

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Sistema de Referencia
de Imágenes Médicas.**

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

Autor(es): Heliodoro Rodríguez Milian.
Nuvia Angélica Estevez Rojas.

Tutor(es): Lic. Rolando Bonal Cáceres.
Lic. Pedro Medina Riesgo.

Ciudad de la Habana, Junio del 2007

DECLARACIÓN DE AUTORÍA.

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

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

Nuvia Angélica Estevez Rojas.

Lic. Rolando Bonal Cáceres.

Heliodoro Rodríguez Milian.

Lic. Pedro Medina Riesgo.

DATOS DE CONTACTO.

Lic. Rolando Bonal Cáceres (email: bonal@uci.cu).

Profesor graduado de Licenciatura en Física. Ha impartido la asignatura Física I y Física II. Posee categoría docente de Asistente. Es especialista de la Dirección de Software para la Salud. Actualmente se desempeña como Jefe del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI).

Lic. Pedro Medina Riesgo (email: medina@uci.cu).

Profesor graduado de Licenciatura en Cibernética Matemática. Ha impartido las asignaturas Programación II y Programación IV. Es profesor de la facultad 7 y se desempeña actualmente como Jefe de Proyecto dentro del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI).

Agradecimientos.

Agradezco a mi familia por confiar en mí siempre y apoyarme en todos mis sueños. Además a todos lo que de una forma u otra hicieron posible este trabajo, a mis amigos, mis profesores, y a mis tutores.

Nuvia Angélica Estevez Rojas.

A mi familia, a mi novia que ha sido un gran apoyo en estos 5 años de carrera, a Alejandro Mario por inspirarme con su ejemplo. A mis profesores de siempre, que han contribuido en mi formación profesional y hoy pueden ver realizado este sueño, a mis amigos de toda la vida que me han brindado su apoyo en todo momento y con los que quisiera contar por siempre por que son los mejores.

Heliodoro Rodríguez Milian.

Agradecemos a la revolución por incluirnos en este maravilloso proyecto.

Dedicatoria.

A mami y papi, y a mi hermano Maiquel; que todo lo son para mi, mi vida y mi orgullo. Y a mi Revolución cubana, la mejor de todas.

Nuvia Angélica Estevez Rojas.

Dedico este trabajo a mis padres y mis abuelos, que siempre han estado cuando los he necesitado y que tanto se han entregado por la realización de este sueño.

Heliodoro Rodríguez Milian.

A nuestro Comandante Fidel.

Resumen.

Actualmente Cuba no cuenta con un sistema que permita a los radiólogos localizar y obtener estudios imagenológicos fuera de una red local, de ahí que el objetivo fundamental de esta investigación consista en desarrollar un sistema que permita guardar y obtener referencias de los estudios imagenológicos realizados en el país, así como de las rutas para la recuperación de estos estudios.

Como parte de la informatización del Sistema Nacional de Salud, se ha concebido la puesta en funcionamiento de una Red Nacional de Imágenes que comunicaría los departamentos imagenológicos de todo el país.

Fundamentado en el estudio de la situación problemática y de las tecnologías, incluyendo los sistemas de este tipo en el mundo, además de las metodologías y herramientas de desarrollo; el Sistema de Referencia de Imágenes Médicas constituye una solución distribuida. Esta alternativa sugiere la instalación de un servidor de referencias nacional junto a una aplicación que permite el acceso a este desde la web. Además del despliegue por cada hospital de un sistema orquestador que controlaría la interacción entre los hospitales y el servidor de referencias.

El Sistema de Referencias de Imágenes Médicas constituye pues, un aporte significativo a la constitución de la Red Nacional de Imágenes, ya que permitirá poseer una referencia de cada estudio imagenológico realizado y con ello poder localizarlo y obtenerlo.

Tabla de Contenidos.

DECLARACIÓN DE AUTORÍA.	2
DATOS DE CONTACTO.....	3
Dedicatoria.....	5
Resumen.	6
Tabla de Contenidos.....	7
Índice de Figuras.	11
Introducción.	2
Idea a Defender.	4
Tareas de la investigación.....	4
Capítulo 1: Fundamentación teórica.....	7
1.1 Estado del arte de las soluciones para la localización y la obtención de estudios radiológicos fuera de una red local.	7
A nivel Mundial.	7
A nivel nacional.....	7
1.2 Descripción del flujo actual de los procesos de transmisión de las imágenes médicas DICOM 3.0 en el Sistema de Imagenología Cubano.....	10
1.3 Metodología, Técnicas y Herramientas a utilizar para el desarrollo del sistema.	11
1.3.1 Metodología de desarrollo RUP (Rational Unified Process).....	12
1.3.2 Lenguaje de Modelado UML (Lenguaje Unificado de Modelación).....	12
1.3.3 Herramienta de modelado Enterprise Architect.	13
1.3.4 Plataforma de desarrollo .Net.....	14
1.3.5 Visual C#.	15
1.3.6 Servicios Web.....	15
1.3.7 Arquitectura Orientada a Servicios. SOA.	16
1.3.8 Patrón de Arquitectura BROKER. Object Request Broker.....	17
1.3.9 El protocolo SOAP.	17
1.3.10 .Net Remoting.....	18

1.3.11 Visual Studio 2005 Team System.	19
1.3.12 PostgreSql.	19
Capítulo 2: Características del sistema.	21
2.1 Problema y situación problemática.	21
2.1.1 Objetivos estratégicos de la organización y oportunidades de negocio que los secundan.	22
2.1.2 Flujo actual de los procesos involucrados en el campo de acción.	22
2.1.3 Análisis crítico de los procesos de localización y transmisión de los estudios.	22
2.2 Objeto de automatización.	23
2.2.1 Descripción de los procesos que serán objeto de automatización.	23
2.2.2 Descripción de los sistemas automatizados que existen vinculados con el campo de acción del proyecto.	23
2.3 Información que se maneja.	24
2.4 Propuesta de sistema.	24
2.4.1 Modelo de Dominio.	25
2.4.1.1 Diagrama del Modelo de Dominio.	26
2.4.2 Requerimientos.	26
2.4.2.1 Dependencias y Relaciones con otros softwares.	26
2.4.2.2 Requerimientos Funcionales.	28
2.4.2.3 Requerimientos No Funcionales.	29
2.4.3 Descripción general del sistema propuesto.	32
2.4.4 Análisis de otras soluciones existentes.	33
2.5 Definición de los casos de uso.	34
2.5.1 Definición de los Actores del Sistema a Automatizar.	34
2.5.2 Definición de los Casos de Uso del Sistema.	35
2.5.3 Diagrama de Casos de Uso del sistema.	38
2.5.4 Descripción de los Casos de Uso.	40
1. Buscar_Referencia.	40
2. Insertar_Referencia.	41
3. AdicionarReporte_Referencia.	42
4. Adicionar_Referencia.	43
5. Localizar_Referencia.	45

6. Guardar_Registro_Búsqueda.....	46
7. Visualizar_Registro_Búsqueda.....	47
8. Búsqueda Web de Referencias.....	48
Capítulo 3: Análisis y diseño del sistema.....	50
3.1 Modelo Arquitectónico.....	50
3.2 Análisis.....	53
3.2.1 Diagrama de Clases de Análisis por Casos de Uso.....	54
3.2.1.1 Paquete Orchestator.....	54
3.2.1.2 Paquete SNRIM.....	56
3.2.1.3 Paquete Sistema de Búsqueda de Referencias.....	58
3.3 Diseño.....	59
3.3.1 Diagramas de Secuencia por Casos de Uso.....	60
3.3.1.1 Paquete Orchestator.....	60
3.3.1.2 Paquete SNRIM.....	64
3.3.1.3 Paquete Sistema de Búsqueda de Referencias.....	67
3.3.2 Descripción de las Clases de Diseño.....	68
3.3.2.1 Paquete Orchestator.....	68
3.3.2.2 Paquete SNRIM.....	81
3.3.2.3 Paquete Sistema de Búsqueda de Referencias.....	84
3.4 Diseño de la BD.....	90
3.4.1 Modelo Entidad Relación.....	90
3.4.1.1 Diagrama Entidad Relación de la Base de Datos de Registro de Búsquedas.....	91
3.4.1.2 Diagrama Entidad Relación de la Base de Datos de Referencias de Estudios Radiológicos.....	92
3.4.2 Descripción de las tablas de la Base de Datos.....	93
3.4.2.1 Base de Datos de Registro de Búsquedas.....	93
3.4.2.2 Base de Datos de Referencias de Estudios Radiológicos.....	94
Capítulo 4: Implementación.....	96
4.1 Diagrama de componentes.....	96
4.2 Modelo de Despliegue.....	97
4.2.1 Diagrama de despliegue.....	98
Conclusiones.....	100

Recomendaciones.....	101
Referencia Bibliográfica.....	102
Bibliografía.....	104
Anexos.....	106
Anexo 1. Glosario de Términos.....	106
Anexo 2: Diagramas de clases de Diseño.....	107
A.2.1 Paquete Orchestator.....	107
A.2.2 Paquete SNRIM.....	111
A.2.3 Paquete Sistema de Búsqueda de Referencias.....	114

Índice de Figuras.

Figura 2 - Diagrama del Modelo De Objetos del Sistema.....	26
Figura 3 – Casos de Uso del Sistema.....	39
Figura 4 - Interacción de los elementos de Cassandra.....	52
Figura 5 – Caso de Uso Buscar_Referencia.	54
Figura 6 – Visualizar_Registro_Búsqueda.	55
Figura 7 – Guardar_Registro_Búsqueda.....	55
Figura 8 – Insertar_Referencia.....	56
Figura 9 – Adicionar_Referencia.....	56
Figura 10 – AdicionarReporte_Referencia.	57
Figura 11 – Localizar_Referencia.	57
Figura 12 – Localizar_Referencia.	58
Figura 13 – Diagrama de Secuencia Buscar_Referencia.	60
Figura 14 - Diagrama de Secuencia Visualizar_Registro_Búsqueda.....	61
Figura 15 - Diagrama de Secuencia Guardar_Registro_Búsqueda.	62
Figura 16 - Diagrama de Secuencia Insertar_Referencia.....	63
Figura 17 - Diagrama de Secuencia Adicionar_Referencia.	64
Figura 18 - Diagrama de Secuencia AdicionarReporte_Referencia.....	65
Figura 19 - Diagrama de Secuencia Localizar_Referencia.....	66
Figura 20 - Diagrama de Secuencia BúsquedaWeb_Referencia.....	67
Figura 21 – Diagrama Entidad Relación del Paquete Orchestator.	91
Figura 22 - Diagrama Entidad Relación del Paquete SNRIM.	92
Figura 23 - Diagrama de Componentes.	97
Figura 24 - Diagrama de Despliegue.....	99
Figura 25 – Diagrama de Clases Buscar_Referencia.....	107
Figura 26 - Diagrama de Clases Visualizar_Registro_Búsqueda.	108
Figura 27 - Diagrama de Clases Guardar_Registro_Búsqueda.....	109
Figura 28 - Diagrama de Clases Insertar_Referencia.....	110
Figura 29 - Diagrama de Clases Adicionar_Referencia.....	111
Figura 30 - Diagrama de Clases AdicionarReporte_Referencia.	112
Figura 31 - Diagrama de Secuencia Localizar_Referencia.....	113
Figura 32 - Diagrama de Clases AdicionarReporte_Referencia.	114

Introducción.

Desde que el 8 de noviembre de 1895 el físico alemán Wilhelm Konrad Röntgen descubriera los rayos X, la imagenología médica ha experimentado más de 100 años de constante evolución hasta llegar a convertirse en el medio de investigación y diagnóstico más utilizado en la actualidad. Sin embargo la necesidad de compartir entre los hospitales los estudios imagenológicos se ha vuelto apremiante en los últimos años; tal es la razón de la preocupación de nuestro país por establecer una red nacional de imágenes, la cual debe permitir a los especialistas de la materia acceder a los estudios ubicados en cualquier localización dentro de la red nacional.

Wilhelm Konrad Röntgen trabajando con tubos de rayos catódicos sometidos a diferencias de voltaje, se dio cuenta de que estos emitían un tipo de radiación capaz de penetrar los más diversos materiales a una distancia considerable. Por ser desconocidos, les bautizó como rayos X. Luego de haber probado sus efectos sobre diversos materiales, decidió experimentar sobre el tejido humano. Para lo cual, le solicitó a su esposa que pusiera su mano entre una placa, donde registraría el resultado y bajo los rayos X: Esta fue la primera radiografía de la historia. (1)

Desde entonces, las aplicaciones de los rayos X no han dejado de desarrollarse y expandirse, al punto de que en la actualidad forman parte indispensable de los recursos de diagnósticos con que dispone la medicina. Esta tecnología se ha ido especializando en materias como la Radiología General, Mamografía, Ultrasonido, Tomografía Computada, Resonancia Magnética y Radiología Intervencionista, entre otros.

Desde que Röntgen descubrió los Rayos "X" en 1895 a la fecha, los médicos han utilizado para realizar los diagnósticos, la información que proporcionan las imágenes obtenidas del interior del cuerpo humano. Durante muchos años, solamente se usaron los rayos descubiertos por Röntgen, pero los investigadores han introducido otras energías y otros métodos de usar las mismas. Así tenemos imágenes diagnósticas por otros medios como el ultrasonido, los isótopos, radioactividad, la termografía, la resonancia magnética nuclear; además de revolucionar la obtención de imágenes con rayos "X" por la nueva tecnología de la llamada tomografía axial computarizada. (2)

La gran cantidad de imágenes para diagnóstico médico que se utilizan en la actualidad ha hecho complicado su manejo, principalmente cuando estas deben imprimirse, archivarse y transmitirse. El desarrollo de estas tecnologías de diagnóstico ha llevado de la mano la evolución de equipos que sean capaces de soportarlas y hacerlas más aprovechables. En ello se han visto envueltas algunas de las compañías de más prestigio internacional como Phillips, con su división Philips Medical System, SIEMENS, General Electric y Kodak, entre otras. Estas empresas se han dedicado con mucha fuerza al desarrollo de PACS¹, y al despliegue de estos, incluyendo desde las instalaciones de red hasta las estaciones de visualización.

Un PACS está compuesto por varios elementos que tienen la capacidad de interactuar entre sí, y con otros sistemas de información como el HIS² o el RIS³. Los componentes de un PACS se dividen fundamentalmente en tres. Un componente para la Generación de las Imágenes, equipos de alta tecnología con interfaz de acceso a las diferentes modalidades como TAC⁴ y Medicina Nuclear. Un Servidor de Base de Datos y Archivos que almacena la información del paciente, el estudio y los datos con los cuales el PACS puede localizar las imágenes almacenadas y un componente de Visualización representado por las estaciones de visualización de los radiólogos.

Nuestro país ha efectuado la compra de equipos médicos de alta tecnología para la obtención de imágenes de alto valor diagnóstico. El sistema Imagis ® se utiliza actualmente en algunos de los hospitales del país para permitir el almacenamiento, reproducción y transmisión de las imágenes médicas. Imagis ® fue desarrollado por el Centro de Biofísica Médica de Santiago de Cuba hace varios años, y presenta varias limitantes, entre ellas, dificultades para procesar algunos tipos de imágenes, por ejemplo las imágenes multiframe. Además no soporta múltiples asociaciones concurrentes, ni tiene la capacidad de mover imágenes desde el servidor hasta los equipos o estaciones asistentes y de post- procesamiento.

A pesar de las condiciones creadas para el desarrollo de la imagenología médica, la necesidad de compartir los estudios entre los hospitales mediante una solución sencilla, ligera y desarrollada, continúa vigente; teniendo en cuenta los niveles de desarrollo tecnológicos con los que cuenta nuestro país en la

¹ Sistema de Almacenamiento y Comunicación de Imágenes

² Sistema de Información Hospitalaria

³ Sistema de Información Radiológica

⁴ Tomografía Axial Computarizada

actualidad. De ahí que el **objetivo general** de la investigación que realizamos sea **desarrollar un sistema que permita guardar y obtener referencias de los estudios imagenológicos realizados en el país, así como de las rutas para la recuperación de estos estudios**; permitiendo a los componentes de un PACS poder efectuar la obtención de dichos estudios.

Analizando lo antes planteado se puede definir que el **objeto de estudio** de la investigación que desarrollamos es la transmisión de información e imágenes de manera estandarizada, en un **campo de acción** que comprende los procesos de transmisión de las imágenes médicas DICOM 3.0⁵ en el Sistema de Imagenología Cubano.

Idea a Defender.

El desarrollo de un sistema que permita localizar todos los estudios radiológicos emitidos desde cualquier localización en el país. Permitiendo a los radiólogos la obtención de cualquiera de los estudios referenciados, y con ello enriquecer el funcionamiento de la Radiología Cubana.

Tareas de la investigación.

Finalmente para darle cumplimiento a los objetivos de la investigación se han trazado las siguientes tareas:

- Investigar la seguridad en Servicios Web.
- Investigar el estado de PostgreSQL.
- Investigación de la transmisión de la información de pacientes según lo planteado por HL7⁶.
- Investigar sobre el funcionamiento actual de los componentes de los sistemas del paquete Cassandra PACS con el objetivo de ver las posibilidades de integración entre ellos.

⁵ Estándar para la visualización, almacenamiento y transmisión de imágenes en medicina.

⁶ Estándar de intercambio de datos electrónicos en el área de la salud

- Elaborar una propuesta de cambio de funcionalidades especificadas para cada sistema del Cassandra PACS.
- Diseñar los sistemas teniendo en cuenta los patrones de diseño que garanticen su fortaleza, reusabilidad y extensibilidad.
- Implementar una aplicación que permita buscar referencias de estudios desde la web.
- Analizar, diseñar e implementar un servicio web para la búsqueda de las referencias.
- Analizar, diseñar e implementar un servicio web para la inserción y actualización de las referencias.
- Analizar, diseñar e implementar un módulo que gestione el intercambio de información entre los hospitales y el modulo de almacenamiento de las referencias de los estudios radiológicos de los pacientes de todo el país.
- Analizar, diseñar e implementar un módulo para el almacenamiento y búsqueda de las referencias de los estudios radiológicos.

El contenido de la investigación se encuentra estructurado en cuatro capítulos:

En el Capítulo 1 se plantean los antecedentes a nivel internacional, nacional, la descripción de los procesos actuales en el marco de la investigación y los conceptos básicos que la justifican; de manera tal que fundamenten el análisis además de la interpretación de los resultados finales.

En el Capítulo 2 se presenta formalmente el problema y la situación problémica, los objetivos estratégicos de la organización desarrolladora junto a las oportunidades de negocio que lo soportan; y el campo de acción con el flujo actual de los procesos involucrados en él. Además, se describe el objeto de automatización, y para ello se explican los procesos que se informatizarán. También se describen los sistemas que ya existen y que se encuentran involucrados en el campo de acción del proyecto. Se realiza una propuesta general de sistema y se analizan las dependencias y relaciones con otros softwares; para ello se especifica el dominio del problema, y se describen formalmente los requerimientos funcionales y no funcionales; finalizando con la especificación de los casos de uso del sistema.

El Capítulo 3 se dedica a explicar detalladamente el desarrollo del Análisis y el Diseño del sistema, para ello se presentan los diagramas de clases de análisis que dan realización a los casos de uso presentados en el capítulo anterior. También se exponen los diagramas clases y de interacción del diseño y se describen las clases Interfaces, Controladoras y Entidades, así como las clases utilizadas en la Base de Datos; de esta se muestran además, los diagramas Entidad Relación de las dos bases de datos que se implementan, una para guardar las referencias de los estudios radiológicos, y otra para guardar los registros de búsquedas.

El Capítulo 4 presenta la distribución física de los componentes que constituyen el sistema. Primero se explica el diagrama de componentes y luego el diagrama de despliegue, donde se demuestra como se distribuirán físicamente los componentes, además de las relaciones de comunicación que existirán entre los nodos.

Capítulo 1: Fundamentación teórica.

En el presente capítulo se plantean los antecedentes a nivel internacional y nacional de los sistemas para la localización y obtención de estudios imagenológicos; la descripción de los flujos de los procesos actuales en el marco de la investigación y los conceptos básicos que la justifican. Todo lo cual, contribuye a fundamentar el análisis del problema y a la interpretación de los resultados finales.

1.1 Estado del arte de las soluciones para la localización y la obtención de estudios radiológicos fuera de una red local.

A nivel Mundial.

En el Mundo se han implementado pocas alternativas con vistas a resolver el problema que se trata de solucionar con la presente investigación. En la actualidad, Canadá cuenta con una alta tecnología imagenológica, una población rala y redes de banda ancha que les ha permitido la instalación de potentes servidores de imágenes localizados regionalmente. Otro intento por encontrar una solución; fue en 1997 en este mismo país, donde se trató de utilizar entre otros el sistema Iris, para el tema del archivado y la restitución en pantalla de imágenes radiológicas. Desarrollado en Suiza y sobre Linux, el sistema permitía centralizar por paciente todos los eventos radiológicos. En principio, los objetivos eran los mismos y se logró con la implementación de los servidores Linux quienes se ocupaban de recuperar las imágenes, archivarlas y servir las a las estaciones de visualización que las necesitaran. Sin embargo no existen registros del éxito de la aplicación de dicho sistema.

A nivel nacional.

Nuestro país no cuenta actualmente con un sistema que permita a los radiólogos localizar y obtener estudios imagenológicos fuera de una red local; sin embargo se está llevando a cabo una alternativa para

facilitar el almacenamiento y la recuperación de los estudios radiológicos. Para ello se han adquirido equipos médicos de última generación, y se ha desarrollado el sistema Cassandra PACS.

El Grupo de Procesamiento de Imágenes y Señales (GPI) de la Universidad de las Ciencias Informáticas está al frente del desarrollo del Cassandra PACS. Además, ha asumido la tarea de desarrollar un sistema informático para dar respuesta a las necesidades de la constitución de una Red Nacional de Imágenes (RNI⁷) que resolvería los problemas de la transmisión radiológica ínter-hospitalaria, utilizando para ello los componentes ya desarrollados del Cassandra PACS.

El Cassandra PACS actualmente consta de cuatro módulos:

Cassandra Viewer.

El visor de imágenes es una aplicación de escritorio desarrollada sobre tecnología .NET con conexión a bases de datos y servidores remotos. Cuenta con capacidad de lectura actualizada y con un equipo de desarrollo amplio y estable para garantizar posteriores adecuaciones de visualización y manipulación establecidas por el estándar DICOM 3.0. Además recoge casi toda la experiencia de usuarios de aplicaciones y sistemas PACS similares a este corroborado a partir de la experiencia de nuestros especialistas de esta área de la salud.

Entre sus características fundamentales podemos mencionar:

- Visualización de imágenes DICOM 3.0.
- Visualización 3D.
- Paletas etiquetadas de colores.
- Paletas etiquetadas de brillo y contraste.
- Filtros.
- Bandeja de Casos.
- Reportes.

⁷ Red Nacional de Imágenes

Cassandra Server.

Es un servidor dedicado al almacenamiento de las imágenes que se encuentren en formato DICOM 3.0. Permite el manejo simultáneo de múltiples asociaciones y agrega el uso de un gestor de bases de datos PostgreSQL para proporcionarle una mayor velocidad de búsqueda de los datos almacenados en el mismo. Además, posee un modo de funcionamiento automático como servicio, permitiendo que sólo se requiera la intervención del administrador en caso de cambios de configuración. Así mismo, da respuestas múltiples configurables ante crisis de capacidad, y un uso administrado de la memoria RAM. También, da solución al problema existente con la recepción y almacenamiento de imágenes multiframe, ya que los servidores hasta este momento en uso no eran capaces de soportarlas.

Cassandra DMail.

DMail es un sistema que constituye una alternativa a la transmisión de imágenes médicas conforme al estándar DICOM 3.0, brindando para ello una interfaz de correo electrónico; aunque la transmisión no se efectúa por SMTP. Además, cuenta con herramientas que posibilitan la anonimización y compresión de los estudios médicos, si el usuario así lo decide. Todo esto se realiza de manera simple y amigable con el objetivo de abstraer a los usuarios de las complejidades de la transmisión conforme al estándar.

DMail brinda todas las funcionalidades básicas de un sistema de su tipo, dígase la creación, mantenimiento, actualización y búsqueda sobre una libreta de contactos, todo el proceso de gestión de mensajes, recibidos, enviados, eliminados, etc.

Uno de los logros fundamentales de este sistema, es que no necesita por parte de los médicos de un conocimiento medio o avanzado en de técnicas de la rama de la informática y las telecomunicaciones, pues la solución es en extremo sencilla y forma parte del que hacer cotidiano de cualquier profesional de la actualidad.

DMail Server:

DMail Server es un servidor de intercambio que debe ser instalado en aquellos equipos intermedios (Proxy, Servidores, etc.) que permitan la comunicación entre aquellas instituciones que deseen insertarse al sistema.

Dmail Client:

Se basa en una aplicación de escritorio que debe ser instalada en las estaciones de trabajo de aquellas instituciones que deseen insertarse al sistema. Mediante esta aplicación los especialistas podrán gestionar sus mensajes, enviar y recibir imágenes médicas. Siempre acorde al estándar.

Xweb.

Cassandra XWeb es una aplicación Web que permite a los especialistas el acceso a los reportes radiológicos emitidos desde el Cassandra Viewer, y el desarrollo de estudios estadísticos sobre el funcionamiento del servicio médico por modalidad. Facilita desde la Web, la visualización de los reportes, así como la descarga de los mismos en formato pdf.

La solución propuesta tiene entre sus objetivos recuperar la información que permita hacer colaborar a los cuatro componentes del Cassandra PACs con sus similares en otros hospitales, con vista a la constitución de la Red Nacional de Imágenes Médicas.

1.2 Descripción del flujo actual de los procesos de transmisión de las imágenes médicas DICOM 3.0 en el Sistema de Imagenología Cubano.

En la mayor parte de los hospitales cubanos, el almacenamiento y la recuperación de imágenes radiológicas se realiza con el sistema Imagis®. Su funcionamiento está basado en tres módulos fundamentales: el módulo de Visualización, el Módulo de Almacenamiento y el Módulo de Transmisión. Sin embargo, en su implementación el visualizador y el módulo de almacenamiento se encuentran físicamente unidos, lo que lo convierte en un mini PACS. En muchos casos, aunque no debiera se

encuentra instalado ligado a un equipo. Esto provoca islas de información en lugares donde debería existir una red imagenológica única. Aunque Imagis® permite la comunicación entre médicos de diferentes hospitales, no se realiza siguiendo los protocolos DICOM establecidos.

Por otra parte, los estudios imagenológicos se le graban al paciente en un dispositivo de almacenamiento, por lo general un CD-ROM que este debería aportar; además son guardados en el servidor imagenológico del centro hospitalario donde se realice. Si el paciente se traslada, lleva sus estudios hacia la estación hospitalaria donde será atendido y ahí otro especialista valorará los estudios almacenados en el CD-ROM. Sin embargo, se deben tener en cuenta los riesgos que implican estos procedimientos, tales como la pérdida de la información en los dispositivos de almacenamiento o de los propios dispositivos; además de que el estándar DICOM 3.0 no establece este tipo de envío de información. Al analizar el flujo de procesos actual se evidencian los inconvenientes y la urgente necesidad de buscar una alternativa para permitir la transmisión ínter hospitalaria segura de los estudios de los pacientes.

La puesta en funcionamiento de **Cassandra PACS** como nueva solución en el tratamiento de estudios e imágenes radiológicas, dígase un visor: **Cassandra Viewer**, un servidor: **Cassandra Server**, un sistema para la transmisión de imágenes: **DMail**, y un generador remoto de reportes radiológicos: **XWeb**; constituye una gran ventaja para nuestro objetivo, ya que supone la colaboración de cuatro componentes independientes en la formación de uno mucho más poderoso, el **Sistema Nacional de Referencia de Imágenes Radiológicas** y el **Cassandra Orchestrator**; sistemas que permitirá la búsqueda y obtención de estudios radiológicos desde cualquier localización del país.

1.3 Metodología, Técnicas y Herramientas a utilizar para el desarrollo del sistema.

Es importante resaltar que la solución propuesta surge después de analizar las diferentes implementaciones de sistemas de este tipo a nivel internacional y de ajustar esas ideas a la infraestructura tecnológica de nuestro país. Con la esperanza que pueda ser aplicable no sólo en Cuba, sino en otros países con condiciones similares. Para su desarrollo se realizó un estudio de las posibles metodologías, técnicas y herramientas a utilizar, teniendo en cuenta la tendencia actual y las potencialidades de cada una de ellas.

1.3.1 Metodología de desarrollo RUP (Rational Unified Process).

Durante el proceso de desarrollo de software se empleó la metodología RUP, por su flexibilidad y adaptabilidad a las condiciones de desarrollo presentadas.

RUP hace énfasis en la adopción de las mejores prácticas del desarrollo de software, lo que permite reducir los riesgos inherentes al desarrollo de una nueva aplicación; de esta manera logramos resultados más predecibles unificando al equipo con procesos comunes que mejoran la comunicación y crean un entendimiento de todas las tareas y responsabilidades.

Las características más importantes de RUP son el desarrollo **iterativo e incremental, dirigido por los Casos de Uso** y que es **centrado en la Arquitectura**; estas garantizan el enfoque del producto final hacia las necesidades reales y concretas del cliente. Este obtiene el producto que satisface todas sus necesidades.

1.3.2 Lenguaje de Modelado UML (Lenguaje Unificado de Modelación).

Actualmente UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en el mundo. Implementa un modelado común para todos los desarrollos, por lo que se crea una documentación también común que cualquier desarrollador con conocimientos del mismo podría entender, independientemente del lenguaje utilizado para el desarrollo.

UML se utiliza para especificar, visualizar, construir y documentar artefactos del sistema de un Software; lo que permite entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Además, capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, y modela una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado permite unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML 2.0

En las versiones previas del UML, se hacía un fuerte hincapié en que no era un lenguaje de programación, un modelo creado mediante UML no podía ejecutarse. En el UML 2.0, esta asunción cambió drásticamente y se modificó el lenguaje, de manera tal que permitiera capturar muchos más comportamientos. De esta forma, permite la creación de herramientas que soporten la automatización y generación de código ejecutable, a partir de modelos UML. (3)

Con ello se logra una diferencia con las versiones anteriores y las herramientas de modelado, abriendo así el paso a algunas como el Enterprise Architect construido sobre las bases específicas de UML.

UML 2.1

Estándar definido por la OMG (Object Management Group), proporciona acceso a los 13 diagramas definidos por UML 2.0, a las Tecnologías MDA (Model Driven Architecture), y un generador de documentación amplio, para ayudarlo a comunicarse y compartir sus modelos eficientemente.

1.3.3 Herramienta de modelado Enterprise Architect.

Enterprise Architect 6.5 (EA) es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. (4)

Es una herramienta que abarca integralmente el ciclo de vida, cubriendo el desarrollo de software desde el relevamiento de los requerimientos, a través de las etapas de análisis, modelos de diseño, prueba y finalmente el mantenimiento y re-uso.

EA es utilizado para el desarrollo de varios tipos de software para un amplio rango de industrias, incluyendo: bancos, desarrollo web, ingeniería, finanzas, medicina, investigación, educación, transporte, ventas, energía, ingeniería electrónica y muchas más. También es utilizado con efectividad para el

entrenamiento en UML y arquitecturas de negocio en empresas de entrenamiento y universidades alrededor del mundo.

1.3.4 Plataforma de desarrollo .Net.

.NET es una plataforma de software que conecta información, sistemas, personas y dispositivos. Comunica además, una gran variedad de tecnologías de uso personal y de negocios, de teléfonos celulares a servidores corporativos, todo esto permitiendo el acceso a información importante, en el momento que se necesiten.

Desarrollado con base en los estándares de Servicios Web XML, .NET permite que los sistemas y aplicaciones, ya sea nuevos o existentes, conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo.

Dentro de las ventajas que reporta el .NET Framework se pueden destacar las siguientes:

- **Código administrado:** El CLR realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- **Interoperabilidad multilinguaje:** El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL).
- **Compilación just-in-time:** El compilador JIT incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- **Seguridad de acceso al código:** Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente del Web sin tener que preocuparse si esto va a estropear el sistema.

- **Despliegue:** Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones. (5)

1.3.5 Visual C#.

Es un lenguaje orientado a objetos simple, elegante y con seguridad en el tratamiento de tipos, que permite crear una gran variedad de aplicaciones. También, proporciona la capacidad de generar componentes de sistema duraderos en virtud de las siguientes características:

- Total compatibilidad entre COM y plataforma para integración de código existente.
- Gran robustez, gracias a la recolección de elementos no utilizados (liberación de memoria) y a la seguridad en el tratamiento de tipos.
- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- Plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas.
- Compatibilidad con XML para interacción con componentes basados en tecnología Web.

C# es uno de los lenguajes de mayor dominio en la comunidad de programadores, sencillo y potente, que combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi; permitiendo con ello, a los desarrolladores, la creación de una gran variedad de aplicaciones. (6)

1.3.6 Servicios Web.

La posibilidad de compartir información o ponerla en un lugar para que otros accedan a ella es interés de la mayoría de las aplicaciones que se desarrollan en la actualidad.

Un servicio Web es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, los sistemas pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. (7)

Los servicios Web se basan en HTTP sobre TCP (Transmission Control Protocol) haciendo uso del puerto 80. Debido a que las organizaciones protegen sus redes mediante firewalls que filtran y bloquean gran parte del tráfico de Internet, se cierran casi todos los puertos TCP salvo el 80, que es el que usan los navegadores. Los servicios Web se comunican por este puerto aprovechando que no resultan bloqueados. Permiten crear buenas interfaces para acceder a las funcionalidades de otros ordenadores en red, desde antes de que existiera SOAP.

Además, son muy prácticos ya que aportan gran independencia entre la aplicación que usa el Servicio Web y el propio servicio. De esta forma, los cambios en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, debido al aumento de la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños.

1.3.7 Arquitectura Orientada a Servicios. SOA.

La arquitectura SOA se ha expandido por la mayoría de los sistemas de gestión de la actualidad. Precisamente, por estar orientada a servicios SOA es muy utilizada por los diseñadores que buscan la agilidad y robustez de su infraestructura.

La arquitectura orientada a servicios (SOA) permite una conectividad flexible de las aplicaciones o recursos representando toda aplicación o recurso en forma de servicio con una interfaz estandarizada. Lo que hace posible el intercambio de información estructurada sobre los eventos del negocio (incluidos los mensajes, los documentos y los 'objetos del negocio') con rapidez y flexibilidad. Esta flexibilidad permite que las aplicaciones nuevas junto a las existentes se combinen con facilidad y rapidez, para responder a las necesidades cambiantes del negocio, y a la capacidad de combinar las aplicaciones. El resultado es la capacidad de desbloquear, integrar y compartir información en grupos de aplicaciones muy aislados, cuyo

rendimiento es bueno dentro de su propio dominio, pero que anteriormente no podían compartir sus datos en toda la empresa. (8)

1.3.8 Patrón de Arquitectura BROKER. Object Request Broker.

El patrón BROKER se utiliza para estructurar sistemas software distribuidos con componentes desacoplados que interaccionan por invocaciones de servicio remotas, donde el broker es el componente responsable de coordinar la comunicación. Diversas plataformas como Microsoft OLE 2.x (Object Linking and Embedding), IBM SOM/DSOM (System Object Model / Distributed System Object Model) o CORBA, comparten este patrón arquitectónico.

Este patrón se utiliza en el contexto de un sistema distribuido, y posiblemente heterogéneo, con componentes independientes que cooperan. Construir sistemas software complejos, mediante un conjunto de componentes desacoplados e ínteroperativos, en lugar de aplicaciones monolíticas; ofrece flexibilidad, mantenibilidad y capacidad de cambio. Así, al repartir la funcionalidad entre un conjunto de componentes independientes, se logra que un sistema potencialmente distribuido y escalable.

Utilizando el patrón BROKER, una aplicación puede acceder a los servicios distribuidos enviando un mensaje al objeto apropiado, en vez de centrarse en las comunicaciones de bajo nivel entre procesos. La arquitectura broker ofrece una gran flexibilidad, permitiendo cambios, incorporaciones y eliminaciones de objetos de forma dinámica. (9)

1.3.9 El protocolo SOAP.

Actualmente, en materia de protocolos de comunicación hay dos grandes tendencias: XML-RPC y SOAP; sin embargo, la mayoría de los Servicios Web públicos, están definidos bajo estándares SOAP.

SOAP(Protocolo de acceso a objetos simples) es un lenguaje de mensajería basado en XML, estandarizado por consorcio W3C, que especifica todas las reglas necesarias para ubicar los Servicio Web XML, integrados en aplicaciones y para establecer la comunicación entre ellos. (10)

El SOAP es un protocolo ligero que permite intercambiar información en un ambiente descentralizado y distribuido. Consiste en tres partes: un sobre que define un marco para describir cuál es en un mensaje y cómo procesarlo, un sistema de las reglas de codificación para expresar casos de uso-definidos, y una convención para representar llamadas y respuestas alejadas del procedimiento. El SOAP está creado con la idea de dar un soporte completo y minucioso a todo tipo de Servicio Web.

1.3.10 .Net Remoting.

Microsoft® .NET Remoting proporciona un marco variado y extensible para que los objetos de distintos dominios de aplicaciones, procesos y equipos se puedan comunicar entre sí sin problemas. Los servicios remotos .NET ofrecen un modelo de programación sencillo, pero muy eficaz, así como compatibilidad en tiempo de ejecución que permite que estas interacciones se realicen de forma transparente. (11)

Las aplicaciones en la plataforma .NET corren en su proceso dentro de su propio espacio de memoria. Cada aplicación tiene la capacidad de exponer objetos al mundo exterior; desde una simple aplicación de consola, una Windows Forms, una residente en Internet Information Server, o un servicio del sistema operativo Windows.

Microsoft .NET Remoting permite a las aplicaciones interactuar con otras más allá de sus dominios. Podemos decir que Remoting permite invocar métodos y pasar objetos más allá de los dominios de la aplicación. (12)

Microsoft® .NET Remoting es un excelente marco que permite compartir objetos entre diversas aplicaciones de ambientes distribuidos. Su versatilidad y facilidad de uso brinda una alternativa frente a DCOM (Distributed Component Object Model), evitando las complicaciones de configuración y comunicación a través de firewalls.

1.3.11 Visual Studio 2005 Team System.

Visual Studio 2005 Team System es uno de los mejores IDE⁸ de desarrollo que se utilizan en la actualidad. Es una plataforma de herramientas del ciclo de vida del desarrollo de software extensible, integrado y productivo; que ayuda a los equipos desarrolladores mediante la mejora de las comunicaciones y la colaboración durante todo el proceso productivo. Además, aporta las siguientes guías:

- Documentación de Team Foundation es un servidor de colaboración de equipo extensible que proporciona seguimiento de elementos de trabajo, control de código fuente, información e instrucciones sobre el proceso.
- Documentación de Team Edition para Architects es un conjunto de herramientas de diseño de aplicaciones integradas para el desarrollo de servicios.
- Documentación de Team Edition para Developers proporciona herramientas de calidad del código y rendimiento que permiten a los equipos generar servicios y aplicaciones confiables y críticos.
- Documentación de Team Edition para Testers proporciona herramientas avanzadas de prueba de carga que permiten a los equipos comprobar el rendimiento de las aplicaciones antes de su implementación. (13)

1.3.12 PostgreSql.

PostgreSql cuenta en la actualidad con más de 15 años de desarrollo activo y una arquitectura probada. Es un motor de bases de datos robusto y fácil de administrar, aplicable a sistemas multiplataformas. Proporciona soporte a SQL92/SQL93, integridad en transacciones y capacidad para extensión de tipos. (14)

Además, PostgreSql posee características importantes como Control de Concurrencia Multi-Versiones (MVCC⁹), recuperación en puntos de tiempo, espacios de tablas, replicación asíncrona, planificador-

⁸ Integrated Development Environment

⁹ Multi-Version Concurrency Control

optimizador de consultas, entre otras. Es altamente escalable, tanto en la cantidad de datos que puede manejar como en la cantidad de usuarios concurrentes que soporta.

En este capítulo, con el objetivo de fundamentar el desarrollo del sistema propuesto, primeramente se realizó el análisis de las condiciones actuales tanto internacionales como nacionales. Se estudiaron y presentaron los sistemas desarrollados en nuestra universidad sobre la transmisión de estudios radiológicos interhospitalarios. Se explicó brevemente, el flujo de los procesos actuales con el objetivo de poner en evidencia el problema de la investigación y la necesidad de hallarle una solución. Además, se fundamentó la selección de las tecnologías y herramientas de modelación y desarrollo utilizadas.

Capítulo 2: Características del sistema.

En este capítulo se presenta formalmente el problema y la situación problémica, los objetivos estratégicos de la organización desarrolladora junto a las oportunidades de negocio que lo soportan. Además, se describe el objeto de automatización, y para ello se explican los procesos que se informatizarán. También se describen los sistemas que ya existen y que se encuentran involucrados en el campo de acción del proyecto. Se realiza una propuesta general de sistema y se analizan las dependencias y relaciones con otros softwares; para ello se especifica el **Dominio** del problema, y se describen formalmente los requerimientos funcionales y no funcionales; finalizando con la especificación de los casos de uso del sistema.

2.1 Problema y situación problémica.

En la actualidad se necesitan resolver los problemas existentes con la **transmisión de los estudios** entre los médicos especialistas en Imagenología.

Los médicos no tienen forma de acceder a estudios anteriores realizados a los pacientes en otras localizaciones; estos transportan sus propias imágenes (en el mejor de los casos) en CD-ROM. Lo que constituye un problema, teniendo en cuenta las pérdidas que presuponen y que no todos los pacientes poseen un dispositivo de almacenamiento adecuado para transportar sus imágenes, o no conocen la necesidad de llevarlo al realizarse el estudio; además de que la lejanía y la transportación también representan limitantes. De ahí que se haya formulado el **problema** que da origen a esta investigación como la siguiente interrogante: **¿Cómo obtener una referencia de todos los estudios imagenológicos existentes en el país?** Permitted con la colaboración de los cuatro componentes del Cassandra PACs la localización y transmisión de los estudios de los pacientes.

2.1.1 Objetivos estratégicos de la organización y oportunidades de negocio que los secundan.

Con el objetivo de elevar la calidad de vida de la población se ha efectuado la compra de equipos médicos imagenológicos de última generación. La UCI se ha visto inmersa en el desarrollo de un sistema capaz de soportar y permitir el funcionamiento íntegro de dichos equipos para lograr la transmisión, almacenamiento y diagnóstico de los estudios imagenológicos. El sistema Cassandra PACS se encuentra actualmente instalado en varios hospitales de nuestro país y en algunas clínicas en Venezuela funcionando de manera estable.

Sin embargo, en nuestro país no existe ningún sistema que permita conocer la ubicación exacta de las imágenes médicas archivadas en los servidores distribuidos en todo el país. Además, tampoco existe alguno que pudiera comprarse debido a la escasa existencia de este tipo de sistemas a nivel internacional y a que están desarrollados para países tecnológicamente más avanzados.

2.1.2 Flujo actual de los procesos involucrados en el campo de acción.

Actualmente en todo el país la transmisión de los estudios médicos no se realiza automáticamente sino por las partes interesadas, pacientes y médicos, en dispositivos externos de almacenamiento, por ejemplo CD-ROM. Estos se queman en las propias estaciones técnicas, lo que conlleva pérdidas de tiempo y recursos.

2.1.3 Análisis crítico de los procesos de localización y transmisión de los estudios.

Debido a que los médicos no tienen forma de acceder a los estudios imagenológicos realizados con anterioridad a sus pacientes en otras localizaciones; los propios pacientes transportan manualmente sus estudios en dispositivos como CD-ROM. Esto supone riesgos de pérdida de los estudios, por no ser un medio adecuado y seguro para la transmisión; además de que la pérdida de tiempo es muchas veces determinante, así como el gasto de recursos que representa.

2.2 Objeto de automatización.

2.2.1 Descripción de los procesos que serán objeto de automatización.

Actualmente el módulo de Cassandra PACS DMail permite la transmisión de los estudios imagenológicos entre hospitales, pero con la supervisión por ambas partes de médicos especialistas. A su vez, desde las estaciones de diagnóstico Cassandra Viewer se pueden hacer búsquedas y obtención de estudios del servidor local del hospital.

Se desea automatizar todo el proceso de **localización de los estudios** de los pacientes que se encuentran en lugares inaccesibles por parte de los médicos en el momento de la consulta y **gestionar la obtención de los mismos**.

2.2.2 Descripción de los sistemas automatizados que existen vinculados con el campo de acción del proyecto.

Cassandra Server.

Permite gestionar la transmisión, el almacenamiento y la disponibilidad de las imágenes médicas dentro de una red imagenológica.

Dmail

Permite la transmisión de imágenes médicas dentro y fuera de una red local, brindando una interfaz de correo al médico, que lo abstrae de toda la complejidad de la transmisión DICOM.

Cassandra Viewer

Herramienta para el diagnóstico de estudios imagenológicos que garantiza la visualización y manipulación de las imágenes generadas por los equipos médicos.

Xweb

Permite acceder a los reportes de diagnóstico emitidos desde el Cassandra Viewer. Además de llevar estadísticas que permiten auditar el funcionamiento de todo el sistema.

2.3 Información que se maneja.

Entre los documentos que se han procesado con la investigación se encuentran el Documento de Visión y el Glosario de Términos.

El **Documento de Visión** es un artefacto que **define la visión del producto** desde las perspectivas del cliente, especificando sus necesidades y características. Además constituye una **base de acuerdo** en cuanto a negociaciones se refiere, en el se plantean entre otras cosas el **propósito**, el **alcance** del proyecto y las **oportunidades de negocio** existentes.

El **Glosario de Términos** es un documento que **describe** los principales **términos utilizados en el proyecto**. Este documento se adjunta en el [Anexo 1](#).

2.4 Propuesta de sistema.

El sistema debe permitir crear las condiciones para que se efectúe la transferencia de estudios entre hospitales dada una referencia de localización. Se plantea la informatización de un servidor nacional de referencias que se actualizaría con la aparición de cada nuevo estudio y reporte de diagnóstico realizado a un determinado estudio; junto a una aplicación web que permitiría el acceso de los especialistas desde cualquier localización. Además se plantea la implementación de un sistema orquestador que sería capaz

de controlar todos los procesos de solicitud de búsqueda de los estudios por los componentes ya terminados del Cassandra PACS, el Cassandra Viewer, el Cassandra Server y el Cassandra Dmail.

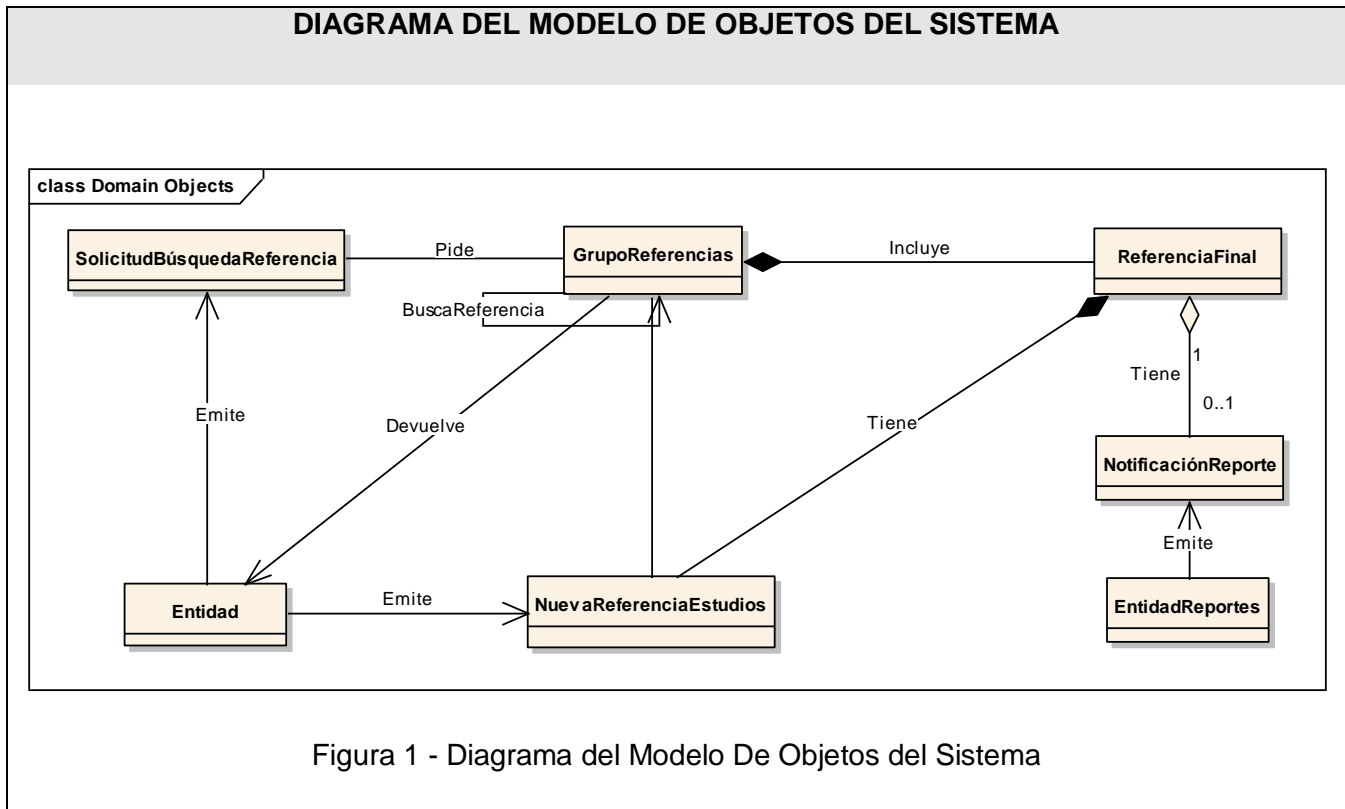
Los sistemas que actualmente existen en el mundo están adaptados para su funcionamiento en condiciones con las que hoy no cuenta nuestro país. Las alternativas estudiadas, como por ejemplo la de Canadá hacen uso de un gran desarrollo tecnológico y los costes del mantenimiento del sistema son muy elevados; lo que convierte al sistema propuesto en una alternativa viable, teniendo en cuenta su adaptabilidad a condiciones tecnológicas no tan desarrolladas y a la inversión que se debería hacer para su despliegue, adecuadas a los intereses y las posibilidades de nuestro país.

2.4.1 Modelo de Dominio.

Las investigaciones previas realizadas al comienzo del desarrollo del sistema no arrojaron los suficientes elementos como para poder describir y modelar un negocio, por lo cual se decidió obtener los conceptos fundamentales y realizar un **Modelo de Dominio**.

El Modelo de Dominio captura los objetos más importantes que existen o los eventos que suceden en el entorno donde estará el sistema. En este caso se definió la **Entidad** como el objeto que emitiría la **Solicitud de Búsqueda**. Esta pide directamente al **Grupo de Referencias** una **Referencia Final**. Es importante resaltar que la **ReferenciaFinal** está compuesta por dos objetos que vienen de lugares diferentes. La **NuevaReferenciaEstudios** que es emitida por la **Entidad**, y la **NotificaciónReporte** que es emitida por la **EntidadReporte**.

2.4.1.1 Diagrama del Modelo de Dominio.



2.4.2 Requerimientos.

Una adecuada obtención y especificación de los requerimientos constituye un factor decisivo en el éxito y mejor utilización del sistema.

2.4.2.1 Dependencias y Relaciones con otros softwares.

La importancia fundamental del sistema propuesto radica precisamente en su carácter sistémico. Es decir, hace que la integración de sus componentes multiplique la utilidad y las potencialidades que tienen por separado cada uno de ellos. Para ello, se hace imprescindible la adaptación de los sistemas existentes a las nuevas funcionalidades y requerimientos.

Las propuestas de cambios de cada uno de los sistemas, se plantea a continuación:

Cassandra Viewer.

Los visores, ubicados en las estaciones de diagnóstico deberán poder recibir las referencias de los estudios solicitados al servidor DICOM, en el caso de que los estudios solicitados no se encontrasen en el, y deberán poder permitirle al especialista conocer dichas referencias (adaptadas para su comprensión), y poder decidir cuales transferir hasta el servidor DICOM local. Después de efectuada la transferencia hacia el servidor este le dará el resultado al visor, y este lo recibirá como si hubiera sido una búsqueda local en el servidor.

Cassandra Server.

El servidor DICOM deberá poderse comunicar con el sistema propuesto para hacerle pedidos de referencias de estudios radiológicos, en el caso de que el estudio solicitado por una estación de diagnóstico Cassandra Viewer no se encontrase en él; y deberá al mismo tiempo poder recibirlas. Además, una vez tenida las referencias, deberá poder transmitírselas a la estación de diagnóstico, desde donde un radiólogo decidirá cual cuales estudios pedir para que sean transferidos. Después de informada la decisión a través del visor y hacia el servidor de Cassandra este deberá poder comunicarse con el servidor de intercambio DMail correspondiente para solicitar la transferencia de el o los estudios radiológicos solicitados, facilitándole la referencia de este; y recibir el o los estudios solicitados una vez transferidos.

Cassandra DMail.

El servidor de intercambio DMail deberá poder recibir solicitudes de transferencia de un estudio radiológico provenientes de un servidor de Cassandra, la solicitud de transferencia contendría la

referencia de localización del estudio a buscar; además deberá poder entregarle a este el estudio después de transferido.

XWeb.

El repositorio de reportes de diagnóstico deberá poder notificarle al sistema cuando a un estudio radiológico se le realice un diagnóstico; así la referencia de este ahora, incluirá información que indique que el estudio en cuestión tiene un reporte de diagnóstico en XWeb. Lo que es de utilidad para el especialista cuando realiza la petición de un estudio desde la estación de diagnóstico y recibe la referencia de este indicando su localización y que tiene un diagnóstico en XWeb que puede revisar.

2.4.2.2 Requerimientos Funcionales.

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea.

Nombre del Requerimiento	Descripción
RF1- Enviar Mensaje.	Es necesario poder enviar mensajes de pedidos y confirmación.
RF2- Recibir Mensaje.	Es necesario poder recibir mensajes de notificación y confirmación.
RF3- Validar Mensaje.	Se necesita validar la procedencia del mensaje para proseguir con la acción solicitada.
RF4- Recibir Referencia.	Se necesita poder recibir los datos de las referencias para redireccionarlas de manera estandarizada y segura a los diferentes destinos.

RF5- Enviar Referencia.	Se hace necesario poder enviar los datos de las referencias de manera estandarizada y segura a los diferentes destinos.
RF6- Guardar Referencia.	Se necesita poder crear una nueva referencia con la ocurrencia de un nuevo estudio.
RF7- Buscar Referencia.	Se necesita poder buscar una referencia.
RF8- Adicionar Reporte a la Referencia.	Se necesita poder realizar cambios sobre una referencia ya existente dado un cambio en el estado del estudio correspondiente.
RF9- Guardar las notificaciones de búsquedas.	Se necesita tener constancia de las búsquedas realizadas por lo que se guardarán notificaciones sobre estas.
RF10- Revisar las notificaciones.	Se necesita poder revisar las notificaciones existentes.
RF11- Buscar una Referencia desde la Web.	Se hace necesario poder buscar una referencia desde la Web.

2.4.2.3 Requerimientos No Funcionales.

Los requerimientos no funcionales describen las características que de una forma u otra limitan al sistema:

Usabilidad.

El sistema deberá ser de fácil manejo utilizando en todo lo posible la experiencia del usuario. Tendrá dos interfaces para interactuar con el usuario, diferentes para los dos componentes, el Servidor de

Referencias que tendrá una interfaz Web, y el Orchestator que tendrá una interfaz Windows sencilla y amigable.

Confiabilidad.

El Servidor de Referencias deberá estar disponible a tiempo completo y de manera estable, estará preparado para soportar alta concurrencia de solicitudes sin que haya caída de los servicios.

El Orchestator deberá funcionar establemente y permitir que las búsquedas de las referencias entre las entidades internas del hospital y el servidor de referencias se efectúen exitosamente.

Seguridad.

El sistema será usado y administrado solamente por trabajadores cuidadosamente designados según su grado de responsabilidad, garantizando la integridad de los procesos que se manejen.

Además el acceso de los usuarios a los sistemas será controlado con nombres de usuario y contraseñas que serán asignadas en dependencia de los privilegios que se les asignen.

Rendimiento.

El sistema debe soportar una alta concurrencia y permitir respuestas rápidas al igual que la velocidad de procesamiento de la información.

Interfaces de Comunicación.

El sistema interactuará con otros que en este caso pertenecen al mismo paquete Cassandra PACs por lo que son compatibles entre ellos. Además el sistema está compuesto por dos subsistemas que implementarán cada uno las interfaces de comunicación adecuadas para permitir las funcionalidades de los subsistemas integrados.

Portabilidad.

El sistema podrá ser utilizado sobre el sistema operativo Windows XP (Recomendado). Recomendamos la adaptación del sistema para permitir su posible ejecución en entorno UNIX, haciendo uso para ello de MONO.

Políticos-culturales.

El sistema actualmente sólo se encuentra implementado en español, pero se planea implementar las versiones correspondientes para otros idiomas. Las interfaces desarrolladas están en correspondencia con el tipo de institución donde será usado.

Legales.

El sistema y toda la documentación generada en su desarrollo pertenecen al Grupo de Procesamiento de Imágenes y Señales de la Universidad de las Ciencias Informáticas.

Software.

- Sistema Operativo Windows XP.
- .NET Framework 2.0 o superior.
- Como motor de base de datos PostgreSQL 8.1 o superior.

Hardware.

Recomendamos como requerimientos mínimos:

Para el servidor donde se instale el sistema orquestador.

- IP real.
- CPU Pentium IV 2,8 GHz.
- Memoria RAM: 512 Mb.
- Adaptador de red: PCI Express 1Gb.

Para el servidor de referencias.

- Memoria RAM: 1 Gb.
- CPU Pentium IV 3,0 GHz.
- RAID 1 ó 5 con discos SCSI de 73 Gb y 10 – 15k RPM.

2.4.3 Descripción general del sistema propuesto.

Teniendo en cuenta que el objetivo fundamental del sistema es permitir la transferencia de estudios entre hospitales dada una referencia de localización de dichos estudios; se plantea la implementación de un servidor nacional de referencias que se actualizaría con la aparición de cada nuevo estudio y de cada nuevo reporte de diagnóstico que se le haga a un determinado estudio, y de una aplicación para el acceso desde la web. Además, la implementación de un sistema orquestador que sería capaz de controlar todos los procesos de solicitud de búsqueda y de transferencia de datos entre los componentes ya terminados del Cassandra PACS: Cassandra Viewer, Cassandra Server, Cassandra Dmail, y XWeb.

A partir de la solicitud de búsqueda de un estudio desde el Cassandra Viewer se inicia una secuencia de eventos que concluye con su obtención. La búsqueda se realiza inicialmente en el servidor Cassandra Server del hospital local, en caso de no encontrarse, el servidor envía una solicitud de búsqueda al Cassandra Orchestator, localizado en un IP real del hospital, quien va a controlar todos los procesos de búsqueda desde el hospital.

El Orchestator, redirecciona la solicitud al Sistema Nacional de Referencia de Imágenes Médicas, que realiza la búsqueda de las referencias del estudio en cuestión; esta referencia contiene la información de la localización del estudio y si este tiene reportes de diagnóstico realizados en XWeb. Una vez completada la búsqueda, el Orchestator devuelve al Cassandra Server el resultado de su solicitud: la referencia.

El Cassandra Server notifica al Cassandra Viewer el resultado de la búsqueda, que estaría constituida por el conjunto de referencias encontradas. En caso de que el médico desee realizar la transferencia del

estudio debe seleccionarlo y ordenar su transferencia. Automáticamente la solicitud de transferencia es emitida hacia el Cassandra Server y este la redirecciona al servidor de intercambio Dmail, localizado donde esté el Cassandra Orchestator y preparado para comunicarse directamente con otro servidor de mismo tipo ubicado en otro hospital.

La solicitud de transferencia es enviada al servidor Dmail receptor, este solicita la transferencia al servidor de Cassandra correspondiente el cual le entrega los datos del estudio solicitados. La transferencia de los datos se efectúa entre los servidores de intercambio Dmail. Luego de la transferencia del estudio, el servidor de intercambio Dmail que se comporta ahora como receptor transfiere lo datos directamente al servidor Cassandra del hospital que inició la solicitud y este directamente transmite la respuesta del estudio al Cassandra Viewer que inició la solicitud.

El sistema deberá permitir la recepción de notificaciones desde XWeb que completen las referencias almacenadas en el servidor de referencia cuando en una estación de diagnóstico, Cassandra Viewer se emita un diagnóstico sobre un estudio determinado y este se almacene en el servidor remoto de reportes, XWeb. La notificación de reporte sobre un estudio determinado se emitirá al servidor de referencias, el cual deberá hacer coincidir la notificación con el estudio en cuestión, con el objetivo de informar cuando se solicite la referencia de un estudio si este tiene o no un reporte de diagnóstico en XWeb.

El sistema deberá permitir la creación de nuevas referencias, ya sean de un nuevo estudio que se almacena, lo cual es informado por el Cassandra Server de cada hospital; o de un estudio que es transferido hacia otro hospital, lo cual no indica que se elimine de su origen, sino que se hace una nueva copia en el otro hospital.

2.4.4 Análisis de otras soluciones existentes.

El problema que trata de resolver esta tesis no ha tenido mucho análisis en el marco mundial, debido a que las posibles soluciones no se corresponden con el interés capitalista predominante en gran parte del resto del mundo; que predicen el consumismo y el enriquecimiento a toda costa. En estos países, no es

de interés obtener imágenes ya realizadas desde otras localizaciones; sino que prefieren volver a hacerlas y cobrar por ello. Sin embargo, países como Canadá con especiales intereses se han propuesto algunas alternativas, como la implantación de servidores regionales de imágenes los cuales son accedidos según las necesidades. A diferencia de Canadá, nuestro país se propone implementar una solución basada en los estudio de las condiciones y tecnologías con las que cuenta y hacerla factible además para países con condiciones similares a las nuestras.

2.5 Definición de los casos de uso.

2.5.1 Definición de los Actores del Sistema a Automatizar.

Nombre del Actor	Descripción
Cassandra Server.	Es el responsable de iniciar los procesos, en el lado del Orchestrator, fundamentales del sistema como insertar una nueva referencia y buscar la referencia de un estudio.
XWeb.	Es el responsable de emitir la notificación del reporte de diagnóstico de un estudio para que esta sea agregada a la referencia del estudio.
Administrador.	Es el rol que representa a las personas que interactuarán con el sistema para revisar las notificaciones que guarde el sistema sobre su funcionamiento.
Orchestrator.	Es el subsistema encargado de iniciar los procesos de insertar y de localizar una nueva referencia. Se generaliza como el actor Subsistema.

Médico.	Es el rol que representa a las personas que interactuarán con el sistema para iniciar los procesos de búsqueda de una referencia desde la Web.
Sistema de Búsqueda de Referencias.	Es el subsistema Web encargado de iniciar la localización de una referencia. Se generaliza como el actor Subsistema.
Subsistema	Es el actor que realiza la funcionalidad de localizar una referencia, se especializa como Orchestator y como Sistema de Búsqueda de Referencias.

2.5.2 Definición de los Casos de Uso del Sistema.

CU-1	Buscar_Referencia.
Actor	Cassandra Server.
Descripción	El Cassandra Server emite un mensaje de solicitud de búsqueda de referencia y el sistema le devuelve la referencia.
Referencia	RF7.

CU-2	Insertar_Referencia.
Actor	Cassandra Server.
Descripción	El Cassandra Server emite una nueva referencia debido a que se almacena un nuevo estudio; la nueva referencia es emitida al sistema, el cual verifica la procedencia del mensaje y en caso valido, el sistema agrega la nueva referencia en el servidor de referencias.
Referencia	RF6.

CU-3	AdicionarReporte_Referencia.
Actor	Xweb
Descripción	XWeb emite un mensaje cuando en él se almacena un nuevo reporte de diagnóstico referente a un estudio. Este mensaje será validado por el sistema y en caso positivo se completará la referencia del estudio dado.
Referencia	RF8.

CU-4	Adicionar_Referencia.
Actor	Orchestator
Descripción	El caso de uso se inicia cuando al sistema llega un mensaje de inserción de una referencia. El sistema busca el estudio correspondiente y le asigna la referencia.
Referencia	RF3, RF1, RF2, RF4, RF5.

CU-5	Localizar_Referencia.
Actor	Susbsistema.
Descripción	Al sistema llega un mensaje de solicitud de búsqueda. Este efectúa la búsqueda y devuelve un resultado.
Referencia	RF7

CU-6	Guardar_Registro_Búsqueda.
Actor	
Descripción	Cuando se realice una búsqueda en el sistema y se obtengan los resultados de la misma el sistema guardará un reporte de que se realizó dicha búsqueda y si se encontraron o no resultados.
Referencia	RF9

CU-7	Visualizar_Registro_Búsqueda
Actor	Administrador
Descripción	El actor buscará los registros que sobre las búsquedas se hayan realizado, esto le permitirá auditar el funcionamiento del sistema.
Referencia	RF10

CU-8	Búsqueda Web de Referencias
Actor	Médico
Descripción	El actor accede a la página de búsqueda de referencias y luego introduce los datos que identifican al paciente e inicia la búsqueda. El sistema muestra primero los pacientes encontrados y el actor selecciona uno y ve sus estudios, luego selecciona el estudio y ve la localización de este. Si lo desease el actor podría seleccionar otro estudio u otro paciente, y el flujo se repetiría.
Referencia	RF11

2.5.3 Diagrama de Casos de Uso del sistema.

Los casos de uso se encuentran separados en tres paquetes, el Orchestator, el SNRIM y el Sistema de Búsqueda de Referencias ya que responden a requerimientos con intereses en lugares diferentes.

DIAGRAMA DE CASOS DE USO DEL SISTEMA

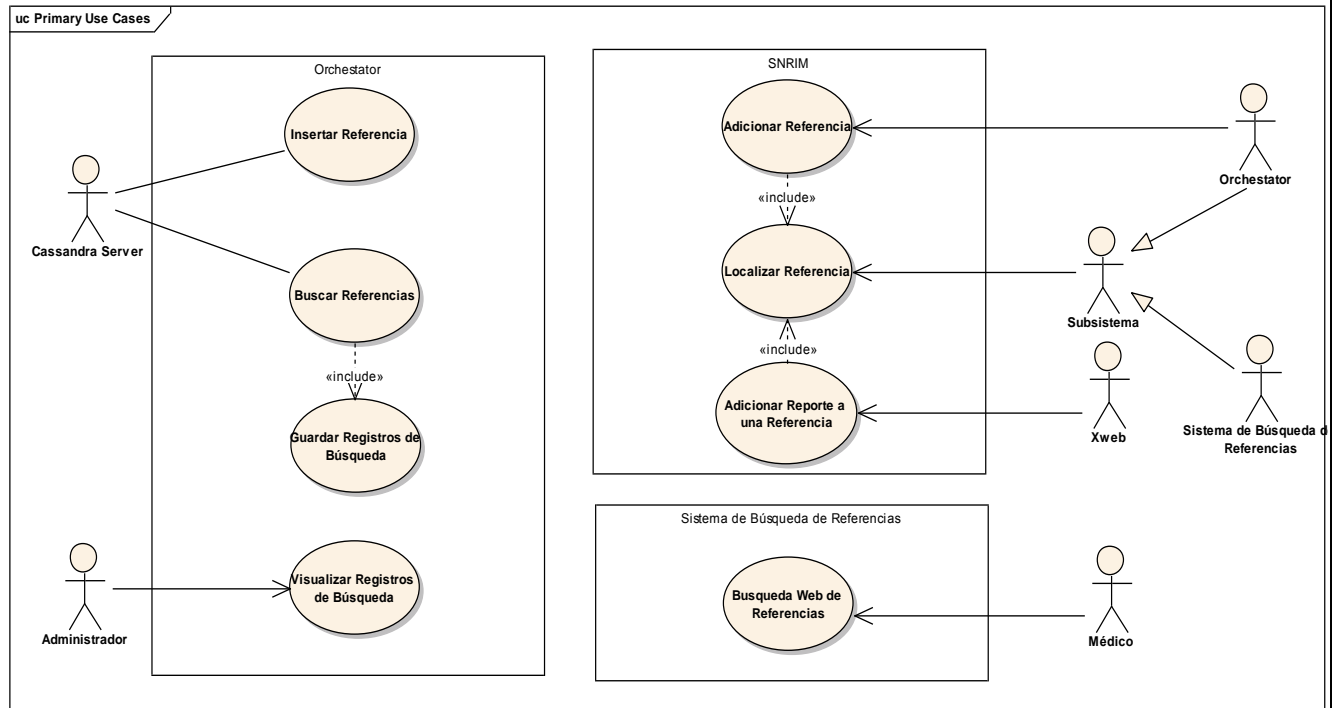


Figura 2 – Casos de Uso del Sistema

2.5.4 Descripción de los Casos de Uso.

1. Buscar_Referencia.

Caso de uso	
Código del Caso de Uso: Buscar_Referencia	
Tipo: Primario.	
Propósito	Buscar la referencia de un estudio determinado almacenada por el sistema en un servidor de referencias.
Actores: Cassandra Server.	
Resumen: El Cassandra Server emite un mensaje de solicitud de búsqueda de referencia y el sistema le devuelve la referencia.	
Referencias	RF6
Precondiciones	
Postcondiciones	Poder tener la referencia buscada.
Acción del actor	Respuesta del sistema
1. El actor envía una solicitud de búsqueda al sistema.	1.1- El sistema recibe la solicitud de búsqueda.
	1.2- El sistema valida la solicitud recibida.
	1.3- El sistema inicia las acciones del CU localizar_Referencia. Ver CU Localizar_Referencia
	1.4- El sistema guarda la solicitud de búsqueda y

	<p>confirma los resultados de la misma.</p> <p>Ver el CU Guardar_Registro_Búsqueda.</p>
	<p>1.5- El sistema devuelve los resultados de la búsqueda.</p>

2. Insertar_Referencia.

Caso de uso	
Código del Caso de Uso: Insertar_Referencia	
Tipo: Primario.	
Propósito	Insertar la nueva referencia de un nuevo estudio que se almacene en un servidor de imágenes.
Actores: Cassandra Server.	
Resumen: El Cassandra Server emite una nueva referencia debido a que se almacena un nuevo estudio; la nueva referencia es emitida al sistema, el cual verifica la procedencia del mensaje y en caso valido, el sistema agrega la nueva referencia en el servidor de referencias.	
Referencias	RF6
Precondiciones	
Postcondiciones	Permite dar inicio al caso de uso Adicionar_Referencia, mediante el que se guarda la referencia.
Acción del actor	Respuesta del sistema
1- Un servidor de Cassandra Server emite una	1.1- El sistema recibe un mensaje con la nueva

nueva referencia sobre un nuevo estudio que ha sido almacenado en este.	referencia.
	1.2- El sistema valida la procedencia del mensaje.
	1.3- El sistema redirecciona el mensaje con la nueva referencia para guardarla.
	1.4- El sistema inicia el Caso de Uso Adicionar_Referencia con el cual se insertará la nueva referencia sobre el estudio en cuestión.
	1.5- El sistema informa al usuario que la nueva referencia ha sido adicionada con éxito.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.2- El sistema valida negativamente el mensaje.
	1.3- El sistema informa al actor que el mensaje no fue procesado y especifica el error.

3. AdicionarReporte_Referencia.

Caso de uso	
Código del Caso de Uso: AdicionarReporte_Referencia.	
Tipo: Primario.	
Propósito	Agregar a la referencia de un estudio una marca que confirme si este tiene o no reporte de diagnóstico en XWeb.

Actores: XWeb.	
Resumen: XWeb emite un mensaje cuando en él se almacena un nuevo reporte de diagnóstico referente a un estudio. Este mensaje será validado por el sistema y en caso positivo se completará la referencia del estudio dado.	
Referencias	RF8
Precondiciones	
Postcondiciones	Poder conocer si un estudio referenciado tiene o no reporte de diagnóstico en XWeb.
Acción del actor	Respuesta del sistema
1- El actor Xweb envía una notificación de reporte de diagnóstico sobre un estudio que está referenciado en el sistema.	1.1- El sistema recibe la notificación de reporte.
	1.2- El sistema valida la procedencia del mensaje.
	1.3- El sistema localiza la referencia del estudio reportado. Ver CU Localizar_Referencia.
	1.4- El sistema agrega a la referencia el dato sobre si el estudio en cuestión está reportado o no.
	1.5- El sistema notifica que la referencia fue marcada con éxito.

4. Adicionar_Referencia.

Caso de uso
Código del Caso de Uso: Adicionar_Referencia.

Tipo: Primario.	
Propósito	Adicionar una referencia en el servidor de referencias.
Actores: Orchestator	
Resumen: El caso de uso se inicia cuando al sistema llega un mensaje de inserción de una referencia. El sistema busca el estudio correspondiente y le asigna la referencia.	
Referencias	
Precondiciones	
Postcondiciones	Permite poder buscar después la referencia insertada.
Acción del actor	Respuesta del sistema
1- El actor Orchestator envía un mensaje con una nueva referencia.	1.1- El sistema recibe el mensaje con la nueva referencia.
	1.2- Busca si existe el estudio correspondiente a la referencia recibida. Ver CU Localizar_Estudio.
	1.3- El sistema no encuentra el estudio correspondiente a la referencia.
	1.4- El sistema crea un nuevo estudio y le asigna la referencia recibida.
	1.5- El sistema notifica que la referencia fue agregada con éxito.

5. Localizar_Referencia.

Caso de uso	
Código del Caso de Uso: Localizar_Referencia.	
Tipo: Primario.	
Propósito	Buscar la o las referencia en el conjunto de referencias.
Actores: Subsistema.	
Resumen: Al sistema llega un mensaje de solicitud de búsqueda. Este efectúa la búsqueda y devuelve un resultado.	
Referencias	RF7.
Precondiciones	Que se haga la solicitud de búsqueda de la referencia.
Postcondiciones	Permite al caso de uso Buscar_Referencia cumplir su finalidad.
Acción del actor	Respuesta del sistema
1- El actor Orchestator envía un mensaje con una solicitud de búsqueda.	1.1- El sistema localiza la o las referencias del estudio a buscar según los parámetros de la búsqueda.
	1.2- El sistema devuelve las referencias encontradas.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1- El sistema no encuentra resultados que coincidan con los datos de la búsqueda.

6. Guardar_Registro_Búsqueda.

Caso de uso	
Código del Caso de Uso: Guardar_Registro_Búsqueda.	
Tipo: Primario.	
Propósito	Guardar reportes sobre las solicitudes de búsquedas que se realicen y si sobre estas se encuentran resultados.
Actores:	
Resumen: Cuando se realice una búsqueda en el sistema y se obtengan los resultados de la misma el sistema guardará un reporte de que se realizó dicha búsqueda y si se encontraron o no resultados.	
Referencias	RF9
Precondiciones	Que se emita una solicitud de búsqueda y que se reciban lo resultados de la misma.
Postcondiciones	Poder buscar después los registros sobre las búsquedas que se hayan efectuado.
Acción del actor	Respuesta del sistema
	1.1- El sistema recibe una notificación sobre una búsqueda realizada y si sobre ella se encontraron resultados o no.
	1.2- El sistema guarda un reporte sobre la notificación recibida.

7. Visualizar_Registro_Búsqueda.

Caso de uso	
Código del Caso de Uso: Visualizar_Registro_Búsqueda.	
Tipo: Primario.	
Propósito	Buscar los eventos que sobre las búsquedas se hayan registrado.
Actores: Administrador.	
Resumen: El actor buscará los registros que sobre las búsquedas se hayan realizado, esto le permitirá auditar el funcionamiento del sistema.	
Referencias	RF10
Precondiciones	Que se hayan efectuado anteriores búsquedas.
Postcondiciones	Permite auditar el sistema.
Acción del actor	Respuesta del sistema
1- El Administrador accede a la interfaz del sistema inicia la búsqueda de los registros relacionado a las búsquedas ocurridas en el sistema. Para ello introduce una fecha o un rango de fecha.	1.1- Valida los datos de la búsqueda
	1.2- Busca los registros según los parámetros pasados por el actor.
	1.3- Devuelve los registros de búsquedas encontrados.

8. Búsqueda Web de Referencias.

Caso de uso	
Código del Caso de Uso: Búsqueda Web de Referencias.	
Tipo: Primario.	
Propósito	Buscar las referencias de un estudio mediante la especificación de los diferentes parámetros de búsqueda.
Actores: Médico.	
Resumen: El actor accede a la página de búsqueda de referencias y luego introduce los datos que identifican al paciente e inicia la búsqueda. El sistema muestra primero los pacientes encontrados y el actor selecciona uno y ve sus estudios, luego selecciona el estudio y ve la localización de este. Si lo desease el actor podría seleccionar otro estudio u otro paciente, y el flujo se repetiría.	
Referencias	RF11
Precondiciones	Que el actor haya accedido al sistema de búsquedas mediante otro que garantice su autenticación.
Postcondiciones	
Acción del actor	Respuesta del sistema
1- El Médico accede a la página búsqueda de referencias e introduce los datos del paciente a buscar.	1.1- El sistema busca todos los pacientes que coincidan con los criterios de búsqueda especificados. Y los muestra al actor
2- El Médico selecciona un paciente para ver sus estudios.	2.1- En ese momento el sistema busca todos los estudios correspondientes al paciente seleccionado. Y los muestra al actor.

3- El Médico selecciona un estudio específico entre el conjunto mostrado.	3.1- El sistema busca todas las localizaciones del estudio seleccionado por el actor y se los muestra.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1- El sistema no encuentra resultados para lo criterios de búsqueda especificados. Finalizando así el caso de uso.

En este capítulo se estableció formalmente la construcción del sistema propuesto, mediante la descripción del problema y la situación problemática. Se describió el objeto de automatización y se realizó una propuesta del sistema teniendo en cuenta el momento de desarrollo que explica el capítulo, y para ello se describieron el Modelo de Dominio y el Modelo de Casos de Uso, así como los requerimiento funcionales y no funcionales que para definir los casos de uso hubo que determinar.

Capítulo 3: Análisis y diseño del sistema.

En el presente capítulo se explica primeramente el tipo de arquitectura implementada y luego se desarrollan el Análisis y el Diseño del sistema, para ello se presentan los diagramas de clases de análisis que dan realización a los casos de uso presentados en el capítulo anterior. Se exponen los diagramas clases y de interacción del diseño y se describen las clases Interfaces, Controladoras y Entidades, así como las clases utilizadas en las bases de datos. Se muestran los diagramas Entidad Relación de las dos bases de datos que se implementan, una para guardar las referencias de los estudios radiológicos, y otra para guardar los registros de búsquedas.

3.1 Modelo Arquitectónico.

El uso de las aplicaciones orientadas a servicios constituye para muchos hoy, un estilo arquitectónico de las tecnologías de la información que les permiten integrar el negocio mediante el enlace de servicios.

Se trata de habilitar la flexibilidad del negocio mediante la integración de los sistemas, datos, aplicaciones, procesos y personas en toda la empresa y más allá de ella, dando a las organizaciones el poder de reducir las complejidades, ampliar el valor de las inversiones en TI¹⁰ existentes y responder dinámicamente a las condiciones cambiantes del negocio.(8a)

SOA es un conjunto de servicios tanto de negocio como tecnológicos que interactúan entre ellos, proporcionando la lógica necesaria para construir aplicaciones de manera rápida y cumpliendo siempre con los principios de la Orientación a Servicios. Proporciona el entorno tecnológico para que las aplicaciones compartan procesos de gestión comunes

El proyecto de la investigación ha sido desarrollado básicamente teniendo en cuenta los fundamentos de la Arquitectura Orientada a Servicios y del patrón BROKER (Object Request Broker), utilizado para estructurar sistemas distribuidos con componentes desacoplados y cooperativos que interactúan por

¹⁰ Tecnologías de Información.

invocaciones de objetos remotos. Sin embargo, se ha prestado atención también las cuatro C arquitectónicas. Los **componentes** o elementos que forman el sistema; las **conexiones** que existen entre estos elementos ya sean dispositivos de redes, protocolos de comunicación u otros; las **restricciones** tanto de comportamiento o de características y finalmente la **configuración** que rige la unión de los elementos.

Teniendo en cuenta que el sistema está compuesto por sistemas desacoplados e independientes es importante señalar la necesidad de un medio de comunicación. El patrón BROKER plantea entre otras cosas que los componentes deben ser capaces de acceder a los servicios proveídos por otros, a través de invocaciones de servicios remotos transparentes en ubicación. Se implementaría un servicio que hostearía un objeto remoto, el cual implementaría una interfaz con la que se comunicaría el servidor DICOM.

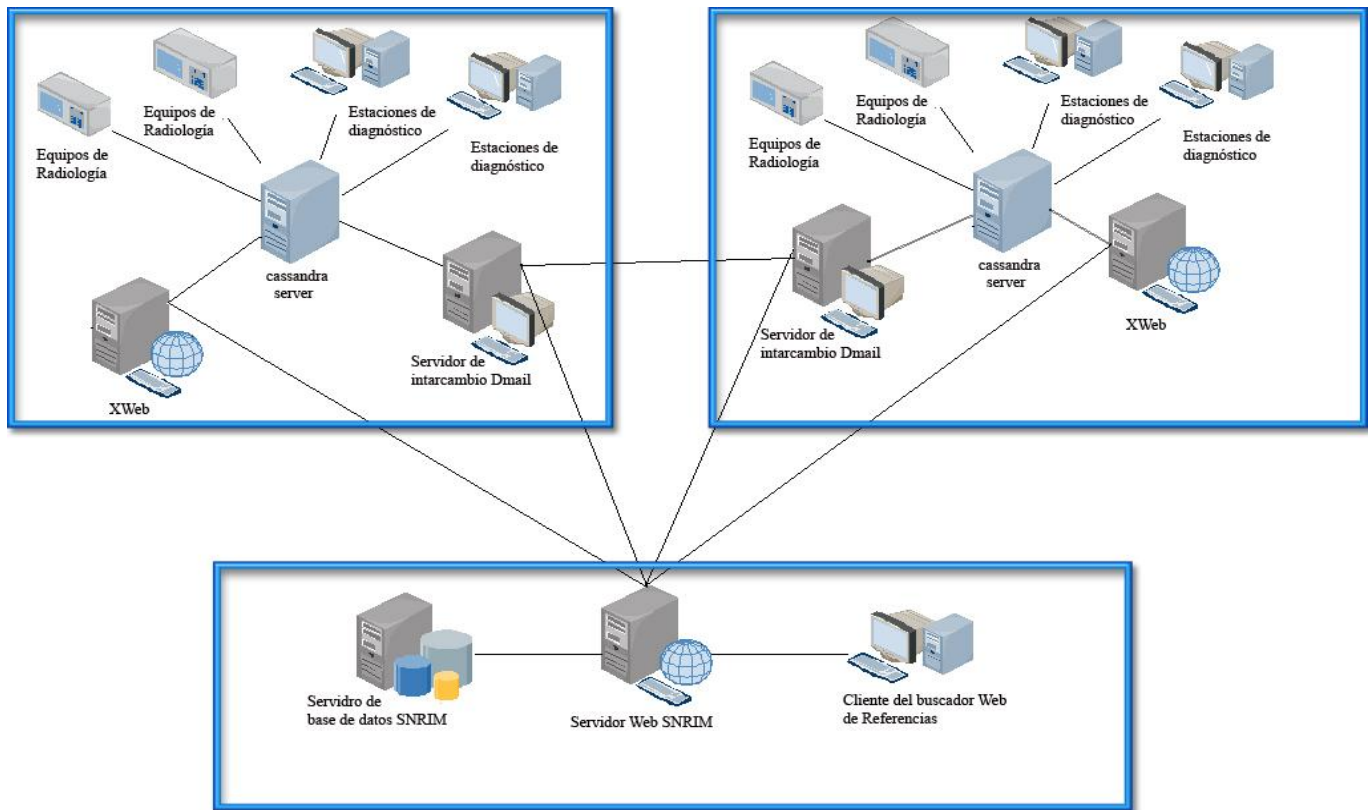


Figura 3 - Interacción de los elementos de Cassandra.

El sistema está compuesto por tres subsistemas que se desplegarían cumpliendo con SOA.

El servidor de referencias, que se instalaría como único componente de su tipo a nivel nacional. En el correrían los servicios web de gestión y de búsqueda. Además, una aplicación web que permitiría realizar búsquedas en el servidor de referencias, instalada en dicho servidor. Y el sistema orquestador, al frente de las gestiones entre el hospital y el servidor de referencias, del cual se instalaría uno en cada hospital. Toda esta distribución ha sido planeada arquitectónicamente para su cumplimiento con todos los requerimientos funcionales y no funcionales.

3.2 Análisis.

Durante el Análisis se analizan los requerimientos funcionales que se hayan definido, refinándolos y estructurándolos con el objetivo de conseguir una descripción más detallada que ayude a estructurar el sistema.

El Modelo de Análisis es la realización del Modelo de Casos de Uso y da paso al Modelo de Diseño. Su objetivo fundamental es la realización de los casos de uso expresándolos en los diagramas de clases del análisis, que en este caso aparecerán organizados en los dos paquetes definidos desde el Modelo de Diseño; el Orchestrator y el SNRIM.

3.2.1 Diagrama de Clases de Análisis por Casos de Uso.

3.2.1.1 Paquete Orchestrator.

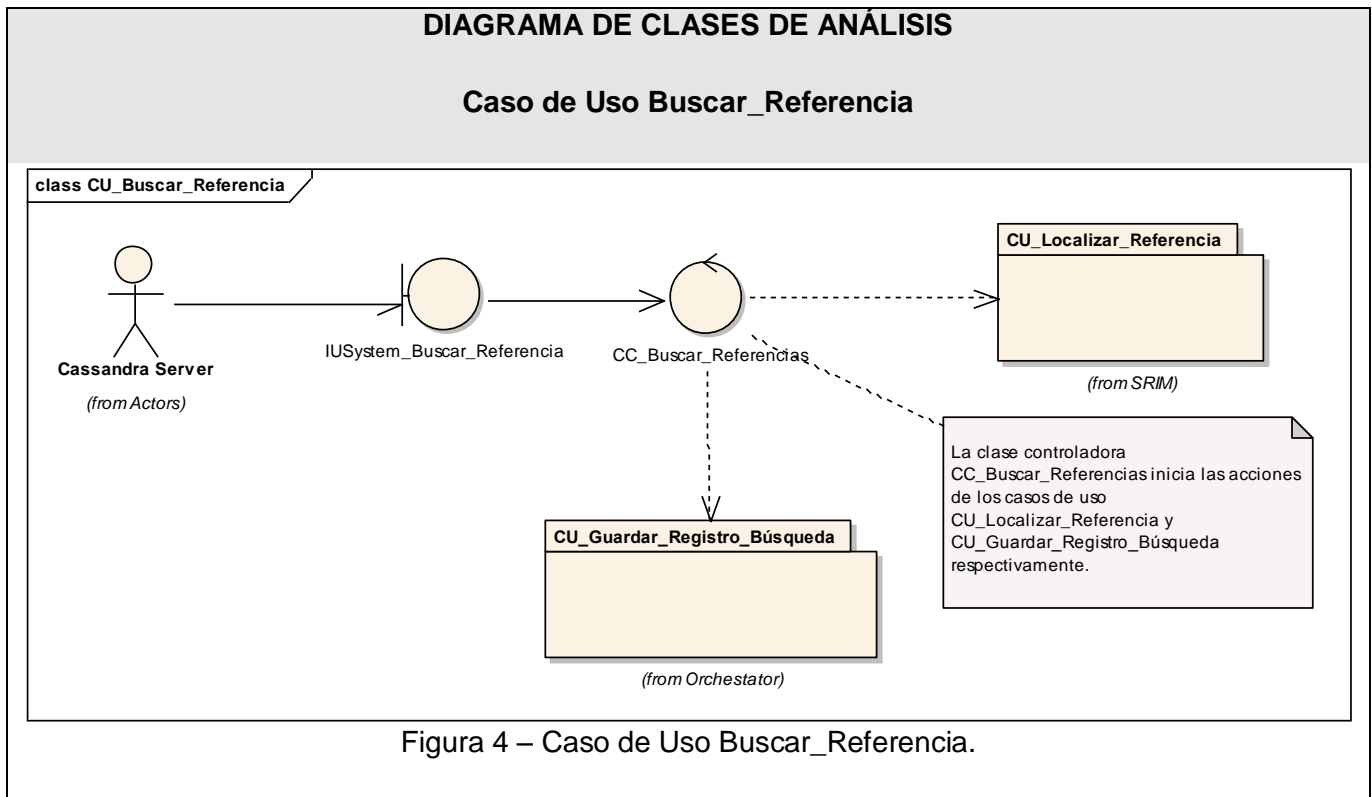


DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso Visualizar_Registro_Búsqueda

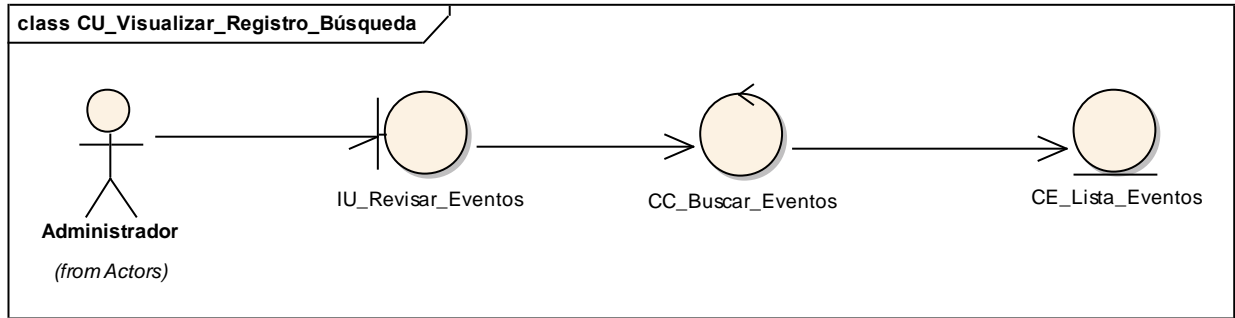


Figura 5 – Visualizar_Registro_Búsqueda.

DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso Guardar_Registro_Búsqueda

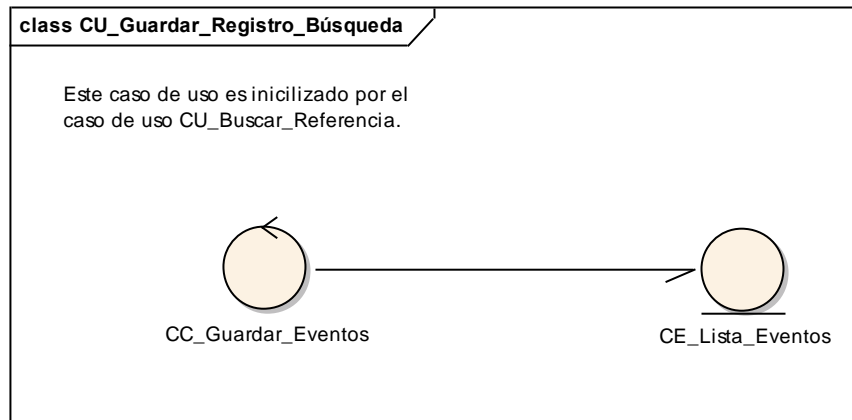


Figura 6 – Guardar_Registro_Búsqueda.

DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso Insertar_Referencia

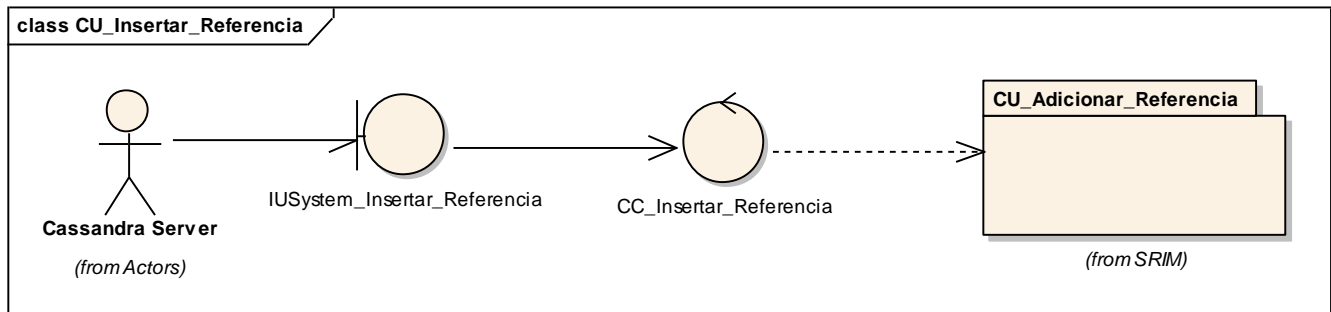


Figura 7 – Insertar_Referencia.

3.2.1.2 Paquete SNRIM.

DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso Adicionar_Referencia

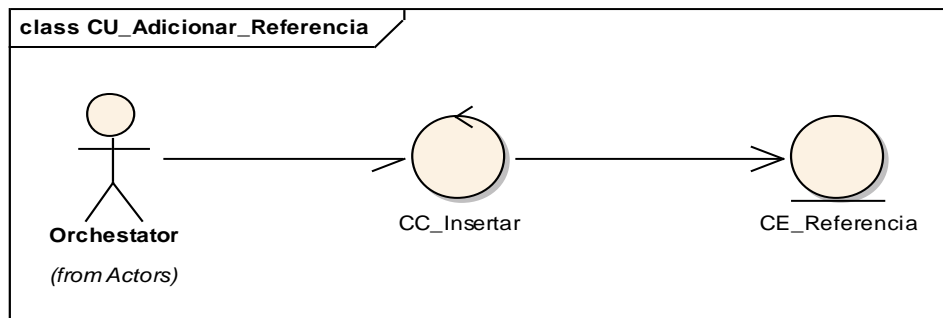


Figura 8 – Adicionar_Referencia.

DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso AdicionarReporte_Referencia

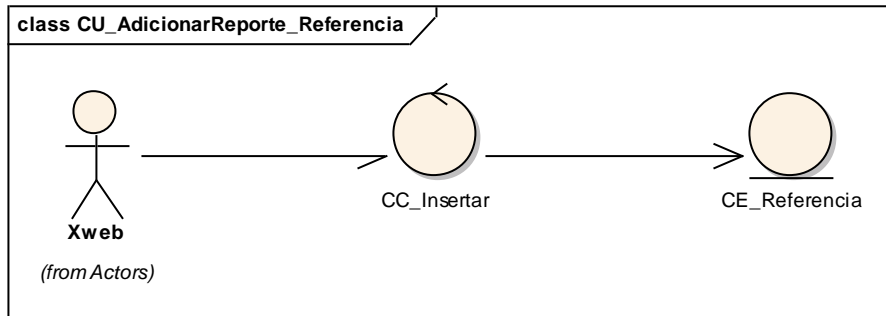


Figura 9 – AdicionarReporte_Referencia.

DIAGRAMA DE CLASES DE ANÁLISIS

Caso de Uso Localizar_Referencia

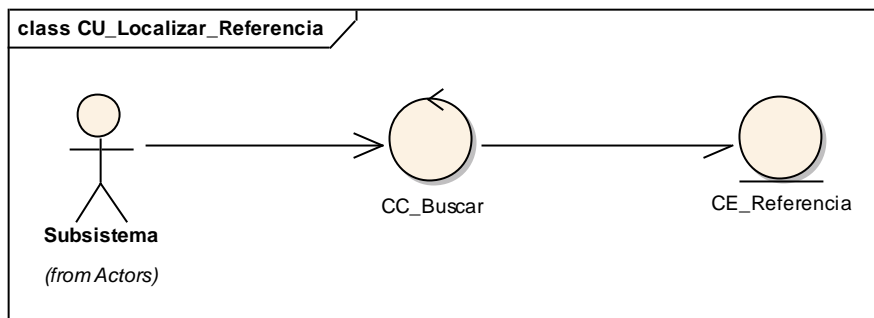


Figura 1 – Localizar_Referencia.

3.2.1.3 Paquete Sistema de Búsqueda de Referencias.

DIAGRAMA DE CLASES DE ANÁLISIS Caso de Uso BusquedaWeb_Referencia

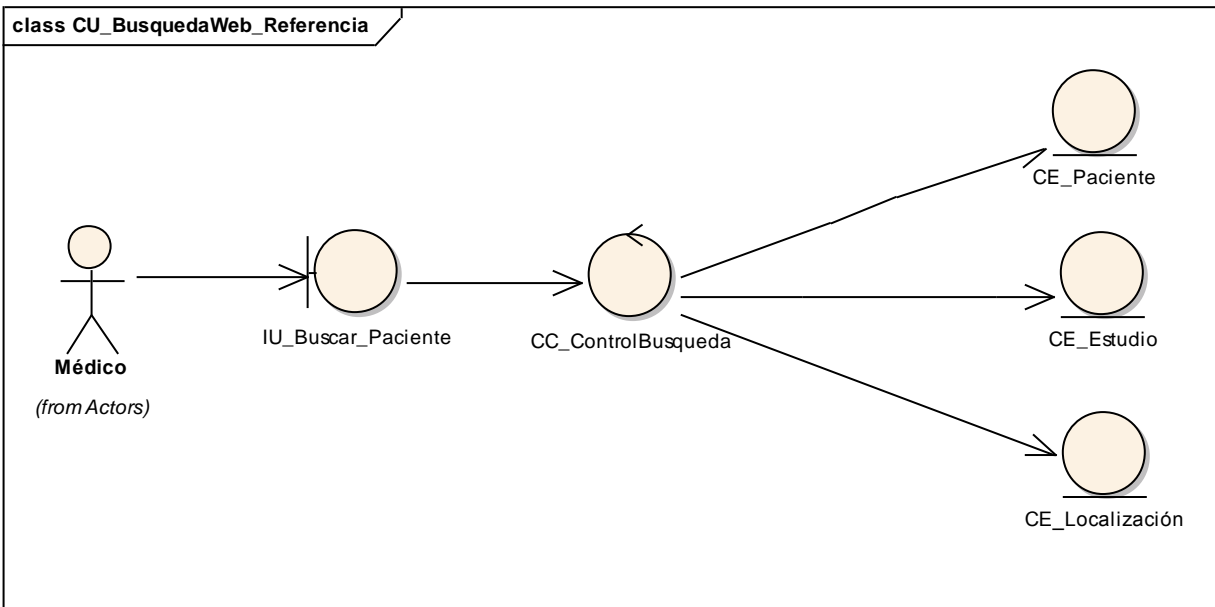


Figura 2 – Localizar_Referencia.

3.3 Diseño.

En el flujo de trabajo de Diseño lo más importante es la elaboración de los diagramas de clases de diseño, que muestran las clases finales que dan realización a los casos de uso modelados con anterioridad. Muchas de las clases definidas en el análisis se mantienen presentes o bien aparecen algunas nuevas para cumplir funciones más específicas de las definidas en el análisis. Los diagramas de clases de diseño muestran las clases, con sus atributos y métodos, y la forma en que se relacionan entre si. [Ver Anexo 2.](#)

Además se realizan los diagramas de interacción que muestran como se relacionan los objetos, con el objetivo de dar cumplimiento a los requerimientos. En este caso, se documentan los diagramas de secuencia que detallan las acciones ordenadas en el tiempo.

3.3.1 Diagramas de Secuencia por Casos de Uso.

3.3.1.1 Paquete Orchestator.

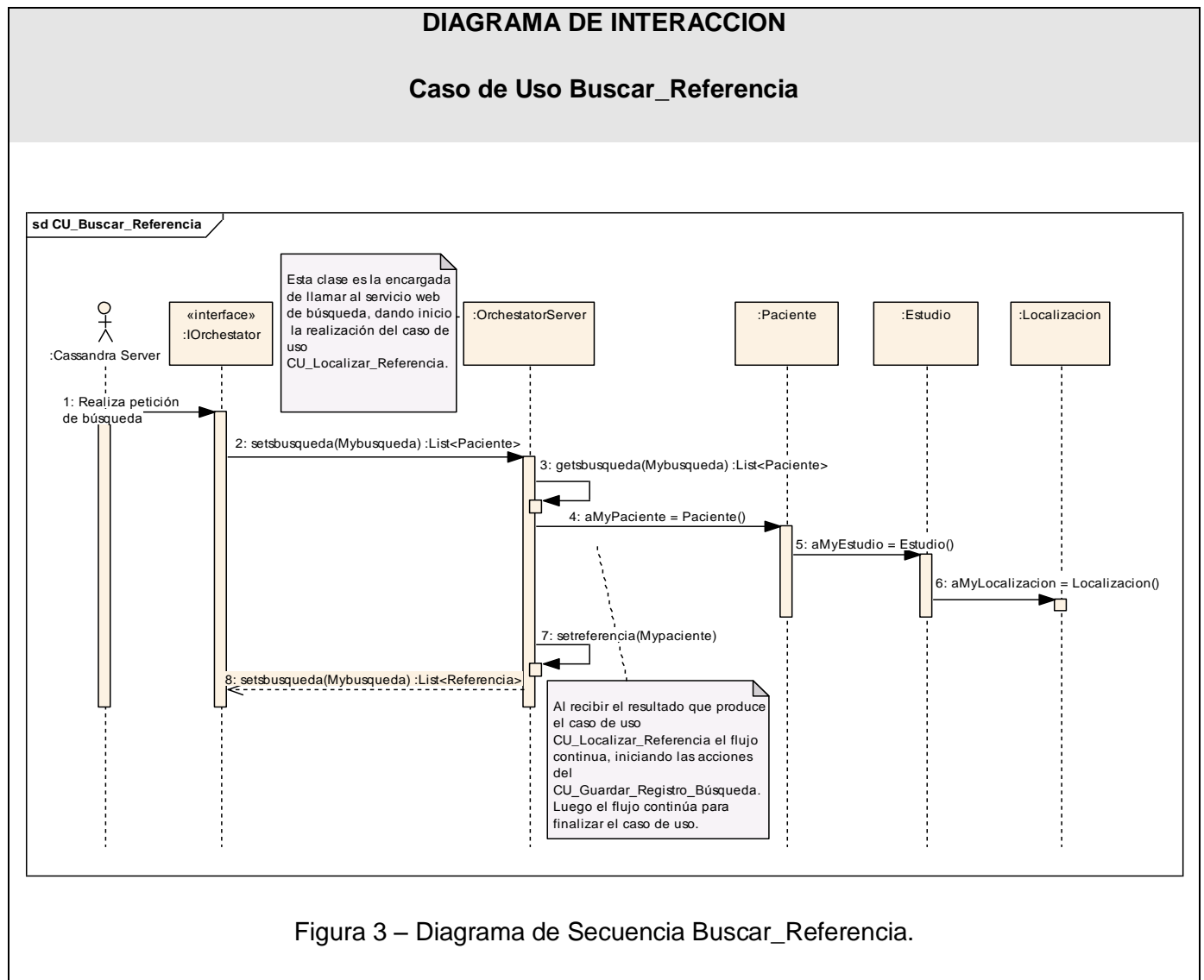


DIAGRAMA DE INTERACCION

Caso de Uso Visualizar_Registro_Búsqueda

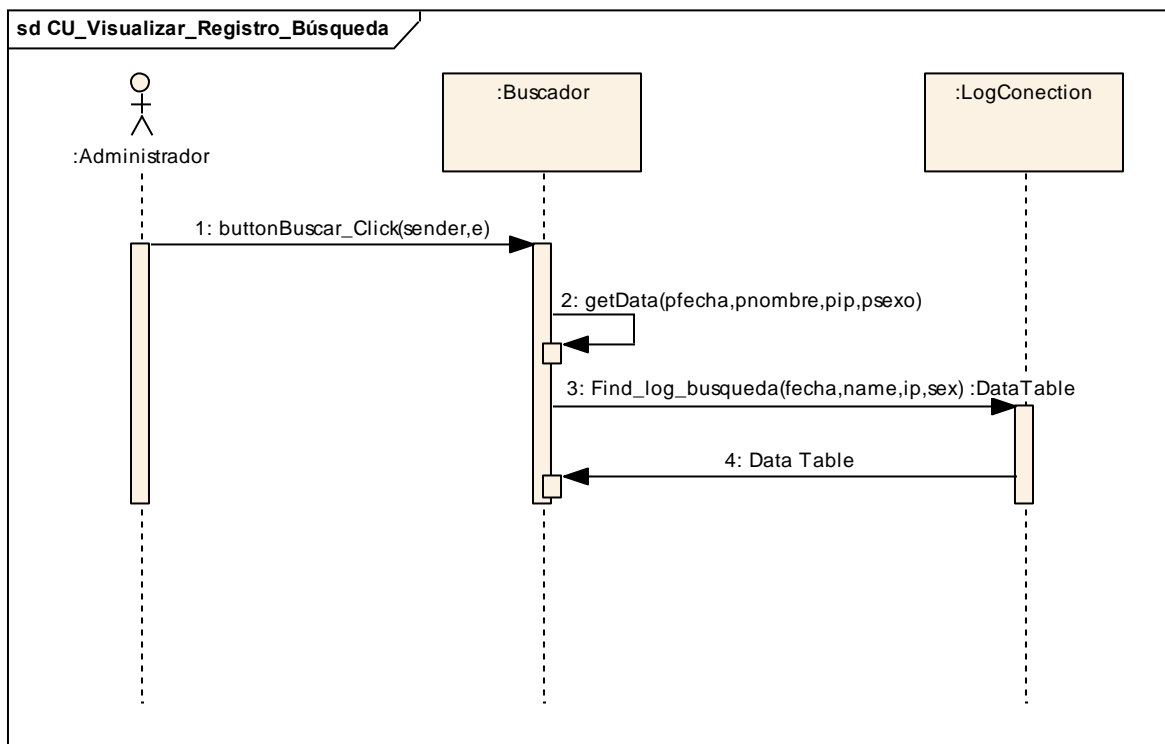


Figura 4 - Diagrama de Secuencia Visualizar_Registro_Búsqueda.

DIAGRAMA DE INTERACCION

Caso de Uso Guardar_Registro_Búsqueda

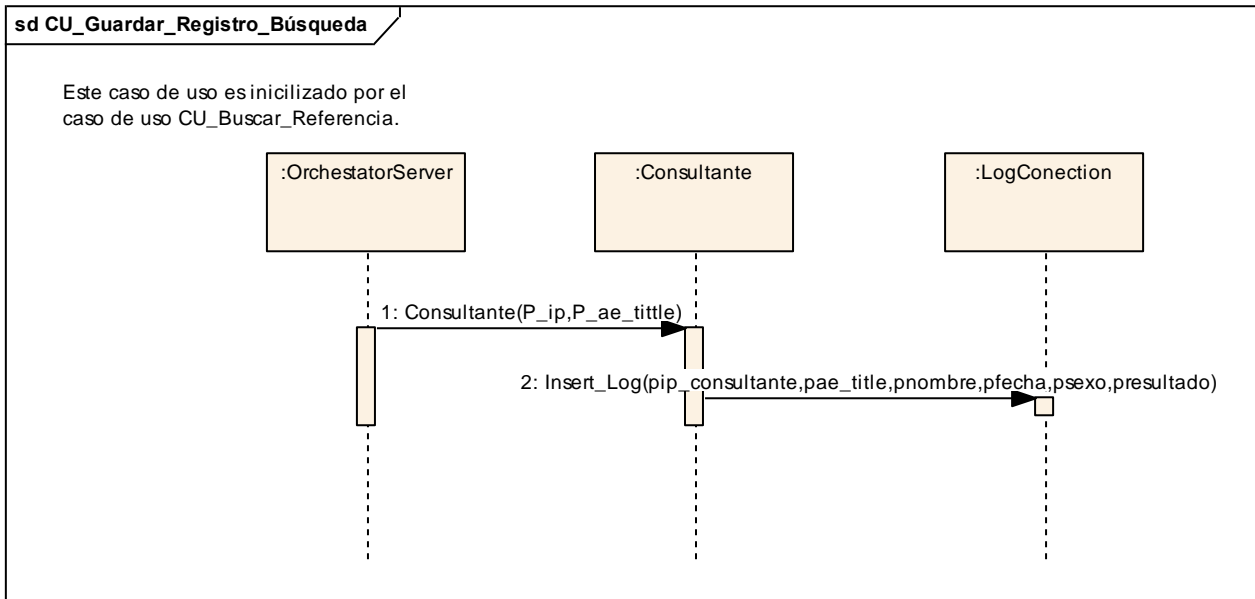


Figura 5 - Diagrama de Secuencia Guardar_Registro_Búsqueda.

DIAGRAMA DE INTERACCION

Caso de Uso Insertar_Referencia

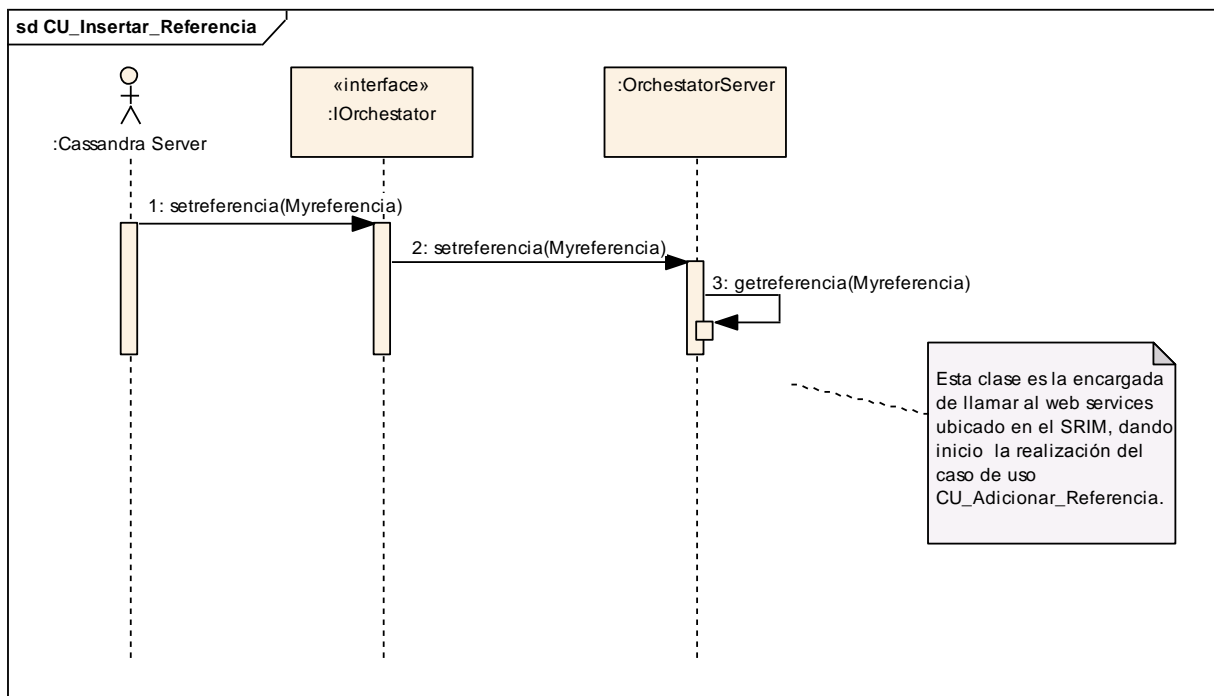


Figura 6 - Diagrama de Secuencia Insertar_Referencia.

3.3.1.2 Paquete SNRIM

DIAGRAMA DE INTERACCION

Caso de Uso Adicionar_Referencia

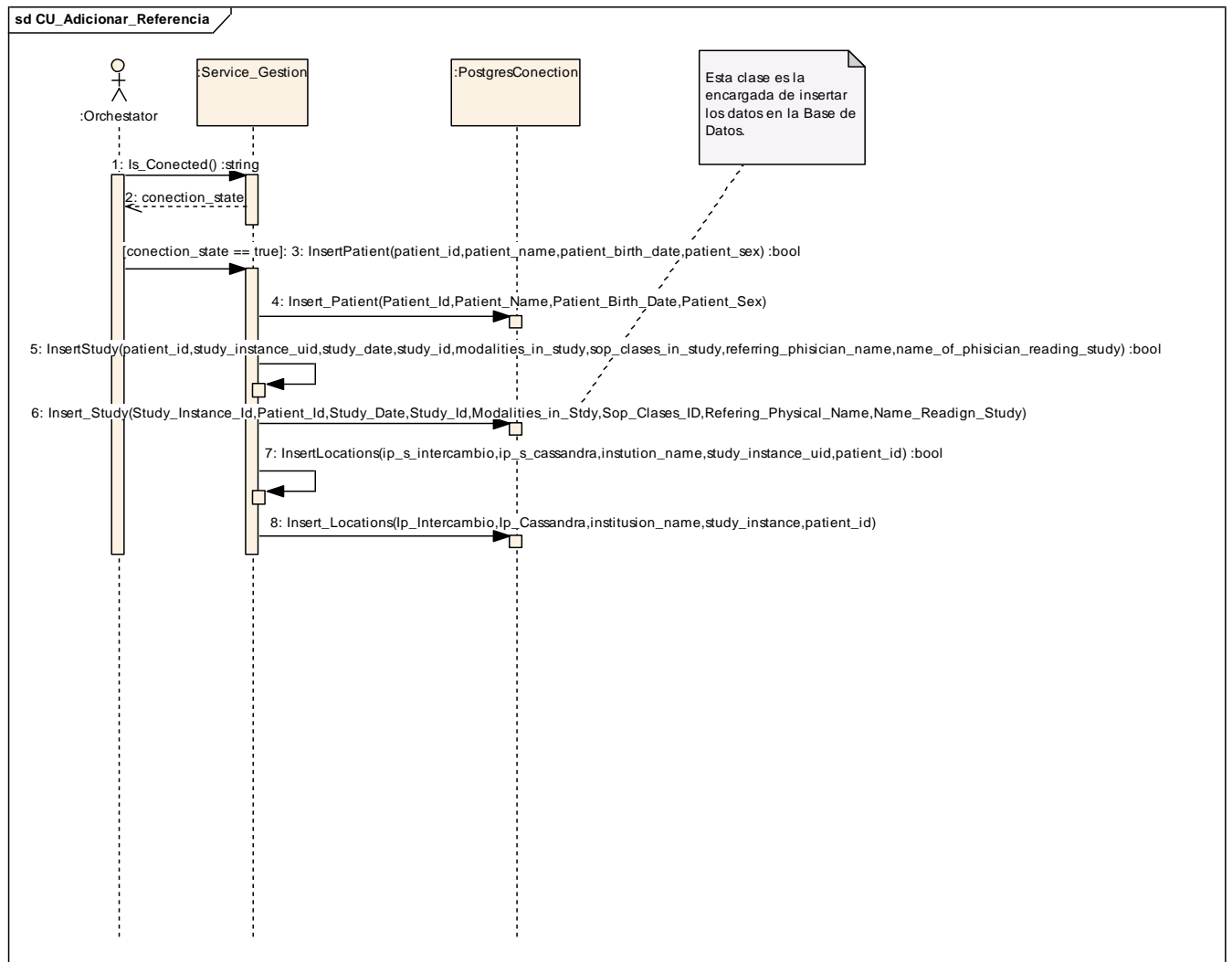


Figura 7 - Diagrama de Secuencia Adicionar_Referencia.

DIAGRAMA DE INTERACCION

Caso de Uso AdicionarReporte_Referencia

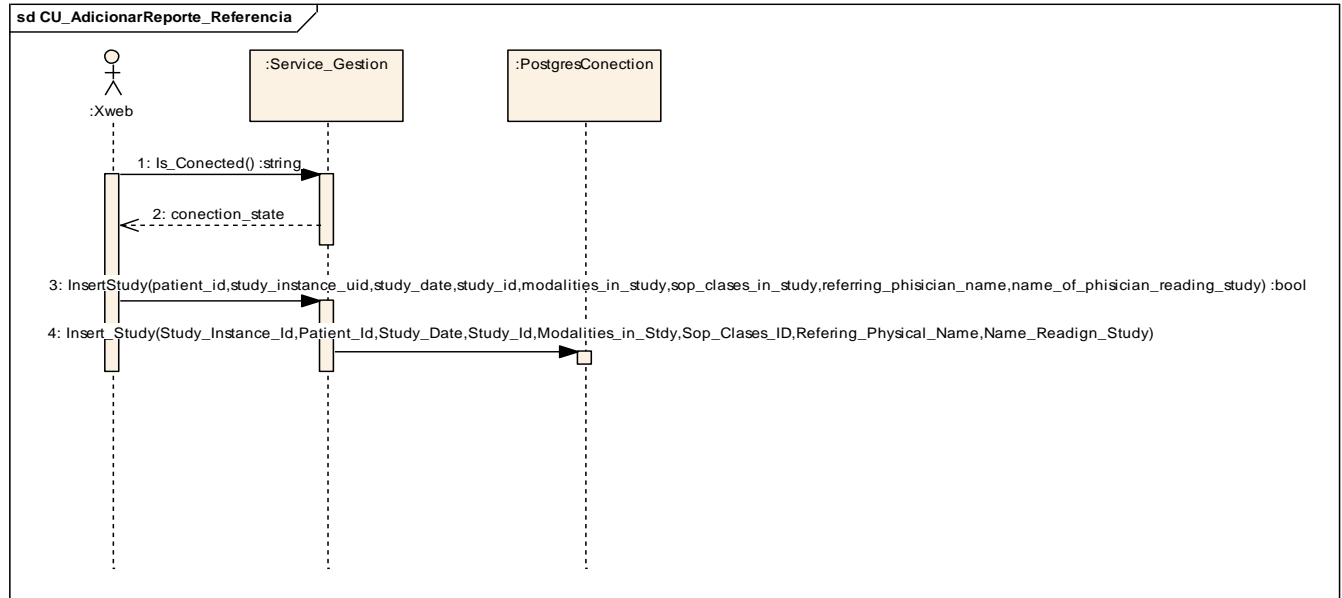


Figura 8 - Diagrama de Secuencia AdicionarReporte_Referencia.

DIAGRAMA DE INTERACCION

Caso de Uso Localizar_Referencia

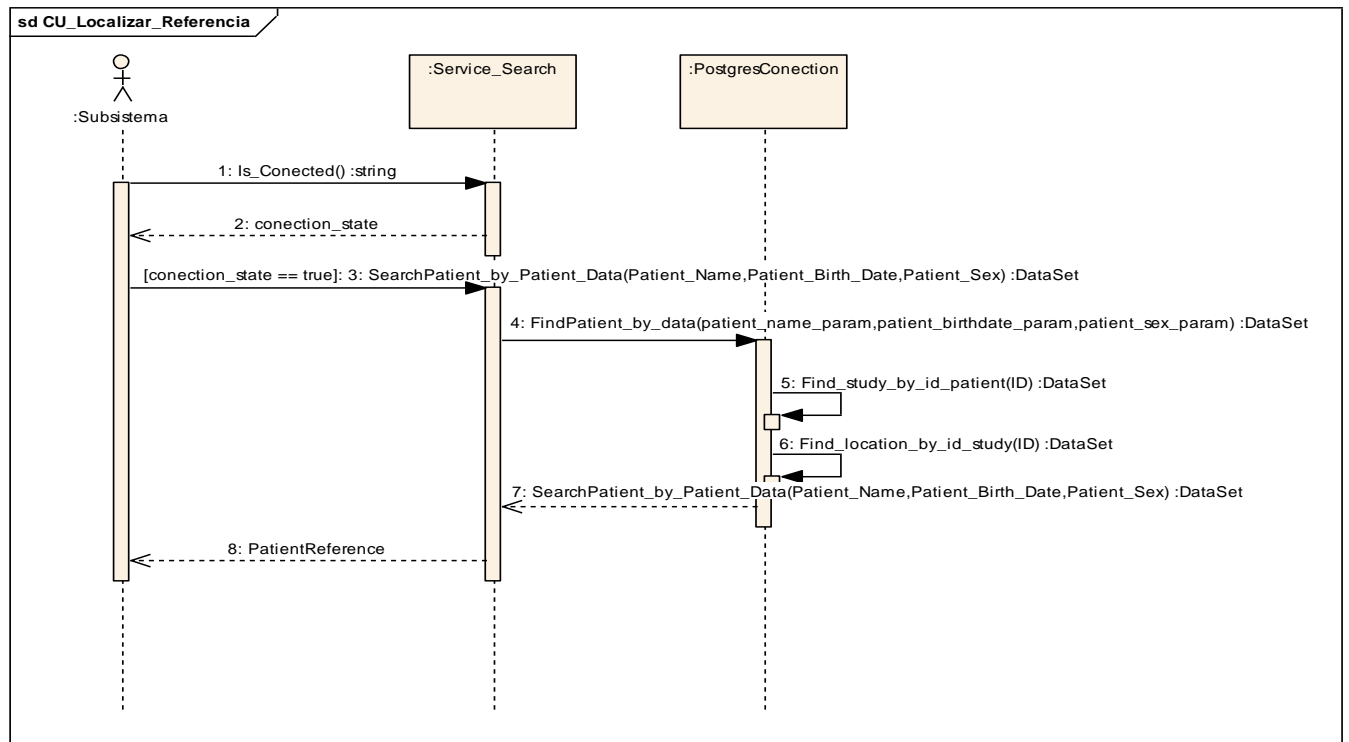
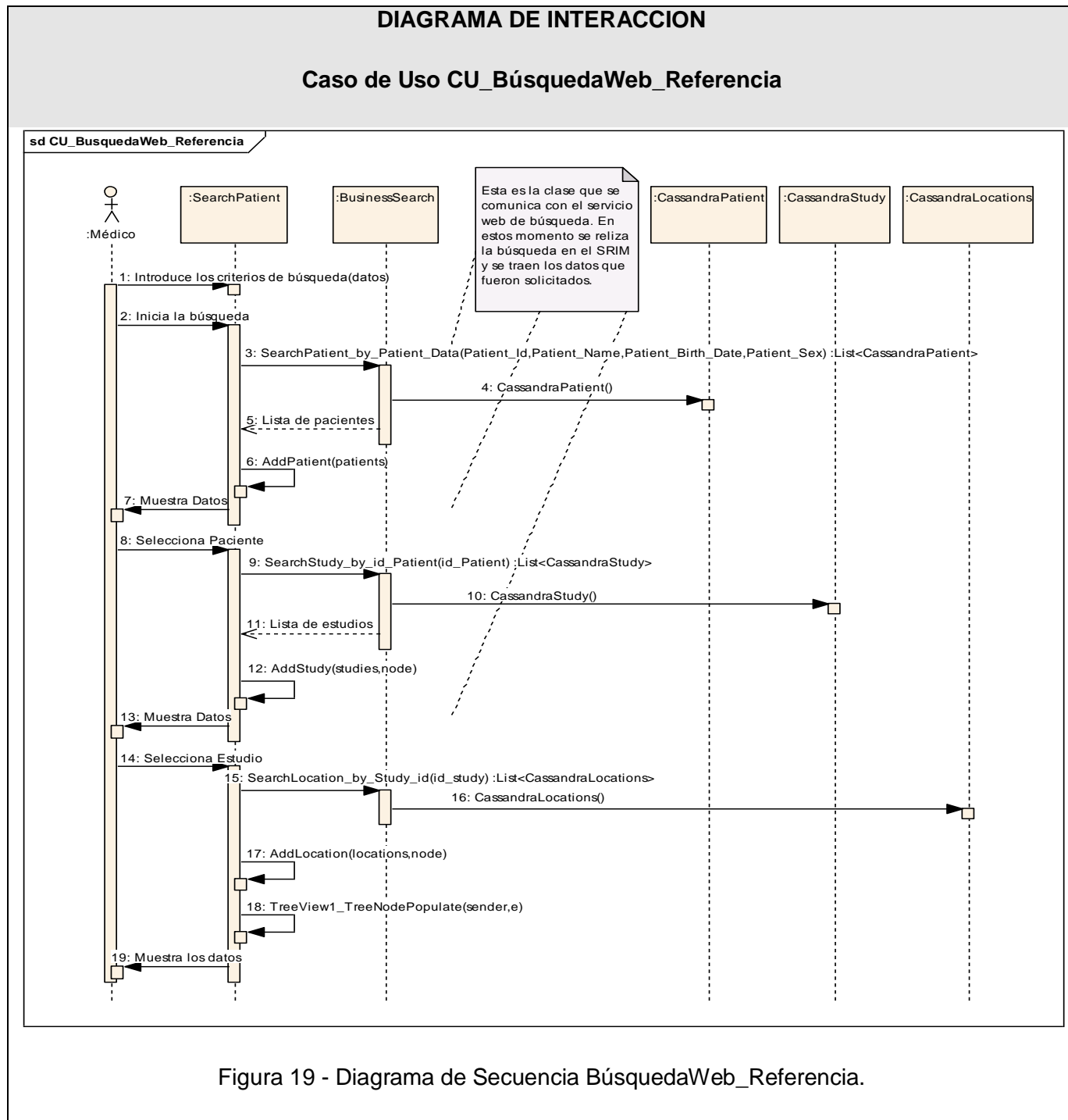


Figura 9 - Diagrama de Secuencia Localizar_Referencia.

3.3.1.3 Paquete Sistema de Búsqueda de Referencias.



3.3.2 Descripción de las Clases de Diseño.

Las clases se describirán según las agrupaciones que se definieron en el Modelo de Diseño.

3.3.2.1 Paquete Orchestator

Orchestator: IOrchestatorServer

Nota: Esta interfaz no está creada con el propósito de polimorfismo como la mayoría de las interfaces, sino de separar la implementación de la misma con el objetivo de cuando cambie el código de la implementación no tener que proporcionar la **dll** a los clientes que la utilicen.

La descripción de los métodos de esta interfaz lo podrá ver en la clase OrchestatorServer

Nombre: IOrchestator	
Tipo de clase: Interfaz de sistema.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	setreferencia(Paciente Mypaciente)
Descripción:	
Nombre:	setsbusqueda(Sbusqueda Mybusqueda)
Descripción:	

Orchestator: OrchestatorServer

Nombre: OrchestatorServer	
Tipo de clase: Control	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	getreferencia(Referencia Myreferencia)
Descripción:	El propósito de este método es pasar la referencia a la capa de acceso a datos.
Nombre:	getsbusqueda(Sbusqueda Mybusqueda)
Descripción:	Este método se encarga de conformar y devolver una lista de Pacientes encontrados.
Nombre:	setreferencia(Paciente Mypaciente)
Descripción:	Se encarga de recibir y validar un Paciente y pasárselo al método getreferencia.
Nombre:	setsbusqueda(Sbusqueda Mybusqueda)
Descripción:	Se encarga de recibir y validar los parámetros de la búsqueda y pasárselos a getsbusqueda.

Orchestator: OrchestatorService

Nombre: OrchestatorService	
Tipo de clase: Control	
Atributo	Tipo
chan	TcpChannel
Para cada responsabilidad:	
Nombre:	OnStart(string[] args)
Descripción:	Este método inicia el servicio Windows que se encarga de hostear y configurar el objeto remoto.
Nombre:	OnStop()
Descripción:	El propósito de este método es detener el servicio Windows y deshostear el objeto remoto.
Nombre:	OrchestatorService()
Descripción:	Este método es el constructor de la clase.

ClasesUtiles

Nombre: Paciente	
Tipo de clase: Entidad	
Atributo	Tipo

pacienteid	string
nombre	string
fecha_nacimiento	DateTime
sexo	char
MyEstudio	List<Estudio>
Para cada responsabilidad:	
Nombre:	Paciente()
Descripción:	Constructor. Inicializa una nueva instancia de Paciente con los valores por defecto.
Nombre:	Pacienteid
Descripción:	Establece o devuelve los valores de pacienteid.
Nombre:	Nombre
Descripción:	Establece o devuelve los valores de nombre.
Nombre:	Fecha_nacimiento
Descripción:	Establece o devuelve los valores de fecha_nacimiento.
Nombre:	Sexo
Descripción:	Establece o devuelve los valores de sexo.

Nombre: Estudio	
Tipo de clase : Entidad	
Atributo	Tipo
study_instance_uid	string
study_date	DateTime
study_id	string
modalities_in_study	string
sop_clases_in_study	string
referring_physician_name	string
name_of_physician_reading_study	string
reporte	bool
MyLocalizacion	List<Localizacion>
Para cada responsabilidad:	
Nombre:	Estudio()
Descripción:	Es el Constructor que inicializa una nueva instancia de Estudio con los valores por defecto.
Nombre:	Study_instance_uid
Descripción:	Establece o devuelve los valores de study_instance_uid.
Nombre:	Study_date

Descripción:	Establece o devuelve los valores de study_date.
Nombre:	Study_id
Descripción:	Establece o devuelve los valores de study_id.
Nombre:	Modalities_in_study
Descripción:	Establece o devuelve los valores de modalities_in_study
Nombre:	Sop_clases_in_study
Descripción:	Establece o devuelve los valores de sop_clases_in_study.
Nombre:	Referring_physician_name
Descripción:	Establece o devuelve los valores de referring_physician_name.
Nombre:	Name_of_physician_reading_study
Descripción:	Establece o devuelve los valores de name_of_physician_reading_study.
Nombre:	Reporte
Descripción:	Establece o devuelve los valores de reporte.

Nombre: Localizacion	
Tipo de clase : Entidad	
Atributo	Tipo
ip_s_intercambio	string
ip_s_cassandra	string
institution_name	string
Para cada responsabilidad:	
Nombre:	Localizacion()
Descripción:	Constructor que inicializa una nueva instancia de Localizacion con los valores por defecto.
Nombre:	Ip_s_intercambio
Descripción:	Establece o devuelve los valores de ip_s_intercambio.
Nombre:	Ip_s_cassandra
Descripción:	Establece o devuelve los valores de ip_s_cassandra.
Nombre:	Institution_name
Descripción:	Establece o devuelve los valores de institution_name.

Nombre: Sbusqueda	
Tipo de clase: Entidad	
Atributo	Tipo
Mypaciente	Paciente
Myestudio	Estudio
Mylocalizacion	Localizacion
tipoB	int
Para cada responsabilidad:	
Nombre:	Sbusqueda()
Descripción:	Constructor que inicializa una nueva instancia de Sbusqueda con los valores por defecto.
Nombre:	Sbusqueda(Paciente aMypaciente,Estudio aMyestudio,Localizacion aMylocalizacion,int aTipoB)
Descripción:	Constructor. Inicializa una nueva instancia de Sbusqueda con los valores proporcionados.
Nombre:	TipoB
Descripción:	Establece o devuelve los valores de tipoB.

Nombre: Consultante	
Tipo de clase: Entidad	
Atributo	Tipo
ip	String
ae_tittle	String
Para cada responsabilidad:	
Nombre:	Consultante()
Descripción:	Constructor. Inicializa una nueva instancia de Consultante con los valores por defecto.
Nombre:	Consultante(string P_ip, string P_ae_tittle)
Descripción:	Constructor. Inicializa una nueva instancia de Consultante con los valores proporcionados.
Nombre:	Ip
Descripción:	Establece o devuelve los valores de ip
Nombre:	Ea_tittle
Descripción:	Establece o devuelve los valores de ae_tittle

Común. Data_Access

Nombre: PostgresConection	
Tipo de clase : Control	
Atributo	Tipo
conn	NpgsqlConnection
pgCommand	NpgsqlCommand
Para cada responsabilidad:	
Nombre:	PostgresConection(String IPData_Server, int Port, String User, String Password, bool SSL, String Data_Base)
Descripción:	Constructor que inicializa una nueva instancia de NpgsqlConnection con los valores proporcionados y se lo asigna a conn creando así una conexión a una base de datos PostgreSQL.
Nombre:	Isconnected()
Descripción:	Retorna true/false en dependencia del estado de la conexión conn.
Nombre:	Insert_Patient(String Patient_Id, String Patient_Name, DateTime Patient_Birth_Date, Char? Patient_Sex)
Descripción:	Método para insertar pacientes en la base de Datos
Nombre:	public void Insert_Study(String Study_Instance_Id, String Patient_Id, DateTime Study_Date, String Study_Id, String Modalities_in_Stdy, String Sop_Classes_ID, String Refering_Physical_Name, String Name_Readign_Study)

Descripción:	Método para Insertar Studios en la Base de datos.
Nombre:	Insert_Locations(String Ip_Intercambio, String Ip_Cassandra, String institusion_name, String study_instance, String patient_id)
Descripción:	Método para Insertar Direcciones en la Base de datos.
Nombre:	Insert_Log(string pip_consultante, string pae_title, string pnombre, DateTime pfecha, char psexo, bool presultado)
Descripción:	Método para Insertar registros de búsquedas
Nombre:	FindPatient_by_data(string patient_name_param, DateTime patient_birthdate_param, char? patient_sex_param)
Descripción:	Método para buscar Pacientes por datos definidos.
Nombre:	Find_study_by_id_patient(string ID)
Descripción:	Método para buscar Estudios por el id del Paciente.
Nombre:	Find_location_by_id_study(string ID)
Descripción:	Método para buscar localizaciones por el id del Estudio.
Nombre:	Find_log_búsqueda(DateTime fecha, string name, string ip, char sex)
Descripción:	Método para buscar Registros de búsquedas por datos definidos.
Nombre:	Delete_Patient(String Patient_Id)
Descripción:	Método para eliminar un Paciente y todos sus Estudios de la Base de datos.

Nombre: LogConection	
Tipo de clase : Control	
Atributo	Tipo
conn	NpgsqlConnection
pgCommand	NpgsqlCommand
Para cada responsabilidad:	
Nombre:	LogConection(String IPData_Server, int Port, String User, String Password, bool SSL, String Data_Base)
Descripción:	Constructor que inicializa una nueva instancia de NpgsqlConnection con los valores proporcionados y se lo asigna a conn creando así una conexión a una base de datos PostgreSQL.
Nombre:	Isconnected()
Descripción:	Retorna true/false en dependencia del estado de la conexión conn.
Nombre:	Insert_Log(string pip_consultante, string pae_title, string pnombre, DateTime pfecha, char psexo, bool presultado)
Descripción:	Método para insertar registros de búsqueda en la base de Datos.
Nombre:	Find_log_busqueda(DateTime fecha, string name, string ip, char sex)
Descripción:	Método para buscar Registros de busquedas por datos definidos.
Nombre:	Find_log_encontrados()

Descripción:	Devuelve la cantidad total de registros de búsquedas que tuvieron éxito.
Nombre:	Find_log_noencontrados()
Descripción:	Devuelve la cantidad total de registros de búsquedas que no tuvieron éxito.
Nombre:	Find_log_cantidadtotal()
Descripción:	Devuelve la cantidad total de registros de búsquedas.

Nombre: Buscador	
Tipo de clase : Interfaz	
Atributo	Tipo
Conexion	Data_Access.PostgresConexion
Para cada responsabilidad:	
Nombre:	Buscador()
Descripción:	Constructor de la clase.
Nombre:	button1_Click(object sender, EventArgs e)
Descripción:	Valida los datos de entrada y llama al método getData
Nombre:	getData(DateTime pfecha, string pnombre, string pip, char psexo)
Descripción:	Realiza una búsqueda en la base de datos a través de la clase

	PostgresConection y muestra los datos en un GridView.
Nombre:	button2_Click(object sender, EventArgs e)
Descripción:	Muestra todos los Registros almacenados en un GridView.
Nombre:	Form1_Load(object sender, EventArgs e)
Descripción:	

3.3.2.2 Paquete SNRIM.

Servicio Web de Gestión

Nombre: Service	
Tipo de clase : Control	
Atributo	Tipo
Conexion	Data_Access.PostgresConection
Para cada responsabilidad:	
Nombre:	Service()
Descripción:	Constructor que inicializa una nueva instancia de Data_Access.PostgresConection con los valores proporcionados y se lo asigna a Conexion creando así una conexión a una base de datos PostgreSQL a través de la clase PostgresConection.
Nombre:	Isconnected()
Descripción:	Retorna true/false en dependencia del estado de la conexión Conexion.

Nombre:	InsertPatient(String patient_id, String patient_name, DateTime patient_birth_date, Char patient_sex)
Descripción:	Método Web para insertar pacientes en la base de Datos a través de la clase PostgresConexion.
Nombre:	InsertLocations(String ip_s_intercambio, String ip_s_cassandra, String instution_name, String study_instance_uid, String patient_id)
Descripción:	Método Web para insertar Localizaciones en la base de Datos a través de la clase PostgresConexion.
Nombre:	InsertStudy(String patient_id, String study_instance_uid, DateTime study_date, String study_id, String modalities_in_study, String sop_clases_in_study, String referring_phisician_name, String name_of_phisician_reading_study)
Descripción:	Método Web para insertar Estudios en la base de Datos a través de la clase PostgresConexion.
Nombre:	UpdateStudy(String patient_id, String study_instance_uid, String ip_s_intercambio, String ip_s_cassandra, String instution_name, String Nip_s_intercambio, String Nip_s_cassandra, String Ninstution_name)
Descripción:	Método Web para actualizar localizaciones de Estudios en la base de Datos a través de la clase PostgresConexion.
Nombre:	DeletePatient(String patient_id)
Descripción:	Método Web para eliminar pacientes en la base de Datos a través de la clase PostgresConexion.

Servicio Web de Búsqueda

Nombre: Service	
Tipo de clase : Control	
Atributo	Tipo
Conexion	Data_Access.PostgresConexion
Para cada responsabilidad:	
Nombre:	Service()
Descripción:	Constructor que inicializa una nueva instancia de Data_Access.PostgresConexion con los valores proporcionados y se lo asigna a Conexion creando así una conexión a una base de datos PostgreSQL a través de la clase PostgresConexion.
Nombre:	Isconnected()
Descripción:	Retorna true/false en dependencia del estado de la conexión Conexion.
Nombre:	SearchPatient_by_Patient_Data(string Patient_Name, string Patient_Birth_Date, char Patient_Sex)
Descripción:	Método Web para buscar pacientes en la base de Datos a través de la clase PostgresConexion.
Nombre:	SearchStudy_by_id_Patient(string id_Patient)
Descripción:	Método Web para buscar estudios en la base de Datos a través de la clase PostgresConexion.

Nombre:	SearchLocation_by_study_id(string id_study)
Descripción:	Método Web para buscar Localizaciones en la base de Datos a través de la clase PostgresConection.

3.3.2.3 Paquete Sistema de Búsqueda de Referencias.

Nombre: BusinessSearch	
Tipo de clase : Control	
Atributo	Tipo
wssearch	Search.Service
Para cada responsabilidad:	
Nombre:	BusinessSearch()
Descripción:	Constructor de la clase.
Nombre:	SearchLocation_by_Study_id(string id_study)
Descripción:	Método que se encarga de
Nombre:	SearchPatient_by_Patient_Data(string Patient_Id, string Patient_Name, string Patient_Birth_Date, char Patient_Sex)
Descripción:	Método que devuelve una lista de paciente según los parámetros que se le pasen al método
Nombre:	SearchStudy_by_id_Patient(string id_Patient)

Descripción:	
Nombre:	SearchStudy_by_modality(string modality)
Descripción:	

Nombre: CassandraLocations	
Tipo de clase : Entidad	
Atributo	Tipo
institution_name	string
ip_s_intercambio	string
ip_s_cassandra	string
ip_cserver	string
Para cada responsabilidad:	
Nombre:	CassandraLocations()
Descripción:	Constructor.
Nombre:	CassandraLocations(string ip_s_intercambio,string ip_s_cassandra,string instution_name)
Descripción:	Constructor.
Nombre:	Ip_s_intercambio()
Descripción:	Ip del servidor de intercambio DMail.

Nombre:	Ip_s_cassandra()
Descripción:	Ip del servidor donde se almacenan los estudios.
Nombre:	Instution_name()
Descripción:	Nombre de la Institución donde se almacenan los estudios.

Nombre: CassandraPatient	
Tipo de clase : Entidad	
Atributo	Tipo
patient_id	string
sex	string
birthDate	string
name	string
studies	List<CassandraStudy>
Para cada responsabilidad:	
Nombre:	CassandraPatient()
Descripción:	Constructor.
Nombre:	CassandraPatient(string patient_id, string name, string birthDate, string sex)
Descripción:	Constructor.

Nombre:	CassandraPatient(string patient_id, string name, string birthDate, string sex,List<CassandraStudy> studies)
Descripción:	Constructor.
Nombre:	Patient_id()
Descripción:	Id del paciente.
Nombre:	Sex()
Descripción:	Sexo del paciente.
Nombre:	BirthDate()
Descripción:	Fecha de nacimiento de paciente.
Nombre:	Name()
Descripción:	Nombre del paciente.
Nombre:	Studies()
Descripción:	Lista de estudios correspondientes al paciente.

Nombre: CassandraStudy	
Tipo de clase : Entidad	
Atributo	Tipo
study_instance_uid	string
date	string

study_id	string
modalities_in_study	string
sop_clases_in_study	string
referring_physician_name	string
name_physician_reading	string
Para cada responsabilidad:	
Nombre:	CassandraStudy()
Descripción:	Constructor.
Nombre:	CassandraStudy(string study_instance_uid, string date, string study_id, string modalities_in_study, string sop_clases_in_study, string referring_physician_name, string name_physician_reading)
Descripción:	Constructor.
Nombre:	CassandraStudy(string id, string date)
Descripción:	Constructor.
Nombre:	Study_instance_uid()
Descripción:	Identificador universal de estudio.
Nombre:	Date()
Descripción:	Fecha del estudio.
Nombre:	Study_id()
Descripción:	Id del estudio.

Nombre:	Modalities_in_study()
Descripción:	Modalidad del estudio.
Nombre:	Sop_clases_in_study()
Descripción:	Identificador par servicio objeto en el estudio.
Nombre:	Referring_physician_name()
Descripción:	Nombre del médico referidor.
Nombre:	Name_physician_reading()
Descripción:	Nombre del médico que realizó el estudio.

Nombre: SearchPatient	
Tipo de clase : Interfaz	
Atributo	Tipo
data	Business_Search.BusinessSearch
Para cada responsabilidad:	
Nombre:	Button1_Click(object sender, EventArgs e)
Descripción:	Busca todos los pacientes, con todos sus datos, de un determinado sexo y en una fecha determinada, y los lista.
Nombre:	AddPatient(List<CassandraPatient> patients)
Descripción:	Adiciona la lista de paciente encontrados a los nodos del árbol en que se muestran finalmente.

Nombre:	AddStudy(List<CassandraStudy> studies, TreeNode node)
Descripción:	Adiciona la lista de estudios encontrados al nodo del paciente correspondiente.
Nombre:	AddLocation(List<CassandraLocations> locations, TreeNode node)
Descripción:	Adiciona la lista de localizaciones encontradas al nodo del estudio correspondiente.
Nombre:	TreeView1_TreeNodePopulate(object sender, TreeNodeEventArgs e)
Descripción:	Conforma el árbol que muestra jerárquicamente los pacientes, sus estudios y las localizaciones de estos últimos.

3.4 Diseño de la BD.

3.4.1 Modelo Entidad Relación.

Modelo Entidad-Relación es un concepto de modelado para bases de datos, propuesto por Peter Chen, mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como entidades (esto es similar al modelo de Programación Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante relaciones. (15)

3.4.1.1 Diagrama Entidad Relación de la Base de Datos de Registro de Búsquedas.

El siguiente diagrama está compuesto sólo por una tabla que almacena los datos referentes a los registros de búsquedas.

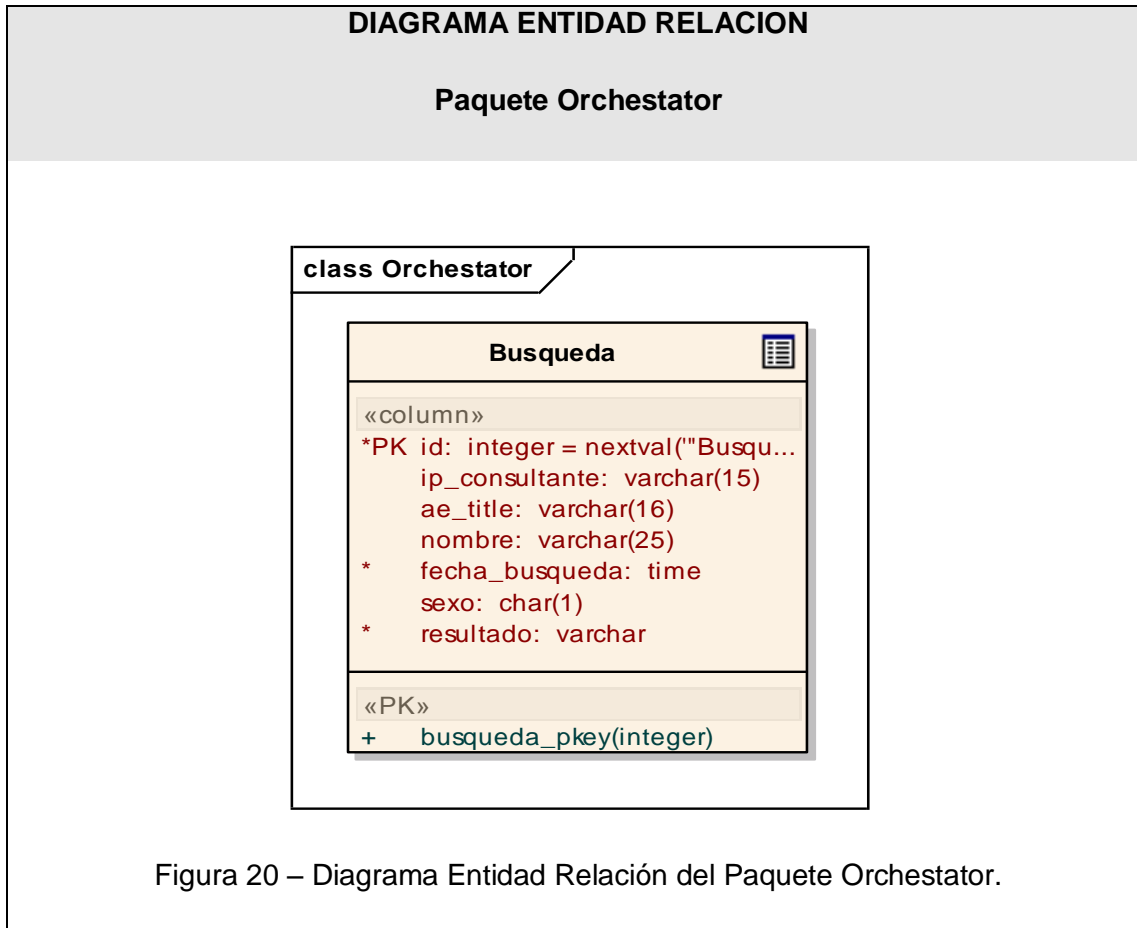
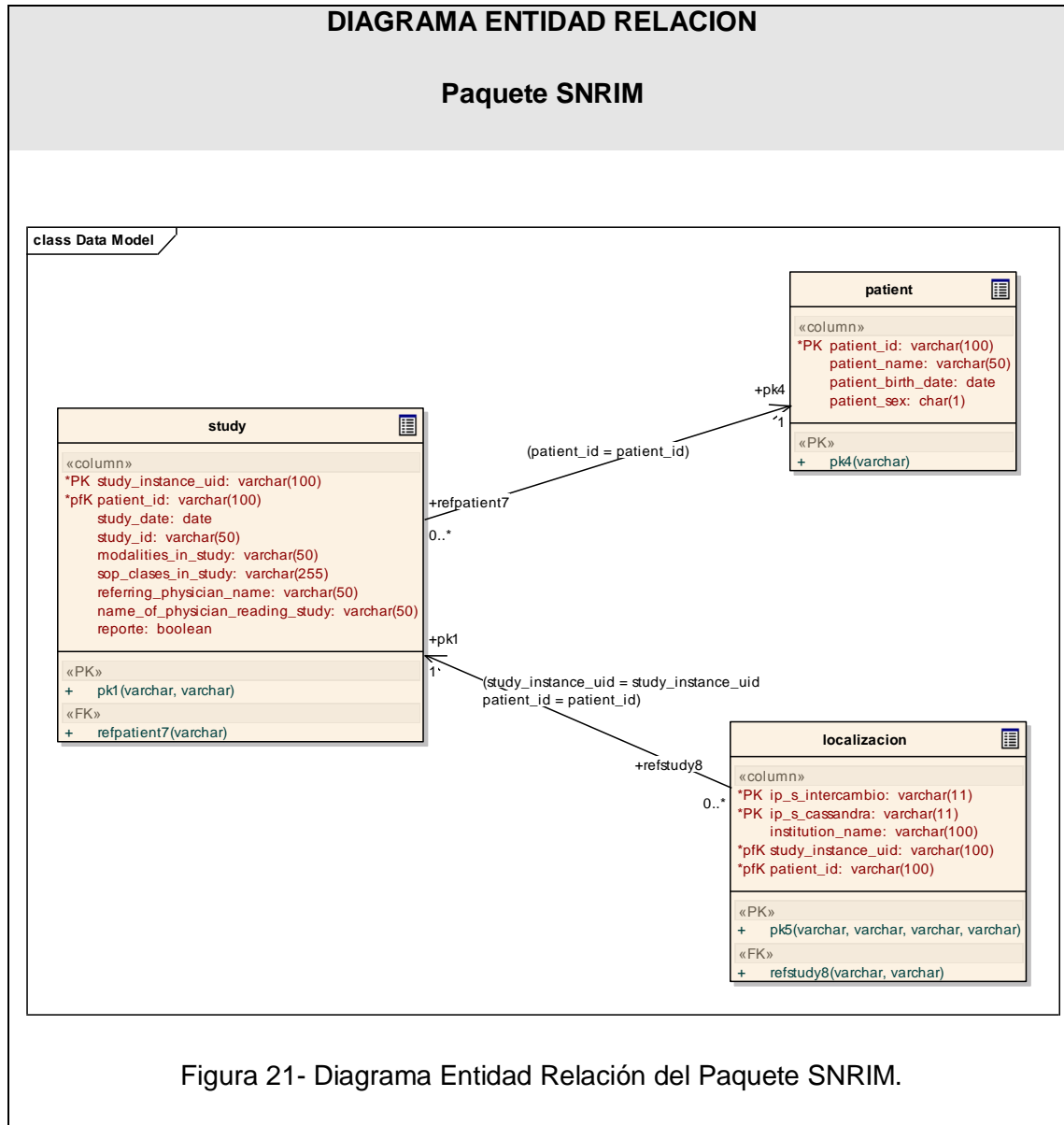


Figura 20 – Diagrama Entidad Relación del Paquete Orchestator.

3.4.1.2 Diagrama Entidad Relación de la Base de Datos de Referencias de Estudios Radiológicos.

En este diagrama se muestran las tablas que contienen la información que constituye en su conjunto las referencias de los estudios.



3.4.2 Descripción de las tablas de la Base de Datos.

En este epígrafe se describirán las tablas de las bases de datos correspondientes con sus respectivos atributos.

3.4.2.1 Base de Datos de Registro de Búsquedas

Nombre: Busqueda		
Descripción: Tabla que almacena la información de los registros de las búsquedas de referencias que se efectúen desde un hospital.		
Atributo	Tipo	Descripción
Id	integer	Id de la búsqueda.
ip_consultante	varchar	Ip del servidor que realiza la búsqueda.
ae_title	varchar	Identificados del Cassandra Server.
nombre	varchar	Nombre del paciente del cual se buscan las referencias.
fecha_busqueda	TIMESTAMP	Fecha de realización de la búsqueda.
sexo	char	Sexo del paciente en cuestión.
resultado	varchar	Dato que indica si la búsqueda tuvo resultados.

3.4.2.2 Base de Datos de Referencias de Estudios Radiológicos.

Nombre: localizacion		
Descripción: Esta tabla almacena todos los datos referentes a la localización física de los estudios radiológicos.		
Atributo	Tipo	Descripción
ip_s_intercambio	varchar	Ip del servidor de intercambio DMail.
ip_s_cassandra	varchar	Ip del servidor de Cassandra.
pnstitution_name	varchar	Nombre de la institución donde se encuentra el estudio.
patient_id	varchar	Id del paciente.
study_instance_uid	varchar	Identificador universal del estudio.

Nombre: patient		
Descripción: Tabla que almacena los datos del paciente.		
Atributo	Tipo	Descripción
patient_id	varchar	Id del paciente
patient_name	varchar	Nombre del paciente
patient_birth_date	date	Fecha de nacimiento
patient_sex	char	Sexo

Nombre: study		
Descripción: Tabla que guarda los datos referentes a los estudios radiológicos.		
Atributo	Tipo	Descripción
study_instance_uid	varchar	Identificador universal del estudio
patient_id	varchar	Id del paciente
study_date	Date	Fecha del estudio
study_id	varchar	Id del estudio
modalities_in_study	varchar	Modalidad del estudio
sop_clases_in_study	varchar	Par servicio objeto en el estudio
referring_physician_name	varchar	Nombre del médico referidor.
name_of_physician_reading_study	varchar	Nombre del médico que realizó el estudio
reporte	boolean	Indica si el estudio ha sido reportado en XWeb.

En este capítulo se mostraron los diagramas de clases de análisis y diseño, así como los diagramas de secuencia que le dan realización a los casos de uso. Además se mostraron los diagramas Entidad Relación de las dos bases de datos que se implementan, con la descripción de las entidades por las que están compuestas respectivamente.

Capítulo 4: Implementación.

En este capítulo se presenta la distribución física de los componentes que constituyen el sistema. Se explica el diagrama de componentes y luego el diagrama de despliegue, demostrando como se distribuirán físicamente los componentes, además de las relaciones de comunicación que existirán entre los nodos.

4.1 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. (16)

4.2.1 Diagrama de despliegue.

El diagrama de despliegue, ilustra la distribución física del sistema: donde se ubicarán los componentes, en qué servidores, máquinas o hardware, y vínculos de red, entre otros. Los componentes se distribuyen por nodos físicos y contienen a los componentes del sistema.

Los componentes de la gestión de las referencias se encontrarían en un servidor web, localizado a nivel nacional, que albergaría además la aplicación para buscar referencias desde la web. Estaría vinculado directamente con el servidor de Base de Datos de Referencias, también localizado nacionalmente.

Con el servidor web interactuarían el de XWeb y el Orchestator, que estaría desplegado en un servidor dentro de cada hospital, específicamente donde esté instalado DMail, aprovechando las características que tendría este.

Finalmente con el Orchestrator interactuaría el servidor DICOM Cassandra Server del hospital.

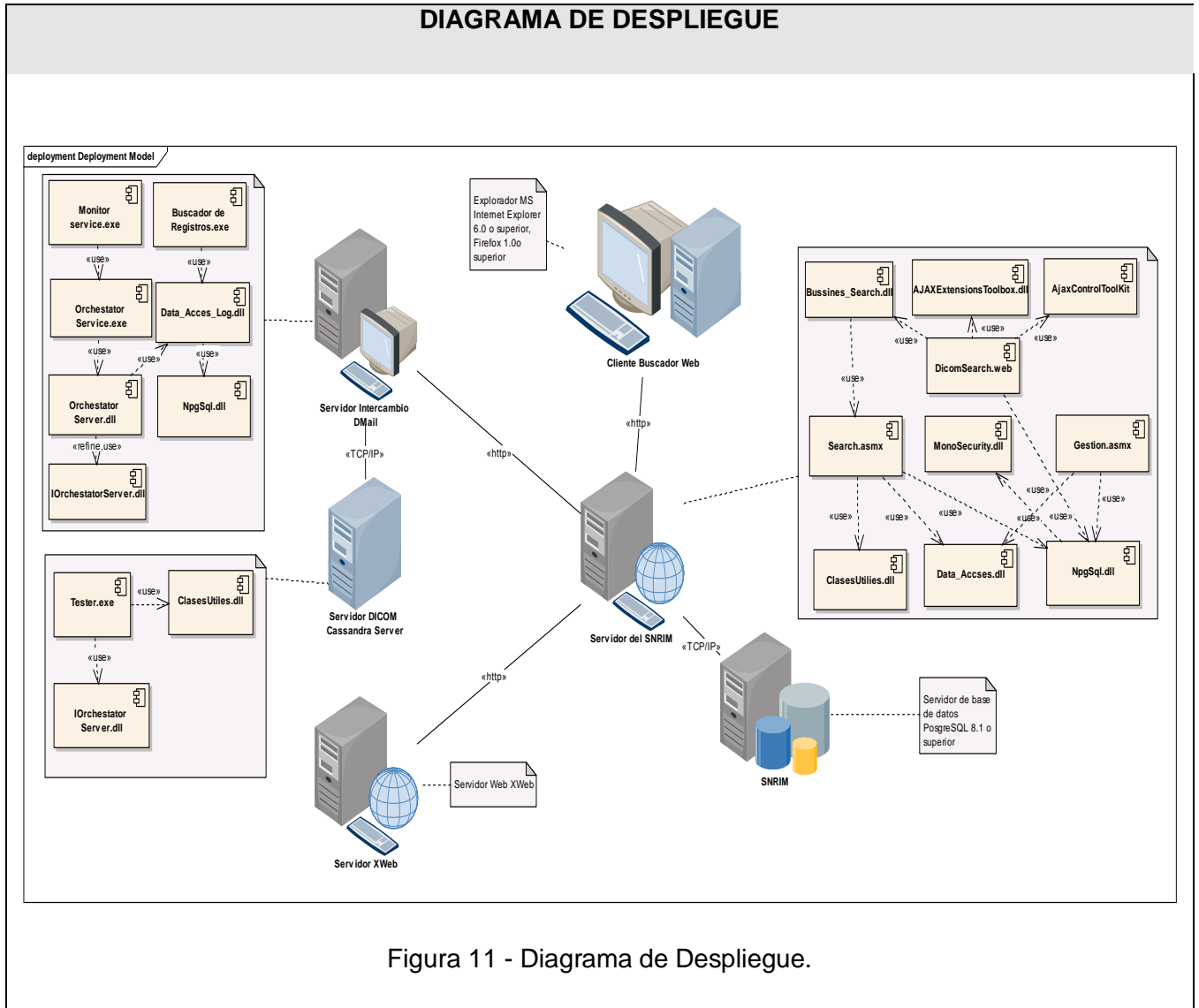


Figura 11 - Diagrama de Despliegue.

En el presente capítulo se presentaron los diagramas de componente y despliegue evidenciando los resultados finales del desarrollo del sistema y su distribución física.

Conclusiones.

La realización del presente trabajo ha permitido desarrollar un sistema capaz de almacenar y obtener referencias de todos los estudios imagenológicos realizados en el país. La solución está compuesta por el Sistema Nacional de Referencia de Imágenes Médicas y el Cassandra Orchestator y viene a satisfacer la necesidad de localizar cualquier estudio que se haya realizado en el país, además de poder realizar posteriormente la transferencia segura de un hospital a otro.

Otro logro de particular importancia es el de lograr el registro de todas las transferencias de imágenes entre hospitales, con lo que se garantiza la auditabilidad del proceso de transferencia entre los hospitales, además de la garantía implícita de confiabilidad y seguridad.

Esta solución permite en resumen, la creciente integración entre los componentes del sistema Cassandra PACS y constituye un paso significativo en el objetivo de lograr la conformación de una Red Nacional de Imágenes, sin perder el principio de responsabilidad hospitalaria por las imágenes almacenadas.

Recomendaciones.

- Someter el sistema a una fase de pruebas de calidad para su posterior despliegue.
- Implementen las funcionalidades o cambios propuestos en el presente trabajo para los sistemas de Cassandra PACS.
- Continuar el desarrollo de los elementos propuestos con vistas a mejorar la eficiencia e incrementar la fiabilidad y la estandarización en las comunicaciones.
- Observar el desarrollo de los estándares medico informáticos con el propósito de adaptarse a cualquier transformación o necesidad de cambio.
- Implementar una versión para sistemas operativos libres, aprovechando las potencialidades de la plataforma MONO.

Referencia Bibliográfica.

1. **Bibliomed.** Radiología. *buenasSalud*. [En línea] Bibliomed, 2000-2007. <http://www.buenasalud.com/lib/ShowDoc.cfm?LibDocID=3083&ReturnCatID=21>].
2. *Historia de la Radiología*. **Medina Delgadillo Edgar**. s.l. : Revista del Instituto Médico Sucre, 1996, Vol. 108.
3. **Anacleto, Lic. Valerio Adrián.** Introducción a UML 2.0. La evolución de la programación hacia la ejecución y validación automática de modelos. *Epidata Consulting*. [En línea] Epidata Consulting, Octubre de 2005. http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.
4. **Sparx Systems.** Index. Sparx Systems. *Sparx Systems*. [En línea] Sparx Systems, 2000-2007. <http://www.sparxsystems.com.ar/index.html>.
5. **Francisco Recio, David Provencio.** Ventajas de .Net. *DesarrolloWeb.com*. [En línea] DesarrolloWeb.com. <http://www.desarrolloweb.com/articulos/1329.php>.
6. **Microsoft.** Conceptos del lenguaje Visual C#. Por qué utilizar C#. *MSDN*. [En línea] Microsoft Corporation, 2007. http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cscon/html/vclrfWhyUseCSharp_PG.asp.
7. **Wikipedia.** Servicio Web. *Wikipedia. La enciclopedia libre*. [En línea] Wikipedia, 2007. http://es.wikipedia.org/wiki/Servicio_Web.
8. **IBM.** Colaboración empresarial con la arquitectura orientada a servicios (SOA). *IBM*. [En línea] IBM, 2007. <http://www-03.ibm.com/systems/es/z/soa/>.
9. **Francisco José García Peñalvo, Jaime González González, Iván Álvarez Navia, María N. Moreno García, Belén Curto Diego, Vidal Moreno Rodilla.** *Fundamentos para el desarrollo de aplicaciones distribuidas basadas en CORBA*. [Document] Salamanca. España. : Departamento de Informática y Automática Universidad de Salamanca. , 2002.

10. **Sacristán, Juan Ignacio Pérez.** Web Services: XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos. *Programación en castellano*. [En línea] Programación en castellano., 2006. <http://www.programacion.net/tutorial/xmlrpcsoap/4/>.
11. **Walzer, Carlos A.** Microsoft .NET Remoting Conceptualmente. *Microsoft.com*. [En línea] Microsoft Corporation, 2003. <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art42.asp>.
12. **Srinivasan, Paddy.** Introducción a Microsoft .NET Remoting Framework. *Microsoft.com*. [En línea] Microsoft Corporation, 2003. <http://www.microsoft.com/spanish/msdn/articulos/archivo/091101/voices/remoting.asp>.
13. **Microsoft.** Visual Studio. Introducción a Visual Studio. *MSDN*. [En línea] Microsoft Corporation, 2007. [http://msdn2.microsoft.com/es-es/library/fda2bad5\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fda2bad5(VS.80).aspx).
14. **Wikipedia, colaboradores de.** Modelo entidad-relación. *Wikipedia, La enciclopedia libre*. [En línea] Wikipedia, La enciclopedia libre, 2007. http://es.wikipedia.org/w/index.php?title=Modelo_entidad-relaci%C3%B3n&oldid=914669.
15. **Consultoría de Seguridad.** Diagramas de Componentes. *creangel.com*. [En línea] Consultoría de Seguridad. <http://www.creangel.com/uml/componente.php>.
16. **Sparx Systems Pty Ltd.** . El Modelo Físico. *sparxsystems Web site*. [En línea] Sparx Systems, 2000-2007. http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html.
17. **IBM.** IBM en España. *Colaboración empresarial con la arquitectura orientada a servicios (SOA)*. [En línea] IBM. <http://www-03.ibm.com/systems/es/z/soa/>.
18. **abarco@entel.es, Antonio Barco.** Principios de la orientación a servicios. *Arquitectura Orientada a Servicios (SOA)*. [En línea] blogspot.com, Diciembre de 2006. http://arquitecturaorientadaaservicios.blogspot.com/2006_06_01_archive.html.
19. **PostgreSQL, Equipo de Desarrollo.** Resumen. *PostgreSQL*. [En línea] Postgres Global Development Group, 1996-1999. <http://es.tldp.org/Postgresql-es/web/navegable/todopostgresql/preface.htm>.

Bibliografía.

PITMAN, N y PILONE, DAN. *UML 2.0 in a Nutshell*. 2005. s.l. : O'Reilly, 0-596-00795-7.

RUMBAUGH, J, JACOBSON, I y BOOCH, G. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000. 84-7829-036-2.

ACR-NEMA. *Estándar DICOM*. National Electrical Manufacturers Association, 2004. p.

BIOFÍSICA MÉDICA. Software Imagis, 2007]. Disponible en: <http://www.biofisicamedica.sld.cu/>

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

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

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

POSTGRESQL.ORG. About, 2007]. Disponible en: <http://www.postgresql.org/about/>

---. PostgreSQL 8.2.4 Documentation, 2007]. Disponible en: <http://www.postgresql.org/docs/current/static/release-8-2.html>

FOWLER, M. *Patterns of Enterprise Application Architecture*. Boston, Addison-Wesley, 2002. 560 p.

POSTGRESQL, EQUIPO DE DESARROLLO. PostgreSQL, 1996-9]. Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/todopostgresql/postgres.htm>

Sparx Systems, 2000-2007. Disponible en: <http://www.sparxsystems.com.ar/index.html>.

PostgreSQL. 1996-1999. Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/todopostgresql/preface.htm>.

Introducción a Visual Studio. *MSDN*. 2007. Disponible en: [http://msdn2.microsoft.com/es-es/library/fda2bad5\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fda2bad5(VS.80).aspx).

IBM, 2007. Disponible en: <http://www-03.ibm.com/systems/es/z/soa/>.

Anexos.

Anexo 1. Glosario de Términos.

Término	Grupo	Definición
DICOM	Técnico	Estándar para la visualización, almacenamiento y transmisión de imágenes en medicina. Norma las especificaciones que deben cumplir los sistemas para la transmisión, almacenamiento y visualización de imágenes médicas.
Cassandra	Técnico	Princesa de Troya (Mitología Griega) a quien le fue otorgado el don de la profecía por el dios Apolo.
PACS	Técnico	Sistema de Almacenamiento y Transmisión de Imágenes. (Picture Archiving and Communication System)
http	Técnico	Protocolo de transferencia de hipertexto (HyperText Transfer Protocol)
HL7	Técnico	Estándar de intercambio de datos electrónicos en el área de la salud, con especial énfasis en las comunicaciones inter-hospitalarias en el área de la información clínica y administrativa.
OMG	Técnico	Object Management Group formado en 1989 con el propósito de crear una arquitectura estándar para objetos distribuidos en redes. Actualmente busca el desarrollo de especificaciones para la industria del software.

Anexo 2: Diagramas de clases de Diseño.

A.2.1 Paquete Orchestator.

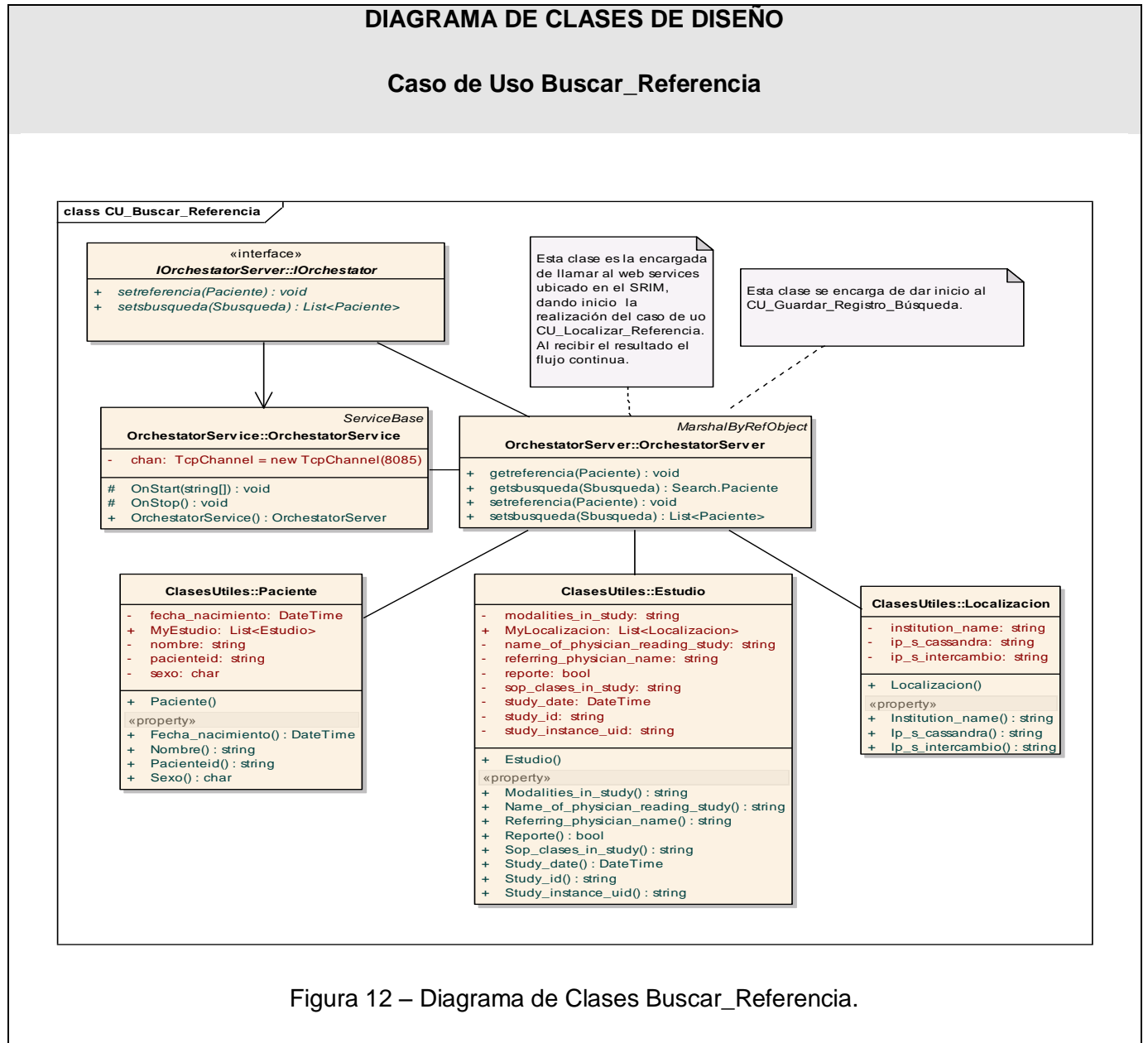


Figura 12 – Diagrama de Clases Buscar_Referencia.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso Visualizar_Registro_Búsqueda

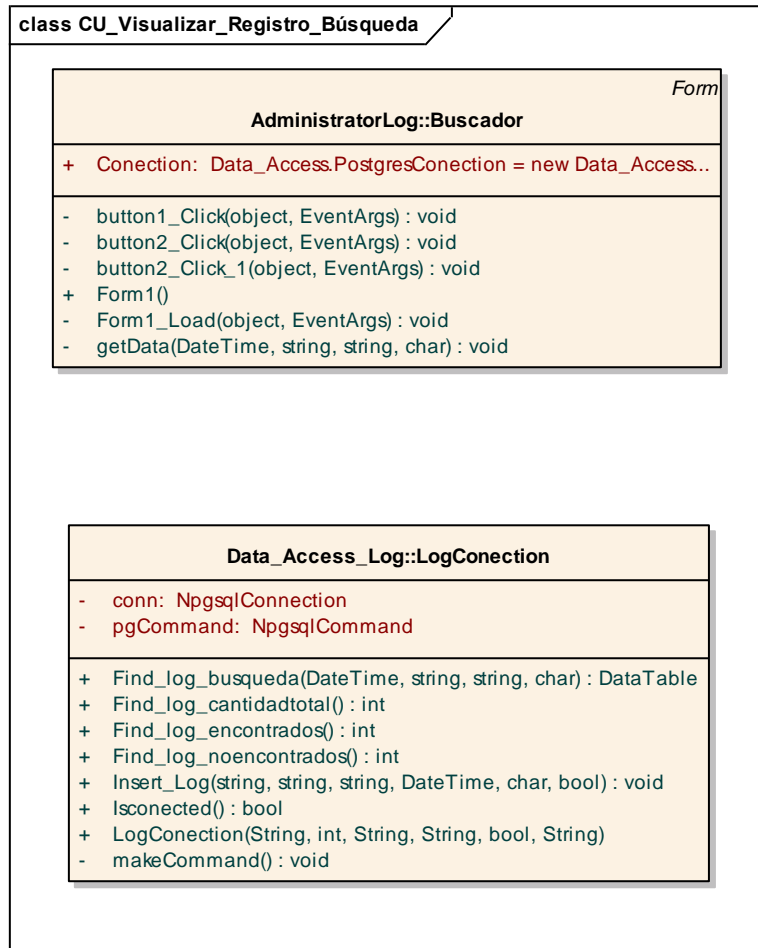


Figura 13 - Diagrama de Clases Visualizar_Registro_Búsqueda.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso Guardar_Registro_Búsqueda

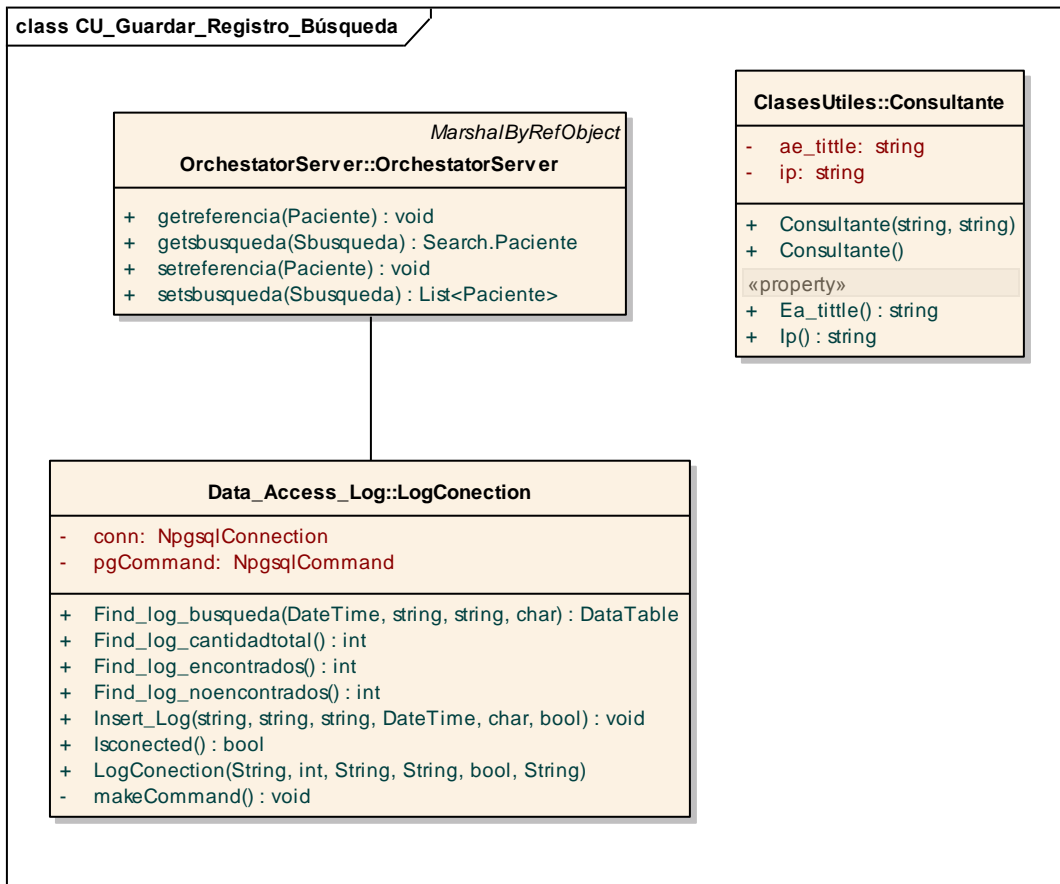


Figura 14 - Diagrama de Clases Guardar_Registro_Búsqueda.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso Insertar_Referencia

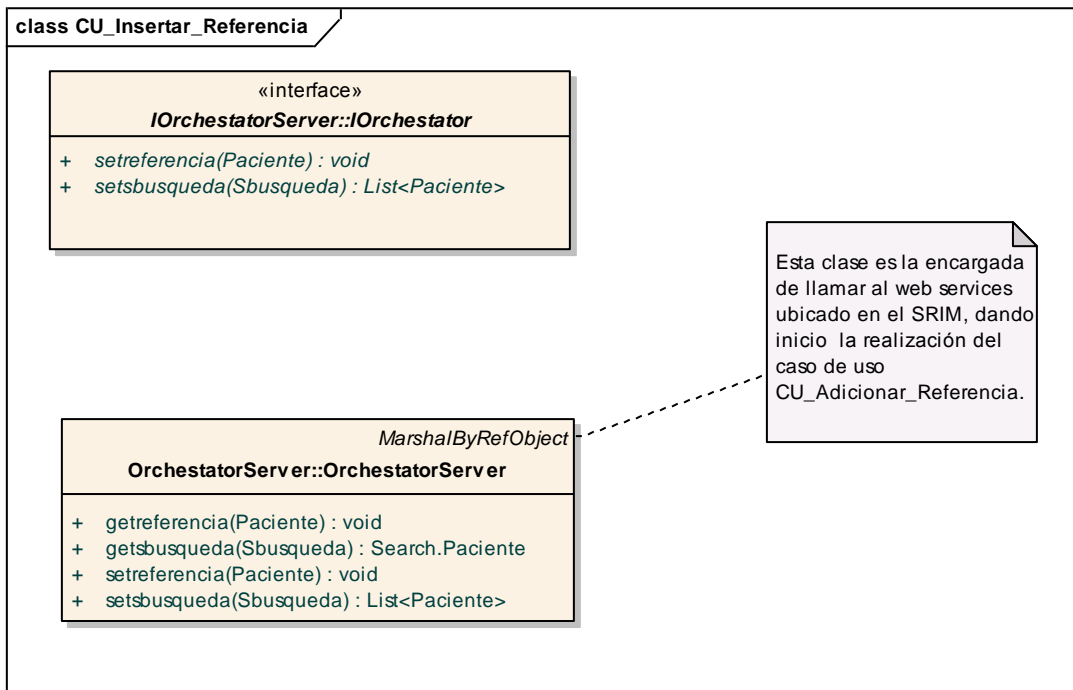


Figura 15 - Diagrama de Clases Insertar_Referencia.

A.2.2 Paquete SNRIM.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso Adicionar_Referencia

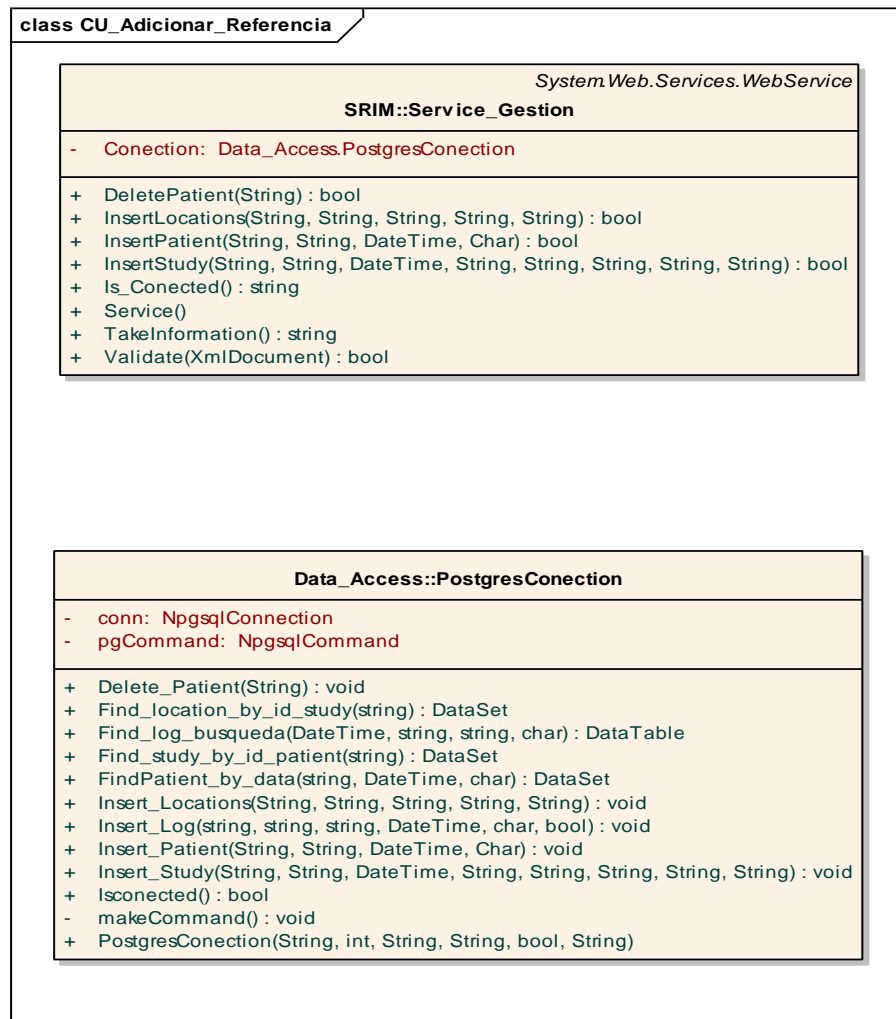


Figura 16 - Diagrama de Clases Adicionar_Referencia.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso AdicionarReporte_Referencia

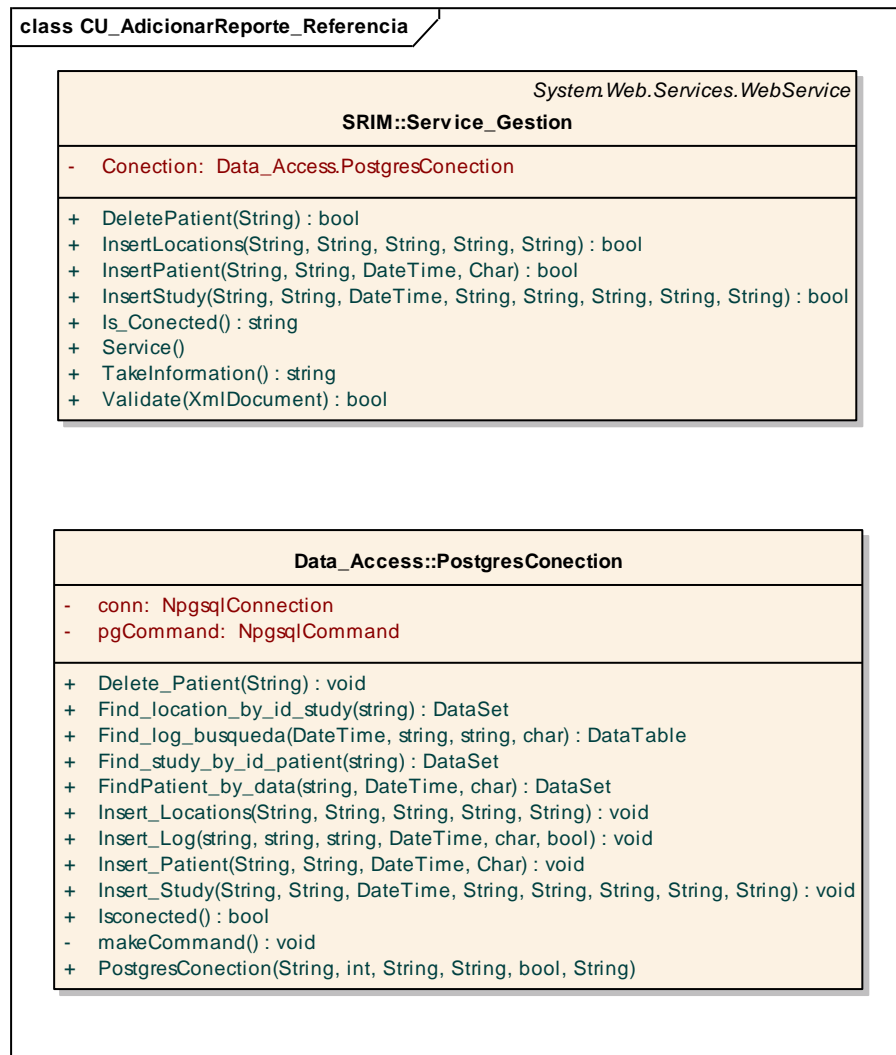


Figura 29 - Diagrama de Clases AdicionarReporte_Referencia.

DIAGRAMA DE CLASES DE DISEÑO

Caso de Uso Localizar_Referencia

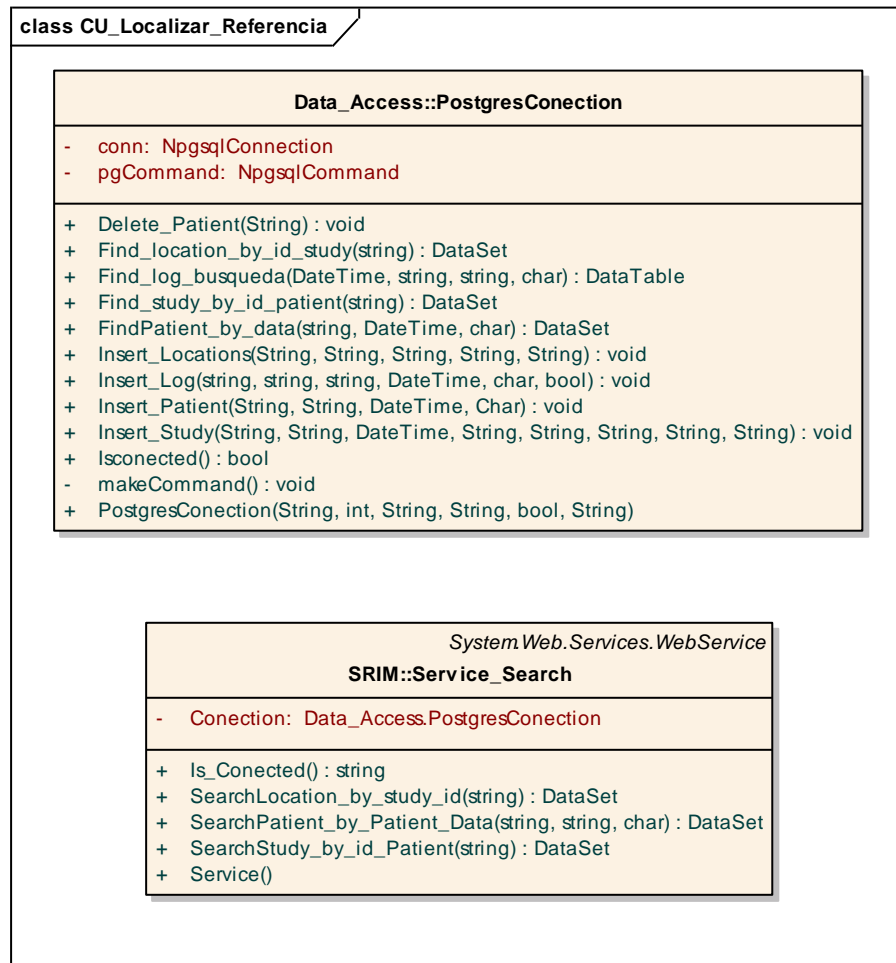


Figura 17 - Diagrama de Secuencia Localizar_Referencia.

A.2.3 Paquete Sistema de Búsqueda de Referencias.

