

Universidad de las Ciencias Informáticas
Facultad 2

Infraestructura de comunicación basada en los perfiles
XDS y PIX para el Centro de Soluciones de Informática
Médica.

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

Autores: Froilán Domínguez González
Alfredo José Chiong Zaldivar

Tutores: Ing. Juan Manuel Rey Alvarez
Ing. Amaya Alvarez Lorenzo

Consultante: MSc. Yovannys Sánchez Corales

La Habana, junio de 2014

“Año 56 de la Revolución

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 25 días del mes de junio del año 2014.

Alfredo José Chiong Zaldívar

Firma del autor

Froilán Domínguez González

Firma del autor

Ing. Juan Manuel Rey Alvarez

Firma del tutor

Ing. Amaya Alvarez Lorenzo

Firma del tutor

DATOS DE CONTACTO

Ing. Juan Manuel Rey Alvarez

Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2011. Especialista, miembro del Departamento de Gestión Hospitalaria del Centro de Informática Médica (CESIM). Actualmente se encuentra vinculado al desarrollo del Sistema de Información Hospitalaria *XAVIA HIS*.

Correo electrónico: jmrey@uci.cu

Ing. Amaya Alvarez Lorenzo

Ingeniera en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2012. Recién graduada en adiestramiento, miembro del Departamento de Gestión Hospitalaria del Centro de Informática Médica (CESIM). Actualmente se encuentra vinculada al desarrollo del Sistema de Información Hospitalaria *XAVIA HIS*.

Correo electrónico: aalorenzo@uci.cu

MSc. Yovannys Sánchez Corales

Ingeniero Informático, graduado en la Ciudad Universitaria “José Antonio Hecheverría” en el año 2005. Máster en Informática Aplicada, especialista y arquitecto, miembro del Departamento de Gestión Hospitalaria del Centro de Informática Médica (CESIM). Actualmente se encuentra vinculado al desarrollo del Sistema de Información Hospitalaria *XAVIA HIS*.

Correo electrónico: yscorales@uci.cu

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

Los sistemas informáticos enmarcados en el ámbito de la salud utilizan estándares sanitarios y buenas prácticas de integración como *HL7* e *IHE*. Esta iniciativa de integración se estructura por dominios de integración como el de Infraestructura de la Tecnología de la Información encargado entre otras funciones de compartir y gestionar la identidad de los pacientes y sus documentos clínicos mediante los perfiles *Patient Identifier Cross-Reference (PIX)* y *Cross-Enterprise Document Sharing (XDS)* respectivamente.

El Centro de Informática Médica de la Universidad de las Ciencias Informáticas desarrolla un Sistema de Información Hospitalaria y un Sistema de Información Radiológica los cuales mantienen una comunicación entre ellos, pero no sigue los estándares sanitarios. Existen algunos trabajos relacionados con la comunicación de sistemas de salud basados en las prácticas definidas por *IHE* como: *InterSystem Healthcare*, *MatchMetrix*, *Open eHealth*, la “Propuesta de aplicación de los perfiles de integración de *IHE* entre los sistemas alas *PACS* - alas *RIS* - alas *HIS*” y el “Modelo de Producción de Software para el Centro de Informática Médica” pero constituyen soluciones a la medida.

Es por ello que el objetivo de este trabajo es desarrollar una infraestructura de comunicación que implemente los perfiles de integración *XDS* y *PIX* del área de Infraestructura TI para el Centro de Informática Médica.

La infraestructura de comunicaciones desarrollada se integra correctamente al Sistema de Información Hospitalaria y arroja un rendimiento de 247.8 peticiones por segundo para 3000 conexiones y de 142.7 peticiones por segundo para 1000 conexiones simuladas.

Palabras claves:

Centro de Informática Médica, *Cross-Enterprise Document Sharing (XDS)*, Historia Clínica Electrónica (HCE), Iniciativa de Integración de Sistemas Sanitarios (*IHE*), *Patient Identifier Cross-Reference (PIX)*.

ÍNDICE

INTRODUCCIÓN	1
CAPITULO 1. FUNDAMENTACIÓN TEÓRICO-METODOLÓGICA SOBRE LAS INFRAESTRUCTURAS DE COMUNICACIONES BASADAS EN PERFILES DE INTEGRACIÓN SANITARIOS	6
1.1 Reseña histórica sobre las infraestructuras de comunicaciones en sistemas sanitarios.	6
1.2 Intercambio de información clínica en sistemas sanitarios.	6
1.3 Iniciativa de Integración de Sistemas Sanitarios.	7
1.3.1 Marco Técnico de la Iniciativa de Integración de Sistemas Sanitarios.	7
1.3.2 Dominios definidos por Iniciativa de Integración de Sistemas Sanitarios.	8
1.3.3 Perfiles de integración del dominio Infraestructura de la Tecnología de la Información.....	10
1.4 Infraestructuras de comunicación sanitarias existentes vinculadas a la investigación.....	16
1.5 Herramientas y Tecnologías.	18
1.5.1 Herramientas para el trabajo con los servicios web	18
1.5.2 Servidores de Aplicaciones.....	20
1.5.3 Canales de Servicios Empresariales.....	21
1.5.4 Entornos de desarrollo Integrado	23
1.5.5 Herramientas de modelado.....	24
1.5.6 Herramientas para realizar pruebas de rendimiento	24
1.6 Servicios web. Arquitectura orientada a servicios	25
1.7 Conclusiones parciales	26
CAPÍTULO 2. INFRAESTRUCTURA DE COMUNICACIÓN BASADA EN PERFILES DE INTEGRACIÓN PIX Y XDS	27
2.1 Componentes esenciales de la infraestructura de comunicaciones propuesta.....	27
2.2 Descripción de los actores presentes en la infraestructura de comunicación.....	28
2.2.1 Fuente de identificación de pacientes.....	28
2.2.2 Consumidor de identificación de paciente.....	32
2.2.3 Administrador de referencias cruzadas de identificación de paciente	33

2.2.4	Fuente de Documentos.....	35
2.2.5	Repositorio de Documentos.....	36
2.2.6	Consumidor de Documentos.....	38
2.2.7	Registro de Documentos	45
2.3	Esquema general de comunicaciones.	49
2.4	Clases principales incluidas en la infraestructura de comunicación.	51
2.5	Conclusiones	54
CAPÍTULO 3. VALIDACIÓN DE LA INFRAESTRUCTURA DE COMUNICACIÓN BASADA EN PERFILES DE INTEGRACIÓN PIX y XDS.....		55
3.1	Descripción de las interfaces correspondientes a los perfiles <i>XDS</i> y <i>PIX</i>	55
3.2	Aplicación de la infraestructura de comunicación en el Sistema de Información Hospitalaria	55
3.3	Pruebas de rendimiento.....	57
3.3.1	Pruebas aplicadas con la transacción Provide and Register Document Set.....	58
3.3.2	Pruebas aplicadas con la transacción Retrive Document Set	60
3.4	Conclusiones	62
CONCLUSIONES		64
RECOMENDACIONES.....		65
REFERENCIA BIBLIOGRAFICA		66

ÍNDICE DE TABLAS

Tabla 1.1 Perfiles de integración <i>IHE</i> y Actores probados por <i>MatchMetrix</i>	17
Tabla 1.2 Comparación de canales de servicios empresariales.....	21
Tabla 2.1 Identificador de las consultas a ejecutar en el Registro de Documentos	38
Tabla 2.2 Parámetros de la consulta <i>FindDocuments</i>	40
Tabla 2.3 Parámetros de la consulta <i>FindSubmissionSets</i>	40
Tabla 2.4 Parámetros de la consulta <i>FindFolder</i>	41
Tabla 2.5 Parámetros de la consulta <i>GetAll</i>	41
Tabla 2.6 Parámetros de la consulta <i>FetDocuments</i>	41
Tabla 2.7 Parámetros de la consulta <i>GetFolders</i>	42
Tabla 2.8 Parámetros de la consulta <i>GetAssociations</i>	42
Tabla 2.9 Parámetros de la consulta <i>GetDocumentosAndAssociations</i>	42
Tabla 2.10 Parámetros de la consulta <i>GetSubmissionSets</i>	42
Tabla 2.11 Parámetros de la consulta <i>GetSubmissionSetAndContents</i>	43
Tabla 2.12 Parámetros de la consulta <i>GetFolderAndContents</i>	43
Tabla 2.13 Parámetros de la consulta <i>GetFolderForDocument</i>	43
Tabla 2.14 Parámetros de la consulta <i>GetRelatedDocuments</i>	44
Tabla 2.15 Parámetros de la consulta <i>GetDocumentsByReferenceld</i>	44
Tabla 2.16 Tipos de asociaciones entre archivos	48
Tabla 2.17 Descripción de la entidad <i>HIS-CESIM</i>	50
Tabla 2.18 Descripción de la entidad <i>Pacientes</i>	50
Tabla 2.19 Descripción de la entidad <i>Sistemas</i>	50
Tabla 2.20 Descripción de la entidad <i>Documentos clínicos</i>	50

ÍNDICE DE FIGURAS

Figura 1.1 Tendencias del uso del canal empresarial <i>WSO2</i>	22
Figura 2.1 Propuesta de infraestructura de comunicaciones del CESIM	28
Figura 2.2 Segmento común del mensaje <i>PRPA_IN201301UV02</i>	29
Figura 2.3 Segmento <i>controlActProcess</i> del mensaje <i>PRPA_IN201301UV02</i>	30
Figura 2.4 Modelo de datos de la identificación de pacientes.....	31
Figura 2.5 Segmento <i>replacementOf</i> del mensaje <i>PRPA_IN201301UV02</i>	32
Figura 2.6 Segmento <i>controlActProcess</i> del mensaje <i>PRPA_IN201309UV02</i>	33
Figura 2.7 Segmento <i>acknowledgement</i> del mensaje <i>MCCI_IN000002UV01</i>	33
Figura 2.8 Segmento <i>controlActProcess</i> del mensaje <i>PRPA_IN201310UV02</i>	34
Figura 2.9 Segmento <i>patient</i> del mensaje <i>PRPA_IN201310UV02</i>	34
Figura 2.10 Segmento <i>asOtherIDs</i> del mensaje <i>PRPA_IN201310UV02</i>	34
Figura 2.11 Segmento <i>asOtherIDs</i> del mensaje <i>PRPA_IN201310UV02</i>	35
Figura 2.12 Mensaje de petición. Transacción <i>Provide And RegisterDocument Set</i>	35
Figura 2.13 Mensaje de petición. Transacción <i>RegisterDocumentSet</i>	36
Figura 2.14 Mensaje de respuesta. <i>Provide And RegisterDocument Set</i>	37
Figura 2.15 Mensaje de respuesta. <i>RetrieveDocumentSet</i>	37
Figura 2.16 Porción del mensaje de petición de la transacción <i>RegistryStoredQuery</i>	39
Figura 2.17 Porción del mensaje de petición de la transacción <i>RegistryStoredQuery</i>	39
Figura 2.18 <i>OR</i> semántico	39
Figura 2.19 <i>AND</i> semántico	39
Figura 2.20 Mensaje de petición. Transacción <i>RetrieveDocumentSet</i>	45
Figura 2.21 Mensaje de respuesta. Transacción <i>RegistryStoredQuery</i>	46
Figura 2.22 Porción del mensaje que representa un documento.....	47
Figura 2.23 Porción del mensaje que representa una carpeta.....	48
Figura 2.24 Porción del mensaje que representa una asociación.....	48

Figura 2.25 Porción del mensaje que representa la referencia a un objeto del registro.....	48
Figura 2.26 Modelo de datos correspondiente al registro de documentos clínicos.....	49
Figura 2.27 Esquema general de comunicaciones.....	50
Figura 2.28 Modelo de paquetes.....	51
Figura 2.29 Paquete de clases Infraestructura de comunicación.....	52
Figura 2.30 Paquete de clases HIS.....	53
Figura 3.1 Modificación del método <code>crearDocumentoClinico</code> de la clase <code>HCManager</code>	56
Figura 3.2 Modificación del método <i>persistir</i> de la clase <i>CrearHC</i>	56
Figura 3.3 Modificación del método <i>persistir</i> de la clase <i>ModificarHistoriaClinica</i>	56
Figura 3.4 Flujo para el registro y modificación de historia clínica.....	57
Figura 3.5 Ejecución de <i>Crear Historia Clínica</i> mediante la Infraestructura de comunicación.....	57
Figura 3.6 Informe agregado para la transacción <i>Provide and Register Document Set</i>	58
Figura 3.7 Gráfico de resultados para la transacción <i>Provide and Register Document Set</i>	59
Figura 3.8 Visualizador <i>Spline</i> para la transacción <i>Provide and Register Document Set</i>	60
Figura 3.9 Informe agregado para la transacción <i>Retrive Document Set</i>	61
Figura 3.10 Gráfico de resultados para la transacción <i>Retrive Document Set</i>	62
Figura 3.11 Visualizador <i>Spline</i> para la transacción <i>Retrive Document Set</i>	62

INTRODUCCIÓN

En la actualidad, la informática tiene un auge acelerado en el campo de la medicina, la cual sirve como base para el conocimiento científico técnico en aras de obtener beneficio de los servicios y atención de la salud, en la sociedad. Esto ha hecho que se creen diversos sistemas informáticos para gestionar la información que se maneja en las diferentes áreas de las instituciones hospitalarias, los cuales tienen como componente esencial la Historia Clínica Electrónica (HCE) (1) para su funcionamiento.

Los sistemas informáticos que gestionan la información generada en un hospital son denominados Sistemas de Información Hospitalaria (por sus siglas en inglés *HIS*) (2). Dichos sistemas permiten la recolección, almacenamiento, procesamiento, recuperación y comunicación de la información relacionada con la atención al paciente y la labor administrativa de todas las actividades del hospital o centro de atención médica.

Existen además en el área de la salud los Sistemas de Almacenamiento, Transmisión y Visualización de imágenes Médicas (por sus siglas en inglés *PACS*) (3), los cuales contribuyen al proceso de diagnóstico para los pacientes en el área de imaginología. Como complemento y mejora en el funcionamiento, la productividad y la precisión de los datos en todo el proceso de tratamiento de imágenes surgen los Sistemas de Información Radiológica (por sus siglas en inglés *RIS*) (4). Estos sistemas pertenecen al servicio de radiodiagnóstico, por tanto capturan y procesan los datos que se obtienen en el departamento de Radiología en una institución de salud.

Con el objetivo de integrar la información generada en los sistemas antes mencionados, así como lograr una correcta interoperabilidad (5) surgen los estándares de comunicación. El intercambio se logra a partir de protocolos como el Protocolo de Acceso Simple a Objetos (por sus siglas en inglés *SOAP*) diseñado para el intercambio de información en un entorno computacional distribuido a través de servicios web. Dos de los estándares más utilizados para la comunicación de sistemas de salud son *HL7 (Health Level Seven)* (6) para compartir documentos clínicos y *DICOM (Digital Imaging Communications in Medicine)* (7) para el trabajo con imágenes médicas digitales.

Los estándares antes mencionados por si solos no garantizan la comunicación, la confidencialidad, la seguridad ni definen los procesos de integración entre sistemas. Con el objetivo de normar su aplicación entre los sistemas surge la Iniciativa de Integración de Sistemas Sanitarios (por sus siglas en inglés *IHE*) (8), el cual establece dominios de integración, como Radiología, Laboratorio e Infraestructura de las Tecnologías de la Información. Cada uno de estos dominios tiene definidos perfiles de integración que

contienen elementos necesarios para lograr una correcta interoperabilidad entre los Sistemas de Información para la Salud.

El Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas desarrolla entre sus productos un Sistema de Información Hospitalaria y un Sistema de Información Radiológica. Actualmente dichos sistemas poseen entre ellos una comunicación basada en servicios web pero con diferentes comportamientos: desde el *HIS* hacia el *RIS* de forma asíncrona a través de una cola de mensajes (*broker-activemq*), sin embargo, en el sentido contrario la comunicación se realiza directamente.

Los servicios web del Sistema de Información Hospitalaria del CESIM exponen las funcionalidades asociadas al trabajo con los usuarios del sistema, información demográfica relacionada con los pacientes, la gestión de citas de radiología, así como la generación de reportes de reposo. Por otra parte los servicios del Sistema de Información Radiológica del CESIM se encargan del trabajo con usuarios del sistema, envío de nomencladores y calendarios para la gestión de citas, así como el reporte de consulta.

Para que la comunicación anteriormente mencionada pueda resultar, es necesario que ambos sistemas garanticen una disponibilidad parcial. Lo anterior puede traer consigo que la integridad de la información que se intercambia pueda ser afectada, es decir puede existir pérdida o alteración de la información.

El *HIS* posee un repositorio de HCE, así como los metadatos asociados a los documentos clínicos como parte de su funcionamiento. Mediante una observación directa, se aprecia que existe fuerte dependencia en la implementación del repositorio de HCE con los metadatos y la base de datos del sistema. Los metadatos contienen la información demográfica de los documentos clínicos, de forma tal que se dificulta la separación hacia un ambiente común para la utilización de estos elementos por otros sistemas como el *RIS*. Todo esto impide una mejor interoperabilidad entre ambos sistemas.

Adicionalmente el *HIS* incluye un mecanismo de Identificación Único de Pacientes (por sus siglas en inglés *MPI*) que se ejecuta como una tarea programada para verificar si existen pacientes duplicados, el mecanismo implementado, también basado en el juicio ingenieril, interrelaciona la información de los pacientes duplicados dándole un identificador único, pero es un algoritmo que contempla muchas variables y tiene un alto costo de ejecución (9). Producto de la infraestructura de comunicación existente entre ambos sistemas y la manera en que opera el *MPI* se dificulta notablemente el rendimiento del sistema durante la ejecución del mismo y no permite el cruce de referencias de identificadores de pacientes de diferentes dominios.

Por las situaciones antes mencionadas se define como **problema científico**: ¿Cómo estandarizar la estrategia de integración entre el Sistema Gestión Hospitalaria y el Sistema de Información Radiológica del Centro de Informática Médica?

Para dar cumplimiento al problema planteado se define como **objeto de estudio**: El intercambio de información clínica en los sistemas para la salud. Enmarcado en el **campo de acción**: El intercambio de información clínica en el área de integración Infraestructura de Tecnología de la Información entre el Sistema Gestión Hospitalaria y el Sistema de Información Radiológica del Centro de Informática Médica.

El **objetivo general** de la investigación es desarrollar una infraestructura de comunicación que implemente los perfiles de integración *XDS* y *PIX* del área de Infraestructura TI para el Centro de Informática Médica.

Para dar cumplimiento al objetivo general, se plantean las siguientes **tareas de investigación**:

1. Identificación de los estándares de comunicación y perfiles de integración en el área de intercambio de documentos clínicos.
2. Análisis de las principales infraestructuras para el intercambio de información clínica basada en estándares y buenas prácticas para la comunicación entre sistemas.
3. Implementación de la infraestructura de comunicación entre el Sistema Gestión Hospitalaria y el Sistema de Información Radiológica del Centro de Informática Médica.
4. Validación de la infraestructura de comunicación desarrollada.

Para lograr el pleno y óptimo desarrollo de la investigación se hace indispensable el empleo y uso de métodos científicos, específicamente los que se evidencian a continuación:

Métodos teóricos: (10) permiten estudiar las características del objeto de investigación que no se observan directamente, además crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad. Se aplican en calidad de enfoque general como estrategia. De estos se utilizarán:

- **Analítico-sintético**: se realizó el estudio de la literatura especializada relacionada con el tema y la exploración de resultados de investigaciones afines, lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.

- **Revisión documental:** se realizó una revisión de los documentos que sustentan el proceso de comunicación entre sistemas de salud a nivel internacional, para una mayor preparación y desarrollo de la investigación.

Métodos empíricos: (10) estos métodos explican las características fenomenológicas del objeto, con ellos es posible representar un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Se aplican como procedimiento en el proceso de investigación. De los métodos empíricos existentes se utilizarán:

- **Entrevistas no estructuradas o libres:** se le realizaron entrevistas a especialistas del Centro de Informática Médica, así como al líder de la comunidad de desarrollo de SOA en la Universidad, con el objetivo de obtener información relacionada con la infraestructura de comunicación y los diferentes conceptos y perfiles a utilizar, lo que contribuye al perfeccionamiento de la propuesta realizada.
- **Observación:** mediante este método se valoraron los avances realizados en el estudio de las diferentes herramientas y tecnologías a usar, además se identificaron los perfiles de integración adecuados para dar respuesta a la problemática y cómo funcionan.

El documento se encuentra estructurado en 3 capítulos, de la siguiente manera:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICO-METODOLÓGICA: Se describen los diferentes conceptos que aborda el campo de acción del problema científico, se muestran breves descripciones de sistemas que implementan la iniciativa de *IHE* para la integración de sistemas de salud dando a conocer el estado del arte y sus características esenciales. Se fundamenta la selección de tecnologías, metodologías y un grupo de herramientas utilizadas en el desarrollo de este trabajo.

CAPÍTULO 2: INFRAESTRUCTURA DE COMUNICACIÓN: Se describe la propuesta de solución que consiste en una arquitectura basada en servicios web para la implementación de los perfiles PIX y XDS del dominio de Infraestructura TI, los cuales son los necesarios para compartir documentos clínicos entre sistemas sanitarios. Se describen los actores que están implicados en los distintos escenarios, la forma en que se intercambiará la información en las transacciones entre dichos actores y la manera en que se emplearán las tecnologías seleccionadas en el capítulo 1.

CAPÍTULO 3: VALIDACIÓN DE LA INFRAESTRUCTURA DE COMUNICACIÓN: Se describe el proceso de validación de la infraestructura de comunicación propuesta teniendo en cuenta la aplicación de dicha Infraestructura en el Sistema de Información Hospitalaria. Además se realiza un análisis de rentabilidad de

los servicios que ofrece dicha plataforma, pruebas de carga mediante la herramienta *Jmeter* así como la correcta comprobación del flujo que siguen los mensajes de los servicios invocados por los diversos actores mediante la herramienta *Soap Tracer*.

CAPITULO 1. FUNDAMENTACIÓN TEÓRICO-METODOLÓGICA.

En el presente capítulo se describen los diferentes conceptos que aborda el campo de acción del problema a resolver, se muestran breves descripciones de sistemas que implementan la iniciativa de *IHE* para la integración de sistemas de salud dando a conocer el estado del arte y sus características esenciales. Se fundamenta la selección de tecnologías, metodologías y un grupo de herramientas utilizadas en el desarrollo de este trabajo.

1.1 Reseña histórica sobre las infraestructuras de comunicaciones en sistemas sanitarios.

La Iniciativa de Integración de Sistemas Sanitarios (*IHE*) (11) se creó en 1998 por parte de usuarios y empresas de Estados Unidos para dar respuesta a los crecientes problemas de interoperabilidad en el dominio de radiología siendo adoptada poco tiempo después por Europa y Asia. Las Sociedades de usuarios *RSNA* (*Healthcare and Information Management Systems Society*) y *HIMSS* (Sociedad Radiológica de Norteamérica) crearon una única plataforma (*IHE*) para que los usuarios y vendedores pudieran definir especificaciones sobre Sistemas de Información Sanitarios que permitan la interoperabilidad entre aplicaciones complejas.

1.2 Intercambio de información clínica en sistemas sanitarios.

Para lograr el intercambio de información clínica en sistemas sanitarios se han creado un conjunto de estándares para normar el almacenamiento y comunicación de la información. Uno de los estándares más utilizados es *HL7* (*Health Level Seven*) (6) para el intercambio de documentos clínicos. Dentro de la iniciativa de *HL7* también se encuentra el estándar Arquitectura Clínica de Documentos (por sus siglas en inglés *CDA*) (6) desarrollado por la organización *HL7* y forma parte de la versión 3 del estándar. *CDA* es un estándar de marcaje para definir la estructura y semántica de un documento clínico que se requiere intercambiar entre distintos sistemas.

El estándar *DICOM* (12) permite que distintos sistemas puedan interactuar en conjunto, mediante la especificación de normas para la manipulación, el almacenamiento, la impresión y la transmisión de información digital en la Imagenología Médica. *DICOM* incluye la definición de un formato de fichero y de un protocolo de comunicación de red. Estos ficheros pueden intercambiarse entre dos entidades que tengan capacidad de recibir imágenes y datos de paciente en formato *DICOM*. El protocolo de comunicación es un protocolo de aplicación que usa *TCP/IP* para la comunicación entre sistema.

En la actualidad la tendencia mundial es a la implementación de un nuevo estándar *FHIR* (*Fast Healthcare Interoperability Resources*) (13). *FHIR* es la nueva propuesta de especificación de estándares *HL7* para el

intercambio electrónico de información en salud; cuyo surgimiento obedece a las dificultades de implementación de *HL7 V3*, a la necesidad de transición de *HL7 V2.x* y los retos que suponen las nuevas tendencias como *Mobile Health* y aplicaciones de telefonía móvil, el intercambio de registros electrónicos de salud o historia clínica electrónica, comunicaciones en la nube, entre otros. *FHIR* combina las mejores características de las líneas de producto *HL7* versión 2, versión 3 y *CDA*, al tiempo que aprovecha los últimos estándares web, aplicado un enfoque estricto de implementación.

A pesar de la tendencia de uso del estándar *FHIR* para la comunicación de información clínica entre sistemas de salud, para dar respuesta a la problemática planteada se escoge *HL7 v2.3*, para lograr la compatibilidad con los sistemas del CESIM los cuales son el objeto de integración. Por otra parte *HL7* es un estándar altamente probado, teniendo mucha aceptación por parte de las instituciones que implementan sistemas de salud, así como por las instituciones que implementan *IHE*.

1.3 Iniciativa de Integración de Sistemas Sanitarios.

La Iniciativa de Integración de Sistemas Sanitarios *IHE* (8) es una iniciativa de profesionales de la salud y de la industria para mejorar la forma de compartir información sanitaria entre los sistemas informáticos. *IHE* promueve el uso coordinado de las normas y estándares establecidos tales como *DICOM* y *HL7* acorde a las necesidades clínicas específicas en el apoyo a la atención de los pacientes. Además está diseñado para estimular la integración de los sistemas de información que apoyan a las instituciones de salud modernas con el objetivo de asegurar que en el proceso de atención al paciente toda la información médica esté correcta y disponible para los profesionales de la salud.

1.3.1 Marco Técnico de la Iniciativa de Integración de Sistemas Sanitarios.

Los Marcos de Trabajo (*Technical Frameworks*) de *IHE* son un grupo detallado de documentos (11) que guían a los desarrolladores de sistemas de información y a los integradores para poder implementar las capacidades establecidas. Estas buenas prácticas de integración permiten una efectiva integración de sistemas, así como facilitar el intercambio apropiado de información médica y de apoyo para el cuidado óptimo del paciente. Uno de los marcos de trabajo es el de infraestructura.

El Marco Técnico de Infraestructura (*Infrastructure Technical Framework*) identifica un subconjunto de componentes funcionales de las empresas de salud llamados actores *IHE*, y especifica sus interacciones en términos de un conjunto de coordinadas transacciones basadas en estándares. La definición *IHE* de un actor no debe tomarse como la definición completa de cualquier producto que pueda ponerlo en práctica, y

no debe tomarse el propio marco para describir exhaustivamente la arquitectura de un sistema de información sanitaria.

Los actores y transacciones descritas en el Marco Técnico de Infraestructura son abstracciones del mundo real en el entorno de los sistemas de información en salud. Mientras algunas de las transacciones son tradicionalmente realizadas por categorías específicas de productos como: Sistemas de Información Hospitalaria, Repositorio de Datos Clínicos, Sistemas de Información Radiológica, Sistemas de Información Clínica o Información Cardiológica. Este marco técnico evita intencionalmente asociar las funciones o actores con las categorías de productos antes mencionadas.

La definición de actores y transacciones proporciona una base para las interacciones entre los componentes funcionales del entorno del sistema de información de salud. En situaciones en las que un producto físico único implementa múltiples funciones, sólo las interfaces entre el producto y las funciones externas en el medio ambiente se consideran significativas por la iniciativa *IHE*.

1.3.2 Dominios definidos por Iniciativa de Integración de Sistemas Sanitarios.

Con el objetivo de organizar la información gestionada por sistemas sanitarios, *IHE* define dominios que pueden ser clínicos u operacionales. En cada dominio los usuarios y proveedores de sistemas de salud identifican las necesidades de integración e intercambio para ese dominio según la experiencia clínica acumulada. Este proceso se realiza con el fin de llegar a un consenso sobre cuáles serán las soluciones acorde a las necesidades y sobre cuales estándares se apoyará cada solución.

Cada dominio incluye una comisión técnica y su principal tarea es desarrollar y documentar las distintas soluciones (conocidas como perfiles de integración). Existe además una comisión de planificación, cuyo principal papel consiste en la planificación, despliegue y la organización de actividades. También mantienen y desarrollan su propio conjunto de documentos del Marco Técnico. La coordinación entre los dominios es responsabilidad del comité de coordinación de dominio, integrado por representantes de cada una de las comisiones y los comités técnicos.

Actualmente *IHE* tiene definido 12 dominios identificados como se mencionan a continuación:

- **Anatomic Pathology** (8) aborda el intercambio de información, flujo de trabajo y la atención al paciente en Patología, incluyendo la patología anatómica. Este dominio está desglosado en 3 perfiles aún en desarrollo.

- **Cardiology** (8) se creó en 2003 para abordar los problemas específicos de flujo de trabajo clínico, el intercambio de información y la mejora de la atención al paciente en el ámbito clínico de la cardiología. El dominio Cardiología de *IHE* se convirtió en estado latente durante el ciclo de desarrollo 2008-2009. Este dominio está desglosado en 4 perfiles estables y 6 perfiles aún en desarrollo.
- **Dental** (8) aborda la implementación de las soluciones de interoperabilidad basadas en estándares para mejorar el intercambio de información, flujo de trabajo y la atención al paciente. Este dominio está desglosado en 2 perfiles estables.
- **Eye Care** (8) se formó en 2005 y es patrocinado por la Academia Americana de Oftalmología (AAO). La misión del dominio *IHE* Eye Care es reunir a las partes interesadas de tecnología de información y profesionales de la salud para aplicar las normas para la comunicación de información de los pacientes de manera eficiente a lo largo de las instalaciones de salud, grandes y pequeñas. Este dominio está desglosado en 4 perfiles estables y 3 perfiles aún en desarrollo.
- **IT Infrastructure** (8) se refiere a la implementación de las soluciones de interoperabilidad basadas en estándares para mejorar el intercambio de información, flujo de trabajo y la atención al paciente. Este dominio está desglosado en 21 perfiles estables y 12 perfiles aún en desarrollo.
- **Laboratory** (8) aborda el intercambio de información y el flujo de trabajo en relación con las pruebas de diagnóstico in vitro en el laboratorio clínico, así como en el punto de atención. Este dominio está desglosado en 6 perfiles estables y 2 perfiles aún en desarrollo.
- **Patient Care Coordination** (8) fue establecido en 2005 acorde a los problemas de integración que atraviesan los proveedores, los problemas de los pacientes o el tiempo. Se ocupa de los aspectos de cuidados clínicos generales, tales como el intercambio de documentos, procesamiento de pedidos, y la coordinación con otros ámbitos de especialidad. Este dominio está desglosado en 4 perfiles estables y 16 perfiles aún en desarrollo.
- **Patient Care Devices** (8) fue creado en 2005 de acuerdo con la integración de los dispositivos médicos en la empresa de salud, desde el punto de atención a la HCE, que podrían resultar en mejoras significativas en la seguridad del paciente y la calidad de la atención. Este dominio está desglosado en 5 perfiles estables y 4 perfiles aún en desarrollo.
- **Pharmacy** (8) aborda el intercambio de información, flujo de trabajo y la atención al paciente en las farmacias, tanto comunitarias como hospitalarias. Este dominio está desglosado en 5 perfiles aún en desarrollo.
- **Quality, Research and PublicHealth** (8) se ocupa del intercambio de información y de los estándares de contenido de historiales médicos electrónicos necesarios para compartir información

relevante para mejorar la calidad en la atención al paciente, la investigación clínica y la vigilancia de la salud pública. Este dominio está desglosado en 15 perfiles aún en desarrollo.

- **Radiation Oncology** (8) aborda el intercambio de información, flujo de trabajo y la atención al paciente en oncología de radiación. Este dominio está desglosado en 2 perfiles estables y 5 perfiles aún en desarrollo.
- **Radiology** (8) se creó en 1998 para abordar los problemas de la interoperabilidad y el intercambio que impactan la calidad de la atención en la información de las imágenes médicas. Este dominio está desglosado en 19 perfiles estables y 10 perfiles aún en desarrollo.

El dominio de Infraestructura de las Tecnologías de la Información es quien define en sus perfiles las buenas prácticas para el intercambio de documentos clínicos entre sistemas de salud existentes en el Centro de Informática Médica. Por lo tanto se ajusta al objetivo que persigue la presente investigación debido a que es un dominio de carácter horizontal a los sistemas de salud mencionados. A continuación se describen los perfiles de integración que pertenecen al dominio TI.

1.3.3 Perfiles de integración del dominio Infraestructura de la Tecnología de la Información.

Los perfiles de integración *IHE* describen una necesidad clínica de integración de sistemas y la solución para llevarla a cabo. Define también los componentes funcionales, los que se denominan actores *IHE*, y especifica con el mayor grado de detalle posible las transacciones propias para cada actor, basadas siempre en estándares como *DICOM* y *HL7*.

Cada perfil de integración *IHE* tiene como objetivo organizar y aprovechar las capacidades de integración que se pueden lograr mediante la aplicación coordinada de las normas de comunicación expuestas en los diferentes estándares. Ellos proporcionan definiciones precisas de cómo las normas pueden aplicarse para satisfacer las necesidades específicas de una situación clínica determinada.

El dominio de Infraestructura de la Tecnología de la Información define 21 perfiles los cuales se describen a continuación:

- **Sharing Value Sets (SVS)** (14) que proporciona un medio a través del cual los sistemas de salud que producen o consumen datos clínicos o administrativos, tales como equipos de diagnóstico por imagen, sistemas de información de laboratorio, o sistemas de registro de salud nacional pueden recibir una nomenclatura uniforme común de gestión centralizada.

- **Retrieve Information for Display (RID)** (15) permite un acceso fácil y rápido a la información del paciente para una mejor atención. Es compatible con el acceso a los documentos persistentes existentes en formatos de presentación de renombre tales como *CDA*, *PDF*, *JPEG*, etc.
- **Enterprise User Authentication (EUA)** (15) define un medio para establecer un nombre por usuario que puede ser utilizado en todos los dispositivos y sistemas en un dominio de integración definido. Esto facilita enormemente la gestión de la autenticación de usuario centralizada y proporciona a los usuarios la comodidad y la velocidad de un inicio de sesión único.
- **Patient Identifier Cross-referencing (PIX)** (15) soporta la referencia cruzada de los identificadores de los pacientes de múltiples dominios. Estas referencias son enviadas por fuentes de identificación de pacientes y utilizadas por sistemas “consumidor de identidades” para relacionar información acerca de un paciente. Esto permite que un clínico tenga visión más completa de la información del paciente.
- **Patient Synchronized Applications (PSA)** (16) permite la selección de un paciente único para el usuario que trabaja en múltiples aplicaciones en una estación de trabajo. Con este perfil de integración la selección de un paciente en cualquiera de las aplicaciones hace que todas las otras aplicaciones sincronicen el mismo paciente.
- **Consistent Time (CT)** (17) define los mecanismos para sincronizar la base de tiempo entre múltiples actores y las computadoras. Varias infraestructuras, seguridades y perfiles de adquisición requieren el uso de una base de tiempo consistente en varios equipos.
- **Patient Demographics Query (PDQ)** (18) ofrece formas para múltiples aplicaciones distribuidas de consultar un servidor de información de paciente para obtener una lista de los mismos, basada en criterios de búsqueda definidos por el usuario.
- **Audit Trail and Node Authentication (ANTA)** (19) establece las características básicas de un Nodo de Seguridad Básica, describe el entorno de seguridad asumido por el nodo de modo que los evaluadores de la seguridad pueden decidir si esto coincide con su entorno.
- **Cross-Enterprise Document Sharing (XDS)** (20) permite que una serie de organizaciones de prestación de asistencia sanitaria que pertenezcan a un dominio de afinidad *XDS* (por ejemplo, una red de hospitales, el sistema de salud de una nación) colaboren en el cuidado de un paciente mediante el intercambio de historias clínicas en forma de documentos a medida que avanzan las actividades de atención a sus pacientes.
- **Personnel White Pages Profile (PWP)** (15) proporciona acceso a la información básica del directorio de usuarios. Esta información tiene un amplio uso entre muchas aplicaciones clínicas y no clínicas a través de las empresas de salud.

- **Cross-Enterprise User Assertion (XUA)** (21) proporciona un medio para comunicar las afirmaciones sobre la identidad de un nodo autenticado (usuario, aplicación, sistema) en las transacciones entre las empresas.
- **Patient Administration Management (PAM)** (22) establece la continuidad e integridad de los datos del paciente, y la información adicional, tal como las personas vinculadas (cuidador primario, familiares, etc.) Coordina el intercambio de registro de pacientes y la información de actualización de los sistemas.
- **Cross-Enterprise Document Reliable (XDR)** (23) proporciona el intercambio de documentos mediante un sistema de mensajería confiable. Esto permite el intercambio de documentos directo entre los *EHR*, *PHR*, y otros sistemas informáticos de salud en la ausencia de una infraestructura de intercambio de documentos, tales como registro de y repositorios de *XDS*.
- **Cross-Enterprise Document Media Interchange (XDM)** (24) proporciona el intercambio de documentos con un archivo común y la estructura de directorio a través de varios medios de comunicación convencionales.
- **Retrieve Form for Data Capture (RFD)** (25) proporciona un método para la recopilación de datos dentro de la aplicación de un usuario para cumplir los requisitos de un sistema externo. *RFD* apoya la recuperación de las formas a partir de una fuente de forma, la visualización y cumplimentación de un formulario, y la devolución de datos de la instancia de la aplicación de visualización de la aplicación de origen.
- **Cross-Community Access (XCA)** (26) apoya los medios para consultar y recuperar los datos médicos pertinentes a los pacientes en poder de otras comunidades. Una comunidad se define como un acoplamiento de instalaciones/empresas que se han comprometido a trabajar juntos utilizando un conjunto común de políticas con el fin de compartir información clínica a través de un mecanismo establecido.
- **Basic Patient Privacy Consents (BPPC)** (27) proporciona un mecanismo para registrar el consentimiento de privacidad del paciente, y un método para los consumidores de contenido a utilizar para hacer cumplir el consentimiento de privacidad apropiada para el uso.
- **Scanned Documents Integration Profile (XDS-SD)** (15) es el encargado de estandarizar los metadatos de la variedad de formas de almacenamiento de información clínica y lograr una asociación estructurada de la misma.
- **Patient Identifier Cross-referencing HL7 V3 (PIXV3)** (28) La funcionalidad de este perfil es igual al perfil *PIX*. La diferencia se encuentra en el formato de los mensajes y en el uso de *SOAP-920* basado en servicios webs.

- **Patient Demographics Query HL7 V3 (PDQV3)** (29) La funcionalidad de este perfil es igual al perfil *PDQ*. La diferencia se encuentra en el formato de los mensajes y en el uso de *SOAP-920* basado en servicios webs.
- **Multi-Patient Queries (MPQ)** (30) define un mecanismo para habilitar las consultas globales a un registro de documentos en función de determinados criterios requeridos por las áreas relacionadas con el análisis de datos, tales como la acreditación de la calidad de los profesionales de atención de salud o centros de salud, recolección de investigación de ensayos clínicos o datos de vigilancia de la salud de la población.

Luego del análisis de cada uno de los perfiles descritos e incluidos en el dominio TI, se seleccionan para ser implementados en el presente trabajo los perfiles *XDS*, el cual permite crear un dominio de afinidad para lograr el intercambio de historias clínicas entre las aplicaciones informáticas del CESIM; así como *PIX* para lograr la identificación de pacientes de múltiples dominios, como pueden ser los atendidos en el área de radiología que son gestionados por el Sistema de Información Radiológica. A continuación se abordan en profundidad cada uno de ellos.

1.3.3.1 Cross-Enterprise Document Sharing (XDS)

El perfil *XDS* les permite a una serie de organizaciones de prestación de asistencia sanitaria que pertenezcan a un dominio de afinidad *XDS* (por ejemplo, una comunidad de la atención) colaboren en el cuidado de un paciente mediante el intercambio de historias clínicas en forma de documentos a medida que avanzan las actividades de prestación de atención de sus pacientes. Un repositorio de documentos federados y un registro de documentos tributan a una búsqueda longitudinal de la información acerca de un paciente en un determinado dominio de afinidad *XDS*. Este perfil define los actores que se destacan a continuación:

Consumidor de Documentos

El Consumidor de Documentos (*Document Consumer*) (31) es todo aquel sistema que inicia una transacción con el fin de obtener uno o más documentos clínicos. Este actor intercambia información con dos actores, el Registro de Documentos (31) y el Repositorio de Documentos (31). El intercambio de información con el Registro de Documentos se lleva a cabo según lo definido por la transacción *Registry Stored Query*, la cual garantiza que el Consumidor de Documentos obtenga en el mensaje de respuesta, una lista de todos los documentos que cumplen con los parámetros especificados en la petición. El intercambio con el Repositorio de Documentos se realiza según la definición de la transacción *Retrieve*

Document Set, pues garantiza que se realice una petición por un documento y que éste se envíe en el mensaje de respuesta.

Fuente de Documentos

La Fuente de Documentos (*Document Source*) (31) es todo aquel sistema que inicia una transacción con el fin de enviar uno o más documentos para su almacenamiento. Este actor solo intercambia información con el actor Repositorio de Documentos y envía la información mediante la transacción *Provide & Register Document Set-b*.

Registro de Documentos

El Registro de Documentos (*Document Registry*) (31) constituye el sistema que recibe una petición para el almacenamiento de los metadatos de uno o más documentos o para la consulta de los que han sido almacenados con anterioridad. Este actor intercambia información con los actores Consumidor de Documentos y con el Repositorio de Documentos. La información que intercambia con el Consumidor de Documentos lo hace mediante la transacción *Registry Stored Query*, pues permite que sean consultados los metadatos de los documentos que han sido almacenados con anterioridad. A través de la transacción *Register Document Set-b* se intercambia información con el actor Repositorio de Documentos del cual recibe los metadatos de los mismos.

Repositorio de Documentos

El Repositorio de Documentos (*Document Repository*) (31) es el sistema que recibe uno o más documentos y sus metadatos con el fin de almacenarlos. Después de recibir dichos documentos, son almacenados y comienzan una transacción con el objetivo de enviar los metadatos recibidos al actor Registro de Documentos.

1.3.3.2 Patient Identifier Cross-referencing (PIX)

PIX soporta la referencia cruzada de los identificadores de los pacientes de múltiples dominios. Estas referencias pueden ser utilizadas por sistemas consumidores de identidad para relacionar información acerca de un paciente que está registrado en más de un sistema y en cada uno posee un identificador distinto.

Además *PIX* está dirigido a empresas de salud de gran tamaño (hospital, una clínica, un consultorio médico, entre otros) Es compatible con la referencia cruzada de identificadores de pacientes de varios dominios de identificación del paciente a través de las siguientes interacciones:

- La transmisión de la información de la identidad del paciente a partir de una fuente de identidad para el Administrador de Referencias Cruzadas (*PIX Manager*).
- La capacidad para acceder a la lista(s) de identificación de pacientes con referencias cruzadas (*PIX*) mediante una consulta/respuesta o mediante una notificación de actualización.

Al encapsular estos comportamientos en un solo actor (Administrador de Referencias Cruzadas), este perfil de integración proporciona la necesaria interoperabilidad, manteniendo la flexibilidad para ser utilizado con cualquier política de referencias cruzadas, así como el algoritmo que se estime conveniente por la empresa. El perfil *PIX* define los siguientes actores para lograr la identificación de pacientes en un dominio de integración:

Fuente de Identificación de Paciente

La fuente de identificación de pacientes (*Patient Identity Source*) (31) es el actor que se encarga de iniciar las transacciones para notificar tanto al Administrador de Referencias Cruzadas de Identificación Pacientes (*Patient Identifier Cross-reference Manager*) y al Registro de Documentos de cualquier acción de identificación relacionado con el paciente, incluyendo la creación, modificación o fusión.

Administrador de Referencias Cruzadas de Identificación de Pacientes

El actor Administrador de Identificación de Referencias Cruzadas de Pacientes (*Patient Identifier Cross-reference Manager*) (31) es el encargado de gestionar la identificación del paciente a través de dominios de identificación de los mismos, proporcionando una lista de identificación del paciente (*alias*) a través de la notificación a una lista configurada de consumidores de identificación de referencias cruzadas de pacientes interesados. Este actor puede, bajo pedido, devolver una lista de identificadores de pacientes correspondientes si existiesen.

Consumidor de Identificación de Referencias Cruzadas de Pacientes

El Consumidor de Identificación de Referencias Cruzadas de Pacientes (*Patient Identifier Cross-reference Consumer*) (31) es quien recibe notificaciones del Administrador de Identificación de Referencias Cruzadas de Pacientes de los cambios a los *alias* de identificación de pacientes. Normalmente, el actor Consumidor de Identificación de Referencias Cruzadas de Pacientes utiliza esta información para mantener los vínculos de información sobre los pacientes en un dominio diferente de identificación del paciente. Además es el encargado de consultar el Administrador de Identificación de Referencias Cruzadas de Pacientes para obtener una lista de identificadores de paciente correspondiente en caso de existir.

1.4 Infraestructuras de comunicación sanitarias existentes vinculadas a la investigación.

Actualmente tanto América del Norte como en la región europea son áreas donde la iniciativa *IHE* ha alcanzado mayor desarrollo. Con el objetivo de potenciar la propuesta de *IHE* se desarrollan diversas actividades destinadas a mejorar la integración entre Sistemas de Información Sanitarios como: *Connectathon* (32) que es un evento internacional donde se conectan sistemas para comprobar el correcto funcionamiento. *IHE* en los sistemas desplegados pertenecientes a las instituciones sanitarias. También incluye un foro de donde usuarios, empresas y administraciones públicas que tienen relación con los Sistemas de Información Sanitarios pueden compartir información con el fin de tratar cualquier temática referente a la integración de los sistemas. Para un mejor entendimiento se decide separar en dos secciones el análisis realizado acerca de otras variantes de infraestructura de comunicación relacionada con la actual investigación.

Internacionales

Inter Systems Health Share® (33) es una plataforma informática de la salud que proporciona tecnología avanzada para la interoperabilidad estratégica y analítica para la acción dentro de una sola instalación, a través de una red de hospitales, o del sistema de salud de una nación entera. Es una solución con pleno objeto comercial, creada en el año 2006 por la empresa *InterSystems* en USA, líder mundial en software para la atención sanitaria. Actualmente se encuentra en su versión 1.5 la cual recibió la certificación de *IHE USA* (34), de interoperabilidad TI sanitaria y de seguridad. Ha sido uno de los primeros productos de tecnología sanitaria en recibir la certificación de la interoperabilidad, ayudando a las organizaciones sanitarias en su avance hacia la atención sanitaria conectada.

MatchMetrix (35) es una plataforma de integración de la compañía *NexGate*, creada en enero del 2012 y se encuentra actualmente en su versión 7.3.1. Incluye un *NextGate EMPI* que funciona como un Índice Maestro de Pacientes en los *IHE Connectathons* donde demostró exitosamente su interoperabilidad con la Referencia Cruzada para Identificar Pacientes y la Búsqueda Demográfica para Pacientes (*PIX / PDQ*). Ha trabajado con numerosos vendedores para cambiar información de identificación del paciente de acuerdo a escenarios clínicos definidos. La siguiente tabla indica los perfiles de integración *IHE* y actores para los que *MatchMetrix* ha sido aprobado. Este producto implementa todas transacciones definidas por la estructura técnica *IHE* para soportar los perfiles *IHE* de Integración y los actores enumerados a continuación:

Tabla 1.1 Perfiles de integración IHE y Actores probados por MatchMetrix

Fuente: Tomado de NextGate (35)

Perfil de Integración Implementado	Actor Implementado
<i>PaTlent IdenTIfier Cross-referencing for MPI</i>	<i>PaTlent IdenTIty Cross-reference Manager</i>
<i>PaTlent IdenTIfier Cross-reference HL7 V3</i>	<i>PaTlent IdenTIty Cross-reference Manage</i>
<i>PaTlent Demographics Query</i>	<i>PaTlent Demographics Supplier</i>
<i>PaTlent Demographics Query HL7 V3</i>	<i>PaTlent Demographics Supplier</i>
<i>Cross Community PaTlent Discovery</i>	<i>Responding Gateway</i>
<i>Audit Trail and Node AuthentIcaTIon</i>	<i>Secure ApplicaTIon</i>
<i>Consistent TIme</i>	<i>TIme Client</i>

La Plataforma de Integración *Open eHealth* (36) es una extensión de la ruta *Apache Camel* que viene con soporte completo para el procesamiento de mensajes y sistemas de información de conexión en el sector sanitario. Está disponible en la versión 2.0 de *Apache License*, la misma se centra en un lenguaje específico de dominio (*DSL*) para implementar los patrones de integración empresarial (*CIE*) en soluciones de integración. *Open eHealth* se basa en los lenguajes de programación *Java* y *Groovy* para el desarrollo de aplicaciones y para el diseño de los *DSL* específicos del proyecto.

Esta plataforma ofrece bloques de construcción de alto nivel para la implementación de perfiles *IHE* como *XDS*, *PIX*, *PDQ* o *XCPD*. Otros estándares soportados son *HL7v2* y *HL7v3* (*CDA* y *CCD*) para los que ofrece lenguajes específicos de dominio (*DSL*) para la creación, el análisis, la representación, la navegación y la modificación.

- **Nacionales**

Con motivo de proponer una forma de integración de los sistemas *HIS* – *RIS* – *PACS* del CESIM se realizó la tesis de grado “Propuesta de aplicación de los perfiles de integración de *IHE* entre los sistemas alas *PACS* - alas *RIS* - alas *HIS*” (37) de los autores Martha Rodríguez López y Raymundo Rodríguez García, que propuso teóricamente la aplicación de los perfiles de integración de *IHE* entre estos sistemas. Esta propuesta analiza algunos de los diferentes perfiles de integración, y profundiza el dominio de radiología, por lo que la solución que propone resuelve solo los problemas de los procesos de interoperabilidad entre los sistemas sanitarios que intercambian Información Radiológica.

Como parte del trabajo de diploma para optar por la categoría de Master en Gestión de Proyectos titulado “Modelo de Producción de Software para el Centro de Informática Médica” del autor Ing. Reinier Alonso

González (38) se propone una arquitectura basada en servicios web poniendo en práctica el dominio de Infraestructura TI a través de un puente (*bridge*) de comunicación. El objetivo de la propuesta es lograr una correcta interoperabilidad entre los sistemas del CESIM haciendo uso de los perfiles *XDS*, *PIX* y *QED*.

A partir del estudio realizado sobre los sistemas y soluciones informáticas relacionadas con el objeto de estudio se arribaron a las siguientes conclusiones.

- La utilización de los sistemas *Inter Systems Health Share®* y *MatchMetrix* no dan solución a la problemática planteada, ya que son plataformas de integración privativas y no se puede extraer sus funcionalidades para lograr el principal requisito que es la comunicación entre los sistemas del CESIM.
- La propuesta de *Open eHealth* no es la adecuada para dar respuesta a la problemática planteada ya que su solución está atada al conjunto de herramientas *Apache Camel* (39), y el marco de trabajo *Spring* (*Spring Framework*) (40) que dificultan su integración con los sistemas del centro y no es posible desplegarla en el conjunto de herramienta de *WSO2*.
- La propuesta realizada en la UCI por los autores Martha Rodríguez López y Raymundo Rodríguez García solo se enfoca en dar solución a la transmisión de imágenes clínicas del área de radiología, no hace énfasis en ninguno de los perfiles definidos dentro del dominio de Infraestructura TI, que son los que responden a la necesidad del intercambio de documentos clínicos entre los sistemas de gestión de información para la salud.
- El *bridge* propuesta en el trabajo “Modelo de Producción de Software para el Centro de Informática Médica” está estrechamente relacionado con el Sistema de Información Hospitalaria del CESIM, por lo tanto dificulta la utilización del mismo por las otras aplicaciones informáticas. Además dicha investigación no está puesta en práctica pues su desarrollo es parcial.

1.5 Herramientas y Tecnologías.

Partiendo del requisito principal de integrar los sistemas del CESIM se acometió al estudio de las tendencias tecnológicas que permiten cumplir el objetivo propuesto. A continuación se describen los elementos principales de las herramientas y de dichas tecnologías.

1.5.1 Herramientas para el trabajo con los servicios web

Entre las herramientas más utilizadas se encuentran *Apache CXF v3.0* (41) la cual es un marco de servicios de código abierto que ayuda a construir y desarrollar servicios utilizando las *API* de

programación de interfaz, como *JAX-RS* y *JAX-WS*. Soporta una variedad de estándares de servicios web, incluyendo *SOAP*, el perfil básico *WS-I*, *WSDL*, *WS-Addressing*, *WS-Policy*, *WS-ReliableMessaging*, *WS-Security*, *WS-SecurityPolicy*, *WS-SecureConversation* y *WS-Trust*.

Otra de estas herramientas es *JBossWS* (42), el cual constituye un marco de trabajo de servicios webs desarrollado como parte del servidor de aplicaciones *Jboss* (*Jboss Server*). La integración con *JBossWS* proporciona el servidor de aplicaciones las tecnologías de servicios web que necesitan para lograr el cumplimiento de la plataforma *Java Enterprise Edition 7* (*Java EE 7*). *JBossWS* incluyen numerosas características, normas de implementación, herramientas para mejorar la facilidad de uso, la gestión de punto final, el monitoreo y seguimiento, proporciona mediante el *JBossWS-CXF*, la mayor parte de las características que vienen con *Apache CXF* (incluyendo *WS-Security*, *WS-Policy*, *WS-Addressing*, *WS-ReliableMessaging*, *WS-Trust*, *MTOM*).

Otra herramienta es *Apache Axis2* v1.5.2 (43), que constituye un motor de servicios web implementado en el lenguaje de programación *C*. Se basa en la arquitectura *Axis2* extensible y flexible, además utiliza su propio modelo de objeto los cuales fluyen en el *API* para *XML* que analiza para alcanzar una velocidad perceptiblemente mayor que versiones anteriores del eje de *Apache*. *Apache Axis2* no solo soporta *SOAP* 1.1 y *SOAP* 1.2, sino que también integra soporte para el popular estilo *REST* (*Representational State Transfer*) para servicios web. Una misma implementación de la lógica de negocios puede ofrecer a la vez una interfaz al estilo *WS-** como también un acceso al estilo *REST/POX* (*Plain Old XML*).

Por su parte *NuSOAP* (44) es un conjunto de herramientas (*Toolkits*) para desarrollar servicios web bajo el lenguaje *PHP*. Está compuesto por una serie de clases que hacen mucho más fácil el desarrollo de dichos servicios. Provee soporte para el desarrollo de clientes (aquellos que consumen los servicios web) y de servidores (aquellos que los proveen). Esta biblioteca es una de las más populares en la construcción de servicios web en *PHP* pues tiene soporte para diferentes versiones de *SOAP*, como también es posible crear servicios web que generen los descriptores de los web services (*WSDL*) y soporte de tipos de datos

Para dar solución a la propuesta se utilizará *Apache CXF* v3.0 con la implementación de *JAX-WS* por la diversidad de protocolos que soporta como:

- *WSDL Bindings*: *MTOM*, *PureXML*, *SOAP 1.1*, *SOAP 1.2*
- *Frontends*: Simple, clientes dinámicos, anotaciones, *JAX-WS/JAX-RS*
- *Data Bindings*: *MTOM*, *SDO*, *XMLBeans*, *JAXB*, *Aegis*
- *Transportes*: Local, *JMS*, *HTTP*, *Camel*

Además la flexibilidad que ofrece permite el uso de la configuración básica (*JAX-WS*) o especificar el motor o mecanismo utilizado (*CXF*) para ajustar los servicios a algunos ya existentes y no dejar ésta tarea al servidor de aplicaciones o a las particularidades de los fabricantes de los mismos. También porque permite controlar el cuerpo de los mensajes establecidos por *IHE* a lo largo de toda la cadena de procesamiento del mismo, es decir, manipular aún la respuesta en el instante previo al que sea enviado al cliente.

1.5.2 Servidores de Aplicaciones

Un servidor de aplicaciones (*Application Server*) (45) es un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Es el encargado de gestionar la mayor parte o la totalidad de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. A continuación se describen algunos de los principales servidores de aplicación en la actualidad:

Oracle GlassFish Server 3.0.1 (46) fue construido sobre el servidor *GlassFish* de código abierto, el cual ofrece una plataforma *Java EE 6* flexible, ligera y extensible. *Oracle GlassFish Server* forma parte de la cartera de *Oracle Fusion Middleware* y es ideal para aplicaciones que requieren un rendimiento rápido y ágil. Permite a las organizaciones crear e implementar aplicaciones web con el perfil web de *Java EE 6* ligero y sacar partido fácilmente de la potencia de toda la plataforma para aplicaciones de empresa. Los desarrolladores también pueden beneficiarse del modelo de programación simplificado y de las mejoras en la productividad que ofrece *Java EE 6*.

Otro de los servidores más utilizados es *Apache Tomcat v7* (47) es un servidor web multiplataforma que funciona como contenedor de *servlets* y que se desarrolla bajo el proyecto denominado *Jackarta* perteneciente a la *Apache Software Foundation* bajo la licencia *Apache 2.0* y que implementa las especificaciones de los *servlets* y de *JavaServer Pages (JSP)* de *Sun Microsystem*. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos establecidos por la *Apache Software Foundation*.

El servidor de aplicaciones de *WSO2 v5.2.1 (WSO2 AS)* (48) es la herramienta para el despliegue de componentes, los cuales son servicios web desarrollados en *axis2*, *jax-ws*, *spring-ws*, entre otros; también sirve para desplegar aplicaciones web tradicionales y servicios *restful*. Las aplicaciones web desplegadas en esta plataforma y que fueran pensadas para ello pueden acceder a diversas funcionalidades que brinda la herramienta.

Basado en los datos expuestos anteriormente además que permite compartir fácilmente la lógica de negocio, datos y procesos a través de todo el ecosistema de TI se escoge como servidor de aplicaciones a utilizar el *WSO2 AS* en su versión 5.2.1.

1.5.3 Canales de Servicios Empresariales

Un canal de servicios empresariales (por sus siglas en inglés *ESB*) (49) (50) es una arquitectura de software de capa intermedia (*middleware*) que proporciona servicios fundamentales para arquitecturas más complejas. Incorpora las funciones necesarias para implementar una arquitectura orientada a servicios (*SOA*). El concepto central de la arquitectura *ESB* es que permite integrar diferentes aplicaciones, esto desacopla los sistemas entre sí, que les permite comunicarse sin depender del conocimiento de otros sistemas existentes.

Entre los distintos *ESB* existentes se encuentra *Mule ESB* (51), el cual es la plataforma de integración de código abierto más popular. El bajo costo de *Mule ESB* junto con una fácil configuración, ampliación y flexibilidad hacen que sea muy popular. Está basado en la plataforma *Java*. Incluye también un poderoso conjunto de esquemas *XML*. La creación de los flujos de integración es de manera sencilla.

También se destaca *JBoss ESB v4.2 GA* (51) que aprovecha las tecnologías *JEMS* como el motor de reglas de negocio *JBoss Drools* para el enrutamiento basado en contenido y mensajería. Este enrutamiento basado en contenido puede utilizar *Jboss Drools* o *XPath*. Soporta *XSLT* y el motor de la transformación *Smooks* para *XML*, además de formatos de datos que no son *XML*. *JBoss ESB* se ejecuta en el servidor de aplicaciones *JBoss Server*.

Otro de los canales más utilizados es *WSO2 ESB v4.7.0* (52) basado en *WSO2 Carbon*, soporta varios protocolos de transporte, admitiendo la comunicación entre diferentes aplicaciones. Incluye transformación y enrutamiento de mensajes, así como la gestión de la seguridad. Permite a los administradores de sistemas y arquitectos de *SOA* de forma sencilla, el enrutamiento de mensajes, la virtualización, la intermediación, la transformación, la programación de tareas, el balanceo de carga, y otras funciones. El tiempo de ejecución ha sido diseñado para ser totalmente asíncrona, sin bloqueo, basado en el motor de mediación *Apache Synapse*. A continuación se muestra una tabla comparativa con los *ESB* antes mencionados.

Tabla 1.2 Comparación de canales de servicios empresariales

Fuente: Tomado de (53)

Características	WSO2 ESB & Plataforma SOA	Mule ESB	JBoss ESB & Plataforma SOA
Soporte de patrones de integración empresarial	Si	Si	Si
Ofrece todas las características necesarias en un ESB (servicios web, transformación de mensajes, mediación, protocolo de enrutamiento de contenido)	Si	Si	Si
Ofrece una plataforma SOA completa y coherente (ESB, MessageBroker, Registro de Gobierno, Proceso de negocio de servidores, servicios de servidor de base de datos, servidor de aplicaciones)	Si	No	Si
Gobierno de SOA	Si	No	No
Espacio gráfico de desarrollo en el ESB	Si	Si	Si
Basado en una arquitectura componible	Si	No	No
Plataforma de integración en la nube (iPaaS)	Si	Si	No
Conectores de la nube y adaptadores heredados	Si	Si	Si
Rendimiento	Alto	Moderado	Moderado
Administración de seguridad e identidades	Si	Limitado	Limitado
Modelo de negocio abierto	Si	Si	No

Luego de analizar algunos de los canales de servicios empresariales más utilizados y teniendo en cuenta la tabla comparativa antes mostrada, el canal de servicio empresarial a utilizar para dar solución en esta investigación es *WSO2 ESB* en su versión 4.7.0, pues demuestra ser el más potente y completo de todos, con más prestaciones y mayor rendimiento. Además un análisis (54) evidenciado en la siguiente gráfica sustenta el motivo de elección.

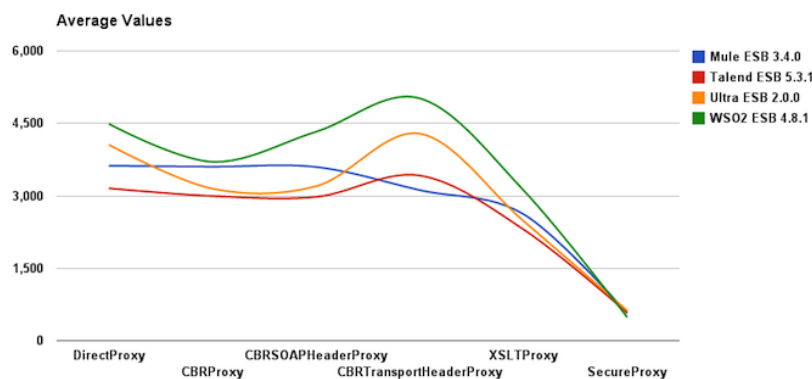


Figura 1.1 Tendencias del uso del canal empresarial *WSO2*

Fuente: Tomado de Desarrollo SOA en la Comunidad *WSO2-Latinoamérica* (54)

1.5.4 Entornos de desarrollo Integrado

Los Entornos de Desarrollos Integrados (por sus siglas en inglés *IDE*) facilitan el desarrollo ágil de programas informáticos para la web o sistemas de escritorios. Entre la gama de entornos de desarrollo existentes se destaca el sistema privativo *Visual Studio* (55), por ser uno de los más utilizados, es de fácil manipulación y posee un lenguaje de programación entendible para cualquier persona. También se destaca el *Netbeans* (56), que es multilenguaje, multiplataforma, permite desarrollar casi todo tipo de aplicaciones java como pueden ser aplicaciones móviles.

JBoss Developer Studio (57) es otro de los entornos de desarrollo comúnmente utilizados, proporciona un rendimiento superior para todo el ciclo de vida de desarrollo. Incluye un amplio conjunto de capacidades de las herramientas y soporte para múltiples modelos y marcos de programación como *Java Enterprise Edition 6 y 7*, *RichFaces*, *Java Server Faces*, *Enterprise Java Beans*, *Java Persistence API*, *Hibernate*, *JAX-RS con REST Easy*, contextos inyección de dependencia, *HTML5*, y muchas otras tecnologías populares. Está totalmente probado y certificado para asegurar que todos sus aditamentos (*plug-ins*), componentes de tiempo de ejecución y sus dependencias son compatibles entre sí.

Eclipse (58) es una herramienta que permite integrar diferentes aplicaciones lo que lo convierte en uno de los entornos integrados de desarrollo más utilizados en la actualidad, es de código abierto y multiplataforma; es una potente y completa plataforma de programación, desarrollo y compilación. La arquitectura de *plugins* de Eclipse permite, además de integrar diversos lenguajes sobre un mismo *IDE*, introducir otras aplicaciones secundarias que pueden resultar útiles durante el proceso de desarrollo como: herramientas *UML*, editores visuales de interfaces, ayuda en línea para librerías, entre otros. La plataforma Eclipse, al ser combinada con *Java Development Tools*, ofrece muchas de las características que se espera de un *IDE*. Posee un editor de sintaxis, la compilación de código incremental, un depurador de nivel fuente tipo *thread-aware*, un navegador de clases, y las interfaces con los sistemas estándar de control de código fuente, como los sistemas de versiones concurrentes.

Para dar solución a la problemática planteada y asumiendo las tecnologías establecidas en el CESIM se utilizará como entorno integrado de desarrollo Eclipse. Además provee funcionalidades en uno de sus aditamentos al Canal de Servicios Empresariales de la *suite WSO2* así como al trabajo con los servicios web *JAX-WS*. De esta manera se reducen los costos de tiempos al personalizar el trabajo sobre el mismo.

1.5.5 Herramientas de modelado

Las herramientas de modelado de sistemas se utilizan para la visualización en diagramas los productos que se van a desarrollar. Entre dichas herramientas se destacan *Enterprise Architect* (59), la misma facilita la realización de las etapas del ciclo de vida de un programa informático tales como análisis, diseño, pruebas y mantenimiento. *Rational Rose* (60) es otra de las herramientas destinadas al modelado de sistemas, con su utilización se puede generar código en diferentes lenguajes de programación (*Java*, *C++* y *C#*) desde un diagrama realizado. También brinda la posibilidad de generar documentación referente al sistema informático, para así obtener una ayuda en cuanto a la implementación del mismo. Además, al obtener el código fuente de un producto informático es posible adquirir los diagramas mediante la ingeniería inversa.

Visual Paradigm (61) es una herramienta de Ingeniería de Software Asistida por Computadora (*CASE - Computer Aided Software Engineering*) que soporta el Lenguaje Unificado de Modelado (*UML - Unified Modeling Language*) y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un sistema. Las ventajas que proporciona *Visual Paradigm* por *UML* son:

- Dibujo: Facilita el modelado de *UML*, puesto que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, porque la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica: Evita errores de sintaxis en el modelado.
- Coherencia entre diagramas: Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas para evitar duplicidades.
- Integración con otras aplicaciones: Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.

La herramienta *Visual Paradigm* por *UML* es la que más se ajusta al perfil de trabajo para darle solución al problema planteado por las ventajas antes mencionadas, además de ser de libre distribución por lo que los elementos como costo de desarrollo y mantenimiento pueden ser reducidos.

1.5.6 Herramientas para realizar pruebas de rendimiento

Con el objetivo de liberar un sistema se le deben realizar pruebas para comprobar su correcto funcionamiento. Para esto existen herramientas que se encargan de monitorear diferentes estados y

configuraciones del producto a examinar, de esta manera se pueden arrojar resultados con los que el cliente puede determinar si cumple con los resultados esperados.

Para medir el rendimiento de los servicios se escoge la es la herramienta de Supervisión de Actividades Empresariales de *WSO2* (por sus siglas en inglés *BAM*) (54) usada para el monitoreo de servicios, servidores y aplicaciones web. A través de su arquitectura permite que los elementos a monitorizar puedan conectarse a ella y enviar los datos requeridos los cuales son procesados y mostrados en un conjunto de graficas que conforman un *dashboard* sumamente útil para comprender el uso que pueden tener las distintas herramientas bajo monitorización.

La herramienta *Jmeter* (62) se escoge para comprobar el correcto funcionamiento del clúster. La misma permite realizar pruebas de carga para analizar y medir el desempeño de las aplicaciones web. Dicha herramienta posibilita verificar el rendimiento en recursos estáticos y dinámicos como pudieran ser archivos y bases de datos bajo diferentes tipos de carga concurrentemente, con la posibilidad de obtener una conclusión más acertada. También muestra las operaciones fallidas durante el proceso de pruebas. Además, posee un funcionamiento de hilos que permite realizar varias operaciones al mismo tiempo, lo que brinda diferentes estadísticas de la prueba que se realiza.

1.6 Plataforma empresarial de Java

La plataforma empresarial de *Java v6* (por sus siglas en inglés *JEE*) (63) permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Su lenguaje de programación *Java* es de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo. Esta plataforma cuenta además con una interfaz de aplicación para la persistencia y acceso a los datos.

1.7 Servicios web. Arquitectura orientada a servicios

La *W3C* define servicio web (64) como un sistema de software diseñado para permitir interoperabilidad máquina a máquina en una red. Los servicios web son sólo *APIs* Web que pueden ser accedidas en una red, como internet, y ejecutadas en un sistema de *hosting* remoto. La descripción de los servicios web se realiza a través de los mensajes que se intercambian entre el proveedor del servicio y el cliente utilizando el Lenguaje de Descripción de Servicios Web (por sus siglas en inglés *WSDL*) (65).

La Arquitectura Orientada a Servicios (por sus siglas en inglés SOA) (66) es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos. Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio las cuales incluyen facilidad y flexibilidad de integración con sistemas legados, alineación directa a los procesos de negocio reduciendo costos de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad. Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

1.8 Conclusiones parciales

Los componentes de los perfiles *XDS* y *PIX* se adecúan de manera correcta al objetivo que se persigue, pues los actores definidos lo constituyen las aplicaciones desarrolladas por el CESIM mientras que el repositorio y los metadatos correspondiente a los documentos clínicos se sustentan en las herramientas *WSO2* identificadas.

Las herramientas que resultaron ser las más adecuadas para desarrollar la solución son Eclipse en su versión 4.2.1 como entorno de desarrollo, *WSO2 AS* como servidor de aplicaciones en su versión 5.2.1 y para la capa intermedia el canal comunicaciones empresarial *WSO2 ESB* en su versión 4.7.0.

CAPÍTULO 2. INFRAESTRUCTURA DE COMUNICACIÓN.

En el presente capítulo se describe la propuesta de solución que consiste en una arquitectura basada en servicios web para la implementación de los perfiles *PIX* y *XDS* pertenecientes al dominio de Infraestructura TI, los cuales son necesarios para compartir documentos clínicos entre los sistemas sanitarios desarrollados por el CESIM. Se describen los actores implicados en los distintos escenarios, la forma en que se intercambiará la información en las transacciones entre dichos actores y la manera en que se emplearán las tecnologías seleccionadas en el capítulo 1.

2.1 Componentes esenciales de la infraestructura de comunicaciones propuesta.

La infraestructura de comunicaciones propuesta se compone de tres partes fundamentales:

- Los sistemas del CESIM: actúan como fuente y consumidor (actores principales) tanto en los procesos de identificación de pacientes como en los relacionados con los documentos clínicos. Los sistemas identificados para ejercer el rol de actor son: el Sistema de Gestión Hospitalaria y el Sistema de Información Radiológica debido las potencialidades de comunicación que ambos sistemas presentan.
- Para la identificación de pacientes se tiene un administrador de referencias cruzadas de identificación de paciente que implementa el perfil *PIX*, ubicado físicamente en un servidor de aplicaciones.
- Con el objetivo de almacenar y gestionar los documentos clínicos y sus metadatos se creó un repositorio de documentos clínicos así como un registro de metadatos (ambos pueden ejercer el rol de actores), siguiendo las especificaciones que establece el perfil *XDS*, ambos ubicados físicamente en un servidor de aplicaciones cada uno.

La comunicación entre los distintos actores se realiza a través del canal de comunicaciones *WSO2 ESB*. La interacción entre los componentes que conforman la infraestructura de comunicación se hace de manera estandarizada y siguiendo las políticas definidas por *SOA* (67). La *Figura 2.1* muestra la interrelación de cada uno de los componentes antes descritos.

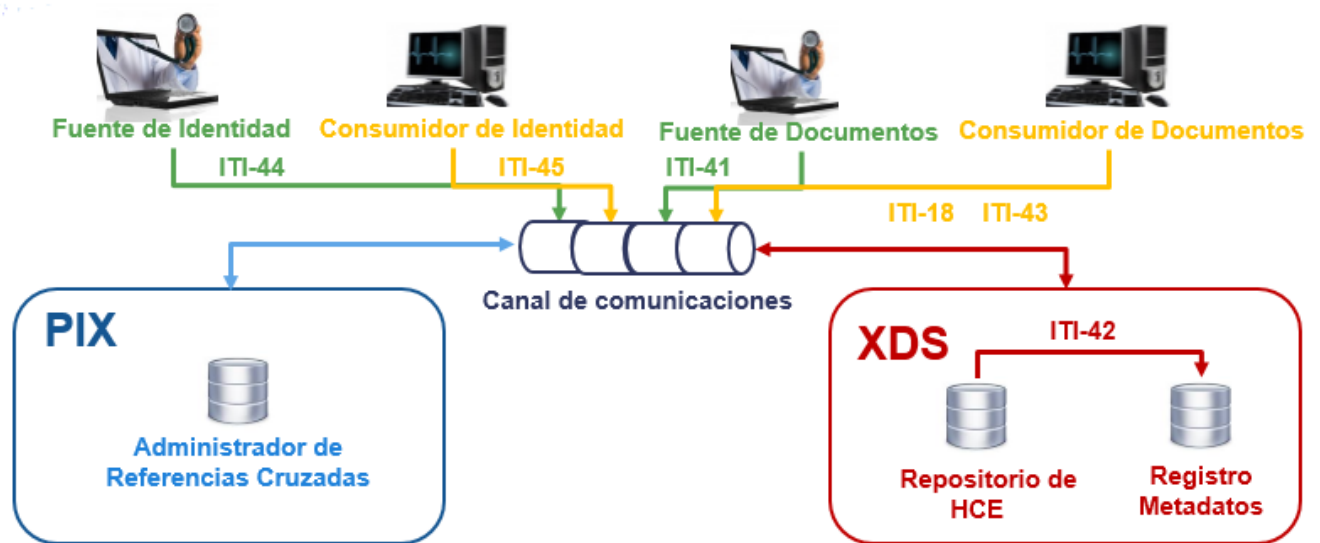


Figura 2.1 Propuesta de infraestructura de comunicaciones del CESIM

Fuente: Elaboración propia

2.2 Descripción de los actores presentes en la infraestructura de comunicación.

A continuación se describe detalladamente la función de cada uno de los actores mencionados en la sección 2.1, así como la interacción entre ellos a través de los mensajes enviados en las transacciones invocadas en cada acción de comunicación reflejada en la propuesta de infraestructura de comunicación.

2.2.1 Fuente de identificación de pacientes

La fuente de identificación de pacientes es el encargado de notificar al Administrador de Referencias Cruzadas de Identificación Pacientes y al Registro de Documentos de cualquier acción de identificación relacionado con el paciente, incluyendo la creación, modificación o fusión. Los sistemas realizan esta notificación a través de la transacción *Patient Identity Feed (ITI-44)* (31) la cual incluye una corroboración de los datos demográficos después de establecer la identidad de un paciente. La transacción *Patient Identity Feed* incluye dos escenarios de ejecución los cuales son descritos a continuación para una mejor comprensión:

Transacción *Patient Identity Feed*: Escenario 1

Este escenario comienza al ocurrir algunos de los siguientes eventos en uno de los actores:

- Admisión de un paciente hospitalizado en un centro hospitalario.

- El registro de un ambulatorio para una visita de las instalaciones.
- Pre-admisión de un paciente (es decir, el registro de la información del paciente antes del ingreso real).
- Actualización de Información del paciente.

El Administrador de Referencias Cruzadas de Identificación de Pacientes sólo deberá realizar la lógica de referencias cruzadas en los mensajes recibidos de la Fuente de Identificación de Pacientes. Para un Dominio de Identificación de Pacientes dado habrá una y sólo una Fuente de Identificación de Pacientes.

La transacción *Patient Identity Feed* se lleva a cabo por el mensaje *HL7 PRPA_IN201301UV02*, cada uno de estos mensajes son reconocidos por el mensaje *HL7 MCCI_IN000002UV01* enviado por el receptor del mensaje a su remitente. A continuación se describe la sintaxis del mensaje *PRPA_IN201301UV02*.

```

5      <id root="22a0f9e0-4454-11dc-a6be-3603d6866807"/>
6      <creationTime value="20070803130624"/>
7      <interactionId root="2.16.840.1.113883.1.6" extension="PRPA_IN201301UV02"/>
8      <processingCode code="P"/>
9      <processingModeCode code="R"/>
10     <acceptAckCode code="AL"/>
11     <receiver typeCode="RCV">
12         <device classCode="DEV" determinerCode="INSTANCE">
13             <id root="1.2.840.114350.1.13.99999.4567"/>
14             <telecom value="https://example.org/PatientFeed"/>
15         </device>
16     </receiver>
17     <sender typeCode="SND">
18         <device classCode="DEV" determinerCode="INSTANCE">
19             <id root="1.2.840.114350.1.13.99998.8734"/>
20         </device>
21     </sender>

```

Figura 2.2 Segmento común del mensaje *PRPA_IN201301UV02*

Fuente: Elaboración propia

En la *Figura 2.2* se muestra un segmento del mensaje enviado por la fuente de identificación de paciente *PRPA_IN201301UV02*. Este segmento es común para todos los mensajes (*PRPA_IN201301UV02*, *PRPA_IN201304UV02*, *PRPA_IN201309UV02*, *PRPA_IN201310UV02*) enviados durante la comunicación entre la fuente y el consumidor de Identificación de Paciente con el Administrador de Referencias Cruzadas de Identificación de Pacientes que son descritos en las siguientes secciones del capítulo.

También el segmento anteriormente mencionado contiene la información general del mensaje como: el identificador, la fecha de envío, el identificador del tipo de mensaje que se envía, para este caso “PRPA_IN201301UV02”, entre otros datos. Además se incluyen los segmentos *receiver* y *sender*, los cuales contienen el identificador de a quién y desde donde se envía el mensaje. A continuación se aborda otro de los segmentos correspondientes al mensaje PRPA_IN201301UV02.

Segmento *controlActProcess*

Otro segmento incluido en el mensaje PRPA_IN201301UV02 es el *controlActProcess*, el cual contiene los datos del evento realizado y es enviado por la fuente de identificación de pacientes. La Figura 2.3 muestra las partes del segmento *controlActProcess* correspondiente al registro de un paciente, lo cual es especificado en el atributo *classCode* de la etiqueta *registrationEvent*. La etiqueta *statusCode* representa el estado de los datos del paciente en el sistema. El segmento *custodian* contiene la información de la institución a la que fue asignado el paciente así como el estado de asignación del mismo.

```
<controlActProcess classCode="CACT" moodCode="EVN">
  <subject typeCode="SUBJ">
    <registrationEvent classCode="REG" moodCode="EVN">
      <id nullFlavor="NA"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <custodian typeCode="CST">
          <assignedEntity classCode="ASSIGNED">
            <id root="1.2.840.114350.1.13.99998.8734"/>
            <assignedOrganization classCode="ORG" determinerCode="INSTANCE">
              <name>Good Health Clinic</name>
            </assignedOrganization>
          </assignedEntity>
        </custodian>
      </registrationEvent>
    </subject>
  </controlActProcess>
```

Figura 2.3 Segmento *controlActProcess* del mensaje PRPA_IN201301UV02

Fuente: Elaboración propia

La etiquetada *Subject1* contiene los datos demográficos enviados por la fuente de identificación del paciente registrado, modificado o fusionado y posteriormente almacenado en el Administrador de Referencias Cruzadas de Identificación de Pacientes. La Figura 2.4 muestra el modelo de datos utilizado.

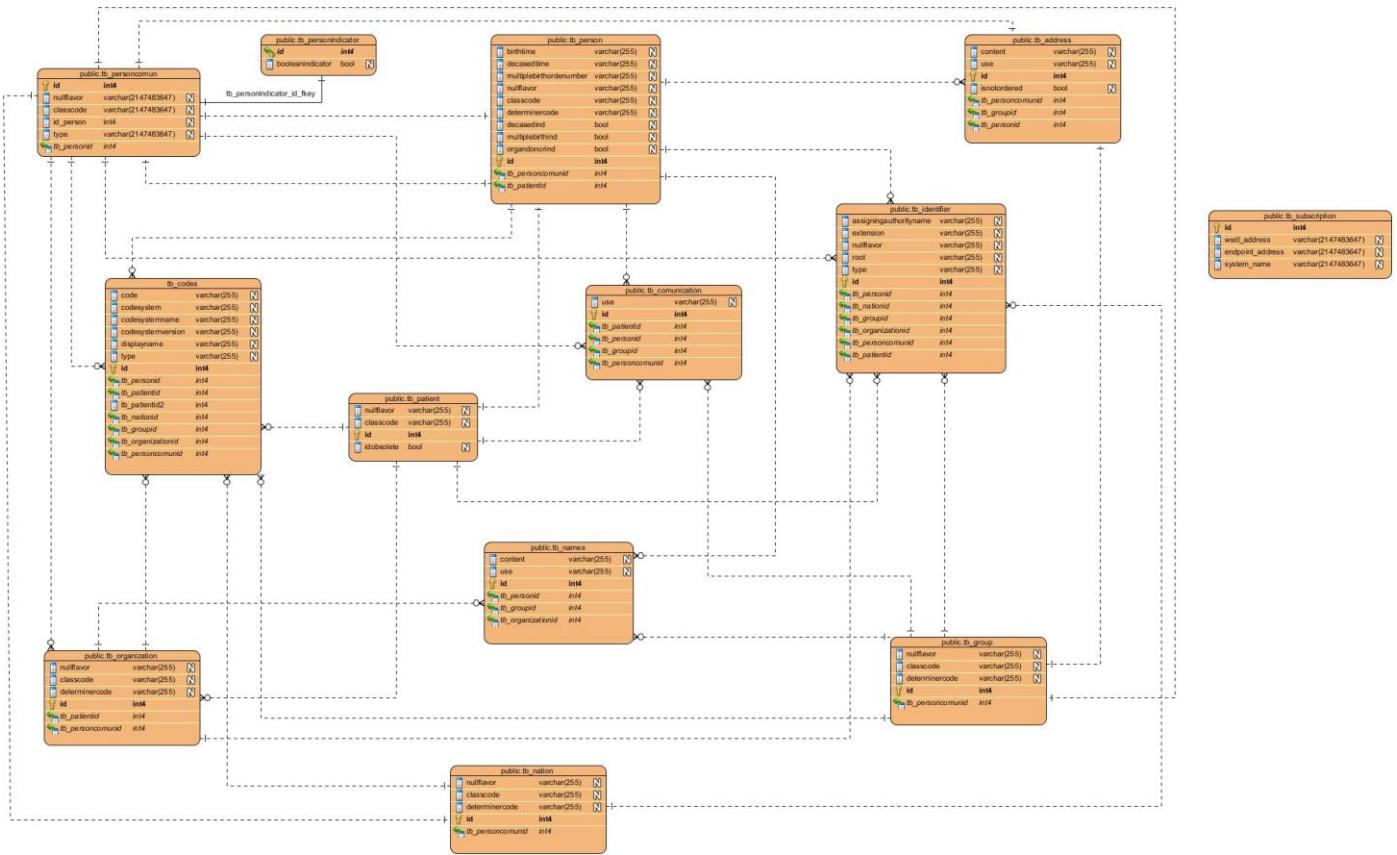


Figura 2.4 Modelo de datos de la identificación de pacientes

Fuente: Elaboración propia

La descripción de las tablas se encuentra en el artefacto Modelo de datos ubicado en el expediente de proyecto del Departamento Gestión Hospitalaria perteneciente al CESIM.

Transacción *Patient Identity Feed*: Escenario 2

En este escenario la Fuente de Identificación de Pacientes inicia la transacción cuando se fusionan los datos de un paciente determinado, la información es enviada mediante el mensaje *PRPA_IN201304UV02*. A diferencia del mensaje *PRPA_IN201301UV02*, se incluye un segmento *replacementOf* en la etiqueta *registrationEvent* como se muestra en la siguiente figura:

```

<replacementOf typeCode="RPLC">
  <priorRegistration classCode="REG" moodCode="EVN">
    <subject1 typeCode="SBJ">
      <priorRegisteredRole classCode="PAT" >
        <id root="1.2.840.114350.1.13.99998.8734" extension="34827G234"/>
      </priorRegisteredRole>
    </subject1>
  </priorRegistration>
</replacementOf>
    
```

Figura 2.5 Segmento *replacementOf* del mensaje *PRPA_IN201301UV02*

Fuente: Elaboración propia

El segmento *replacementOf* del mensaje *PRPA_IN201301UV02* es donde la fuente de identificación de pacientes especifica el identificador del paciente que se fusiona. La sección limitada por la etiqueta *priorRegistration* especifica el último evento realizado con el paciente, ya sea de registro o modificación de datos. La etiqueta *priorRegisteredRole* contiene el identificador que poseía el paciente en la última actualización que tuvo en el sistema.

2.2.2 Consumidor de identificación de paciente

El Consumidor de Identificación de Referencias Cruzadas de Pacientes es quien consulta al Administrador de Identificación de Referencias Cruzadas de Pacientes para obtener una lista de identificadores de un paciente determinado mediante la transacción *PIX Query (ITI-45)* (31) en la cual se envía el identificador del paciente para obtener los datos del mismo.

Esta transacción se lleva a cabo por el mensaje *HL7 PRPA_IN201309UV02*, el cual será reconocido por el mensaje *HL7 PRPA_IN201310UV02* enviado por el Administrador de Identificación de Referencias Cruzadas de Pacientes con la lista de identificadores. A continuación se describen los segmentos pertenecientes al mensaje *PRPA_IN201309UV02*, el mensaje *PRPA_IN201310UV02* es descrito en la sección 2.2.3.

Cómo es mencionado en la sección 2.2.1, el mensaje *PRPA_IN201309UV02* posee la misma estructura inicial que resto de los mensajes descritos en el capítulo. La *Figura 2.2* muestra el segmento *controlActProcess* perteneciente a dicho mensaje.

```
<controlActProcess classCode="CACT" moodCode="EVN">
  <code code="PRPA_TE201309UV02" codeSystem="2.16.840.1.113883.1.6"/>
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id root="1.2.840.114350.1.13.99997.2.7766" extension="USR5568"/>
    </assignedPerson>
  </authorOrPerformer>
  <queryByParameter>
    <queryId root="1.2.840.114350.1.13.99999.4567.34" extension="33452"/>
    <statusCode code="new"/>
    <responsePriorityCode code="I"/>
    <parameterList>
      <patientIdentifier>
        <value root="1.2.840.114350.1.13.99997.2.3412" extension="38273N237"/>
        <semanticsText>Patient.Id</semanticsText>
      </patientIdentifier>
    </parameterList>
  </queryByParameter>
</controlActProcess>
```

Figura 2.6 Segmento *controlActProcess* del mensaje *PRPA_IN201309UV02*

Fuente: Elaboración propia

En el cuerpo del segmento *controlActProcess* se encuentra la información de la consulta que desea realizar el Consumidor de Identificación de Pacientes, en la etiqueta *authorOrPerformer* se encuentra el identificador del usuario que realiza la consulta. En el segmento *queryByParameter* se definen los datos específicos de la consulta, el identificador único de la consulta, la prioridad de respuesta, así como el identificador que tiene el paciente en su sistema para realizar la búsqueda del resto de los identificadores en el dominio de integración.

2.2.3 Administrador de referencias cruzadas de identificación de paciente

El Administrador de Identificación de Referencias Cruzadas almacena los identificadores de los pacientes de los diferentes sistemas del CESIM, así como los datos demográficos necesarios para realizar el cruce de referencias como es mostrado en la *Figura 2.4*. Este actor responde a los mensajes *PRPA_IN201301UV02* y *PRPA_IN201309UV02* con los mensajes *MCCI_IN000002UV01* y *PRPA_IN201310UV02* respectivamente. A continuación se describe la estructura de cada mensaje cuya estructura inicial es igual a la descrita en los epígrafes anteriores del presente capítulo. La *Figura 2.6* muestra los datos que se le envían a la Fuente de Identificación de Pacientes con el código de aceptación y el identificador del mensaje de notificación de registro, modificación o fusión de pacientes.

```
<acknowledgement>
  <typeCode code="CA"/>
  <targetMessage>
    <id root="22a0f9e0-4454-11dc-a6be-3603d6866807"/>
  </targetMessage>
</acknowledgement>
```

Figura 2.7 Segmento *acknowledgement* del mensaje *MCCI_IN000002UV01*

Fuente: Elaboración propia

El segmento *controlActProcess* del mensaje *PRPA_IN201310UV02* contiene la información referente los identificadores del paciente que se solicitaron y son definido en la etiqueta *subject* como se muestra en la *Figura 2.8*.

```
<controlActProcess classCode="CACT" moodCode="EVN">
  <code code="PRPA_TE201310UV02"/>
  <subject typeCode="SUBJ">
  </subject>
  <queryAck>
  </queryAck>
  <queryByParameter>
  </queryByParameter>
</controlActProcess>
```

Figura 2.8 Segmento *controlActProcess* del mensaje *PRPA_IN201310UV02*

Fuente: Elaboración propia

También se enviará el estado de respuesta a la consulta (*Figura 2.11*), además de los parámetros recibidos en el mensaje *PRPA_IN201309UV02* como se describe en la *Figura 2.6*.

El segmento *patient* que se muestra en la *Figura 2.8* contiene los datos del paciente solicitado, así como de la institución en la que se registró el mismo. En este mensaje no es necesaria la devolución de los datos demográficos del paciente, por lo que la etiqueta *patientPerson* solo contendrá la lista de identificadores que tiene el paciente en el dominio de integración como es mostrado en la *Figura 2.9*.

```
<patient classCode="PAT">
  <id root="1.2.840.114350.1.13.99998.8734" extension="34827G409"/>
  <statusCode code="active"/>
  <patientPerson classCode="PSN" determinerCode="INSTANCE">
    <name nullFlavor="NA">
    </name>
    <asOtherIDs classCode="PAT">
      <id root="1.2.840.114350.1.13.99997.2.3412" extension="38273N237"/>
      <scopingOrganization classCode="ORG" determinerCode="INSTANCE">
        <id root="1.2.840.114350.1.13.99997.2.3412"/>
      </scopingOrganization>
    </asOtherIDs>
    <asOtherIDs classCode="CIT">
      <id root="2.16.840.1.113883.4.1" extension="999-99-4452"/>
      <scopingOrganization classCode="ORG" determinerCode="INSTANCE">
        <id root="2.16.840.1.113883.4.1"/>
      </scopingOrganization>
    </asOtherIDs>
  </patientPerson>
  <providerOrganization classCode="ORG" determinerCode="INSTANCE">
    <id root="1.2.840.114350.1.13.99998.8734"/>
    <name>Good Health Clinic</name>
    <contactParty classCode="CON">
      <telecom value="tel:+1-342-555-8394"/></telecom>
    </contactParty>
  </providerOrganization>
</patient>
```

Figura 2.9 Segmento *patient* del mensaje *PRPA_IN201310UV02*

Fuente: Elaboración propia

La *Figura 2.10* muestra la etiqueta *asOtherIDs*, la cual contiene el identificador del paciente en una institución determinada en las etiquetas *id* y *scopingOrganization* respectivamente, el modelo de datos correspondiente a estos elementos se describe en la *Figura 2.4*.

```
<asOtherIDs classCode="PAT">
  <id root="1.2.840.114350.1.13.99997.2.3412" extension="38273N237"/>
  <scopingOrganization classCode="ORG" determinerCode="INSTANCE">
    <id root="1.2.840.114350.1.13.99997.2.3412"/>
  </scopingOrganization>
</asOtherIDs>
```

Figura 2.10 Segmento *asOtherIDs* del mensaje *PRPA_IN201310UV02*

Fuente: Elaboración propia

El segmento *queryAck* mostrado en la *Figura 2.11* notifica al Consumidor de Identificación de Pacientes el estado de respuesta de la consulta, enviando el identificador de la consulta enviada en el mensaje *PRPA_IN201310UV02* y el código de respuesta a la misma.

```
<queryAck>
  <queryId root="1.2.840.114350.1.13.99999.4567.34" extension="33452"/>
  <queryResponseCode code="OK"/>
</queryAck>
<queryByParameter>
```

Figura 2.11 Segmento *asOtherIDs* del mensaje *PRPA_IN201310UV02*

Fuente: Elaboración propia.

2.2.4 Fuente de Documentos

La fuente de Documentos (*Document Source*) intercambia información con el Repositorio de Documentos (*Document Repository*) mediante la transacción *Provide And Register Document Set (ITI-41)* con el fin de enviar un documento y sus metadatos para que sea registrado. El mensaje de petición que envía la fuente de Documentos está representado por la etiqueta *ProvideAndRegisterDocumentSetRequest* que contiene una etiqueta *Document* que representa la información del documento en una secuencia de *bytes* y otra *SubmitObjectsRequest* que contiene una *RegistryObjectList* dentro de la cual se especifican todos los metadatos de los documentos. A continuación se muestra el ejemplo de un mensaje:

```
<ProvideAndRegisterDocumentSetRequest>
  <lcm:SubmitObjectsRequest>
    <rim:RegistryObjectList>
      <rim:ExtrinsicObject id="Document01" mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
        <rim:RegistryPackage id="SubmissionSet01">
          <rim:Classification id="cl10" classifiedObject="SubmissionSet01" classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"/>
          <rim:Association id="as01" associationType="HasMember" sourceObject="SubmissionSet01" targetObject="Document01">
        </rim:RegistryObjectList>
    </lcm:SubmitObjectsRequest>
    <Document id="Document01">UjBsR09EbGhjZ0dTQXNlUjBUNBRU1tQ1p0dU1GUxhBUzhi</Document>
  </ProvideAndRegisterDocumentSetRequest>
```

Figura 2.12 Mensaje de petición. Transacción *Provide And RegisterDocument Set*.

Fuente: Elaboración propia

Cuando el Repositorio de Documentos recibe un mensaje de este tipo, se almacena el documento físico con la información contenida dentro de la etiqueta *Document* y envía la información correspondiente a los metadatos hacia el registro de documentos mediante la transacción *RegistrarDocumentSet* (Ver sección 2.2.2 Repositorio de Documentos). Al finalizar el proceso, el Repositorio de Documentos envía un mensaje informando el estado de la transacción que fue iniciada por la fuente de Documentos (Ver sección 2.2.2 Repositorio de Documentos).

2.2.5 Repositorio de Documentos

El Repositorio de Documentos intercambia información con la Fuente de Documentos mediante la transacción *ProvideAndRegisterDocumentSet* y con el Registro de Documentos mediante la transacción *RegisterDocumentSet (ITI-42)*. Cuando el repositorio recibe una petición proveniente del actor fuente de documentos con el fin de registrar una colección de documentos, el repositorio almacena físicamente el documento y envía un mensaje de petición al actor registro de documentos para el almacenamiento de los metadatos de los documentos que recibió de la fuente. Este mensaje de petición se define en la transacción *RegisterDocumentSet* y tiene la siguiente estructura:

```

<lcm:SubmitObjectsRequest>
  <rim:RegistryObjectList>
    <rim:ExtrinsicObject id="Document01" mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
    <rim:RegistryPackage id="SubmissionSet01">
      <rim:Classification id="c110" classifiedObject="SubmissionSet01" classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"/>
      <rim:Association id="as01" associationType="HasMember" sourceObject="SubmissionSet01" targetObject="Document01">
    </rim:RegistryObjectList>
  </lcm:SubmitObjectsRequest>

```

Figura 2.13 Mensaje de petición. Transacción *RegisterDocumentSet*.

Fuente: Elaboración propia

En la imagen anterior se muestra dentro de la etiqueta *RegistryObjectList* la lista de los metadatos los cuales son especificados usando las etiquetas *ExtrinsicObject*, *RegistryPackage*, *Association*, *ObjectRef* (Ver sección 2.2.4 Registro de Documentos).

Después de realizar la petición antes descrita al registro, el repositorio esperará la respuesta la cual contiene el estado de la transacción. Adicionalmente, si el registro encontró errores el mensaje se forma por la etiqueta *RegistryResponse* especificando el valor *urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failed* para el atributo *status* y dentro de esta una colección de etiquetas *RegistryError* con cada uno de los errores que se encontraron como se muestra en la *Figura 2.14*. En caso contrario adquiere el valor *urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success*.

```

1 <RegistryResponse status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure">
2   <RegistryErrorList highestSeverity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error">
3     <RegistryError errorCode="XDSPatientIdDoesNotMatch"
4       codeContext="Patient Id in Document (Document1) does not match SubmissionSet"
5       location=""
6       severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
7     <RegistryError errorCode="XDSRegistryMetadataError"
8       codeContext="RegistryPackage (SubmissionSet) is not labeled as SubmissionSet or Folder"
9       location=""
10      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
11   </RegistryErrorList>
12 </RegistryResponse>

```

Figura 2.14 Mensaje de respuesta. *Provide And RegisterDocument Set*.

Fuente: Elaboración propia

Cada etiqueta *RegistryError* está compuesta por cuatro atributos específicos (68)

- *errorCode*: Indica el tipo de error que se produjo. Estos errores se pueden encontrar en la *Tabla 4.2.4.1 del libro IHE_ITI_TF_Vol3, página 84*. Este campo es obligatorio.
- *codeContext*: Una descripción con más detalles del error que se produjo. Este campo es obligatorio.
- *location*: Indica el módulo, línea o lugar donde se produjo el error. Este campo es opcional.
- *severity*: Indica el tipo es error o precaución. Este campo es obligatorio.

El mensaje anterior es el mismo mensaje de respuesta enviado a la fuente de documentos por el registro, y de manera adicional los errores encontrados. El repositorio de documentos recibe la petición del consumidor de documentos para recuperar documentos que son almacenados previamente. Estas peticiones contendrán el identificador del documento que se desea recuperar (Ver *sección 2.2.3 Consumidor de Documentos*). A estas peticiones el repositorio responderá con un mensaje que tiene la siguiente estructura:

```

<RetrieveDocumentSetResponse>
  <rs:RegistryResponse status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"/>
  <DocumentResponse>
    <RepositoryUniqueId>1.3.6.1.4...1000</RepositoryUniqueId>
    <DocumentUniqueId>1.3.6.1.4...2300</DocumentUniqueId>
    <mimeType>text/xml</mimeType>
    <Document>UjBsR09EbGhjz0dTQxNQUBUUNBRU1tQ1p0dU1GUzhEUVzhi</Document>
  </DocumentResponse>
  <DocumentResponse>
    <RepositoryUniqueId>1.3.6.1.4...1000</RepositoryUniqueId>
    <DocumentUniqueId>1.3.6.1.4...2300</DocumentUniqueId>
    <mimeType>text/xml</mimeType>
    <Document>UjBsR09EbGhjz0dTQxNQUBUUNBRU1tQ1p0dU1GUzhEUVzhi</Document>
  </DocumentResponse>
</RetrieveDocumentSetResponse>

```

Figura 2.15 Mensaje de respuesta. *RetrieveDocumentSet*.

Fuente: Elaboración propia

Como se puede apreciar en la figura anterior el mensaje de respuesta contiene una definición *RetrieveDocumentSetResponse* (ITI-43) que representa el mensaje de respuesta a la transacción *RetrieveDocumentSet*. Esta etiqueta contiene las etiquetas *RegistryResponse* que permite saber el estado de la transacción y una colección de etiquetas *DocumentResponse* que representan los

documentos que forman parte de la respuesta a la petición para la recuperación de estos documentos. Cada definición *DocumentResponse* contiene las cuatro etiquetas siguientes:

- *RepositoryUniqueId*: representa el identificador del repositorio del cual se recupera el documento.
- *DocumentUniqueId*: representa el campo *uniqueId* del documento que se recupera.
- *MimeType*: representa el tipo de archivo o extensión del archivo que se recupera.
- *Document*: representa la información en una secuencia de *bytes* del archivo que se recupera.

2.2.6 Consumidor de Documentos

El Consumidor de Documentos (*Document Consumer*) intercambia información mediante las transacciones *RegistryStoredQuery* (ITI-18) y *RetrieveDocumentSet* con el fin de recuperar los documentos clínicos que están almacenados en el repositorio. Además se comunica con el Registro de Documentos mediante la transacción *RegistryStoredQuery* donde se especifica cuál de las 14 posibles consulta se desea ejecutar así como los parámetros necesarios para realizar la búsqueda de los documentos. A continuación se muestra la tabla donde se relacionan las consultas con el identificador de cada una.

Tabla 2.1 Identificador de las consultas a ejecutar en el Registro de Documentos
Fuente: IHE_ITI_TF_Vol2a (31)

Nombre de la Consulta	Identificador de la Consulta
FindDocuments	urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d
FindSubmissionSets	urn:uuid:f26abbc-b-ac74-4422-8a30-edb644bbc1a9
FindFolders	urn:uuid:958f3006-baad-4929-a4deff1114824431
GetAll	urn:uuid:10b545ea-725c-446d-9b95-8aeb444eddf3
GetDocuments	urn:uuid:5c4f972b-d56b-40ac-a5fcc8ca9b40b9d4
GetFolders	urn:uuid:5737b14c-8a1a-4539-b659-e03a34a5e1e4
GetAssociations	urn:uuid:a7ae438b-4bc2-4642-93e9-be891f7bb155
GetDocumentsAndAssociations	urn:uuid:bab9529a-4a10-40b3-a01ff68a615d247a
GetSubmissionSets	urn:uuid:51224314-5390-4169-9b91-b1980040715a
GetSubmissionSetAndContents	urn:uuid:e8e3cb2c-e39c-46b9-99e4-c12f57260b83
GetFolderAndContents	urn:uuid:b909a503-523d-4517-8acf-8e5834dfc4c7
GetFoldersForDocument	urn:uuid:10cae35a-c7f9-4cf5-b61efc3278ffb578
GetRelatedDocuments	urn:uuid:d90e5407-b356-4d91-a89f-873917b4b0e6
FindDocumentsByReferenceld	urn:uuid:12941a89-e02e-4be5-967cce4bfc8fe492

De esta manera si el Consumidor de Documentos desea ejecutar la consulta *FindDocuments* con identificador *urn: uuid: 14d4debf-8f97-4251-9a74-a90016b0af0d*, el mensaje que es enviado al Registro de Documentos tiene la siguiente estructura:

```
<rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">
</rim:AdhocQuery>
```

Figura 2.16 Porción del mensaje de petición de la transacción RegistryStoredQuery

Fuente: Elaboración propia

Cada consulta tiene un grupo de parámetros de especificación obligatorio por parte del consumidor de documentos y otro grupo opcionales. A continuación se muestra la sección de un mensaje donde se especifica el valor para el parámetro *\$XDSDocumentEntryPatientId*.

```
<rim:Slot name="$XDSDocumentEntryPatientId">
  <rim:ValueList>
    <rim:Value>st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

Figura 2.17 Porción del mensaje de petición de la transacción RegistryStoredQuery

Fuente: Elaboración propia

Estas consultas también soportan *AND* y *OR* semánticos sobre algunos de sus atributos. Para especificar un *OR* semántico dentro de una etiqueta *Slot* se especificaran más de un valor. El siguiente ejemplo devuelve los documentos que su atributo *EntryUUID* tiene valor 'a' o 'b'.

```
<rim:Slot name="$XDSDocumentEntryEntryUUID ">
  <rim:ValueList>
    <rim:Value>('a')</rim:Value>
    <rim:Value>('b')</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

Figura 2.18 OR semántico

Fuente: Elaboración propia

Para especificar un *AND* semántico se especificaran más de una etiqueta *Slot* con el mismo nombre. El siguiente ejemplo devuelve los documentos que su atributo *EntryUUID* tiene valor 'a' y 'b'.

```
2 <rim:Slot name="$XDSDocumentEntryEntryUUID">
3   <rim:ValueList>
4     <rim:Value>('a')</rim:Value>
5   </rim:ValueList>
6 </rim:Slot>
7 <rim:Slot name="$XDSDocumentEntryEntryUUID">
8   <rim:ValueList>
9     <rim:Value>('b')</rim:Value>
10  </rim:ValueList>
11 </rim:Slot>
```

Figura 2.19 AND semántico

Fuente: Elaboración propia

A continuación se muestran las tablas donde se presentan los parámetros específicos para cada consulta. Las tablas tiene tres columnas donde la primera corresponde al nombre del parámetro, la segunda

contendrá una “R” si el parámetro es obligatorio o una “O” si el parámetro es opcional y la tercera columna contendrá una “M” si el parámetro soporta valores múltiples.

Consulta *FindDocuments*

Esta consulta retorna una colección de objetos de tipo *XDSDocumentEntry* que coincidan con los parámetros especificados por el Consumidor de Documentos. La *Tabla 2.2* muestra en la primera columna los nombres de los parámetros que se le pueden pasar a la consulta. En la segunda se especifica si el parámetro es obligatorio (“R”) o si es opcional (“O”) y en la tercera columna se especifica con una letra “M” si el campo acepta valores múltiples.

Tabla 2.2 Parámetros de la consulta *FindDocuments*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryPatientId	R	–
\$XDSDocumentEntryClassCode	O	M
\$XDSDocumentEntryTypeCode	O	M
\$XDSDocumentEntryPracticeSettingCode	O	M
\$XDSDocumentEntryCreationTimeFrom	O	–
\$XDSDocumentEntryCreationTimeTo	O	–
\$XDSDocumentEntryServiceStartTimeFrom	O	–
\$XDSDocumentEntryServiceStartTimeTo	O	–
\$XDSDocumentEntryServiceStopTimeFrom	O	–
\$XDSDocumentEntryServiceStopTimeTo	O	–
\$XDSDocumentEntryHealthcareFacilityTypeCode	O	M
\$XDSDocumentEntryEventCodeList	O	M
\$XDSDocumentEntryConfidentialityCode	O	M
\$XDSDocumentEntryAuthorPerson	O	M
\$XDSDocumentEntryFormatCode	O	M
\$XDSDocumentEntryStatus	R	M

Consulta *FindSubmissionSets*

Esta consulta retorna una colección de objetos de tipo *XDSSubmissionSet* que coincidan con los parámetros especificados por el Consumidor de Documentos.

Tabla 2.3 Parámetros de la consulta *FindSubmissionSets*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSSubmissionSetPatientId	R	–
\$XDSSubmissionSetSourceId	O	M
\$XDSSubmissionSetSubmissionTimeFrom	O	–
\$XDSSubmissionSetSubmissionTimeTo	O	–
\$XDSSubmissionSetAuthorPerson	O	–
\$XDSSubmissionSetContentType	O	M
\$XDSSubmissionSetStatus	R	M

Consulta *FindFolders*

Esta consulta retorna una colección de objetos de tipo *XDSFolder* que coincidan con los parámetros especificados por el Consumidor de Documentos.

Tabla 2.4 Parámetros de la consulta *FindFolder*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSFolderPatientId	R	–
\$XDSFolderLastUpdateTimeFrom	O	–
\$XDSFolderLastUpdateTimeTo	O	–
\$XDSFolderCodeList	O	M
\$XDSFolderStatus	R	M

Consulta *GetAll*

Devuelve todos los contenidos que coincidan con los parámetros que fueron especificados en la petición. Los objetos retornados son de tipo *XDSSubmissionSet*, *XDSDocumentEntry*, *XDSFolder* y *Association*.

Tabla 2.5 Parámetros de la consulta *GetAll*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$patientId	R	--
\$XDSDocumentEntryStatus	R	M
\$XDSSubmissionSetStatus	R	M
\$XDSFolderStatus	R	M
\$XDSDocumentEntryFormatCode	O	M
\$XDSDocumentEntryConfidentialityCode	O	M

Consulta *GetDocuments*

Recupera una colección de objetos de tipo *XDSDocumentEntry*. Los objetos *XDSDocumentEntry* son seleccionados ya sea por su atributo *entryUUID* o *uniqueId*.

Tabla 2.6 Parámetros de la consulta *GetDocuments*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryEntryUUID	O	M
\$XDSDocumentEntryUniqueid	O	M
\$homeCommunityId	O	--

Consulta *GetFolders*

Recupera una colección de objetos de tipo *XDSFolder*. Los objetos *XDSFolder* son seleccionados por su *entryUUID* o *uniqueId*.

Tabla 2.7 Parámetros de la consulta GetFolders

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSFolderEntryUUID	O	M
\$XDSFolderUniqueid	O	M
\$homeCommunityId	O	--

Consulta GetAssociations

Recupera objetos *Association* cuyo atributo *sourceObject* o *targetObject* coincide con el parámetro *\$uuid* especificado por el actor *DocumentConsumer*.

Tabla 2.8 Parámetros de la consulta GetAssociations

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$uuid	R	M
\$homeCommunityId	O	-

Consulta GetDocumentsAndAssociations

Recupera una colección de objetos *XDSDocumentEntry* y *Association*. Los objetos *XDSDocumentEntry* son seleccionados por su atributo *entryUUID* o *uniqueid*. Esta son las consultas *GetDocuments* y *GetAssociations* combinadas en una consulta.

Tabla 2.9 Parámetros de la consulta GetDocumentosAndAssociations

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryEntryUUID	O	M
\$XDSDocumentEntryUniqueid	O	M
\$homeCommunityId	O	--

Consulta GetSubmissionSets

Recupera objetos *XDSSubmissionSet* usados para enviar colecciones de objetos *XDSDocumentEntry* y *XDSFolder*. Los objetos *XDSDocumentEntry* y *XDSFolder* de interés son identificados por su *UUIDs* en el parámetro *\$uuid*.

Tabla 2.10 Parámetros de la consulta GetSubmissionSets

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$uuid	R	M
\$homeCommunityId	O	--

Consulta GetSubmissionSetAndContents

Recupera un objeto *SubmissionSet* y su contenido. Los objetos *SubmissionSet* son seleccionados por su atributo *entryUUID* o *uniqueId*. Los objetos *DocumentEntry* pueden restringirse por su atributo *formatCode* y *confidentialityCode*

Tabla 2.11 Parámetros de la consulta *GetSubmissionSetAndContents*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSSubmissionSetEntryUUID	O	--
\$XDSSubmissionSetUniqueid	O	--
\$XDSDocumentEntryFormatCode	O	M
\$XDSDocumentEntryConfidentialityCode	O	M
\$homeCommunityId	O	--

Consulta *GetFolderAndContents*

Recupera un objeto *XDSFolder* y su contenido. El objeto *XDSFolder* es seleccionado por su atributo *entryUUID* o *uniqueId*. Los objetos *DocumentEntry* Los objetos devueltos pueden ser restringidos por sus atributos *formatCode* y *confidentialityCode*.

Tabla 2.12 Parámetros de la consulta *GetFolderAndContents*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSFolderEntryUUID	O	--
\$XDSFolderUniqueid	O	--
\$XDSDocumentEntryFormatCode	O	M
\$XDSDocumentEntryConfidentialityCode	O	M
\$homeCommunityId	O	--

Consulta *GetFoldersForDocument*

Recupera objetos *XDSFolder* que contienen el objeto *XSDSDocumentEntry* especificado en la consulta. El objeto *XSDSDocumentEntry* se especifica mediante su atributo *entryUUID* o *uniqueId*.

Tabla 2.13 Parámetros de la consulta *GetFolderForDocument*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryEntryUUID	O	--
\$XDSDocumentEntryUniqueid	O	--
\$homeCommunityId	O	--

Consulta *GetRelatedDocuments*

Recupera objetos *XSDSDocumentEntry* que están relacionados con documentos específicos mediante objetos de *Association*. También devuelve los objetos *Association*. El Consumidor de Documentos especifica el *UUID* o el *uniqueId* del documento para buscar todas sus relaciones.

Tabla 2.14 Parámetros de la consulta *GetRelatedDocuments*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryEntryUUID	O	--
\$XDSDocumentEntryUniquelid	O	--
\$AssociationType	R	M
\$homeCommunityId	O	--

Consulta *FindDocumentsByReferenceld*

Esta consulta retorna objetos *XDSDocumentEntry* que coincidan con los parámetros especificados por el actor Consumidor de documentos. Es semánticamente idéntica a la consulta *FindDocuments* excepto que el parámetro *\$XDSDocumentEntryReferenceldList* contiene uno o más valores para comparar con el atributo *referenceldList* de los *DocumentEntry* que están almacenados en el registro.

Tabla 2.15 Parámetros de la consulta *GetDocumentsByReferenceld*

Fuente: IHE_ITI_TF_Vol2a (31)

Nombre del parámetro	Opción	Mult
\$XDSDocumentEntryPatientId	R	--
\$XDSDocumentEntryReferenceldList	R	M
\$XDSDocumentEntryClassCode	O	M
\$XDSDocumentEntryTypeCode	O	M
\$XDSDocumentEntryPracticeSettingCode	O	M
\$XDSDocumentEntryCreationTimeFrom	O	--
\$XDSDocumentEntryCreationTimeTo	O	--
\$XDSDocumentEntryServiceStartTimeFrom	O	--
\$XDSDocumentEntryServiceStartTimeTo	O	--
\$XDSDocumentEntryServiceStopTimeFrom	O	--
\$XDSDocumentEntryServiceStopTimeTo	O	--
\$XDSDocumentEntryHealthcareFacilityTypeCode	O	M
\$XDSDocumentEntryEventCodeList	O	M
\$XDSDocumentEntryConfidentialityCode	O	M
\$XDSDocumentEntryAuthorPerson	O	M
\$XDSDocumentEntryFormatCode	O	M
\$XDSDocumentEntryStatus	R	M

Después de que el mensaje es recibido por el Registro de Documentos este ejecuta la consulta solicitada y le envía al Consumidor de Documentos las instancias que se encontraron (Ver *sección 2.2.2*). Con esta información el Consumidor de Documentos tiene la información de cada objeto encontrado incluyendo el identificador del repositorio donde se encuentra el documento físico y el identificador del mismo. Con estos datos el Consumidor de Documentos comienza la transacción *RetrieveDocumentSet* que le permite enviarle al registro el identificador del documento que desea recuperar. Este mensaje tiene la forma siguiente:

```

<RetrieveDocumentSetRequest>
  <DocumentRequest>
    <RepositoryUniqueId>1.3.6.1.4...1000</RepositoryUniqueId>
    <DocumentUniqueId>1.3.6.1.4...2300</DocumentUniqueId>
  </DocumentRequest>
  <DocumentRequest>
    <RepositoryUniqueId>1.3.6.1.4...1000</RepositoryUniqueId>
    <DocumentUniqueId>1.3.6.1.4...2301</DocumentUniqueId>
  </DocumentRequest>
</RetrieveDocumentSetRequest>

```

Figura 2.20 Mensaje de petición. Transacción *RetrieveDocumentSet*.

Fuente: Elaboración propia

En el ejemplo anterior el Consumidor de Documentos le solicita el documento con identificador “1.3.6.1.4...2300” al repositorio “1.3.6.1.4...1000” y el documento con identificador “1.3.6.1.4...2301” al mismo repositorio. Cuando el repositorio recibe esta petición responde con un mensaje (Ver sección 2.2.5 Repositorio de Documentos) que contiene el identificador del repositorio que se está consultando, el identificador del documento que se desea recuperar, el tipo de archivo y una secuencia de *byte* que representa la información del archivo.

2.2.7 Registro de Documentos

El Registro de Documentos (*Document Registry*) mediante las transacciones *RegistryStoredQuery* y *RegisterDocumentSet* responde a las peticiones del Consumidor de Documentos y del Repositorio de Documentos. Cuando el Registro de Documentos recibe una petición del Consumidor de Documentos para la ejecución de una consulta, los elementos que cumplen con los parámetros especificados son devueltos en un mensaje que tiene la siguiente estructura:


```

1 <query:AdhocQueryResponse status="Success">
2   <rim:RegistryObjectList>
3     <rim:ExtrinsicObject
4       id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
5       isOpaque="false"
6       mimeType="text/xml"
7       objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
8       status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"/>
9
10    <rim:ObjectRef id="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"/>
11    <rim:ObjectRef id="urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"/>
12    <rim:ObjectRef id="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"/>
13    <rim:ObjectRef id="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"/>
14    <rim:ObjectRef id="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"/>
15    <rim:ObjectRef id="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"/>
16    <rim:ObjectRef id="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"/>
17    <rim:ObjectRef id="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"/>
18  </rim:RegistryObjectList>
19 </query:AdhocQueryResponse>

```

Figura 2.21 Mensaje de respuesta. Transacción *RegistryStoredQuery*.

Fuente: Elaboración propia

En la porción del mensaje de respuesta anterior la etiqueta *AdhocQueryResponse* corresponde a la definición del mensaje de respuesta y la etiqueta *RegistryObjectList* corresponde a la lista de objetos encontrados en el registro. Dentro de esta última etiqueta podrán aparecer las siguientes etiquetas:

- *ExtrinsicObject*: representan un documento que fue encontrado, esta etiqueta incluye todos los metadatos del documento. A continuación se muestran en etiquetas *Slot* algunas de las características del documento.

```

<rim:ExtrinsicObject
  id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
  isOpaque="false"
  mimeType="text/xml"
  objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
  status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">
  <rim:Slot name="URI">
    <rim:ValueList>
      <rim:Value>http://localhost:8080/XDS/Repository/08a15a6f-5b4a-42de-8f95-89474f83abdf.xml</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="authorInstitution">
    <rim:ValueList>
      <rim:Value>Fairview Hospital</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="creationTime">
    <rim:ValueList>
      <rim:Value>200412261119</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="hash">
    <rim:ValueList>
      <rim:Value>4cf4f82d78b5e2aac35c31bca8cb79fe6bd6a41e</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="sourcePatientId">
    <rim:ValueList>
      <rim:Value>jd12323^^^wsh</rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:ExtrinsicObject>

```

Figura 2.22 Porción del mensaje que representa un documento.

Fuente: Elaboración propia

- *RegistryPackage*: representa una carpeta en la que se contiene uno o muchos documentos.

```

<rim:RegistryPackage id="SubmissionSet01">
  <rim:Slot name="submissionTime">
    <rim:ValueList>
      <rim:Value>20041225235050</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString value="Physical"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString value="Annual physical"/>
  </rim:Description>
  <rim:Classification id="c108" classificationScheme="urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d" classifiedObject="SubmissionSet01">
    <rim:Slot name="authorPerson">
      <rim:ValueList>
        <rim:Value>Sherry Dopplemeyer</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="authorInstitution">
      <rim:ValueList>
        <rim:Value>Cleveland Clinic</rim:Value>
        <rim:Value>Berea Community</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="authorRole">
      <rim:ValueList>
        <rim:Value>Primary Surgon</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:Classification>
</rim:RegistryPackage>

```

Figura 2.23 Porción del mensaje que representa una carpeta.

Fuente: Elaboración propia

- *Association*: representa una asociación entre elementos del registro. Esta asociación puede ser entre una carpeta y un archivo o entre dos archivos.
 - Carpeta y archivo.
 - ebRIM HasMember urn:oasis:names:tc:ebxmlregrep:
 - AssociationType:HasMember
 - Dos archivos: Si la asociación es entre dos archivos, esta podrá ser una de las siguientes:

Tabla 2.16 Tipos de asociaciones entre archivos

Fuente: IHE_ITI_TF_Vol3 (68)

HL7 Document Relationship Replace	urn:ihe:iti:2007:AssociationType:RPLC
HL7 Document Relationship Transform	urn:ihe:iti:2007:AssociationType:XFRM
HL7 Document Relationship Append	urn:ihe:iti:2007:AssociationType:APND
HL7 Document Relationship Transform and Replace	urn:ihe:iti:2007:AssociationType:XFRM_RPLC
IHE Signature	urn:ihe:iti:2007:AssociationType:signs

- *APND (append)*: El presente documento es una adición al documento padre.
- *RPLC (replace)*: El documento actual es un reemplazo del documento padre.
- *XFRM (transform)*: El presente documento es una transformación del documento padre.
- *XFRM_RPLC (transformwithreplace)*: El documento actual es a la vez una transformación y una sustitución del documento padre.
- *Signs*: El presente documento es una firma digital que firma el documento padre.

```
<rim:Association id="as01" associationType="HasMember" sourceObject="SubmissionSet01" targetObject="Document01">
  <rim:Slot name="SubmissionSetStatus">
    <rim:ValueList>
      <rim:Value>Original</rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:Association>
```

Figura 2.24 Porción del mensaje que representa una asociación.

Fuente: Elaboración propia

- *ObjectRef*: representan la referencia a un objeto del registro, esta etiqueta solo contiene el identificador del objeto pero no sus metadatos por lo que es utilizada en la comunicación para evitar que los mensajes sean demasiado extensos.

```
<rim:ObjectRef id="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"/>
```

Figura 2.25 Porción del mensaje que representa la referencia a un objeto del registro.

Fuente: Elaboración propia

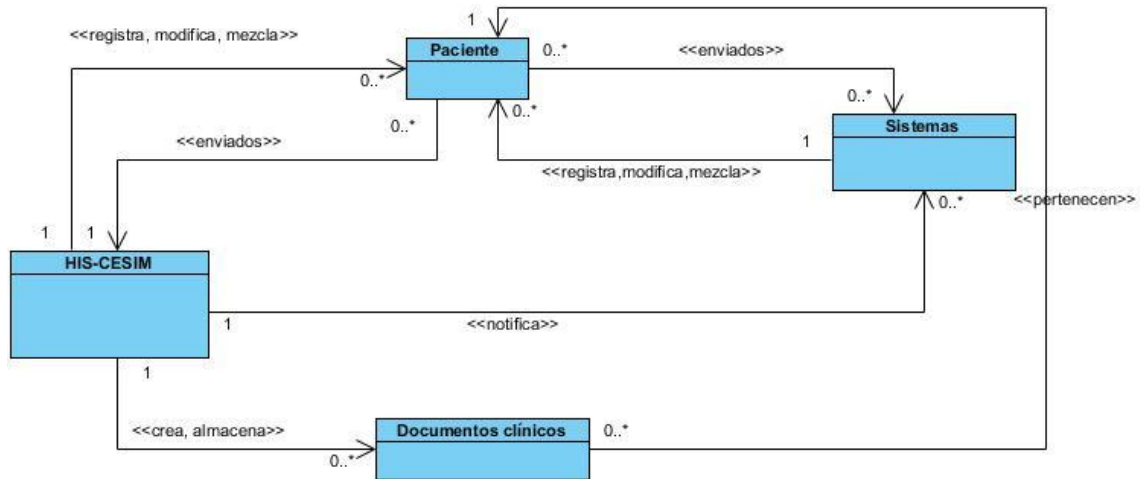


Figura 2.27 Esquema general de comunicaciones

Fuente: Elaboración propia

A continuación se describen cada una de las clases mostradas en el esquema general de comunicaciones. Los atributos definidos son el nombre y la descripción de la clase:

Tabla 2.17 Descripción de la entidad HIS-CESIM

Nombre de la Clase	HIS-CESIM
Descripción	Sistema de Información Hospitalaria desarrollado en el CESIM.

Tabla 2.18 Descripción de la entidad Pacientes

Nombre de la Clase	Pacientes
Descripción	Persona que recibe los servicios que brinda una institución hospitalaria.

Tabla 2.19 Descripción de la entidad Sistemas

Nombre de la Clase	Sistemas
Descripción	Sistemas de gestión de la información de la salud.

Tabla 2.20 Descripción de la entidad Documentos clínicos

Nombre de la Clase	Documentos clínicos
Descripción	Historia clínica del paciente en formato de CDA.

2.4 Clases principales incluidas en la infraestructura de comunicación.

En el desarrollo de la investigación se crearon un conjunto de clases para gestionar el acceso a los datos, los servicios y la mensajería. El siguiente modelo representa la estructura de paquetes utilizada:

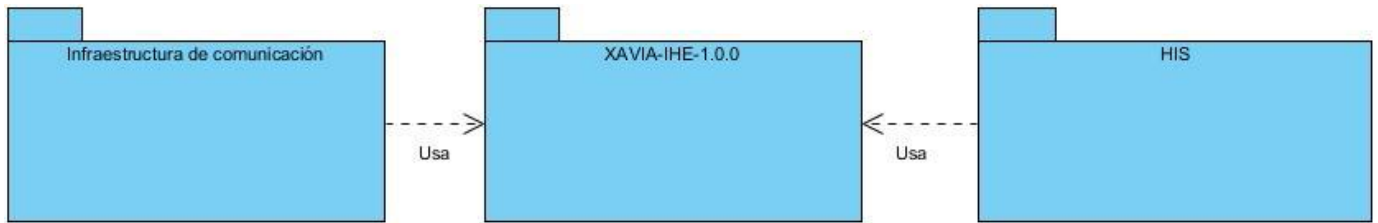


Figura 2.28 Modelo de paquetes

Fuente: Elaboración propia

La infraestructura de integración contiene las clases implementadas en los servicios para gestionar las transacciones y el acceso a los datos. El paquete correspondiente a los mensajes *XAVIA-IHE-1.0.0* es quien contiene la estructura de clases que propone *IHE* para estandarizar la comunicación entre los sistemas de salud (*HIS*, *RIS*, entre otros), este se obtuvo a partir del mapeo de los *WSDL* publicados en el sitio oficial de dicha iniciativa de integración. El paquete *HIS* contiene las clases creadas así como las modificadas necesarias para la conexión con la infraestructura de integración. A continuación se muestran las clases contenidas en los paquetes *Infraestructura de comunicación* y *HIS*:

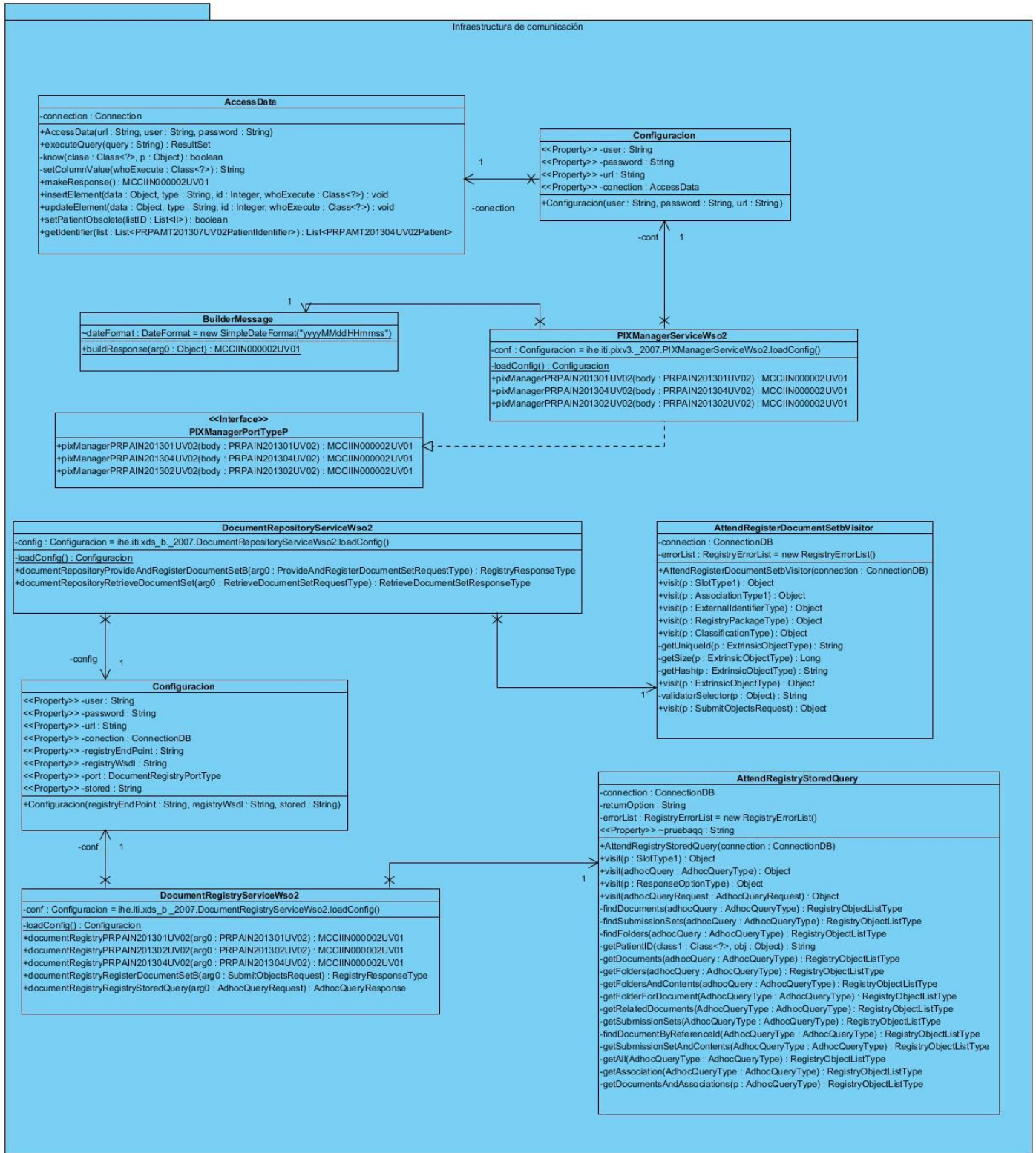


Figura 2.29 Paquete de clases Infraestructura de comunicación

Fuente: Elaboración propia

La descripción de las tablas se encuentra en el artefacto Modelo de diseño ubicado en el expediente de proyecto del Departamento Gestión Hospitalaria perteneciente al CESIM.

2.5 Conclusiones parciales

La utilización del *WSO2 ESB* como parte de la infraestructura de comunicaciones resulta ser esencial pues reduce el tráfico de peticiones realizadas por los sistemas del CESIM.

El diseño de la infraestructura de comunicaciones demuestra ser escalable debido a que se sustenta sobre una arquitectura orientada a servicios.

CAPÍTULO 3. VALIDACIÓN DE LA INFRAESTRUCTURA DE COMUNICACIÓN.

En el presente capítulo se describe el proceso de validación de la infraestructura de comunicación propuesta teniendo en cuenta la aplicación de dicha Infraestructura en el Sistema de Información Hospitalaria, el análisis de rentabilidad de los servicios que ofrece dicha plataforma, pruebas de carga mediante la herramienta *Jmeter* así como la correcta comprobación del flujo que siguen los mensajes de los servicios invocados por los diversos actores mediante la herramienta *Soap Tracer*. Los experimentos antes mencionados se describen en las siguientes secciones.

3.1 Descripción de las interfaces correspondientes a los perfiles XDS y PIX

A partir de la jerarquía de clases definida en la sección 2.4 del capítulo 2 se generaron las interfaces mediante el *WSDL* correspondientes a cada uno de los servicios web. En los anexos 1, 2 y 3 se muestran cada una de las interfaces antes mencionadas. Estas interfaces cumplen con las que propone la Iniciativa de Integración de Sistemas Sanitarios *IHE* en su sitio oficial, por tanto se garantiza que otros componentes de sistemas sanitarios puedan interactuar con la infraestructura de comunicación propuesta de manera estandarizada.

3.2 Aplicación de la infraestructura de comunicación en el Sistema de Información Hospitalaria

Para comprobar las funcionalidades de búsqueda y obtención de HCE así como los metadatos incluidos en la propuesta de solución, se integró dicha infraestructura a los módulos Admisión y Visor de Historia Clínica del Sistema de Información Hospitalaria. Se implementaron las clases *XDSRepositoryConsumer*, *XDSRegistryConsumer* y *PIXConsumer* con las transacciones normadas por *IHE* y mencionadas en la sección 2.4 del capítulo 2 las cuales ofrecen las funcionalidades básicas de los perfiles correspondientes. Dichas clases son configuradas para que se invoquen cuando el servidor de aplicaciones que contiene el *HIS* se inicializa, es decir, en el contexto de aplicación.

Para insertar un documento clínico definido en el perfil XDS a través de la infraestructura de comunicación se modificó el método *crearDocumentoClínico* de la clase *HCMManager* correspondiente al Visor de Historia Clínica como se muestra en la Figura 3.1. La sentencia destacada invoca el método insertar documento en el repositorio incluido en la infraestructura de comunicación.

```

else {
    this.filesManager.deleteFile(filePath);
    HcDocumentoClinico_visorhc docDb = this.documentMetadata.saveDocumentMetadata(doc);
    this.filesManager.saveFileToRepository(signedFilePath, docDb.getHcPersona().getNumeroHc(), docDb.getIdentificadorDocumento());
    try {
        xdsRepositoryConsumer.insertDocument(doc, clase, signedFilePath);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return signedFilePath;
}

```

Figura 3.1 Modificación del método `crearDocumentoClinico` de la clase `HcManager`

Fuente: Elaboración propia

Para el caso de insertar un paciente definido en el perfil *PIX* incluida en la infraestructura propuesta se modificó el método *persistir* de la clase *CrearHC* del módulo de Admisión como se muestra en la siguiente figura:

```

//Notificando a servidor de referencias cruzadas
try {
    pixConsumer.PRPA_IN201301UV02(persona);
} catch (Exception e) {
    System.err.print(e.getMessage());
    System.err.print("----- Error notificando identificación de pacientes -----");
    e.printStackTrace();
}

```

Figura 3.2 Modificación del método *persistir* de la clase *CrearHC*

Fuente: Elaboración propia

Para el caso de modificar un paciente definido en el perfil *PIX* incluida en la infraestructura propuesta se modificó el método *persistir* de la clase *ModificarHistoriaClinica* del módulo de Admisión como se muestra en la siguiente figura:

```

//Notificando al Administrador de referencias cruzadas
try {
    pixConsumer.PRPA_IN201302UV02(persona);
} catch (Exception e) {
    System.err.print(e.getMessage());
    System.err.print("----- Error notificando identificación de paciente modificado -----");
    e.printStackTrace();
}

```

Figura 3.3 Modificación del método *persistir* de la clase *ModificarHistoriaClinica*

Fuente: Elaboración propia

La Figura 3.5 ilustra la ejecución de las funcionalidades con las actualizaciones antes descritas. A la izquierda se muestra la ejecución del registro de una historia clínica mediante la funcionalidad *Crear Historia Clínica* del módulo Admisión que sigue el flujo mostrado en la Figura 3.4, mientras que a la derecha la captura de los mensajes de notificación de identificación de paciente y de registro del documento clínico enviados a la infraestructura en el *Soap Tracer* del *WSO2 ESB*.

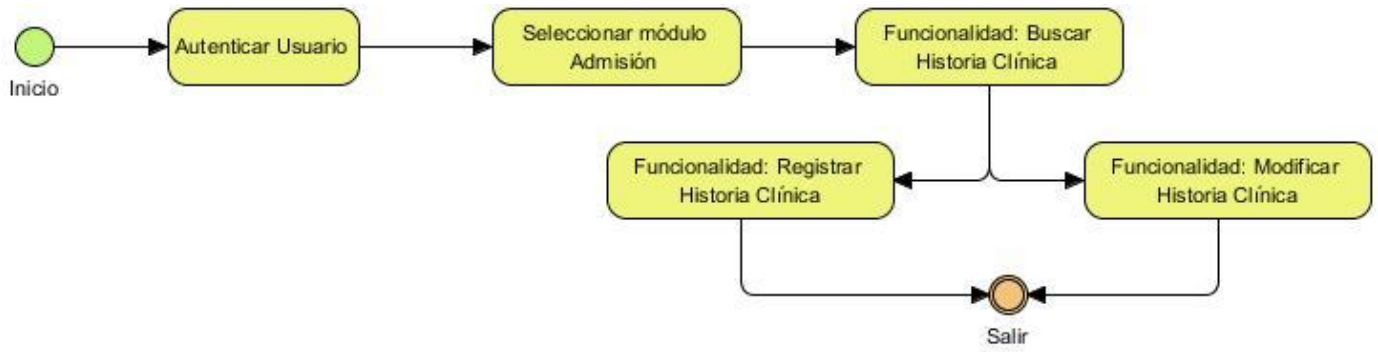


Figura 3.4 Flujo para el registro y modificación de historia clínica

Fuente: Elaboración propia

A continuación se visualiza lo ejecución de la funcionalidad descrita anteriormente:

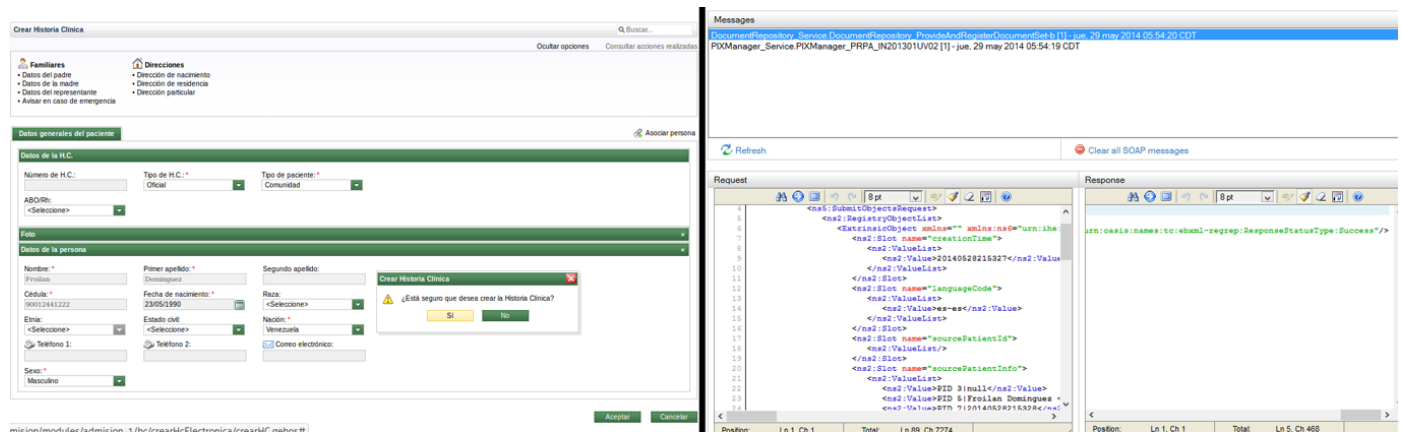


Figura 3.5 Ejecución de *Crear Historia Clínica* mediante la Infraestructura de comunicación

Fuente: Elaboración propia

3.3 Pruebas de rendimiento

Las pruebas de rendimiento o de estrés son las pruebas que se realizan desde una perspectiva para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Existen dos tipos de pruebas de rendimiento: pruebas de carga y de estrés.

Las pruebas de carga se ejecutan para comprender el comportamiento de una aplicación ante una carga determinada. La carga puede estar compuesta por el número de usuarios esperados en producción o un número de transacciones durante un tiempo determinado. El resultado de esta prueba nos ofrece el tiempo de respuesta de todas las transacciones críticas.

Las pruebas de estrés son utilizadas normalmente para someter a la aplicación al límite de su funcionamiento mediante la ejecución de un número de usuarios muy superior al esperado, o bien mediante la substracción de recursos (también conocidas como pruebas negativas donde se simula por ejemplo el fallo de un servidor en *cluster*). Este "*test de stress*" tiene como finalidad determinar la robustez de una aplicación cuando la carga es extrema, y ayuda a los administradores a comprobar la configuración de las alarmas del sistema entre otras cosas. En este tipo de pruebas los tiempos de respuesta de la aplicación no son importantes y tienden a ser ignorados. Otro posible objetivo de este tipo de pruebas es determinar el límite real de la aplicación en cuanto a número de usuarios concurrentes, número de transacciones por segundo, entre otros.

Para garantizar el correcto funcionamiento de la infraestructura de comunicaciones propuesta, se realizaron pruebas para determinar la capacidad de procesamiento, estabilidad y rapidez del mismo con el apoyo del programa *Jmeter*. Las pruebas se aplicaron con las transacciones *Provide and Register Document Set* y *Retrieve Document Set* del perfil *XDS* a la infraestructura de comunicación.

3.3.1 Pruebas aplicadas con la transacción *Provide and Register Document Set*

Se aplicaron pruebas de rendimiento a la infraestructura de comunicación propuesta a través de la transacción *Provide and Register Document Set*. Se definieron 1000 peticiones para dicha transacción según los indicadores: *media*, *mediana*, *% error*, *rendimiento*, y *tráfico de red*. Los resultados arrojados se muestran a continuación:

Informe Agregado									
Nombre: Informe Agregado									
Comentarios									
Escribir todos los datos a Archivo									
Nombre de archivo		Navegar...		Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log <input type="checkbox"/> Sólo Errores <input type="checkbox"/> Éxitos		Configurar			
Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Petición We...	1000	390	364	896	22	1677	0,00%	142,7/sec	14,5
Total	1000	390	364	896	22	1677	0,00%	142,7/sec	14,5

Figura 3.6 Informe agregado para la transacción *Provide and Register Document Set*

Fuente: Herramienta *Jmeter*

Los resultados arrojados en el informe anterior muestran que el rendimiento de la infraestructura de comunicación es de 142,7 peticiones por segundo con una media de 390, lo cual se evidencia gráficamente en la Figura 3.7.

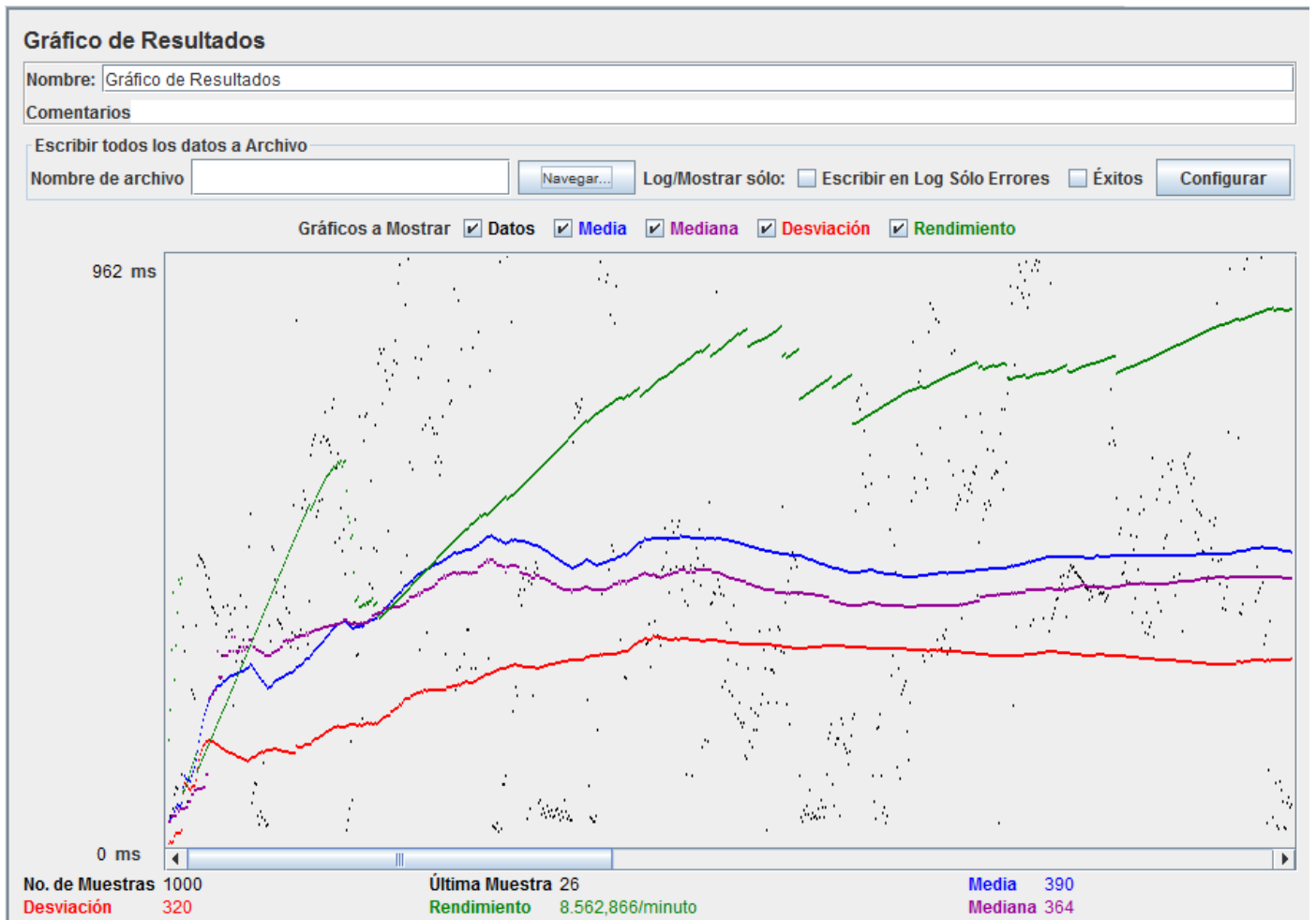


Figura 3.7 Gráfico de resultados para la transacción *Provide and Register Document Set*

Fuente: Herramienta *Jmeter*

Para obtener un grado mayor de detalle con respecto al comportamiento de la infraestructura de comunicaciones se generó una gráfica de *splain* la cual evidencia los picos mínimo y máximo de respuesta a las peticiones realizadas.

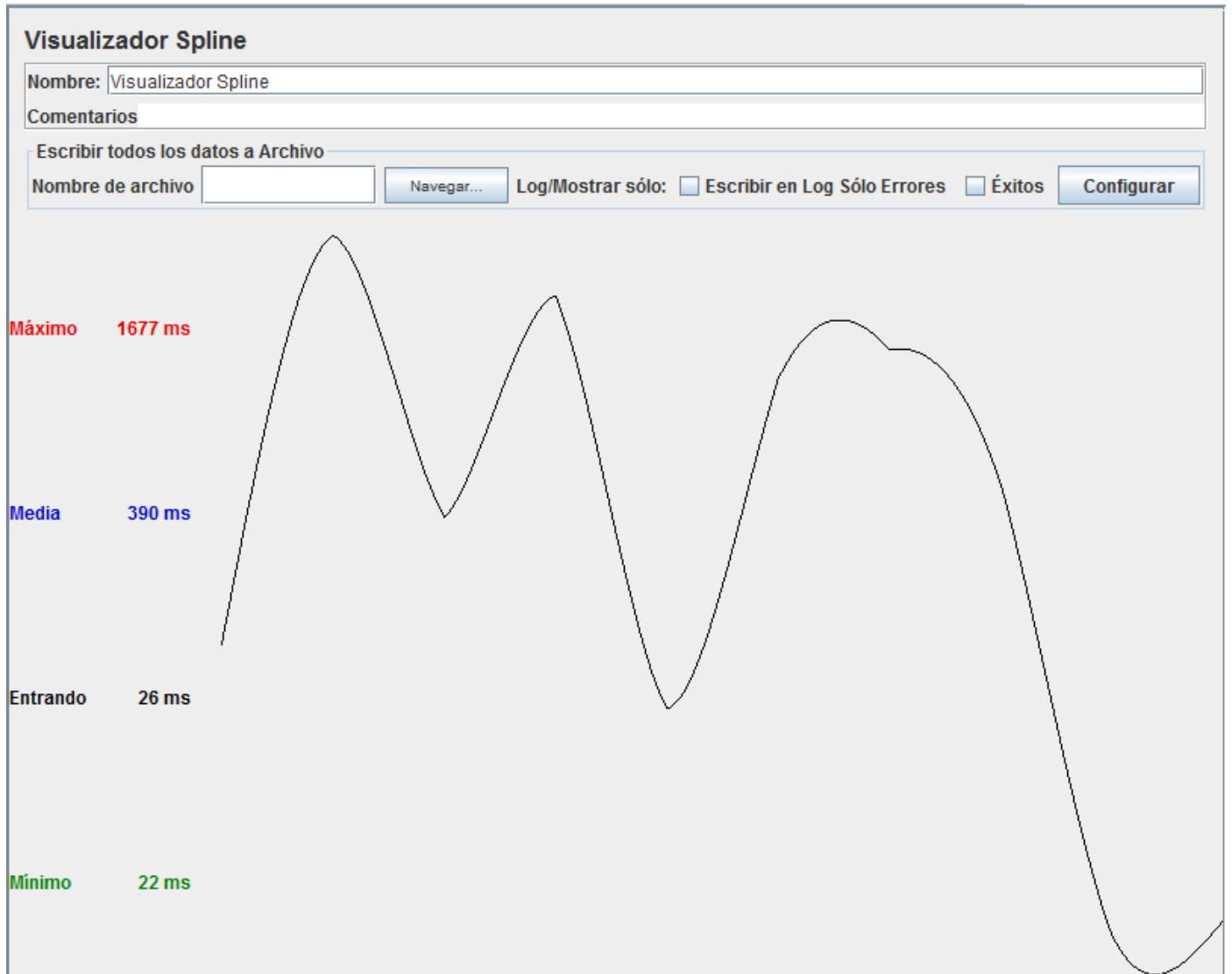


Figura 3.8 Visualizador *Spline* para la transacción *Provide and Register Document Set*

Fuente: Herramienta *Jmeter*

3.3.2 Pruebas aplicadas con la transacción *Retrive Document Set*

De manera adicional a los resultados arrojados por las pruebas de rendimiento a la infraestructura de comunicación propuesta a través de la transacción *Provide and Register Document Set* se realizaron las pruebas a la transacción *Retrive Document Set*, en esta ocasión simulando 3000 peticiones. Los resultados arrojados se muestran a continuación:

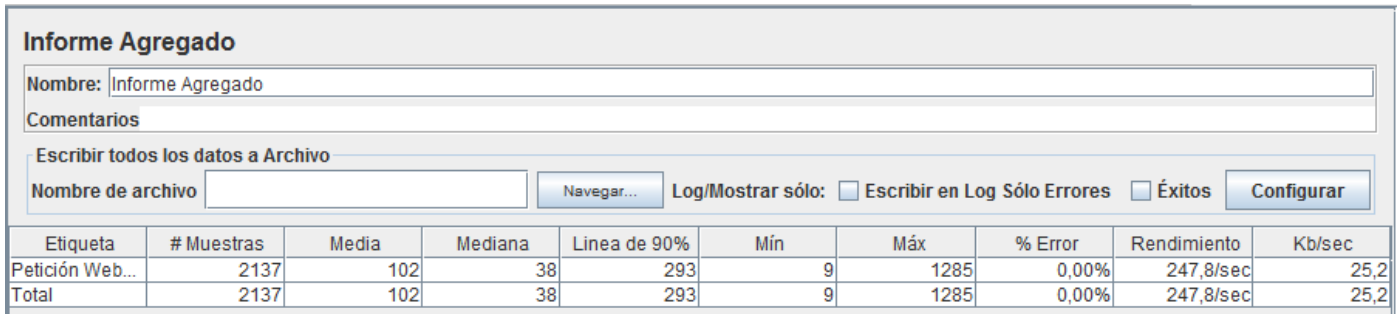


Figura 3.9 Informe agregado para la transacción *Retrive Document Set*

Fuente: Herramienta *Jmeter*

Los resultados arrojados en el informe anterior muestran que el rendimiento de la infraestructura de comunicación es de 247,8 peticiones por segundo con una media de 102, lo cual se evidencia gráficamente en la Figura 3.10.

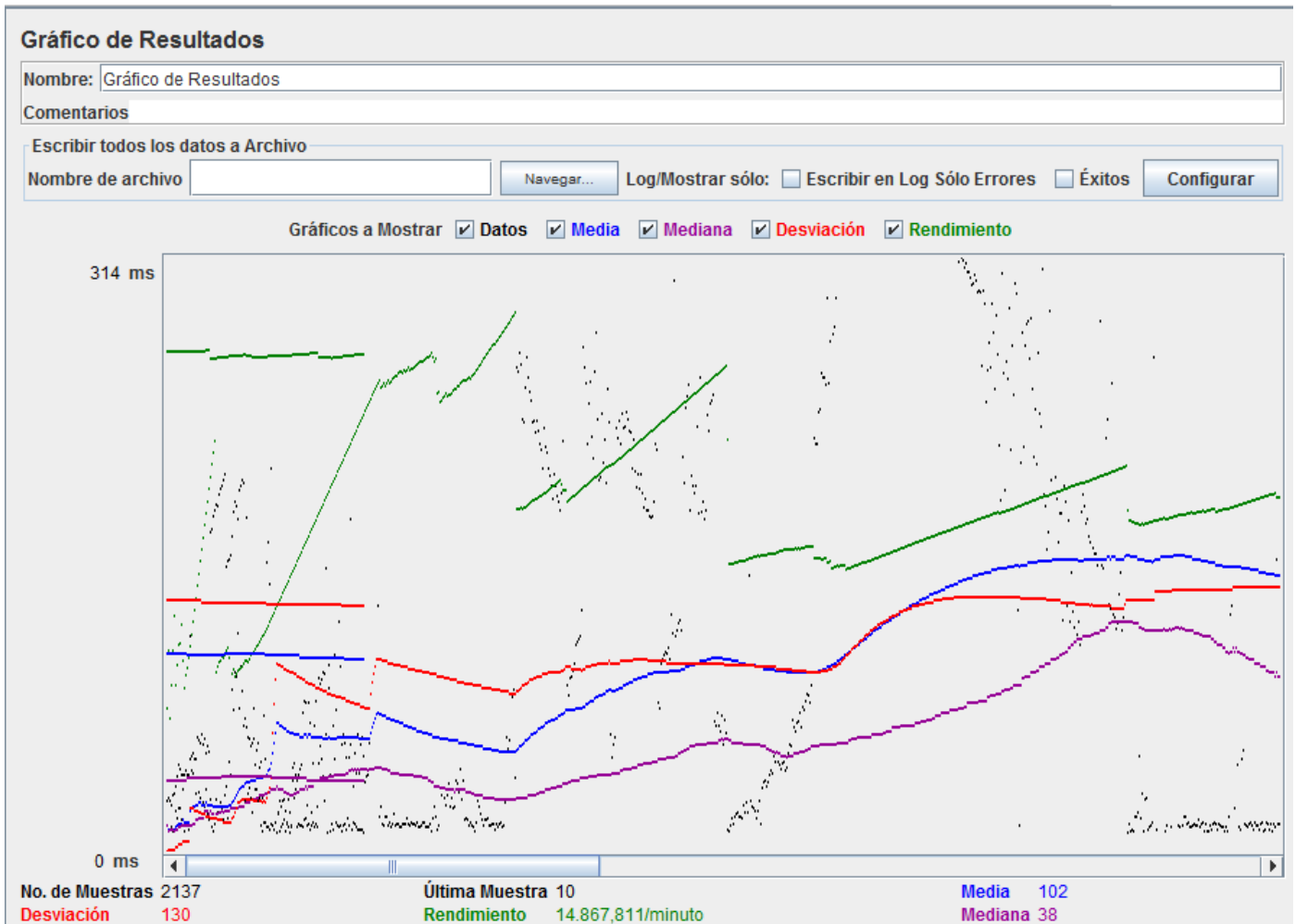
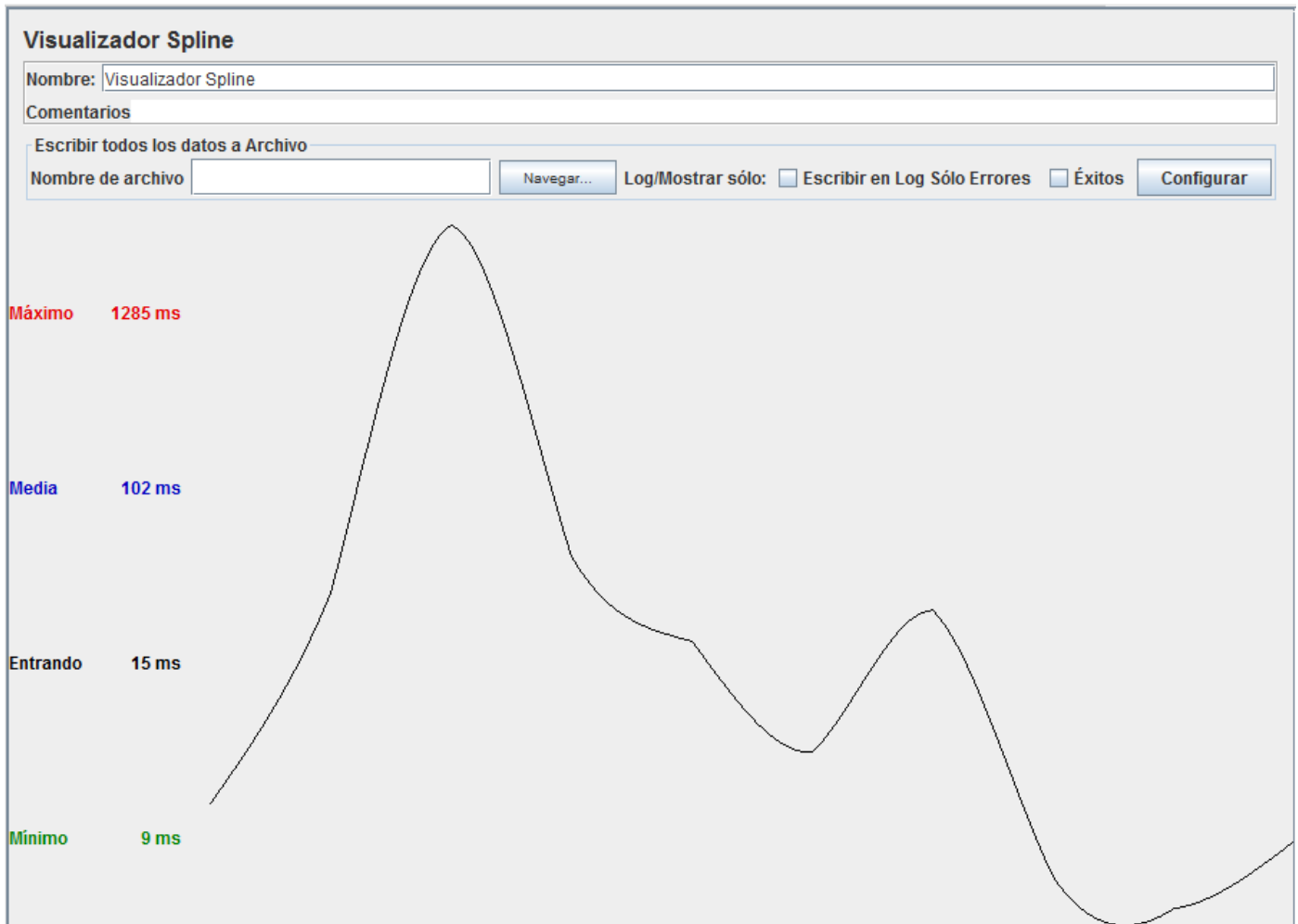


Figura 3.10 Gráfico de resultados para la transacción *Retrive Document Set*Fuente: Herramienta *Jmeter*

Como evidencia de los picos mínimos y máximos de respuesta de la infraestructura de comunicaciones se realizó nuevamente la gráfica *spline*.

Figura 3.11 Visualizador *Spline* para la transacción *Retrive Document Set*Fuente: Herramienta *Jmeter*

3.4 Conclusiones parciales

En este capítulo se realizaron las validaciones al producto desarrollado, las mismas arrojaron las siguientes conclusiones:

La definición del contexto de aplicación en el que se enmarcan las clases *XDSRepositoryConsumer*, *XDSRegistryConsumer* y *PIXConsumer* demuestra no afectar el rendimiento del *HIS* integrado a la infraestructura de comunicaciones.

Las pruebas de rendimiento realizadas permitieron comprobar la eficiencia de la infraestructura de comunicaciones, arrojando rápidas respuestas para la cantidad de usuarios estimados.

CONCLUSIONES

Con el desarrollo de la infraestructura de comunicación basada en los perfiles *XDS* y *PIX* para el Centro de Informática Médica, se concluye lo siguiente:

A partir del análisis de los distintos dominios de integración de *IHE* se determinó escoger el dominio Infraestructura de Tecnología de la Información, el cual se adecúa a los escenarios compartidos de las aplicaciones desarrolladas en el Centro de Informática Médica.

El estudio realizado acerca de las infraestructuras de comunicaciones identificadas por la literatura consultada, sustenta la creación de la infraestructura propuesta, pues dichas investigaciones se desarrollan bajo plataformas cerradas.

La infraestructura de comunicación basada en los perfiles de integración *XDS* y *PIX* pertenecientes al dominio TI permite compartir el repositorio de Historia Clínica Electrónica junto a sus metadatos a los distintos sistemas del CESIM, así como una identificación de los pacientes de manera estandarizada.

Los resultados arrojados por los distintos experimentos expuestos en la sección de resultados demuestran un correcto funcionamiento de la infraestructura de comunicación propuesta debido a que el rendimiento es de 247.8 peticiones por segundo para 3000 conexiones y de 142.7 peticiones por segundo para 1000 conexiones simuladas.

RECOMENDACIONES

Luego de haber cumplido los objetivos propuestos se recomienda implementar los perfiles *ATNA* y *XUA* del dominio de Infraestructura TI para la gestión de la auditoría y autenticación única de usuarios respectivamente. Además, integrar los perfiles *PDQ* y *SVS* desarrollados como parte de otros trabajos de investigación en la infraestructura de comunicación e implementar las funcionalidades necesarias para consumir los servicios que brinda la plataforma de comunicación en los sistemas del Centro de Informática Médica.

REFERENCIA BIBLIOGRAFICA

1. **España, Ministerio de Sanidad.** Sitio oficial del Ministerio de Sanidad España. [En línea] Ministerio de Sanidad, Servicios Sociales e Igualdad, 22 de 12 de 2009. [Citado el: 20 de 02 de 2014.] <http://www.msssi.gob.es/profesionales/hcdsns/home.htm>.
2. **Cerritos, Antonio.** *Sistema de Información Hospitalaria.* México D.F : s.n., 2009.
3. **Bordils i Rovira, Francisco y Chavarría Díaz, Miguel.** *Almacenamiento y transmisión de imágenes. PACS.* 2004.
4. **López, Andrés M Manzaneque.** *RIS (Sistema de Información Radiológica) y PACS (Sistema de Comunicación y Archivo de Imágenes Médicas).* 2010.
5. **IEEE.** *IEEE.* 1990.
6. **Spain, HL7.** Health Level Seven Spain. [En línea] [Citado el: 17 de 12 de 2009.] <http://www.hl7spain.org/VerPagina.asp?IDPage=0>.
7. **Manufacture, National Electrical.** *Digital Imaging and Communications in Medicine (DICOM).* Virginia 22209, USA : s.n., 2003.
8. **IHE International, Inc.** IHE Integrating the Healthcare Enterprise. [En línea] IHE International, 2013. [Citado el: 10 de 01 de 2014.] http://ihe.net/Technical_Frameworks/#IT.
9. **Sánchez, Carlos Pedro Soberats.** *Sistema de detección de personas duplicadas en un Sistema de Gestión de Informática Médica.* Centro de Informática Médica, Universidad de las Ciencias Informáticas. La Habana : s.n., 2011. pág. 80, Tesis de grado.
10. **Reidl, Lucy y Peresmitré, Gilda Gómez.** *METODOLOGÍA DE INVESTIGACIÓN EN CIENCIAS SOCIALES.* s.l. : U.N.A.M., 2013. pág. 100.
11. **IHE International, Inc.** *IHE IT Infraestructure (ITI) Technical Framework.* 2013.
12. **C2C Consultoría TSIS, S.L. The eHealth Company.** C2C The eHealth Company. [En línea] 2013. [Citado el: 26 de 05 de 2014.] www.c2ctis.com/archives/231.
13. **M., Mario Enrique Cortés.** HL7 en Español. [En línea] 12 de 12 de 2013. [Citado el: 25 de 02 de 2014.] <http://hl7es.blogspot.com/2013/12/fhir-el-nuevo-miembro-de-la-familia-hl7.html>.
14. **IHE Internacional, Inc.** *IHE_ITI_TF.* 2014. 2013. pág. 199. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
15. —. *IHE_ITI_TF.* 2013. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
16. —. *IHE_ITI_TF.* 2013. pág. 55. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
17. —. *IHE_ITI_TF.* 2013. pág. 59. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.

18. —. *IHE_ITI_TF*. 2013. pág. 61. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
19. —. *IHE_ITI_TF*. 2013. pág. 67. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
20. —. *IHE_ITI_TF*. 2013. pág. 81. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
21. —. *IHE_ITI_TF*. 2013. pág. 117. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
22. —. *IHE_ITI_TF*. 2013. pág. 127. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
23. —. *IHE_ITI_TF*. 2013. pág. 140. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
24. —. *IHE_ITI_TF*. 2013. pág. 145. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
25. —. *IHE_ITI_TF*. 2013. pág. 151. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
26. —. *IHE_ITI_TF*. 2013. pág. 171. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
27. —. *IHE_ITI_TF*. 2013. pág. 183. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
28. —. *IHE_ITI_TF*. 2013. pág. 222. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
29. —. *IHE_ITI_TF*. 2013. pág. 227. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
30. —. *IHE_ITI_TF*. 2013. pág. 231. Vol. 1 [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
31. —. *IHE_ITI_TF*. 2013. Vol. 2a [En línea] 2013. [Citado el: 10 de 12 de 2013.] http://ihe.net/Technical_Frameworks/#IT.
32. **IHE International**. Integrating the Healthcare Enterprise. [En línea] 2013. [Citado el: 13 de 04 de 2014.] www.ihe.net/connectathon.
33. **InterSystems**. InterSystems HealthShare. [En línea] 05 de 09 de 2013. [Citado el: 25 de 03 de 2014.] <http://www.hospitaldigital.com/2013/09/05/intersystems-healthshare-recibe-la-certificacion-de-interoperabilidad-ti-sanitaria-de-ihe-usa-e-icsa-labs/>.

34. **ICSA Labs.** ICSA Labs. [En línea] 2014. [Citado el: 26 de 05 de 2014.] <http://www.icsalabs.com/technology-program/ihe-usa-certifiacion>.
35. **NextGate Solutions, Inc.** NEXTGATE. [En línea] NextGate Solutions, Inc, 2014. [Citado el: 21 de 3 de 2014.] <http://www.nextgate.com/es/our-products/ihe-compliance/>.
36. **Open eHealth.** Open eHealth. [En línea] 2014. [Citado el: 26 de 05 de 2014.] <http://www.openehealth.org/display/ipf2/Home>.
37. **Martha Rodríguez López, Raymundo Rodríguez García.** *Propuesta de aplicación de los perfiles de integración de IHE entre los sistemas alas PACS - alas RIS - alas HIS.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. Trabajo de Diploma.
38. **González, Reinier Alonso.** *Modelo de producción de software para el Centro de Informática Médica.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. págs. 49-53, Tesis de Maestría.
39. **Apache Software Foundation.** Apache Camel. [En línea] 2004-2014. [Citado el: 11 de 12 de 2013.] <https://caml.apache.org>.
40. **GoPivotal, Inc.** Spring. [En línea] 2014. [Citado el: 10 de 04 de 2014.] projects.spring.io/spring-framework/.
41. **Apache.** Apache CXF. [En línea] [Citado el: 23 de 02 de 2014.] <http://cxf.apache.org/>.
42. **Comunity, Jboss.** JBoss Web Services. [En línea] Redhat, Inc, 23 de 02 de 2014. <https://www.jboss.org/jbossws>.
43. **Foundation, Apache Software.** The Apache Software Foundation. [En línea] Apache Software Foundation, 2009. <http://axis.apache.org/axis2/c/core>.
44. **Digital, Noción.** Noción Digital. [En línea] azuanet, 2007. [Citado el: 20 de 01 de 2010.] <http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>.
45. **InterSystem.** InterSystem Spain. [En línea] 2014. [Citado el: 20 de 05 de 2014.] http://www.intersystems.es/page/es/overview_servidor_aplicaciones.html.
46. **Oracle.** Oracle Store. [En línea] 2013. [Citado el: 20 de 05 de 2014.] <https://shop.oracle.com/pls/ostore/product?p1=OracleGlassFishServer>.
47. **Apache.** Apache Tomcat. [En línea] 2014. [Citado el: 20 de 05 de 2014.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.
48. **WSO2 Corporation.** WSO2. [En línea] 2013. [Citado el: 20 de 05 de 2014.] <http://wso2.com/products/application-server/>.
49. **WSO2.** WSO2 Oficial Site. [En línea] WSO2, 2014. [Citado el: 24 de 02 de 2014.] <http://wso2.org/library/articles/2013/01/esb-performance-65>.
50. **Rouse, Margaret.** SearchSOA. [En línea] Octubre de 2007. [Citado el: 26 de 05 de 2014.] <http://searchsoa.techtarget.com/definition/enterprise-service-bus>.

51. **Cobia.** Cobia Blog. [En línea] 2013. [Citado el: 24 de 02 de 2014.] <http://blog.cobia.net/cobiacom/2012/04/18/how-to-pick-an-esb-an-enterprise-service-bus-evaluation-framework/>.
52. **Inc, W.** *Welcome to WSO2 Enterprise Service Bus (ESB) v4.0.3 2011.* 2012.
53. **Cobiacom.** IT Business Navigation Charts. [En línea] 08 de 2012. [Citado el: 20 de 02 de 2014.] <http://blog.cobia.net/cobiacom/2012/08/01/esb-comparison/>.
54. **Osorio, Jorge Infante.** Desarrollo SOA en la Comunidad WSO2-Latinoamérica. [En línea] Blogger, 2014. [Citado el: 02 de 05 de 2014.] desarrollosa.blogspot.com/2014/03/el-esb-de-wso2-se-considera-el-esb-mas.html.
55. **Studio, Visual.** Visual Studio. [En línea] [Citado el: 23 de 02 de 2014.] <http://www.visualstudio.com/>.
56. **Netbeans.** Netbeans. [En línea] Oracle Corporation and/or its affiliates, 2013. [Citado el: 23 de 02 de 2014.] <https://netbeans.org/features/index.html>.
57. **Comunity, Jboss.** Jboss Comunity. [En línea] RedHat. [Citado el: 23 de 02 de 2014.] <https://www.jboss.org/products/jbds>.
58. **Storkel, Scott. O'Reilly.** Java.com. [En línea] O'Reilly Media, Inc, 23 de 01 de 2012. <http://onjava.com/pub/a/onjava/2002/12/11/eclipse.html>.
59. **Enterprise Architect.** Herramienta de diseño UML y herramienta CASE UML para desarrollo de software. [En línea] 2012. [Citado el: 26 de 11 de 2013.] <http://sparxsystems.com.ar/products/ea.html>.
60. **Rational Rose Enterprise.** Paquete Rational Rose. [En línea] 2011. [Citado el: 26 de 11 de 2013.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
61. **Paradigm, Visual.** Free Download Manager. [En línea] 6.0, 2009. [Citado el: 15 de 12 de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
62. **Apache.** Apache Jmeter. [En línea] 2012. [Citado el: 10 de 12 de 2013.] <http://jmeter.apac>.
63. **Oracle.** Your First Cup Java. [En línea] 2012. [Citado el: 12 de 06 de 2014.] <http://docs.oracle.com/javase/6/firstcup/doc/gkhoy.html#gcrlz>.
64. **W3C.** W3C. [En línea] 2013. [Citado el: 26 de 05 de 2014.] <http://www.w3.org/standards/webofservices/>.
65. —. WSDL CURRENT STATUS. [En línea] 2014. [Citado el: 20 de 05 de 2014.] http://www.w3.org/standards/techs/wsdl#w3c_all.
66. **Dianella Rosales Pérez, Raul Marquez Cabrera.** *Propuesta de Modelo de Estructura y Evolución para un Centro de Excelencia SOA.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. Trabajo de Diploma.
67. **Pelaez, Juan.** Arquitectura Orientada a Servicios (SOA). [En línea] 27 de 05 de 2009. [Citado el: 26 de 05 de 2014.] <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-orientada-a-servicios-soa/>.
68. **IHE Internacional, Inc.** *IHE_ITI_TF.* 2013. Vol. 3.

ANEXOS

```

- <wsdl:definitions name="DocumentRepository_Service" targetNamespace="urn:ihe:iti:xds-b:2007">
- <wsdl:types>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:ihe:iti:xds-b:2007" version="1.0"></xs:schema>
+ <xs:schema targetNamespace="http://www.w3.org/XML/1998/namespace" version="1.0"></xs:schema>
</wsdl:types>
+ <wsdl:message name="DocumentRepository_ProvideAndRegisterDocumentSet-b"></wsdl:message>
+ <wsdl:message name="DocumentRepository_ProvideAndRegisterDocumentSet-bResponse"></wsdl:message>
+ <wsdl:message name="DocumentRepository_RetrieveDocumentSet"></wsdl:message>
+ <wsdl:message name="DocumentRepository_RetrieveDocumentSetResponse"></wsdl:message>
+ <wsdl:portType name="DocumentRepository_PortType"></wsdl:portType>
+ <wsdl:binding name="DocumentRepository_ServiceSoapBinding" type="tns:DocumentRepository_PortType"></wsdl:binding>
+ <wsdl:service name="DocumentRepository_Service"></wsdl:service>
</wsdl:definitions>

```

Anexo 1. WSDL correspondiente al repositorio del perfil XDS según lo establecido por IHE

```

- <wsdl:definitions name="DocumentRegistry_Service" targetNamespace="urn:ihe:iti:xds-b:2007">
- <wsdl:types>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0" version="1.0"></xs:schema>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:hl7-org:v3" version="1.0"></xs:schema>
+ <xs:schema targetNamespace="http://www.w3.org/XML/1998/namespace" version="1.0"></xs:schema>
</wsdl:types>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201301UV02"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201301UV02Response"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201304UV02"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_RegistryStoredQueryResponse"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_RegistryStoredQuery"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201302UV02"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201304UV02Response"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_RegisterDocumentSet-bResponse"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_PRPA_IN201302UV02Response"></wsdl:message>
+ <wsdl:message name="DocumentRegistry_RegisterDocumentSet-b"></wsdl:message>
+ <wsdl:portType name="DocumentRegistry_PortType"></wsdl:portType>
+ <wsdl:binding name="DocumentRegistry_ServiceSoapBinding" type="tns:DocumentRegistry_PortType"></wsdl:binding>
+ <wsdl:service name="DocumentRegistry_Service"></wsdl:service>
</wsdl:definitions>

```

Anexo 2. WSDL correspondiente al registro del perfil XDS según lo establecido por IHE

```

- <wsdl:definitions name="PIXManager_Service" targetNamespace="urn:ihe:iti:pixv3:2007">
- <wsdl:types>
+ <xs:schema elementFormDefault="qualified" targetNamespace="urn:hl7-org:v3" version="1.0"></xs:schema>
</wsdl:types>
+ <wsdl:message name="PIXManager_PRPA_IN201304UV02"></wsdl:message>
+ <wsdl:message name="PIXManager_PRPA_IN201302UV02Response"></wsdl:message>
+ <wsdl:message name="PIXManager_PRPA_IN201302UV02"></wsdl:message>
+ <wsdl:message name="PIXManager_PRPA_IN201301UV02Response"></wsdl:message>
+ <wsdl:message name="PIXManager_PRPA_IN201304UV02Response"></wsdl:message>
+ <wsdl:message name="PIXManager_PRPA_IN201301UV02"></wsdl:message>
+ <wsdl:portType name="PIXManager_PortType"></wsdl:portType>
+ <wsdl:binding name="PIXManager_ServiceSoapBinding" type="tns:PIXManager_PortType"></wsdl:binding>
+ <wsdl:service name="PIXManager_Service"></wsdl:service>
</wsdl:definitions>

```

Anexo 3. WSDL correspondiente al perfil *PIX* según lo establecido por *IHE*