



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática

Universitat Politècnica de València

Análisis y diseño de la aplicación Orión-RIS

Proyecto Final de Carrera

Ingeniero Informático

Autor: David Fuentes Muñoz

Director: Dra. Dña. María José Castro Bleda

2 de septiembre de 2012

Índices de imágenes	1
Acrónimos	2
1 Introducción	3
1.1 Motivación	3
1.2 Contexto de la aplicación	3
1.3 Objetivos	5
2 Especificación de requisitos	7
2.1 Introducción	7
2.1.1 Propósito	7
2.1.2 Ámbito del sistema	7
2.1.3 Visión general del documento	8
2.2 Descripción general	8
2.2.1 Perspectiva del producto	8
2.2.2 Funciones del producto	10
2.2.3 Características de los usuarios	18
2.2.4 Restricciones	19
2.2.5 Suposiciones y dependencias	20
2.2.6 Requisitos futuros	21
2.3 Requisitos específicos	21
2.3.1 Interfaces externas	21
2.3.2 Funciones	22
2.3.3 Requisitos de rendimiento	30
2.3.4 Restricciones de diseño	31
2.3.5 Atributos del sistema	31
2.3.6 Otros requisitos	32
3 Análisis	33
3.1 Caso de uso del administrativo	33
3.2 Caso de uso del técnico radiólogo/A.T.S	34
3.3 Caso de uso del radiólogo	34
3.4 Caso de uso del administrador	35
4 Diseño	37
4.1 Aspectos generales	37
4.2 Interfaz de usuario	38
4.3 Servidor	39
4.4 Motor de persistencia	39
4.5 Base de datos	40
5 Implementación	43
5.1 Introducción	43
5.2 Interfaz de usuario	43
5.3 Servidor	45
5.4 Motor de persistencia	47
6 Valoración	51
7 Bibliografía	53

Índices de imágenes

<i>Ilustración 1: SISAN</i>	4
<i>Ilustración 2: Integración de Orión-RIS con otros sistemas</i>	10
<i>Ilustración 3: Pantalla de citación</i>	12
<i>Ilustración 4: Pantalla de Gestión de Cita</i>	13
<i>Ilustración 5: Pantalla de Captura de Actividad</i>	15
<i>Ilustración 6: Pantalla de Informado</i>	17
<i>Ilustración 7: Pantalla de Administración</i>	18
<i>Ilustración 8: Caso de Uso del perfil Administrativo</i>	33
<i>Ilustración 9: Caso de Uso del perfil de Técnico Radiólogo</i>	34
<i>Ilustración 10: Caso de Uso del perfil de Radiólogo</i>	35
<i>Ilustración 11: Caso de Uso del perfil de Administrador</i>	35
<i>Ilustración 12: Arquitectura de Orión-RIS</i>	37
<i>Ilustración 13: Relación de las principales tablas.</i>	41
<i>Ilustración 14: Comportamiento del patrón de diseño Observer</i>	44
<i>Ilustración 15: Estructura de paquetes en el servidor</i>	45
<i>Ilustración 16: Operaciones de persistencia disponibles.</i>	46

Acrónimos

HIS: Hospital Information System, en algunos hospitales se llama IRIS y otros HIGIA, pero básicamente son iguales.

RIS: Radiology Information System, antes de la llegada de Orión-RIS se llamaba simplemente RIS.

PACS: Picture Archiving and Communication System, sistema de almacenamiento y gestión de la imágenes del hospital.

AVS: Agencia Valenciana de Salud.

SISAN: Sistema de Información Sanitario de la Generalitat Valenciana.

Rhapsody: Es el software que hace de “traductor” entre los distintos sistemas que cooperan dentro de SISAN.

SIP: Sistema de Información Poblacional, núcleo importante dentro de SISAN, en el que muchas aplicaciones se apoyan.

JDOQL: Es el propio lenguaje de consultas orientado a objetos que utiliza JDO.

1 Introducción

El presente documento presenta la memoria del sistema de información de Orión-RIS. Un sistema con el que se trabaja en el sector sanitario, más concretamente en los servicios de radiología de los hospitales.

Aplicación cuya propiedad corresponde a la Agencia Valenciana de Salud y que cuyo desarrollo fue adjudicado en sus inicios a la empresa Steria Ibérica S.A.U. en el año 2006. Desde entonces y hasta el día de hoy Steria ha sido la empresa adjudicataria para realizar el desarrollo funcional de la aplicación, su mantenimiento y para dar soporte técnico a la Unidad de Informática de los distintos hospitales.

La estructura de este documento detallará el contexto de la aplicación y su interacción con el resto de sistemas de SISAN para facilitar a los profesionales el desarrollo de su actividad.

Describiremos la especificación de requisitos el análisis técnico de la aplicación, su diseño e implementación.

Analizaremos a lo largo de este documento la importancia que tiene Orión-RIS para el éxito futuro de Orión-Clínic.

1.1 Motivación

La motivación principal es mejorar el sistema de información que gestionaba el servicio de radiología de los hospitales en la Comunidad Valenciana y dotar a los profesionales radiólogos de una estación radiológica con la que operar con más comodidad, eficacia y eficiencia.

Seguidamente veremos que este proyecto, Orión-RIS, sería la “punta de flecha” de un proyecto todavía más ambicioso y grande que pretende gestionar toda la actividad de un hospital de una forma integrada.

1.2 Contexto de la aplicación

Para situar a Orión-RIS en un contexto concreto es necesario hablar de tres sistemas de información donde cada uno se sitúa dentro del anterior, como podemos ver en la Ilustración 1: SISAN más adelante:

- **SISAN (Sistema de Información Sanitario de la Generalitat Valenciana):** Lo forman un gran conjunto de subsistemas, son soluciones software que dan soporte a las distintas actividades del sector sanitario y a los distintos perfiles profesionales que en este sector desarrollan su actividad. Muchos de estos sistemas que están

dentro de este marco, SISAN, se comunican entre ellas, están integradas y cooperan de alguna manera entre sí para que este sistema ofrezca la visión de un único sistema compacto al servicio del sector sanitario en su conjunto.

- **Orión-Clínic:** Es una aplicación que forma parte del conjunto de aplicaciones llamadas aplicaciones clínico-asistenciales. En concreto Orión-Clínic es el sistema de información que se compone del sistema Orión-RIS, el sistema HIS (Hospital Information System) y el de farmacia. Así pues Orión-Clínic cubre gran parte de las necesidades del sistema clínico-asistenciales junto con Abucasis II, sistema utilizado en los Centros de Atención Primaria.
- **Orión-RIS:** La importancia de Orión-RIS dentro de Orión-Clínic reside en que Orión-RIS fue la “punta de flecha”. Así pues fue la primera aplicación que se desarrolló y la que sería la base tecnológica de Orión-Clínic.

La primera implantación y pilotaje de Orión-RIS stand-alone fue en el año 2006 en el Hospital de San Juan de Alicante. A partir de ese momento se empezó a implantar en el resto de Hospitales de la Comunidad Valenciana que eran de gestión pública hasta un total de 22 hospitales, así como de los respectivos Centros de Especialidades donde también existe una actividad radiológica.

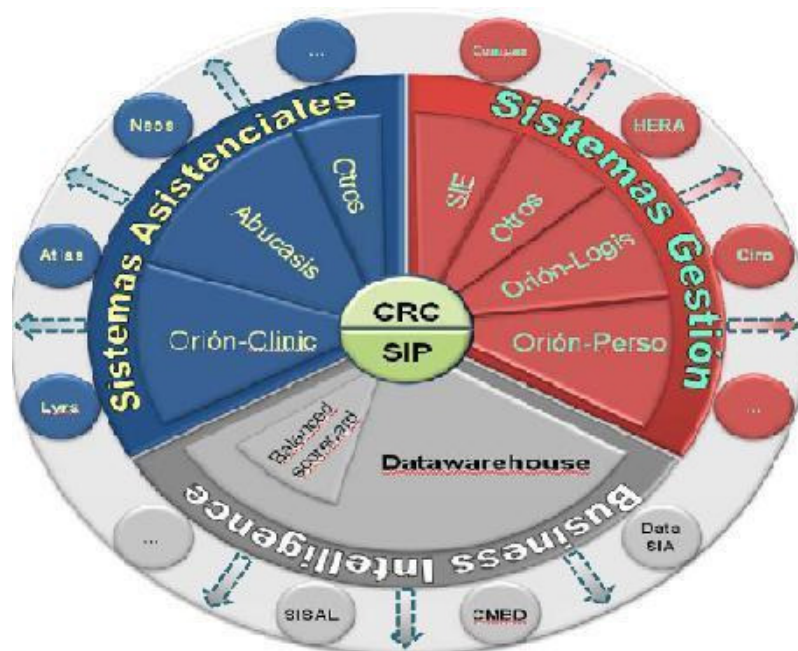


Ilustración 1: SISAN

1.3 Objetivos

Los objetivos de Orión-RIS son varios:

- Existe la necesidad de una renovación tecnológica en el sector sanitario que utiliza la mayoría del personal sanitario en los hospitales.
- El servicio de radiología es uno de los servicios con más actividad de los hospitales. Y se pretende dotar al personal del servicio de radiología de una herramienta capaz de agilizar su actividad y digitalizarla.

Sobre todo se mejora mucho la forma en el que el radiólogo desarrolla su actividad creando una estación radiológica más potente y sofisticada. En esta estación el radiólogo será capaz de visualizar la imagen del paciente que pretende informar, redactar el informe mediante un dictáfono firmando el informe de manera digital y mandando este informe a un sistema PACS que lo pondrá a disposición del facultativo especialista que lo quiera consultar desde cualquier consulta.

- Otro objetivo, a un nivel más alto, de esta aplicación es ir introduciendo el que será el futuro Orión-Clínic haciendo que su futura puesta en marcha sea lo más cómoda posible.

2 Especificación de requisitos

2.1 Introducción

La especificación de requisitos es una descripción completa del comportamiento del sistema que se va a desarrollar, en nuestro caso un sistema para la gestión del servicio de rayos de los hospitales.

Este apartado ha sido realizado siguiendo el estándar IEEE 830 de 1998. Un estándar que ha sido criticado por muchos autores y desde múltiples puntos de vista pero es la forma más extendida para realizar el análisis de requisitos.

2.1.1 Propósito

El análisis de requerimientos tiene el objetivo de proporcionar tanto al cliente como al propio desarrollador o analista una visión funcional completa de lo que será la aplicación. Sus requerimientos funcionales, así como las restricciones y límites de la aplicación quedaran reflejados en este apartado.

Su interacción con otros sistemas así como la interacción con los distintos usuarios será también claramente descrita.

2.1.2 Ámbito del sistema

La aplicación software que en este apartado se describe es un sistema de información para la gestión del servicio de radiología de un hospital y en la mayoría de casos también para los centros de especialidades que están asociados con este hospital y en los que existe actividad radiológica.

Es necesario puntualizar para una mejor comprensión que la estructura departamental se encuentra definida o dividida principalmente por departamentos, hasta un total de 24 departamentos de salud, a lo largo de la Comunidad Valenciana. Cada departamento se compone de Centros de Atención Primaria (CAP) y Centros de Especialidades (CCEE) asociados a un hospital.

La aplicación Orión-RIS, solo operará en aquellos centros donde exista actividad radiológica, normalmente hospitales y CCEE.

2.1.3 Visión general del documento

En los siguientes apartados de esta ERS hablaremos de las restricciones que afectan a los requisitos y de aquello que de alguna manera puede afectar a su funcionalidad.

2.2 Descripción general

En esta sección se detalla el contexto de la aplicación, con que sistemas interacciona para llevar a cabo su objetivo y presentaremos imágenes de las pantallas de poder tener una visión general. Más adelante describiremos los requisitos de manera detallada.

2.2.1 Perspectiva del producto

En la reciente historia de esta aplicación habría que aclarar que, hoy por hoy, existen todavía dos vertientes de este mismo producto. Por un lado el que empezó en el año 2006 que es el Orión-RIS stand-alone. Por otro lado está el subsistema Orión-RIS que pertenece a un sistema mayor que es el ya mencionado Orión-Clínic.

Existe el propósito de ir prescindiendo de Orión-RIS stand-alone, mientras se vaya implantando Orión-Clínic. Así pues en hospitales como el Hospital Doctor Arnau de Vilanova, en el que Orión-Clínic está actualmente funcionando, la aplicación Orión-RIS stand-alone ya no existe.

En la Ilustración 2: *Integración* de Orión-RIS con otros sistemas, podemos ver que otros subsistemas de SISAN son los que cooperan con Orión-RIS para que éste pueda llevar a cabo su cometido dentro del servicio de rayos.

Podemos distinguir dos tipos de sistemas con los que interacciona:

- **Periféricos hardware:**

Monitores barco: El mantenimiento de este equipamiento corre a cargo de la empresa proveedora de PACS. Una vez el radiólogo selecciona la exploración que quiere informar, Orión-RIS genera un mensaje xml con el SIP y el NHC del paciente así como el identificador de la prueba. Lo manda a un puerto local donde un programa de PACS se encuentra escuchando. Con esta información PACS puede buscar en su sistema las imágenes asociadas al estudio o exploración y mostrarlas en el monitor para que el radiólogo realice el diagnóstico.

Dictáfono: La empresa que proporciona el componente java necesario para poder embeberlo dentro de Orión-RIS y que proporciona también los dictáfonos así como las licencias para que los radiólogos puedan utilizarlos es PHILIPS.

Lector de chips de tarjeta: El teclado utilizado es el Cherry 8040 para la lectura del chip de las tarjetas con las que los radiólogos realizan la firma digital una vez quieren cerrar un informe.

Este teclado dispone también de un lector de bandas para pasar la tarjeta SIP de los pacientes y realizar una búsqueda del paciente sin errores de transcripción, mayormente usado por los administrativos.

- **Sistemas software:**

Rhapsody: Motor de Integración utilizado por SISAN, para posibilitar la comunicación entre los distintos sistemas o subsistemas de SISAN

HIS (Hospital Information System): Este sistema permite la gestión de pacientes (altas, bajas y modificaciones) mientras que Orión-RIS no lo permite. Las tablas de pacientes de ambos sistemas deben de estar siempre sincronizadas de manera que una sea el espejo de la otra. Este requisito tiene que ver con la futura puesta en marcha de Orión-Clínica. La forma elegida para que ambos sistemas mantengan la información de pacientes de manera integral y consistente es mediante mensajería a través de Rhapsody.

SIP: Es un sistemas del que muchas aplicaciones dentro de sanidad se apoyan. En Orión-RIS si un paciente nuevo, no registrado todavía en el hospital donde acude quisiera una cita, se buscaría por su SIP, entonces el sistema enviará al sistema SIP una petición de búsqueda, éste responderá a la petición con el paciente que corresponda al SIP introducido, y Orión-RIS de forma transparente solicitará a HIS el alta de un nuevo paciente.

PACS (Picture Archiving and Communication System): Principalmente es el sistema que se encarga de almacenar y distribuir las imágenes radiológicas aunque también tiene otros cometidos.

A este sistema se mandará toda la mensajería de mantenimiento de paciente que se genera desde el HIS correspondiente.

Otro tipo de información que se envía desde Orión-RIS hacia PACS son los informes una vez ejecutada la firma de éstos y las citas que se generan desde Orión-RIS, para que éste las distribuya en las diferentes modalidades del servicio de rayos. Así se consigue que los Técnicos Radiólogos tengan disponible en la modalidad que corresponda la lista diaria de citas con la que se dispone a trabajar y poder asociar la imagen que se genere con la cita correspondiente.

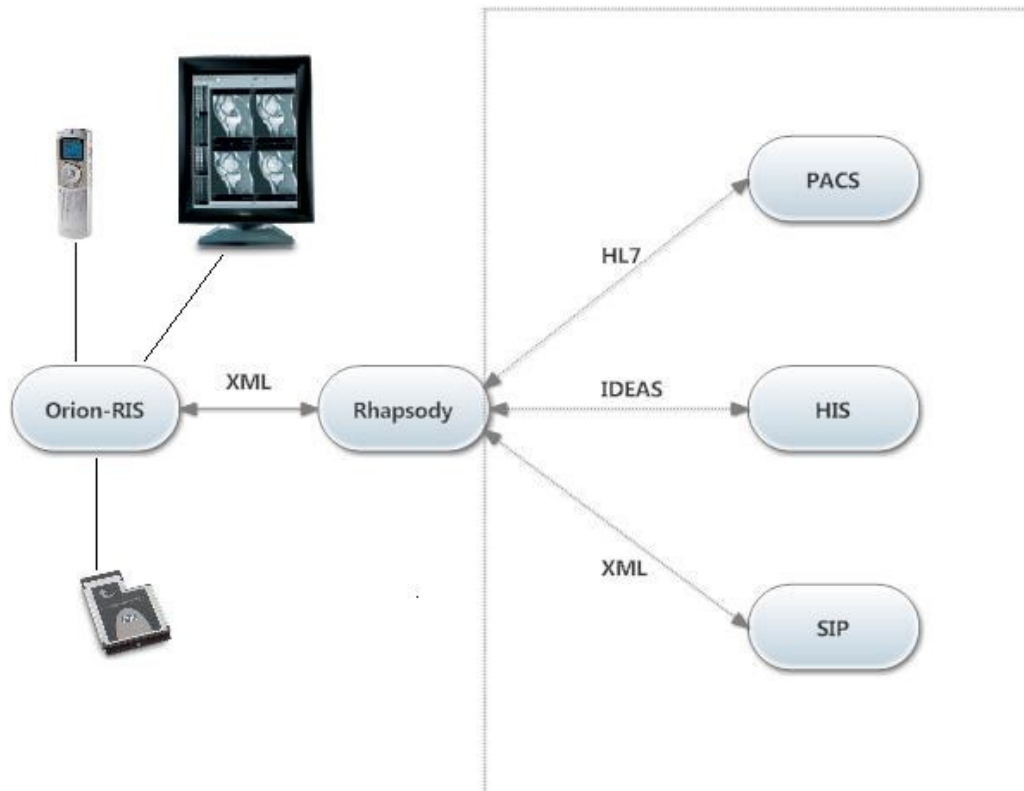


Ilustración 2: Integración de Orión-RIS con otros sistemas

2.2.2 Funciones del producto

Dividida por casos de uso sería:

- **Citación:**

El administrativo o técnico radiólogo podrán generar una cita programada o una cita urgente.

Para ello dispondremos en una misma pantalla de la información necesaria para realizar una cita ya sea programada o urgente.

Necesitamos un componente que haga posible la búsqueda del paciente y que lo haga posible por medio de varios campos; SIP, NHC, nombre y apellidos, NIF.

También dispondremos de un componente tipo árbol donde podremos encontrar cualquier centro de la comunidad valenciana así como los servicios del propio hospital para poder así seleccionar quien es el solicitante de la prueba a realizar.

Dispondremos de un componente adicional para añadir información opcional referente a la cita como por ejemplo; códigos CIE, fecha de solicitud, fecha deseada, si está embarazada..., etc.

Otro componente obligatorio será el que contenga todas las pruebas que el usuario logueado puede citar, por la sección en la que trabaja.

Así mismo dependiendo de qué prestaciones elijamos tendremos un componente donde nos aparecerá la agendas o agendas donde se puede realizar la prueba elegida.

Por último, dispondremos de un componente que será un calendario de la agenda para poder visualizar la disponibilidad o en caso de querer citar de manera manual poder elegir el día y la hora de la cita si se rechaza la propuesta inicial que realiza el sistema.

Ilustración 3: Pantalla de citación

- **Gestión de citas:**

El administrativo podrá anular, reprogramar o imprimir la cita deseada.

Al igual que en la pantalla de citación aquí dispondremos del mismo componente de paciente para realizar la búsqueda de la cita que queremos.

Más abajo tendremos el componente que muestra las agendas, para poder seleccionar una en el proceso de reprogramación.

En el centro de la pantalla se mostrara el resultado de la búsqueda, que será el conjunto de citas del paciente buscado, citas pendientes, realizadas, canceladas y reprogramadas.

En la parte derecha de la pantalla tendremos toda la información referente a la cita seleccionada bien para consulta o para modificar desde la opción “Modificar” del menú contextual.

Paciente

SIP: 2151181

FUENTES MUÑOZ, DAVID
F. nacimiento 14/05/1979

Ver datos administrativos
Ver Citas del Paciente
Buscar otro paciente

Agendas Citables

- HOSPITAL DOCTOR PESET
 - RADIODIAGNOSTICO
 - RADIODIAGNOSTICO
 - ABCDEF-AGENDA CITACIÓN
 - ECO1-ECOGRAFIA 1
 - KRCL-A LA CARCEL
 - SALA3-SALA 3

Fecha	Hora	Prestación	Agenda	Estado
PENDIENTE				
03/05/2011	15:00	02 228 06 490;	ABCDEF	CITADA
FINALIZADA				
11/01/2010	22:00	02 225 02 020...	SALA3	REALIZADA
07/05/2010	14:30	02 225 02 020;	ABCDEF	REALIZADA
07/05/2010	08:50	02 228 06 460;	SIL	INFORMADA
13/05/2010	08:40	02 229 08 100;	SIL	REALIZADA
28/05/2010	00:12	02 229 08 070;	ABCDEF	REALIZADA
26/10/2010	13:25	02 225 01 080;	DFM	INFORMADA
26/10/2010	13:15	02 225 01 030;	DFM	INFORMADA
02/11/2010	08:00	02 229 08 080...	SIL	CITADA
04/11/2010	18:17	02 225 02 020;	KRCL	INFORMADA
05/11/2010	08:00	02 225 02 020;	SIL	INFORMADA
05/11/2010	08:00	02 225 02 020;	KRCL	ANULADA
23/02/2011	09:06	02 225 02 020;	ABCDEF	CITADA
07/04/2011	12:45	02 227 05 030;	ECO1	CITADA
07/04/2011	11:55	02 227 05 030;	ECO1	CITADA
07/04/2011	13:35	02 227 05 030;	ECO1	CITADA
07/04/2011	11:05	02 227 05 030;	ECO1	CITADA
07/04/2011	12:45	02 227 05 030;	ECO1	CITADA
07/04/2011	10:15	02 227 05 030;	ECO1	CITADA

Datos de la cita no editables

Datos de registro

Cita:
Fecha:
Registrador:

¿Quién demanda la exploración?

Cambiar solicitante

Prestaciones

Descripción	Late.

Datos de la cita

Datos clínicos | Alergias | Observaciones

Diag. [] CIE

Prioridad Normal

1er. diag. Embarazada

F. Solicitud []

F. Deseada []

Guardar

FUENTES MUÑOZ, DAVID - Sección: NIVEL 2 Marcado de llegada a Sala 13:55:04

Ilustración 4: Pantalla de Gestión de Cita

- **Captura de actividad.**

Para las citas programadas, una vez llega el día en el que se realiza la exploración al paciente el técnico radiólogo deberá capturar la cita para marcar las exploraciones como realizadas.

La pantalla tendrá también el componente de paciente bien para buscar citas de ese paciente o para visualizar la información referente al paciente cuando seleccionamos una cita de la lista de citas.

Bajo el componente de paciente tendremos un componente que muestra las agendas con las que el usuario logueado trabaja y permitir así que éste seleccione una para empezar a trabajar con ella.

Un poco más abajo disponemos de un filtro para poder mostrar solo las pendientes de capturar, solo la capturadas o ambas. Y seguidamente el botón "Actualizar citas" que finalmente nos mostrara los resultados de aplicar nuestra selección.

Por último en esta parte de la pantalla tenemos el componente "Personal", con el que se pretende registrar al usuario técnico radiólogo o A.T.S. que realiza la prueba así como seleccionar si fuera necesario un radiólogo como responsable de informar las pruebas.

La parte que ocupa el centro y la derecha de la pantalla puede ser dividida en dos partes la parte de arriba es donde se mostrará el resultado de la búsqueda al buscar por paciente o por agenda. Y la parte de abajo en la que nos mostrara la información importante sobre la cita que arriba hemos seleccionado, como por ejemplo las pruebas que incluye.

DFM - ORION - RIS Módulo de Captura de Actividad - DPTO. 10 VALENCIA - DR. PESET

Módulos Preferencias Ayuda

Paciente Activo

SIP

FUENTES MUÑOZ, DAVID

F. nacimiento 14/05/1979 ♂

Ver datos administrativos

Buscar otro paciente

Lista de Trabajo

-Equipos-

ABCDEF-AGENDA CITACIÓN
ECO1-ECOGRAFÍA 1
KRCL-A LA CARCEL
PRUEBA-PRUEBA
SALA3-SALA 3

-Citas-

Día seleccionado 09-may-2011

-Estado-

Todos

Personal

-Realizador-

ENFERMERO

DESCONOCIDO

-Sección/Radiólogo por defecto-

RADIODIAGNOSTICO

HORA	SIP	PACIENTE	PRESTACIÓN	ESTADO	H
08:00	2151181	FUENTES MUÑOZ, DAVID	02 229 08 080;02 225 02 020	PENDIENTE	
09:06	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	PENDIENTE	
11:05	2151181	FUENTES MUÑOZ, DAVID	02 227 05 030	PENDIENTE	
11:55	2151181	FUENTES MUÑOZ, DAVID	02 227 05 030	PENDIENTE	
12:45	2151181	FUENTES MUÑOZ, DAVID	02 227 05 030	PENDIENTE	
13:35	2151181	FUENTES MUÑOZ, DAVID	02 227 05 030	PENDIENTE	
12:45	2151181	FUENTES MUÑOZ, DAVID	02 227 05 030	PENDIENTE	
15:00	2151181	FUENTES MUÑOZ, DAVID	02 228 06 490	PENDIENTE	
10:00	2151181	FUENTES MUÑOZ, DAVID	02 225 02 120	CAPTURADA	09
10:00	2151181	FUENTES MUÑOZ, DAVID	02 225 01 040	CAPTURADA	09
00:00	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	PENDIENTE	
09:12	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	CAPTURADA	09
13:21	2151181	FUENTES MUÑOZ, DAVID	02 229 08 100	CAPTURADA	13
01:30	2151181	FUENTES MUÑOZ, DAVID	02 229 08 270	PENDIENTE	
02:45	2151181	FUENTES MUÑOZ, DAVID	02 229 09 030	PENDIENTE	
08:00	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	CAPTURADA	08
22:00	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	PENDIENTE	
03:45	2151181	FUENTES MUÑOZ, DAVID	02 229 08 100	PENDIENTE	
04:45	2151181	FUENTES MUÑOZ, DAVID	02 229 08 090	PENDIENTE	
05:45	2151181	FUENTES MUÑOZ, DAVID	02 229 08 210	PENDIENTE	
06:15	2151181	FUENTES MUÑOZ, DAVID	02 229 10 040	PENDIENTE	
09:00	2151181	FUENTES MUÑOZ, DAVID	02 229 10 160A	PENDIENTE	
08:25	2151181	FUENTES MUÑOZ, DAVID	02 225 02 020	CAPTURADA	08

-Detalle de la cita-

Diagnóstico

Datos Clínicos Embarazada

-Prestaciones-

Cod/Num...	Descripción	Contraste	Proyec...	Realiz...	Motivo	Later.
02 229 1...	ANGIOPLASTIA ARTERIAL	No	Indef...			

-Materiales-

Código	Descripción	Tipo	Cantidad	Unidad Métrica

Realizador Sección/Radiólogo

Fecha Realización Hora Realización Duración Radiación

FUENTES MUÑOZ, DAVID - RADIODIAGNOSTICO 8:58:38

Ilustración 5: Pantalla de Captura de Actividad

- **Informado.**

El doctor radiólogo podrá buscar la exploración o exploraciones que quiere informar utilizando distintos filtros que tiene disponible, aunque el más común es la búsqueda por paciente.

Una vez encontrado creará un nuevo informe, utilizando el dictáfono o una serie de macros existentes o bien editando mediante el teclado.

Como podemos ver en la captura de pantalla tenemos en esta pantalla dos partes la parte izquierda donde tenemos el componente de paciente para realizar la búsqueda por paciente.

Además tenemos varios filtros para realizar la búsqueda de las pruebas ya realizadas y listas para informar como por ejemplo el filtro por agendas o el de modalidades.

Un poco más abajo tenemos ayuda para realizar la opción de búsqueda de informes que abrirá una nueva ventana donde podremos realizar la búsqueda por varios campos. También tenemos en el mismo apartado una opción “Mis informes” que listará todos los informes del radiólogo que inició la sesión, firmados y pendientes de firmar.

En la parte central de la pantalla tenemos el panel donde veremos el resultado de las búsquedas que vayamos realizando. Básicamente hay tres posibles listados, el de pruebas, el de informes y el listado mixto de pruebas e informes que será el resultado de una búsqueda por paciente que muestra directamente el historial radiológico del mismo.

En esta misma parte central pero en una nueva pestaña se abrirá si así lo necesitamos el editor de informes que dispone a su vez de varios ayudadores para la edición. Desde el mismo editor se puede guardar y finalmente firmar si se confirma que es definitivo.

DFM - ORION - RIS Módulo de Informado - DPTO. 10 VALENCIA - DR. PESET

Módulos Preferencias Ayuda

Paciente Activo

SIP 2151181

FUENTES MUÑOZ, DAVID
F. nacimiento 14/05/1979 ♂

Ver datos administrativos

Historial Radiológico

Historial Clínico

Buscar otro paciente

Bandeja de Exploraciones Pendientes

Mis Exploraciones

Exploraciones por Modalidad

Exploraciones por Agenda

Grupos funcionales

Informes Remotos

Exploraciones desechadas

Informes Realizados

Mis Informes

Búsqueda de Informes

Gestionar entregas

Comparación de Pruebas

Historial Radiológico

Exploraciones Desechadas

Exploraciones	Prestacion	Asignada A	Fecha Realizacion	Num. Cita
La semana Pasada				
DILATACIÓN DE ESTENOSIS ...	RADIOLOGICO		28/04/2011 13:21	1015631
DACRIOCISTOGRAFÍA	RADIOLOGIA		29/04/2011 08:00	1015637
DACRIOCISTOGRAFÍA	RADIOLOGIA		29/04/2011 08:00	1015652
Este mes				
DOPPLER ARTERIAL PERIFÉRI...	RADIOLOGIA		24/05/2011 09:12	1015605
COLANGIOGRAFÍA INTRAVEN...	RADIOLOGIA		24/05/2011 09:12	1015618
SERIE OCLUSIVA	RADIOLOGIA		24/05/2011 09:12	1015621
DACRIOCISTOGRAFÍA	RADIOLOGIA		24/05/2011 09:12	1015629
Año pasado				
DACRIOCISTOGRAFÍA	RADIOLOGIA		07/05/2010 08:50	1015502
COLANGIOGRAFÍA TRANSPAR...	MEDICINA INTERNA		13/05/2010 08:40	1015500
DILATACIÓN DE ESTENOSIS ...	MEDICINA INTERNA		13/05/2010 08:40	1015501
COLANGIOGRAFÍA TRANSPAR...	RADIOLOGICO		20/05/2010 12:59	1015240

Fecha informe	F. exploración ¹	Radiólogo
Año pasado		
15/04/2011	07/05/2010	FUENTES MUÑOZ, DAVID
TC DE CRÁNEO SIN C		
15/04/2011	07/05/2010	FUENTES MUÑOZ, DAVID
15/04/2011	07/05/2010	FUENTES MUÑOZ, DAVID
15/04/2011	26/10/2010	FUENTES MUÑOZ, DAVID
RX DE CRÁNEO, RX ABDOMEN SIMPLE		
15/04/2011	26/10/2010	FUENTES MUÑOZ, DAVID
15/04/2011	26/10/2010	FUENTES MUÑOZ, DAVID
04/11/2010	04/11/2010	PELAEZ NIETO, PEDRO
DACRIOCISTOGRAFÍA		
15/04/2011	04/11/2010	FUENTES MUÑOZ, DAVID
DACRIOCISTOGRAFÍA		
15/04/2011	04/11/2010	FUENTES MUÑOZ, DAVID
15/04/2011	04/11/2010	FUENTES MUÑOZ, DAVID

FUENTES MUÑOZ, DAVID en delegación de JI, JI, JI

11:36:09

Ilustración 6: Pantalla de Informado

- **Administración.**

Modulo donde el administrador podrá gestionar las agendas, definiendo sus horarios y exploraciones que en ellas se realizan. Gestión de usuarios, profesionales, prestaciones y del calendario laboral. Todo lo necesario para desarrollar la actividad del servicio de rayos.

Esta pantalla difiere de la del resto de pantallas ya que tenemos varias partes que gestionar, para poder acceder a cada una de ellas se ha dividido la parte izquierda de la pantalla en un menú donde las funcionalidades más importantes serán; Agendas, Prestaciones, Gestión de Usuarios, Gestión de Profesionales y Calendario Laboral.

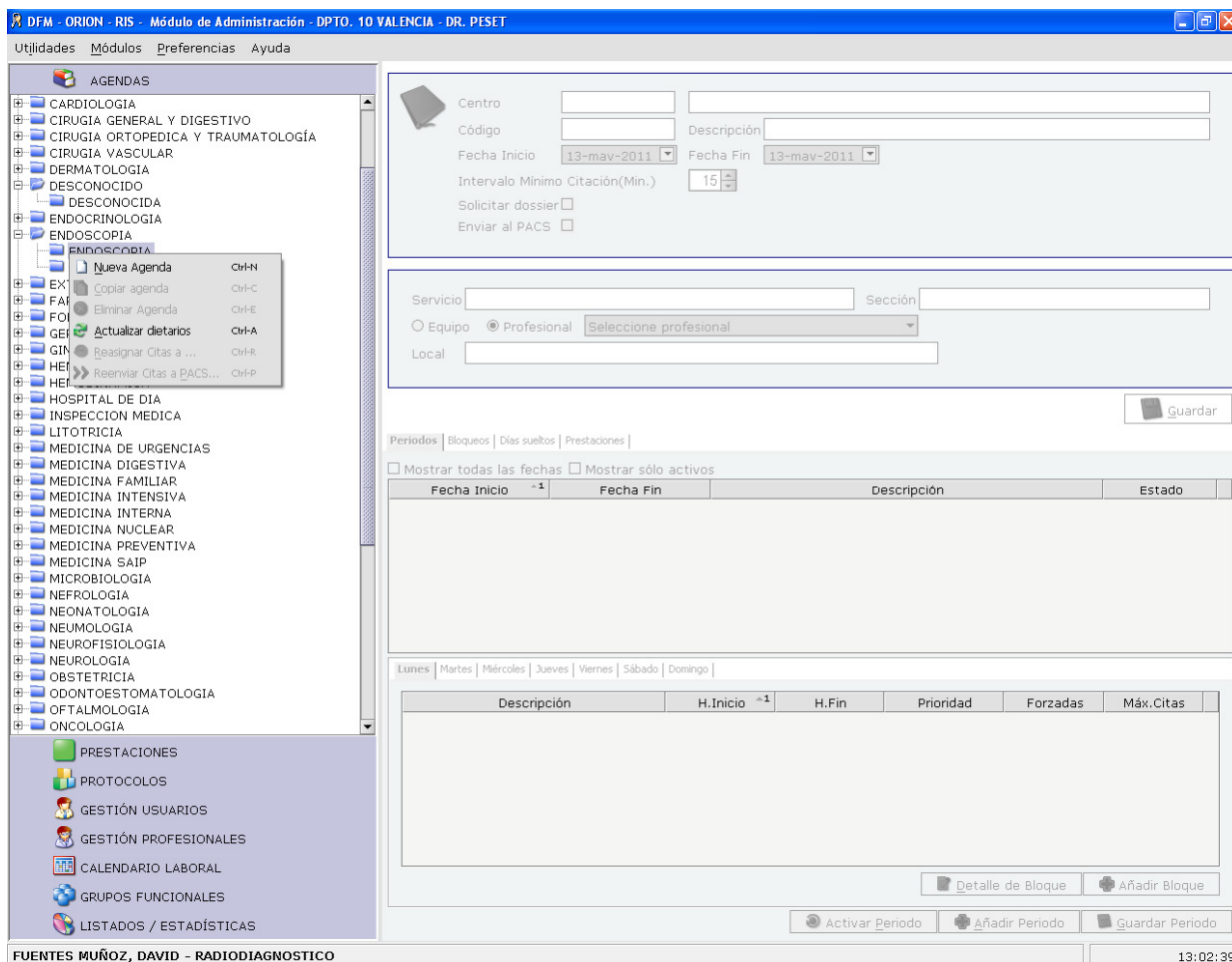


Ilustración 7: Pantalla de Administración

2.2.3 Características de los usuarios

En cuanto a los perfiles de usuarios y las limitaciones que tiene cada uno el sistema ha ido evolucionando desde sus principios. Reflejando cada vez más la realidad del servicio de radiología en las posibilidades que ofrece la aplicación.

Así pues tenemos varios perfiles dentro de la aplicación:

- **Administrativo/a:** Su labor más significativa es la de dar citas al paciente, la mayoría de veces son citas programadas aunque también pueden realizar citas urgentes. También realiza la tarea de gestión de citas, esto es, anulaciones, reprogramaciones y modificaciones. Y por último puede necesitar entrar al modulo de informado para consultar algún informe o imprimir copias si lo necesitara.
- **Técnico radiólogo:** Es un usuario de un perfil técnico el cual se encarga de realizar la exploración al paciente. Por ello ha de saber

manejar y tener conocimientos acerca de la radiología y los aparatos que las realizan. En la aplicación se encargará de citar urgente al paciente, sobre todo aquellos que se encuentran ingresados y que no tienen cita previa. Y para los que vienen con cita previa la tarea será la de capturar la cita, marcándola como realizada.

- **Enfermero/a o A.T.S.:** Este es uno de los perfiles relativamente nuevo ya que en un principio no figuraba dentro de Orión-RIS, pero por petición de algunos hospitales se decidió incorporarlo. Funcionalmente puede realizar las mismas tareas que el técnico radiólogo
- **Radiólogo:** Este usuario es doctor y es el encargado de realizar el diagnostico de las exploraciones realizadas. Firmará sus informes una vez terminados para que este sea enviado a PACS y pueda verse desde las consultas.
- **Facultativo residente:** Es el radiólogo recién titulado. Estará como residente entre dos y tres años. Puede crear informes pero no podrá firmar. La firma la realiza el radiólogo más experimentado capacitado para ello.
- **Facultativo residente con firma:** En este caso el residente si tiene la opción de firmar.
- **Administrador:** Es el usuario que se encarga de gestionar las agendas, prestaciones, usuarios y calendario laboral. La parte que más tiempo ocupa al administrador es la gestión de las agendas ya que se tienen que adaptar a las necesidades de la carga de trabajo. El resto de opciones, una vez realizado los ajustes al comienzo de la implantación, no son muy variables. Este perfil puede ser llevado a cabo por cualquier persona del servicio de radiología con ciertas aptitudes para el manejo de las aplicaciones y, muy importante, que sepa muy bien el funcionamiento del servicio y del personal que en él trabaja.

2.2.4 Restricciones

En este apartado vamos a ver algunas de las limitaciones que nos podemos encontrar, limitaciones que pueden ser de distinta naturaleza.

- **Java:** Para la ejecución del cliente Orión-RIS utilizamos una API de Java propietaria de IBM, la versión 1.5.

Hemos detectado que si intentamos lanzar el comando `javaws` de la jre de alguna versión de java superior a la 1.5 ésta descarga

podría dar error. Así que para asegurar la correcta descarga y ejecución de la aplicación utilizaremos la javaws que está dentro de la jre de ibm que Steria facilita a cada hospital.

- **Hardware:** Los monitores deben de poder mostrar una resolución de al menos 1024x1280. Si intentamos ejecutar la aplicación con una resolución menor probablemente algunos componentes de la interfaz se superpongan imposibilitando llevar a cabo la tarea de manera adecuada.

Los puestos clientes normales, aquellos donde trabajan los técnicos y los administrativos, deben de tener una memoria RAM mínima de 1G. Para el caso de los puestos clientes que forman parte de la estación radiológica deben de tener como mínimo 2G de RAM, puesto que en esos equipo corre también el visor de imágenes de PACS junto con Orión-RIS.

- **Limitaciones de Steria:** La parte de desarrollo y mantenimiento de la aplicación es responsabilidad de Steria Ibérica S.A.U. desde el 2006 hasta la fecha.

La gestión de la base de datos Informix de Orión-RIS es responsabilidad de otra empresa, la cual lleva también la gestión de otras BBDD de la Consellería de Sanitat.

La administración del servidor de aplicaciones Websphere la realiza otra empresa diferente. Su tarea es la configuración y el despliegue de las versiones que desde el equipo de Orión-RIS se generan.

- **Comunicaciones:** Cualquier comunicación que se lleva a cabo desde o hacia Orión-RIS se realiza a través del Motor de Integración Rhapsody, que es administrado por la empresa adjudicataria correspondiente.
- **Lenguaje de programación:** El lenguaje de programación utilizado es Java. Utilizando principalmente dos ediciones Java: Java Standard Edition (Java SE) y Java Enterprise Edition (Java EE).

Para el desarrollo de la interfaz se han utilizado componentes de la API SWING también componentes JIDE y JGoodies.

2.2.5 Suposiciones y dependencias

Existen algunos cambios que requerirían ciertas modificaciones para que Orión-RIS pudiera funcionar como hasta ahora. Un ejemplo sería el sistema operativo. Orión-RIS fue concebido para correr sobre un puesto

cliente Windows, aunque la aplicación este desarrollada sobre Java y por este motivo es una aplicación portable, sí que es cierto que hay muchas partes que tienen en cuenta la estructura de directorios de Windows, por ejemplo cuando se descarga la aplicación se realiza en una ruta especificada por el archivo orion.jnlp que es específica de Windows.

En cuanto a los requisitos funcionales de la aplicación cabe señalar que existe un comité de seguimiento el cual recoge las peticiones que los distintos hospitales han ido creando a través del bugzilla, y que una vez reunidos se decide que peticiones se llevan a cabo por ser de interés común para todos los hospitales. Esto marcará que los requisitos funcionales lleven por un camino u otro.

2.2.6 Requisitos futuros

En el escenario actual donde la movilidad de los pacientes por los distintos centros de salud es un hecho, se hace necesaria la posibilidad de acceder a toda la información posible, desde cualquier centro que se necesite.

Actualmente hay un proyecto en marcha llamado HSE (Historia de Salud Electrónica), y que podría resumirse como la centralización de informes de los pacientes.

Es un sistema centralizado de informes radiológicos y clínicos donde se encontrarán alojados todos los informes de los pacientes, de forma que si necesito recuperar información de un paciente concreto solo tendré que solicitar a través de Orión-RIS, éste pasará la consulta al sistema HSE y si localiza algún informe sobre dicho paciente nos facilitará información básica sobre los informes. En consecuencia solicitar a través de Orión-RIS que informe nos interesa, para que HSE nos recupere toda la información del informe elegido.

Para llevar a cabo este cometido será necesario implementar unos servicios web capaz de recuperar el informe completo para mandarlo al sistema centralizado para que éste lo envíe al hospital que lo solicite.

2.3 Requisitos específicos

2.3.1 Interfaces externas

- **Interfaces de usuario:** El usuario interactúa con el sistema Orión-RIS a través de una interfaz software basada en componentes SWING de Java.

- **Interfaces hardware:** El usuario tendrá a su disposición, en el caso de un usuario normal, un teclado especial que tiene la capacidad de leer la banda magnética de la tarjeta SIP, cargando en Orión-RIS los datos del paciente.

Para el caso de un radiólogo, en su estación radiológica, disponen de un dictáfono que utilizará para dictar el diagnóstico final, también podrán firmar el informe digitalmente mediante el lector de chips de tarjeta, que puede encontrarse en el mismo teclado donde está el lector para la banda magnética. Formando parte de esta estación radiológica también encontramos dos monitores especiales para la visualización de las imágenes radiológicas, llamados monitores barco como el del enlace siguiente:

<http://www.barco.com/es/products-solutions/pantallas-monitores-y-estaciones-de-trabajo/pantallas-m%C3%A9dicas/pantallas-de-diagn%C3%B3stico> .

- **Interfaces software:**

Otros componentes software que permiten junto a Orión-RIS ser un sistema completo es por ejemplo la API que permite al dictáfono escribir dentro de un componente SWING y dar opciones de control, tales como play, o stop.

Hay que mencionar también el programa que permite leer el chip de las tarjetas que es el siemens CardOS así como los certificados necesarios que hay que obtener de la pagina web de GVA.

Recordamos como ya comentamos en otros puntos que existen otros sistemas software en los que Orión-RIS se apoya para llevar a cabo su tarea, estos son principalmente, Rhapsody, PACS, HIS y el SIP.

- **Interfaces de comunicaciones:**

Todos los mensajes que Orión-RIS envía y recibe son mensajes XML, y todos ellos se envían y reciben de o hacia el sistema de información Rhapsody, el cual podríamos llamar nuestro traductor para entendernos con el resto de sistemas con los que cooperamos.

2.3.2 Funciones

Pasamos a describir todas las acciones que debe realizar el software, o sea, de qué forma se aceptan y procesan la información de entrada para producir una salida o resultado esperado. Vamos a especificar en cada acción en cuál de las cinco pantallas o módulos disponibles se ejecuta la acción descrita.

Autenticación de usuario:

- **Módulo:** Para acceder a la aplicación.
- **Actor:** Todos los perfiles deben de autenticarse
- **Introducción:** El usuario necesita hacer uso de la aplicación.
- **Entradas:** Hay que introducir el NIF del usuario y una contraseña que desde la respectiva Unidad de Informática se le ha proporcionado.
- **Proceso:** El sistema valida el usuario con el NIF introducido y la contraseña contra el LDAP, si la validación es correcta entonces se busca el usuario con ese NIF en la tabla usuarios de Orión-RIS.
- **Salidas:** Si pasa ambas validaciones el usuario accede a la aplicación con el perfil asignado. Si falla la primera validación se avisa de que ésta no ha sido exitosa, si es la segunda la que falla se informa de que el usuario no existe en Orión-RIS.

Cita automática:

- **Módulo:** Citación
- **Actor:** Administrativo
- **Introducción:** Se necesita citar a un paciente.
- **Entradas:** Hay que introducir el paciente, el servicio o en su caso el centro que demanda la exploración, la exploración o exploraciones y por ultimo y de forma opcional algunos datos de la cita, como por ejemplo la fecha deseada.
- **Proceso:** Con esta información el sistema busca no el hueco más cercano a la fecha deseada o la fecha de hoy, sino el hueco más óptimo para rellenar así de forma efectiva los huecos de la agenda.
- **Salidas:** Una ventana en la que se propone un día y una hora y se solicita la confirmación de la cita o el rechazo para poder buscar de manera manual en un calendario.

Cita manual:

- **Módulo:** Citación
- **Actor:** Administrativo

- **Introducción:** Se necesita citar a un paciente.
- **Entradas:** Hay que introducir el paciente, el servicio o en su caso el centro que demanda la exploración, la exploración o exploraciones y por ultimo y de forma opcional algunos datos de la cita, como por ejemplo la fecha deseada.
- **Proceso:** Con esta información el sistema busca no el hueco más cercano a la fecha deseada o la fecha de hoy, sino el hueco más óptimo para rellenar así de forma efectiva los huecos de la agenda. En este caso y a diferencia del anterior el administrativo rechazará la propuesta del sistema e ira de forma manual al calendario para elegir un día y una hora que se ajuste al paciente.
- **Salidas:** Una ventana en la que se confirma el día y la hora de la cita y donde se puede aceptar o cancelar la cita ya generada.

Cita Urgente:

- **Módulo:** Citación.
- **Actor:** Administrativo, técnico radiólogo o enfermero.
- **Introducción:** El paciente se encuentra preparado para realizarle la prueba y hay que registrar en ese mismo instante los datos necesarios para su realización.
- **Entradas:** Hay que introducir el paciente, el servicio o en su caso el centro que demanda la exploración, la exploración o exploraciones y por ultimo y de forma opcional algunos datos de la cita, en este caso la fecha deseada no tiene sentido pues la cita urgente es para el mismo instante en el que se realiza
- **Proceso:** Se genera un cita urgente y en una ventana se solicita algunos datos más; que técnico o enfermero es el responsable de realizar la prueba y qué servicio, o si se quiere especificar más, que radiólogo informará la exploración.
- **Salidas:** Una última ventana que pide confirmación de la cita o en su defecto la cancelación de la misma.

Anular cita:

- **Módulo:** Gestión de Citas.
- **Actor:** Administrativo.

- **Introducción:** Después de dar una cita, hay que cancelarla por el motivo que sea.
- **Entradas:** Hay que seleccionar la cita que queremos anular, para ello tenemos dos opciones; buscarla por paciente o buscarla por agenda. Una vez encontrada, sobre la cita botón derecho y en el menú contextual elegimos la opción de “Anular cita”.
- **Proceso:** El sistema procede a cambiar el estado de la cita de introducida a anulada
- **Salidas:** La cita cambia del estado introducida a anulada. Lo podemos ver en una de las columnas de la cita, si la hemos buscado por paciente o en el Tooltip si la hemos buscado por agenda.

Reprogramar cita:

- **Módulo:** Gestión de Citas.
- **Actor:** Administrativo.
- **Introducción:** Necesitamos modificar el día y/u hora de una cita ya introducida en el sistema.
- **Entradas:** Hay que seleccionar la cita que queremos anular, para ello tenemos dos opciones; buscarla por paciente o buscarla por agenda. Una vez encontrada, sobre la cita botón derecho y en el menú contextual elegimos la opción de “Reprogramar cita”, después tenemos que seleccionar la agenda en la que queremos, puede ser la misma u otra, y después dentro del calendario elegir día y hora.
- **Proceso:** El sistema guarda la nueva hora y/o día de la cita.
- **Salidas:** Una ventana donde nos pregunta si queremos imprimir la cita modificada.

Capturar una cita:

- **Módulo:** Captura
- **Actor:** Técnico radiólogo o enfermero (A.T.S.)
- **Introducción:** Llega el paciente para realizarse la prueba y hay que confirmar que se le ha realizado la prueba.

- **Entradas:** Seleccionamos una agenda de las disponibles, elegimos la fecha, sacamos el listado de citas, seleccionamos una cita y sobre ella el botón derecho, pulsamos la única opción que hay, “Capturar cita” y si todo esta correcto confirmamos la operación.
- **Proceso:** El estado de la cita cambia de introducida a realizada.
- **Salidas:** Vemos como el color de la cita cambia de negro a naranja y una de las columnas cambia de citada a capturada.

Modificar una cita:

- **Módulo:** Captura
- **Actor:** Técnico radiólogo o enfermero (A.T.S.)
- **Introducción:** Llega el paciente para realizarse la prueba y se necesita modificar las pruebas que se iban a realizar, añadir o eliminar.
- **Entradas:** Seleccionamos la cita que deseamos modificar, seleccionamos la opción del menú contextual “Modificar cita”. Si queremos eliminar hay que marcarla como “No realizada” y si queremos añadir alguna debemos añadirla pulsando el botón “+”.
- **Proceso:** La cita ahora puede contener nuevas pruebas o las pruebas que tenía pueden estar marcadas como “No realizada”.
- **Salidas:** Se guarda la cita con el resultado de la modificación y se queda preparada para la captura.

Anular actividad:

- **Módulo:** Captura
- **Actor:** Técnico radiólogo o enfermero (A.T.S.)
- **Introducción:** Ninguna de las pruebas se va a realizar bien por error, bien por inasistencia del paciente o bien porque no procede.
- **Entradas:** Seleccionamos la cita que deseamos modificar, seleccionamos la opción del menú contextual “Modificar cita”. Hay que marcar cada una de las pruebas como “No realizada” y confirmar esta modificación.

- **Proceso:** La cita tiene marcadas todas sus pruebas como “No realiza”.
- **Salidas:** Se guarda la cita con el resultado de la modificación y se queda preparada para la captura. Este tipo de citas también deben quedar capturadas.

Crear informe:

- **Módulo:** Informado
- **Actor:** Radiólogo, facultativo residente, residente con firma.
- **Introducción:** Una vez realizada la prueba el radiólogo la tiene disponible para informar.
- **Entradas:** El radiólogo deberá de seguir estos pasos; primero buscar la exploración que quiere informar, sobre la exploración botón derecho del ratón y del menú contextual seleccionamos “Nuevo informe”, se abre el editor y el radiólogo puede editar el informe, una vez finalizado lo guarda y si quiere confirmar el diagnóstico lo debe firmar.
- **Proceso:** El sistema guarda un informe. Si lo ha firmado se guarda también la firma digital y se envía el informe a PACS.
- **Salidas:** El informe desaparece de la bandeja de exploraciones y ahora aparece un informe en la bandeja de “Mis informes” del radiólogo.

Buscar prueba:

- **Módulo:** Informado
- **Actor:** Radiólogo, facultativo residente, residente con firma.
- **Introducción:** El radiólogo pretende informar una prueba ya capturada y se dispone a buscarla.
- **Entradas:** El radiólogo dispone de varias formas para encontrar la prueba, las más utilizada es la que utiliza algún identificador del paciente; NHC, SIP, NIF o nombre y apellidos. Tiene sin embargo otras posibilidades, por agenda, por tipo de prueba o por modalidad.
- **Proceso:** Una vez elegido la forma de buscar la prueba, filtramos y buscamos en la bandeja de entrada la prueba que queríamos.

- **Salidas:** Un numero de pruebas cuyo patrón coinciden con el filtro utilizado.

Consultar informes:

- **Módulo:** Informado
- **Actor:** Radiólogo, facultativo residente, residente con firma.
- **Introducción:** El radiólogo pretende consultar un informe ya realizado.
- **Entradas:** Hay un botón “Búsqueda de informes” que nos abrirá un dialogo con múltiples campos para realizar la búsqueda.
- **Proceso:** Una vez elegido la forma de buscar el informe o los informes, filtramos y buscamos en la bandeja de entrada el informe o informes que queríamos.
- **Salidas:** Un número de informes cuyo patrón coinciden con el filtro utilizado.

Anular informe:

- **Módulo:** Informado.
- **Actor:** Radiólogo.
- **Introducción:** Hay un informe erróneo y tenemos que anularlo.
- **Entradas:** Se busca el informe que se pretende anular, o en la bandeja personal de “Mis informes”, éstos aparecen por orden cronológico. Otra alternativa sería buscar por paciente. Una vez encontrado se selecciona se pulsa el botón derecho del ratón y se selecciona la opción de “Anular informe”.
- **Proceso:** El informe cambia de estado a informe anulado.
- **Salidas:** El informe quedará sin validez clínica, no pudiéndose imprimir ni ver el contenido del mismo.

Crear agenda:

- **Módulo:** Administración.
- **Actor:** Administrador.

- **Introducción:** Vamos a crear una agenda para poder introducir citas en ella.
- **Entradas:** Elegiremos la opción de crear nueva agenda del menú contextual. Introduciremos los datos que definen la agenda y definiremos unos periodos y unos tramos horarios.
- **Proceso:** El sistema creará unos dietarios, resultado de la definición de unos tramos horarios y su activación.
- **Salidas:** Una agenda con un calendario disponible para citar.

Crear una prestación:

- **Módulo:** Administración.
- **Actor:** Administrador.
- **Introducción:** Necesitamos crear una nueva prestación/exploración que se empezará a realizar en nuestro centro.
- **Entradas:** Rellenaremos el formulario con los datos básicos de la prestación, y asociaremos nuestra prestación a una prestación de catalogo.
- **Proceso:** El sistema guarda la prestación.
- **Salidas:** Prestación disponible para introducirla en una cita.

Dar de alta un usuario:

- **Módulo:** Administración.
- **Actor:** Administrador.
- **Introducción:** Un nuevo usuario necesita trabajar con Orión-RIS.
- **Entradas:** Dar de alta un nuevo usuario requiere rellenar los datos básicos del formulario; nombre, apellidos, DNI y una de las partes más importantes, el perfil de usuario. También es aquí donde le damos acceso a los módulos que necesite, así como el módulo de inicio que se abrirá al arrancar la aplicación.
- **Proceso:** El sistema da de alta un usuario y si este es radiólogo también da de alta un profesional asociado al usuario.

- **Salidas:** El usuario podrá entrar en Orión-RIS y empezar a trabajar.

Crear el calendario laboral:

- **Módulo:** Administración.
- **Actor:** Administrador.
- **Introducción:** Antes de empezar a definir agendas es necesario crear al menos el calendario laboral del año en curso.
- **Entradas:** Creamos un nuevo calendario e introducimos los días festivos y su definición, podemos marcar los días como “siempre festivos” para que el próximo calendario herede los festivos.
- **Proceso:** El sistema crea un calendario laboral disponible para crear agendas citables.
- **Salidas:** Calendario laboral con sus festivos correspondientes.

2.3.3 Requisitos de rendimiento

Para un mejor rendimiento se han cacheado ciertas estructuras de datos que el sistema utiliza muy frecuentemente y que sería bastante pesado si cada vez que fueran requeridas se efectuara una llamada al servidor con la correspondiente consulta a la base de datos y generación de la estructura de datos solicitada. Tenemos en nuestro sistema dos niveles de cacheo, en el cliente en el que se carga cada vez que arranca una nueva instancia de Orión-RIS y otro nivel en el lado del servidor, este nivel permanece en el servidor y facilita la estructura solicitada a cada instancia de cliente que la solicite.

Por poner unos ejemplos, una de las estructuras que se mantiene es el conjunto de prestaciones que cada sección puede citar, otra sería la estructura departamental de la Agencia Valenciana de la Salud.

La aplicación no debería de tardar más de un minuto en cargar todas estas estructuras en la cache si ésta está vacía.

En el apartado de las comunicaciones con otros sistemas, el escenario que implica más intercambio de mensajes e involucra a más sistemas no debería de tardar más de dos segundos en dar una respuesta.

Este escenario es el que comienza con la búsqueda por SIP de un paciente, este paciente no se encuentra en nuestra base de datos así que tiene que consultar al sistema SIP si se encuentra el paciente con el SIP

indicado, si la respuesta es afirmativa nos habrá devuelto los datos del paciente, así que con esos datos solicitamos al HIS que nos de de alta un NHC (dato propio de cada centro) provisional para poder citarlo, una vez recibido el nuevo NHC Orión-RIS lo da de alta en nuestro sistema y se podría citar al paciente.

2.3.4 Restricciones de diseño

Puesto que es imposible que una misma aplicación se ajuste a 20 hospitales diferentes, cada uno con distintas formas de trabajo, se creará una tabla donde se podrá parametrizar la aplicación y así poder personalizarla, activando o desactivando algunas funcionalidades a gusto de cada uno de los hospitales.

Por poner un ejemplo, se puede ocultar la propuesta de cita que el sistema realiza al usuario que está citando haciendo que este vaya directamente al calendario para elegir manualmente el día y la hora.

2.3.5 Atributos del sistema

Podemos decir que es un sistema muy fiable y robusto gracias en parte a que cada centro tiene el servidor de aplicaciones en una máquina que está dentro de un clúster. Lo que podemos esperar de un sistema clúster es:

- Alto rendimiento
- Alta disponibilidad
- Balanceo de carga
- Escalabilidad

Además la arquitectura de servidor, basada en la especificación J2EE garantiza un funcionamiento idóneo para un sistema de esta envergadura. Esta arquitectura también proporciona facilidad a la hora de mantener la aplicación debido a la división por capas del servidor.

Para dotar de una mayor fiabilidad Consellería apuesta por un motor de base de datos de garantías como es Informix, un servidor de aplicaciones como Websphere de IBM y todo esto trabajando en el marco de una red propia de Sanidad llamada ARTERIAS, lo cual da una seguridad extra a toda la información que en ella circula.

Otro punto más en cuanto a la seguridad es cada usuario accede a Orión-RIS mediante un usuario que será su NIF y una contraseña proporcionada

por el propio centro. El usuario y la contraseña introducida serán validadas contra un servidor LDAP.

Los radiólogos además tendrán una tarjeta única, personal e intransferible y junto a ella se le facilita un PIN y un PUK, con la que podrán firmar informes que llevarán así su firma digital.

2.3.6 Otros requisitos

Como veremos más adelante será necesario mantener una serie de tablas para garantizar una parte del funcionamiento con respecto a la gestión de los pacientes.

Orión-RIS arrancará con su propia tabla de pacientes pero tenemos que tener en cuenta que en un futuro lo que se pretende es unificar tanto el HIS, el RIS y farmacia en Orión-Clínic, por lo que se quiere es mantener la información de paciente totalmente sincronizada en ambos sistemas, para que cuando se lance el sistema Orión-Clínic con una única tabla de pacientes no haya problemas con la información de RIS con respecto a la del HIS.

Para esta tarea se desarrolla una serie de mensajería intercambiada siempre por medio de Rhapsody, y que conseguirá que cada vez que se realice una operación sobre un paciente, ya sea alta , baja o modificación ésta se mande a través del middleware para sincronizar las tablas de pacientes de ambos sistemas.

Orión-RIS por su parte remitirá esta información a PACS, para que también este tenga la información referente a los pacientes actualizada.

3 Análisis

La fase de análisis es la parte en la que construimos el modelo del sistema que queremos desarrollar. Este modelo puede construirse a partir de una serie de modelos, cada uno con un objetivo que describe diferentes características. Obteniendo así un modelo completo y comprensible desde diferentes puntos de vista.

Tenemos el diagrama de clases; un diagrama estático que define la estructura del sistema, los casos de uso; que describen el comportamiento, el diagrama de secuencia; utilizado para definir la interacción entre los objetos del sistema. Todos estos diagramas se construirán basándose en una notación estándar, el lenguaje UML (Unified Modeling Language).

En este documento utilizaremos los diagramas de caso de uso, para mostrar la relación de los actores con los requisitos funcionales del sistema.

3.1 Caso de uso del administrativo

Como muestra el diagrama estas son las acciones que puede realizar el administrativo, todas referentes a la cita, será el usuario que debe acordar con el paciente el día y la hora que se realizará la prueba.

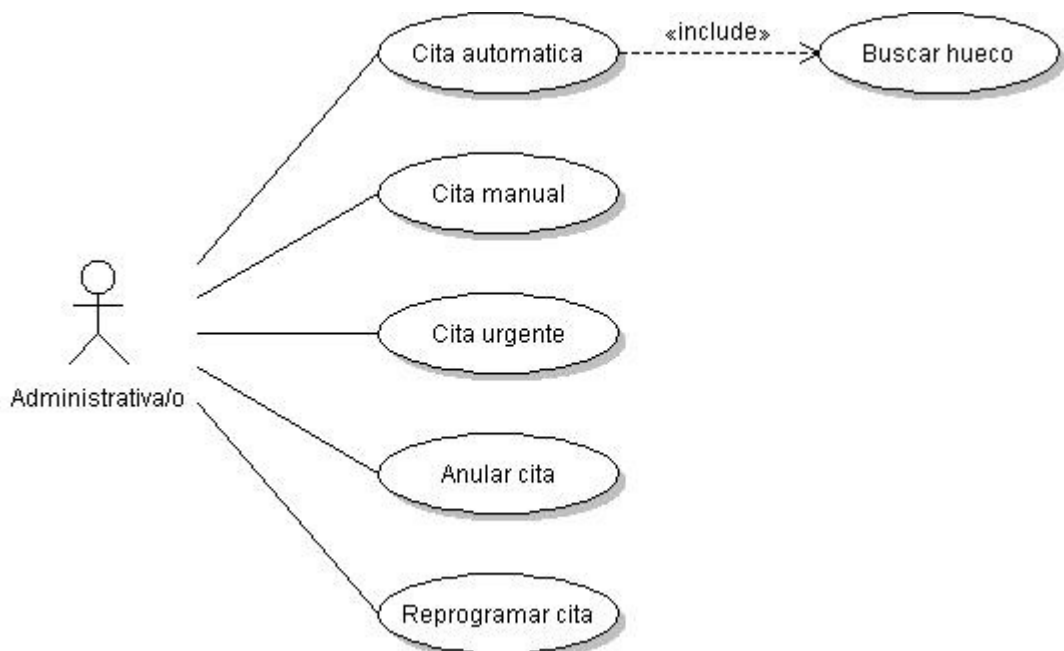


Ilustración 8: Caso de Uso del perfil Administrativo

3.2 Caso de uso del técnico radiólogo/A.T.S

Este que es un perfil más técnico, tiene que manejar también la maquina que ejecuta las radiografías, ecos o resonancias. Es importante recordar que hasta que la actividad no esté capturada no aparecerá en la bandeja de entrada de pruebas pendientes de informar del radiólogo.

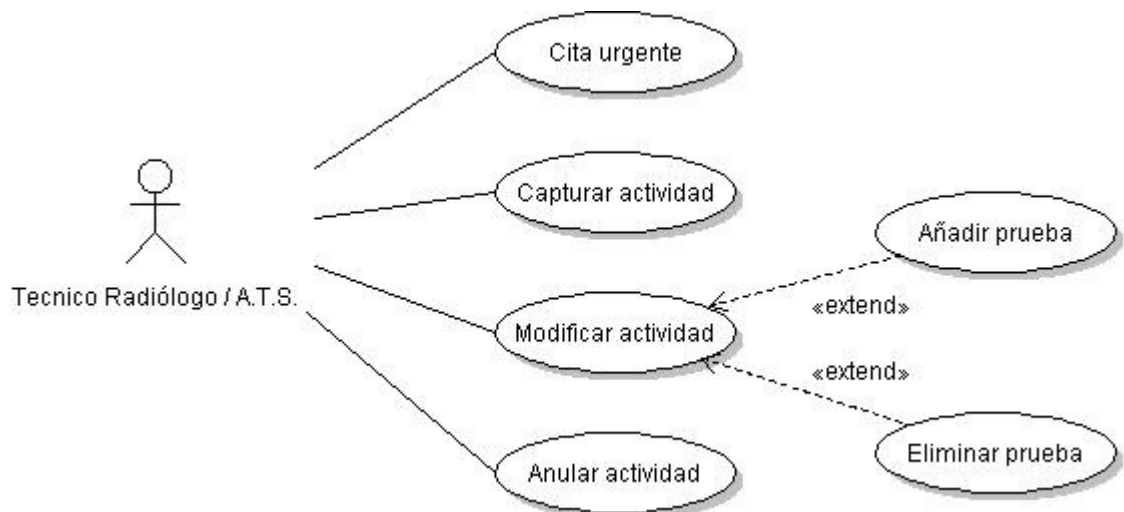


Ilustración 9: Caso de Uso del perfil de Técnico Radiólogo

3.3 Caso de uso del radiólogo

Sin salir de la misma pantalla el radiólogo puede buscar pruebas e informes, crear informes y firmarlos digitalmente, todo ello con la ayuda de los monitores barco y el dictáfono. Todo este conjunto de recursos es lo que llamamos estación radiológica.

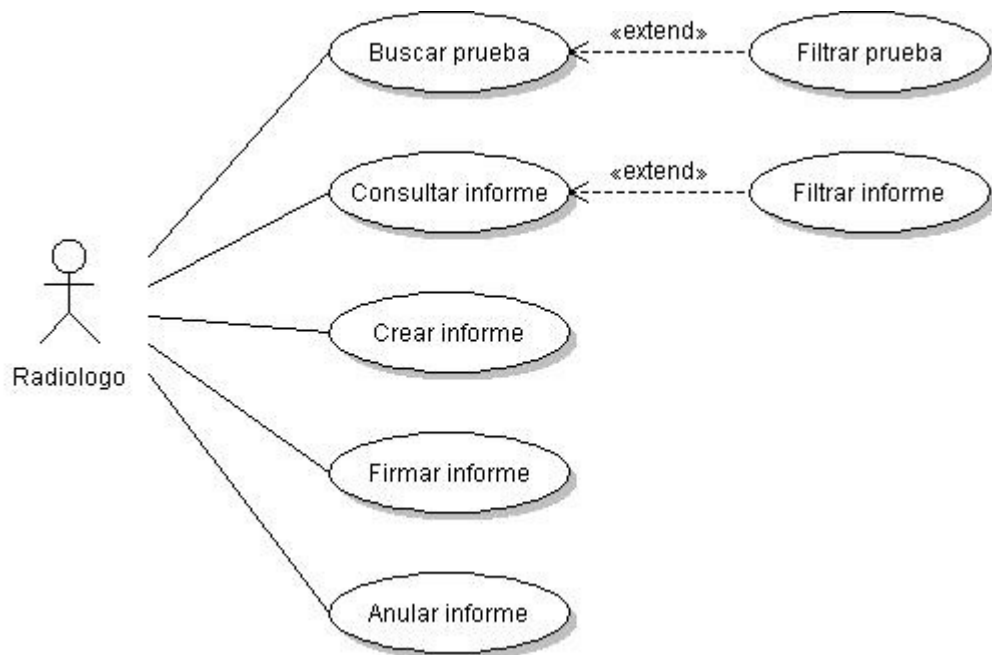


Ilustración 10: Caso de Uso del perfil de Radiólogo

3.4 Caso de uso del administrador

El usuario administrador puede ser cualquier usuario de la aplicación, lo único que se le pide es que tenga un gran conocimiento del servicio de rayos y de la forma de trabajo así como de los diferentes usuarios. También es importante que sea capaz de mantener una comunicación fluida con el resto de responsables del servicio para el buen funcionamiento del mismo.

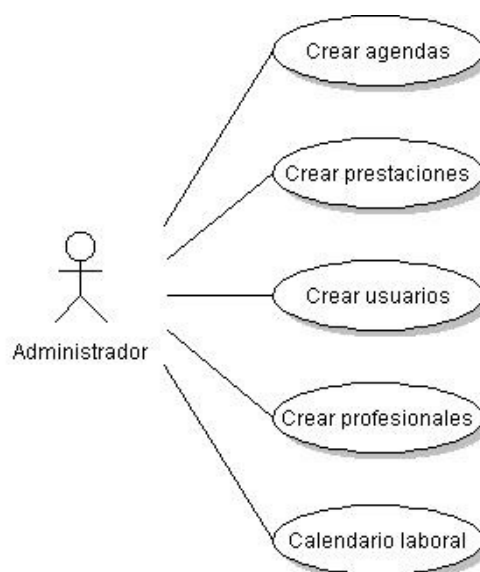


Ilustración 11: Caso de Uso del perfil de Administrador

4 Diseño

4.1 Aspectos generales

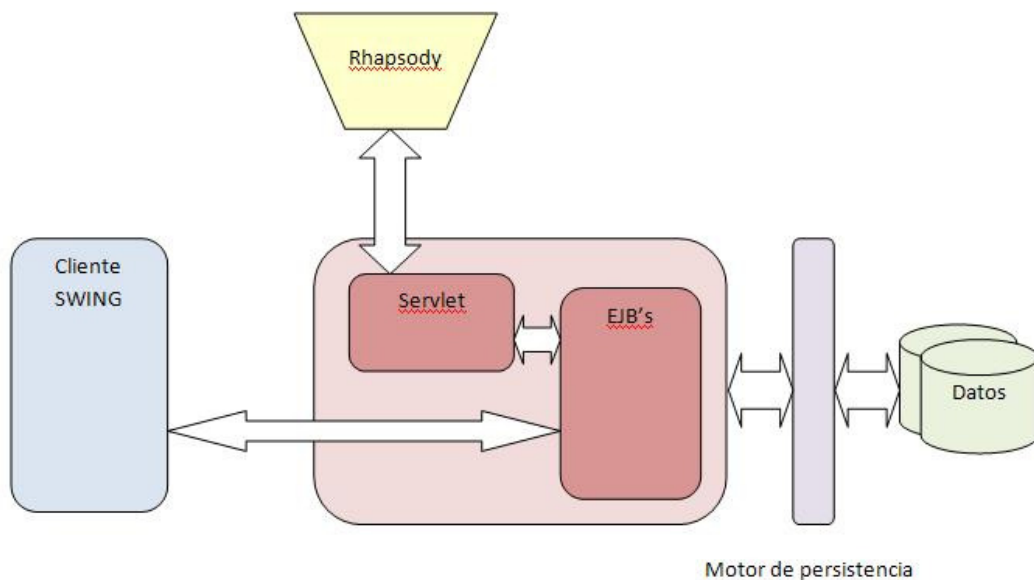


Ilustración 12: Arquitectura de Orión-RIS

Vamos a desgranar los aspectos de diseño del aplicativo Orión-RIS. Una aplicación que se basa en la especificación J2EE. Para ello nos hace falta un servidor que pueda correr un servidor de aplicaciones J2EE, en nuestro caso se opta por maquinas UNIX y el servidor de aplicaciones WebSphere.

Las ventajas de utilizar una solución basada en la especificación J2EE son varias:

- Soporte de múltiples sistemas operativos: Al ser una plataforma basada en el lenguaje Java, es posible desarrollar arquitecturas basadas en J2EE utilizando cualquier sistema operativo donde se pueda ejecutar una maquina virtual Java.
- Organismo de control: La plataforma J2EE está controlada por el JCP, un organismo formado por más de 500 empresas. Entre las empresas que lo forman están las más importantes del mundo informático (SUN, IBM, Oracle, SPA, HP, AOL, etc.) lo que la apuesta que se ha realizado por esta tecnología en los últimos años y el interés para que sea un éxito y un estándar de referencia..

- Facilidades para el desarrollador: La existencia de contenedores web, y contenedores EJB's hace que el desarrollador se centre mas en el problema de desarrollo y no en aspectos de más alto nivel ya que estos contenedores proveen al desarrollador de una serie de tareas automáticas y una serie de herramientas y API's para llevar a cabo el desarrollo de manera más rápida y eficaz.
- Arquitectura N-capas: Hace que el mantenimiento de la aplicación sea mucho más sencillo, más legible y comprensible.

En el lado del cliente, como ya sabemos, hemos utilizado la tecnología SWING de Java para construir nuestra interfaz de usuario. Una solución aceptada por sus ventajas:

- Es una opción más rápida en respuesta que la de una aplicación web.
- También suele ser ésta una solución más robusta y fiable que las aplicaciones web.
- Y pueden ser más seguras algo que en el entorno en el que trabajamos y con la información que se maneja es algo muy importante.
- Está basada únicamente en Java, por lo que un programador Java puede perfectamente desarrollar componentes SWING. En el entorno web el programador tiene que lidiar con más de un lenguaje, HTML, CSS, JavaScript, PHP, etc.

4.2 Interfaz de usuario

La interfaz estará dividida por cinco pantallas, en cada una de ellas el usuario podrá llevar a cabo su tarea sin salir de la misma.

- **Citación:** En esta pantalla el usuario tiene disponible todos los componentes y recursos necesarios para dar o buscar la información requerida para poder dar una cita ya sea automática, manual o una cita urgente.
- **Gestión de citas:** En estas pantallas los usuarios pueden buscar toda la información referente a las citas y realizar si se desea anulaciones reprogramaciones o modificaciones de cierta información de la cita, como por ejemplo quien la demanda.
- **Captura de actividad:** En esta pantalla el técnico sacará el listado de las citas por aparato, e irá seleccionando una a una conforme vayan pasando los pacientes y se vayan ejecutando las distintas pruebas.

- **Informado:** EL radiólogo dispone de una bandeja de entrada de pruebas pendientes de informar, como si fuera la bandeja del correo electrónico. El dispone de varios filtros que puede utilizar. Desde esta pantalla también puede consultar informes, crear nuevos y firmarlos.
- **Administración:** En una misma pantalla tiene todas las opciones de gestión de la aplicación, agendas, prestaciones, usuarios, profesionales y el calendario laboral.

Para acceder a los servicios del servidor se construirá una capa de acceso, una por cada caso de uso.

4.3 Servidor

En el lado del servidor hemos utilizado unas capas que separan la lógica de la aplicación, de otra capa en la que se realiza directamente, mediante el uso de la persistencia JDO, el acceso a los datos, esto es, obtención, eliminación y modificación de información de la base de datos.

Así pues el servidor tendrá dos niveles de interfaces, una para permitir el acceso desde el cliente y la segunda para permitir el acceso de los servicios a las operaciones de persistencia, creando así una capa de separación que hará que el servidor tenga una estructura más ordenada y fácil de mantener.

Como se aprecia en la *Ilustración 12: Arquitectura de Orión-RIS*, el servidor dispone de un servlet que principalmente atiende los mensajes que provienen del HIS, son mensajes que pasan por el middleware se transforman a mensajes XML y se envían a la instancia de Orión-RIS que toca según el entorno en el que nos encontramos.

Una vez recibido el mensaje XML se pasa al servicio correspondiente dentro del contenedor EJB y desde allí se atiende la petición, se realizan las operaciones correspondientes y se notifica a PACS la operación que se acaban de realizar.

El resto de operaciones son las que provienen del usuario normal del servicio de rayos son peticiones que atacan directamente a los servicios que se encuentran en el contenedor EJB, sin pasar por servlet.

4.4 Motor de persistencia

El acceso a un sistema de datos persistente, como el de una base de datos relacional, es un punto crítico en el diseño de nuestra aplicación por ser esta decisión clave para un funcionamiento óptimo de nuestro producto.

Tenemos que tener en cuenta para realizar una correcta elección que nuestra base de datos es una base de datos de más 90 tablas con complejas

relaciones entre sí. Así que elegir un acceso mediante JDBC sería muy costoso para el programador, es una opción válida para aplicaciones pequeñas con pocos accesos a la base de datos, pero muy costosa cuando es una aplicación grande con muchos accesos a la base de datos.

Un framework ORM te permite gestionar la concurrencia y las transacciones de forma automática, y permite construir un acceso a los datos de forma clara y ordenada, y eso en una aplicación grande es algo imprescindible.

El motor de persistencia elegido para la puesta en marcha de este proyecto es el motor de persistencia XALIA LIDO que es un producto JDO, o una implementación de la especificación JDO. XGALIA LIDO implementa de manera completa el estándar JDO 1.0. En el año que se decide el uso de este motor de persistencia JDO es la opción más completa y más estable del momento.

4.5 Base de datos

La base de datos constara de algunas tablas maestras cuya información será la misma para cada uno de los centros donde la aplicación se ponga en marcha, tales como:

- calles: Nombre de todas las calles de la comunidad valenciana
- municipios: Municipios codificados.
- prestaciones_catalogo: Que consta de un catalogo corporativo de pruebas de la Agencia Valenciana de la Salud.
- servicios_crc: Códigos y descripciones de los servicios corporativos.
- departamentos: Tiene la información codificada de todos los departamentos de la Comunidad Valenciana.

Otro conjunto de tablas se copian de los sistemas HIS de los hospitales para mantener y no perder cierta información acerca de los pacientes, tales como:

- compensables_seguibles: Mantiene la información de aquellos pacientes a los que se les ha atendido pero que pertenecen a otro centro.
- fusiones_pacientes: Mantiene información acerca de un procedimiento muy normal en los hospitales como es el de fusionar un número de historia provisional con uno definitivo.

A continuación vamos a representar las relaciones más importantes que existen en nuestra base de datos:

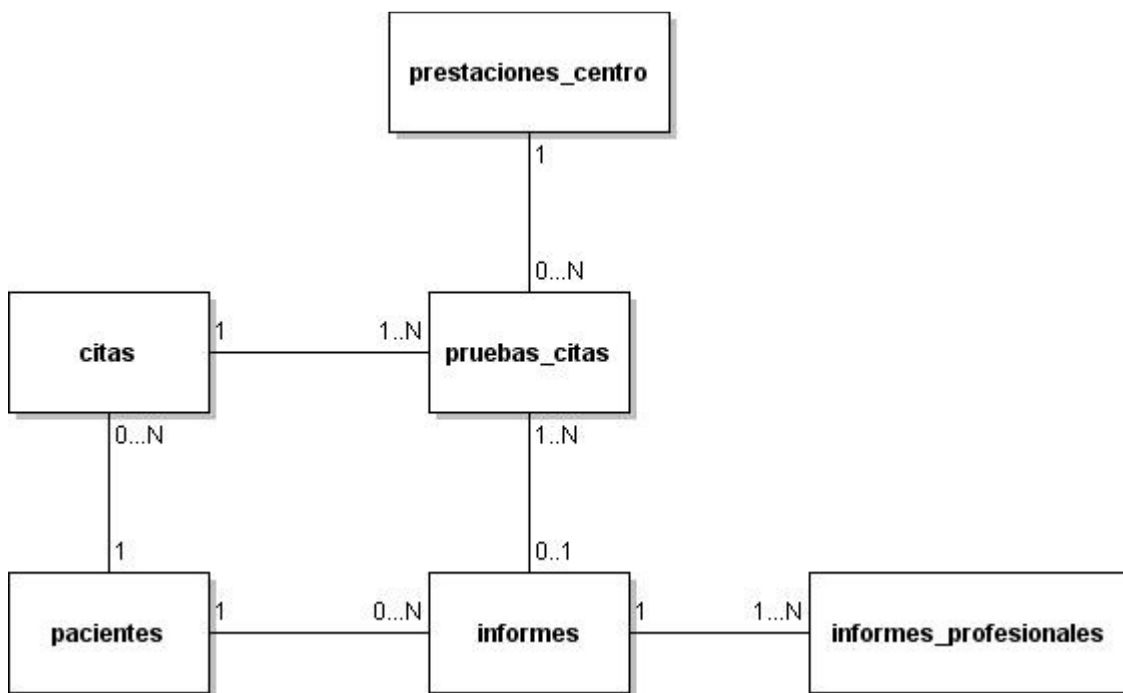


Ilustración 13: Relación de las principales tablas.

Podemos decir que estas tablas y sus relaciones representan la base de Orión-RIS, los pacientes, sus citas compuestas de pruebas_citas, los informes que se realizan sobre dichas pruebas_citas son la información más importante.

Es importante la relación entre citas y pruebas_citas para poder dar citas con más de una prueba y porque como hemos visto anteriormente se pueden hacer cancelaciones o modificaciones tanto a nivel de cita como a nivel de exploración (pruebas_citas).

La relación entre informes e informes_profesionales indica que un informe puede tener varios autores, o sea, que más de un radiólogo puede contribuir en la redacción del informe, sin embargo el radiólogo que firma el informe solo puede ser uno, y éste quedará registrado en un campo de la propia tabla informes.

5 Implementación

5.1 Introducción

Este punto tratará de explicar sobre aspectos de la configuración de la aplicación y algunos ejemplos de implementación con código, explicará el uso de algunos patrones de diseño que se han utilizado como solución a problemas de desarrollo software que aparecen en un contexto similar.

Y se darán ejemplos también de alguna entidad persistente para comprobar su implementación.

5.2 Interfaz de usuario

En lado del cliente hacemos uso de algunas soluciones aportadas por los patrones de diseño.

Singleton: Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Muchas clases cliente quieren usar varias clases servidoras, y deben saber cuál es exactamente la que le proporciona cada servicio. El sistema se volvería muy complejo, porque habría que relacionar todas las clases cliente con todas y cada una de las clases servidoras.

Como en Orión-RIS, cada pantalla que tenemos accede mayormente a un conjunto muy concreto de servicios del servidor se implementará un singleton por cada caso de uso.

Tendremos al final un único objeto por cada caso de uso que dará acceso a los servicios relacionados con ese caso de uso.

Para el de citación por ejemplo tenemos la clase CitacionDelegate en la que se implementa el siguiente método que es clave para el correcto funcionamiento del singleton.

```
public synchronized static CitacionDelegate getInstance()
throws Exception {
    if (citacionDelegate == null) {
        citacionDelegate = new CitacionDelegate();
    }
    return citacionDelegate;
}
```

Como se puede observar en la lógica del método, para obtener la instancia de ésta clase hay que invocar al método getInstance(), si existe ya una, te la devolverá y sino creará una nueva consiguiendo así que en todo momento exista una única instancia de esta clase.

Una vez obtenida la instancia podremos llamar a los métodos que nos dan acceso a los servicios del servidor para realizar distintas operaciones como:

```
public List retrieveCitasDia(int agendaKey, Date date)
public List retrieveAgendasCitables()
public int saveCita(CitaDTO cita)
public CalendarioAgendaDTO retrieveCargaDeTrabajo
```

Observer: Se trata de un patrón de comportamiento que suele observarse en frameworks de interfaces gráficas orientadas a objetos como SWING. Es una solución al problema que se plantea cuando en una misma pantalla tienes componentes SWING distintos y el comportamiento de uno depende de la información que otro componente obtenga o de como se comporte.

Para que quede algo más claro presento el siguiente diagrama de secuencia:

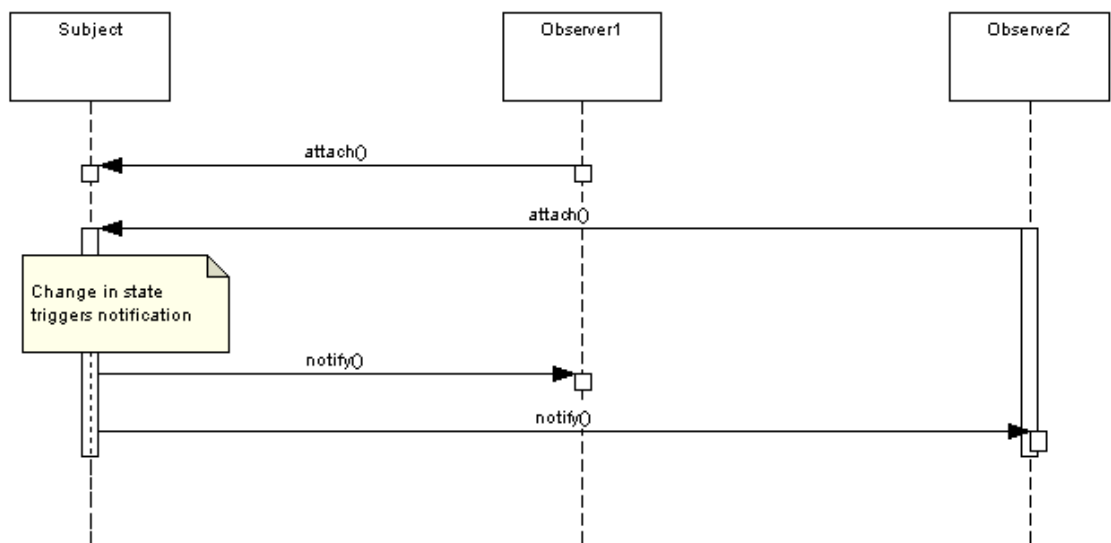


Ilustración 14: Comportamiento del patrón de diseño Observer

En cuanto al comportamiento general del cliente es importante destacar el papel de la clase `UIContext.java`, que es la que facilita la configuración de las distintas instancias dependiendo de las necesidades de cada centro. Así pues esta clase recoge todos los valores de la tabla `configuracion_instancia`. Por ejemplo en la tabla tenemos un parámetro llamado `ocultar_propuesta` cuyo valor puede estar a `true` o `false`.

Por defecto será una propiedad puesta a `false`, para que la aplicación muestre al usuario la propuesta de cita que el sistema a encontrado. Si se quiere ocultar este paso para que el usuario pueda elegir directamente el día y la hora en el calendario de la agenda no hay más que cambiar este valor de la base de datos por `true`.

Se inicializa el valor de la variable en `UIContext.java`

```
ocultarPropuesta = false;
```

Después se recoge el valor que tiene en la base de datos para comprobar si es `true`.

```
if(instanciaOrionProperties.getProperty("ocultar_prop")!=null) {
    if(instanciaOrionProperties.getProperty ("ocultar_prop").equals("true"))
    {
        ocultarPropuesta = true;
    }
}
```

Y más tarde desde la parte del cliente donde se hace uso de la ventana de propuesta se consulta al método de `UIContext.java` qué valor tiene:

```
public boolean isOcultarPropuesta() {
    return ocultarPropuesta;
}
```

5.3 Servidor

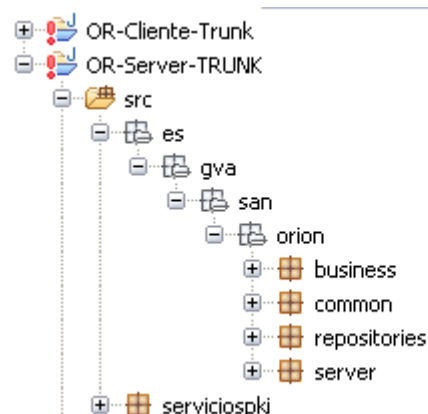


Ilustración 15: Estructura de paquetes en el servidor

La estructura del servidor tiene el aspecto que presenta en la figura de arriba, con cuatro paquetes dentro de “`es.gva.san.orion`” como:

- **business:** Dentro del paquete business tenemos las clases u objetos entidad utilizados por el motor de persistencia para ser mapeados con las tablas de la base de datos.
- **common:** Dentro de este paquete se ubican todas aquellas clases de uso general y compartido por el resto de clases del servidor. Así pues podemos ver en este paquete una clase de constantes como RisConstantes.

También tenemos las clases necesarias para crear la conexión del motor de persistencia y la clase JDOUtil.java que tiene implementado todas las operaciones disponibles y que se resumen en la imagen siguiente:



Ilustración 16: Operaciones de persistencia disponibles.

También disponemos de una serie de clases muy importantes en nuestra aplicación como son las clases DTO, son aquellas clases que no son clases entidades (objetos persistentes) pero las representan en cierta medida, muchas veces con información más acotada, simplificando y disminuyendo el tamaño de estos objetos que son los que viajan del servidor al cliente y viceversa.

Por último en este paquete tenemos la clase `BaseSessionBean` que implementa la interface `SessionBean`, utilizados estos métodos por el contenedor EJB.

- **repositories:** En este paquete y dividido por casos de uso, tenemos por cada caso de uso una interface y la clase java que implementa esa

interface. Estos métodos lo que hacen es formar la JDOQL para generar adecuadamente la consulta, eliminación, actualización o una nueva entrada en la base de datos, a través de los métodos de la JDOUtils.java.

Por ejemplo, un método implementado sería el siguiente:

```
public SituacionPrueba
findSituacionPruebaByCodigoSituacion(PersistenceManager pm, int
codSituacionPrueba) {
SituacionPrueba situacionPrueba = null;
Object[] params = new Object[]{new Integer(codSituacionPrueba) };

List situacionesPruebas = JDOUtil.find(pm, SituacionPrueba.class,
"codigoSituacionPrueba==codSituacionPrueba", null,
"int codSituacionPrueba", params);

if (situacionesPruebas.size() > 0) {
    situacionPrueba = (SituacionPrueba)
    situacionesPruebas.iterator().next();
}
return situacionPrueba;
}
```

- **server:** En este paquete están los servicios o métodos que podemos invocar desde el cliente. Junto con esta implementación encontraremos las interfaces necesarias según la especificación EJB 2.0.

También tenemos el conjunto de clases que hacen posible la comunicación con el motor de integración y que hará posible mandar mensajes tanto al PACS como al HIS y al SIP.

Y luego dentro del proyecto pero en otra carpeta diferente como “es” se encuentra el paquete serviciospki, paquete utilizado para el proceso de firma electrónica.

5.4 Motor de persistencia

Para mostrar cómo hemos solucionado esta parte de la aplicación lo más fácil será ver un ejemplo sencillo de mapeo de un objeto entidad con su correspondiente tabla.

Como hemos comentado antes las clases que representan las tablas se encuentran en “es.gva.san.orion.business”.

Por ejemplo la clase Periodos.java, cuyo significado dentro de la aplicación va ligado al de agenda, ya que cuando se define un horario para la agenda hay que definir un periodo que puede ser desde un día hasta varios años.

Esta clase cuya definición es:

```
public class Periodos implements Serializable {
    ...

    private static final long serialVersionUID = 2765740961462L;
```

Lo primero a comentar aquí es el hecho de que todos los objetos entidad implementarán la interface `Serializable`, esto se realiza de este modo para evitar un funcionamiento no esperado en la gestión de los objetos cuando estos son detachados. Junto con esto, darle un valor a la variable `serialVersionUID` es también importante para el correcto funcionamiento, según la información recogida de http://db.apache.org/jdo/attach_detach.html.

Junto con la clase que representa una de las tablas tendremos dos archivos en la misma ruta, `lido-package.xml` y `package.jdo`.

Dentro de `lido-package.xml` entre otras clases tenemos la que ahora nos interesa que es `periodos`.

```
<class name="Periodos" entity="PERIODOS">
  <primary-key>
    <field name="codigoPeriodo" attribute="PERI_KEY" initial-
      value="database"/>
  </primary-key>
  <field name="fechaFin" attribute="PERI_FECHA_FIN">
    <mapper class="{date}">
      <param key="type" value="date-only"/>
    </mapper>
  </field>
  <field name="situacion" attribute="PERI_SITUACION"/>
  <field name="fechaInicio" attribute="PERI_FECHA_INICIO">
    <mapper class="{date}">
      <param key="type" value="date-only"/>
    </mapper>
  </field>
  <field name="descripcionPeriodo" attribute="PERI_DESC"/>
  <field name="agenda">
    <associate field="codigoAgenda" to-attribute="AGEN_KEY"/>
  </field>
  <field name="bloques">
    <unordered-collection entity="BLOQUES">
      <associate field="codigoPeriodo" to-attribute="PERI_KEY"/>
    </unordered-collection>
  </field>
</class>
```

Este es el mapeo de los campos de la clase con el correspondiente en la base de datos.

Dentro del archivo `package.jdo` tenemos la parte que corresponde a `periodos`:

```
<class name="Periodos" identity-type="application" objectid-
class="PeriodosID">
  <field name="codigoPeriodo" primary-key="true"/>
  <field name="fechaFin"/>
  <field name="situacion"/>
  <field name="fechaInicio"/>
  <field name="descripcionPeriodo"/>
  <field name="agenda"/>
```

```
<field name="bloques">  
  <collection element-type="es.gva.san.orion.business.agenda.Bloque"/>  
</field>  
</class>
```

Esta definición es la que hace que el objeto sea persistente ya que es el objeto que busca el motor de persistencia para confirmar que efectivamente es un objeto persistente.

A partir de este momento ya podríamos recuperar objetos persistentes y trabajar con ellos. Para enviar información de estos objetos al cliente crearemos un objeto DTO con la información necesaria.

6 Valoración

Personalmente me enorgullece la presentación de esta memoria que refleja lo que es la aplicación que ha significado mi primera experiencia laboral cuyo funcionamiento desde su primera implantación en el año 2006 ha sido todo un éxito reconocido por los responsables por parte de Consellería, los cuales destacan el gran rendimiento de la aplicación y la gran labor de el equipo del que formo parte.

Además el conocimiento de la tecnología Java y el desarrollo de aplicaciones como ésta han sido un gran reto y me han valido para mejorar profesionalmente. Así pues ahora se abre un camino por el que me gustaría seguir, que es el del desarrollo de aplicaciones empresariales.

Después de todos estos años, a día de hoy puedo decir que todo este esfuerzo ha merecido la pena.

7 Bibliografía

Keogh, James. *J2EE: Manual de Referencia*. 2003. McGraw-Hill / Interamericana de España.

Stelting, Stephen; Maassen, Olav. *Patrones de Diseño Aplicados a Java*. 2003. Pearson Alhambra.

Elliot, James; Eckstein, Robert; Loy, Marc; Wood, David; Cole, Brian. *JAVA SWING*. 2002. O'REILLY.

<http://docs.oracle.com/javase/tutorial/uiswing/> - Página de referencia para la creación de componentes SWING.

<http://docs.oracle.com/javase/6/docs/api/> - API de JAVA.

<http://www.wikipedia.org/> - Enciclopedia online.

http://www.w3schools.com/xml/xml_xsl.asp - Referencia de tecnologías web, en este caso utilizada para consultar aspectos de XML.

<http://standards.ieee.org/findstds/standard/830-1998.html>

<http://www.theserverside.com>

