

Universidad de las Ciencias Informáticas

Facultad 2



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

Desarrollo del servidor de listas de trabajo DICOM
para la solución PACS-RIS del Centro de
Informática Médica

Autores: Claudia Jimenez Heredia
Alexy Dumenigo Aguila

Tutores: Ing. Luis Eduardo González Abreu
Ing. Maikel Sánchez Dieguez

Ciudad de La Habana, 25 de Junio de 2015
“Año 57 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 25 días del mes de Junio del año 2015

Autores:

Claudia Jimenez Heredia

Alexy Dumenigo Aguila

Tutores:

Ing. Luis Eduardo González Abreu

Ing. Maikel Sánchez Dieguez

DATOS DE CONTACTO

Tutores:

Ing. Luis Eduardo González Abreu: Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2010. Es especialista del Departamento de Desarrollo de Aplicaciones del Centro de Informática Médica de la Universidad de las Ciencias Informáticas donde se desempeña como líder del equipo de desarrollo del sistema RIS. Correo electrónico: legonzalez@uci.cu

Ing. Maikel Sánchez Dieguez: Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2010. Es especialista del Departamento de Desarrollo de Componentes del Centro de Informática Médica de la Universidad de las Ciencias Informáticas donde se desempeña como líder del equipo de desarrollo del sistema PACS. Correo electrónico: msdieguez@uci.cu

DEDICATORIA

Claudia

Como culminación de mis estudios universitarios, dedico esta tesis a mis padres por haberme dado tanto amor y cariño durante todos estos años.

Por enseñarme lo que significa tener una familia.

A mi hermano, por ser mi eterno compañero.

Alexy

Dedico esta tesis a mis padres, que me hicieron quien soy y me alentaron a seguir mi propio camino. De ellos aprendí la importancia del trabajo, del esfuerzo y de ser cada día mejores en lo que hacemos.

A mi hermano, para que no dude en perseguir sus propias metas.

AGRADECIMIENTOS

De Claudia

A mami y a papi por dedicar tanto esfuerzo y sacrificio para que yo pudiera tener un futuro mejor, por tanto amor.

A Alfredito por ser más que un hermano mi amigo.

A Iraldo por apoyarme en tantos momentos difíciles y darle alegría a mi vida.

A Arianna, Mayelin, Yisel, Maidevis, Bexi, Amigo y Aramis por ser esas personitas que llegaron para quedarse.

A la gente de mi aula, los que están y los que no por tantos momentos juntos.

A todo el que ayudó a mi preparación como futura profesional.

A mi partner Alexy, por los momentos buenos y malos en el desarrollo de esta tesis.

De Alexy

A mis padres y hermano, que me apoyaron en todas las formas posibles a lo largo de la carrera.

A mis tíos María Ofelia y Aquino, por brindarme un segundo hogar.

A mis compañeros y amigos, mi otra familia, por los cinco años de soportarnos mutuamente, como corresponde a toda buena familia.

A aquellas personas que de un modo u otro contribuyeron a mi formación y al desarrollo de esta tesis

De forma general los autores de esta tesis quieren dedicar un agradecimiento especial:

A los tutores, Maikel y Luis Eduardo, que nos guiaron en todo el proceso, por sus buenos consejos y por su paciencia.

A los miembros del tribunal, por las críticas oportunas y constructivas, que dieron forma a este trabajo.

A la Revolución y a Fidel, por la idea de una universidad donde hemos podido seguir nuestra vocación y superarnos como profesionales y como seres humanos

RESUMEN

Una lista de trabajo DICOM contiene datos de pacientes y de procedimientos a realizar en un departamento de imagenología. Los servidores de listas de trabajo DICOM permiten que la información del estudio y del paciente se mantenga consistente tanto en el RIS como en el PACS. En el Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas (UCI) se desarrolla una solución PACS-RIS. Como parte de dicha solución, se implementó el componente alas PACSWorklist el cual se encuentra acoplado al RIS. Dicho componente solo permite, a los equipos de adquisición, consultar su lista de trabajo. Además no cuenta con un panel de configuración para su administración. El objetivo de la presente investigación es desarrollar un componente de software encargado del manejo de las listas de trabajo DICOM. Para ello se realizó un estudio de los perfiles de integración propuestos por IHE. Dicho estudio determinó la utilización del perfil Flujo de trabajo programado en dicho componente. El desarrollo del componente estuvo guiado por la metodología RUP. Se utilizó C# como lenguaje de programación, Enterprise Architect 7.5 para el modelado, Visual Studio 2013 como herramienta de desarrollo y PostgreSQL 9.3 como gestor de bases de datos. Al concluir el desarrollo se obtuvo un componente que permite el manejo de las listas de trabajo DICOM entre los sistemas PACS y RIS. El mismo podrá ser utilizado por sistemas de cualquier fabricante, que hagan uso de los servicios definidos por el estándar DICOM para el intercambio de información entre sistemas médicos.

Palabras claves:

DICOM, IHE, listas de trabajo, perfil de integración, servidor

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS SERVIDORES DE LISTAS DE TRABAJO DICOM EN LOS DEPARTAMENTOS DE IMAGENOLÓGÍA.....	6
1.1. SISTEMAS DE ALMACENAMIENTO Y COMUNICACIÓN DE IMÁGENES (PACS)	6
1.2. SISTEMA DE INFORMACIÓN RADOLÓGICA (RIS)	6
1.3. IMAGEN DIGITAL Y COMUNICACIONES EN MEDICINA (DICOM)	7
1.3.1. Surgimiento de DICOM	7
1.3.2. Tipos de objetos y servicios DICOM	8
1.3.3. Protocolo de comunicación DICOM	9
1.3.4. Declaración de conformidad de DICOM	9
1.4. INTEGRANDO LAS EMPRESAS SANITARIAS (IHE).....	10
1.4.1. ¿Qué es IHE?	10
1.4.2. Flujo de trabajo programado (Scheduled Workflow (SWF))	11
1.5. SISTEMAS INFORMÁTICOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN	14
1.6. LENGUAJES, HERRAMIENTAS, TECNOLOGÍAS Y METODOLOGÍA A UTILIZAR	16
CAPÍTULO 2. CARACTERÍSTICAS DEL SERVIDOR DE LISTAS DE TRABAJO DICOM	21
2.1. PROPUESTA DEL SERVIDOR DE LISTAS DE TRABAJO DICOM	21
2.2. MODELO DEL DOMINIO	23
2.3. ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE	24
2.3.1. Requisitos funcionales	25
2.3.2. Requisitos no funcionales	27
2.4. MODELO DE CASOS DE USO DEL SISTEMA	29
2.4.1. Definición de los actores del sistema	29
2.4.2. Diagrama de casos de uso del sistema	30
2.4.3. Especificación de los casos de uso del sistema	31
CAPÍTULO 3. DISEÑO DEL SERVIDOR DE LISTAS DE TRABAJO DICOM	36
3.1. DESCRIPCIÓN DE LA ARQUITECTURA. FUNDAMENTACIÓN.....	36
3.2. PATRÓN ARQUITECTÓNICO	39
3.3. DISEÑO	40
3.3.1. Patrones de diseño	40
CAPÍTULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DEL SERVIDOR DE LISTAS DE TRABAJO DICOM	44

4.1. MODELO DE DATOS	44
4.1.1. Descripción de las tablas de la base de datos	45
4.2. IMPLEMENTACIÓN.....	46
4.2.1. Diagrama de despliegue	46
4.2.2. Diagrama de componentes	48
4.3. TRATAMIENTO DE ERRORES	50
4.4. ESTÁNDAR DE CODIFICACIÓN	51
4.5. VALIDACIÓN DEL SERVIDOR DE LISTAS DE TRABAJO DICOM.....	53
4.6. DECLARACIÓN DE CONFORMIDAD DICOM.....	56
RESULTADOS OBTENIDOS	57
CONCLUSIONES	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXOS	64
ANEXO 1. DIAGRAMAS DE CLASES DEL CASO DE USO GESTIONAR MODALIDAD	64
ANEXO 2. DESCRIPCIÓN DE LAS CLASES DEL DISEÑO.....	64
ANEXO 3. DIAGRAMAS DE SECUENCIA.....	66
ANEXO 4. DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS	67
GLOSARIO DE TÉRMINOS	69

FIGURAS

Figura 1.1 Adquisición en RAD-5.....	13
Figura 1.2 RAD-6 Paso del procedimiento en la modalidad en progreso	14
Figura 2.1 Flujo de trabajo del componente Celsus Worklist.....	22
Figura 2.2 Diagrama del Modelo del dominio del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	24
Figura 2.3 Requisitos funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	26
Figura 2.4 Requisitos no funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	29
Figura 2.5 Diagrama de casos de uso del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	31
Figura 3.1 Arquitectura basada en los estilos Cliente-Servidor y N-Capas del componente Celsus Worklist	38
Figura 3.2 Patrón Singleton utilizado en el desarrollo del componente Celsus Worklist	41
Figura 3.3 Patrón Controlador utilizado en el desarrollo del componente Celsus Worklist	42
Figura 3.4 Patrón Fabricación Pura utilizado en el desarrollo del componente Celsus Worklist	42
Figura 4.1 Modelo de datos del componente Celsus Worklist	45
Figura 4.2 Diagrama de despliegue del componente Celsus Worklist.....	47
Figura 4.3 Diagrama de componentes de Celsus Worklist.....	49
Figura 4.4 Método Create de la clase “AllowedModalitiesController”	51
Figura 4.5 Fragmento de código de la clase “AllowedModalitiesMap”	52
Figura 4.6 Fragmento de código de la clase “WorklistQueries”	52
Figura 4.7 Fragmento de código de la clase “WorklistManager”	52
Figura 4.8 Captura de pantalla al botón <i>Ping RIS</i> de la aplicación Modality Emulator 3.1.5.....	54
Figura 4.9 Captura de pantalla al botón DICOM Echo de la aplicación Modality Emulator 3.1.5	54
Figura 4.10 Captura de pantalla a la vista Select MWL Response de la aplicación Modality Emulator 3.1.5 ..	54
Figura 4.11 Captura de pantalla a la página principal de la aplicación Modality Emulator 3.1.5	55
Figura 4.12 Captura de pantalla al estado <i>IN PROGRESS</i> del estudio en el RIS	55
Figura 4.13 Captura de pantalla al botón <i>Send MPPS Progress</i> de la aplicación Modality Emulator 3.1.5	55
Figura 4.14 Captura de pantalla al estado <i>DISCONTINUED</i> del estudio en el RIS	56
Figura 5 Diagrama de clases del diseño del caso de uso Gestionar modalidad.....	64
Figura 6 Diagrama de secuencia Adicionar modalidad	66

TABLAS

Tabla 2.1 Entidades y conceptos del Modelo del dominio del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM	23
Tabla 2.2 Requisitos funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM	25
Tabla 2.3 Requisitos no funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	27
Tabla 2.4 Definición de los actores referentes al servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM.....	30
Tabla 2.5 Descripción del caso de uso Gestionar modalidad.....	31
Tabla 4.1 Descripción de la tabla “tbl_allowed_modalities” perteneciente a la base de datos del componente Celsus Worklist	45
Tabla 4.2 Descripción de los elementos del diagrama de despliegue del componente Celsus Worklist	47
Tabla 4.3 Descripción de los elementos que conforman el diagrama de componentes del servidor de listas de trabajo DICOM	49
Tabla 6 Descripción de la clase AllowedModalitiesController	64
Tabla 7 Descripción de la tabla “tbl_remote_scp_configuration”	67
Tabla 8 Descripción de la tabla “tbl_local_configuration”	67
Tabla 9 Descripción de la tabla “tbl_log”	68
Tabla 10 Descripción de la tabla tbl_user	68

INTRODUCCIÓN

La radiología es la rama de la salud que se encarga de generar imágenes del interior del cuerpo mediante diferentes agentes físicos para fines diagnósticos (1). En 1972, el británico Hounsfield (2) presenta en Londres el primer tomógrafo computarizado, en el cual la imagen no es analógica, como en la radiología convencional, sino digital. A partir de este momento comienzan los primeros trabajos con la radiología digital, trayendo consigo ventajas tanto para el paciente como para el personal sanitario, entre las que se incluyen, la disminución en la dosis de radiación a la que se expone el paciente, la rapidez del proceso y la disponibilidad de la información que puede ser utilizada por el radiólogo (3).

La creciente tendencia hacia el uso de la radiología digital, especialmente en el diagnóstico por imagen, y el constante intercambio de información entre sistemas informáticos médicos, puso de relieve la necesidad de estandarizar los protocolos de comunicación y los formatos de la información en sanidad. Es por esto que surge el estándar DICOM¹. El mismo fue desarrollado conjuntamente por ACR² y NEMA³. Es un estándar de comunicación entre sistemas de información y a la vez un formato de almacenamiento de imágenes médicas. (4)

Los ficheros DICOM son la base de los sistemas conocidos en el mundo como Sistemas de Almacenamiento y Comunicación de Imágenes (PACS⁴). Un PACS está compuesto por equipos de adquisición de imágenes médicas⁵, estaciones de visualización y procesamiento de la información entre otros, todos integrados por las redes digitales y software de aplicación (5).

¹ Digital Imaging and Communication in Medicine

² American College of Radiology

³ National Electrical Manufacturers Association

⁴ Picture Archiving and Communication System

⁵ En lo adelante los términos equipo de adquisición de imágenes médicas y modalidad, se utilizan para representar al equipo que obtiene las imágenes médicas, en un estudio imagenológico a un paciente determinado

Los sistemas PACS no cubren todas las necesidades requeridas por los departamentos de diagnóstico por imágenes, para complementarlos se utilizan los Sistemas de Información Radiológica (RIS⁶). Estos se desempeñan en la gestión de la información que se genera en los departamentos de radiología, permitiendo la informatización de la lista de trabajo de los equipos de adquisición de imágenes médicas y especialistas de la institución, la organización del flujo de trabajo de los departamentos de imagenología, la homogenización de los reportes de estudios imagenológicos realizados a los pacientes, de los reportes estadísticos de la institución y las hojas de cargo por servicios. (6)

El RIS se encarga de proporcionar al PACS la información sobre las citas existentes. A su vez el PACS notificará al RIS que el estudio ha sido realizado y completado para posteriormente proporcionar al radiólogo las imágenes de la exploración realizada, de forma que este, pueda elaborar el informe correspondiente en el RIS.

La correcta comunicación entre el RIS y el PACS es esencial en el funcionamiento de cualquier departamento de imagenología, pues permite mantener una integridad referencial entre pacientes y procedimientos en ambos sistemas. La forma en que interactúan dichos sistemas se ha estandarizado con el surgimiento y uso de IHE⁷, una iniciativa que tiene como finalidad, mejorar la comunicación entre los diferentes sistemas sanitarios. Los sistemas desarrollados de conformidad con IHE se comunican mejor entre sí y son más fáciles de aplicar, lo que permite un mejor uso de la información (7).

Con el objetivo de impulsar la informatización de la sociedad cubana, en el 2002 se crea la Universidad de las Ciencias Informáticas (UCI). Esta universidad posee varios centros de desarrollo de software entre los que se encuentra el Centro de Informática Médica (CESIM). Una de las aplicaciones que se desarrolla en dicho centro es la solución PACS-RIS, la cual se encuentra en proceso de despliegue en varios hospitales de Cuba y de la República Bolivariana de Venezuela.

⁶ Radiology Information System

⁷ Integrating the Healthcare Enterprise

Ante el problema de mantener identificados a los pacientes y estudios de manera única en la solución PACS-RIS desarrollada por el CESIM, se creó el componente alas PACSWorklist. Este posee un servicio básico que solo permite a los equipos de adquisición de imágenes médicas, consultar su lista de trabajo, por lo que, una vez que se comience la ejecución del estudio en dicho equipo, no es posible actualizar el estado del mismo en el RIS y en el PACS. Esto trae como consecuencia que las listas de trabajo solicitadas siempre envíen los datos de las mismas citas a los equipos de adquisición. Debido a esto no se puede llevar a cabo un control sobre los estudios para saber en qué estado se encuentran.

Este sistema se encuentra acoplado al RIS desarrollado por el CESIM. Esto hace que, en caso de que el RIS evolucione, no se puede seguir contando con el mismo servidor de listas de trabajo sin tener que realizarle modificaciones, ni podrá ser utilizado por un RIS de otro fabricante.

La solución alas PACSWorklist no posee un panel de configuración, haciendo difícil su administración y configuración por parte del personal encargado en las instituciones hospitalarias. Esto trae como consecuencia que presente problemas para su configuración y mantenimiento.

Tomando en cuenta la situación descrita anteriormente, surge el siguiente **problema a resolver**: el funcionamiento del componente alas PACSWorklist del Centro de Informática Médica dificulta el manejo de las listas de trabajo DICOM. En correspondencia con el problema se define el **objeto de estudio**: listas de trabajo DICOM. El **campo de acción** está centrado en: manejo de las listas de trabajo DICOM acorde con lo planteado por la normativa internacional IHE.

Para dar solución al problema, se plantea como **objetivo general**: desarrollar un componente de software encargado de manejar las listas de trabajo DICOM acorde con lo planteado por la normativa internacional IHE.

En aras de cumplir el objetivo planteado se proponen las siguientes **tareas de la investigación**:

1. Caracterización de las tendencias actuales del desarrollo de servidores de listas de trabajo DICOM con vistas a la definición de los requisitos del componente.
2. Caracterización de los elementos de las listas de trabajo DICOM, propuestos por la normativa internacional IHE, para definir los requisitos de integración con la solución PACS-RIS.

3. Asimilación de las herramientas, tecnologías y metodología empleadas para el desarrollo de la solución PACS-RIS.
4. Generación de los artefactos correspondientes a las fases de desarrollo de software, propuestos por la metodología de desarrollo Proceso Unificado de Desarrollo de Software (RUP).
5. Implementación del componente de software aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.
6. Elaboración de la declaración de conformidad con el estándar DICOM para el componente implementado.

Los métodos científicos utilizados en la investigación son:

Métodos Teóricos

Histórico-Lógico: se utiliza con el objetivo de realizar un análisis histórico acerca de las tendencias en el desarrollo de servidores de listas de trabajo DICOM. Como parte inicial de la investigación se realiza un estudio del arte de la problemática planteada, teniendo en cuenta los sistemas existentes que hacen uso de las listas de trabajo DICOM. En este estudio se analizan las principales características de estas soluciones.

Análisis y Síntesis: se hace uso del mismo para determinar, a partir de un estudio de los elementos que conforman las listas de trabajo DICOM, las relaciones y nexos esenciales que existen entre los mismos. Esto permite conocer los datos necesarios a utilizar en el intercambio de información entre los sistemas PACS y RIS.

Inductivo-Deductivo: a partir de un análisis del funcionamiento del componente alas PACSWorklist, se evalúan los elementos que constituyen la problemática existente y se definen los aspectos particulares que se utilizarán para elaborar la propuesta de solución.

Modelación: con el objetivo de representar de forma simplificada la realidad y descubrir nuevas relaciones y cualidades del objeto, se crean diagramas que posibilitan un mejor entendimiento de los procesos.

El presente trabajo de diploma se encuentra estructurado en cuatro capítulos:

CAPÍTULO 1. Fundamentación Teórica de los servidores de listas de trabajo DICOM en los departamentos de imagenología, se realiza la fundamentación teórica de los elementos que se utilizarán en el posterior desarrollo del servidor de listas de trabajo DICOM para la solución PACS-RIS. En este se explican los conceptos asociados al campo de acción que intervienen en la investigación. Se realiza, además, un estudio de los sistemas informáticos existentes que utilizan los servicios de listas de trabajo DICOM a nivel internacional. Asimismo se describen las herramientas, tecnologías y metodología a utilizar en el proceso de desarrollo del componente.

CAPÍTULO 2. Características del servidor de listas de trabajo DICOM, se propone la solución al problema planteado, se detallan sus principales características y flujo de trabajo. Además se describe el modelo del dominio del servidor de listas de trabajo DICOM y los requisitos funcionales y no funcionales que el sistema deberá cumplir. Se presenta además, el diagrama de casos de uso así como la descripción de los mismos y de los actores que intervienen en cada uno de ellos.

CAPÍTULO 3. Diseño del servidor de listas de trabajo DICOM, se realiza el modelado del diseño de la solución planteada y se describe la arquitectura a utilizar brindando una serie de características que justifican la misma. Se realiza el modelado del diseño a partir de los diagramas de clases del diseño y de los diagramas de secuencia de los casos de uso.

CAPÍTULO 4. Implementación y validación del servidor de listas de trabajo, se realiza una descripción de los elementos que corresponden a la fase de implementación. Se describe el modelo de datos en el que se muestra la estructura que almacena la información requerida por el sistema. Se muestra el diagrama de despliegue y el de componentes así como los elementos que conforman los mismos y las relaciones entre ellos. Se describen las estrategias de codificación y además cómo se llevará a cabo el tratamiento de errores en el sistema. De igual forma se realiza la validación del sistema implementado, permitiendo asegurar que los resultados generados fueron los que se esperaban.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS SERVIDORES DE LISTAS DE TRABAJO DICOM EN LOS DEPARTAMENTOS DE IMAGENOLOGÍA

En el presente capítulo se exponen las bases teóricas y conceptuales que se utilizarán en el posterior desarrollo del servidor de listas de trabajo DICOM para la solución PACS-RIS. Se explican los principales conceptos asociados al campo de acción que intervienen en la investigación, así como otros elementos necesarios para la comprensión del funcionamiento de los servidores de listas de trabajo. Por otra parte, se realiza un estudio sobre sistemas PACS y RIS que utilizan las listas de trabajo en el mundo. Asimismo se describen las tecnologías y herramientas, así como la metodología que se utilizará en el proceso de desarrollo del software.

1.1. Sistemas de almacenamiento y comunicación de imágenes (PACS)

Un sistema de almacenamiento y comunicación de imágenes (PACS) está compuesto por equipos de adquisición de imágenes médicas, estaciones de visualización y procesamiento de la información entre otros, todos integrados por las redes digitales y software de aplicación. (7) Estos se encargan de controlar la información generada durante el proceso de adquisición de imágenes médicas. Es decir, toda la información relacionada de forma directa con los estudios, las propias imágenes y los detalles de su proceso de creación, el envío a las estaciones de diagnóstico, las características de estas estaciones, y hasta su posterior impresión y distribución. (8).

Con el objetivo de que los sistemas PACS funcionen correctamente con modalidades y estaciones de trabajo de diferentes fabricantes, existen una serie de estándares de imagen digital que se han definido para ello. Entre estos se encuentra el estándar DICOM. (9)

1.2. Sistema de información radiológica (RIS)

Un RIS tiene como principal función, la gestión de la información radiológica dentro de las instituciones hospitalarias. Estos sistemas reducen los errores provocados por el registro manual de la información, aumentando de este modo el nivel de seguridad y mejorando la atención del paciente, ya que informatiza toda la actividad radiológica del mismo. (10)

Un RIS permite, además, visualizar las listas de trabajo por cada modalidad. Esta lista de trabajo está constituida por los estudios de imágenes pendientes. Generalmente se puede acceder a ella electrónicamente y contiene detalles sobre las tareas a realizar, tales como el nombre del paciente, número de identificación (ID), número de registro y otros datos relevantes que se hayan introducido. Puede o no determinar concretamente el programa y el equipo en que se va a realizar dicho trabajo. (11) Los sistemas RIS se comunican con los PACS a través de estándares establecidos como DICOM y HL7⁸. Esta comunicación debe ser en ambos sentidos para mantener actualizada la información del paciente y el estado de los estudios e informes asociados a estos en todo momento.

1.3. Imagen digital y comunicaciones en medicina (DICOM)

En este epígrafe se muestra cómo DICOM define sus objetos y servicios y cómo estos se unen para crear las unidades funcionales del mismo. Además se realiza una breve descripción de cómo funciona DICOM como un protocolo de comunicación entre sistemas médicos, permitiendo el intercambio de información entre estos. Por último se define qué es una declaración de conformidad con el estándar DICOM y los elementos que contiene.

1.3.1. Surgimiento de DICOM

La introducción de las imágenes digitales en la década de 1970 y el constante intercambio de información entre sistemas informáticos médicos de diversos fabricantes, trajo consigo que la ACR y NEMA se unieran para crear un comité. Este tuvo como objetivo crear un estándar para la transmisión de imágenes médicas y la asociación de su información. Con el objeto de hacer patente su relación hereditaria respecto al estándar fundado por ACR y NEMA, la versión original de DICOM ha sido denominada DICOM versión 3.0. (12) Los objetivos de DICOM son: alcanzar la compatibilidad y mejorar el flujo de trabajo entre los sistemas de imágenes y otros sistemas de información, en entornos de atención médica en todo el mundo (12)

⁸ Health Level Seven

1.3.2. Tipos de objetos y servicios DICOM

Los objetos DICOM se denominan Definición de Objetos de Información, por sus siglas en inglés IOD (*Information Object Definition*). Estos objetos se clasifican en compuestos, los cuales corresponden a varias entidades del mundo real, y en simples o normalizados, que corresponden a una única entidad. Cada uno de los IOD compuestos está formado por varios IOD normalizados. Además, maneja dos tipos de servicios (DIMSE⁹): los servicios compuestos DIMSE-C y los servicios normalizados DIMSE-N. Los servicios son las acciones que se pueden aplicar a los objetos, copiar, almacenar, seleccionar y escribir, son ejemplos de acciones posibles. (4, 13)

Los servicios incluidos en DIMSE-C son: C-STORE, C-FIND, C-GET, C-MOVE y C-ECHO; los de DIMSE-N son: N-EVENT-REPORT, N-GET, N-SET, N-ACTION, N-CREATE y N-DELETE. Así, un mensaje DICOM está compuesto de un grupo de comandos y un grupo de datos que es dependiente del primero; en el grupo de comandos se indica el servicio DIMSE que se ejecuta con los datos. (14)

Los tipos de servicio se combinan con los objetos IOD y definen las unidades funcionales de DICOM. Estas combinaciones servicio-objeto se denominan clases (SOP¹⁰). De esta manera DICOM define cuales son las operaciones que pueden ser ejecutadas y sobre qué objetos. (15)

A través de las clases SOP se efectúa el intercambio de información. La base de estos intercambios es la utilización de protocolos cliente/servidor. Cada vez que dos aplicaciones o equipos deciden conectarse para intercambiar información, uno de los dos desarrolla el papel de nodo proveedor de la clase de servicio (SCP¹¹), mientras que el otro nodo, toma el papel de usuario de la clase de servicio (SCU¹²) (13).

⁹ DICOM Message Service Element

¹⁰ Service-Object Pair

¹¹ Service Class Provider

¹² Service Class User

1.3.3. Protocolo de comunicación DICOM

El protocolo de comunicación DICOM utiliza el modelo TCP/IP para definir su propio lenguaje de red, que especifica cómo formatear e intercambiar objetos DICOM (imágenes médicas y su información asociada) entre equipos dentro de un entorno hospitalario. Como resultado, el modelo TCP/IP ve aumentada su funcionalidad al añadir una capa de aplicación DICOM con protocolos específicos. La capa de aplicación DICOM se organiza en dos niveles (16):

- El nivel superior: presenta los servicios de alto nivel, que consiste en las clases y el protocolo DIMSE.
- El nivel inferior: presenta los servicios de bajo nivel, que consisten en primitivas para la negociación de la asociación entre entidades de aplicación y el intercambio de estructuras para la gestión de dicha asociación, más conocido como PDU por sus siglas en inglés (*Protocol Data Unit*). Esta capa es conocida como protocolo de capa superior DICOM, por sus siglas en inglés DICOM UL (*DICOM Upper Layer*).

Según lo que especifica el modelo TCP/IP, las aplicaciones de red pueden comunicarse entre sí mediante la definición de su interfaz de red. Para poder usar esta interfaz en la red, debe tener asignada una dirección IP que identifique al dispositivo y un puerto donde enviar los datos.

DICOM denomina una aplicación de red DICOM como entidad de aplicación (AE) y añade un campo para asignarle un nombre que la identifique denominado *AE Title*. Por tanto para configurar un dispositivo en una red DICOM es importante definir los campos *AE Title*, dirección IP y puerto.

1.3.4. Declaración de conformidad de DICOM

La declaración de conformidad de DICOM, es la certificación de cumplir el estándar y debe ser descrita para cada modalidad y dispositivo y para cada versión del producto, indicando además para cada servicio DICOM el tipo correspondiente (cliente, servidor o ambos). Las principales clases de servicio DICOM son las siguientes (13):

- Clase de almacenamiento (*Class Storage*)
- Clase de consulta y recuperación (*Class Query & Retrieve*)
- Impresión (*Class Print*)

- Gestión de listas de trabajo (*Basic Worklist Management*)
- Paso del procedimiento realizado por la modalidad (*Modality Performed Procedure Step*)

La clase de Gestión de listas de trabajo permite el acceso a las listas de trabajo. Una lista de trabajo presenta información relacionada con una lista de tareas a realizar (pacientes citados) para la modalidad.

1.4. Integrando las empresas sanitarias (IHE)

Como parte de la presente investigación en este epígrafe se mencionan los elementos que conforman la iniciativa IHE. De todos los perfiles propuestos por IHE, en este documento se describe el perfil, Flujo de trabajo programado ya que es el que más se ajusta a la problemática planteada. De igual forma se describen las transacciones que define dicho perfil para el manejo de las listas de trabajo DICOM. Estas transacciones se especifican en el marco técnico de IHE.

1.4.1. ¿Qué es IHE?

IHE es una iniciativa de profesionales de la salud y empresas proveedoras de equipos y software médicos. Esta iniciativa tiene como objetivo mejorar la forma en que los sistemas de información para la salud se comunican. La herramienta práctica que ofrece IHE es la definición de perfiles de integración, los cuales se definen como una agrupación de actores, transacciones y vocabulario, que utilizan estándares como DICOM y HL7 para la integración de sistemas de manera que proporcionen una mayor interoperabilidad y disminuyan los errores en el flujo de trabajo. (17)

Los perfiles de integración IHE describen una necesidad clínica de integración de sistemas y la solución para llevarla a cabo. Definen los componentes funcionales, a los que se les llama actores IHE, y las posibles transacciones que cada actor deberá llevar a cabo. (17)

Estos perfiles, permiten gestionar el conjunto integrado de sistemas de información necesario para proporcionar una mejor atención sanitaria. Su alternativa sería el desarrollo de interfaces hechas a medida para cada instalación, lo que resulta más costoso y requiere el mantenimiento de estas interfaces durante toda la vida útil del sistema. Los perfiles de integración definen claramente cómo deben encajar todas las piezas basándose en estándares aceptados globalmente. (18)

1.4.2. Flujo de trabajo programado (*Scheduled Workflow (SWF)*)

SWF es un perfil de integración que propone IHE. SWF establece un flujo de información continuo en el proceso de realización de una prueba de imagen. El mismo especifica las transacciones que mantienen la consistencia de la información sobre el paciente desde su registro hasta la visualización de las imágenes, pasando por la petición, la cita, la adquisición de imágenes y su almacenamiento. (19)

Este perfil permite determinar cuáles imágenes y objetos se han almacenado y están disponibles para permitir que se realicen los posteriores pasos del flujo de trabajo. Además, este perfil hace posible que coincidan las imágenes, reportes de diagnóstico y otros elementos que se hayan adquirido durante un caso de trauma a un paciente que no ha sido previamente identificado. En el ejemplo del caso de trauma, permite la futura reconciliación de los datos obtenidos del paciente con su historia clínica electrónica, haciendo posible su posterior utilización. (19)

En el perfil SWF, la modalidad presupone que el RIS sostiene varios servicios DICOM, utilizando atributos específicos y respondiendo a los mensajes de manera particular. Estos servicios son:

- MWL¹³ DICOM como SCP
- MPPS¹⁴ DICOM como SCP

Mediante el uso del servicio MWL de DICOM se puede transferir información sobre peticiones y pacientes a los equipos de adquisición. SWF asegura poder realizar un seguimiento del estado del flujo de trabajo (en progreso, completado, interrumpido). IHE necesita que la modalidad envíe mensajes MPPS al RIS y al PACS. Si el RIS no está capacitado para soportar MPPS DICOM, no podrá recibir actualizaciones del estado de los estudios. (18)

El marco técnico de trabajo que define IHE está basado en actores que interactúan mediante transacciones. Los actores son sistemas de información o componentes de estos, que producen, gestionan o actúan sobre

¹³ Modality Worklist

¹⁴ Modality Performed Procedure Step

información relativa a las operaciones de la empresa. Por otro lado, las transacciones son interacciones entre actores que transfieren la información requerida a través de mensajes estandarizados. (17)

Los principales actores de los cuales hace uso SWF son: el DSS/OF¹⁵, el *PPS Manager*¹⁶ y la Modalidad. Estos actores utilizan las transacciones definidas en el marco técnico de trabajo de IHE: RAD-5, RAD-6 y RAD-7.

RAD-5 Consulta de la lista de trabajo de la modalidad (*Query Modality Worklist*) es una transacción utilizada por los clientes de listas de trabajo y el actor DSS/OF. Este último es el encargado de la gestión de pacientes y de las órdenes de los estudios a realizar, además es responsable de las actualizaciones de los procedimientos programados y de responder a las solicitudes de las listas de trabajo. Esta transacción se realiza bajo dos circunstancias (20):

1. Planificar una cita. (Adquisición)

Cuando el paciente se presenta para el procedimiento programado, el especialista que lo atiende, revisa la información referente al procedimiento a realizar, la exactitud del procedimiento solicitado, los posibles comentarios añadidos por el especialista que remite, entre otros aspectos. La Modalidad de adquisición la constituyen los sistemas que adquieren y crean las imágenes médicas. En la misma se utilizan los mensajes C-FIND de DICOM MWL para realizar consultas al DSS/OF y obtener los Pasos del Procedimiento Programado. De esta forma la Modalidad de adquisición toma el rol de SCU y el DSS/OF el de SCP. La lista

¹⁵ Department System Scheduler/Order Filler

¹⁶ Performed Procedure Step Manager

se descarga hacia la Modalidad de adquisición y el tecnólogo verifica la información. Estos datos se incluyen en el encabezado de las imágenes que se generen. (20) Ver Figura 1.1.

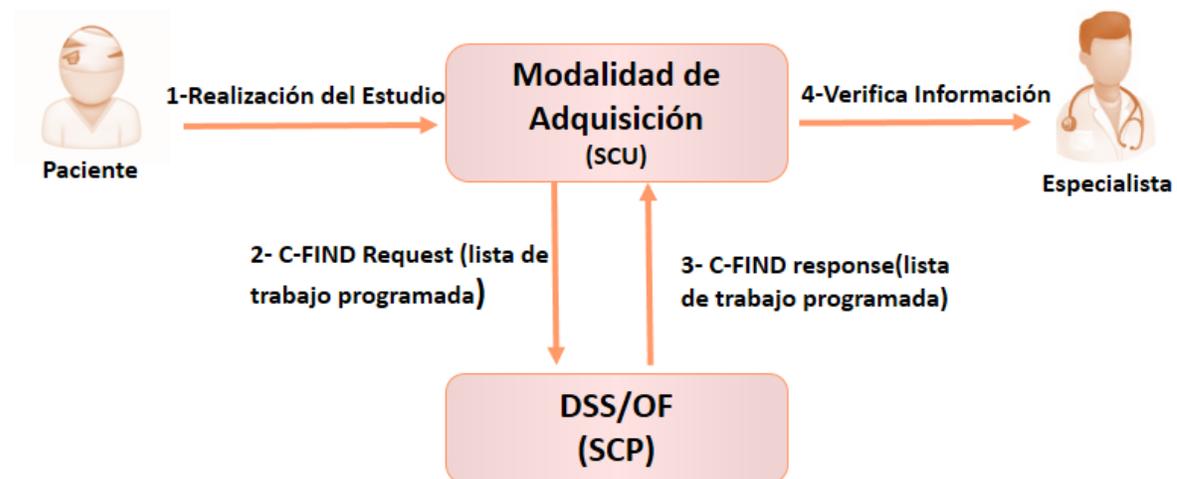


Figura 1.1 Adquisición en RAD-5

(fuente: elaboración propia)

2. Planificar la importación de objetos de evidencia ya existentes o información en formato duro. (Importación)

En el caso de la importación, esta se realiza con objetos DICOM o se crean objetos DICOM como producto de la misma, por ejemplo la digitalización de videos médicos. Esta puede ser planificada como parte de una cita o al recibirse un medio físico con imágenes del paciente (PDI¹⁷), las cuales son requeridas en una consulta.

RAD-6 Paso del procedimiento en la modalidad en progreso (*Modality Procedure Step In Progress*) es otra transacción utilizada por los actores DSS/OF, los de la Modalidad de adquisición y el PPS Manager. Este último se encarga de actualizar el estado del paso del procedimiento programado. Inicialmente, la Modalidad de adquisición utiliza el servicio N-CREATE para informarle al *PPS Manager* que el procedimiento programado

¹⁷ Portable Data for Imaging

ha sido iniciado y se encuentra en progreso. Seguidamente el *PPS Manager* hará uso del servicio N-CREATE para actualizar dicha información en el DSS/OF. (21) Ver Figura 1.2.



Figura 1.2 RAD-6 Paso del procedimiento en la modalidad en progreso
(fuente: elaboración propia)

RAD-7 Paso del procedimiento en la modalidad completado o descontinuado (*Modality Procedure Step Completed/Discontinued*) al igual que la anterior es utilizada por los actores DSS/OF, *PPS Manager* y los actores de la Modalidad de adquisición. La Modalidad de adquisición utiliza el servicio N-SET para informarle al *PPS Manager* que el procedimiento programado ha sido completado o interrumpido. El *PPS Manager* utiliza el servicio N-SET para inmediatamente actualizar la información del procedimiento programado en el DSS/OF. (22) Ver Figura 1.3.



Figura 1.3 RAD-7 Paso del procedimiento en la modalidad completado o descontinuado
(Fuente: elaboración propia)

1.5. Sistemas informáticos existentes vinculados al campo de acción

Durante la investigación se realizó un estudio de las principales tendencias en el desarrollo de servidores de listas de trabajo DICOM. Como parte de dicha investigación, en este epígrafe se describen algunos sistemas informáticos existentes que utilizan las listas de trabajo DICOM en el mundo. En el mismo se destacan las principales funcionalidades de estos, lo que permitirá determinar requisitos que cumplirá el componente de software a desarrollar.

Xcelera Connect Release 1.3, perteneciente a los sistemas médicos de Philips, es un motor de interfaz altamente programable que conecta al Sistema de Gestión de Imagen de Xcelera (IMS¹⁸) y las modalidades que soportan el estándar DICOM con el sistema de información hospitalaria (HIS¹⁹) y/o el sistema de información cardiológica (CIS²⁰). Soporta los servicios de gestión de las listas de trabajo DICOM (MWLM²¹) y de procedimiento realizado en la modalidad (MPPS) como SCP y SCU. Xcelera Connect permite un flujo de información entre pacientes y exámenes desde modalidades y hacia ellas, a través de la administración de listas de trabajo y pasos de procedimiento programado por modalidad, conforme a lo que establece DICOM. (23)

iSite PACS Worklist es otra solución perteneciente a Philips. La misma interactúa con la infraestructura existente del centro de salud donde se encuentre, lo que permite que pueda ser utilizado por sistemas de diversos fabricantes. En un centro donde se encuentre la solución iSite PACS Worklist el manejo de listas de trabajo se realiza en el PACS ya que el mismo se integra con el HIS y el RIS preexistente, esto es posible mediante el uso de un dispositivo intermedio denominado bróker. Este dispositivo acepta los mensajes HL7 procedentes del RIS, traduce o mapea los datos para producir mensajes DICOM, y éstos son transmitidos al PACS. De esta forma el PACS y las modalidades pueden aceptar datos del RIS, lo que permite la colaboración de los tres, por ejemplo, en la creación y gestión de las listas de trabajo. (24)

IMPAX RIS DICOM Server en su versión 6, es el servicio de conectividad para dispositivos de imagen del sistema RIS de Agfa. Actúa como un gestor de las listas de trabajo en la modalidad asumiendo el rol de SCP y además como un gestor del procedimiento realizado en la modalidad, por sus siglas en inglés MPPS Manager (*Modality Performed Procedure Step Manager*). En los sistemas de Agfa, IMPAX RIS DICOM Server

¹⁸ Xcelera Image Management System

¹⁹ Hospital Information Systems

²⁰ Cardiology Information System

²¹ DICOM Modality Worklist Management

solo se instala cuando el servicio MPPS es requerido o cuando no hay un PACS de Agfa. Normalmente las listas de trabajo son manejadas por las aplicaciones *Connectivity Manager* o PACS Bróker. (25)

HealthE PACS Worklist es una solución desarrollada por *Healthline Information Systems* basada en IHE, HL7 y DICOM. El mismo actúa como una interfaz intérprete entre el RIS, el HIS y el PACS que permite a las modalidades consultar su lista de trabajo. Entre sus funciones se encuentran la de indicar cuándo el procedimiento programado ha sido concluido en el equipo de adquisición. Además permite el trabajo con múltiples RIS y HIS posibilitando la correcta asociación de la información en cada uno de ellos. Entre las principales desventajas que posee este sistema es que ha sido desarrollado para trabajar con las estaciones de trabajo pertenecientes a Cedara, una empresa líder en el desarrollo de visores para imágenes médicas. (26)

En el mundo existen varios sistemas de imágenes médicas desarrollados por poderosas compañías como Philips, Agfa, Kodak entre otras, que hacen uso de los servicios de listas de trabajo. La mayoría de estos sistemas han sido desarrollados teniendo en cuenta las necesidades de clientes específicos. De igual forma no hacen uso de las normas definidas por IHE para la utilización de estándares como DICOM y HL7. La utilización de estas normas les permitiría una integración ideal donde exista un flujo de información libre de errores, permitiendo los mejores resultados en la atención al paciente.

La utilización de alguno estos sistemas no resolvería el problema identificado para la solución PACS-RIS pues no satisfacen la problemática descrita. Por otra parte, estos sistemas tienen como principal desventaja que son sistemas propietarios, lo que traería como consecuencia costos excesivos en el uso y mantenimiento de los mismos. Esto trae consigo que no se pueda modificar el código de los mismos para su adaptación al problema planteado.

1.6. Lenguajes, herramientas, tecnologías y metodología a utilizar

A continuación se describen las herramientas, tecnologías y la metodología a utilizar, en el posterior desarrollo del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM. Las mismas han sido las definidas por el proyecto de imágenes médicas perteneciente a dicho centro.

Lenguaje Unificado de Modelado 2.1 (UML): es utilizado para especificar, visualizar construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas.(27). Este

lenguaje prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, además describe la semántica esencial de lo que estos diagramas y símbolos significan. Las fases de desarrollo de los sistemas que soportan UML son: Análisis de requisitos, Análisis, Diseño, Implementación y Pruebas. El uso de UML trae muchas ventajas entre las que se encuentran: mejora en los tiempos de desarrollo de los sistemas, modelado de sistemas utilizando conceptos orientados a objetos y mejor soporte a la planeación y al control de proyectos. (28)

Lenguaje de programación C# 5.0: está diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en el Framework .NET. Es un lenguaje sencillo, moderno, proporciona seguridad y está orientado a objetos. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de *Common Language Runtime* (CLR). Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones. C# se presenta como Visual C# en el conjunto de programas Visual Studio .NET. Visual C# utiliza plantillas de proyecto, diseñadores, páginas de propiedades, asistentes de código, un modelo de objetos y otras características del entorno de desarrollo. (29)

Microsoft .NET Framework 4.5: es un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: *Common Language Runtime* (CLR), que es el que controla las aplicaciones en ejecución, y la biblioteca de clases de Framework .NET, que proporciona una biblioteca de código probado y que puede ser utilizada por los desarrolladores en sus propias aplicaciones. Proporciona independencia e interoperabilidad entre lenguajes, por lo que se puede interactuar con otras aplicaciones y componentes de Framework.NET independientemente del lenguaje con el fueron desarrolladas. Ofrece varios servicios entre ellos (30):

- Administración de la memoria
- Amplia biblioteca de clases
- Compatibilidad de versiones
- Ejecución en paralelo

Enterprise Architect 7.5: es una herramienta de análisis y diseño basado en UML 2.1 la cual cubre las diferentes etapas de desarrollo del software, desde la obtención de los requisitos, diseño del modelo y

pruebas. Esta aplicación ha sido desarrollada con una amplia gama de funciones para todo tipo de profesionales. Posee características avanzadas, rápidas y flexibles que permiten diseñar, implementar, probar, mantener modelos utilizando UML y otros estándares para el modelado. (31)

Microsoft Visual Studio 2013: constituye el IDE de programación por excelencia en la plataforma .NET. Es un completo y complejo sistema integrado de gestión en la programación de soluciones informáticas. Con el pasar de las versiones y el crecimiento en cada una de ellas, Visual Studio ha unificado en una sola herramienta servidores de gestión de ciclo de vida, laboratorios de prueba, sistemas de integración continua, repositorios de código compartido avanzadas, etc. (32)

Visual Studio 2013 permite desarrollar aplicaciones sobre la base de la última versión de ASP.NET, en la que se pueden mezclar las diferentes tecnologías de la misma. Además se puede utilizar un solo ambiente y un solo lenguaje de programación, con lo cual el programador puede desarrollar para todas las plataformas manteniendo un solo código y enfocando su conocimiento en un solo lenguaje. Con Visual Studio 2013 se pueden construir aplicaciones para todos los dispositivos, plataformas y sistemas operativos soportados, además de realizar decenas de operaciones y validaciones de depuración las cuales permiten encontrar fallos de manera fácil. (32)

ASP.NET MVC 5: es un Framework de aplicaciones web que implementa el patrón arquitectónico modelo-vista-controlador (MVC). Este modelo de desarrollo Web unificado incluye los servicios necesarios para crear aplicaciones Web empresariales con el mínimo de código. Forma parte del Framework.NET. Con ASP.NET el código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime*, entre ellos Microsoft Visual Basic y C#. ASP.NET MVC es una alternativa al modelo de formularios Web Forms. El mismo posee poca complejidad y se integra perfectamente con las características existentes de ASP.NET como páginas maestras, seguridad y autenticación. (33)

PostgreSQL 9.3: es un sistema gestor de bases de datos objeto-relacional. Utiliza un modelo cliente-servidor y multiprocesos en vez de multihilos por lo que, de existir algún fallo en alguno de los procesos no se afectará el resto y el sistema continuará funcionando. Entre las principales características de PostgreSQL se encuentran: consultas complejas, disparadores, vistas actualizables e integridad transaccional. La versión 9.3 de este sistema ha sido una de las más potentes y robustas del mercado debido a su buen funcionamiento con grandes cantidades de datos y con altas concurrencias de usuarios accediendo a la vez al sistema. Esta

versión además amplía la fiabilidad de PostgreSQL, disponibilidad y capacidad de integración con otras bases de datos. El hecho de que el código esté disponible para todos hace posible que los equipos de desarrollo puedan extender o personalizar PostgreSQL sin costos adicionales. (34)

TortoiseSVN 1.8: es una herramienta para programadores que permite controlar y catalogar las diferentes versiones lanzadas de un programa durante su proceso de desarrollo. TortoiseSVN es un cliente de código abierto para el sistema de control de versiones Apache Subversion. Almacena archivos y directorios a lo largo del tiempo, los cuales son almacenados en un repositorio central. El mismo trabaja a través del explorador de Windows, integrándose en el menú contextual y dando acceso a sus opciones desde ahí. Además, crea una pequeña base de datos en la que irá guardando todas las modificaciones que sufre un mismo archivo de forma que en todo momento se pueda saber qué cambios se realizaron sobre él, cuándo, quién fue el autor de las modificaciones entre otras muchas características. (35)

Proceso Unificado Relacional (RUP): es un proceso de desarrollo de software que unido UML constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Esta metodología tiene como ventaja que es adaptable al contexto y a las necesidades de cada organización. (36)

El proceso de software definido por RUP posee tres características esenciales: está dirigido por los casos de uso, está centrado en la arquitectura y es iterativo e incremental. Los casos de uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. (36)

Que sea centrado en la arquitectura permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. RUP tiene como estrategia un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. (37)

Mediante la herramienta Enterprise Architect 7.5 y UML 2.1 se modelarán los elementos correspondientes a las fases de desarrollo de software. Este modelado permitirá un mayor entendimiento de las componentes internos del sistema así como de sus relaciones. Las diferentes fases de desarrollo del software serán las

propuestas por la metodología de desarrollo de software RUP, la cual orientará los artefactos que se deben generar en cada una de ellas. Con el uso del entorno integrado de desarrollo Visual Studio 2013 y la tecnología ASP.NET MVC 5, se desarrollarán las funcionalidades asociadas al servidor de listas de trabajo DICOM para la solución PACS-RIS. Además como gestor de bases de datos se utilizará PostgreSQL 9.3. Por último para controlar las diferentes versiones por las que transite el desarrollo de la solución se utilizará TortoiseSVN 1.8.

Al finalizar el presente capítulo en el cual se realizó, entre otros aspectos, un estudio sobre algunos sistemas que utilizan las listas de trabajo DICOM en el mundo, se pudo concluir que la utilización de los mismos no daría solución a la problemática planteada. Además, los sistemas analizados no son de código libre por lo que no pueden ser modificados. El estudio de estos sistemas, permitió concluir que es necesario el desarrollo de un componente encargado del manejo de las listas de trabajo DICOM en la solución PACS-RIS. Asimismo, en el estudio de los perfiles de integración de IHE, se determinó que el perfil de flujo de trabajo programado es el que más se ajusta al problema planteado. Se expusieron las herramientas, tecnologías y metodología a utilizar en el posterior desarrollo de las funcionalidades del servidor de listas de trabajo DICOM, teniendo en cuenta lo establecido por el proyecto de imágenes médicas perteneciente al CESIM.

CAPÍTULO 2. CARACTERÍSTICAS DEL SERVIDOR DE LISTAS DE TRABAJO DICOM

En este capítulo se presenta la propuesta del sistema a desarrollar describiendo sus principales características y su flujo de trabajo. Se muestra el Modelo del dominio en el cual se representan visualmente los objetos y clases conceptuales relacionados con el tema de investigación, con el objetivo de lograr un mayor entendimiento del contexto del mismo. Se identifican los requisitos funcionales y no funcionales a partir de los cuales se representan los casos de uso, con el objetivo de estructurar el diagrama de casos de uso.

2.1. Propuesta del servidor de listas de trabajo DICOM

Después de haber llevado a cabo una investigación sobre el manejo de las listas de trabajo DICOM, acorde con lo planteado por IHE y teniendo en cuenta las dificultades presentadas por el componente Worklist existente, se propone la solución Celsus Worklist. Esta solución permitirá el manejo de las listas de trabajo DICOM entre los sistemas PACS y RIS del CESIM.

Mediante Celsus Worklist se podrá intercambiar información entre los equipos de adquisición de imágenes médicas y la solución PACS-RIS. Está basada en el perfil propuesto por IHE: flujo de trabajo programado (SWF). Este perfil garantiza que durante todo el proceso de atención al paciente, la información se mantenga consistente y libre de errores. De igual forma SWF, especifica cómo se transferirá la información requerida a través de mensajes estandarizados entre los diferentes actores que intervengan en el proceso. La principal función de Celsus Worklist será la de actuar como intermediario entre los sistemas PACS y RIS, posibilitando la correcta asociación entre paciente e imagen.

Celsus Worklist poseerá una interfaz web la cual hará posible la interacción con el sistema. Mediante la misma, se deben registrar los componentes que se conectarán al servidor para garantizar su acceso y el correcto flujo de información. Los servicios que ofrecerá Celsus Worklist podrán ser configurados en esta interfaz, para permitir su utilización por otros sistemas que no hayan sido desarrollados por el CESIM y que posean los servicios DICOM necesarios.

Celsus Worklist deberá estar conectado con un RIS que implemente el servicio de listas de trabajo DICOM en la modalidad (MWL), el cual permitirá, a los equipos de adquisición, consultar su lista de trabajo mediante el

servicio C-FIND. Este RIS, al igual que el PACS al que se conectará, poseerá el servicio MPPS de DICOM. El servicio MPPS en el RIS y en el PACS permitirá que, a través de Celsus Worklist, estos sistemas puedan recibir las actualizaciones del estado del estudio que se está realizando hasta su culminación.

Celsus Worklist tendrá entre sus funciones permitir que los equipos de adquisición se conecten y consulten, utilizando el servicio C-FIND, su lista de trabajo para el día en cuestión. La misma incluirá los datos de los pacientes y estudios, los cuales serán proveídos por el RIS. Una vez obtenidos estos datos a través de Celsus Worklist, en el equipo de adquisición, el Especialista en imagenología dará inicio a la realización del estudio correspondiente y el estado del mismo cambiará a “*In Progress (En Progreso)*”.

El estado del estudio se irá actualizando durante su realización, para lo cual el equipo enviará mensajes MPPS a Celsus Worklist, quien a su vez se encargará de mantener actualizados al PACS y al RIS, hasta que haya cambiado a “*Complete/Discontinued (Completado o Descontinuado)*”. En la Figura 2.1 se muestra más claramente este proceso. Atendiendo a la leyenda mostrada en la figura, el tipo de flecha indica el mensaje DICOM que se utiliza en cada momento. Los mensajes N-CREATE y N-SET pertenecen al servicio MPPS.

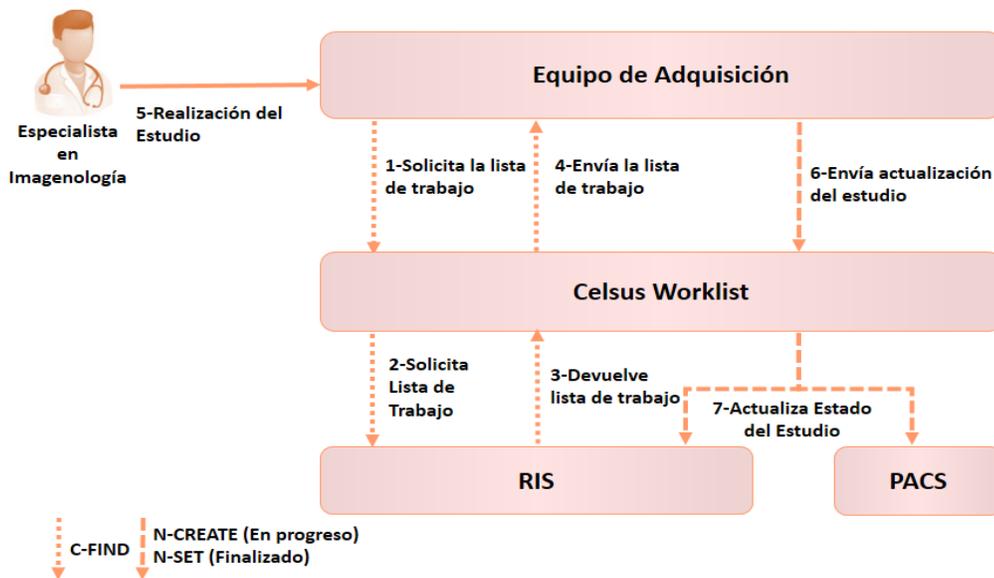


Figura 2.1 Flujo de trabajo del componente Celsus Worklist
(Fuente: elaboración propia)

El uso del componente Worklist a desarrollar, traerá ventajas a los departamentos de imagenología donde se encuentre. La utilización del mismo evitará errores en el flujo de información, ya que no se tendrán que introducir manualmente los datos del paciente en el equipo de adquisición, debido a que estos se obtendrán directamente del RIS. La solución propuesta no será dependiente del sistema RIS desarrollado por el CESIM, lo cual posibilita que pueda ser utilizado por cualquier RIS que cuente con los servicios establecidos por el estándar DICOM para el uso de las listas de trabajo en las modalidades. Además si se desea evolucionar el RIS de igual manera se podrá seguir contando con el mismo servidor de listas de trabajo.

2.2. Modelo del dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases. Pueden mostrar(27):

- Objetos del dominio o clases conceptuales
- Asociaciones entre las clases conceptuales
- Atributos de las clases conceptuales

Con el objetivo de lograr una mejor comprensión del Modelo del dominio del servidor de listas de trabajo DICOM, a continuación se describen los principales conceptos asociados al mismo.

Tabla 2.1 Entidades y conceptos del Modelo del dominio del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM

(Fuente: elaboración propia)

Entidades y conceptos	Descripción
Equipo de adquisición de imágenes médicas	Constituye el equipo a través del cual se le realiza el estudio al paciente. Es el encargado de consultar al componente alas PACSWorklist para obtener su lista de trabajo, además es el que envía al PACS las imágenes con los datos del estudio realizado.
RIS	Es el sistema encargado de la gestión de la información radiológica, contiene las listas de trabajo para los equipos de adquisición de imágenes médicas.

Entidades y conceptos	Descripción
Componente alas PACSWorklist	Es un componente que pertenece al RIS. El mismo cuenta con un servicio básico que permite a los equipos consultar su lista de trabajo.
PACS	Es el sistema que se encarga del almacenamiento, la visualización y la transmisión de las imágenes médicas que se obtienen en los equipos de adquisición, producto de los estudios realizados a los pacientes.

Diagrama del modelo del dominio

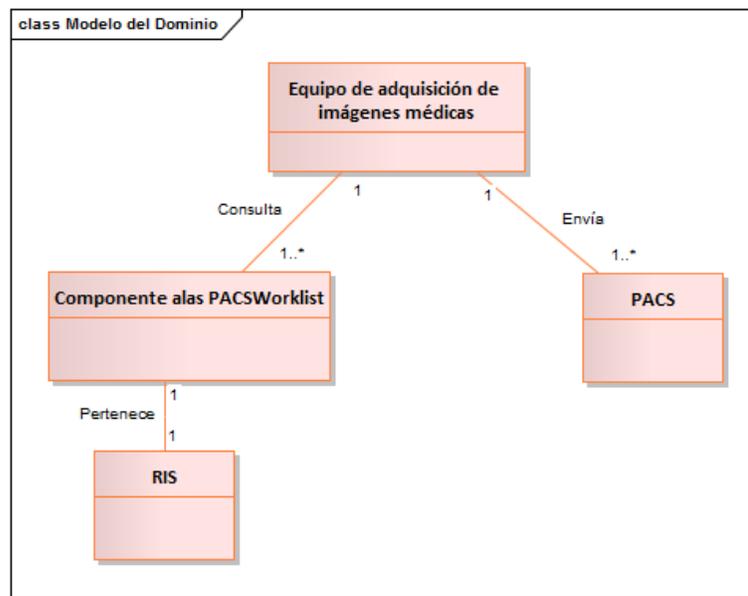


Figura 2.2 Diagrama del Modelo del dominio del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM
(Fuente: elaboración propia)

2.3. Especificación de los requisitos del software

Los requerimientos o requisitos de un sistema son la descripción de los servicios proporcionados por este y sus restricciones operativas. Estos requerimientos reflejan las necesidades de un cliente que hará uso del

sistema en cuestión. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le denomina ingeniería de requisitos. (38)

2.3.1. Requisitos funcionales

Los requisitos funcionales describen la interacción entre el sistema y su ambiente independientemente de su implementación. Estos requisitos dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactarlos. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en determinadas situaciones. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (38) Los requisitos funcionales del servidor de listas de trabajo a desarrollar, se muestran en la tabla 2.2.

Tabla 2.2 Requisitos funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM (Fuente: elaboración propia)

Número	Requisito	Descripción
RF 1	Iniciar servidor.	Permite iniciar el servidor.
RF 2	Detener servidor.	Permite detener el servidor.
RF 3	Solicitar lista de trabajo.	Permite solicitar la lista de trabajo del equipo correspondiente.
RF 4	Actualizar estado del estudio.	Permite actualizar en el RIS y en el PACS el estado del estudio.
RF 5.1	Adicionar nodo DICOM.	Permite adicionar un nodo DICOM que se conectará al servidor ya sea un RIS o un PACS.
RF 5.2	Eliminar nodos DICOM.	Permite eliminar un nodo DICOM que se encuentra conectado al servidor.
RF 5.3	Modificar nodo DICOM.	Permite modificar la conexión de un nodo DICOM.

Número	Requisito	Descripción
RF 6.1	Mostrar configuración del servidor local.	Permite mostrar los valores de la conexión del servidor local.
RF 6.2	Editar configuración del servidor local.	Permite editar los valores de la conexión del servidor local.
RF 7	Mostrar log del sistema.	Permite mostrar los log del sistema.
RF 8.1	Adicionar modalidad.	Permite adicionar una modalidad que se conectará al servidor.
RF 8.2	Modificar modalidad.	Permite modificar la modalidad que se conectará al servidor.
RF 8.3	Eliminar modalidad.	Permite eliminar una modalidad seleccionada.

Para una mejor comprensión de los requisitos funcionales, se agrupan los mismos por paquetes. Ver Figura 2.2.

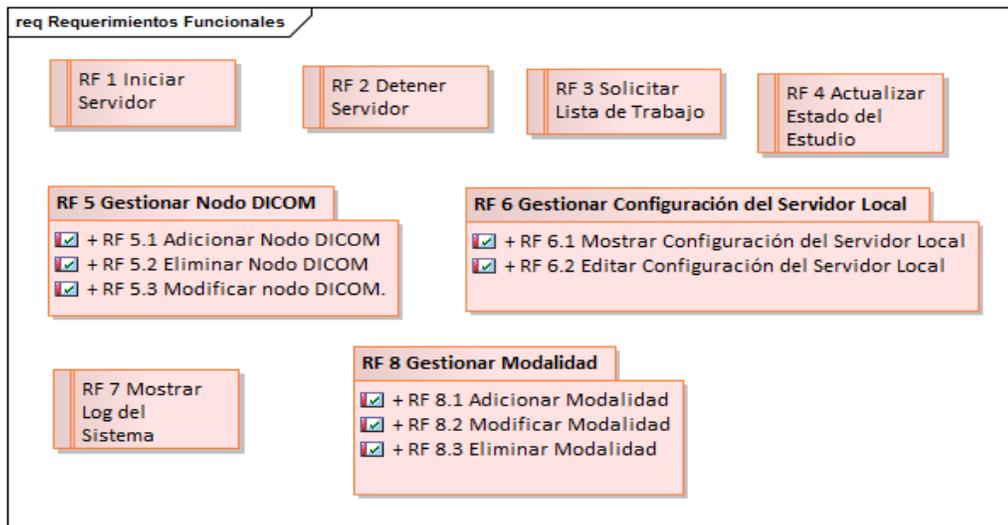


Figura 2.3 Requisitos funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM (fuente: elaboración propia)

2.3.2. Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente se aplican a características o servicios individuales del sistema. Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.(39) Los requisitos no funcionales del servidor de listas de trabajo a desarrollar, se muestran en la tabla 2.3.

Tabla 2.3 Requisitos no funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM (fuente: elaboración propia)

Número	Requisito	Descripción
RNDI 1	Lenguaje de programación C#.	Para el desarrollo de las funcionalidades se utilizará C# como lenguaje de programación, pues está diseñado y optimizado para la plataforma .NET.
RNDI 2	Entorno integrado de desarrollo Visual Studio 2013.	Se utilizará Microsoft Visual Studio 2013 como entorno integrado de desarrollo. Este soporta C# como lenguaje de programación.
RNDI 3	Uso de Enterprise Architect 7.5.	Se utilizará el Enterprise Architect en su versión 7.5 como herramienta de modelado.
RNDI 4	Uso PostgreSQL 9.3.	Se utilizará como gestor de bases de datos PostgreSQL en su versión 9.3.
RNDI 5	Uso del Framework.Net 4.5.	El lenguaje de programación C# depende de este Framework y además presenta mejoras en cuanto a administración y rendimiento.

Número	Requisito	Descripción
RNDI 6	Uso de UML 2.1.	Se utilizará UML como lenguaje de modelado en su versión 2.1.
RNDI 7	Uso de Tortoise SVN 1.8.	Se utilizará la herramienta TortoiseSVN en su versión 1.8 para el control de versiones, la cual puede integrarse al entorno integrado de desarrollo seleccionado.
RNDI 8	Uso de la metodología Rational Unified Process (RUP).	El proceso de desarrollo del servidor de listas de trabajo será guiado por la metodología RUP.
RNU 1	Facilidad de empleo para los usuarios de la aplicación.	El sistema debe contar con una interfaz diseñada de forma que se facilite el empleo de la misma por los usuarios que accedan.
RNU 2	Debe soportar el idioma inglés además del español.	El sistema debe ser capaz de cambiar de idioma según lo necesite el usuario.
RNFO 1	Memoria RAM de 4gb o superior.	La aplicación necesita al menos 4gb de memoria RAM para su correcto funcionamiento.
RNL 1	El componente y la documentación son propiedad del proyecto de imágenes médicas del CESIM.	El componente y la documentación generada durante el desarrollo del software son propiedad del proyecto de imágenes médicas del CESIM.
RNS 1	Autenticación de usuarios.	El componente deberá implementar un mecanismo de seguridad para la autenticación de los usuarios permitidos.

En la Figura 2.3 se muestran los requisitos no funcionales agrupados por paquetes.

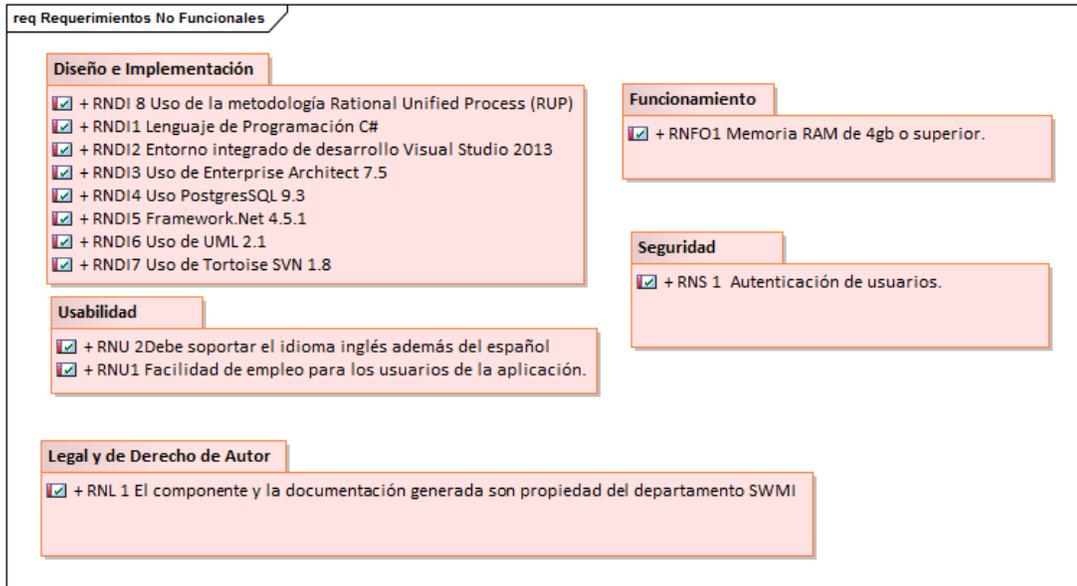


Figura 2.4 Requisitos no funcionales del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM (fuente: elaboración propia)

2.4. Modelo de casos de uso del sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Estos diagramas están compuestos por actores que representan un tipo de usuario del sistema, entiéndase por usuario cualquier cosa externa que interactúa con el mismo. Además estos diagramas están compuestos por los casos de uso, los cuales son tareas que deben llevarse a cabo con el apoyo del sistema que se está desarrollando. (38, 40)

2.4.1. Definición de los actores del sistema

Los actores asociados a la presente investigación se muestran en la Tabla 2.4.

Tabla 2.4 Definición de los actores referentes al servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM

(Fuente: elaboración propia)

Actor	Justificación
 Administrador	Representa el usuario que posee los permisos para trabajar con el servidor.
 Cliente	Representa al especialista que se encargará de realizarle el estudio al paciente en el equipo de adquisición de imágenes médicas. También puede representar a un equipo de adquisición de imágenes médicas que automáticamente es capaz de realizar dichas acciones.

2.4.2. Diagrama de casos de uso del sistema

En la Figura 2.5 se muestra el Diagrama de casos de uso del sistema, referente al servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM. Este diagrama representa la interacción de los usuarios con el sistema.

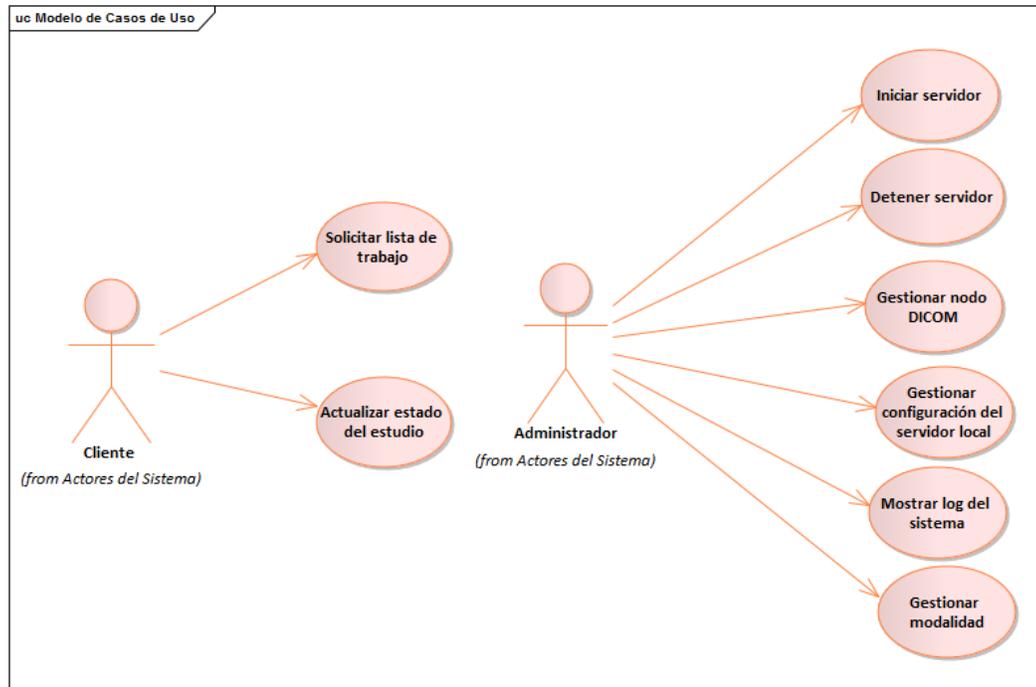


Figura 2.5 Diagrama de casos de uso del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM (fuente: elaboración propia)

2.4.3. Especificación de los casos de uso del sistema

A continuación se muestra la descripción del caso de uso Gestionar modalidad, el resto de las descripciones se pueden encontrar en el documento CESIM_PACS-RIS_010114a_Especificacion_de_casos_de_uso perteneciente al expediente del proyecto.

Tabla 2.5 Descripción del caso de uso Gestionar modalidad

(Fuente: elaboración propia)

Objetivo	El objetivo de este caso de uso es adicionar, eliminar o editar una modalidad.
Actores	Administrador (Inicia).
Resumen	El caso de uso inicia cuando el Administrador selecciona la opción “Modalidades Autorizadas” en el menú superior de la aplicación. El sistema muestra un listado con

	las modalidades autorizadas existentes. Brinda además, las opciones de adicionar una nueva modalidad, editar alguna existente o eliminarla.
Referencias	RF 8.1, RF 8.2, RF 8.3.
Complejidad	Baja.
Prioridad	Alta.
Precondiciones	No procede.
Postcondiciones	Se editó una modalidad. Se adicionó una modalidad. Se eliminó una modalidad.
Flujo de eventos	
Flujo básico Gestionar Modalidad	
<ol style="list-style-type: none"> 1. El Administrador selecciona la opción “Modalidades Autorizadas” en el menú superior de la aplicación. 2. El Sistema muestra la página solicitada con las opciones correspondientes. 3. El Administrador selecciona la operación que desee realizar. <ul style="list-style-type: none"> - Si selecciona la opción “Adicionar Modalidad”, ir a la Sección 1: “Adicionar modalidad”. - Si selecciona la opción “Editar Modalidad”, ir a la Sección 2: “Editar modalidad”. - Si selecciona la opción “Eliminar Modalidad”, ir a la Sección 3: “Eliminar modalidad”. 4. Termina el caso de uso. 	
Flujos alternos	
No aplica.	
Sección 1: “Adicionar modalidad”	
<ol style="list-style-type: none"> 1. El Sistema muestra en la vista “Adicionar Modalidad” los campos para introducir los datos para crear la nueva modalidad. <ul style="list-style-type: none"> - Called AE Title. 	

- IpAddres.

2. El **Administrador** introduce los datos de la nueva modalidad.
3. El **Administrador** selecciona la opción "Aceptar".
4. El **Sistema** valida los datos introducidos.
5. El **Sistema** registra los nuevos datos introducidos.
6. El **Sistema** muestra el mensaje: " La modalidad se ha añadido a la lista de modalidades".

Flujos alternos

4a Datos incorrectos

1. El **Sistema** muestra el mensaje: " Los datos son incorrectos".
2. Regresar al paso 2 del flujo básico.

4b Campos vacíos

1. El **Sistema** resalta en rojo los campos vacíos y muestra el mensaje: "Los campos resaltados son obligatorios".
2. Regresar al paso 2 del flujo básico.

Sección 2: "Editar modalidad"

1. El **Administrador** selecciona de la página "Modalidades Autorizadas", en la tabla donde se listan los datos de las modalidades, la opción "Editar" para modificar una modalidad.
2. El **Sistema** muestra los datos actuales de la modalidad seleccionada en la vista "Modificar Modalidad".
3. El **Administrador** modifica los valores que desea.
4. El **Administrador** selecciona la opción "Aceptar".
5. El **Sistema** valida los datos introducidos.
6. El **Sistema** guarda los nuevos datos introducidos.

7. El **Sistema** muestra el mensaje: " Se ha actualizado exitosamente".

Flujos alternos

4a Datos incorrectos

1. El **Sistema** muestra el mensaje: "Los datos son incorrectos".
2. Regresar al paso 3 del flujo básico.

4b Campos Vacíos

1. El **Sistema** resalta en rojo los campos vacíos y muestra el mensaje: "Los campos resaltados son obligatorios".
2. Regresar al paso 3 del flujo básico.

Sección3: "Eliminar modalidad"

1. El **Administrador** selecciona de la página "Modalidades Autorizadas", en la tabla donde se listan los datos de las modalidades, la opción "Eliminar" para eliminar una modalidad.
2. El **Sistema** muestra la vista "Eliminar Modalidad" con las opciones correspondientes.
3. El **Administrador** selecciona la opción "Eliminar".
4. El **Sistema** muestra el mensaje "¿Está seguro que desea eliminar la modalidad seleccionada?".
5. El **Administrador** selecciona la opción "Aceptar".
6. El **Sistema** elimina la modalidad seleccionada.
7. El **Sistema** muestra el mensaje: " Se ha eliminado exitosamente de la lista".

3a Selecciona la opción "Cancelar"

1. El **Sistema** muestra el listado de las modalidades conectadas.
2. Regresar al paso 2 del flujo básico.

Relaciones	Casos de Uso Incluidos	No procede.
	Casos de Uso Extendidos	No procede.
Asuntos pendientes	No procede.	

A partir de la investigación realizada se pudo determinar la propuesta de solución para la problemática descrita. Teniendo en cuenta el análisis del flujo actual de información existente en los sistemas PACS-RIS del CESIM, se obtuvo el Modelo del dominio del servidor de listas de trabajo DICOM. En este modelo se representaron los conceptos encontrados y las relaciones definidas entre ellos. El mismo permitió determinar las características que debe cumplir el servidor de listas de trabajo DICOM. Las funcionalidades del componente Celsus Worklist fueron obtenidas a partir de la descripción de los casos de uso del sistema.

CAPÍTULO 3. DISEÑO DEL SERVIDOR DE LISTAS DE TRABAJO DICOM

En este capítulo se describe la arquitectura y los patrones de diseño que se utilizarán en el desarrollo del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM. Se realiza el diseño de la solución mediante los diagramas de clases del diseño y la descripción de las clases que los conforman. Además se muestran los diagramas de secuencia de los casos de uso.

3.1. Descripción de la arquitectura. Fundamentación

La arquitectura de software se refiere a la estructura de un sistema, la cual está compuesta de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. Se necesita una arquitectura de software para organizar el desarrollo, fomentar la reutilización, comprender y hacer evolucionar el sistema. La descripción de la arquitectura está constituida por la descripción de las partes del sistema que es importante que comprendan todos los desarrolladores para que éstos puedan realizar cambios y correcciones al software sin romper con la misma. (36)

Teniendo en cuenta las características del componente que se desea desarrollar y los servicios que brindará, se definieron dos estilos arquitectónicos a utilizar. La estructura lógica del componente estará representada por una arquitectura n-capas, mientras que la estructura física por la arquitectura cliente-servidor.

Arquitectura cliente-servidor: es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. (41)

Celsus Worklist es un componente que actuará como servidor de listas de trabajo DICOM. Al mismo se conectarán equipos de adquisición clientes que accederán a los servicios DICOM que este proveerá. Esto permitirá que dichos equipos, puedan acceder a las listas de trabajo que tienen asignadas y además que puedan enviar actualizaciones del estado de los estudios, a través de Celsus Worklist, hacia los sistemas RIS y PACS. De igual forma, este sistema contará con una interfaz web que permite el acceso al sistema de los usuarios de la aplicación.

Arquitectura n-capas: propone separar los componentes de la aplicación según sus funciones. La programación por capas se refiere a un estilo de programación, que tiene como objetivo separar la lógica de

diseño de la lógica de negocios. Una de las ventajas que tiene este estilo es que el desarrollo del software se puede llevar a cabo en varios tipos de niveles, de esta forma un cambio en algún nivel no afectará los demás o las afectaciones de estos serán mínimas. Además ayuda a diferenciar entre los diferentes tipos de tareas que deben realizar los elementos que conforman el componente, ofreciendo un diseño que maximiza la reutilización y provee mejoras en cuanto al mantenimiento del código. (42) A continuación se describen las capas por las que estará compuesta la aplicación.

Capa de presentación: en esta capa se presentan los resultados de las operaciones a los usuarios y se valida la entrada y salida de datos. Envía la información que obtiene a la capa de dominio para su procesamiento. La comunicación de esta capa se realiza con la capa inmediata inferior, la de dominio.(43)

La capa de presentación estará conformada por las vistas y las clases controladoras. Las diferentes vistas muestran las opciones de iniciar o detener los servicios prestados por el componente, adicionar las modalidades se conectarán al sistema o editar, en una nueva ventana, algunas de las que se muestran en el listado de las modalidades existentes. De igual forma, cuenta con la opción de adicionar los nodos DICOM que se conectarán al servidor o editar la conexión de alguno de los existentes.

Capa del dominio: esta capa reúne los aspectos del software que se deben automatizar o apoyan a los procesos que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y de las restricciones que se aplican al sistema.(43) Es la encargada de comunicar la capa de datos con la capa de presentación. En esta capa es donde se realiza la gestión de los equipos de adquisición y de los nodos DICOM que se conectarán al servidor. Contiene las entidades del dominio.

Capa de servicios: se encuentran los servicios definidos en el estándar DICOM para el intercambio de información con los sistemas que se conectarán, estos servicios serán consumidos por el sistema propuesto que además los proveerá. Esta capa se abstrae de toda la lógica del sistema y realiza las acciones encargadas de controlar los servicios DICOM. Estos servicios serán los que se consumen durante los procesos de dar respuesta a las solicitudes de listas de trabajo, que realizan los equipos de adquisición y de actualizar el estado de los estudios en el RIS y en el PACS.

Capa de acceso a datos: será la encargada de cargar, eliminar, modificar y persistir la información registrada. En esta capa se registran los parámetros de configuración del servidor así como los sistemas con los que se

conectará. Esto es posible mediante el uso de NHibernate un framework de persistencia de datos, que abstrae al programador del gestor de bases de datos utilizado y la persistencia de las entidades del sistema obtenidas mediante el proceso de mapeo.

Capa de infraestructuras transversales: esta capa contiene funcionalidades comunes que se utilizan en las diferentes capas que componen la estructura lógica de la aplicación. (43) En esta capa se realiza el registro de los log que se generan durante los diferentes eventos del sistema. Además contiene los aspectos propios de la configuración del componente Celsus Worklist.

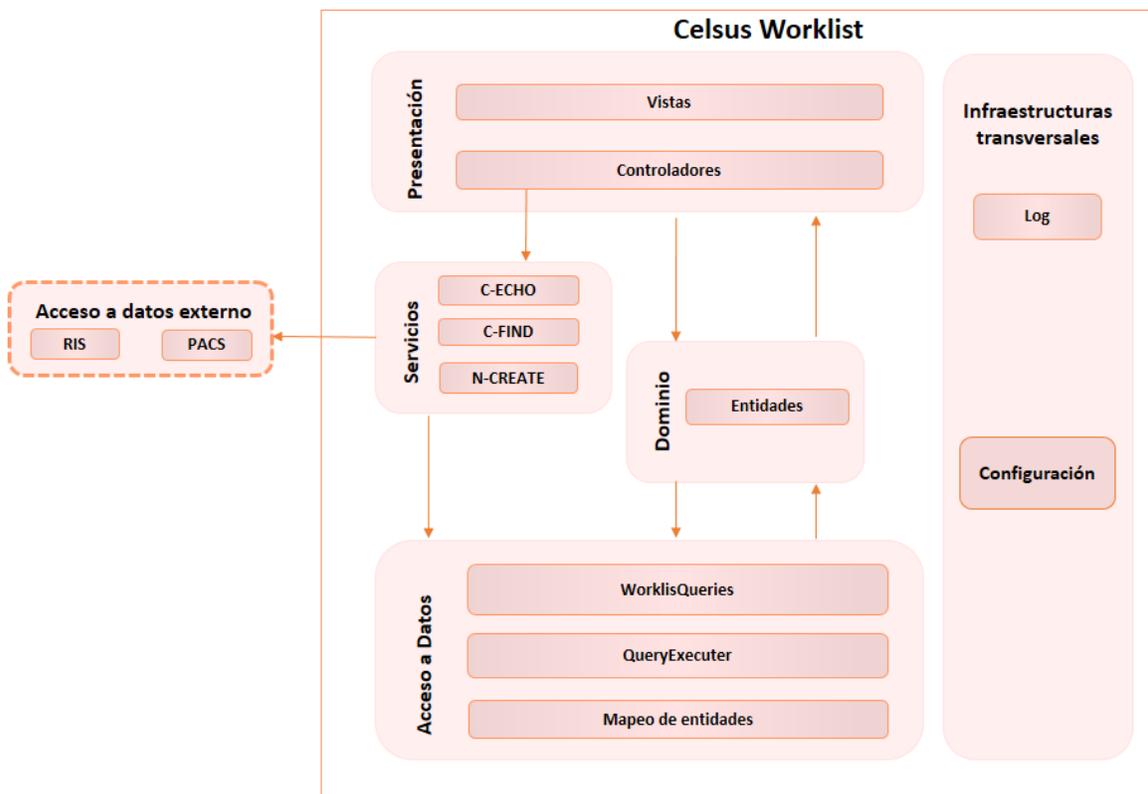


Figura 3.1 Arquitectura basada en los estilos Cliente-Servidor y N-Capas del componente Celsus Worklist (Fuente: elaboración propia)

3.2. Patrón arquitectónico

Los patrones arquitectónicos capturan la experiencia comprobada en el desarrollo del software y ayudan a promover buenas prácticas de diseño. Teniendo en cuenta las herramientas y tecnologías utilizadas en el desarrollo de las funcionalidades asociadas al componente Celsus Worklist, se definió la utilización del patrón arquitectónico Modelo-Vista-Controlador (MVC). Este patrón de arquitectura de software separa los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

El patrón arquitectónico MVC separa la aplicación en tres componentes principales (33):

- **Modelo:** representa las reglas de negocio de la aplicación. Se encarga de mantener la persistencia de los datos, guardando o recuperando la información independiente de medio utilizado (ficheros XML, bases de datos entre otros).
- **Vista:** representa los componentes que muestran la interfaz de la aplicación, mostrando la información obtenida a partir del modelo, de manera que el usuario pueda visualizarla. Básicamente las vistas contienen el código de presentación que se va a enviar al navegador.
- **Controlador:** representa los componentes que se encargan de la interacción con el usuario, actuando de intermediario entre el usuario, el modelo y la vista. El controlador recoge las peticiones del usuario, interactúa con el modelo y finalmente selecciona que vista es la adecuada para mostrar los datos en cuestión.

Como se puede observar en la Figura 3.1, el controlador recibe el evento proveniente de la vista y lo envía al modelo para su procesamiento, que en este caso sería la capa de acceso a datos, pasando antes por la capa de dominio. Una vez procesados los datos, en esta capa, se envían a la vista para presentar el resultado de la operación al usuario. Este patrón arquitectónico se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

3.3. Diseño

El diseño es la etapa de la ingeniería de software que consiste en producir un modelo o representación técnica del software que se va a desarrollar. El diseño de software se encuentra en el núcleo técnico de la ingeniería y se aplica de manera independiente al modelo que se utilice. Una vez que se analizan y describen los requisitos, el diseño de software es la última acción de la ingeniería correspondiente dentro de la actividad de modelado, la cual establece una plataforma para la generación de código y pruebas. (40)

El diseño tiene como finalidad transformar los requisitos obtenidos en un diseño que sea capaz de representar cómo será el funcionamiento del sistema propuesto. En esta etapa se genera el artefacto Modelo del Diseño. En el [Anexo 1](#) se puede encontrar el diagrama de clases del diseño del caso de uso Gestionar Modalidad y la descripción de las clases del mismo en el [Anexo 2](#). Cada una de las clases del diseño describe un elemento del dominio del problema, con enfoque en los aspectos visibles para el usuario o cliente.

Además se realizaron los diagramas de secuencia para mostrar el flujo de información entre los diferentes elementos del componente. En el [Anexo 3](#) se puede encontrar el diagrama de secuencia del caso de uso Gestionar modalidad. Los diagramas de clases del diseño y de secuencia del resto de los casos de uso, se pueden encontrar en el documento CESIM_PACS-RIS_010215_Modelo_de_diseño perteneciente al expediente del proyecto.

3.3.1. Patrones de diseño

Un patrón de diseño describe una estructura que resuelve un problema de diseño particular dentro de un contexto específico. (40) En otras palabras, los patrones de diseño son una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Para el desarrollo de las funcionalidades del servidor de listas de trabajo se definieron los siguientes patrones:

Singleton: garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. El patrón de diseño Singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. (44) En el sistema propuesto este patrón es utilizado para la creación de una instancia única de la clase “DbFactory”. Esta clase es la que garantiza la conexión a la base de datos, realiza el mapeo de las entidades y crea las sesiones de “NHibernate”.

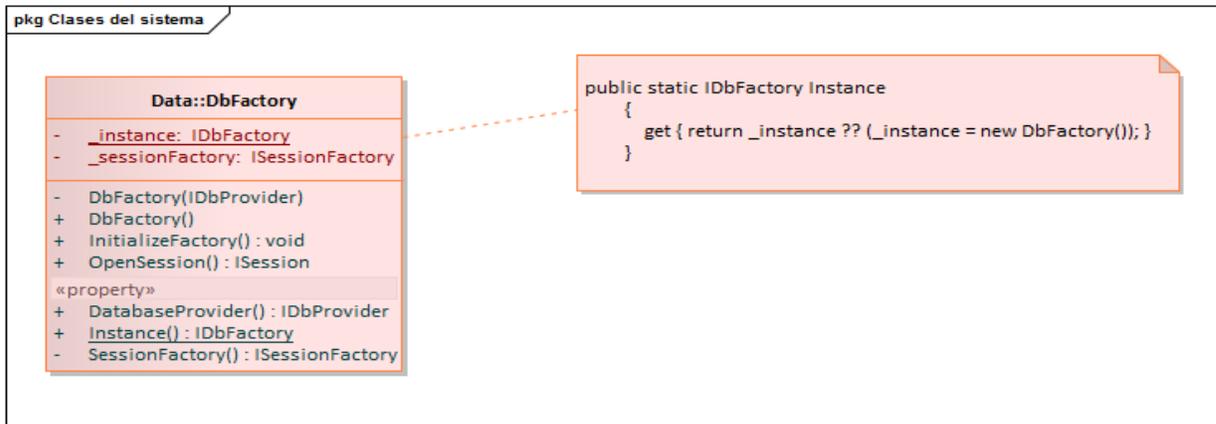


Figura 3.2 Patrón Singleton utilizado en el desarrollo del componente Celsus Worklist
(Fuente: elaboración propia)

Los patrones para asignar responsabilidades, GRASP²² son utilizados para describir los principios fundamentales de la asignación de responsabilidades a objetos. (45) Estos patrones consisten en elegir clases determinadas y decidir cómo estas deben interactuar.

Controlador: es utilizado para controlar el flujo del sistema mediante clases específicas. Este patrón se presenta ante el problema de asignar un responsable ante un evento de entrada al sistema, el cual es generado por un actor del mismo. (45). El patrón controlador actúa como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que esta reciba los datos del usuario y los envíe a las distintas clases según el método llamado.

En el sistema propuesto se crearon las clases controladoras “AllowedModalitiesController”, “ConfigurationController”, “LogController”, “RemoteScpController” y “ServiceController” que permiten separar la capa del dominio de la de presentación. Estas clases serán las encargadas de gestionar las acciones que ocurran durante los eventos generados por los actores externos al sistema. Un ejemplo de la aplicación de este patrón se evidencia cuando el administrador necesita adicionar una modalidad al servidor. La clase

²² General Responsibility Assignment Software Patterns

“AllowedModalitiesController” se encargará de recoger la información de la nueva modalidad, obtenida en la vista y enviarla a la clase “WorklistManager” para su procesamiento. Ver Figura 3.3.

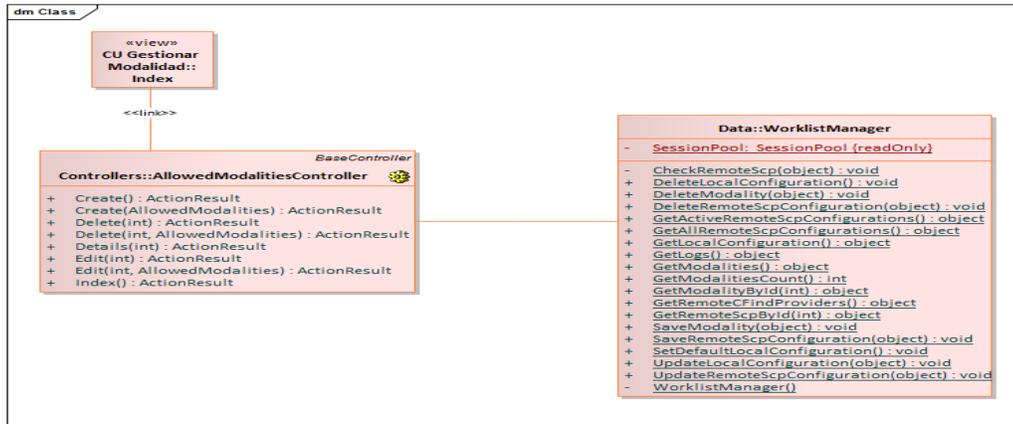


Figura 3.3 Patrón Controlador utilizado en el desarrollo del componente Celsus Worklist (Fuente: elaboración propia)

Fabricación Pura: este patrón brinda una solución al problema de asignar un conjunto de responsabilidades a una clase que no representa nada en el dominio del problema. El uso de este patrón trae ventajas como la existencia de una mayor cohesión, ya que se tienen clases dedicadas a una responsabilidad en particular y se baja el acoplamiento lo que permite reutilizar y mantener las unidades de código. (45)

Ante el problema de realizar el manejo de los log que se generarán durante los diferentes eventos que ocurrirán en el sistema, se creó la clase “NLogLogger”. Esta clase es la responsable de inicializar los log que registrarán los sucesos del sistema. Además se encarga de almacenar los mismos en la base de datos de la aplicación.

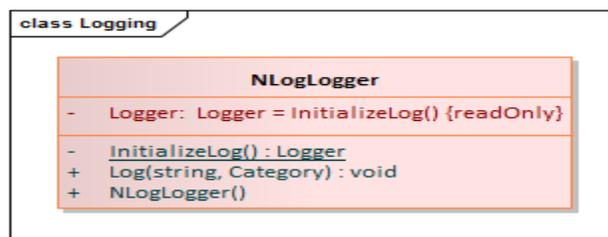


Figura 3.4 Patrón Fabricación Pura utilizado en el desarrollo del componente Celsus Worklist (Fuente: elaboración propia)

Teniendo en cuenta las características del componente a desarrollar y los servicios que brindará se definió que la arquitectura estará basada en los estilos arquitectónicos cliente-servidor y n-capas. A partir de los diagramas de clases del diseño y de secuencia se obtuvo una mejor visión sobre los elementos del sistema propuesto y las relaciones entre ellos. Esto permite obtener elementos que se utilizarán en la fase de la implementación. El diseño estará basado en el patrón Singleton. Además, para la asignación de responsabilidades se determinó el uso de los patrones Controlador y Fabricación pura.

CAPÍTULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DEL SERVIDOR DE LISTAS DE TRABAJO DICOM

En el presente capítulo se realiza una descripción de los elementos que corresponden a la fase de implementación. Se describe el modelo de datos en el que se muestra la estructura que almacena la información requerida por el sistema. Se muestra el diagrama de despliegue y el de componentes así como los elementos que conforman los mismos y las relaciones entre ellos. Se describen las estrategias de codificación y además cómo se llevará a cabo el tratamiento de errores en el sistema. De igual forma se realiza la validación del sistema implementado, permitiendo asegurar que los resultados generados fueron los que se esperaban.

4.1. Modelo de datos

Una parte importante del modelado del sistema, es la definición de la forma lógica que contendrá los datos procesados por este. Esto es posible mediante los denominados modelos de datos. Estos permiten a los ingenieros de software identificar objetos de datos y sus relaciones mediante una notación gráfica. Este modelo hace uso de los diagramas de entidad relación. (38)

En el contexto del análisis estructurado, este diagrama entidad relación define todos los datos que se introducen, se almacenan, se transforman y se producen dentro de una aplicación. El modelo de datos estudia estos, independientemente del proceso que los transforma. El mismo se compone de tres piezas de información: el objeto de datos, los atributos que describen el modelo de datos y la relación que conecta objetos de datos entre sí. (38)

El objeto de datos o también llamados entidades, constituye una representación de cualquier composición de información compuesta que deba comprender el software. Las entidades están compuestas de atributos. El último elemento del modelo de datos lo constituyen las relaciones entre las entidades. Estas relaciones no poseen existencia propia en el mundo real que se está modelando, pero son necesarias para reflejar las interacciones existentes entre las entidades. (46) En la figura 4.1 se muestra el modelo de datos del componente Celsus Worklist.

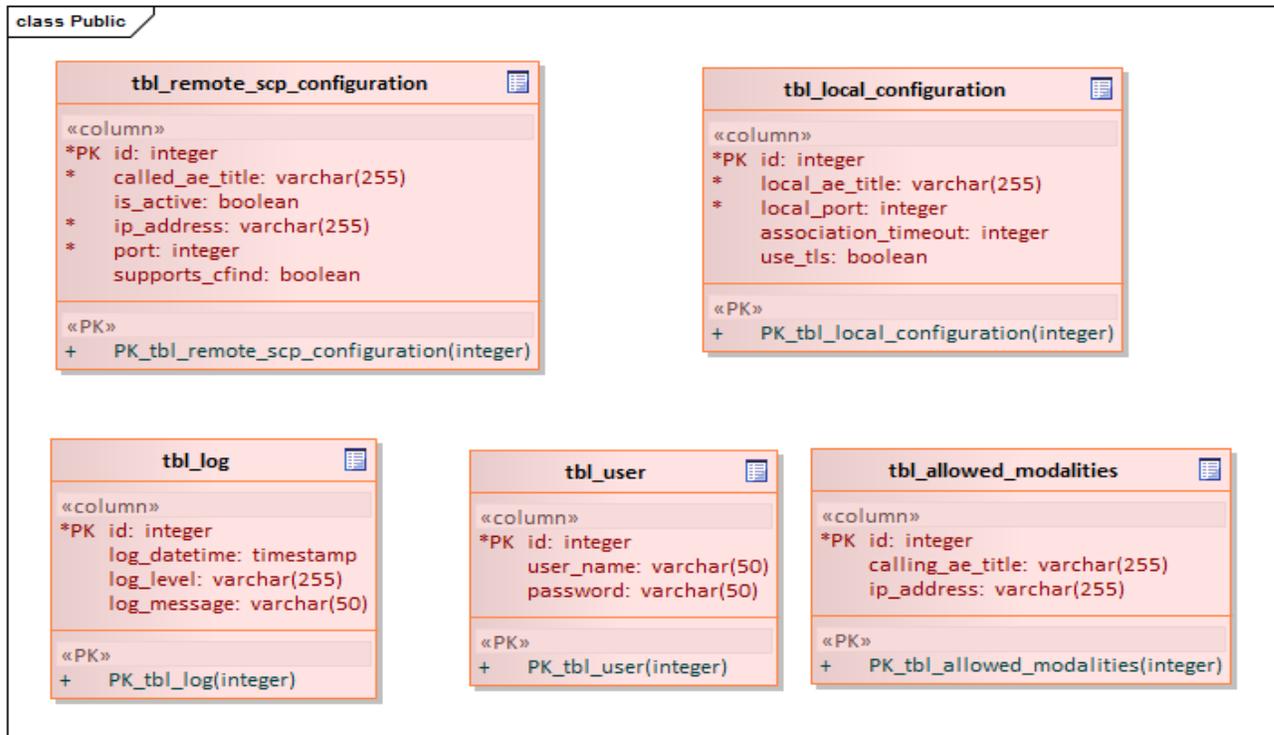


Figura 4.1 Modelo de datos del componente Celsus Worklist

(Fuente: elaboración propia)

4.1.1. Descripción de las tablas de la base de datos

A continuación se presentan la descripción de la tabla “tbl_allowed_modalities”. El resto de las descripciones puede encontrarse en el [Anexo 4](#).

Tabla 4.1 Descripción de la tabla “tbl_allowed_modalities” perteneciente a la base de datos del componente Celsus Worklist

(Fuente: elaboración propia)

Nombre: tbl_allowed_modalities		
Descripción: Esta tabla será la encargada de almacenar los datos de las modalidades que se conecten al servidor de listas de trabajo DICOM.		
Atributo	Tipo	Descripción
Id	Integer	Identificador de la tabla.

Nombre: tbl_allowed_modalities		
called_ae_title	Varchar	Nombre que identifica a una aplicación de red DICOM. En este caso la modalidad.
ip_address	Varchar	Dirección IP de la modalidad.

4.2. Implementación

Las vistas físicas de un sistema modelan la estructura de la implementación de la aplicación por sí misma, su organización en componentes, y su despliegue en nodos de ejecución. Existen dos vistas físicas: la vista de implementación y la vista de despliegue.

4.2.1. Diagrama de despliegue

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos. Un nodo es un recurso de ejecución, tal como una computadora. Esta vista, permite determinar las consecuencias de la distribución y de la asignación de recursos. La vista de despliegue se representa en diagramas de despliegue. (28)

En la Figura 4.2 se muestra el diagrama de despliegue del servidor de listas de trabajo DICOM para la solución PACS-RIS del CESIM. Para una mayor descripción de los elementos del mismo Ver Tabla 4.2.

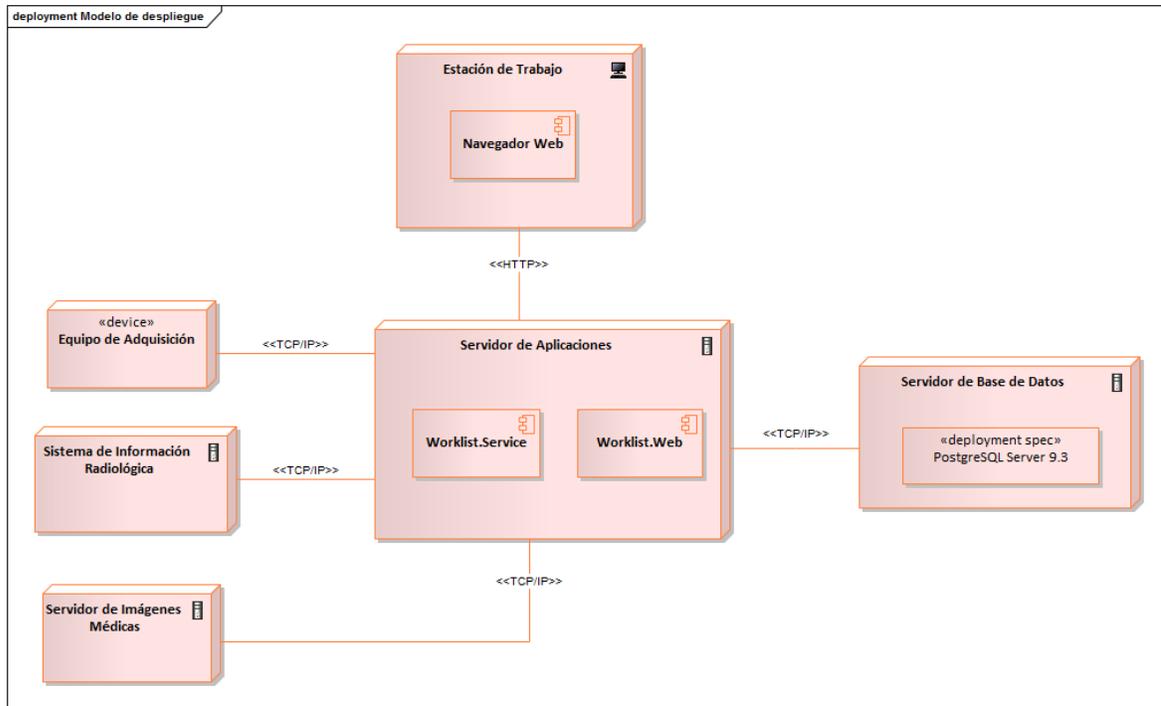


Figura 4.2 Diagrama de despliegue del componente Celsus Worklist
(Fuente: elaboración propia)

Tabla 4.2 Descripción de los elementos del diagrama de despliegue del componente Celsus Worklist
(Fuente: elaboración propia)

Nodo	Descripción
Estación de trabajo	Estación que permite la configuración del componente Celsus Worklist mediante una interfaz Web.
Servidor de aplicaciones	Proporciona los servicios que permiten la ejecución y disponibilidad de la aplicación desplegada.
Servidor de base de datos	Almacena los datos que registrará el componente Celsus Worklist.
Servidor de imágenes médicas	Almacena las imágenes y estudios que se realizan en los equipos de adquisición.

Nodo	Descripción
Sistema de información radiológica	Almacena la información referente a los pacientes y las citas.
Equipo de adquisición	Adquiere las imágenes médicas que se obtienen en la realización de un estudio imagenológico a los pacientes.

4.2.2. Diagrama de componentes

La vista de implementación modela los componentes de un sistema a partir de los cuales se construye la aplicación. También modela la asignación de clases y de otros elementos del modelo a los componentes. La vista de implementación se representa en diagramas de componentes.

La vista de componentes muestra un conjunto de componentes disponibles (librería) con sus dependencias, este es el material a partir del cual se puede ensamblar un sistema. De esta forma, cada componente se conecta a otros componentes cuyos servicios utiliza, estas conexiones deben ser consistentes con las interfaces de los componentes. (28) En la Figura 4.2 se muestra el Diagrama de componentes de Celsus Worklist.

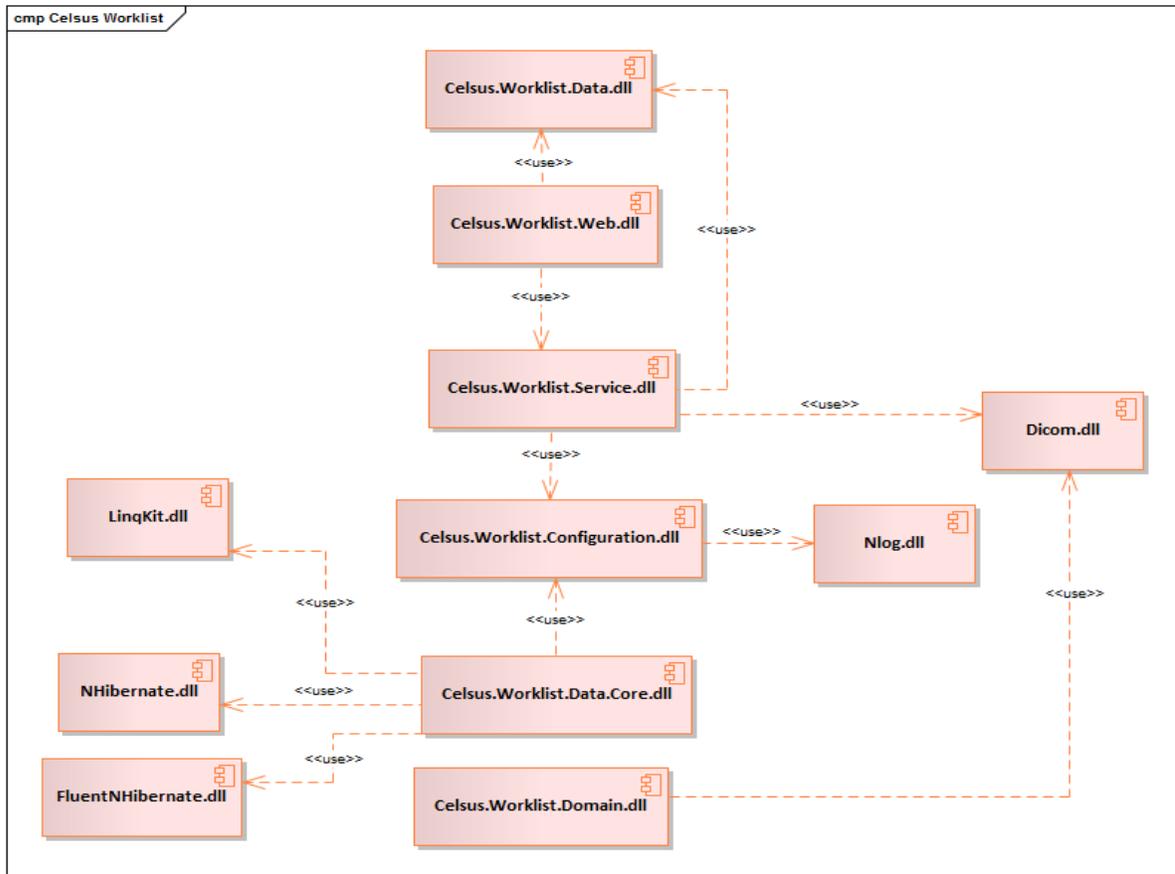


Figura 4.3 Diagrama de componentes de Celsus Worklist

(Fuente: elaboración propia)

En la siguiente tabla se muestra la descripción de cada uno de los elementos del diagrama de componentes.

Tabla 4.3 Descripción de los elementos que conforman el diagrama de componentes del servidor de listas de trabajo DICOM

(Fuente: elaboración propia)

Componente	Descripción
Celsus.Worklist.Configuration.dll	Componente que se encarga de la configuración del sistema, así como de los logs del mismo.

Componente	Descripción
Celsus.Worklist.Data.Core.dll	Componente que ejecuta las consultas a la base de datos y se encarga de manejar las sesiones en la misma.
Celsus.Worklist.Data.dll	Componente que permite la conexión con la base de datos.
Celsus.Worklist.Domain.dll	Se encarga de las interfaces de las entidades, de los log y de los mensajes C-FIND.
Celsus.Worklist.Service.dll	Componente que contiene las clases que controlan los servicios DICOM en la aplicación.
Celsus.Worklist.Web.dll	Componente que contiene todas las vistas y las clases controladoras de la aplicación.
DICOM.dll	Librería que garantiza el manejo de los servicios DICOM y la transmisión de los datos en este formato.
FluentNHibernate.dll	Librería que se utiliza para crear y gestionar los mapeos de entidades entre la aplicación y la base de datos
LinqKit.dll	Librería que permite implementar consultas SQL a la base de datos desde la aplicación.
NHibernate.dll	Librería que se encarga del manejo de las sesiones y la comunicación con la base de datos, garantizando el mapeo objeto-relacional (ORM).
Nlog.dll	Librería que permite registrar los logs del sistema.

4.3. Tratamiento de Errores

La realización de un correcto tratamiento de errores, permite preparar al software ante situaciones para las cuales no ha sido diseñado. Las características de control de excepciones del lenguaje C#, proporcionan una manera de afrontar cualquier situación inesperada o excepcional que se presente mientras se ejecuta un

programa. El control de excepciones utiliza las palabras clave **try catch** para intentar realizar acciones que podrían plantear problemas y controlar errores cuando considere que sea razonable.(47)

En el desarrollo de las funcionalidades asociadas al componente Celsus Worklist, se utilizó el manejo de excepciones mediante los bloques **try catch**. Estos fueron utilizados en las clases controladoras para realizar un tratamiento de errores ante la entrada de los datos de los usuarios. A continuación se muestra un fragmento de código perteneciente a la clase “AllowedModalitiesController”, en la misma se puede observar lo anteriormente descrito.

```
public ActionResult Create(AllowedModalities collection)
{
    if(ModelState.IsValid)
    {
        try
        {
            WorklistManager.SaveModality(collection);
            Success(collection.CallingAETitle + WDictionary.AllowedModalitiesCreate, true);
        }
        catch (Exception ex)
        {
            Danger(ex.Message, true);
        }

        return RedirectToAction("Index");
    }
    return View();
}
```

Figura 4.4 Método Create de la clase “AllowedModalitiesController”

(Fuente: elaboración propia)

4.4. Estándar de codificación

Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada(48) El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad, es establecer un estándar de codificación, sobre el que se efectuarán luego revisiones del código. La adopción de un estándar de codificación, sólo es viable si se sigue desde el principio hasta el final del proyecto de software. No es práctico, ni prudente, imponer un estándar de codificación una vez iniciado el trabajo.(48)

Los nombres de los métodos y de las clases se escribieron utilizando el estilo *UpperCamelCase* el cual establece, para las palabras compuestas, la escritura en letra inicial mayúscula de cada una de ellas. Para nombrar las variables se determinó la utilización de minúsculas garantizando, en la medida de lo posible, ser descriptivas al igual que los métodos y las clases, de forma que al leerlas se conozca su propósito. Los nombres de las tablas se definieron con letra inicial minúscula antecedita de `tbl_`. En caso de que el nombre sea compuesto, las otras palabras también se escribieron en minúsculas.

En el método “Create” de la clase controladora “AllowedModalitiesController” se puede apreciar el uso de lo anteriormente descrito. Ver Figura 4.4. Otro ejemplo lo constituye en la clase “AllowedModalitiesMap” donde se puede observar el uso de los estilos definidos.

```
public AllowedModalitiesMap()  
{  
    Table("tbl_allowed_modalities");  
}
```

Figura 4.5 Fragmento de código de la clase “AllowedModalitiesMap”

(Fuente: elaboración propia)

Para nombrar los métodos además se determinó que los mismos incluyeran una descripción de los valores devueltos, por ejemplo en el método que se muestra a continuación de la clase “WorklistQueries”.

```
public static AllowedModalities GetModalityById(ISession session, int id)
```

Figura 4.6 Fragmento de código de la clase “WorklistQueries”

(Fuente: elaboración propia)

Se definió el uso de las llaves de apertura alineadas verticalmente cerrándolas donde los pares de llaves se alinean, como se muestra a continuación:

```
public static List<AllowedModalities> GetModalities(ISession session)  
{  
    return QueryExecuter.GetEntityItems<AllowedModalities>(session, null, null, null);  
}
```

Figura 4.7 Fragmento de código de la clase “WorklistManager”

(Fuente: elaboración propia)

4.5. Validación del servidor de listas de trabajo DICOM

La validación tiene como objetivo asegurar que el sistema de software satisfice las expectativas del cliente. Analiza la conformidad de los resultados generados con los esperados. Las pruebas de validación intentan demostrar que el software satisface sus requerimientos. (38)

Mediante el uso de la herramienta Modality Emulator en su versión 3.1.5 se pudo probar un flujo completo de trabajo del componente Celsus Worklist. Esta herramienta permitió emular un equipo de adquisición con los servicios DICOM necesarios.

Modality Emulator 3.1.5 es una herramienta que permite a los ingenieros y desarrolladores emular un entorno médico con escenarios, que permitan la prueba de determinado software. La misma permite simular las funcionalidades DICOM de un equipo de adquisición de imágenes médicas. Tiene como principal función probar y verificar la comunicación con todos los servicios DICOM que una modalidad utiliza realmente. Además se pueden probar los servicios que provee la propia aplicación que se realiza. Entre las principales funciones que posee se encuentran (49):

- Consultar la lista de trabajo de la modalidad que simula
- Enviar actualizaciones del estado de los procedimientos mediante el uso del servicio MPPS

Para la utilización del emulador primero se configuraron los parámetros de conexión del equipo de adquisición, estos son los que permiten la conexión de este al componente Celsus Worklist. Una vez establecidos los valores del puerto, dirección IP y *AE title* del servidor de listas de trabajo y del RIS al cual se le va a realizar la solicitud, se comenzó la realización del flujo de trabajo.

Para todos los casos probados:

1. Se verificó la conexión con el RIS a través de Celsus Worklist, resultando satisfactoria.

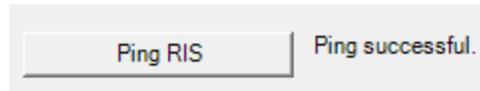


Figura 4.8 Captura de pantalla al botón *Ping RIS* de la aplicación Modality Emulator 3.1.5
(Fuente: elaboración propia)

2. Se verificó correctamente el funcionamiento de los servicios DICOM.



Figura 4.9 Captura de pantalla al botón DICOM Echo de la aplicación Modality Emulator 3.1.5
(Fuente: elaboración propia)

3. Se obtuvieron las listas de trabajo solicitadas por el equipo de adquisición.

Patient Name	Patient ID	Accession Nr	Requested Proc ID	Scheduled Proc Step Start Date	Modality	Scheduled Station AE Title
Patients Name	PatientID	123456	RPID	20060330	CR	AE Title
aaron ramsey	99	20140115-E...	70.73	20150512	US	Eco_I
Aleida Torres H...	34	20130314-Q...	25.26	20150512	US	Eco_I
Aleida Torres H...	34	20130706-O...	60.63	20150512	US	Eco_I
Aleida Torres H...	34	20130709-IK...	63.66	20150512	US	Eco_I
Aleida Torres H...	34	20130917-...	67.70	20150512	US	Eco_I
Martha Andrade...	108		83.86	20150512	US	Eco_I
Lisandra Castro ...	35	20130318-V...	49.52	20150512	US	Eco_I
Aleida Torres H...	34	20130708-X...	62.65	20150512	US	Eco_I
aaron ramsey	99	20140614-B...	76.79	20150512	US	Eco_I
Aleida Torres H...	34	20130318-L...	51.54	20150512	US	Eco_I
Juan Gomez	59	20130314-X...	28.29	20150512	US	Eco_I
Aleida Torres H...	34	20130315-K...	46.49	20150512	US	Eco_I
Carlos Herrera D...	39	20130315-M...	42.44	20150512	US	Eco_I
Aleida Torres H...	34	20130403-SI...	58.61	20150512	US	Eco_I
Claudia Fuentes	58	20130314-N...	21.22	20150512	US	Eco_I
Aleida Torres H...	34	20130314-B...	30.31	20150512	US	Eco_I
Carlos Herrera D...	39	20130315-Y...	38.40	20150512	US	Eco_I
Arelis Gomez T...	21	20130326-C...	54.57	20150512	US	Eco_I
Pepe Rodriguez...	61	20130314-L...	26.27	20150512	US	Eco_I
Juan Argel	57	20130314-T...	27.28	20150512	US	Eco_I
Aleida Torres H...	34	20130315-R...	46.48	20150512	US	Eco_I

Figura 4.10 Captura de pantalla a la vista Select MWL Response de la aplicación Modality Emulator 3.1.5
(Fuente: elaboración propia)

4. Una vez obtenida la lista de trabajo, se seleccionó el paciente al cual se le desea realizar el estudio en el equipo.
5. El equipo envió un mensaje N-CREATE indicando el comienzo del estudio.

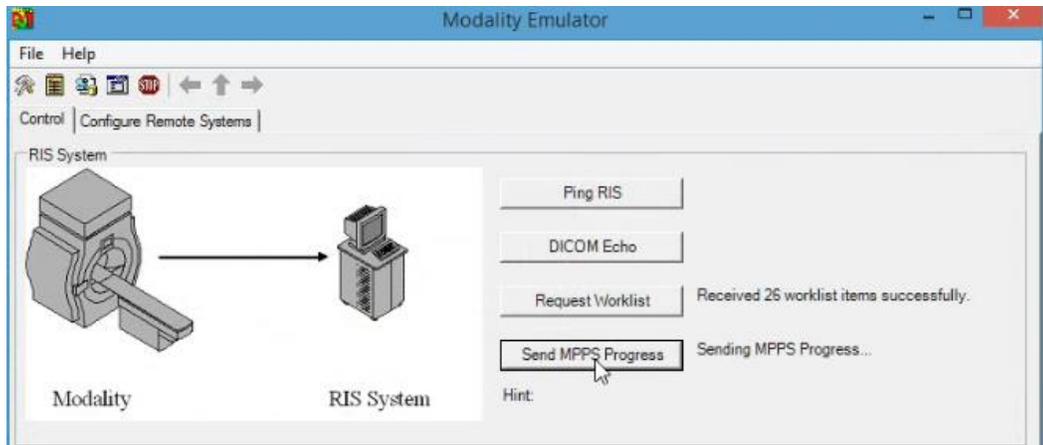


Figura 4.11 Captura de pantalla a la página principal de la aplicación Modality Emulator 3.1.5 (Fuente: elaboración propia)

6. Se envió una actualización del estado del estudio al RIS.
7. El estado del estudio en el RIS cambió a "In Progress".



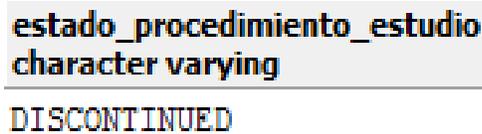
Figura 4.12 Captura de pantalla al estado *IN PROGRESS* del estudio en el RIS (Fuente: elaboración propia)

8. Se discontinuó la realización del estudio.



Figura 4.13 Captura de pantalla al botón *Send MPPS Progress* de la aplicación Modality Emulator 3.1.5 (Fuente: elaboración propia)

9. El estado del estudio, el valor de la cita y la hora de finalización del estudio cambiaron en el RIS.



```
estado_procedimiento_estudio
character varying
DISCONTINUED
```

Figura 4.14 Captura de pantalla al estado *DISCONTINUED* del estudio en el RIS

(Fuente: elaboración propia)

Finalizado el desarrollo de esta prueba, se pudo concluir que se puede establecer una correcta comunicación entre una modalidad y el RIS a través de Celsus Worklist. Esta comunicación se realizó mediante los servicios que establece DICOM para la comunicación entre sistemas médicos. La misma, estuvo guiada por las transacciones que establece el perfil SWF para el intercambio de información entre la modalidad y el RIS.

Asimismo, queda evidenciado que el componente Celsus Worklist, es capaz de actuar como intermediario entre los equipos de adquisición y el RIS, permitiendo el manejo de las listas de trabajo DICOM. Esto garantiza la consistencia de la información sobre el paciente, durante todo el proceso. Queda demostrado, además, que dicho componente puede ser utilizado por sistemas de cualquier fabricante, siempre y cuando cuenten con los servicios que define DICOM, para el intercambio de información entre sistemas médicos.

4.6. Declaración de conformidad DICOM

Para cada modalidad, dispositivo y versión de cada producto que se desarrolle utilizando el estándar DICOM, se debe crear una declaración de conformidad. La misma debe indicar para cada servicio DICOM que se utilice el tipo correspondiente (cliente, servidor o ambos). La declaración de conformidad del componente Celsus Worklist se puede encontrar en el expediente del proyecto. Esta permitirá, a los responsables de la red imagenológica del centro de salud donde se encuentre, conocer cómo se integrará el componente Celsus Worklist con los demás equipos y sistemas de apoyo al diagnóstico médico.

A partir de los elementos analizados en este capítulo se pudo obtener una visión sobre cómo será la estructura encargada del manejo de los datos en el sistema. De igual forma se mostró una vista de la organización de los componentes del mismo, y su despliegue en nodos de ejecución. Se definió que el tratamiento de errores será posible mediante el manejo de excepciones. Se especificaron los estándares de codificación a utilizar. Por último, la validación del componente Celsus Worklist permitió concluir que el mismo puede ser utilizado para el manejo de listas de trabajo entre los sistemas PACS y RIS.

RESULTADOS OBTENIDOS

Con la implementación del componente Celsus Worklist se obtienen los siguientes resultados:

- Se elimina la entrada manual de los datos de los estudios a realizar en los equipos de adquisición de imágenes médicas, evitando errores en el flujo de información de los departamentos de imagenología.
- El componente Celsus Worklist podrá ser utilizado por cualquier RIS y PACS, que cuente con los servicios establecidos por el estándar DICOM para el uso de las listas de trabajo en los equipos de adquisición de imágenes médicas.
- El estado del estudio imagenológico será conocido en el RIS, desde su inicio a su culminación en el equipo de adquisición de imágenes médicas.
- Se garantiza una interacción con el sistema mediante su interfaz web, lo que permite la administración y configuración del mismo por parte del personal encargado en las instituciones hospitalarias.

CONCLUSIONES

Con el desarrollo de las funcionalidades asociadas al componente Celsus Worklist se concluye lo siguiente:

- Las funcionalidades del componente Celsus Worklist fueron definidas teniendo en cuenta la problemática existente y las tendencias actuales en el desarrollo de servidores de listas de trabajo DICOM.
- El componente desarrollado permite la consistencia de la información del paciente, durante la realización del estudio imagenológico y la integración del mismo con los sistemas de apoyo al diagnóstico médico, debido a la utilización del perfil propuesto por IHE, flujo de trabajo programado y las transacciones definidas por este.
- A partir de las características del componente se estableció una arquitectura basada en los estilos arquitectónicos cliente-servidor y n-capas.
- Con la culminación de la presente investigación se obtuvo un componente capaz de realizar el manejo de las listas de trabajo DICOM entre los sistemas PACS y RIS.
- La declaración de conformidad DICOM elaborada, permite a los responsables de la red imagenológica del centro de salud donde se encuentre, conocer cómo se integrará el componente Celsus Worklist con los demás equipos y sistemas de apoyo al diagnóstico médico.

RECOMENDACIONES

Al concluir la implementación del componente Celsus Worklist para la solución PACS-RIS del CESIM se recomienda:

- Implementar los servicios DICOM, que permitan el intercambio de información con el componente Celsus Worklist, en la solución PACS del CESIM.
- Integrar el componente Celsus Worklist al sistema de autenticación unificada.
- Realizar pruebas al componente en un entorno hospitalario real.

REFERENCIAS BIBLIOGRÁFICAS

1. BRAVO, Camilo. Radiología digital vs convencional. [En línea]. 2013. [Accedido 18 octubre 2014]. Disponible en: <http://es.slideshare.net/desskrgr/radiologia-digital-vs-convencional>
2. BRITISH ENGINEER. Sir Godfrey Newbold Hounsfield. *British engineer* [En línea]. 2013. Disponible en: <http://www.britannica.com/EBchecked/topic/272989/Sir-Godfrey-Newbold-Hounsfield>
3. IAEA. Radiografía Digital. [En línea]. 2013. [Accedido 18 octubre 2014]. Disponible en: https://rpop.iaea.org/RPOP/RPoP/Contentes/InformationFor/HealthProfessionals/1_Radiology/DigitalRadiography.htm
4. CLINIC-CLOUD. ¿Qué es el formato DICOM? Las claves del estándar en imágenes médicas. [En línea]. 2014. [Accedido 19 octubre 2014]. Disponible en: <https://clinic-cloud.com/formato-dicom-que-es-estandar-imagenes-medicas/>
5. DÍAZ, Carlos Guzmán y AGUILAR, Denys Bárbaro Vega. SLD076 SISTEMA PARA EL ALMACENAMIENTO Y TRANSMISION DE IMAGENES MEDICAS VERSION 3.0. [En línea]. 2013. Disponible en: www.informaticahabana.com
6. CASTRO, Arelys Rivero y NOGUERA, Alejandro Hernández. Revista Cubana de Informática Médica - Propuesta de aplicación para el registro de estudios imagenológicos de modalidades no DICOM. [En línea]. 2012. Vol. 4, no. 2. [Accedido 19 octubre 2014]. Disponible en: http://scielo.sld.cu/scielo.php?pid=S1684-18592012000200006&script=sci_arttext
7. FAIMBE, H.K.Huang. *PACS and Imaging Informatics*. Segunda. Wiley-BlackWell, 2010.
8. DUBNER, Dr. Fernando Zalduondo. SOCRAD Sociedad Radiológica de Puerto Rico y American College of Radiology. [En línea]. 2014. Disponible en: <http://www.socrad.com/patients-corner/29-modalidades-de-imagenes-medicas>
9. VILLALOBOS, José Ángel Córdova, DOMÍNGUEZ, Maki Esther Ortiz y RÉTIZ, María Luisa González. *Sistemas para archivo y comunicación de imágenes (PACS)* [En línea]. 2009. Salud. Disponible en: http://www.cenetec.salud.gob.mx/descargas/equipo_guias/guias_tec/41gt_PACS.pdf
10. LÓPEZ, Martha Rodríguez y GARCÍA, Raymundo Rodríguez. *Propuesta de aplicación de perfiles de integración IHE entre los sistemas alas PACS-alas RIS-alas HIS*. 2010.
11. GARCÍA, Lorena Pérez. Sistemas de información en el servicio de radiología. *SlideShare* [En línea]. 2010. Disponible en: <http://es.slideshare.net/natachasb/ris-3434028>

12. KOWALCZYK, Luiza, CARRINO, John A. y SOLOMON, Harry. *Digital Imaging and Communication in Medicine.Strategic Document* [En línea]. 2014. Disponible en: <http://dicom.nema.org>
13. DIAZ, Miguel Chavarri y LLORÉIS, R. Maximiliano Lloret. *Diagnóstico por la imagen* [En línea]. 2014. Disponible en: www.conganat.org
14. NEMA. DIMSE Service. *The Association of Electrical Equipment and Medical Imaging Manufacturers* [En línea]. 2013. Disponible en: http://medical.nema.org/dicom/2013/output/chtml/part07/sect_7.5.html
15. DICOM LIBRARY. SOPs. *DICOM Library* [En línea]. 2011. Disponible en: <http://www.dicomlibrary.com/dicom/sop/>
16. ZAMORA, Ana Belén Peinado. *Aplicación web para gestión y optimización de dosis a pacientes en pruebas radiodiagnósticas en el HGUCR* [En línea]. 2013. Disponible en: <https://ruidera.uclm.es/xmlui/handle/10578/3678>
17. ESPAÑA, IHE. ¿Qué es IHE? [En línea]. 2014. Disponible en: <http://www.ihe-e.org>
18. AVRAHAM, Ellie, BARTON, Frederik, DICKMANN, Christoph, JAIN, Sanjay y LEGGET, Tori. *IHE Radiology User's Handbook* [En línea]. 2005. Disponible en: http://www.ihe.net/Resources/upload/ihe_radiology_users_handbook_2005edition.pdf
19. IHE RADIOLOGY TECHNICAL COMMITTEE. *IHE Radiology Technical Framework Supplement.Scheduled Workflow.b (SWF.b)* [En línea]. 2014. IHE International, Inc. Disponible en: http://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_SWF.b.pdf
20. INC, IHE International. Query Modality Worklist. En: *IHE Radiology Technical Framework IHE RAD TF-2. 2011. p. 60-69.*
21. INC, IHE International. Modality Procedure Step In Progress. En: *IHE Radiology Technical Framework IHE RAD TF-2. 2011. p. 70-82.*
22. INC, IHE International. Modality Procedure Step Completed/Discontinued. En: *IHE Radiology Technical Framework IHE RAD TF-2. 2011. p. 83-90.*
23. PHILIPS. *DICOM Conformance Statement Xcelera Connect Release 1.3 L2* [En línea]. 2008. PHILIPS. Disponible en: http://incenter.medical.philips.com/doclib/enc/fetch/2000/4504/577242/577256/588723/5144873/5144488/5144587/DICOM_Conformance_Statement_Xcelera_Connect_R1.3L2.pdf%3fnodeid%3d5148757%26vernum%3d1

24. PHILIPS. *DICOM Conformance Statement iSite PACS*. 2008. Philips Medical Systems Radiology Informatics.
25. AGFA HEALTH CARE. IMPAX RIS DICOM Server. *Agfa Health Care* [En línea]. 2014. Disponible en: <http://www.agfahealthcare.com>
26. HEALTHLINE INFORMATION SYSTEMS. HealthE PACS Worklist. [En línea]. 2014. Disponible en: <http://www.healthlineis.com/MissionStatement.html>
27. LARMAN, Craig. *UML y Patrones .Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2. Prentice hall, 2002.
28. RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady. *El lenguaje unificado de modelado. Manual de referencia*. Addison Wesley, 2000.
29. MICROSOFT DEVELOPER NETWORK. Introducción al lenguaje C# y .NET Framework. *Microsoft Developer Network* [En línea]. 2015. Disponible en: <https://msdn.microsoft.com>
30. NETWORK, Microsoft Developer. Framework.NET. *Microsoft Developer Network* [En línea]. 2015. Disponible en: <https://msdn.microsoft.com/es-es/library/hh425099%28v=vs.110%29.aspx>
31. NAIDU, Venugopal. Enterprise Architect 7.5. [En línea]. 2014. Disponible en: <http://enterprise-architect.software.informer.com/7.5/>
32. QUIJANO, Juan. Visual Studio 2013. [En línea]. 2013. Disponible en: www.genbetadev.com/herramientas/visual-studio-2013
33. MICROSOFT. Información general sobre ASP.NET MVC. *Microsoft Developer Network* [En línea]. 2014. Disponible en: <https://msdn.microsoft.com/es-es/library/dd381412%28v=vs.108%29.aspx>
34. POSTGRESQL. PostgreSQL. *PostgreSQL* [En línea]. 2015. Disponible en: <http://www.postgresql.org>
35. CANO, Fernando P. Najera. TortoiseSVN Un cliente de Subversion para Windows Versión 1.8. [En línea]. 2015. Disponible en: http://tortoisesvn.net/docs/nightly/TortoiseSVN_es/help-onepage.html#tsvn-preface-about
36. JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. *Proceso Unificado de Desarrollo de Software*. Adison Wesley, 2000. ISBN 84-7829-0362.
37. KRUCHTEN, Philippe. *The Rational Unified Process an Introduction*. Segunda. Addison Wesley, 2000. ISBN 0-201-70710-1.
38. SOMMERVILLE, Ian. *Ingeniería del Software*. 7ma. España, 2005. ISBN 8478290745.

39. CAN, Carolina Novelo. Requerimientos Funcionales y no funcionales. [En línea]. 2010. Disponible en: <http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales#scribd>
40. PRESSMAN, Roger S. *Ingeniería de Software. Un enfoque práctico*. 6ta. 2005. ISBN 9701054733.
41. © RA-MA. *Conceptos generales de la arquitectura de aplicaciones web* [En línea]. 2014. Disponible en: www.ra-ma.es
42. BUSCHMAN, Frank, MEUNIER, Regine, ROHNER, Hans, SOMMERLAD, Peter y STAL, Michael. *Pattern-Oriented Software Architecture. A System of Pattern*. Jhon Wiley & Sons Canada, 1997. ISBN 0471958697.
43. LLORENTE, Cesar de la Torre, CASTRO, Unai Zorrilla, BARROS, Miguel Ángel Ramos y NELSON, Javier Calvarro. *Guía de arquitectura N-capas orientada al dominio con .NET*. España, 2010. ISBN 9788493696398.
44. MUHLRAD, Daniel. *Patrones de Diseño*. 2008.
45. LARMAN, Craig. Fase de Diseño. En: *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Prentice hall, 2002. p. 185-217.
46. PRESSMAN, Roger S. *Ingeniería de Software. Un enfoque práctico*. 5ta. 2002. ISBN 8448132149.
47. Excepciones y control de excepciones (Guía de programación de C#). *Microsoft Developer Network* [En línea]. 2014. Disponible en: <https://msdn.microsoft.com/es-es/library/ms173160.aspx>
48. MICROSOFT DEVELOPER NETWORK. Revisiones de código y estándares de codificación. *Microsoft Developer Network* [En línea]. 2003. Disponible en: <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>
49. SOURCEFORCE.NET. Modality Emulator. *DVtk Project* [En línea]. 2012. Disponible en: <http://dicom.dvtk.org/modules/wiwimod/index.php?page=Modality+Emulator&cmenu=products>

ANEXOS

Anexo 1. Diagramas de clases del caso de uso Gestionar modalidad

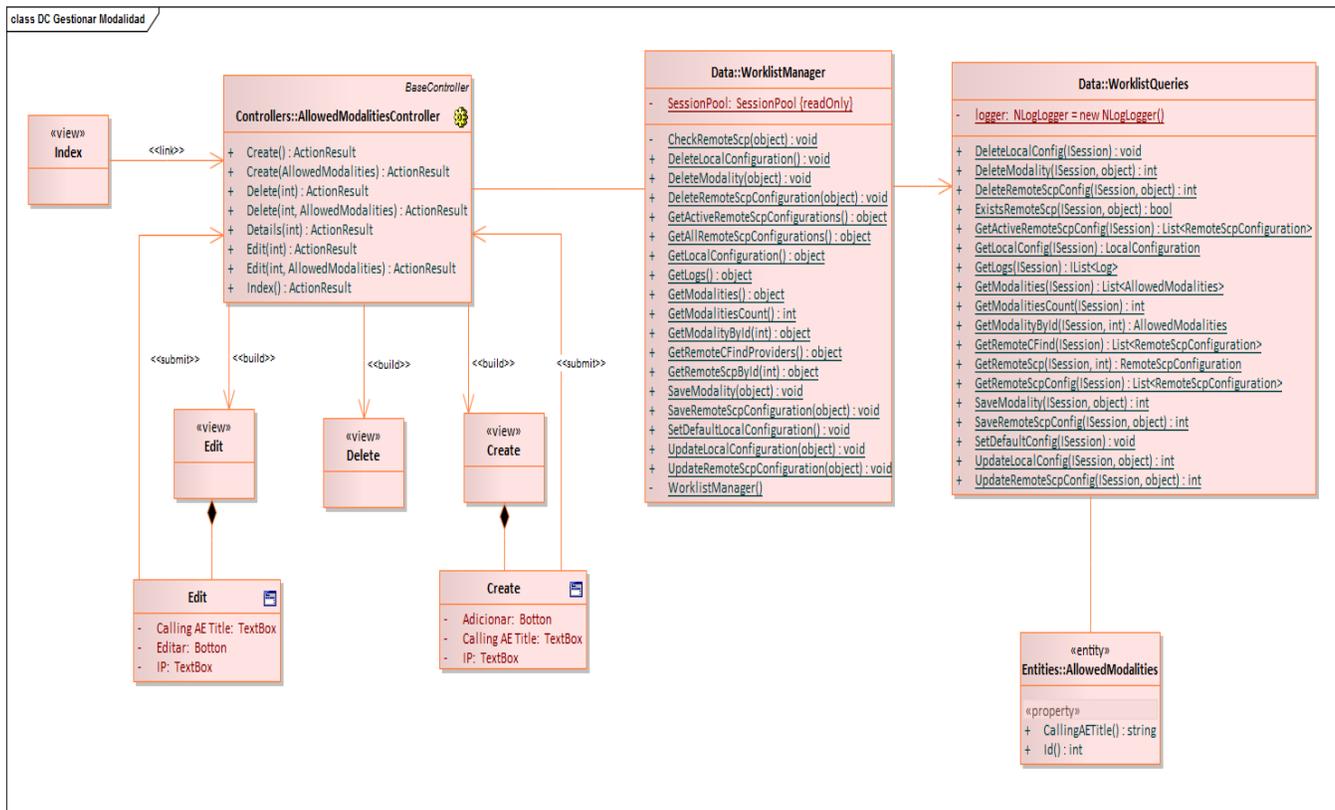


Figura 5 Diagrama de clases del diseño del caso de uso Gestionar modalidad (fuente: elaboración propia)

Anexo 2. Descripción de las clases del diseño

Tabla 6 Descripción de la clase AllowedModalitiesController

(Fuente: elaboración propia)

Nombre: AllowedModalitiesController
Tipo de clase: controladora

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Create (collection: AllowedModalities)
Descripción:	Permite crear una modalidad.
Nombre:	Delete (id: int, collection: AllowedModalities)
Descripción:	Permite eliminar una modalidad seleccionada.
Nombre:	Details (id: int)
Descripción:	Muestra los detalles de una modalidad determinada.
Nombre:	Edit (id: int, collection: AllowedModalities)
Descripción:	Permite editar una modalidad.

Anexo 3. Diagramas de Secuencia

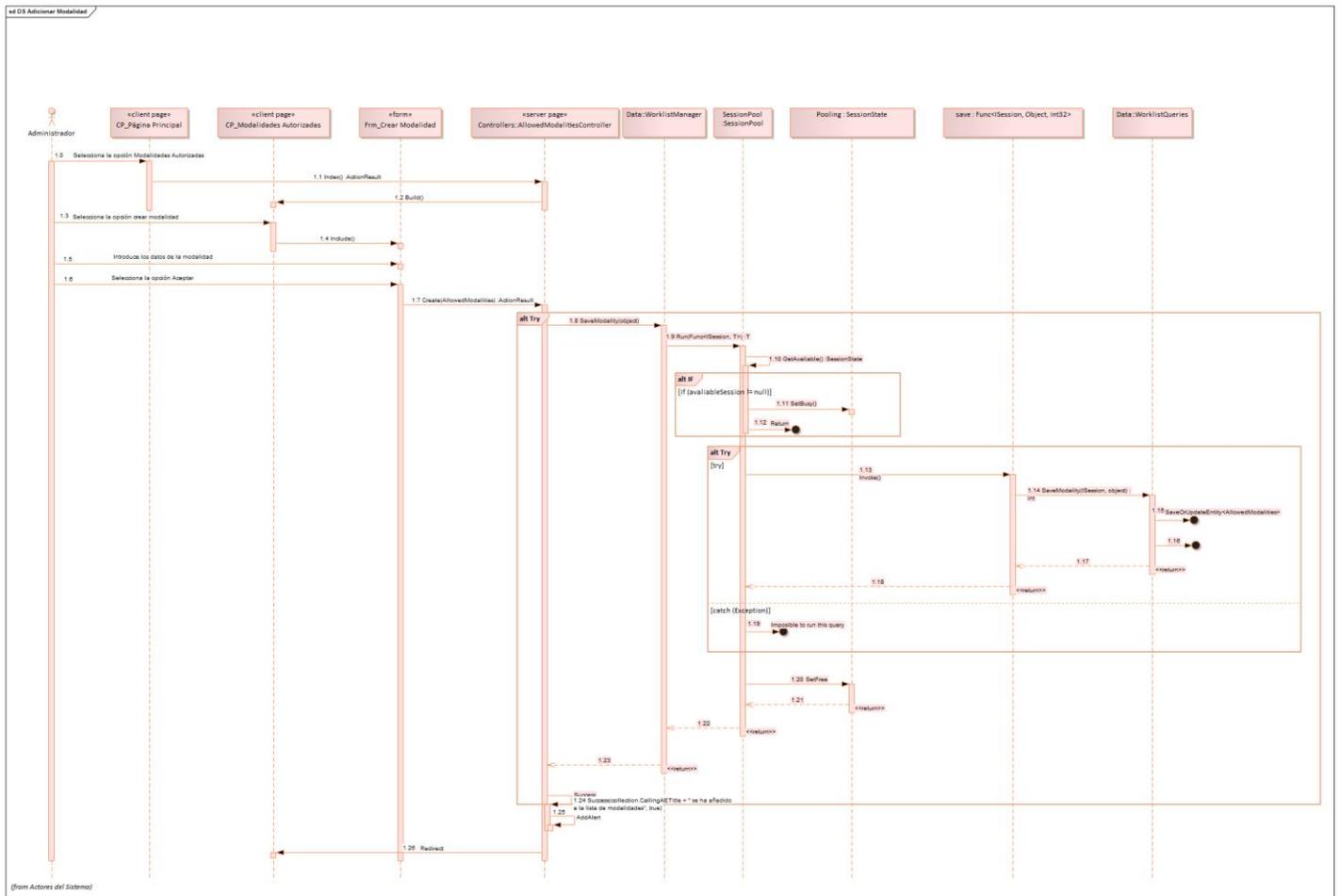


Figura 6 Diagrama de secuencia Adicionar modalidad (fuente: elaboración propia)

Anexo 4. Descripción de las tablas de la base de datos

Tabla 7 Descripción de la tabla “tbl_remote_scp_configuration”

(Fuente: elaboración propia)

Nombre: tbl_remote_scp_configuration		
Descripción: Esta tabla será la encargada de almacenar los datos de los nodos DICOM que se conectarán al componente Celsus Worklist.		
Atributo	Tipo	Descripción
Id	Integer	Identifica cada nodo DICOM que se encuentra conectado al servidor de listas de trabajo.
called_ae_title	Varchar	Nombre que identifica a una aplicación de red DICOM. En este caso los nodos DICOM.
is_active	Boolean	Especifica si la conexión con el nodo DICOM está activa o no.
ip_address	Varchar	Dirección IP del nodo DICOM.
Port	Integer	Puerto por donde se conecta el nodo DICOM.
supports_cfind	boolean	Indica si soporta el envío de mensajes c-find.

Tabla 8 Descripción de la tabla “tbl_local_configuration”

(Fuente: elaboración propia)

Nombre: tbl_local_configuration		
Descripción: Esta tabla será la encargada de almacenar los datos la configuración local del servidor de listas de trabajo DICOM.		
Atributo	Tipo	Descripción
Id	Integer	Identificador de la tabla.
local_ae_title	varchar	Nombre que identifica a una aplicación de red DICOM.
local_port	Integer	Puerto por donde se conecta el servidor.
association_timeout	Integer	Tiempo máximo establecido para la asociación.

Nombre: tbl_local_configuration

use_tls	boolean	Especifica si utiliza el protocolo TLS.
---------	---------	---

Tabla 9 Descripción de la tabla “tbl_log”

(Fuente: elaboración propia)

Nombre: tbl_log

Descripción: Esta tabla será la encargada de almacenar el registro de los logs generados por el servidor de listas de trabajo DICOM.

Atributo	Tipo	Descripción
Id	Integer	Identificador de la tabla.
log_datetime	timestamp	Fecha hora del log.
log_level	Varchar	Especifica las categorías de los logs (<i>Debug, Info, Warning, Error</i>).
log_message	Varchar	Es la descripción de evento que se ha producido.

Tabla 10 Descripción de la tabla “tbl_user”

(Fuente: elaboración propia)

Nombre: tbl_user

Descripción: Esta tabla será la encargada de almacenar los usuarios que se registran en la aplicación.

Atributo	Tipo	Descripción
Id	Integer	Identificador de la tabla.
user_name	Varchar	Nombre del usuario.
password	Varchar	Contraseña del usuario.

GLOSARIO DE TÉRMINOS

Bróker: dispositivo o aplicación que “soluciona” problemas de conectividad, entre un sistema que no cumple una especificación determinada y otros que sí la cumplen. Los mensajes enviados hacia o desde el sistema pasan a través del bróker, que los traduce de dicha especificación a un lenguaje que el sistema pueda entender.

Dirección IP: (IP es un acrónimo para *Internet Protocol*) número con el cual se identifica una computadora conectada a una red.

Framework: estructura predefinida para la creación de aplicaciones. Puede estar formado por un conjunto de librerías y clases o por una arquitectura que facilita el desarrollo de software.

HIS (*Hospital Information System*): sistema de información hospitalario.

HL7 (*Health Level 7*): estándar establecido para el intercambio, administración e integración de datos referentes a la atención sanitaria a pacientes así como a la prestación, gestión y evaluación de los servicios sanitarios.

Log: registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación.

Mantenibilidad: propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla.

Marco técnico de IHE: documento que define los Perfiles de Integración de IHE, los problemas y los casos de uso a los que se refieren los Actores y Transacciones que intervienen en ellos. Proporciona también instrucciones precisas de implementación para cada transacción.

Proveedor de Clase de Servicio (*Service Class Provider, SCP*): sistema o aplicación que proporciona un Servicio DICOM (podría decirse que es el “servidor” de un servicio).

Puerto: interfaz para comunicarse con un programa a través de una red.

RNDI: Requisito no Funcional de Diseño e Implementación.

RNFO: Requisito no Funcional de Funcionamiento.

RNL: Requisito no Funcional de Derechos Legales y de Autor.

RNU: Requisito no Funcional de Usabilidad.

RNS: Requisito no Funcional de Seguridad

Transacción: intercambio de información entre actores. El Marco técnico describe cómo utilizar los estándares establecidos (HL7, DICOM) para cada transacción en el intercambio de información.

Usuario de Clase de Servicio (*Service Class User, SCU*): sistema o aplicación que utiliza un servicio DICOM (podría decirse que es el “cliente” de un servicio).