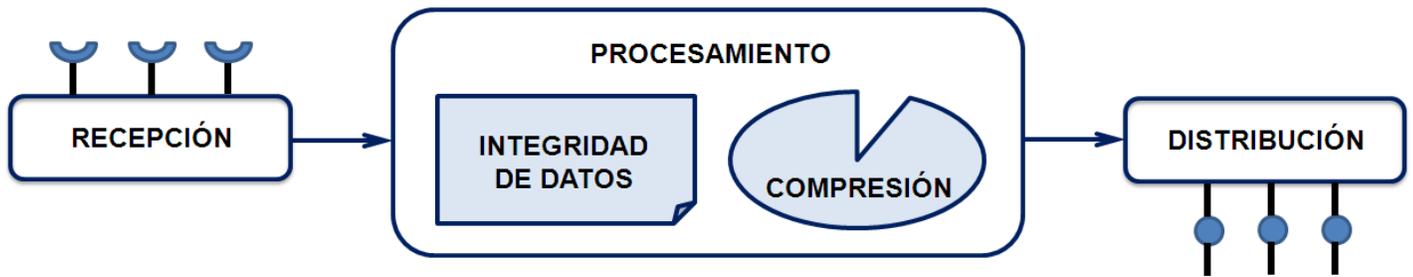


Fig. 3.2. Arquitectura del sistema

El sistema está compuesto por un conjunto de componentes que realizan el almacenamiento, procesamiento y posteriormente la distribución o enrutamiento de las imágenes. Cada uno trabaja de forma independiente y notifica al próximo cuando concluyen sus tareas.

El componente de **Recepción** se encarga de recibir las imágenes generadas por los equipos de adquisición. Al recibir las imágenes, comienza a almacenarlas temporalmente, cada vez que almacena todas las imágenes de un paciente, notifica al componente de **Procesamiento** avisándole que ya puede comenzar a trabajar con las imágenes de dicho paciente. El componente de **Procesamiento** realiza tareas de pre-procesamiento a las imágenes, como son compresión y verificación de la integridad de los datos, según las reglas definidas en el sistema para estas imágenes. Cuando este componente termina de procesar las imágenes de un paciente, notifica al componente de **Distribución** que ya las imágenes fueron terminadas de procesar, este último se encarga de su distribución o enrutamiento hacia el servidor y/o estaciones de trabajo del PACS, según las reglas.

El registro de los sucesos que tienen lugar en el sistema es un aspecto importante. Existen muchas bibliotecas de registro disponibles para las aplicaciones basadas en .NET, y entre las más avanzadas y ampliamente usadas se encuentra Log4net.

Log4net es una librería de registro de código abierto para ayudar al programador a exportar logs o registros en una variedad de objetivos de salida (errores, información, depuración, alertas, entre otros). (43) Esta librería fue utilizada para registrar los eventos y sucesos del sistema de forma persistente.

Log4net es parte del proyecto Apache Logging Services, el cual fue creado con la intención de proporcionar servicios de bitácora multilenguaje con el propósito de depuración y auditoría de aplicaciones. (44)

Esta librería ofrece soporte para múltiples plataformas, es configurada mediante un fichero XML y presenta una arquitectura de registro jerárquica. El registro jerárquico brinda flexibilidad al desarrollo basado en componentes. Cada componente tiene su log o registro. Las propiedades de cada uno estos logs pueden ser establecidas como el desarrollador lo necesite. (45)

3.2. Diseño

En la fase de diseño se modela el sistema y se define y construye la arquitectura. El diseño proporciona una imagen completa del software, desde una perspectiva de implementación. Define la relación entre los elementos estructurales más importantes del software, los estilos arquitectónicos y patrones de diseño que pueden usarse para satisfacer los requisitos definidos por el sistema, incluyendo los no funcionales. (46)

El objetivo de esta fase es convertir los requisitos del software en especificidades que describen cómo va a ser implementado el sistema. El modelo del diseño describe la realización de los casos de uso, donde intervienen los diagramas de clases de diseño y los diagramas de colaboración y/o secuencia.

En los diagramas de clases de diseño se modelan las clases, sus atributos y operaciones y las relaciones que existen entre las mismas. Para un mejor entendimiento de los requerimientos y de los procesos planteados para el desarrollo del sistema, se realizaron los diagramas de clases del diseño para cada caso de uso, que pueden ser encontrados en el [Anexo 3](#).

Los diagramas de secuencia permiten una mejor comprensión de las actividades que se llevan a cabo en los casos de uso, estos se muestran en el [Anexo 4](#).

3.2.1. Descripción de las clases

La descripción de las clases del sistema brinda información acerca de la interacción de las clases involucradas en la realización de los casos de uso. Las descripciones de estas clases se encuentran en el [Anexo 5](#).

En este capítulo se presentaron los diagramas de clases del diseño y diagramas de secuencia de los casos de uso más críticos, especificando los flujos de eventos e interacciones entre las clases y se explicaron

brevemente las clases empleadas. Se definió la arquitectura de la aplicación y se expusieron las características de la misma.

CAPÍTULO 4. IMPLEMENTACIÓN

El objetivo de este capítulo es describir los temas referentes a la implementación del sistema, quedarán expuestos los componentes así como la forma en que estos interactúan entre sí para brindar al cliente las funcionalidades del sistema. De igual forma puede ser vista la propuesta para el despliegue de la aplicación en una clínica radiológica.

4.1. Diagrama de componentes

Un diagrama de componentes modela los aspectos físicos de un sistema, muestra las organizaciones y dependencias lógicas entre los componentes de un software, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

Para un mejor entendimiento del diagrama de componentes, se realizó uno por cada componente del sistema, los cuales se muestran en las siguientes figuras.

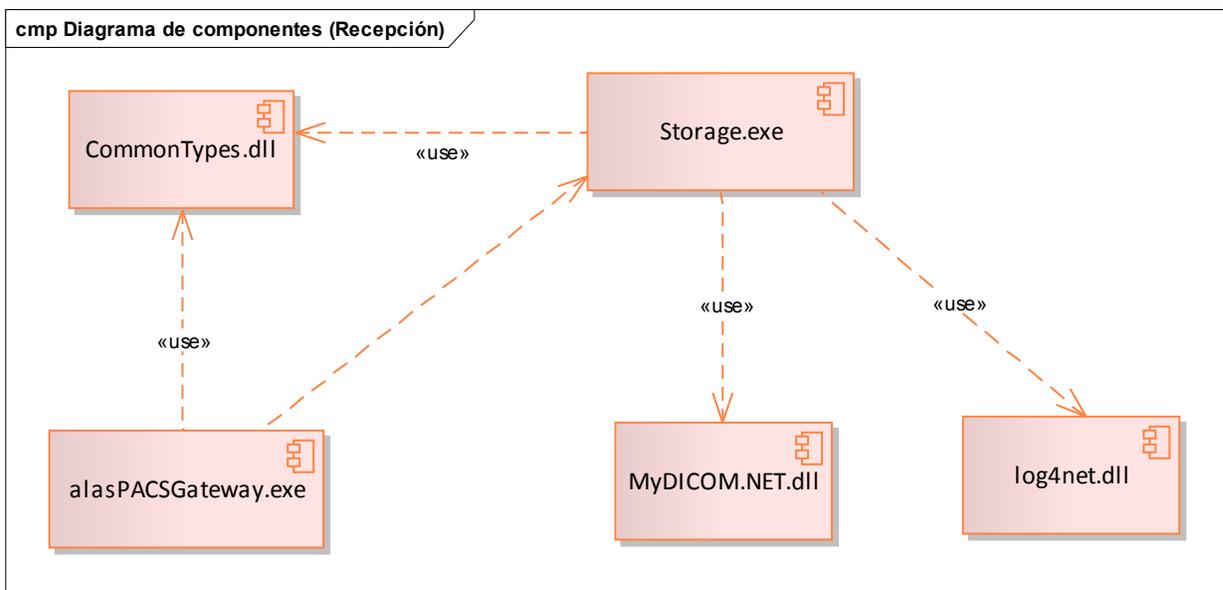


Fig. 4.1. Diagrama de componentes de **Recepción**

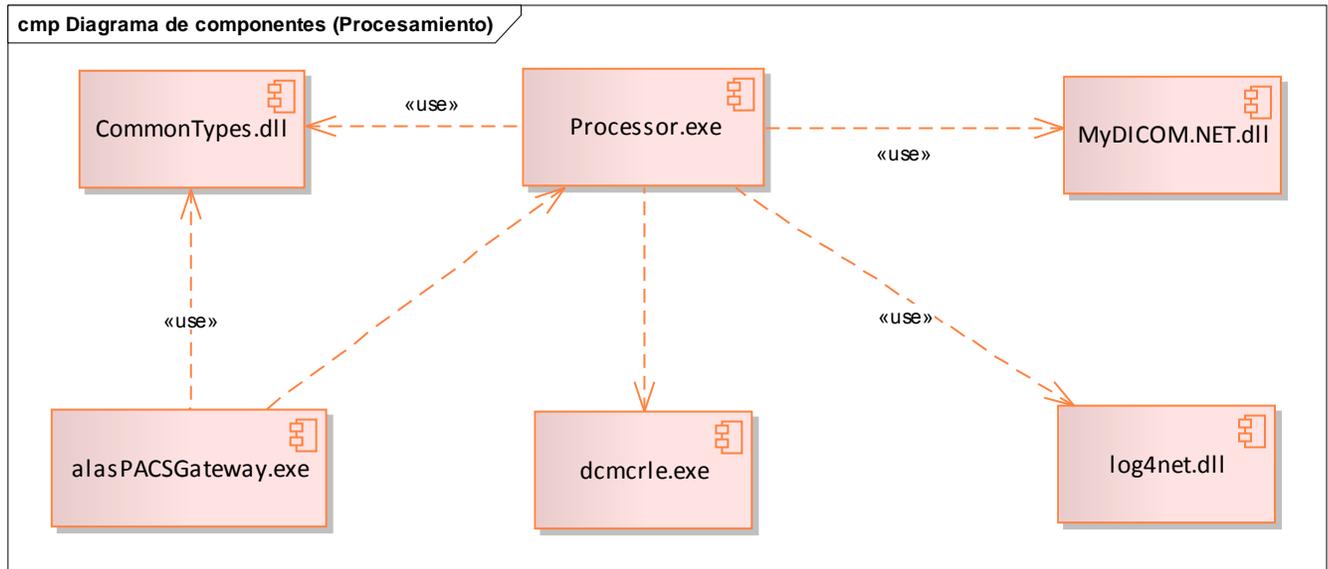


Fig. 4.2. Diagrama de componentes de **Procesamiento**

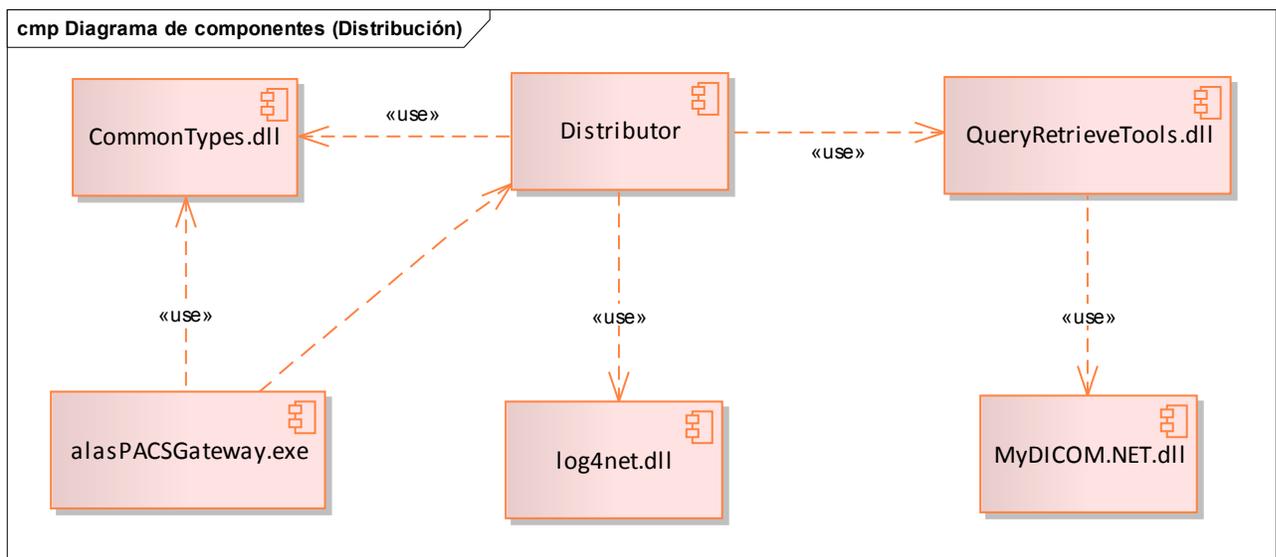


Fig. 4.3. Diagrama de componentes de **Distribución**

Los componentes principales Storage.exe, Processor.exe y Distributor.exe se encargan de la recepción y almacenamiento, procesamiento y distribución de las imágenes, respectivamente. Interactúan con el componente alasPACSGateway.exe que permite la configuración de las reglas y del sistema, y con la librería CommonTypes.dll, que provee las clases para realizar dicha configuración. Estos componentes principales utilizan log4net.dll para registrar los sucesos que ocurren en el sistema. El componente dcmcrle.exe es utilizado por Processor.exe para comprimir las imágenes.

Se emplearon las librerías QueryRetrieveTools.dll para enviar las imágenes a través de la red; y MyDICOM.NET.dll para realizar el almacenamiento, la lectura y escritura de las imágenes.

4.2. Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema, muestra las relaciones físicas entre los componentes de hardware y software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (procesos y objetos que se ejecutan en ellos).

En la **Fig. 4.2** se muestra el modelo de despliegue para el sistema.

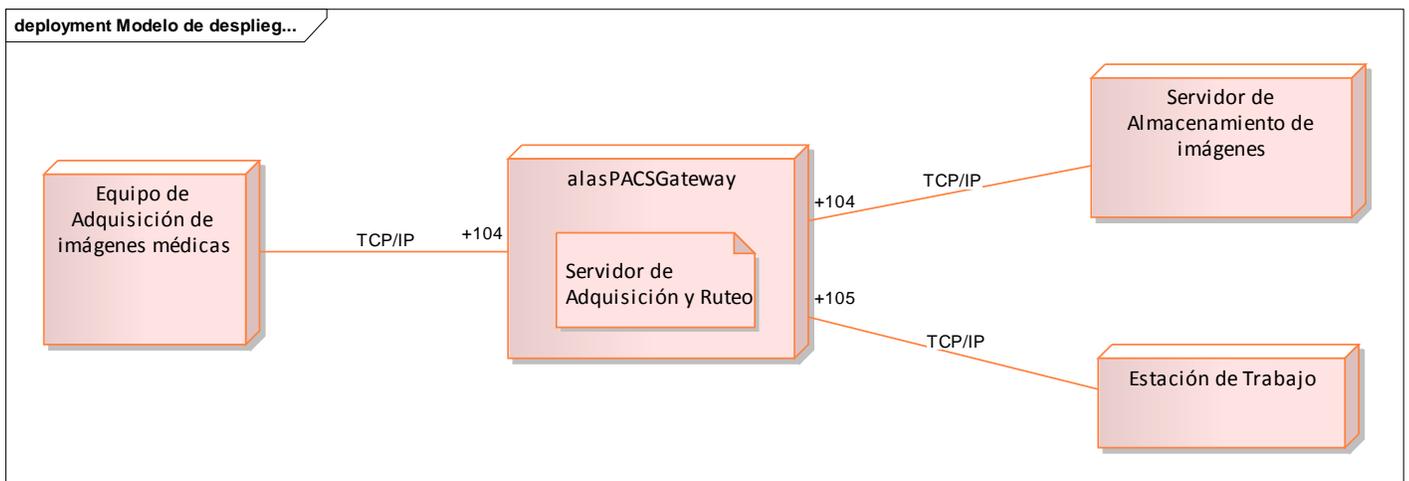


Fig. 4.2. Modelo de despliegue

4.3. Pruebas preliminares realizadas al sistema

Se realizaron pruebas preliminares al sistema desarrollado para comprobar su rendimiento y capacidad para la recepción, el procesamiento y la distribución de las imágenes. Se simuló un ambiente de un departamento radiológico, donde se tomaron varias computadoras como los equipos de adquisición y estaciones de destino y se configuraron varias reglas para la recepción, procesamiento y distribución de estudios imagenológicos de varias modalidades y tamaños. El sistema fue instalado en una computadora Pentium D, CPU de 3.00 GHz, 1 GB de RAM conectada a una red de 100 Mbps.

Se efectuaron pruebas con 3 estudios de angiografía, siendo esta modalidad la de mayor tamaño de entre los tipos que existen. Estos estudios, cada uno con una capacidad promedio entre 100 y 200 MB, de 60 a 120 imágenes, de 2 MB cada una, fueron distribuidos a 7 estaciones. El más grande, de 107 imágenes, con un tamaño de 214 MB fue recepcionado y procesado en 35 segundos y distribuido en 5:03 minutos, con un tiempo total entre la recepción y su llegada a todas las estaciones destino, de 5:38 minutos, con un tamaño final de 132 MB, representando el 61 % del total de su tamaño original.

En el caso de las tomografías, la modalidad de mayor cantidad de imágenes, se realizaron pruebas con 2 estudios, de entre 100 y 350 MB. Una de las pruebas realizadas arrojó un tiempo total, entre la recepción y distribución a las estaciones de trabajo, de 4:15 minutos. Igualmente, se efectuaron pruebas para las resonancias y ultrasonidos, estos tipos de estudios son los de menor tamaño. Ambos llegaron a su destino correspondiente en aproximadamente en 2:03 y 00:32 minutos, respectivamente.

Para las pruebas con los estudios de angiografía se tomaron una cantidad de 7 estaciones de destino y en los restantes casos, 5 estaciones. La lista de resultados y tiempos obtenidos durante las pruebas se muestra en el [Anexo 6](#).

A partir del análisis de las pruebas realizadas y a través de la siguiente expresión, donde **1440 minutos** representa la cantidad de minutos que tiene un día, ***T_m*** representa el tiempo de envío de un estudio imagenológico de una modalidad y ***C_m*** la capacidad del estudio, se obtuvo la capacidad de transmisión diaria de imágenes del sistema por modalidad.

$$\text{Capacidad de Transmisión Diaria} = \frac{1440 \text{ min}}{T_m \text{ (min)}} * C_m \text{ (MB)}$$

Para realizar el cálculo de la capacidad de transmisión diaria se tomaron estudios de diferentes modalidades que fueron utilizados en las pruebas. Para el caso de las angiografías, con un estudio de 214 MB que fue recepcionado, procesado y distribuido en un tiempo total de 05:38 minutos, se obtuvo 54 570 MB de transmisión diaria. La lista de las capacidades obtenidas se muestra en el [Anexo 7](#).

Se tomó como referencia el Centro de Alta Tecnología (CAT) de Venezuela, “Dr. Salvador Allende”, una clínica especializada en radiología donde se realizan estudios imagenológicos de tomografía, resonancia magnética y ultrasonido. Este centro es considerado el mayor de los que existen en Venezuela por la cantidad de estudios que realiza diariamente, que no exceden de 50 tomografías, 40 resonancias magnéticas y 30 ultrasonidos. Por tanto, con la suma de la cantidad de estudios de cada modalidad por el tamaño de los estudios utilizados para el cálculo de la capacidad, se obtuvo que la clínica tiene una capacidad de transmisión diaria de 6 927 MB, lo cual se muestra en el [Anexo 8](#).

A partir del cálculo de la capacidad de transmisión diaria para una clínica como la antes mencionada, se calculó el promedio de la capacidad de transmisión diaria del sistema para estas modalidades, obteniendo como resultado 22 387 MB, mostrado en el [Anexo 9](#), una capacidad por encima de la que se genera en la clínica tomada como referencia.

Con el estudio y las pruebas realizadas se demuestra que la velocidad de ruteo del sistema es suficiente para las condiciones existentes en un ambiente como el explicado anteriormente.

En el presente capítulo se presentaron los diagramas de componentes y despliegue, los cuales reflejan el resultado del desarrollo del software, los elementos que componen el sistema, la interacción entre ellos y la forma en que estos serán desplegados físicamente. Se expusieron los resultados obtenidos durante pruebas preliminares realizadas al sistema.

CONCLUSIONES

Se realizó un estudio de las tendencias actuales en la distribución de imágenes médicas donde se obtuvo que los sistemas internacionales no son económicamente viables y que el sistema que existe en Cuba no resuelve el problema planteado en la presente investigación.

Con el modelado de una propuesta de un flujo de trabajo se consiguió la normalización y distribución de las imágenes en un ambiente de un departamento radiológico.

El estilo arquitectónico empleado, Secuencial en Lote, permitió separar los procesos de recepción, procesamiento y distribución en varios componentes, brindando mayor estabilidad y escalabilidad al sistema.

Con la realización de este trabajo ha sido posible desarrollar un servidor de adquisición y ruteo de imágenes médicas; el sistema alas PACSGateway, versión 1.0, de fácil configuración y manipulación, capaz de obtener, almacenar y enviar imágenes médicas de acuerdo al estándar DICOM 3.0. Garantiza que las imágenes médicas adquiridas pasen por un proceso de normalización, permitiendo un menor costo en cuanto a almacenamiento y que sean distribuidas a través de la infraestructura de red a las estaciones clientes que las requieran, logrando un equilibrio en la carga del servidor y la red.

RECOMENDACIONES

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

Permitir que las tareas que se realizan tanto en el componente de procesamiento como en el componente de distribución se puedan ejecutar simultáneamente, o sea, que se procese más de una tarea a la vez en las colas de procesamiento y distribución. Con esto se logra un incremento de la Capacidad de Transmisión Diaria del sistema.

Definir la interoperabilidad entre el sistema y el alas RIS (Sistema de Información Radiológica) para agregar reglas de distribución, conociendo, a partir de la comunicación con el alas RIS, el médico que indica el estudio imagenológico, así como el especialista que lo debe informar. Para ello, el alas RIS debe incluir elementos en la configuración del perfil de los usuarios, que permitan especificar a los médicos si desean recibir las imágenes de aquellos estudios imagenológicos que indicaron a sus pacientes, así como la estación de trabajo de preferencia para aquellos especialistas que deben realizar los informes de los estudios.

REFERENCIAS BIBLIOGRÁFICAS

1. **González, Lázaro Rodríguez, Durañona, Yanoksy Yero.** *Servidor de Imágenes Médicas (Cassandra Server)*. Cuba : s.n., 2007.
2. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 9)*. 2010.
3. **Gamboa, André José Yero, Guerra, Alejandro Pérez.** *alasPACSWorklist Servidor de Listas de Trabajo*. Cuba : s.n., 2009.
4. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 1)*. 2010.
5. *Ídem a la referencia 4.*
6. *Ídem a la referencia 4.*
7. *Ídem a la referencia 3.*
8. *Ídem a la referencia 2.*
9. *Ídem a la referencia 2.*
10. *Ídem a la referencia 2.*
11. *Ídem a la referencia 3.*
12. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 6)*. 2010.
13. *Ídem a la referencia 12.*
14. *Ídem a la referencia 12.*
15. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 1ra Edición (Capítulo 8)*.
16. *Ídem a la referencia 15.*
17. **Matilde, Césari.** Sistemas basados en reglas. [En línea]
http://ai.frm.utn.edu.ar/micesari/files/SISTEMAS_BASADOS_REGLAS.pdf.

18. *Ídem a la referencia 17.*
19. **Escuela Politecnica Nacional de Ecuador.** *Escuela Politecnica Nacional de Ecuador.* [En línea] <http://dspace.epn.edu.ec/bitstream/123456789/1139/2/T11004%20CAP%204.pdf>.
20. *Ídem a la referencia 19.*
21. **Ramsoft.** [En línea] <http://www.ramsoft.com>.
22. *Ídem a la referencia 21.*
23. *Ídem a la referencia 21.*
24. **Ltd. IMAGE Information Systems.** *Administration Guide iQ-Router 1.0.* 2008.
25. *Ídem a la referencia 24.*
26. *Ídem a la referencia 24.*
27. *Ídem a la referencia 24.*
28. **Images On Call.** [En línea] <http://www.imagesoncall.com/dicom.asp>.
29. *Ídem a la referencia 28.*
30. *Ídem a la referencia 1.*
31. *Ídem a la referencia 3.*
32. **Alvarez, Liset María Barreras, Aguilera, Leydis Morales.** *Componente para grabado en media de estudios imagenológicos.* Cuba : s.n., 2010.
33. **Subversion.** [En línea] <http://subversion.apache.org/features.html>.
34. **TortoiseSVN.** [En línea] <http://tortoisesvn.tigris.org>.
35. **Rational.** [En línea] <http://www.rational.com.ar/herramientas/rup.html>.
36. *Ídem a la referencia 35.*
37. **González, Luis Eduardo Abreu, Fonseca, Mónica Guzmán.** *Módulo de gestión para los reportes estadísticos.* Cuba : s.n., 2010.
38. *Ídem a la referencia 32.*

39. *Ídem a la referencia 37.*
40. **Reynoso, Billy.** *Introducción a la Arquitectura de Software.* [Presentación Power Point] Buenos Aires : Universidad de Buenos Aires.
41. —. *Architect Academy: Seminario de Arquitectura de Software.* [Presentación de Power Point] Buenos Aires : Universidad de Buenos Aires.
42. **Escuela de Ingeniería Civil de Informática.** [En línea] 2010. [Citado el: 2 de marzo de 2011.] http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/Arquitectura_Software.pdf.
43. [En línea] [Citado el: 7 de marzo de 2011.] <http://paginawebhostingydominio.com/Log4net%20Creacion%20de%20Logs>.
44. [En línea] [Citado el: 7 de marzo de 2011.] <http://logging.apache.org/log4net/index.html>.
45. [En línea] [Citado el: 7 de marzo de 2011.] <http://logging.apache.org/log4net/release/features.html>.
46. **Pressman, Roger.** *Un Enfoque Práctico. (Capítulo 9).*
47. **Gutiérrez, Prof. José Manuel.** *Sistemas Expertos Basados en Reglas.* Universidad de Cantabria : s.n.
48. **Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato.** *Version Control with Subversion For Subversion 1.6.* 2010.
49. [En línea] <http://www.visus.com/products/modality-gateways.html>.
50. **Elaborado por CESIM - UCI. Todos los Derechos Reservados.** *ESTRATEGIA PARA LA SELECCIÓN DE CASOS DE USO, PROCESOS DEL NEGOCIO O DOMINIO.* Cuba : s.n., 2009.
51. **ACC/HIMSS/RSNA.** *Manual de Usuario IHE-Radiología.* España : s.n.
52. **RamSoft, Inc.** *Gateway Router Quick Start Guide.*
53. —. *Sistemas para el Almacenamiento y Transmisión de Imágenes Médicas.*
54. **Rivero, Arelys Castro, Hernández, Alejandro Noguera.** *Aplicación para el registro de estudios imagenológicos: Solución para equipos de Oftalmología.* Cuba : s.n., 2009.

BIBLIOGRAFÍA

1. **ACC/HIMSS/RSNA**. *Manual de Usuario IHE-Radiología*. España : s.n.
2. **Alvarez, Liset María Barreras, Aguilera, Leydis Morales**. *Componente para grabado en media de estudios imagenológicos*. Cuba : s.n., 2010.
3. *Architect Academy: Seminario de Arquitectura de Software*. [Presentación de Power Point] Buenos Aires : Universidad de Buenos Aires.
4. **Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato**. *Version Control with Subversion For Subversion 1.6*. 2010.
5. **Elaborado por CESIM - UCI. Todos los Derechos Reservados**. *ESTRATEGIA PARA LA SELECCIÓN DE CASOS DE USO, PROCESOS DEL NEGOCIO O DOMINIO*. Cuba : s.n., 2009.
6. En línea] [Citado el: 7 de marzo de 2011.] <http://logging.apache.org/log4net/index.html>.
7. En línea] [Citado el: 7 de marzo de 2011.] <http://logging.apache.org/log4net/release/features.html>.
8. En línea] [Citado el: 7 de marzo de 2011.] <http://paginawebhostingydominio.com/Log4net%20Creacion%20de%20Logs>.
9. En línea] <http://www.visus.com/products/modality-gateways.html>.
10. **Escuela de Ingeniería Civil de Informática**. [En línea] 2010. [Citado el: 2 de marzo de 2011.] http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/Arquitectura_Software.pdf.
11. **Escuela Politecnica Nacional de Ecuador**. *Escuela Politecnica Nacional de Ecuador*. [En línea] <http://dspace.epn.edu.ec/bitstream/123456789/1139/2/T11004%20CAP%204.pdf>.
12. **Gamboa, André José Yero, Guerra, Alejandro Pérez**. *alasPACSWorklist Servidor de Listas de Trabajo*. Cuba : s.n., 2009.
13. **González, Lázaro Rodríguez, Durañona, Yanoksy Yero**. *Servidor de Imágenes Médicas (Cassandra Server)*. Cuba : s.n., 2007.
14. **González, Luis Eduardo Abreu, Fonseca, Mónica Guzmán**. *Módulo de gestión para los reportes estadísticos*. Cuba : s.n., 2010.

15. **Gutiérrez, Prof. José Manuel.** *Sistemas Expertos Basados en Reglas.* Universidad de Cantabria : s.n.
16. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 9).* 2010.
17. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 1).* 2010.
18. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 2da Edición (Capítulo 6).* 2010.
19. **H. K. Huang, D.Sc., FRCR (Hon.), FAIMBE.** *Pacs and Imaging Informatics Basic Principles and Applications, 1ra Edición (Capítulo 8).*
20. *Images On Call.* [En línea] <http://www.imagesoncall.com/dicom.asp>.
21. **Ltd. IMAGE Information Systems.** *Administration Guide iQ-Router 1.0.* 2008.
22. **Matilde, Césari.** *Sistemas basados en reglas.* [En línea] http://ai.frm.utn.edu.ar/micesari/files/SISTEMAS_BASADOS_REGLAS.pdf.
23. **Pressman, Roger.** *Un Enfoque Práctico. (Capítulo 9).*
24. **RamSoft, Inc.** *Gateway Router Quick Start Guide.*
25. *Ramsoft.* [En línea] <http://www.ramsoft.com>.
26. *Rational.* [En línea] <http://www.rational.com.ar/herramientas/rup.html>.
27. **Reynoso, Billy.** *Introducción a la Arquitectura de Software.* [Presentación Power Point] Buenos Aires : Universidad de Buenos Aires.
28. **Rivero, Arelis Castro, Hernández, Alejandro Noguera.** *Aplicación para el registro de estudios imagenológicos: Solución para equipos de Oftalmología.* Cuba : s.n., 2009.
29. *Sistemas para el Almacenamiento y Transmisión de Imágenes Médicas.*
30. *Subversion.* [En línea] <http://subversion.apache.org/features.html>.
31. *TortoiseSVN.* [En línea] <http://tortoisesvn.tigris.org>.

GLOSARIO DE TÉRMINOS

Acquisition Gateway: Puerta de enlace de adquisición de imágenes.

ACR: Colegio Americano de Radiología.

AE Title: Application Entity Title, representación utilizada para identificar los nodos DICOM para comunicarse.

BPMN: Business Process Modeling Notation.

BSD: Berkeley Software Distribution. Licencia de software permisiva.

Compresión Wavelet: algoritmo de compresión con pérdida. Procedimiento de codificación que tiene como objetivo representar cierta cantidad de información utilizando una menor cantidad de la misma, siendo imposible una reconstrucción exacta de los datos originales. La compresión con pérdida sólo es útil cuando la reconstrucción exacta no es indispensable para que la información tenga sentido.

CT: Estudio imagenológico de tomografía.

Destino: Son los clientes a los cuales el sistema envía las imágenes después de procesadas, estos pueden ser el servidor de almacenamiento de imágenes del PACS o estaciones de trabajo

DICOM: Estándar para el intercambio, almacenamiento, visualización e impresión de imágenes médicas.

Fuente: Equipo de adquisición de imágenes médicas que puede enviar imágenes al sistema.

HIMSS: Healthcare Information and Management Systems Society.

HL7: Health Level Seven.

IHE: Integrating the Healthcare Enterprise.

JPEG Lossless: variantes del formato JPEG que comprimen la imagen sin pérdida de datos.

JPEG, JPEG 2000, JPEG Lossy: algoritmos de compresión con pérdida para reducir el tamaño de los archivos de imágenes. Al descomprimir o visualizar la imagen no se obtiene exactamente la misma imagen de la que se partía antes de la compresión.

MONO: Es el nombre de un proyecto de código abierto para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET. Mono posee importantes componentes útiles para desarrollar software, entre ellos, un compilador para el lenguaje C#.

MR: Estudio imagenológico de resonancia magnética.

NEMA: Asociación Nacional de Productores de Equipos Eléctricos.

PACS: Sistema para el Almacenamiento, Transmisión y Visualización de Imágenes Médicas.

Perfil de Integración IHE: Describen las necesidades clínicas de integración y la solución para llevarlas a cabo.

RIS: Sistema de Información Radiológica. Es un sistema encargado de la gestión de la información generada y manipulada como resultado de los procesos de negocio de carácter radiológico (imagenológico). Tiene la responsabilidad de manejar la información demográfica de los pacientes, programar las citas y la entrega de reportes de diagnóstico.

RLE: Run-Length Encoding. Algoritmo de compresión sin pérdida, permite la recuperación exacta de la imagen original del archivo comprimido.

RSNA: Sociedad de Radiología de Norteamérica.

SCP: Proveedor de clases de almacenamiento. Elemento de un Acquisition Gateway que tiene como propósito recibir la solicitud de almacenamiento desde los dispositivos de imágenes o el módulo PACS.

SCU: Proveedor de clases de usuario. Elemento de un Acquisition Gateway que tiene como propósito enviar (rutear) las imágenes almacenadas al servidor del PACS.

US: Estudio imagenológico de ultrasonido.

XA: Estudio imagenológico de angiografía.

ANEXOS

Anexo 1 Trazabilidad de los casos de uso con los requisitos

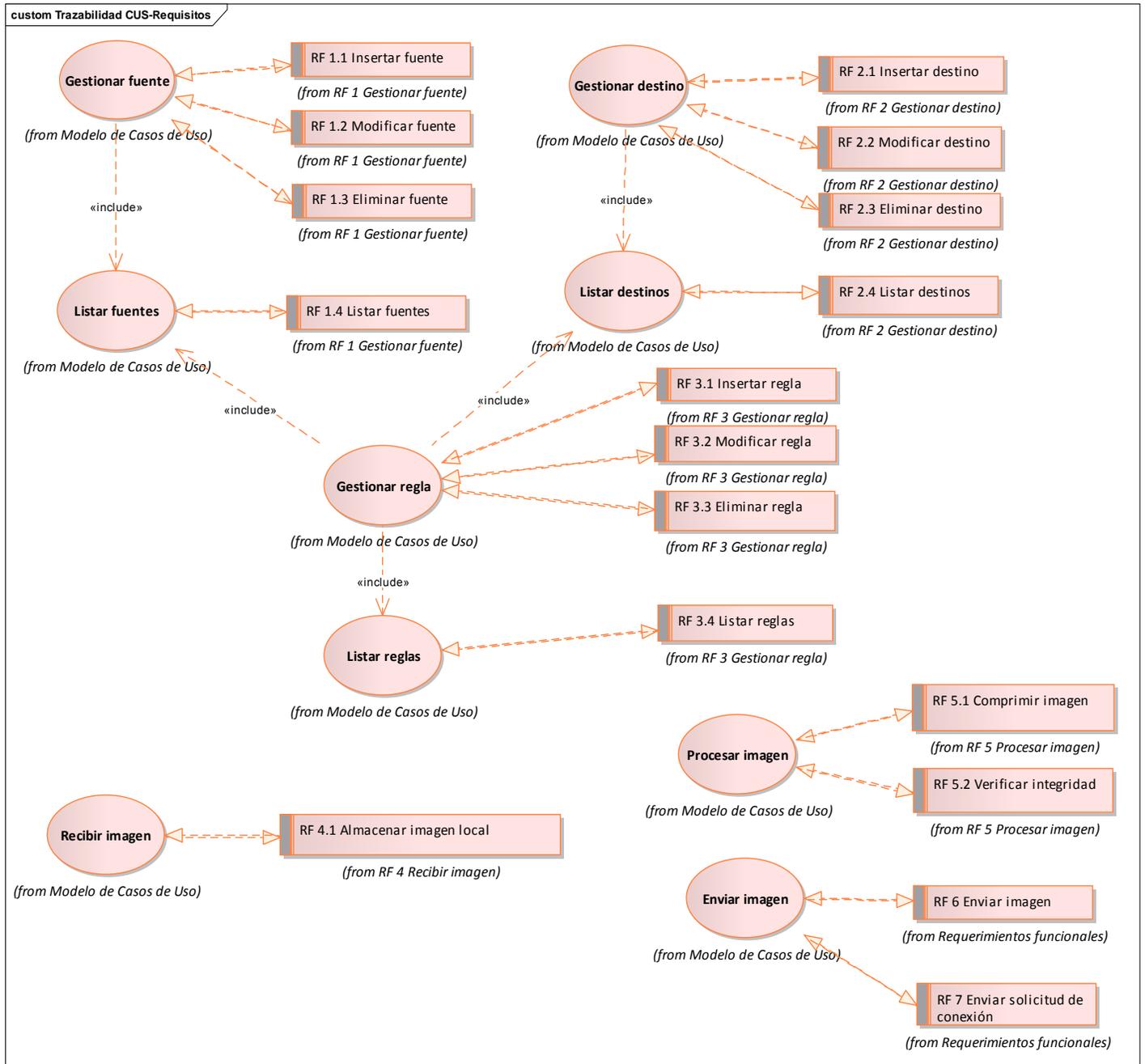


Fig. 1. Trazabilidad de casos de uso con requisitos

Anexo 2 Descripción ampliada de los Casos de Uso del Sistema

Caso de uso – Gestionar regla

CU # 3. Gestionar regla	
Objetivo	Insertar, modificar y eliminar regla.
Actores	Administrador del sistema (inicia).
Resumen	El administrador inserta los datos de una regla. Una regla especifica las tareas de procesamiento que se pueden realizar a una imagen y el destino hacia donde se debe enviar. También podrá modificar y eliminar una regla existente.
Complejidad	Baja
Prioridad	Crítico
Referencias	RF 3.1, RF 3.2, RF 3.3
Precondiciones	
Postcondiciones	Se realizó la operación seleccionada exitosamente.
Flujo de eventos	
Flujo básico < Insertar regla>	
<ol style="list-style-type: none"> 1. El Administrador selecciona la pestaña <i>Reglas</i>. 2. El Sistema muestra una ventana con las opciones para gestionar las reglas. 3. El Sistema muestra todas las fuentes (Ver CU Listar fuentes) y destinos (Ver CU Listar destinos). 4. El Administrador presiona la opción Nuevo. 5. El Administrador llena los campos de una regla (nombre, compresión, fuente, destino y campos a chequear de una imagen), luego presiona la opción Adicionar. 	

<ol style="list-style-type: none"> 6. El Sistema muestra todas las reglas (Ver CU Listar reglas) y un mensaje diciendo que la regla fue insertada correctamente. 7. Termina el escenario.
<p>Flujos alternos</p>
<p><i>4a. Alguno o todos los campos de la regla que se inserta están vacíos.</i></p>
<ol style="list-style-type: none"> 1. El Sistema muestra un mensaje informando el campo o los campos que deben ser llenados. 2. El Sistema regresa al paso 4 del Flujo básico.
<p><i>4b. El origen de la regla que se inserta ya forma parte de otra regla existente en el sistema.</i></p>
<ol style="list-style-type: none"> 1. El Sistema muestra un mensaje informando que cada regla debe tener un origen único. 2. El Sistema regresa al paso 4 del Flujo básico.
<p><i>4c. La regla que se inserta ya existe en el sistema.</i></p>
<ol style="list-style-type: none"> 1. El Sistema muestra un mensaje informando que el nombre de la regla ya existe y este debe ser único. 2. El Sistema regresa al paso 4 del Flujo básico.
<p>Flujo básico < Modificar regla ></p>
<ol style="list-style-type: none"> 1. El Administrador selecciona la pestaña <i>Gestión de Reglas</i>. 2. El Sistema muestra una ventana con las opciones para gestionar las reglas. 3. El Sistema muestra todas las fuentes (Ver CU Listar fuentes) y destinos (Ver CU Listar destinos). 4. El Sistema muestra todas las reglas (Ver CU Listar reglas). 5. El Administrador selecciona la regla con doble clic. 6. El Sistema muestra los datos de la regla seleccionada. 7. El Administrador llena la información con los nuevos datos y presiona la opción <i>Modificar</i> o selecciona la regla, presiona clic derecho y la opción <i>Modificar</i>. 8. El Sistema modifica la regla. 9. El Sistema muestra un mensaje informando que la regla ha sido modificada correctamente.

10. Termina el escenario.		
Flujos alternos		
<i>7a. El origen de la regla que se inserta ya forma parte de otra regla existente en el sistema.</i>		
<ol style="list-style-type: none"> 1. El Sistema muestra un mensaje informando que cada regla debe tener un origen único. 2. El Sistema regresa al paso 7 del Flujo básico. 		
<i>7b. La regla que se inserta ya existe en el sistema.</i>		
<ol style="list-style-type: none"> 1. El Sistema muestra un mensaje informando que el nombre de la regla ya existe y este debe ser único. 2. El Sistema regresa al paso 7 del Flujo básico. 		
Flujo básico < Eliminar regla >		
<ol style="list-style-type: none"> 1. El Administrador selecciona la pestaña <i>Reglas</i>. 2. El Sistema muestra una ventana con las opciones para gestionar las reglas. 3. El Sistema muestra todas las reglas (Ver CU Listar reglas). 4. El Administrador selecciona la regla, presiona clic derecho y la opción <i>Eliminar</i>. 5. El Sistema elimina la regla. 6. El Sistema muestra un mensaje informando que la regla ha sido eliminada correctamente. 7. Termina el escenario. 		
Relaciones	CU incluidos	CU Listar fuentes CU Listar destinos CU Listar reglas
	CU extendidos	No aplicable
Requisitos funcionales	no	RNIU 1, RNIU 2

Asuntos pendientes	No Aplicable
---------------------------	--------------

Caso de uso – Recibir imagen

CU # 7. Recibir imagen	
Objetivo	El sistema debe almacenar las imágenes que son enviadas por un equipo de adquisición.
Actores	Equipo de Adquisición (inicia).
Resumen	Un equipo de adquisición envía imágenes al sistema, el cual debe almacenarlas localmente.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF 4.1
Precondiciones	
Postcondiciones	Se realizó la operación seleccionada exitosamente.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El Sistema recibe una solicitud de almacenamiento de un equipo de adquisición (fuente). 2. El Sistema verifica si la fuente se encuentra registrada en el sistema. 3. El Sistema almacena las imágenes localmente. 4. El Sistema registra el suceso. 5. Termina el escenario. 	

<i>2a. La fuente no existe en el sistema.</i>		
1. El Sistema no realiza el almacenamiento de las imágenes.		
Relaciones	CU incluidos	No aplicable
	CU extendidos	No aplicable
Requisitos funcionales	no	RNS 1, RNS 2, RNDI 6, RNF 1, RNI 1, RNE 1
Asuntos pendientes	No Aplicable	

Caso de uso – Procesar imagen

CU # 8. Procesar imagen	
Objetivo	El sistema debe procesar todas las imágenes que están almacenadas.
Actores	Reloj (inicia).
Resumen	El sistema debe realizar las tareas de procesamiento que están configuradas a las imágenes almacenadas. Se registra el evento sucedido.
Complejidad	Alta
Prioridad	Crítico
Referencias	RF 5.1, RF 5.1
Precondiciones	
Postcondiciones	Se realizó la operación seleccionada exitosamente.

Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El Sistema revisa las tareas de procesamiento configuradas en las reglas para la imagen para comenzar el procesamiento. 2. El Sistema modifica los valores de los campos de la imagen configurados en la regla. 3. El Sistema realiza una compresión a la imagen, de acuerdo al tipo de compresión configurado en la regla. 4. El Sistema registra el suceso. 5. Termina el escenario. 	
<i>2a. No se especifican campos a modificar en la regla asociada a la imagen.</i>	
<ol style="list-style-type: none"> 1. No se modifican los campos de la imagen. 	
<i>3a. No se especifica compresión en la regla asociada a la imagen.</i>	
<ol style="list-style-type: none"> 1. No se comprime la imagen. 	
Relaciones	CU incluidos No aplicable
	CU extendidos No aplicable
Requisitos funcionales no	RNF 1, RNE 1, RNS 2
Asuntos pendientes	No Aplicable

Caso de uso – Enviar imagen

CU # 9. Enviar imagen	
Objetivo	El sistema debe enviar todas las imágenes procesadas hacia sus destinos,

	configurados en las reglas.	
Actores	Reloj (inicia).	
Resumen	El sistema envía las imágenes procesadas hacia su destino y luego las elimina. Se registra el evento sucedido.	
Complejidad	Alta	
Prioridad	Crítico	
Referencias	RF 6	
Precondiciones		
Postcondiciones	Se realizó la operación seleccionada exitosamente.	
Flujo de eventos		
Flujo básico		
<ol style="list-style-type: none"> 1. El Sistema revisa los destinos configurados en las reglas para la imagen para comenzar el envío. 2. El Sistema envía una solicitud de conexión al o los destinos. 3. El Sistema realiza una solicitud de almacenamiento al o los destinos. 4. El Sistema envía la imagen. 5. El Sistema elimina la imagen. 6. El Sistema registra el suceso. 7. Termina el escenario. 		
<i>2a. Si el sistema no puede establecer una conexión con algún destino.</i>		
<ol style="list-style-type: none"> 1. La imagen se envía posteriormente. 		
Relaciones	CU incluidos	CU Enviar solicitud de conexión

	CU extendidos	No aplicable
Requisitos funcionales	no	RNS 2, RNDI 7, RNF 1, RNI 1, RNE 1
Asuntos pendientes		No Aplicable

Anexo 3 Diagramas de clases del diseño

Caso de uso – Gestionar regla

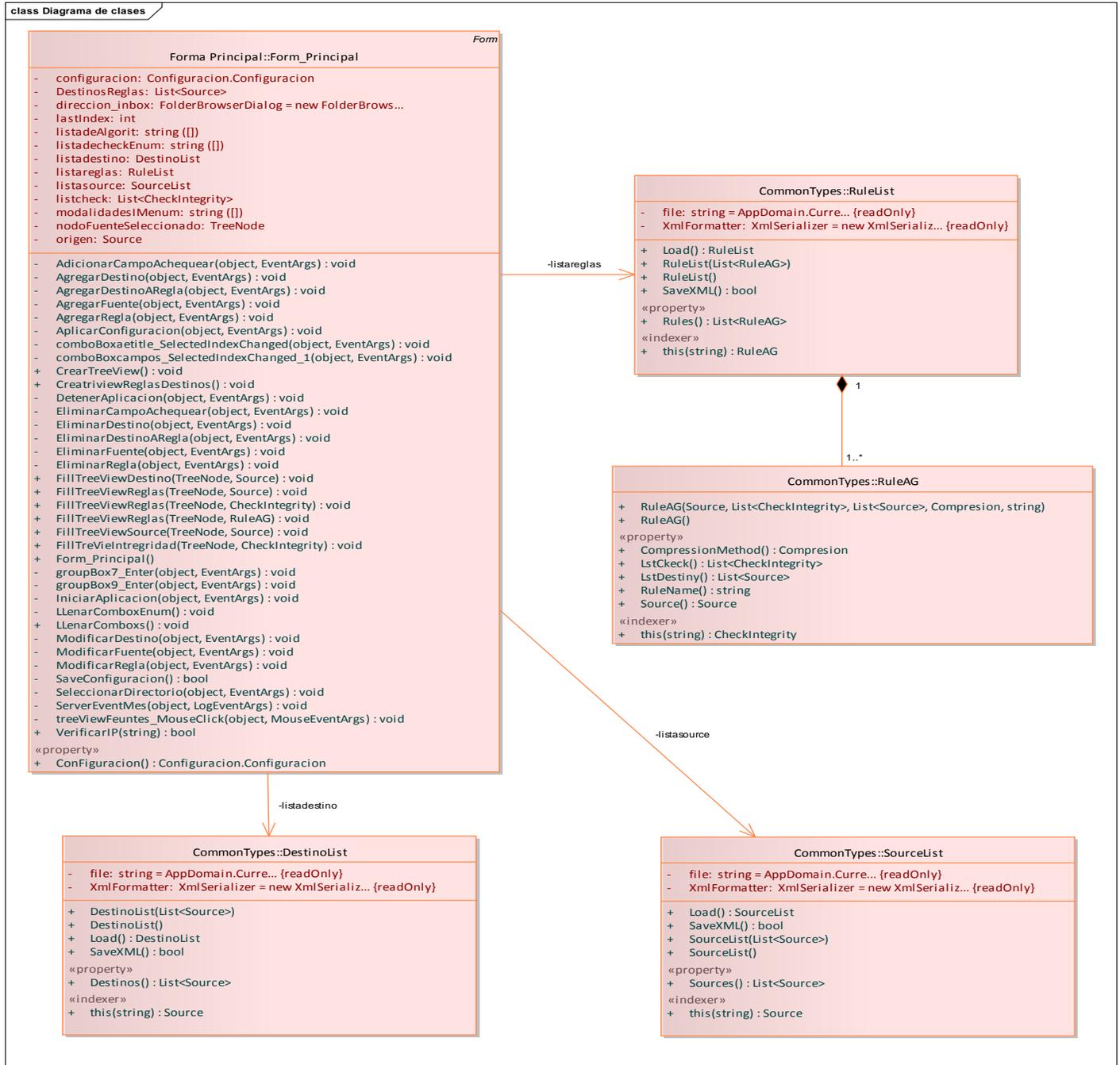


Fig. 2. Diagrama de clases de CU Gestionar regla

Caso de uso – Recibir imagen

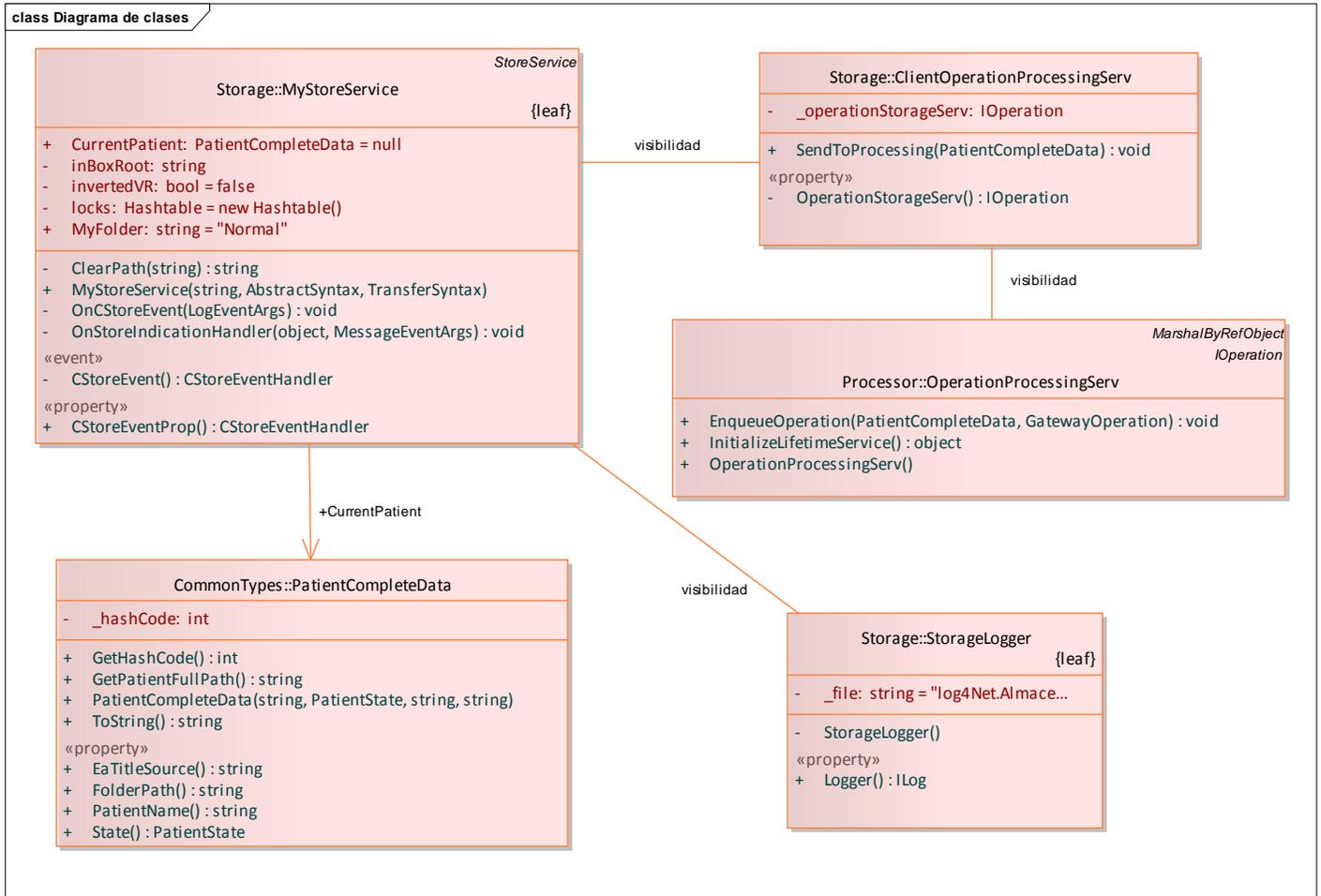


Fig. 3. Diagrama de clases de CU Recibir imagen

Caso de uso – Procesar imagen

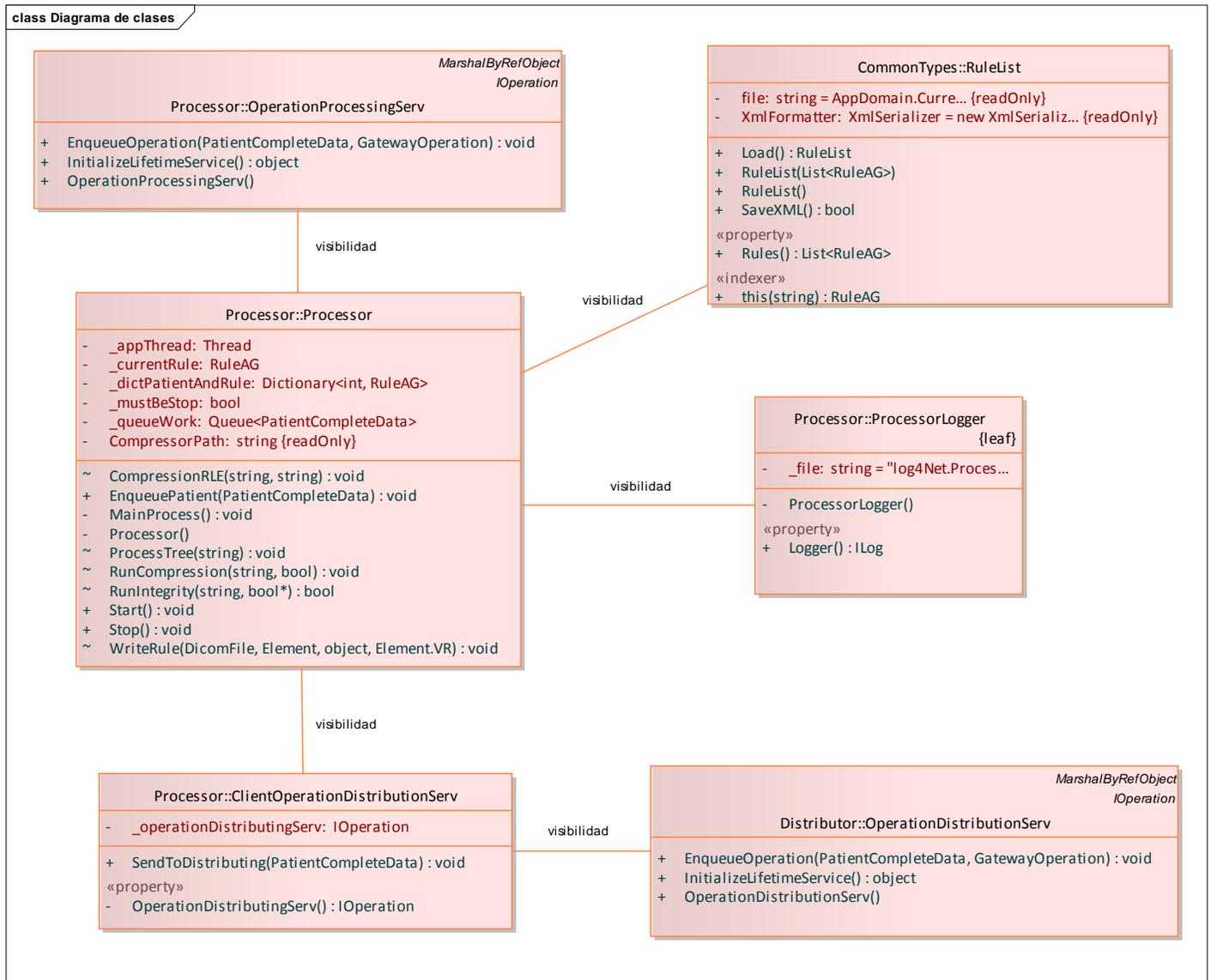


Fig. 4. Diagrama de clases de CU Procesar imagen

Caso de uso – Enviar imagen

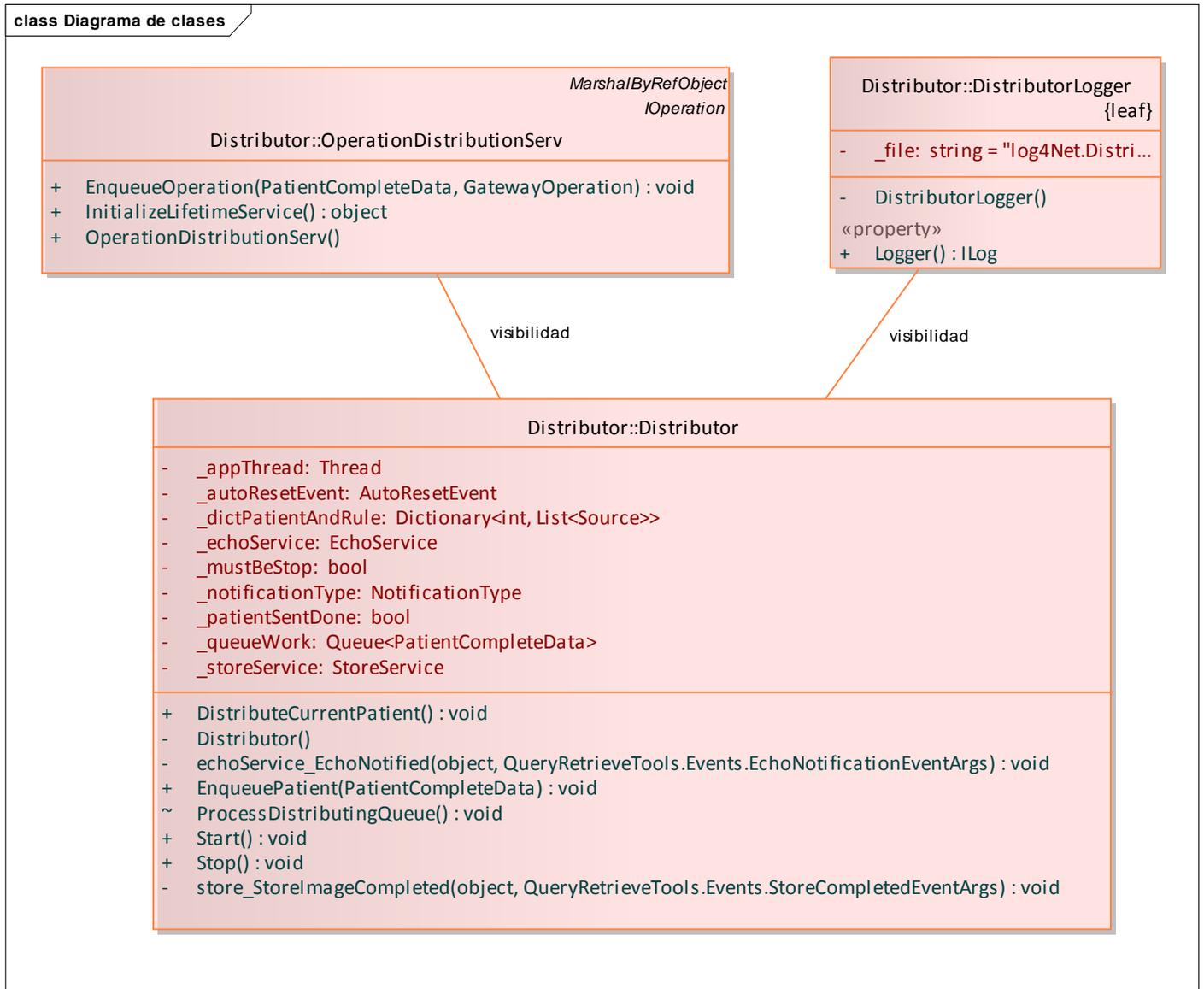


Fig. 5. Diagrama de clases de CU Enviar imagen

Anexo 4 Diagramas de secuencia

Caso de uso – Recibir imagen

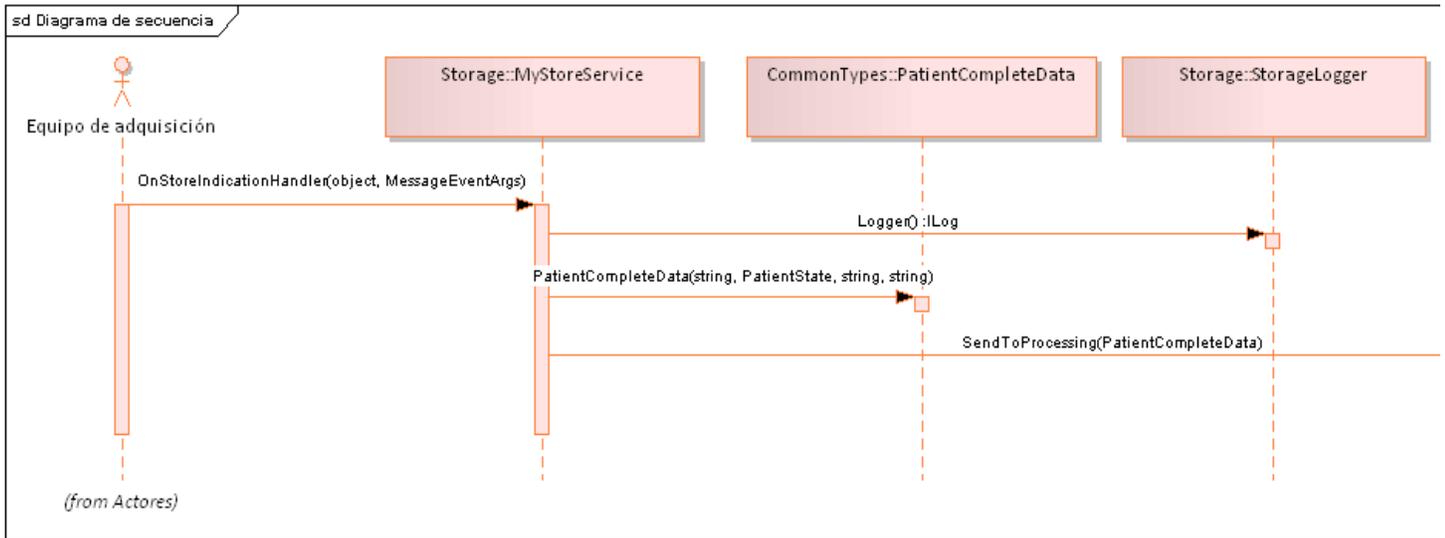


Fig. 6.1. Diagrama de secuencia de CU Recibir imagen

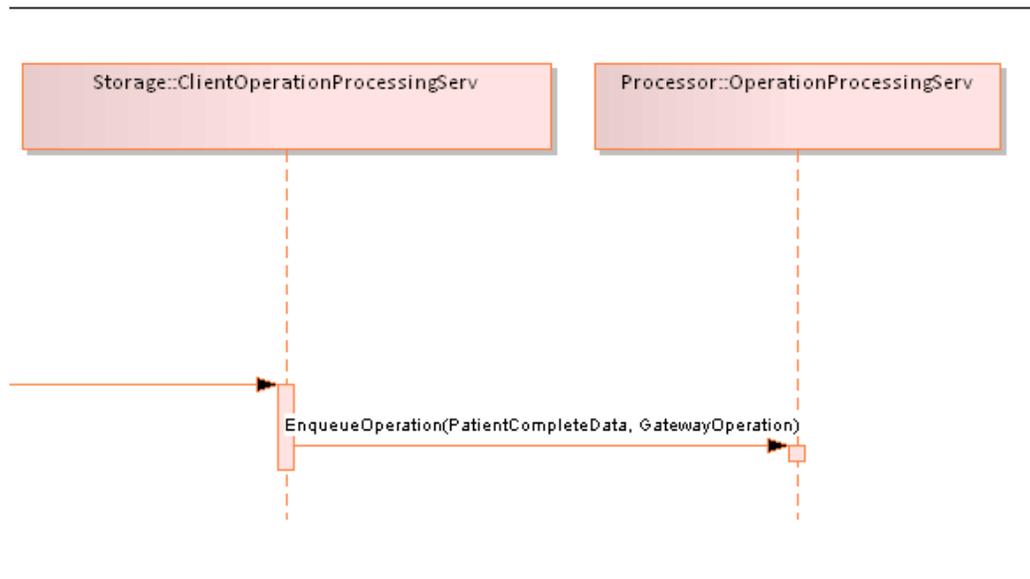


Fig. 6.2. Diagrama de secuencia de CU Recibir imagen

Caso de uso – Procesar imagen

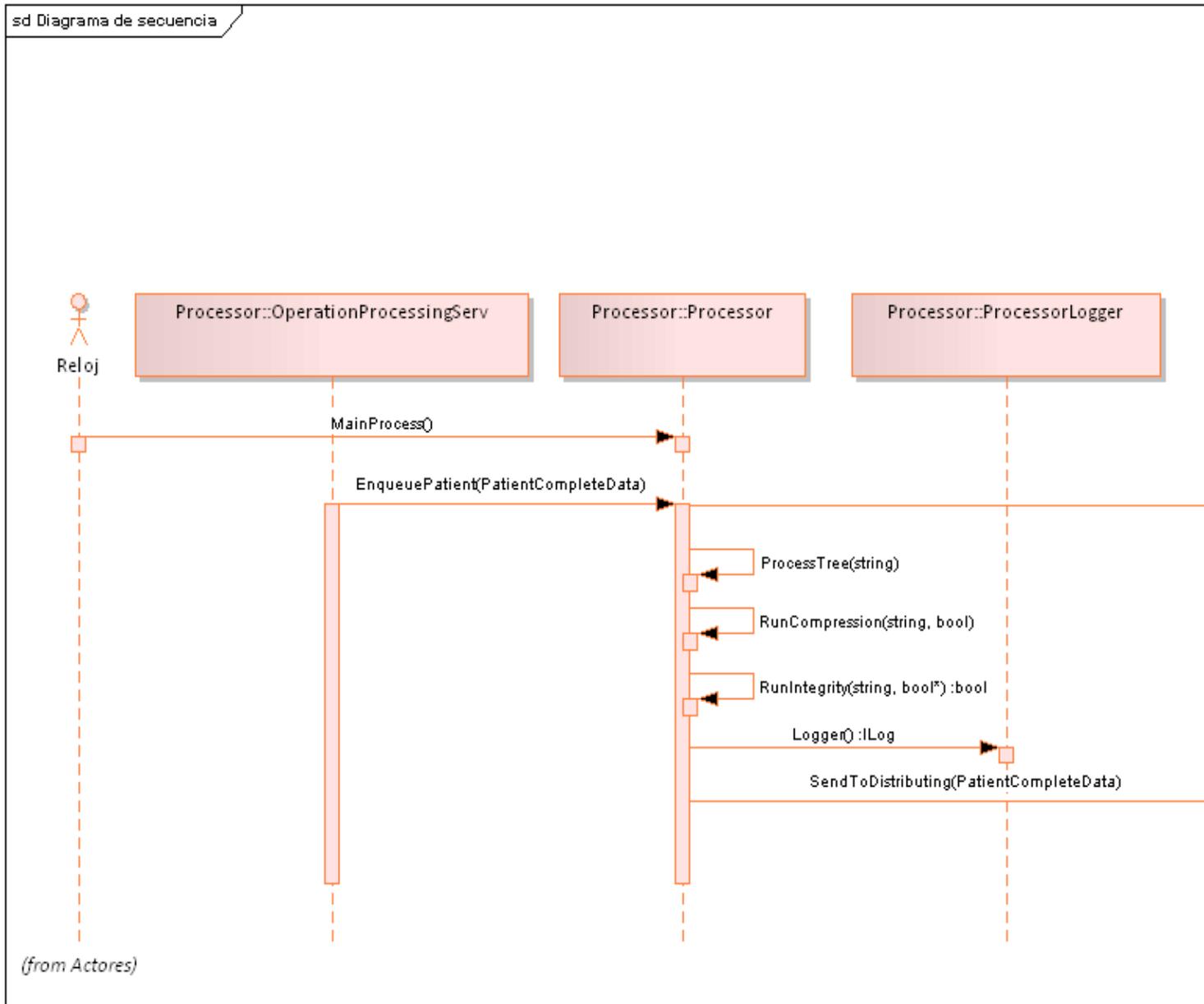


Fig. 7.1. Diagrama de secuencia de CU Procesar imagen

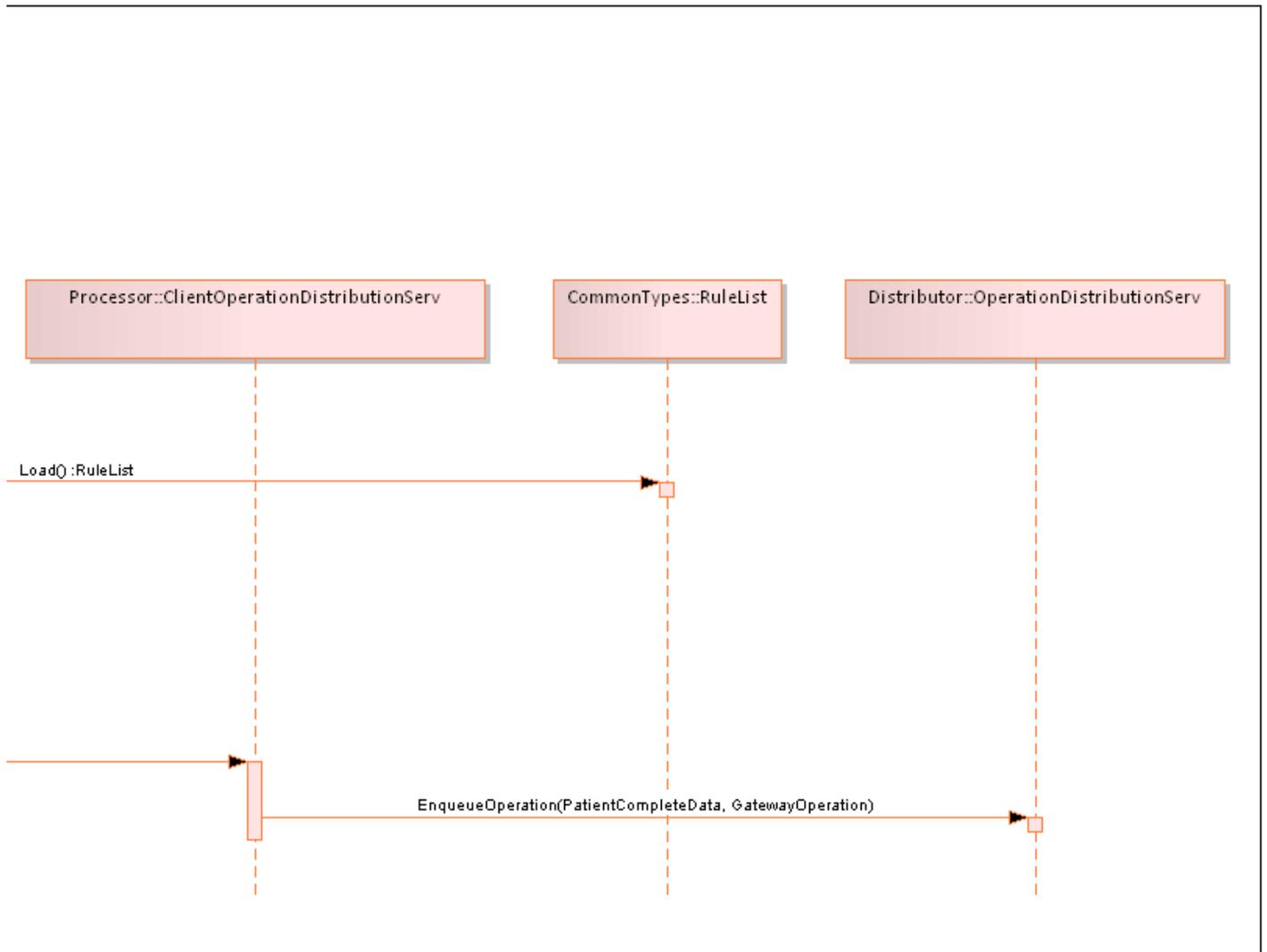


Fig. 7.2. Diagrama de secuencia de CU Procesar imagen

Caso de uso – Enviar imagen

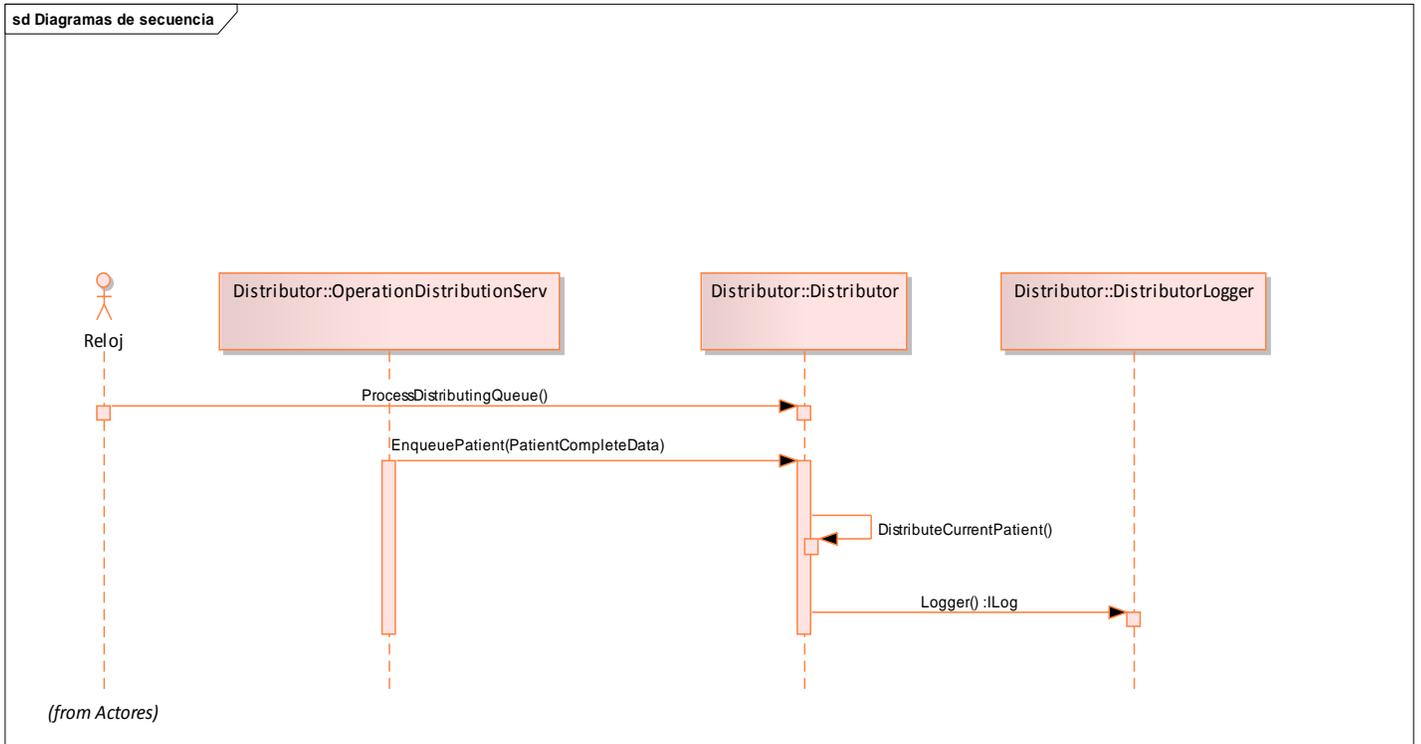


Fig. 8. Diagrama de secuencia de CU Enviar imagen

Anexo 5 Descripción de las clases

Nombre: DestinoList	
Tipo de clase: controladora	
Atributo	Tipo
file	string
XmlFormatter	XmlSerializer
Para cada responsabilidad:	
Nombre:	DestinoList (reglas: List<Source>)
Descripción:	Constructor de la clase.
Nombre:	DestinoList
Descripción:	Constructor de la clase.
Nombre:	Load
Descripción:	Carga de un fichero XML los destinos que tiene configurado el sistema.
Nombre:	SaveXML
Descripción:	Salva en un fichero XML los destinos que tiene configurado el sistema.
Nombre:	This (name: string)
Descripción:	Buscar un destino dado en la lista de destinos del sistema.

Nombre: Source	
Tipo de clase: entidad	
Atributo	Tipo
Modalidad	Modalidades

Alias	string
AETitle	String
Ip	string
Puerto	int
Para cada responsabilidad:	
Nombre:	Source (modalidad: Modalidades, alias: string, AETitle: string, ip: string, puerto: int)
Descripción:	Constructor de la clase.
Nombre:	Source (alias: string, aeATitle: string, ip: string, puerto: int)
Descripción:	Constructor de la clase.
Nombre:	Source
Descripción:	Constructor de la clase.
ToString	ToString
Descripción:	Convierte el objeto de la clase.

Nombre: Form_Principal	
Tipo de clase: interfaz	
Atributo	Tipo
listasource	SourceList
listadestino	DestinoList
listareglas	RuleList
DestinosReglas	List<Source>
listcheck	List<CheckIntegrity>

origen	Source
nodoFuenteSeleccionado	TreeNode
modalidadesIMenum	string
listadecheckEnum	string
listadeAlgorit	string
configuracion	Configuracion.Configuracion
lastIndex	int
direccion_inbox	FolderBrowserDialog
Para cada responsabilidad:	
Nombre:	AgregarFuente (sender: object, e: EventArgs)
Descripción:	Crea una fuente con los datos introducidos por el usuario y la agrega a la lista de fuentes. Invoca al método SaveXML de la clase SourceList.
Nombre:	AgregarDestino (sender: object, e: EventArgs)
Descripción:	Crea un destino con los datos introducidos por el usuario y la agrega a la lista de destinos. Invoca al método SaveXML de la clase DestinoList.
Nombre:	AgregarRegla (sender: object, e: EventArgs)
Descripción:	Crea una regla con los datos introducidos por el usuario y la agrega a la lista de reglas. Invoca al método SaveXML de la clase RuleList.
Nombre:	VerificarIP (ip: string)
Descripción:	Se verifica la entrada de una dirección IP válida.
Nombre:	ModificarFuente (sender: object, e: EventArgs)
Descripción:	Actualiza los datos una fuente seleccionada con los nuevos datos introducidos por el usuario. Invoca al método SaveXML de la clase SourceList.

Nombre:	ModificarDestino (sender: object, e: EventArgs)
Descripción:	Actualiza los datos un destino seleccionado con los nuevos datos introducidos por el usuario. Invoca al método SaveXML de la clase DestinoList.
Nombre:	ModificarRegla (sender: object, e: EventArgs)
Descripción:	Actualiza los datos una regla seleccionada con los nuevos datos introducidos por el usuario. Invoca al método SaveXML de la clase RuleList.
Nombre:	EliminarFuente
Descripción:	Elimina una fuente seleccionada por el usuario. Invoca al método SaveXML de la clase SourceList.
Nombre:	EliminarDestino
Descripción:	Elimina uno destino seleccionado por el usuario. Invoca al método SaveXML de la clase DestinoList.
Nombre:	EliminarRegla
Descripción:	Elimina una regla seleccionada por el usuario. Invoca al método SaveXML de la clase RuleList.
Nombre:	FillTreeViewSource (t: TreeNode, s: Source)
Descripción:	Mostrar las fuentes existentes en el sistema en la interfaz de usuario.
Nombre:	FillTreeViewDestino (t: TreeNode, s: Source)
Descripción:	Mostrar los destinos existentes en el sistema en la interfaz de usuario.
Nombre:	FillTreeViewReglas (t: TreeNode, s: RuleAG)
Descripción:	Mostrar las reglas existentes en el sistema en la interfaz de usuario.

Nombre: SourceList

Tipo de clase: controladora	
Atributo	Tipo
file	string
XmlFormatter	XmlSerializer
Para cada responsabilidad:	
Nombre:	SourceList (reglas: List<Source>)
Descripción:	Constructor de la clase.
Nombre:	SourceList
Descripción:	Constructor de la clase.
Nombre:	Load
Descripción:	Carga de un fichero XML las fuentes que tiene configurado el sistema.
Nombre:	SaveXML
Descripción:	Salva en un fichero XML las fuentes que tiene configurado el sistema.
Nombre:	This (name: string)
Descripción:	Buscar una fuente dada en la lista de fuentes del sistema.

Nombre: RuleList	
Tipo de clase: controladora	
Atributo	Tipo
file	string
XmlFormatter	XmlSerializer
Para cada responsabilidad:	

Nombre:	RuleList (reglas: List<RuleAG>)
Descripción:	Constructor de la clase.
Nombre:	RuleList
Descripción:	Constructor de la clase.
Nombre:	Load
Descripción:	Carga de un fichero XML las reglas que tiene configurado el sistema.
Nombre:	SaveXML
Descripción:	Salva en un fichero XML las reglas que tiene configurado el sistema.
Nombre:	This (name: string)
Descripción:	Buscar una regla dada en la lista de reglas del sistema.

Nombre: RuleAG	
Tipo de clase: entidad	
Atributo	Tipo
Source	Source
LstCkeck	List<CheckIntegrity>
LstDestiny	List<Source>
CompressionMethod	Compresion
RuleName	string
Para cada responsabilidad:	
Nombre:	RuleAG (source: Source, check: List<CheckIntegrity>, destiny: List<Source>, compress: Compresion, ruleName: string)

Descripción:	Constructor de la clase.
Nombre:	RuleAG
Descripción:	Constructor de la clase.

Nombre: MyStoreService	
Tipo de clase: controladora	
Atributo	Tipo
CurrentPatient	PatientCompleteData
inBoxRoot	string
invertedVR	bool
locks	Hashtable
MyFolder	string
Para cada responsabilidad:	
Nombre:	MyStoreService
Descripción:	Constructor de la clase.
Nombre:	OnCStoreEvent (e: LogEventArgs)
Descripción:	Devuelve una instancia del evento CStoreEvent.
Nombre:	OnStoreIndicationHandler (service: object, args: MessageEventArgs)
Descripción:	Crea la estructura de directorios donde se almacenarán las imágenes de un paciente y luego almacena estas localmente.
Nombre:	ClearPath (path: string)
Descripción:	Valida el nombre de los directorios donde se almacenan las imágenes.

Nombre: PatientCompleteData	
Tipo de clase: entidad	
Atributo	Tipo
_hashCode	int
Para cada responsabilidad:	
Nombre:	PatientCompleteData (folderPath: string, state: PatientState, ea: string, patient: string)
Descripción:	Constructor de la clase.
Nombre:	GetPatientFullPath
Descripción:	Devuelve la dirección donde está almacenado un paciente.
Nombre:	GetHashCode
Descripción:	Devuelve un código hash como identificador de ese paciente.

Nombre: ClientOperationProcessingServ	
Tipo de clase: controladora	
Atributo	Tipo
_operationStorageServ	IOperation
Para cada responsabilidad:	
Nombre:	SendToProcessing (patient: PatientCompleteData)
Descripción:	Invoca al método EnqueueOperation de la clase OperationProcessingServ para adicionar el paciente a la cola de trabajo de la clase Processor.

Nombre: StorageLogger

Tipo de clase: controladora	
Atributo	Tipo
_file	string
Para cada responsabilidad:	
Nombre:	StorageLogger
Descripción:	Constructor de la clase, configura la estructura, dirección y capacidad del fichero log.
Nombre:	Logger
Descripción:	Crea el tipo de suceso y su descripción.

Nombre: Processor	
Tipo de clase: controladora	
Atributo	Tipo
CompressorPath	string
_queueWork	Queue<PatientCompleteData>
_appThread	Thread
_dictPatientAndRule	Dictionary<int, RuleAG>
_currentRule	RuleAG
_mustBeStop	bool
Para cada responsabilidad:	
Nombre:	Processor
Descripción:	Constructor de la clase.

Start	Start
Descripción:	Inicia un hilo de ejecución que verifica si hay elementos en la cola de trabajo para comenzar el procesamiento.
Nombre:	Stop
Descripción:	Detiene el hilo de ejecución.
Nombre:	MainProcess
Descripción:	Extrae el primer elemento de la cola de trabajo para comenzar el procesamiento y busca la regla correspondientes a dicho paciente. Invoca al método ProcessTree.
Nombre:	EnqueuePatient (patientCompleteData: PatientCompleteData)
Descripción:	Adicionar un paciente a la cola de trabajo.
Nombre:	ProcessTree (patientFolder: string)
Descripción:	Manda a procesar la estructura de directorios donde están almacenadas las imágenes de un paciente.
Nombre:	RunIntegrity (filepath: string, [out] mustBeCompressed: bool)
Descripción:	Verifica los campos de una imagen, estos campos están definidos en la regla para esa imagen.
Nombre:	WriteRule (file: DicomFile, element: Element, value: object, vr: Element.VR)
Descripción:	Escribe en los campos de una imagen los datos que fueron definidos en la regla para esa imagen.
Nombre:	RunCompression (path: string, mustBeCompressed: bool)
Descripción:	Extrae de la regla el tipo de compresión que se le tiene que realizar a la imagen e invoca al método que realiza dicho tipo de compresión.
Nombre:	CompressionRLE (inputFile: string, outputFile: string)
Descripción:	Aplica compresión RLE a la imagen.

Nombre: OperationProcessingServ	
Tipo de clase: controladora	
Atributo	Tipo
_hashCode	int
Para cada responsabilidad:	
Nombre:	OperationProcessingServ
Descripción:	Constructor de la clase.
Nombre:	EnqueueOperation (patient: PatientCompleteData, operation: GatewayOperation)
Descripción:	Invoca al método EnqueuePatient de la clase Processor para adicionar el paciente a la cola de trabajo de la clase Processor.

Nombre: ClientOperationDistributionServ	
Tipo de clase: controladora	
Atributo	Tipo
_operationDistributingServ	IOperation
Para cada responsabilidad:	
Nombre:	SendToDistributing (patient: PatientCompleteData)
Descripción:	Invoca al método EnqueueOperation de la clase OperationDistributingServ para adicionar el paciente a la cola de trabajo de la clase Distributor.

Nombre: ProcessorLogger	
Tipo de clase: controladora	

Atributo		Tipo
_file		string
Para cada responsabilidad:		
Nombre:	Logger	
Descripción:	Crea el tipo de suceso y su descripción.	
Nombre:	ProcessorLogger	
Descripción:	Constructor de la clase, configura la estructura, dirección y capacidad del fichero log.	

Nombre: Distributor		
Tipo de clase: controladora		
Atributo		Tipo
_queueWork		Queue<PatientCompleteData>
_appThread		Thread
_dictPatientAndRule		Dictionary<int, List<Source>>
_mustBeStop		bool
_notificationType		NotificationType
_patientSentDone		bool
_echoService		EchoService
_storeService		StoreService
_autoResetEvent		AutoResetEvent
Para cada responsabilidad:		

Nombre:	DistributeCurrentPatient
Descripción:	Extrae el primer elemento de la cola de trabajo para comenzar la distribución y busca la regla correspondiente a dicho paciente. Envía el paciente hacia su(s) destino(s).
Nombre:	Distributor
Descripción:	Constructor de la clase.
Nombre:	echoService_EchoNotified (sender: object, e: QueryRetrieveTools.Events.EchoNotificationEventArgs)
Descripción:	Envía una solicitud de conexión hacia la o las estaciones destinos antes de comenzar la distribución.
Nombre:	EnqueuePatient (patientCompleteData: PatientCompleteData)
Descripción:	Adicionar un paciente a la cola de trabajo
Nombre:	ProcessDistributingQueue
Descripción:	Revisa la cola de trabajo y si existen elementos invoca al método DistributeCurrentPatient.
Nombre:	Start
Descripción:	Inicia un hilo de ejecución que verifica si hay elementos en la cola de trabajo para comenzar la distribución.
Nombre:	Stop
Descripción:	Detiene el hilo de ejecución.
Nombre:	store_StoreImageCompleted (sender: object, e: QueryRetrieveTools.Events.StoreCompletedEventArgs)
Descripción:	Notifica cuando el envío de un paciente es completado.

Nombre: DistributorLogger	
Tipo de clase: controladora	
Atributo	Tipo
_file	string
Para cada responsabilidad:	
Nombre:	Logger
Descripción:	Crea el tipo de suceso y su descripción.
Nombre:	DistributorLogger
Descripción:	Constructor de la clase, configura la estructura, dirección y capacidad del fichero log.

Nombre: OperationDistributionServ	
Tipo de clase: controladora	
Atributo	Tipo
_hashCode	int
Para cada responsabilidad:	
Nombre:	OperationDistributionServ
Descripción:	Constructor de la clase.
Nombre:	EnqueueOperation (patient: PatientCompleteData, operation: GatewayOperation)
Descripción:	Invoca al método EnqueuePatient de la clase Distributor para adicionar el paciente a la cola de trabajo de la clase Distributor.

Anexo 6 Resultados obtenidos de las pruebas realizadas

Modalidad	Cantidad de imágenes	Tamaño Total (MB)	Tiempos				Tamaño Comprimido (MB)
			Recepción	Procesamiento	Distribución	Total	
XA	107	214	00:15	00:20	05:03	05:38	132
XA	66	132	00:06	00:12	03:11	03:29	67.7
XA	2	120	00:15	00:25	01:50	02:30	36
CT	209	110	00:07	00:15	03:53	04:15	59.5
CT	622	312	00:45	01:04	13:15	15:04	192
DX	21	298	00:15	00:59	05:58	07:12	144
MR	63	34	00:02	00:07	01:54	02:03	15.9
MR	49	15.5	00:02	00:06	01:25	02:03	7.72
US	1	2.25	00:01	00:01	00:30	00:32	336 KB

Anexo 7 Resultados obtenidos del cálculo de la Capacidad de Transmisión Diaria por modalidad

Modalidad	Tm (minutos)	Cm (MB)	Capacidad de Transmisión Diaria (MB)	Conversión (GB)
XA	05:38	214	54 570	53
CT	04:15	110	37 180	36
MR	02:03	34	23 868	23
US	00:32	2.25	6 113	5.9

Anexo 8 Resultados obtenidos del cálculo de la Capacidad de Transmisión Diaria del CAT “Dr. Salvador Allende”

Modalidad	Cantidad de estudios	Tamaño de un estudio (MB)	Capacidad de Transmisión (MB)
CT	50	110	5 500
MR	40	34	1 360
US	30	2.25	67.5
Total			6 927.5

Anexo 9 Promedio de la Capacidad de Transmisión Diaria del sistema para las modalidades de tomografía, resonancia magnética y ultrasonido

Modalidad	Capacidad de Transmisión Diaria (MB)
CT	37 180
MR	23 868
US	6 113
Promedio de Transmisión Diaria	22 387