Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Integración del Sistema de Información de Radiodiagnóstico Corporativo de la Conselleria de Sanidad de la Comunidad Valenciana, Orion-RIS, con el Sistema regional de Historia de Salud Electrónica

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Diego J. Domínguez Carralero

Tutor: Juan Luis Posadas Yagüe

2014-2015

Resumen

HSE es el sistema centralizado de información de Historia de Salud Electrónica, que contiene las actuaciones y episodios, en el área de la salud, que se realizan durante la vida de un paciente.

Orion-RIS es el sistema distribuido de información de Radiodiagnóstico corporativo de la Conselleria de Sanidad de la Comunidad Valenciana, que actualmente se encuentra en producción en 20 hospitales de toda la región, entre los que se encuentran el Hospital Politécnico Universitario La Fe, el Hospital General de Alicante, el Hospital Clínico de Valencia, el Hospital General de Castellón, etc. Abarca todo el ciclo de vida en una cita a un paciente para la realización de un estudio de radiodiagnóstico, desde la creación de la cita, a la realización de la actividad y finalmente el informado de la misma.

En la sociedad actual en que la información es clave, se pretende enriquecer la información de la que se dispone en la Historia de Salud Electrónica, añadiendo la información de los informes de actividad de radiodiagnóstico, cualquiera que sea su departamento de origen, lo que permitirá la consulta de la actividad de rayos desde cualquier aplicación que se integre con HSE, ahora o en el futuro.

La integración se llevará a cabo mediante la publicación e invocación de Webservices, con WS-Security, utilizando la implementación Apache CXF (framework completo de código abierto para servicios web).

Palabras clave: Orion-RIS, HSE, radiodiagnóstico, WebServices, integración, CXF.

Tabla de contenidos

Índice de Imágenes

Índice de Tablas

1. Introducción

1.1 Presentación

Las organizaciones sanitarias en general, y los hospitales en particular, son entidades muy grandes y complejas, en las que conviven literalmente centenares de aplicaciones informáticas, con complicadas interacciones entre ellas. Tienen que enfrentarse a tecnologías siempre en evolución, y responder con agilidad a los cambios en la regulación y la presión de la administración con soluciones flexibles, robustas y sencillas.

Esta situación conlleva un coste económico muy elevado, con la construcción, mantenimiento y adaptación de sistemas, y otro coste, que puede no resultar tan obvio, en tiempo, en la definición de circuitos de compartición de la información, en workflows de trabajo, con resultados de pruebas que dependen unas de otras, y cuya gestión resulta muy compleja.

La integración constituye la clave para la reducción paulatina de estos costes, con la puesta en común de la mayor cantidad posible de información, evitando redundancias y omisiones, para que la atención sanitaria sea lo mejor posible tanto para los pacientes como para los profesionales de la salud que desarrollan su trabajo con multitud de estas herramientas informáticas.

En este trabajo de fin de grado nos centraremos en definir y realizar la integración entre dos sistemas de información:

- HSE es el sistema centralizado de información de Historia de Salud Electrónica, que contiene las actuaciones y episodios, en el área de la salud, que se realizan durante la vida de un paciente.
- Orion-RIS es el sistema distribuido de información de Radiodiagnóstico corporativo de la Conselleria de Sanidad de la Comunidad Valenciana.

1.2 Objetivos

La Conselleria de Sanidad emplea cientos de miles de euros en licencias de productos destinados a facilitar dichas integraciones, entre aplicaciones construidas con diferentes tecnologías y capacidades de comunicación, y que se ejecutan apoyadas sobre software propietario.

La actual situación económica, en la que es necesario más que nunca controlar el presupuesto económico, hace necesaria una renovación tecnológica, dotando a las aplicaciones de mecanismos más modernos de comunicaciones e integraciones, y adoptando soluciones estándar en cuanto a configuraciones en lugar de optar por soluciones propietarias, con el objetivo de eliminar intermediarios y proveedores, y poder realizar progresivamente las migraciones a software libre.

Con la integración con HSE se pretende poner una primera piedra en la modernización y estandarización de las integraciones de Orion-RIS, adoptando tecnologías y sistemas de acreditación y seguridad más actuales.

Al mismo tiempo, se pretende potenciar las capacidades funcionales de Orion-RIS, evitando de este modo su funcionamiento en modo exclusivamente departamental. Se pretende integrar la consulta de informes de radiodiagnóstico realizados en hospitales diferentes dentro de la propia aplicación, facilitando de este modo a los profesionales la consulta y seguimiento de las características del paciente, las pruebas que se le han realizado, y los hallazgos encontrados en dichas pruebas.

1.3 Descripción del documento

Se describe a continuación a grandes rasgos la estructura del documento que recoge el proyecto de Integración de Orion-RIS con la Historia de Salud Electrónica:

Apartado 1: Es el apartado actual, que recoge la introducción al proyecto.

Apartado 2: En él se realiza la presentación de las aplicaciones implicadas.

Apartado 3: Lista los requisitos establecidos para la integración.

Apartado 4: Recoge el diseño técnico y funcional realizado en el desarrollo del proyecto.

Apartado 5: Muestra detalles de la implementación de la integración realizada.

Apartado 6: Especifica las pruebas realizadas.

Apartado 7: Recoge las conclusiones obtenidas en el desarrollo de este proyecto.

2. Presentación de las aplicaciones

2.1 Historia de Salud Electrónica

La Agencia Valenciana de Salud inicia el proyecto de Historia de Salud Electrónica (HSE) en el 2008 en una participación con el proyecto Historia Clínica Digital del Sistema Nacional de Salud (HCDSNS). Este novedoso proyecto permite el acceso a documentos clínicos de pacientes que estén disponibles en otras comunidades autónomas. El nodo local de HCDSNS de la Comunidad Valenciana servirá de proyecto piloto para la creación de un sistema regional de intercambio de documentos clínicos, a través de la aplicación Historia de Salud Electrónica.

El proyecto de Historia de Salud Electrónica tiene como finalidad garantizar a los ciudadanos y a los profesionales sanitarios el acceso a aquella información clínica relevante para la atención sanitaria de un paciente desde cualquier sistema de información utilizado en el ámbito de la Comunidad Valenciana, asegurando a los ciudadanos que la consulta de sus datos queda restringida a quién está autorizado para ello, y evitando que los pacientes atendidos en diversos centros se sometan a exploraciones y procedimientos de innecesaria repetición.

El sistema HSE, como queda reflejado en la Figura 1, es un elemento troncal en el plan de sistemas de información de la Conselleria de Sanidad de la Generalitat Valenciana como base del proceso asistencial, fundamentalmente soportado por los sistemas Orion-RIS (Radiodiagnóstico), Orion-Clinic (Historia Clínica), SIA (Atención Primaria), Alta Hospitalaria y Cordex (Emergencias), y como elemento vertebrador de los procesos colaborativos existentes entre los diferentes ámbitos asistenciales, facilitando la interoperabilidad entre los sistemas presentes en la Comunidad Valenciana, y los de otras regiones a través del nodo local HCDSNS.

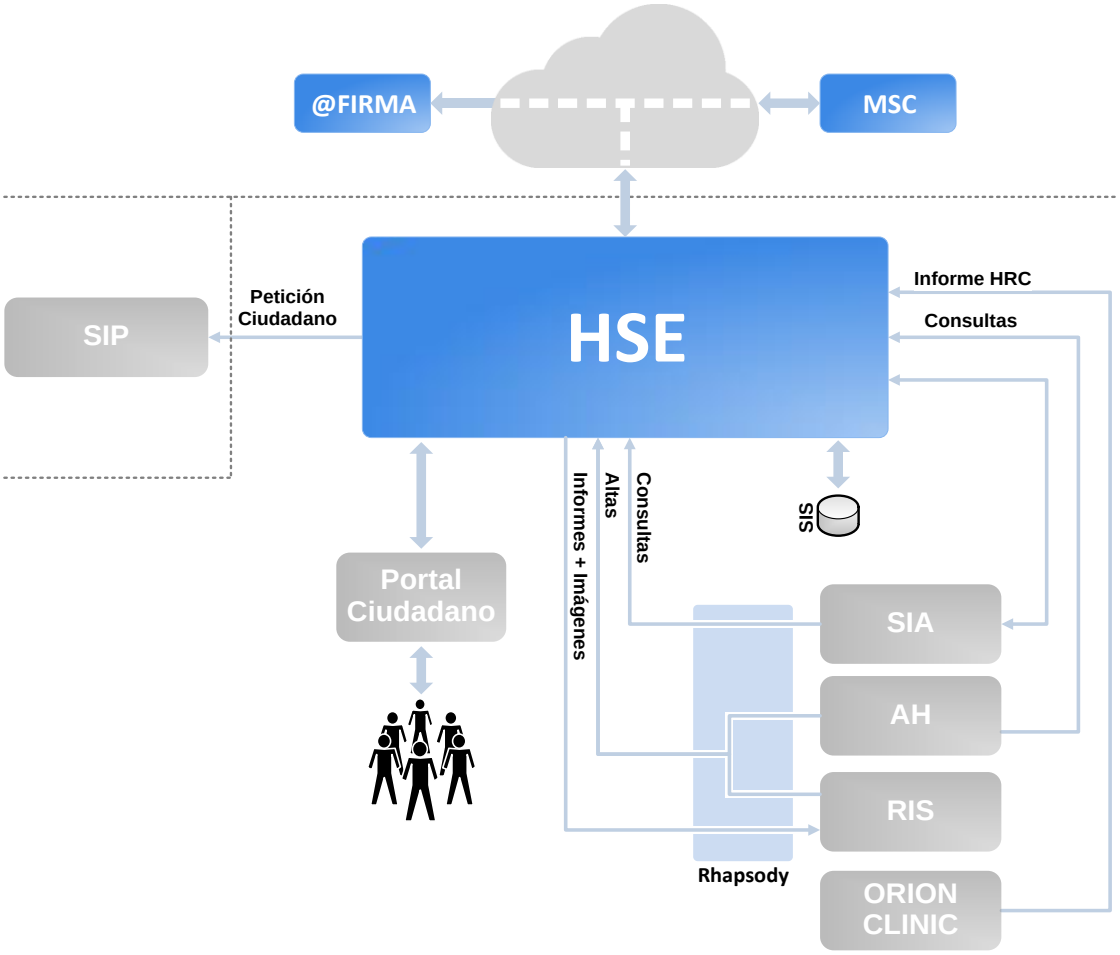

Figura 1: Esquema de Integración de HSE

La Historia de Salud Electrónica (HSE) se entiende como un complejo conjunto de información personal relacionada con los diferentes estados de salud y enfermedad de los ciudadanos que se genera a lo largo de la vida y que se registra, almacena y utiliza en un entorno digital en todo su ciclo vital con objeto de mejorar el estado de salud de una población y de sus individuos.

El sistema HSE ha sido diseñado para ser capaz de operar con los sistemas de información de salud de otras regiones, participando activamente en grupos de interoperabilidad y pilotos de proyectos nacionales (Historia electrónica del Sistema Nacional de Salud) y europeos (epSOS proyecto europeo de eSalud; European Patients – Smart open Services) de salud.

Facilita la interoperabilidad entre las diferentes aplicaciones asistenciales utilizadas en el ámbito de la Comunidad Valenciana, permitiendo el acceso de los profesionales sanitarios a través de sus aplicaciones habituales a toda la información que sea relevante para la atención sanitaria de los ciudadanos, independientemente de dónde se haya generado, posibilitando de manera proactiva:

- Que el usuario sea informado por su aplicación habitual de la información relevante disponible de cualquier paciente al que va a atender.
- Que el usuario sea informado por su aplicación habitual de acontecimientos médicos relevantes (ingreso hospitalario, alta hospitalaria,…) ocurridos a pacientes de su cupo en cualquier lugar del Sistema Nacional de Salud, aunque dichos pacientes no estén citados ese día (buzón de notificaciones).
- Que el ciudadano tenga acceso online a los datos de salud, propios o de sus representados, que se encuentren disponibles en formato digital en alguno de los Sistemas de Información que se integren en HSE.
- El aumento de la seguridad y calidad de atención de los pacientes, asegurando la precisión de los datos clínicos, reduciendo la incidencia de los errores médicos y ahorrando costes al evitar servicios duplicados.

Todo esto, por supuesto con las medidas de seguridad necesarias para garantizar al ciudadano la confidencialidad de los datos de carácter personal relativos a su salud, permitiéndole auditar el registro de accesos realizados a sus datos.

Entre los datos que facilita, se encuentran los siguientes:

- Historia Clínica Resumida.
- Informes de atención en Urgencias.
- Informes clínicos de alta de hospitalización.
- Informes de resultados de pruebas de laboratorio.
- Informes de pruebas de Imagen (Radiodiagnóstico, Med. Nuclear).
- Informes de resultados de otras pruebas diagnósticas.
- Informe de Cuidados de Enfermería.
- Portal del Ciudadano.
- Otros: Guías clínicas, Gestor de alarmas sanitarias, Gestor de Peticiones Corporativo, Teleasistencia…

HSE consta de cuatro entornos de operación, con características y usos diferentes:

- **PRO:** El entorno de producción es el utilizado por los usuarios finales tanto médicos como profesionales. Este entorno está ubicado en el centro de informática de sanitario y es accesible dentro de la red corporativa. La URL del entorno es https://hseavs.san.gva.es

- **PRE:** El entorno de preproducción de HSE es utilizado únicamente por el equipo de calidad de la Conselleria de Sanidad y no será público para ninguna integración. Este entorno está ubicado en el centro de informática de sanitario y es accesible dentro de la red corporativa. La URL del entorno es https://hseavspre.san.gva.es

- **TEST:** El entorno de TEST de HSE es utilizado tanto para desarrollar y probar las integraciones en curso como para realizar la homologación o certificación de la integración que permitirá el paso a producción. Este entorno está ubicado en el centro de informática de sanitario y es accesible dentro de la red corporativa. La URL del entorno es https://hseavstest.san.gva.es

- **INT:** El entorno de integración es utilizado únicamente durante el desarrollo de la integración y la fase de pruebas iniciales. Este entorno está ubicado en las instalaciones de la empresa proveedora de desarrollo y no está incluido en la red corporativa. Sobre este entorno existe flexibilidad completa para habilitar comunicaciones HTTP o HTTPS y del mismo modo también se puede activar o desactivar la seguridad bajo demanda y así evitar problemas que puedan interferir durante el desarrollo de la integración.

2.2 Orion-RIS

La Agencia Valenciana de Salud inicia en 2005 el proyecto Orion, un ambicioso plan de modernización de los sistemas de información hospitalarios.

Se elige como punta de lanza para este proyecto global el servicio de Radiodiagnóstico por su gran importancia en el ámbito hospitalario, en el que se constituye como un servicio trasversal, del que dependen en buena medida muchos otros servicios que hacen uso de sus técnicas y resultados.

Orion-RIS (Orion – Sistema de Información de Radiodiagnóstico, por sus siglas en inglés) se convierte de este modo en la aplicación corporativa de imagen de radiodiagnóstico de la Conselleria de Sanidad, que se utiliza por primera vez en el Hospital Policlínico Universitario La Fe de Valencia, concretamente en el área de Infantil, el 6 de Junio de 2007, y que actualmente se emplea en 22 de los principales hospitales de la Comunidad Valenciana, junto con sus correspondientes centros de Especialidades y centros de Salud.

La aplicación es utilizada en los servicios de radiodiagnóstico de los hospitales para dar soporte a las tareas propias del servicio, abarcando el ciclo de vida completo de un episodio, agrupadas bajo módulos de pantalla única, en los que el personal tiene al alcance de su vista todo lo necesario para la realización de sus funciones, entre las que se encuentran:

- Citación de Pacientes: Bajo el módulo de Citación se engloban tareas como la citación de pacientes para la realización de pruebas, la reprogramación de las mismas, el registro de solicitudes de cita para aquellas que no puedan citarse sin una valoración previa, y otras funcionalidades propias de la gestión, como la generación de listados, etiquetas, etc.

- Captura de actividad: Englobadas en el módulo de captura de actividad se encuentran las tareas de realización de las pruebas por parte de los técnicos, la generación y actualización de listas de trabajo, el registro de materiales y pluses utilizados, y la modificación de pruebas programadas

- Administración y configuración de la aplicación: Bajo el módulo de administración se configuran ciertas tablas “maestras” que Orion-RIS necesita para operar: codificaciones, secciones, prestaciones, usuarios, agendas, calendarios, etc.

- Informado: En el módulo de informado se agrupan las funcionalidades necesarias para la elaboración de informes por parte de los radiólogos.

Orion-RIS dispone de cuatro entornos de operación, con características y usos diferentes:

- **PRO:** El entorno de producción es el utilizado por los usuarios finales tanto radiólogos como profesionales de otros tipos. Este entorno está ubicado en cada uno de los hospitales que lo utilizan y únicamente es accesible dentro de la propia red del hospital.

- **PRE:** El entorno de preproducción es utilizado únicamente por la unidad de informática de los hospitales y por el equipo de desarrollo de la aplicación, únicamente desde la red de la Conselleria de Sanidad y no será público para ninguna integración.

- **TEST:** El entorno de TEST es utilizado tanto para desarrollar y probar las versiones en curso como para realizar la homologación o certificación de las posibles integraciones. Este entorno está ubicado en el centro de informática de la Conselleria de Sanidad y sólo es accesible dentro de la red corporativa.

- **DESA:** El entorno de desarrollo consta de servidores de aplicaciones en local, en las instalaciones de la empresa proveedora. Todos los demás recursos necesarios (bases de datos, motores de integración, etc) se encuentran en el centro de informática de la Conselleria de Sanidad, siendo únicamente accesibles mediante una red punto a punto entre la empresa proveedora y el centro de informática.

Ya que será en Orion-RIS donde deben llevarse a cabo las modificaciones para la integración con HSE, es necesario presentar con un poco más de detalle, tanto el entorno tecnológico de aquella, como las características funcionales del módulo de informado, que será el módulo afectado por dicha integración, y también brevemente el modelo de datos.

2.2.1 Descripción Funcional del Módulo de Informado

El usuario principal del módulo de Informado será el Médico Radiólogo.

En el acceso a dicho módulo, el radiólogo podrá seleccionar la exploración que desea informar, mediante el uso de diferentes criterios, siendo el más habitual la búsqueda del paciente por su número único de identificación SIP (Sistema de Información Poblacional) que lo identifica inequívocamente en todo el territorio regional o NHC (Número de Historia Clínica del paciente, que lo identifica inequívocamente en el propio hospital). Otro método, todavía más habitual, es la selección de la prueba de una bandeja de exploraciones pendientes de informar asignadas al radiólogo.

Una vez seleccionada la exploración, creará un nuevo informe de hallazgos, utilizando para ello un mecanismo de reconocimiento de voz integrado en el sistema (Speech Magic, de Nuance) o utilizando el teclado y con la ayuda de una serie de macros de texto configurables.

El radiólogo tiene a su disposición también herramientas para consultar otros informes realizados en el mismo centro, tanto por él mismo, como por sus colegas.

Como se puede apreciar en la Figura 2, la pantalla se encuentra dividida en dos zonas, la izquierda donde se encuentran los diferentes componentes y filtros de búsqueda, y la derecha, más espaciosa, en la que se obtiene el resultado de la búsqueda realizada y que permitirá aplicar las acciones sobre los resultados, ya sea imprimir o consultar informes ya realizados, firmar informes no definitivos, o crear nuevos informes.

Básicamente en esta zona se presentan tres clases de listas, el de prestaciones pendientes de informar, el de informes definitivos o provisionales y una combinación de ambos.

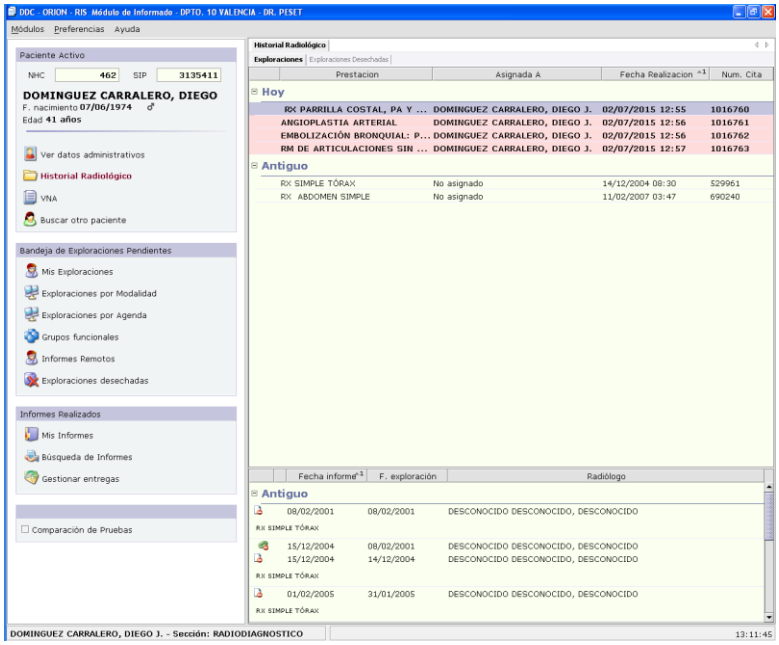

Figura 2: Módulo de Informado Orion-RIS

Sobre esta misma zona, pero en una nueva pestaña que se superpone a los listados, se abrirá, si así se desea, el editor de informes para su creación, tal y como se puede observar en la Figura 3.

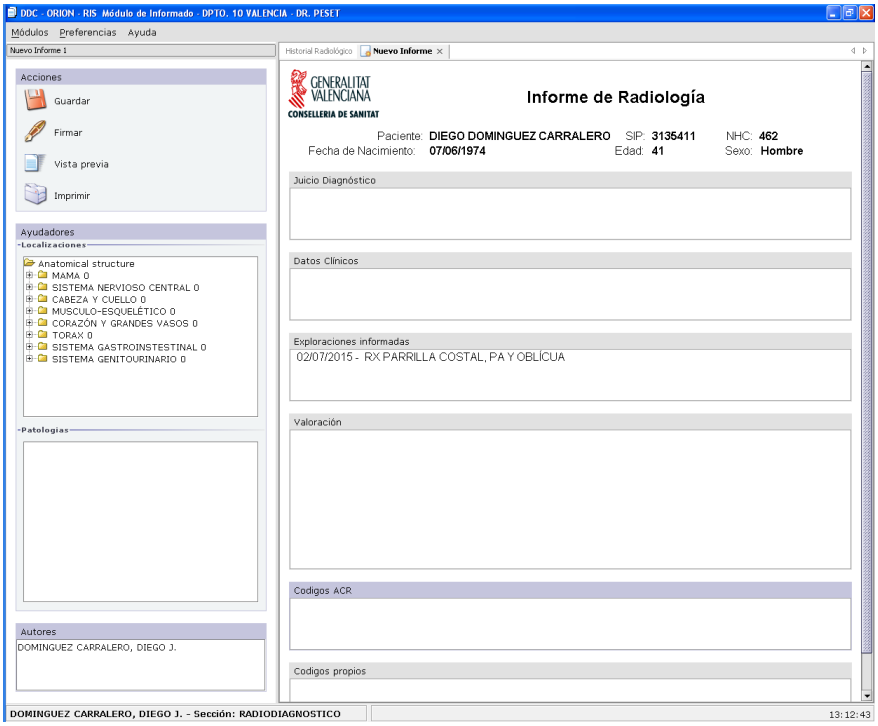

Figura 3: Editor de Informes Orion-RIS

Una vez realizada la firma digital, el informe se envía a un sistema de almacenamiento especial para imagen diagnóstica denominado PACS (Picture Archiving and Communication System, por sus siglas en inglés) dedicado al archivado digital de imágenes médicas y para la transmisión de las mismas a estaciones de visualización dedicadas, con monitores de definición ultra-alta o entre ellas a través de una red informática. Como tecnología sanitaria está regulada en Europa por la directiva 93/42/EEC.

2.2.2 Entorno Tecnológico

Orion-RIS es una aplicación J2EE de 3 capas, Presentación, Servicios y Persistencia, en la que se incluye la base de datos relacional, como se presenta en la figura 4.

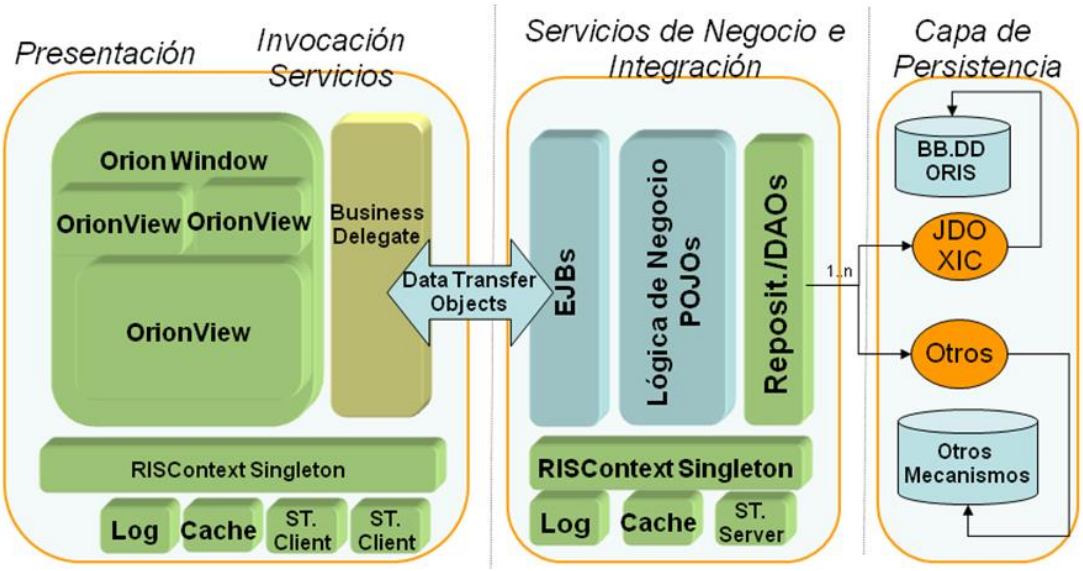

Figura 4: Arquitectura de capas de Orion-RIS

La parte de servicios se ejecuta sobre un Servidor de Aplicaciones IBM Websphere versión 7, aprovechando toda la infraestructura que nos brinda este: seguridad, transaccionalidad, escalabilidad, pool de conexiones, etc.

El cliente es una aplicación de escritorio Java, basado en Swing que invoca servicios implementados por EJBs sin estado. (EJB2.1)

Los servicios utilizan el modelo de dominio que representa las entidades y sus relaciones. Son POJOS que se persisten mediante una herramienta de mapeo objeto relacional (JDO) en la base de datos Informix.

Este modelo nos brinda una seguridad Http básica, gestionada por el contenedor, mediante usuario y contraseña.

Las integraciones entre Orion-RIS y otras aplicaciones, se realizan a través de un Servlet instanciado en el servidor de aplicaciones Webphere, que se accede exclusivamente a través del motor de integración corporativo de la Conselleria de Sanidad, que se presentará más adelante, en el apartado 2.3 Rhapsody.

Dicho motor de integración se encarga de encaminar peticiones desde/hacia sistemas externos, que en su mayoría utilizan una mensajería desarrollada por la propia Agencia Valenciana de Salud, denominada IDEAS. El motor de integración se encarga de convertir esos mensajes IDEAS a un lenguaje más actual y versátil, XML, y de realizar la invocación del servlet.

2.2.3 Modelo de Datos

El modelo de datos de Orion-RIS, convenientemente normalizado para evitar la redundancia de datos, disminuir problemas de actualización y proteger la integridad de los datos, es muy extenso y complejo, con más de 200 tablas, y además, se halla protegido por los derechos de autor de la Conselleria de Sanidad, por lo que no puede reproducirse en su totalidad.

A efectos de permitir la comprensión de la jerarquía necesaria, se presentan brevemente las tablas clave afectadas, imprescindibles para comprender el mecanismo de integración.

- **Pacientes –** Almacena todos los datos clave del paciente. Identificación por sus claves únicas (NHC y SIP), direcciones de contacto, etc.
- **Citas** – En la tabla citas se almacenan todas las visitas programadas al hospital de un paciente, en una fecha concreta, y para realizarse unas exploraciones determinadas.
- **Pruebas Citas** - En la tabla pruebas citas, se almacenan cada una de esas exploraciones a las que se hace referencia en el punto anterior.
- **Informes** – En la tabla informes se almacenan las valoraciones que los radiólogos realizan sobre las imágenes resultantes de las exploraciones realizadas. Un informe puede incluir exploraciones de una o varias citas y no tiene por qué cubrir todas las exploraciones de una.

Las relaciones entre las tablas se indican de modo esquemático a continuación, obviando las partes del modelo cuya relación con la integración no es directa:

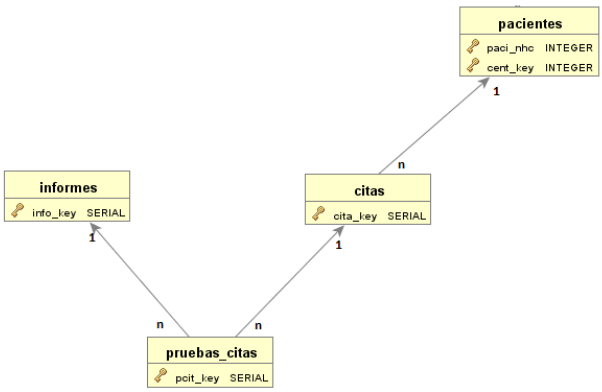

Figura 5: Relaciones entre las tablas implicadas

Cuando un paciente acude al hospital a que se le realicen pruebas de radiodiagnóstico, se considera que tiene una cita. La relación entre pacientes y citas es uno a muchos, un paciente puede tener múltiples citas, pero una cita sólo se le realiza a un paciente.

Las citas están compuestas por pruebas_citas en una relación también de una a muchas. Una cita contiene o puede contener varias pruebas citas, pero cada prueba cita es de una sola cita. Las pruebas citas son personalizaciones (por cita y paciente) de las prestaciones que se realizan en el hospital.

Los informes se crean para las pruebas_citas, en una relación de uno a muchas. Un informe puede aportar la valoración a varias pruebas_citas, sean de una o de varias citas diferentes, pero siempre del mismo paciente, pero una misma prueba no puede informarse en varios informes diferentes.

2.3 Rhapsody

A diferencia de las dos aplicaciones anteriores, el motor de integración Rhapsody es una aplicación comercial, propiedad de Orion Health (a pesar de la coincidencia en el nombre con Orion-RIS, no hay ninguna relación entre ellos) que se utiliza en la Conselleria de Sanidad para la gestión y el intercambio de mensajes entre las aplicaciones, bases de datos, y sistemas hospitalarios externos.

El motor de integración Rhapsody acepta mensajes en la mayoría de los formatos que se encuentran en un entorno sanitario. A continuación, estos mensajes se transforman y se transmiten a los destinos correctos con la garantía absoluta de que el mensaje se envía.

Las rutas son independientes entre sí, de manera que se podrá interrumpir una ruta en concreto, sin interrumpir el flujo de mensajes en las otras rutas.

Rhapsody archiva todos los mensajes en la base de datos y no los elimina hasta que haya recibido un mensaje que confirme la entrega de los mismos, permitiendo visualizar, modificar o reenviar los mensajes archivados posteriormente.

Rhapsody puede transmitir de manera selectiva mensajes de diferentes formatos a varios destinos, así como notificar a los administradores mediante diferentes métodos de comunicación en el supuesto de que suceda algún error.

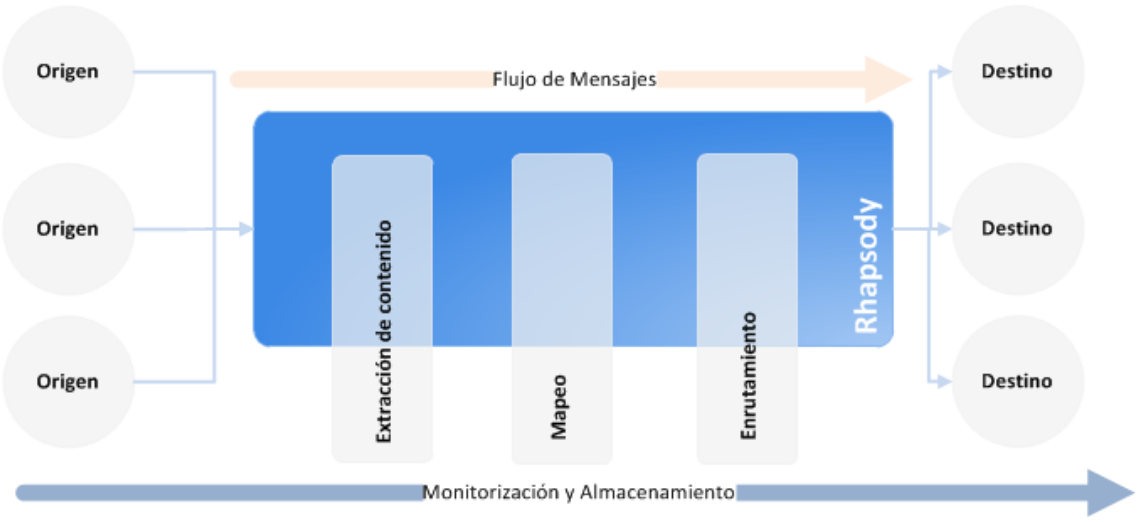

Figura 6: El bus de integración Rhapsody

3. Análisis de Requisitos

3.1 Introducción

Se describe en este apartado la especificación de requisitos del software (ERS) a desarrollar, según la estructura definida por el estándar IEEE 830-1998.

3.1.1 Propósito

Se pretende definir a continuación los requisitos funcionales y no funcionales de la integración, orientados al equipo de desarrollo, y que deberán establecer las bases para su implementación. Del mismo modo, esta definición servirá como acuerdo con la Conselleria de Sanitat para detallar la funcionalidad que se desea.

3.1.2 Ámbito

El objetivo de la integración es el diseño de un mecanismo que permita obtener los informes de un paciente que constan en la Historia de Salud Electrónica y presentarlos al usuario de la aplicación Orion-RIS, distinguiendo de algún modo los informes realizados en local, frente a los que son objeto de esta consulta, y por otro lado, la implementación de un mecanismo de donación de los informes creados en la base de datos de Orion-RIS.

3.1.3 Ácrónimos

Nombre	Descripción
RQxxx_YY	Requisito Funcional nº xxx, perteneciente al hito de entrega YY
CUxxxz_YY	Caso de uso nº z, perteneciente al requisito nº xxx, del hito de entrega YY
RNxxx	Requisito no Funcional nº xxx

Tabla 1: Acrónimos

3.1.4 Definiciones

Nombre	Descripción
Servicio WEB	tecnología que mediante un conjunto de estándares y protocolos, realiza un intercambio de datos entre aplicaciones.
SOAP	Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
XML	Lenguaje que organiza la información por medio de etiquetas.
SSL	protocolo criptográfico que proporciona comunicaciones seguras por una red.
X509	X509 es un estándar para infraestructuras de claves públicas.
API	Conjunto de rutinas y funciones ofrecidas por un sistema, en forma de librerías.

JAX-WS	API de Java para la creación de servicios web.
CXF	Framework completo, de código abierto para servicios web.
WS-Security	Protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web.
Quartz	framework Open Source, con licencia Apache 2.0, para la planificación de tareas en Java
Paciente	En la medicina y en general en las ciencias de la salud, el paciente es alguien que sufre dolor o malestar.
Radiólogo	Especialista en radiología, usuario de la aplicación Orion-RIS
Informe	Descripción, oral o escrita, de las características y circunstancias de un suceso o asunto, en este contexto se refiere a la descripción de los hallazgos de relevancia médica encontrados en el paciente.

Tabla 2: Definiciones

3.1.5 Referencias

Documentación interna y guías de estilo de Conselleria de Sanidad y del proyecto Orion-RIS.

IEEE 830-1998: Recommended Practice for Software Requirements Specifications.

3.1.6 Visión General de la Especificación de Requisitos

La presente especificación de requisitos se compone de tres secciones claramente diferenciadas:

- La sección actual (primera) contiene la introducción al apartado de especificación de requisitos.

- La segunda sección, que realiza una adaptación del estándar definido en la norma IEEE 830-1998, para adaptarlo a la guía de estilo de la Conselleria de Sanidad, y que presenta los requisitos en un contexto amplio y detallado.

- La tercera describe los requisitos con un elevado nivel de detalle, para que los desarrolladores puedan cumplir con las necesidades especificadas. Esta definición permitirá del mismo modo a los equipos de pruebas tener información suficiente para la realización de los test necesarios que garanticen el buen funcionamiento del producto.

3.2 Descripción General

3.2.1 Perspectiva

Orion-RIS es un producto independiente que necesita enviar y obtener información a través de servicios web ofrecidos por el sistema de información de la Historia de Salud Electrónica.

Desde unos puntos concretos de la aplicación, establecidos en el módulo de informado de la misma, se realizarán una serie de peticiones de datos, con parámetros tales como el paciente a consultar o el número de informe a visualizar.

3.2.2 Funcionalidades

Orion-RIS debe proporcionar a los usuarios que lo utilicen una información precisa sobre los informes de imagen de radiodiagnóstico disponibles para un determinado paciente.

El usuario (radiólogo) tiene dos puntos de contacto con la nueva integración: la solicitud de todos los informes que constan para un determinado paciente, que se ofrecerán en forma de lista, y la consulta de un informe concreto, cuya visualización se realizará como un PDF en una ventana emergente.

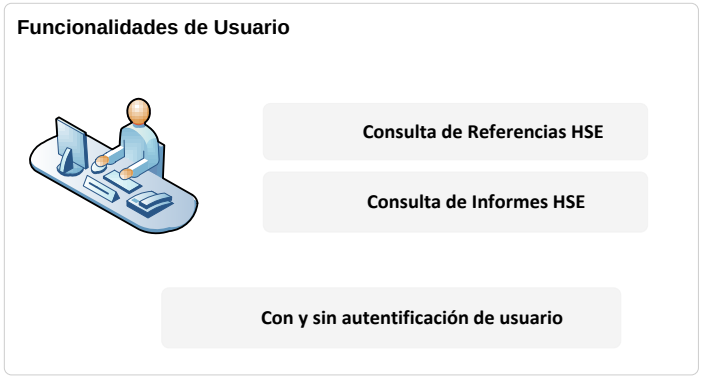

Figura 7: Funcionalidades de Usuario

Además la aplicación deberá ser capaz de enviar información a la historia de salud sobre los informes que se vayan realizando en el sistema de un modo autónomo.

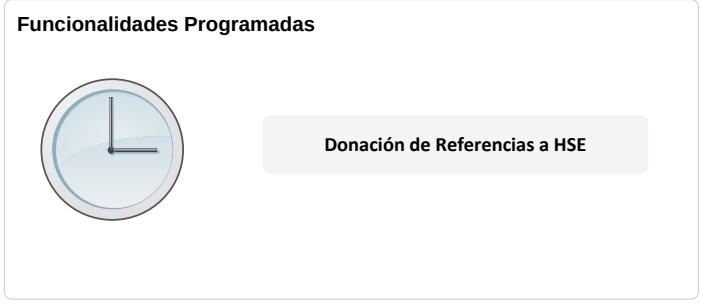

Figura 8: Funcionalidades Programadas

3.2.3 Usuarios

Sólo los usuarios de tipo radiólogo estarán autorizados y capacitados para el uso de las nuevas funcionalidades. Se adaptará para ello el módulo de Informado, con menús adaptados a las guías de estilo existentes, de modo que las modificaciones no interfieran en su trabajo diario, y que las modificaciones sean accesibles de un modo rápido y sencillo.

3.2.4 Restricciones

La funcionalidad requerirá de subida de versión de la aplicación Orion-RIS.

3.2.5 Suposiciones y Dependencias

Se asume el correcto funcionamiento de los servicios ofrecidos por la historia de salud electrónica, y de la red de comunicaciones de la Conselleria de Sanidad (Arterias).

Cualquier cambio en la especificación de los servicios, llamadas o parámetros específicos, deberá ser notificado, para su adaptación en el código de Orion-RIS por parte del equipo de desarrollo.

3.3 Requisitos específicos

Se detallan a continuación, con un nivel de detalle suficiente para las necesidades del equipo de desarrollo, los requisitos funcionales y no funcionales establecidos.

3.3.1 Requisitos de Interfaz de usuario

Las modificaciones deberán realizarse de modo que no supongan una ruptura ni en las líneas de diseño de la aplicación, ni en el modo de trabajo de este, por lo que el manejo de las opciones deberá ser intuitivo y sencillo.

Los controles se incluirán en el módulo de informado, concretamente en el componente destinado a la información del paciente, y los resultados se presentarán siguiendo la tónica de la bandeja de entrada de informes, respondiendo a los mismos métodos de entrada e interacción con el usuario, teclado y ratón, mediante click izquierdo para selección y ejecución de acciones y click derecho para menú contextual.

3.3.2 Requisitos Funcionales

- **Requisito RQ000401_H1:** Creación del entorno para la invocación segura de Servicios Web, bajo las directrices marcadas por la Agencia Valenciana de Salud.

- **Requisito RQ000402_H1:** Comprobación de la identidad del profesional solicitante vía tarjeta criptográfica.

- **Requisito RQ000403_H1:** Implementación de las invocaciones a los Servicios Web proporcionados por la Historia de Salud Electrónica:
    - **Caso de Uso CUF0004031_H1:** Visualización en Orion-RIS de las referencias a informes radiológicos por paciente disponibles en HSE.
    - **Caso de Uso CUF0004032_H1:** Visualización en Orion-RIS de un informe concreto de un paciente disponible en HSE.
    - **Caso de Uso CUF0004033_H1:** Donación de referencias de los informes creados en la instancia local de Orion-RIS.

3.3.3 Requisitos No Funcionales

- **Requisito RN000401:** El producto resultante debe ser robusto y fiable, teniendo en cuenta las posibles excepciones y su tratamiento.

3.4 Conclusiones

Se han definido en este capítulo requisitos, características y funciones que debe cumplir el producto. Se presentará a continuación, en el apartado siguiente “Diseño Técnico y Funcional”, una definición más detallada del sistema.

4. Diseño Técnico y Funcional

4.1 Requisito RQ000401_H1: Creación del entorno para la invocación segura de Servicios Web.

Un Servicio Web es una tecnología que mediante un conjunto de estándares y protocolos, realiza un intercambio de datos entre aplicaciones. Permite que aplicaciones diferentes, desarrolladas en lenguajes de programación distintos, y ejecutadas sobre cualquier plataforma, intercambien datos a través de una red de comunicaciones. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web.

Bajo las directrices de la Agencia Valenciana de Salud, toda comunicación entre los sistemas deberán realizarse mediante un canal de comunicación seguro HTTPS/SSL, y la autentificación de los sistemas deberá realizarse mediante la utilización de certificados digitales X.509 v3.

Se detallan a continuación las tecnologías y elementos necesarios para el establecimiento de esta invocación segura:

4.1.1 SOAP

Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C y es uno de los protocolos utilizados en los servicios Web.

Básicamente SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir.

Define un conjunto de reglas Conjunto de reglas de codificación para expresar instancias de tipos de datos y unas convenciones para representar llamadas a procedimientos y sus respuestas.

Este protocolo está basado en XML y se conforma de tres partes:

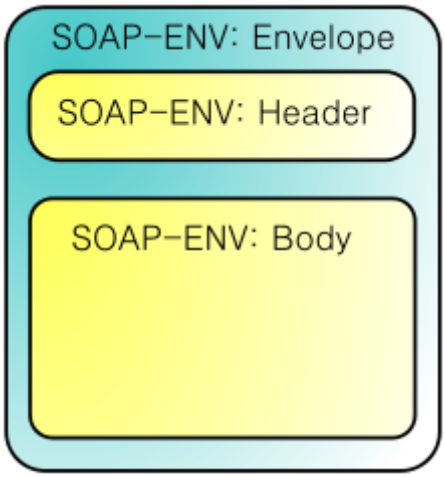

Figura 9 : Estructura de mensaje SOAP

- Envelope (obligatoria): raíz que de la estructura, es la parte que identifica al mensaje SOAP como tal.
- Header: esta parte es un mecanismo de extensión ya que permite enviar información relativa a como debe ser procesado el mensaje. Es una herramienta para que los mensajes puedan ser enviados de la forma más conveniente para las aplicaciones. El elemento "Header" se compone a su vez de "Header Blocks" que delimitan las unidades de información necesarias para el header.
- Body (obligatoria): contiene la información relativa a la llamada y la respuesta.
- Fault: bloque que contiene información relativa a errores que se hayan producido durante el procesado del mensaje y el envio desde el "SOAP Sender" hasta el "Ultimate SOAP Receiver"

El protocolo SOAP tiene tres características principales:

- Extensibilidad (se pueden utilizar extensiones, como por ejemplo las referentes a seguridad).
- Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS).
- Independencia (SOAP permite cualquier modelo de programación).

4.1.2 Secure Sockets Layer (SSL)

Para garantizar la seguridad, las comunicaciones entre sistemas se encriptarán mediante el protocolo SSL v3, garantizando la privacidad de la información y la exposición de la misma a terceros no autorizados.

Secure Sockets Layer (SSL; en español «capa de conexión segura») y su sucesor Transport Layer Security (TLS; en español «seguridad de la capa de transporte») son protocolos criptográficos que proporcionan comunicaciones seguras por una red.

Se SSL proporciona autenticación y privacidad de la información entre extremos mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación.
- Intercambio de claves públicas y autenticación basada en certificados digitales.
- Cifrado del tráfico.

Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar. Dentro de esta negociación, las partes intervinientes pueden establecer restricciones a los algoritmos que aceptarán en la comunicación, e incluso establecer un orden de preferencia entre los mismos, de forma que algoritmos considerados débiles no sean elegibles, o se dé prioridad a algoritmos con mejor rendimiento frente a otros con implementaciones más lentas.

4.1.3 X.509

X509 es un estándar para infraestructuras de claves públicas (en inglés, Public Key Infrastructure o PKI), publicado oficialmente en 1988. X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación. Asume un sistema jerárquico estricto de autoridades certificantes para emisión de certificados. Esto contrasta con modelos de redes de confianza, como PGP, donde cualquier nodo de la red puede firmar claves públicas, y avalar la validez de certificados de claves de otros.

Dichos certificados identificarán a cada uno de las aplicaciones que acceden al sistema (certificados de aplicación) y a los propios servidores (certificados de servidor).

4.1.4 Java API for XML Web Services (JAX-WS)

Java API for XML Web Services (JAX-WS) es una API de Java para la creación de servicios web. Es parte de la plataforma Java EE de Sun Microsystems.

La implementación de referencia de JAX-WS se desarrolla como un proyecto de código abierto y forma parte del proyecto GlassFish, un servidor de aplicaciones Java EE de código abierto.

Los distintos servidores de aplicaciones incluyen distintas implementaciones del estándar, en algunos casos opensource y en otros casos propietarias.

4.1.5 Apache CXF

Framework completo, de código abierto para servicios web. Se originó como combinación de dos proyectos de código abierto: Celtix desarrollado por IONA Technologies (adquirida por Progress Software en 2008) y XFire desarrollado por un equipo basado en Codehaus. Estos proyectos fueron combinados por personas que trabajaban juntas en Apache Software Foundation. El nombre CXF se deriva de la combinación de los nombres de proyecto "Celtix" y "XFire".

4.1.6 WS-Security

WS-Security (Seguridad en Servicios Web) es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web. En abril de 2004 el estándar WS-Security 1.0 fue publicado por Oasis-Open. En 2006 fue publicada la versión 1.1.

Originalmente desarrollado por IBM, Microsoft, y VeriSign, el protocolo es ahora llamado oficialmente WSS y está desarrollado por un comité en Oasis-Open.

El protocolo contiene especificaciones sobre cómo debe garantizarse la integridad y seguridad en mensajería de Servicios Web. El protocolo WSS incluye detalles en el uso de SAML y Kerberos, y formatos de certificado tales como X.509.

4.1.7 Certificados de Servidor

El entorno de Producción de Orion-RIS deberá disponer de certificados de servidor emitidos por la Agencia de Tecnología y Certificación Electrónica de la Comunidad Valenciana (ACCV).

La emisión de certificados de servidor se solicitará mediante ticket en el sistema de gestión de incidencias corporativo de la Agencia Valenciana de Salud (CA-Ticketing) o directamente a la ACCV siguiendo la guía http://www.accv.es/administracion-publica/certificados/servidor-con-soporte-ssl/

El certificado de servidor se utilizará para el establecimiento de la conexión segura entre la instancia de Orion-RIS y el servidor central de HSE.

Los certificados de servidor son fácilmente reconocibles porque se identifican con un icono distintivo en los navegadores. Por ejemplo, en la siguiente imagen se puede ver como el Navegador Chrome, actuando como un cliente de HTTP, ha aceptado el certificado con nombre "hseavstest.san.gva.es" porque ha podido verificar su identidad al haber sido emitido por la entidad intermedia ACCVCA-120.

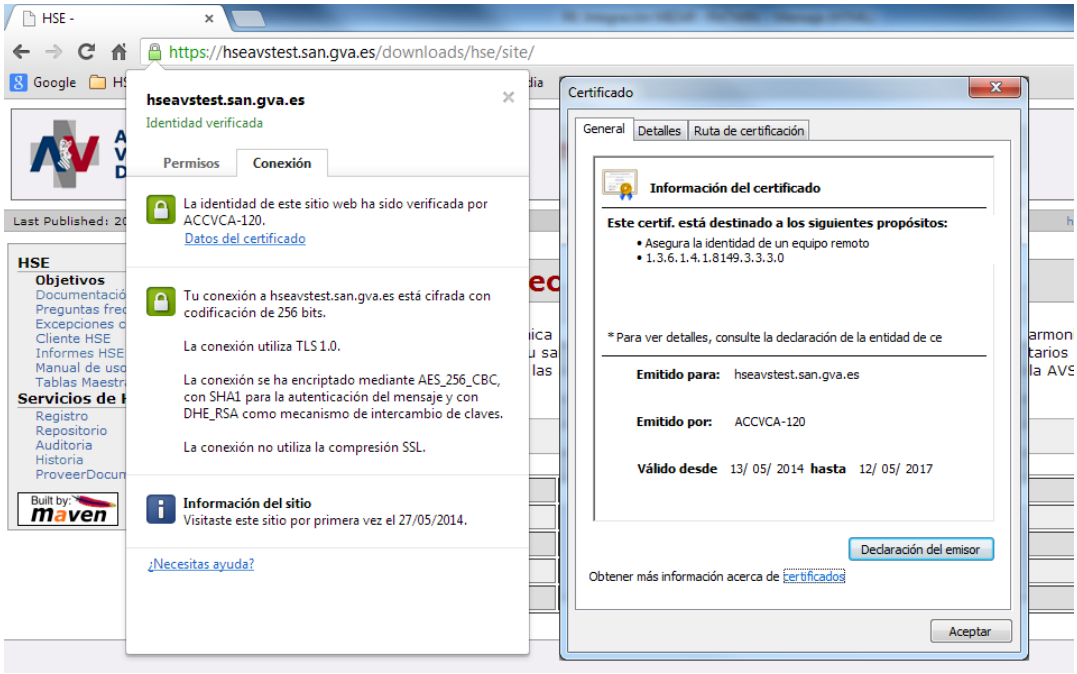

Figura 10: Comprobación de certificados

Cuando en lugar de utilizar un navegador para establecer una conexión HTTPS, se establece la conexión desde una aplicación asistencial, como es nuestro caso en la integración con Orion-RIS, en la negociación del canal SSL para establecer la conexión será necesario que la aplicación asistencial confíe en el certificado de servidor o en sus autoridades intermedias para poder establecer la conexión.

Para ello se deberá configurar en la aplicación asistencial el almacén de certificados de confianza (Truststore) para añadir los certificados de las autoridades intermedias y raíz.

En tecnología Java este almacén de certificados está ubicado por defecto en $JAVA_HOME/lib/security/cacerts aunque se puede personalizar, con el contenido de la propiedad "javax.net.ssl.trustStrore".

4.1.8 Certificados de Aplicación

En un entorno como el sanitario, en el que se producen sinergias e integraciones entre muchas aplicaciones (sólo en un ámbito intrahospitalario pueden coexistir ya más de 100 aplicaciones) resulta de vital importancia poder identificar de manera segura quién es quién en dicha integración. Para este propósito utilizaremos los certificados de aplicación.

Será necesario disponer de certificados digitales de aplicación distintos para cada instancia y cada entorno de Orion-RIS (PRO, PRE y TEST de cada departamento), que unidos a los necesarios para los entornos de PRE y TEST centrales, sumarán aproximadamente un centenar.

Dichos certificados de aplicación deberán solicitarse mediante ticket en CA, o directamente a la ACCV, siguiendo la guía http://www.accv.es/administracion-publica/certificados/de-aplicacion/

A continuación se presentan un par de ejemplos de certificados existentes, mediante su atributo SUBJECT que identifica unívocamente al certificado de aplicación y que deberá especificarse en la solicitud a la ACCV de los nuevos certificados:

- C=ES, O=CONSELLERIA SANITAT GVA, OU=Aplicaciones, CN=HISTORIA DE SALUD ELECTRONICA TEST - CONSELLERIA DE SANITAT GVA : Identifica a la aplicación HSE del entorno de TEST.

- C=ES, O=Generalitat Valenciana, OU=Aplicaciones, CN=PORTAL HISTORIA DE SALUD ELECTRONICA - CONSELLERIA DE SANIDAD: Identifica a la aplicación PortalHSE del entorno de PRO.

- C=ES, O=Generalitat Valenciana, OU=Aplicaciones, CN=ORION RIS PRE - HOSPITAL LA FE CONSELLERIA DE SANIDAD: Identifica a la aplicación Orion-RIS de la FE del entorno de PRE.

El certificado de aplicación deberá almacenarse en un almacén de certificados, que será utilizado por Orion-RIS, disponiendo en este punto de:

- La ruta completa donde se encuentra el almacén de certificados: en tecnología java el almacén de certificados tiene la extensión “jks”. En nuestra integración concreta, el almacén de certificados se denominará orionris.jks

- Contraseña para acceder a los certificados dentro del almacén.
- Alias del certificado dentro del almacén: Un almacén de certificados permite albergar más de un certificado digital y para poder identificar un certificado concreto dentro del almacén será necesario identificarlo mediante el alias.
- Contraseña del certificado: Esta contraseña deberá ser distinta a la del almacén de certificados.

4.2 Requisito RQ000402_H1: Comprobación de la identidad del profesional solicitante vía tarjeta criptográfica.

HSE tiene una política de seguridad de acceso a sus servicios Web basada en asignación de permisos a certificado digital de aplicación.

Sin embargo, la asignación de permisos por servicio (radiodiagnóstico, cardiología, neumología, pediatría, etc), categoría profesional (médico adjunto, médico residente, enfermera, técnico auxiliar, etc.) u otro criterio recae sobre Orion-RIS, que es quien realiza la identificación del usuario final.

De este modo, se establece una relación de confianza entre HSE y Orion-RIS mediante la cual se delega el control de acceso a esta última, que vela por que no existan accesos indebidos por parte de sus usuarios finales a la información que se proporciona en HSE, identificando para ello a sus usuarios mediante certificado digital de persona física de la ACCV, la Fabrica Nacional de Moneda y Timbre o la Dirección General de Policía para aquellos lugares de la aplicación donde se invoquen los servicios web proporcionados por HSE.

La pulsación del enlace de visualización de informes de ámbito nacional en la Historia de Salud Electrónica, deberá desencadenar la comprobación de la identidad del solicitante, mediante una ventana emergente para la consulta de su tarjeta criptográfica y pin, como se muestra en la Figura 6.

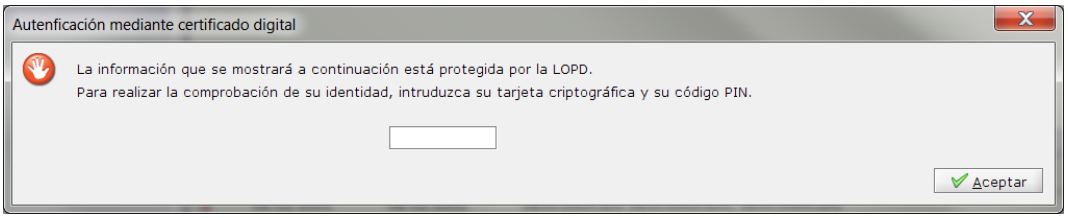

Figura 11: Validación de tarjeta criptográfica

Una vez validado, a través de las librerías de firma digital proporcionadas por la ACCV y cuyo uso ya está implementado en Orion-RIS para la realización de la firma digital de informes, el usuario se enviará como un parámetro en la invocación a los Servicios Web para garantizar la identidad del peticionario.

4.3 Requisito RQ000403_H1: Implementación de las invocaciones a los Servicios Web proporcionados por la Historia de Salud Electrónica.

4.3.1 Caso de Uso CUF0004031_H1: Visualización en Orion-RIS de las referencias a informes radiológicos por paciente disponibles en HSE.

En el módulo de Informado añadiremos en el panel del paciente dos nuevas opciones, que nos permitirán realizar la invocación segura del servicio web proporcionado por HSE obtenerReferenciasResultadosImagen, bien con alcance autonómico (sólo consultará los informes realizados en la comunidad valenciana) o con alcance global (informes realizados en todo el territorio nacional).

En la Figura 9 se presenta el interfaz tras la aplicación de las modificaciones.

Figura 12: Interfaz con las nuevas opciones de consulta en HSE

Los informes recuperados con esta consulta se visualizaran en una bandeja de informes equivalente a la bandeja de Informes normal, pero que constará de la siguiente estructura:

Figura 13: Interfaz con los campos de los informes de HSE

Fecha de Realización del informe

Centro de Realización del informe.

Radiólogo Responsable del informe.

Prestación/prestaciones informadas.

Indicador de informe oculto.

Los informes recuperados serán de todos los centros, exceptuando los del propio, ya que estos constan en el historial radiológico local del paciente.

La última columna contendrá un icono distintivo () para aquellos informes que el paciente ha decidido que sean ocultados de su Historia de Salud.

En concreto es necesaria la implementación de la operación especificada a continuación, y cuya documentación completa puede consultarse en el site https://hseavstest.san.gva.es

- ConsultaService.obtenerReferenciasResultadosImagen, que obtendrá todas las referencias de un paciente que constan en HSE.

4.3.2 Caso de Uso CUF0004032_H1: Visualización en Orion-RIS de un informe concreto de un paciente disponible en HSE.

Haciendo click derecho sobre cualquiera de las referencias de informes recuperados, se abre un menú contextual con una única opción por el momento, “Abrir informe”, que realiza la invocación al servicio obtenerInformeResultadosImagen.

Si por el contrario se pulsa sobre el icono que indica que el paciente ha decidido ocultar un informe de su historia, dicho informe podrá visualizarse, aunque para ello será necesaria la aceptación consciente de una petición de visualización de informes ocultos (Figura 14). Este acceso quedará registrado por motivos legales.

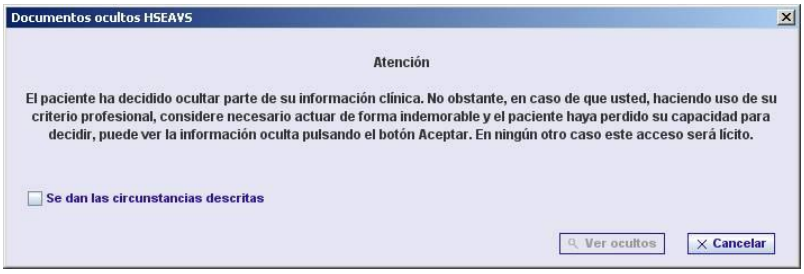

Figura 14: Aceptación de visualización de Informes Ocultos

Con la pulsación en el botón de ver los ocultos se invocará al servicio web obtenerInformeResultadosImagen con un parámetro adicional que indica que se muestre el documento oculto.

La visualización del informe se llevará a cabo en una nueva ventana emergente que mostrará el contenido del informe en PDF.

En concreto es necesaria la implementación de la operación especificada a continuación, y cuya documentación completa puede consultarse en el site https://hseavstest.san.gva.es

- InformesService.obtenerInformeResultadosImagen, que obtendrá el detalle de la referencia que especifiquemos.

4.3.3 Caso de Uso CUF0004033_H1: Donación de referencias de los informes creados en la instancia local de Orion-RIS.

Actualmente es el motor de integración, Rhapsody, quien intercepta los informes que navegan desde el RIS al PACS e inyecta dicho informe en forma de alta a HSE.

Bajo el paradigma futuro de la eliminación de Rhapsody, es necesario que Orion-RIS realice esta operación directamente, mediante un proceso que incluye la invocación de un WebService ofrecido por HSE, RegistroService, con la operación crearReferencia.

Esta donación se realizará de modo desatendido y desincronizado de la firma de informes, para evitar la saturación de las comunicaciones. Mediante un cron o proceso similar, se enviarán las donaciones de referencias que consten como no enviadas. Se utilizará un mecanismo para controlar los informes donados, o los que han provocado error, que podrán reenviarse periódicamente.

Como se muestra ver en la Tabla 3, el modelo de datos de la tabla de informes en Orion-RIS se ampliará con dos nuevos campos:

Info_key	...	HSE_Fallos_Entrega	HSE_Entregado
info_key_1	...	3	FALSE
info_key_2	...		TRUE
info_key_3	...	2	TRUE

Tabla 3: Campos añadidos al modelo de informes

El campo de fallos en la entrega servirá para mantener un contador que nos permita alertar de que un informe concreto no se puede enviar, para que se revise de modo manual, pudiendo en el futuro incluirse un indicador relacionado en el cuadro de mando de la aplicación.

El campo de entregado nos servirá para dejar de enviar los informes ya entregados.

La lógica del proceso de donación de referencias será la siguiente:

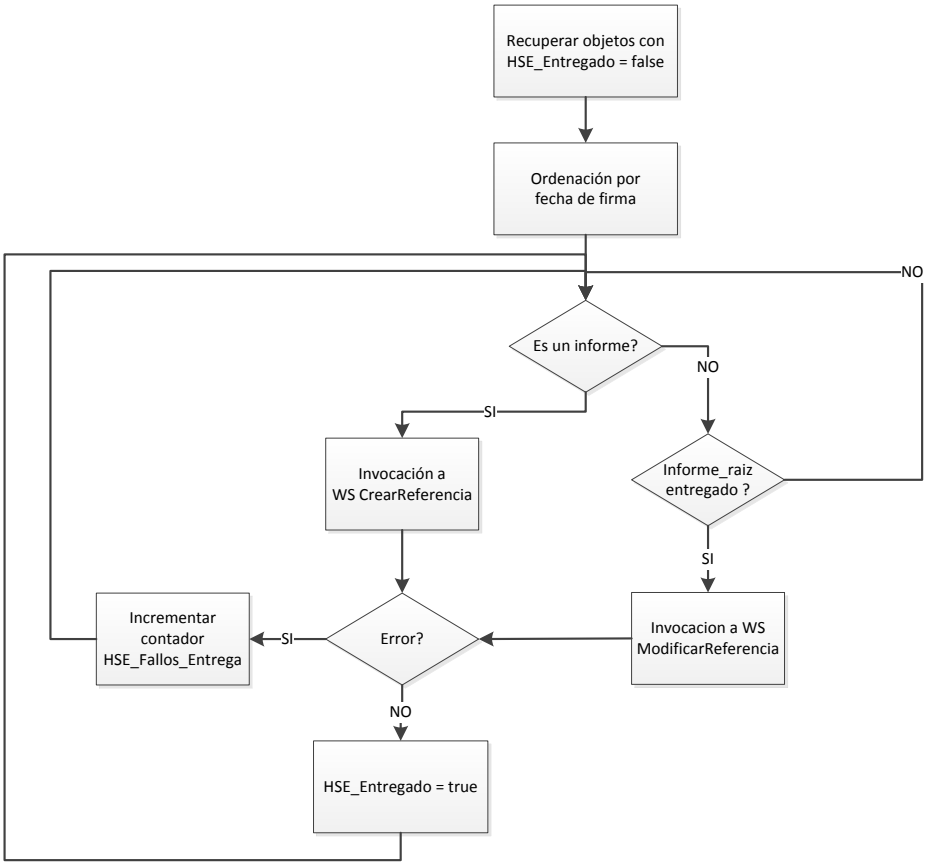

Figura 15: Diagrama de flujo del proceso de donación de referencias

Con el fin de separar la operación normal de Orion-RIS de las donaciones de referencias de informes a la Historia de Salud Electrónica, y no provocar sobrecarga en la red de comunicaciones en los periodos de máxima actividad, es necesario construir un mecanismo de envío desatendido en background, con planificación configurable en función de las ventanas de servicio asignadas para los procesos por parte de los administradores de sistemas de la Agencia Valenciana de Salud.

Para la construcción del citado mecanismo de envío se opta por el empleo de Quartz, framework de planificación de tareas que ya se encuentra en uso en otras aplicaciones desarrolladas para Conselleria de Sanidad.

Es un framework Open Source, con licencia Apache 2.0, para la planificación de tareas en Java, usado en importantes y conocidos proyectos como JBoss, Cocoon, Apache Jakarta, etc.

Es compatible con proyectos J2SE y J2EE, y permite la planificación flexible de tareas incluso mediante un fichero de configuración externo, lo que resulta decisivo para su elección como framework para este mecanismo. Quartz mantiene el estado de las tareas programadas, incluso en caso de fallos y reinicio de servidores.

Se basa en dos interfaces principales:

org.quartz.Job

Definir una tarea es tan sencillo como implementar la interface org.quartz.Job.

Simplemente es necesaria la implementación de un método y el lanzamiento de una excepción en caso de error para que Quarz reintente su ejecución o no, en función de la configuración especificada.

org.quartz.Trigger

Es una clase abstracta que define los instantes en que la tarea debe ser ejecutada. Las diferentes clases que implementan esta interfaz definen el modo en el que se configurará el lanzamiento de la tarea, programático, manual, mediante fichero de configuración externo, etc.

5. Detalles de Implementación

5.1 Requisito RQ000401_H1: Creación del entorno para la invocación segura de Servicios Web.

El servidor de aplicaciones provee muchos mecanismos que hacen de la configuración de seguridad un proceso muy sencillo, mediante la aplicación de soluciones propietarias. Sin embargo, y en aras de la futura compatibilidad en este momento en el que las licencias de Software están en entredicho, se opta por la configuración de un modo más artesanal, con el proceso que se describe a continuación:

5.1.1 Configurar la seguridad "Unrestricted SDK JCE policy files" en la instalación de Java.

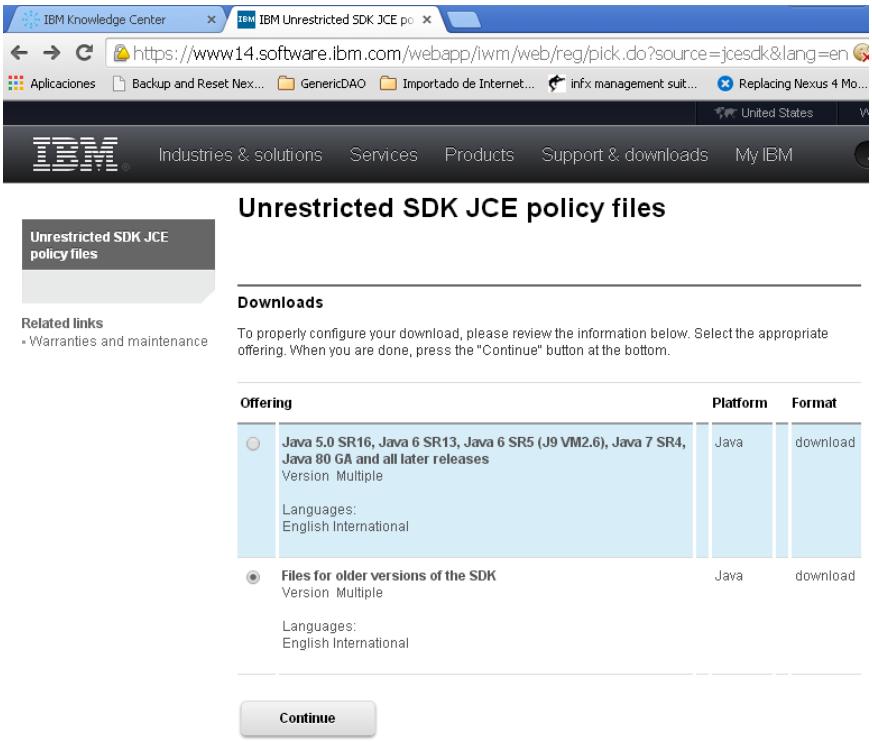

Figura 16: Descarga de las políticas de seguridad

El JDK estándar viene configurado por defecto con unas políticas de seguridad que contienen ciertas restricciones sobre el tamaño de las claves que pueden usarse, generalmente limitadas a 128 bits. Sin embargo, la política de seguridad de la Agencia Valenciana de Salud, obliga a que las claves sean de 2048 bits, por lo que es necesario

actualizar las políticas usadas en la máquina virtual que ejecuta el servidor de aplicaciones. El modo más sencillo de hacerlo es, tal y como puede observarse en la Figura 16, descargar el fichero correspondiente a la versión de Java que soporta el Websphere Application Server en ejecución.

Dicha descarga se realizará desde la página Web de IBM: https://www14.software.ibm.com/webapp/iwm/web/reg/pick.do?source=jcesdk&lang=en_US

Tras ello, deberá copiarse y descomprimirse el archivo obtenido, en la ruta de la máquina virtual Java que ejecuta el servidor de aplicaciones las librerías Java:

```
/opt/IBM/WebSphere7/AppServer/java/lib
```

Figura 17: Ruta de la máquina virtual Java de Websphere Application Server

Cambiar el propietario y asignarle permisos:

```
chown *.* websphere:websphere

chmod +x local_policy.jar

chmod +x US_export_policy.jar
```

Figura 18: Asignación de permisos a los archivos de la Java Policy

Una vez reiniciado el servidor de aplicaciones, este estará en disposición de aceptar la criptografía que se especifica por parte de la Agencia Valenciana de Salud.

5.1.2 Importación del certificado de aplicación en el almacén de certificados OrionRIS.jks proporcionado por Soporte Orion-RIS

Para que las unidades de informática puedan realizar el despliegue de la versión de Orion-RIS que provee la integración con la Historia de Salud Electrónica, previamente deben disponer del certificado de aplicación, cuya solicitud deben realizar a la ACCV.

Junto con el certificado llegará un password que habrá que introducir en la base de datos, en la tabla de configuración_instancia de la aplicación, tabla que recoge valores de configuración propios de cada entorno.

Será necesario importar el certificado privateKeyFile.p12, que será el proporcionado por la ACCV en el almacenORIS.jks proporcionado por el soporte de la aplicación, mediante el comando:

```
keytool -importkeystore -srckeystore privateKeyFile.p12 -srcstoretype PKCS12 -
destkeystore almacenORIS.jks
```

Figura 19: importación del certificado de aplicación en el almacén

La ejecución de este comando requerirá tanto la contraseña del almacén de certificados, que será proporcionada por el equipo de soporte de la aplicación, como la contraseña del propio certificado, que será proporcionada por la ACCV.

5.1.3 Configuración del certificado de Servidor para el establecimiento de las conexiones SSL.

Ya que el Servidor de aplicaciones se dedica en exclusiva a Orion-RIS, se establecerán las configuraciones utilizando los almacenes por defecto. En otro caso, en que el servidor de aplicaciones diese servicio a aplicaciones diferentes, sería necesaria la creación de almacenes propios, a nivel tanto de Nodo como de Cell.

En primer lugar, en la configuración de certificados SSL y gestión de claves, en el Almacén por defecto del nodo, debe importarse el certificado de servidor, generado por la ACCV para cada Servidor de Orion-RIS, como se muestra en la Figura 20.

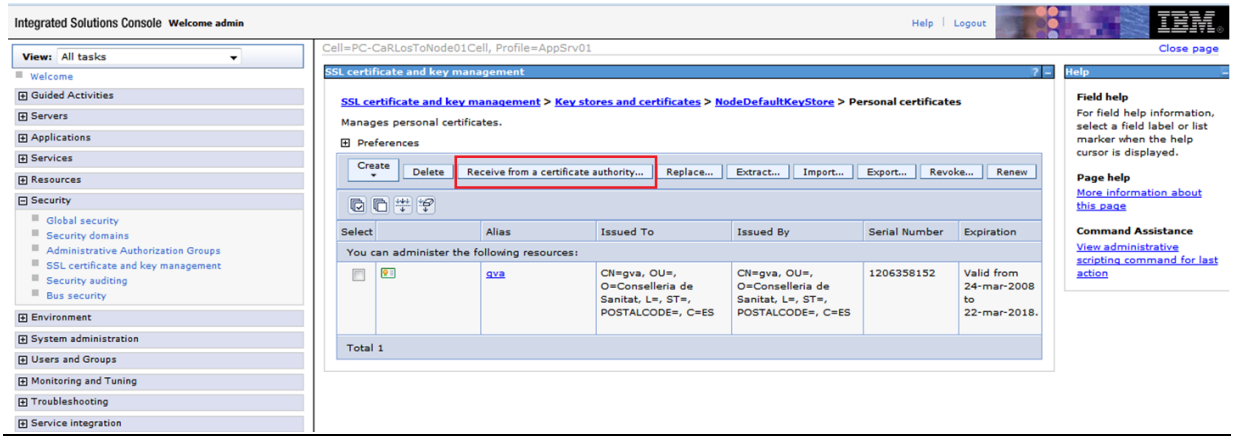

Figura 20: Importación del certificado de servidor

Una vez importado dicho certificado, se asignará a todas las comunicaciones salientes que se realicen por el protocolo HTTP, como se muestra en las figuras 21 y 22.

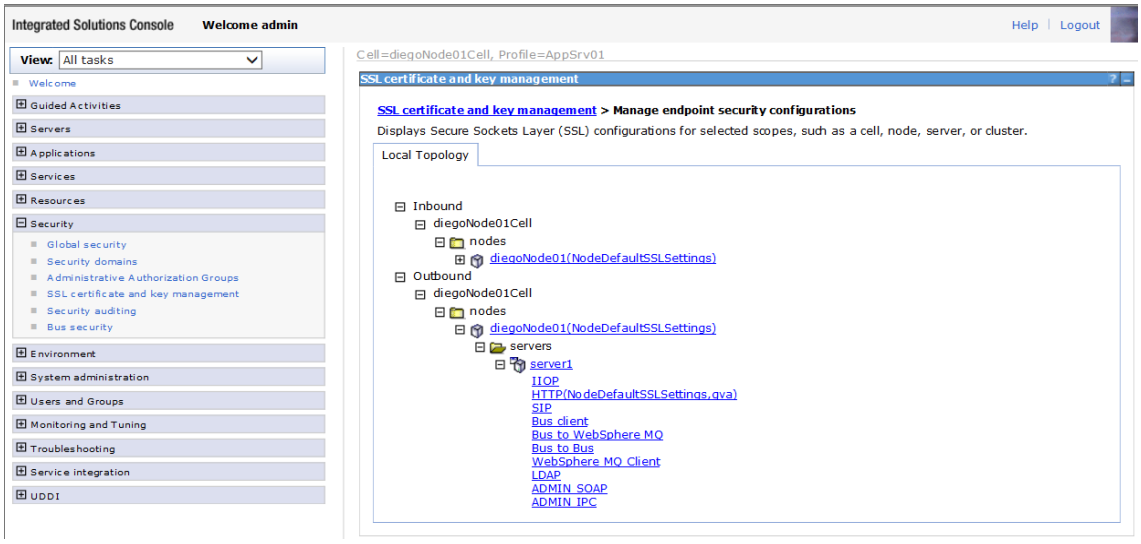

Figura 21: Gestión de los canales de entrada y salida

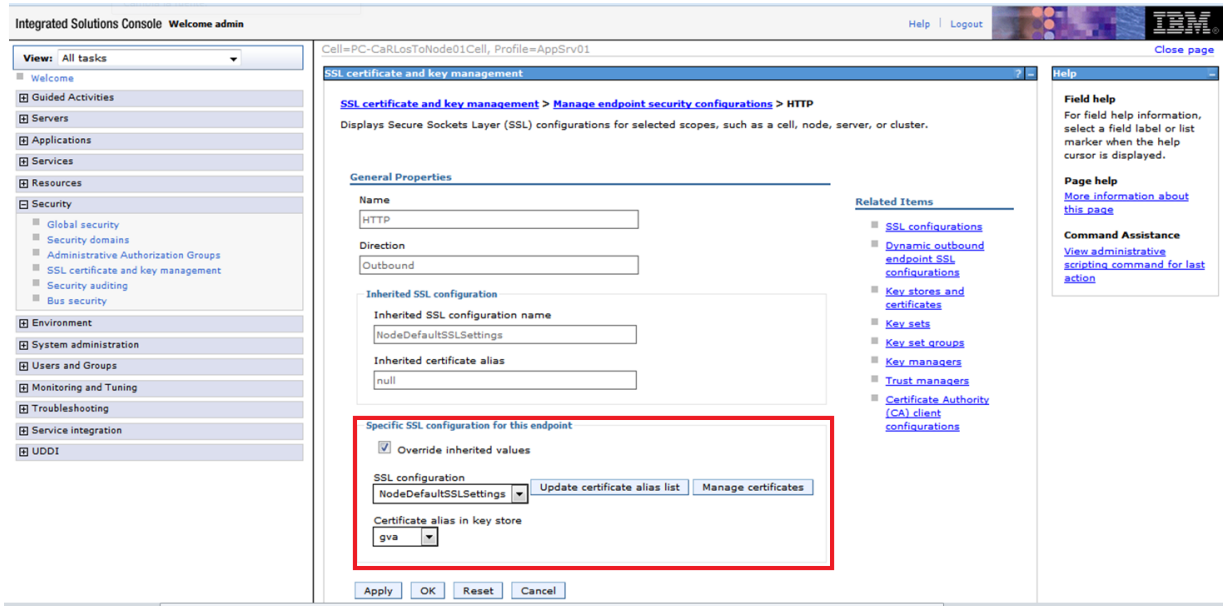

Figura 22: Asignación del certificado a punto de salida HTTP

5.1.4 Configuración de las autoridades certificadoras de confianza

Para poder utilizar los certificados digitales, ya sea en un navegador web o en un servidor de aplicaciones, deben registrarse en el sistema correspondiente las claves públicas de los certificados digitales de la Agencia de Tecnología y Certificación Electrónica (ACCV). Son necesarias para verificar que el certificado digital que se va a utilizar ha sido emitido por una Autoridad de Certificación en la que se confía.

Para permitir validar los certificados emitidos por la ACCV, tanto de HSE como de Orion-RIS es necesaria la importación de los certificados intermedios de la autoridades confianza que permiten la comprobación la validez de la emisión de un certificado. Estos certificados se descargan del sitio web de la ACCV (http://www.accv.es/ayuda/descargar-certificados-digitales/).

El proceso requiere la importación de todas los certificados de la cadena de certificación de la ACCV: **rootca.crt, accv-ca1.crt, accv-ca2.crt, accvraiz1.cer, accvca110.cer y accvca120.cer** en el Almacén de certificados de confianza (NodeDefaultTrustStore) por defecto del Nodo del Servidor de Aplicaciones Websphere, ya que este último es exclusivo para Orion-RIS. De otro modo, sería recomendable la creación de un almacén propio de certificados por aplicación o seguir manteniendo un almacén único, en función de los requisitos de las aplicaciones que deban convivir.

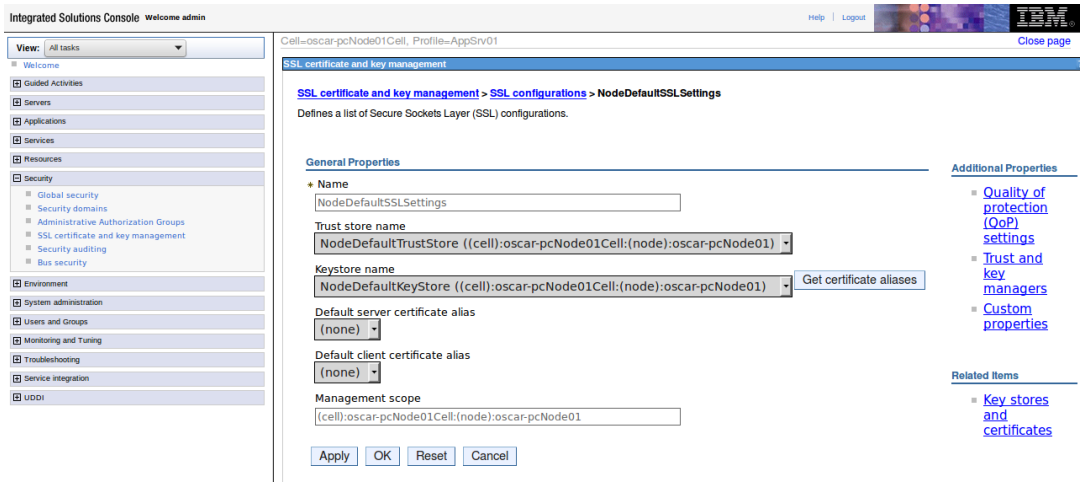

Figura 23: Configuración SSL de NodeDefaultTrustStore

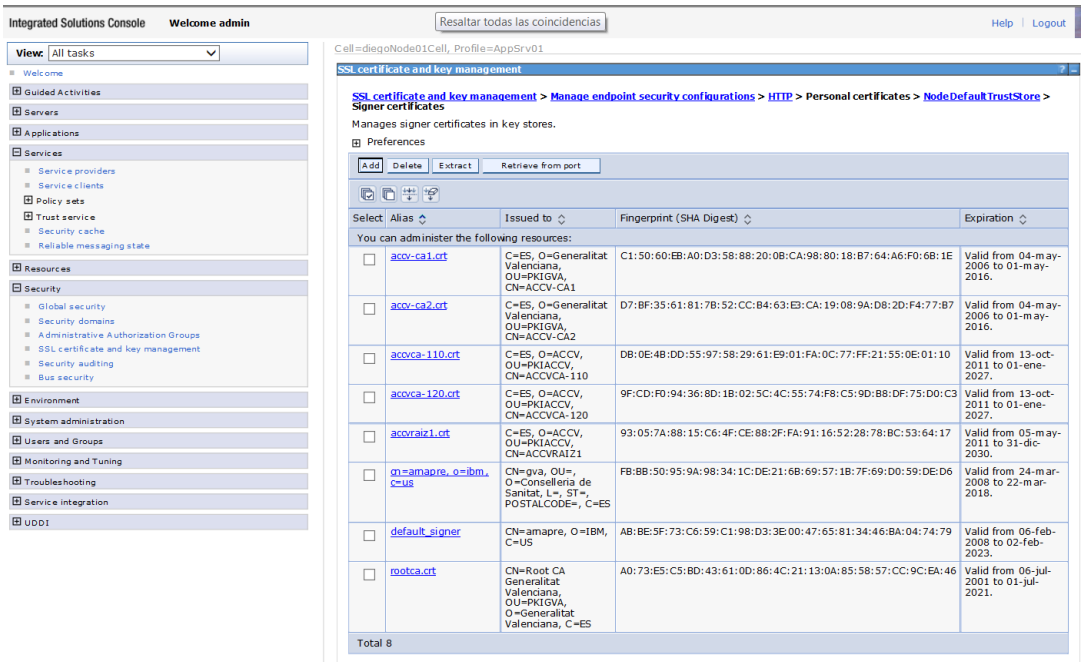

Figura 24: Importación de certificados de confianza en el NodeDefaultTrustStore

5.2 Requisito RQ000403_H1: Implementación de las invocaciones a los Servicios Web proporcionados por la Historia de Salud Electrónica.

5.2.1 CUF0004031_H1, CUF0004032_H1 y CUF0004033_H1 Implementación de código.

Se detalla a continuación, a modo de ejemplo, la implementación en código del método construido para la invocación del Servicio Web crearReferencia, cuyo objetivo es la donación a HSE del número de referencia del informe, junto con datos asociados al mismo.

El resto de métodos construidos para la invocación del resto de servicios seguirán el mismo patrón, por lo que no se detallará su código.

Se omiten, por motivos de confidencialidad y propiedad intelectual del código fuente, que pertenece a Conselleria de Sanidad, fragmentos de código completos, que no tienen relación con la integración que se describe.

```
Public Boolean crearReferencia(String informeId) {

log.debug("Inicio crearReferencia. codigo informe:>" + informeId + "<");

// Creación del proxy para invocar al WS
JaxWsProxyFactoryBean proxy = new JaxWsProxyFactoryBean();

try {
            …

// Recojo de un properties que se carga de una tabla de configuración en base de datos el EndPoint del WebService
String urlHSE = Application.getInstance().getInstanciaOrionProperties().getProperty("hse_registro_service");

// Y el Alias del certificado a utilizar
String aliasKeystore = Application.getInstance().getInstanciaOrionProperties().getProperty("hse_alias");


// Asigna el endpoint obtenido
proxy.setAddress(urlHSE);

// Le indicamos la interfaz a que responde el WS que consumiremos
proxy.setServiceClass(es.gva.san.orion.server.hse.registroService.RegistroService.class);

// Una vez finalizada la configuración del proxy, se instancia el mismo
RegistroService registroCliente = (RegistroService) proxy.create();

// Para la configuración de seguridad discriminamos entre nuestro entorno
```

```
//local de desarrollo, en Windows, y el entorno final de ejecución
if (System.getProperty("os.name").startsWith("Windows")) {
	pathProperties = "es/gva/san/orion/server/hse/securityRIS-
DESA.properties";
	log.debug("asigno el properties securityRIS-DESA");
} else {
	pathProperties = "es/gva/san/orion/server/hse/securityRIS.properties";
	log.debug("asigno el properties securityRIS");
}


// Parámetros de configuración de seguridad del WSS4JInInterceptor
Map<String, Object> properties = new HashMap<String, Object>();
properties.put(WSHandlerConstants.ACTION, "Timestamp Signature");
properties.put("signaturePropFile", pathProperties);

// Instanciación del WSS4JInInterceptor que efectúa la firma y el timestamp
//de los mensajes de salida
WSS4JInInterceptor inInterceptor = new WSS4JInInterceptor(properties);

// Se instancia el cliente del WS, y se configura para que emplee https
Client cliente = ClientProxy.getClient(registroCliente);
HTTPConduit http = (HTTPConduit) cliente.getConduit();
TLSClientParameters params = new TLSClientParameters();
params.setUseHttpsURLConnectionDefaultSslSocketFactory(true);
http.setTlsClientParameters(params);

// Se asigna el WSS4JInInterceptor al cliente creado
Endpoint endpoint = cliente.getEndpoint();
endpoint.getInInterceptors().add((Interceptor<? extends Message>)
inInterceptor);

// Parámetros de configuración de seguridad del WSS4JOutInterceptor
properties = new HashMap<String, Object>();
properties.put(WSHandlerConstants.ACTION, "Timestamp Signature");
properties.put(WSHandlerConstants.USER, aliasKeystore);
properties.put(WSHandlerConstants.SIG_PROP_FILE, pathProperties);
properties.put(WSHandlerConstants.PW_CALLBACK_CLASS,
	"es.gva.san.orion.server.hse.callback.PasswordCallback");
properties.put(WSHandlerConstants.SIG_KEY_ID, "DirectReference");


// Instanciación del WSS4JOutInterceptor que comprueba la firma y el
//timestamp de los mensajes
WSS4JOutInterceptor outInterceptor = new WSS4JOutInterceptor(properties);
endpoint.getOutInterceptors().add((Interceptor<? extends Message>)
outInterceptor);

// Preparamos una peticion para el servicio
CrearReferenciaVO crearReferenciaVO = new CrearReferenciaVO();

// Construcción del objeto CrearReferenciaVO, que es el objeto intercambiado
		…


//Invocacion del WebService
log.debug("invoco al servicio de crearReferencia");

ResultadoCreacionVO resultado =
```

```
registroCliente.crearReferencia(crearReferenciaVO);

log.debug("Fin crearReferencia. referencia:>"+
resultado.getCodigoReferencia() + "<");

// Guardo la referencia
informe.setReferenciaHSE(resultado.getCodigoReferencia());
} catch (HSEException_Exception ex) {
      // Tratamiento de las excepciones

            …
}
```

Figura 25: Código ejemplo de la invocación de Servicio Web

5.2.2 CUF0004033_H1 Implementación de la donación programada

Se detalla a continuación, a modo de ejemplo, la implementación en código de los procedimientos de Quartz, cuyo objetivo es la donación a HSE del número de referencia del informe, mediante la invocación del método creado en el punto anterior, que a su vez invoca al servicio web asociado.

```
// implementa la interfaz org.quartz.Job
public class CreaReferenciaJob implements org.quartz.Job {

    /** The logger. */
    private static Logger logger = Logger.getLogger(CreaReferenciaJob.class);

// método execute (de la interfaz) y propagación de la excepción asociada
    public void execute(JobExecutionContext arg0) throws
JobExecutionException {

      try {
          logger.debug("Inicio CreaReferenciaJob execute");

                  …
// invocación al método crearReferencia construído en el punto anterior
// que a su vez, invoca al servicio web correspondiente
             iservice.crearReferencia(cod);
             logger.debug("Fin CreaReferenciaJob execute");
          }
      } catch (Exception e) {
          logger.debug("Error CreaReferenciaJob execute: >" + e.getMessage()
                + "<");
          e.printStackTrace();
      }
    }
}
```

Figura 26: Implementación de una tarea de Quartz

```
// Planificador de las tareas
public class PlanificaTareas {
    private Scheduler scheduler;
    private String cronSchedule;

                    …

    public void start() throws org.quartz.SchedulerException,
          java.text.ParseException {
     try {
// Definimos la tarea
       JobDetail job = JobBuilder.newJob(CreaReferenciaJob.class)
             .withIdentity("CreaReferenciaJob").build();

// conseguimos el fichero de configuración de las tareas
      propiedades.load(new FileInputStream(" .../quartz.properties"));

       cronSchedule = propiedades.getProperty("cronSchedule");
     } catch (IOException ex) {
            log.error(" No se ha cargado el quartz.properties. ", ex);
            log.debug("Cargo un cron por defecto. >0 0 2 ? * SUN,WED<");
            cronSchedule = "0 0 2 ? * SUN,WED";
     }

// Configuramos el Trigger que avisará al planificador de cuándo debe
// ejecutar la tarea.
          CronTrigger cronTrigger = TriggerBuilder
                .newTrigger()
                .withIdentity("creareferenciacrontrigger",
                      "creareferenciacrontriggergroup1")
                .withSchedule(
                      CronScheduleBuilder.cronSchedule(cronSchedule))
                .build();

// Obtenemos el planificador
       scheduler = org.quartz.impl.StdSchedulerFactory.getDefaultScheduler();

// La tarea definida en JobDetail será ejecutada en los instantes
       scheduler.scheduleJob(job, cronTrigger);

// Iniciamos las tareas planificadas en el Sheduler
       scheduler.start();
    }

    /**
     * Detiene el proceso de planificación
     */
    public void stop() {
      try {
          scheduler.shutdown();
      } catch (Exception ex) {
                        …
      }
    }
```

Figura 27: Planificador de tareas en Quartz

La planificación de las ejecuciones se detallará en el fichero quartz.properties, que se dejará en una ruta concreta dentro de la estructura de carpetas para que sea accesible por el servidor de aplicaciones. Mediante esta configuración, se obtiene la flexibilidad de poder modificar la planificación en función de la disponibilidad de recursos de red.

```
#############################################
#       Core
#############################################
#org.quartz.scheduler.instanceName = SIP_ClusteredScheduler
#org.quartz.scheduler.instanceId = AUTO

#The scheduler is local, which means it can't be accessed using RMI
org.quartz.scheduler.rmi.export = false
org.quartz.scheduler.rmi.proxy = false

#A maximum of X jobs can be run simultaneously
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 10
org.quartz.threadPool.threadsInheritContextClassLoaderOfInitializingTh
read = true

#############################################
#       PLUGINS
#############################################

# Logging
org.quartz.plugin.triggHistory.class =
org.quartz.plugins.history.LoggingTriggerHistoryPlugin
org.quartz.plugin.triggHistory.triggerFiredMessage = Trigger {1}.{0}
fired job {6}.{5} at: {4, date, HH:mm:ss MM/dd/yyyy}
org.quartz.plugin.triggHistory.triggerCompleteMessage = Trigger
{1}.{0} completed firing job {6}.{5} at {4, date, HH:mm:ss MM/dd/yyyy}
with resulting trigger instruction code: {9}

# Definición declarativa de los jobs
#org.quartz.plugin.jobInitializer.class =
org.quartz.plugins.xml.XMLSchedulingDataProcessorPlugin


org.quartz.plugin.jobInitializer.failOnFileNotFound = true
# scanInterval property -The interval (in seconds) at which to scan
for changes to the file.
#If the file has been changed, it is re-loaded and parsed. To disable
scanning set it to 0.
# si habilitamos esta property se produce el error de job duplicado:
org.quartz.plugin.jobInitializer.scanInterval = 0
org.quartz.plugin.shutdownhook.class =
org.quartz.plugins.management.ShutdownHookPlugin
org.quartz.plugin.shutdownhook.cleanShutdown = true

#############################################
#       Expresion cron para inicializar el planificador
#############################################
cronSchedule = 0 0 2 ? * SUN,WED
```

Figura 28: Ejemplo de fichero de configuración de Quartz

La sintaxis de Quartz para el cronSchedule no es más que una cadena de texto compuesta por varios campos, con un orden concreto, separados por espacios en blanco.

"Segundos" "Minutos" "Horas" "Día del mes" "Mes" "Día de la semana" "Año"

Los campos pueden contener alguno de los valores permitidos, o caracteres especiales que se indicarán en la siguiente tabla.

Campo	Obligatorio	Valores	Caracteres
Segundos	S	0-59	, – * /
Minutos	S	0-59	, – * /
Horas	S	0-23	, – * /
Día del mes	S	1-31	, – * ? / L W
Mes	S	1-12 o JAN-DEC	, – * /
Día de la semana	S	1-7 o SUN-SAT	, – * ? / L #
Año	N	Vacío | 1970-2099	, – * /

Tabla 4: Valores del planificador de Quartz

Es importante recordar que las semanas comienzan en Domingo (valor 1)

El valor de los caracteres especiales se especifica a continuación:

- * : Selecciona todos los valores de un campo (por ejemplo cada hora, cada minuto).
- ? : Selecciona sin un valor específico cuando se puede utilizar (es similar a decir cualquiera).
- : Selecciona rango de valores (por ejemplo 4-6 que es de 4 a 6).
- , : Selecciona valores específicos (por ejemplo MON,WED,FRI es decir los lunes, miércoles y viernes).
- / : Selecciona incrementos a partir del primer valor (por ejemplo 0/15 que es cada 15 minutos comenzando desde el minuto 0 -> 15, 30 ,45).
- L (Día del mes) : Selecciona el último día del mes.
- L (Día de la semana) : Selecciona el último día de la semana (7 / sabado / SAT).
- XL (Día de la semana) : Selecciona el último día de ese tipo del mes (por ejemplo 6L -> el último viernes del mes).
- W : Selecciona el día de la semana (de lunes a viernes) más cercano al día (weekday).
- LW : Selecciona el último weekday del mes.
- # : Selecciona la posición de un día del mes (por ejemplo 6#3 -> el tercer viernes del mes).

La configuración elegida, por tanto: cronSchedule = 0 0 2 ? * SUN,WED
establece la ejecución todos los lunes y los miércoles de todos los meses, a las 2:00 horas.

6. Pruebas Realizadas

Para realizar las pruebas de funcionamiento de los servicios Web, se utilizará la herramienta SoapUI, de SmartBear, en su versión 5.0.0.

SoapUI es una solución multiplataforma de Testing Funcional, libre y de código abierto que permite, a través de una interfaz gráfica, crear y ejecutar pruebas funcionales con gran agilidad.

Para realizar las pruebas a los servicios web ofrecidos por la Historia de Salud Electrónica, creamos un proyecto, mediante la conexión, desde el propio SoapUi, a la URL: https://hseavstest.san.gva.es/hse/services/ConsultasService_V3_0

Desde la aplicación SoapUI 5.0.0 se realiza la creación de un nuevo proyecto

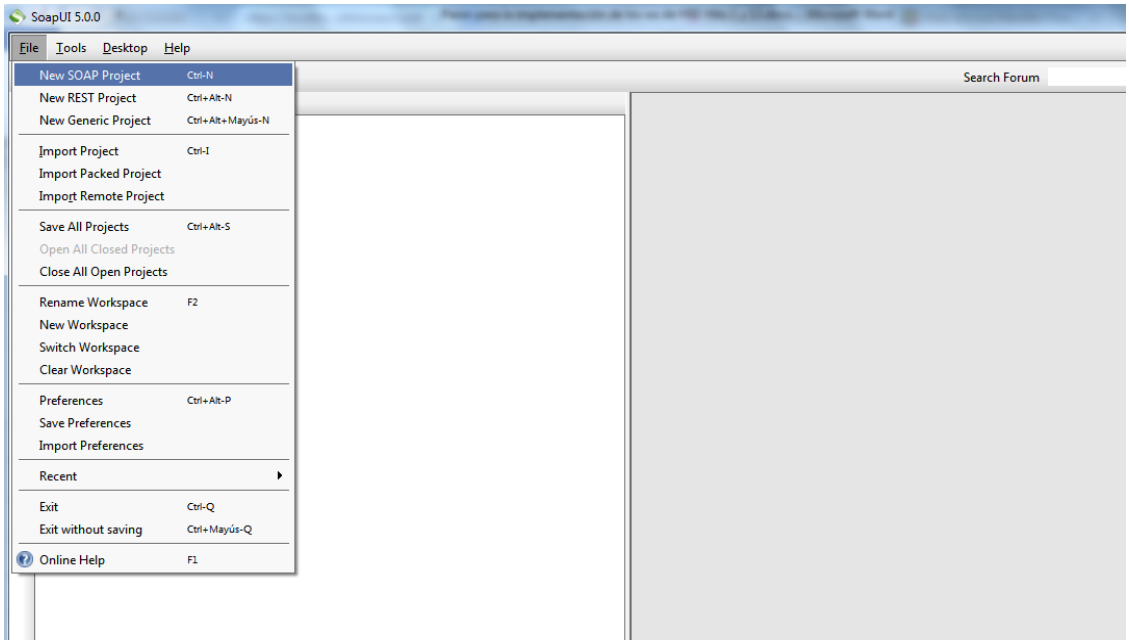

Figura 29: Creación de un nuevo proyecto

New Soap Project > introducción del nombre del proyecto y URL donde está disponible el WSDL

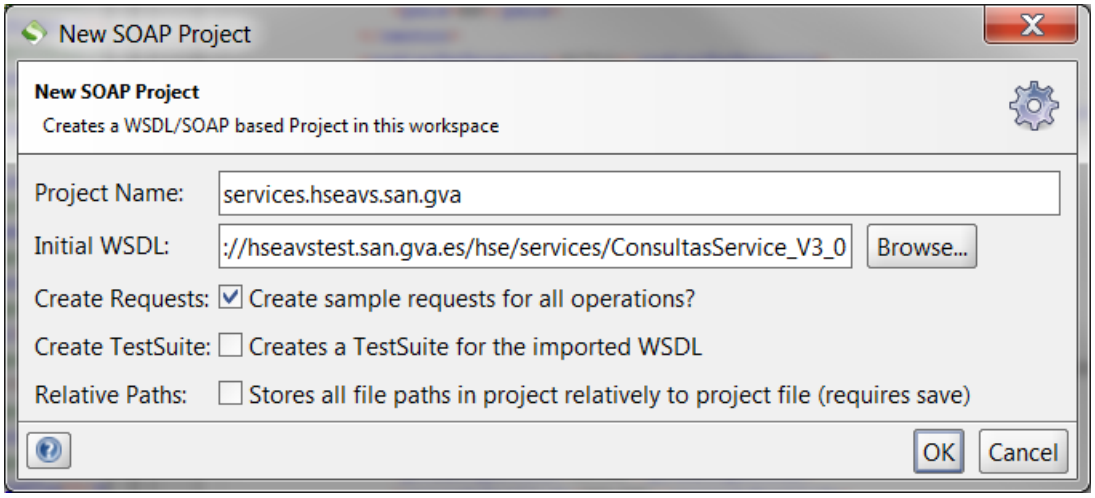

Figura 30: Creación de un nuevo proyecto II

Como resultado, se obtiene la estructura de proyecto con las llamadas a los servicios disponibles ya implementados

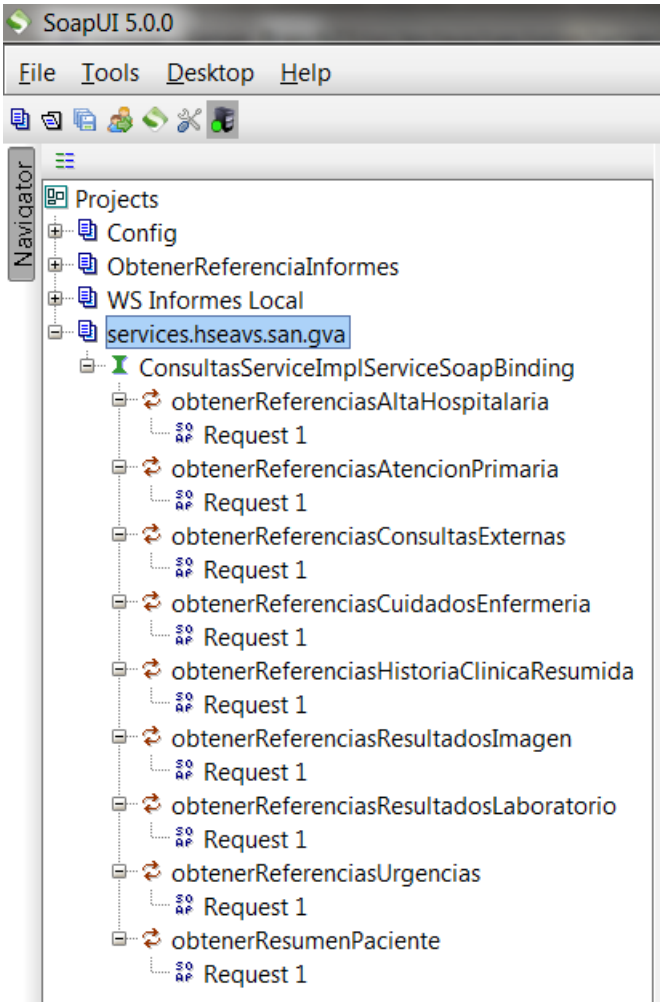

Figura 31: Proyecto creado con las llamadas implementadas

Es necesario realizar la configuración de seguridad, tanto los certificados a usar, como los mecanismos de encriptación, según las directrices marcadas por la Agencia Valenciana de Salud.

Mediante doble click sobre el nuevo proyecto soap, se presenta la ventana de configuración. En su pestaña WS-Security configuración y dentro de esta en la sección keystore se añade el almacén de claves OrionRIS.jks, y SoapUI solicita la password del almacén de claves.

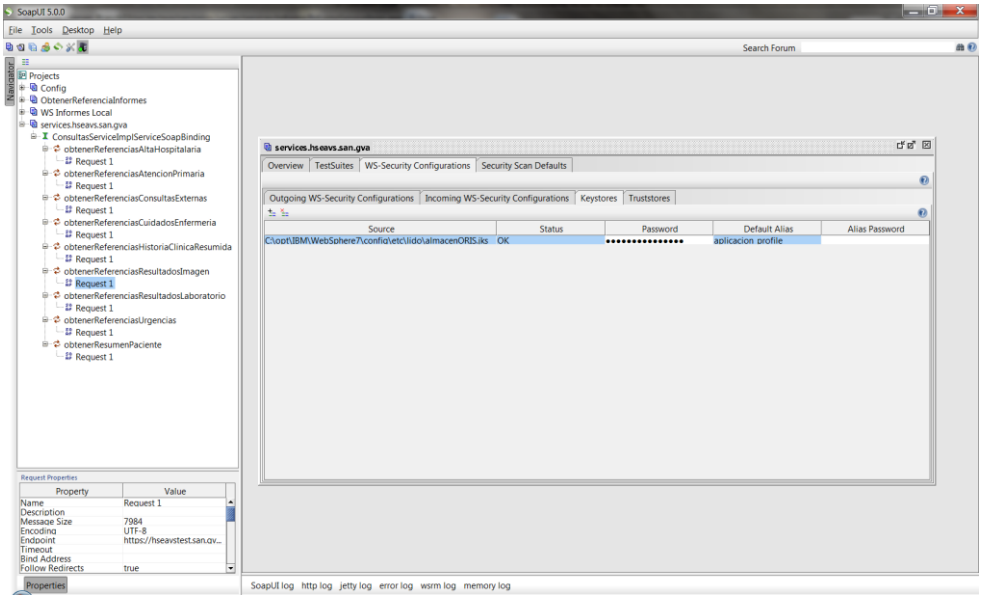

Figura 32: Configuración del almacén de certificados

Mediante la configuración del Outgoing WS-Security, estableceremos el certificado de aplicación que se extraerá del almacén definido en el paso anterior.

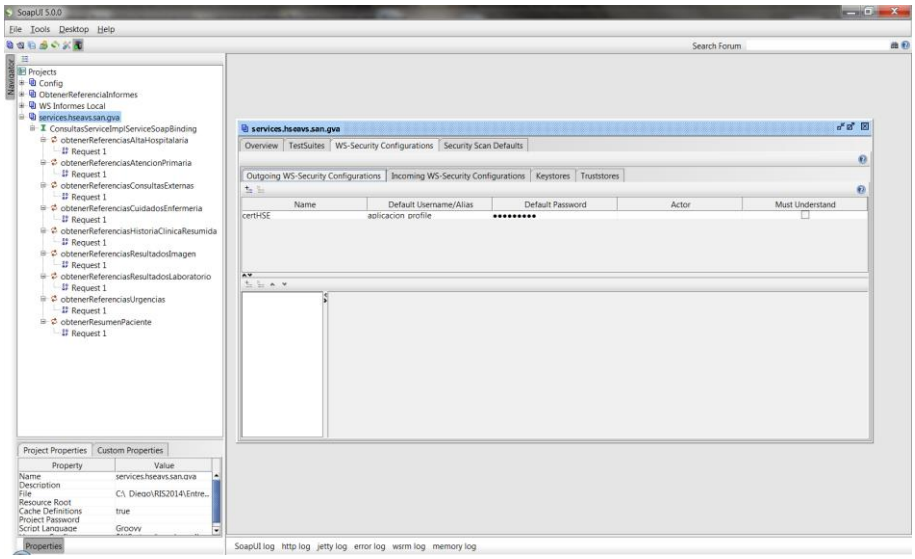

Figura 33: Configuración de certificado de aplicación

Es necesario añadir un campo Timestamp y un campo Signature, en los que se configurarán los algoritmos y características que marcan las directrices de la Agencia Valenciana de Salud.

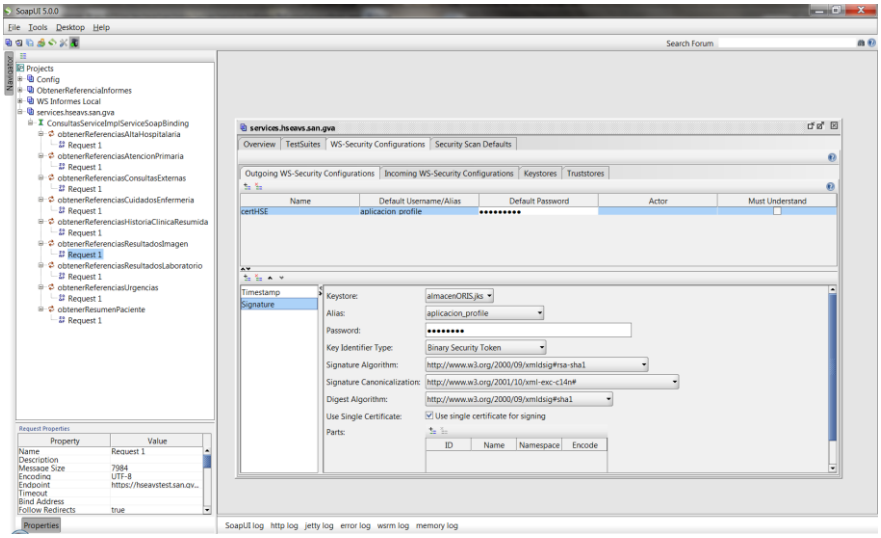

Figura 34: Configuración de algoritmos de encriptación

Por último, hay que modificar los valores en el mensaje para realizar la petición, sustituyendo los ? con los que SoapUI implementa los mensajes, por los valores correctos, y en el Request del servicio, en su pestaña AUTH, asignar una autorización BASIC y en el outgoing WSS asignar la configuración establecida en el punto anterior.

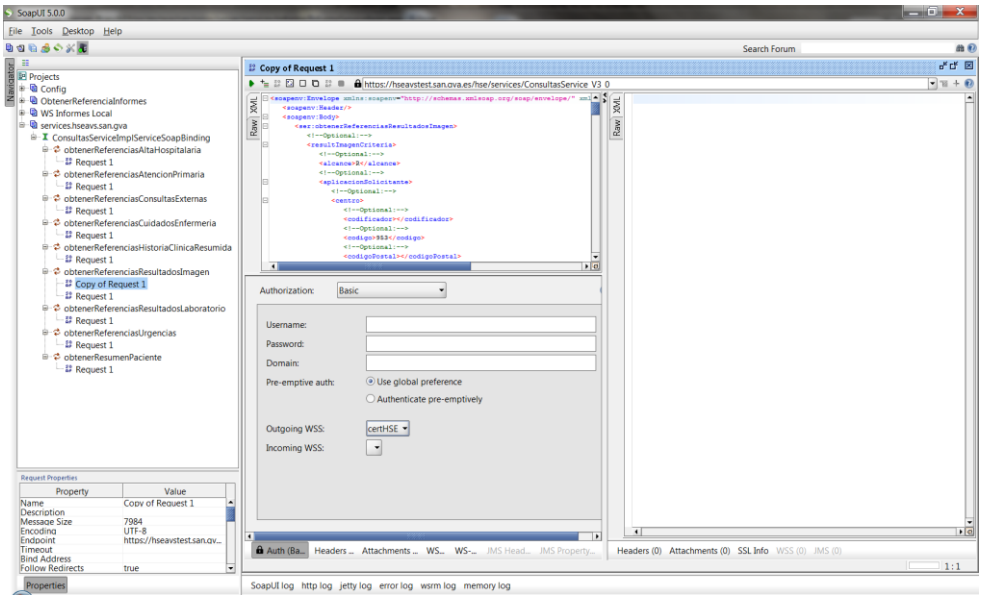

Figura 35: Establecimiento de las características de autentificación para la petición a un Servicio Web

Tras pulsar el  , primer icono de la barra de la ventana del Request de soapUI, se realiza la invocación al servicio correspondiente, y se obtiene la respuesta en la parte derecha de la ventana.

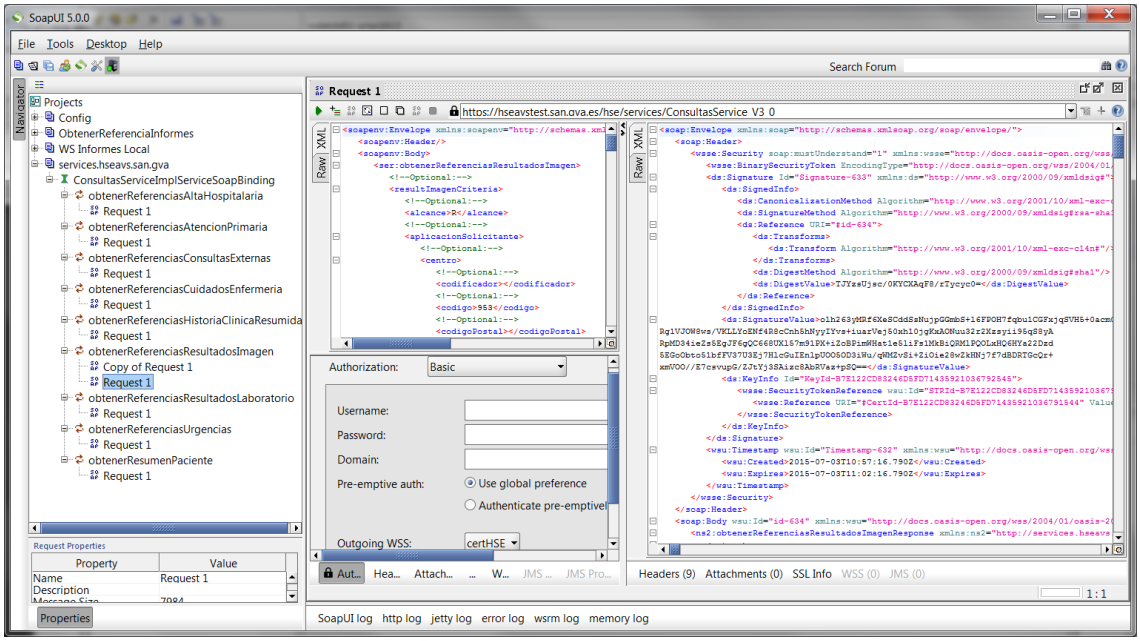

Figura 36: Ejecución de la invocación

En la respuesta se pueden observar las cabeceras de seguridad, y el mensaje de respuesta.

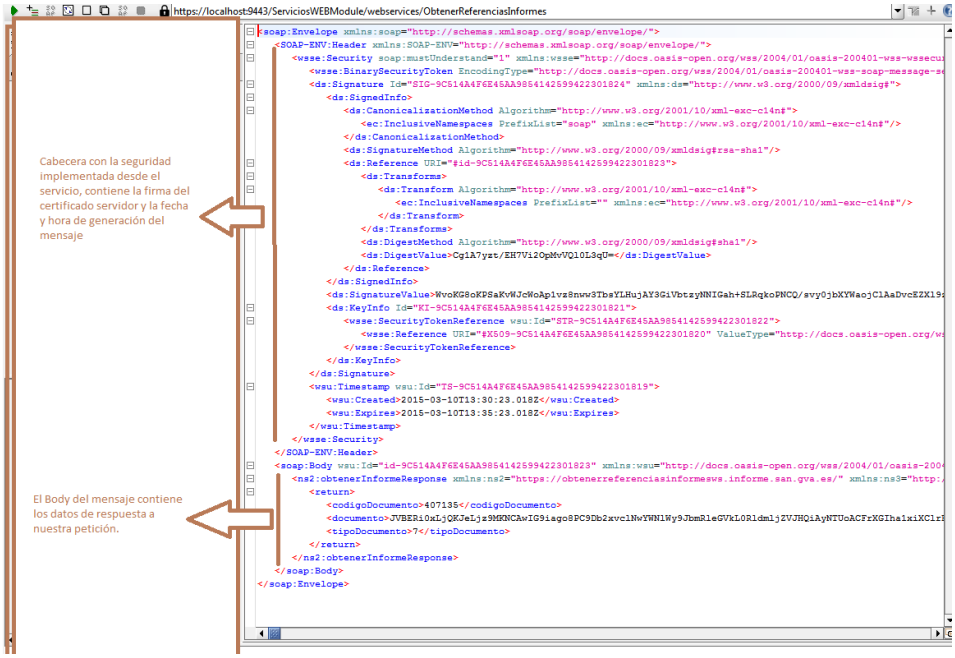

Figura 37: Mensaje de respuesta

Desde la pestaña SSL se puede consultar la información de firma utilizada para el canal de comunicación.

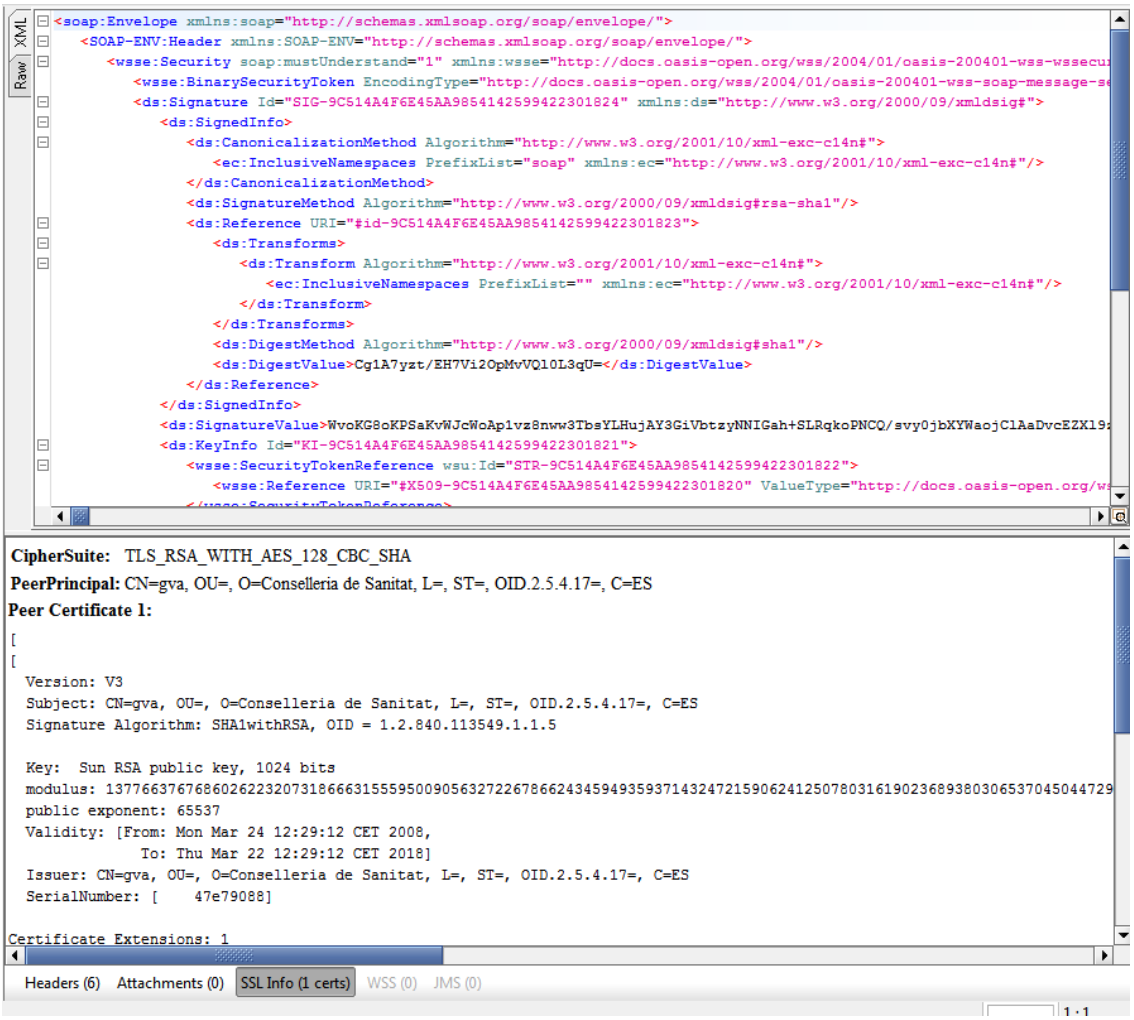

Figura 38: Información de seguridad

7. Conclusiones

7.1 Introducción

Con este apartado se pone fin al presente documento, exponiendo las dificultades principales que se han encontrado en el desarrollo del proyecto, así como las soluciones encontradas para solventarlos.

Se indica además la posible evolución de las integraciones de Orion-RIS, de la que la presente no es más que la primera piedra en el camino.

7.2 Problemas y soluciones

El principal problema que se ha encontrado en el desarrollo del presente proyecto, ha sido la dificultad para conseguir información de utilidad desde los equipos de soporte y desarrollo de las aplicaciones. Este aspecto, aunque no es deseable, es perfectamente comprensible, dado que son empresas rivales, y ofrecer demasiada información podría proporcionar ventaja a las empresas competidoras de cara a futuras adjudicaciones de contrato.

Otro problema importante se ha encontrado en la falta de disponibilidad de los entornos en Conselleria de Sanidad. Los entornos de Test están muy solicitados para las pruebas de gran cantidad de aplicaciones e integraciones. Los juegos de datos de prueba necesarios son de mala calidad u obsoletos, en concreto los informes almacenados en HSE Test no correspondían a pacientes de la base de datos de Orion-RIS Test. Para solventar la situación y poder realizar las pruebas necesarias hubo que desarrollar un proceso de sustitución de números de historia y códigos de centro online, que procesaba la petición previa a la invocación a los servicios web.

Por último, la cantidad de nuevas tecnologías usadas en la integración ha supuesto una carga de trabajo bastante elevada, que ha alargado los tiempos en principio planificados para el desarrollo de la integración.

7.3 Evolución

Cumplido el objetivo de construir con éxito la primera solución de integración al margen del motor de integración, se desea, por parte de la Conselleria de Sanitat, continuar con este proceso, sustituyendo progresivamente las integraciones que actualmente utilizan dicho motor, como podrían ser las de SIP o PACS, o abordando ya directamente al margen de este las nuevas integraciones con otras aplicaciones como Alta Hospitalaria, el Visor de Historia Clínica, etc, con el fin de enriquecer la experiencia de los usuarios y la información asistencial y clínica que ofrece Orion-RIS.

Referencias

Historia de Salud Electrónica. (2008). Conselleria de Sanitat
Fecha de consulta: 8 de mayo de 2015
http://www.dgfc.sgpg.meh.es/sitios/dgfc/es-ES/ipr/fcp0713/c/bp/ac/ac2012/Documents/BPAC2012CV_2.pdf

Documentación interna Orion-RIS (2005).*Conselleria de Sanitat*.
Fecha de consulta: 16 de mayo de 2015

El motor de Integración Rhaspsody.(2015). Orion-Health
Fecha de consulta: 22 de mayo de 2015
https://orionhealth.com/

Servicios Web (2015). *W3C Consortium*.
Fecha de consulta: 2 de junio de 2015
http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb

Soap. *Wikipedia, la enciclopedia libre*.
Fecha de consulta: 4 de junio de 2015
https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol

Criptografía Asimétrica (2015, 6 de junio). *Wikipedia, la enciclopedia libre*.
Fecha de consulta: 7 de junio de 2015
https://es.wikipedia.org/wiki/Criptografia_asimetrica

Java Api for XML WebServices (2015, 1 de junio). *Wikipedia, la enciclopedia libre*.
Fecha de consulta: 8 de junio de 2015
https://es.wikipedia.org/wiki/JAX-WS

Apache CXF (2015). *The Apache Software fundation*.
Fecha de consulta: 10 de junio de 2015
http://cxf.apache.org/

WS-Security (2013, 10 de marzo). *Wikipedia, la enciclopedia libre*.
Fecha de consulta: 21 de junio de 2015
https://es.wikipedia.org/wiki/WS-Security

Quartz Scheduler (2015). *Quartz Enterprise Scheduler .NET*.
Fecha de consulta: 06 de julio de 2015
http://www.quartzscheduler.net/

Agradecimientos

A mi familia por el continuo apoyo y soporte anímico.

A Blanca, por todo el tiempo que le he robado.

A mi tutor, el Doctor Juan Luis Posadas Yagüe, sin cuyo apoyo y guía no habría sido posible la realización de este trabajo fin de grado.