



Universidad de las Ciencias Informáticas
Facultad 7

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

TÍTULO: MÓDULO CONSULTA EXTERNA DEL SISTEMA
DE INFORMACIÓN HOSPITALARIA alas HIS

Autores: Arturo Iván Morffe Zaldívar
Leonardo García Ferrer

Tutora: Lic. Dainerys Castañero Rodríguez

Ciudad de La Habana, Junio 2009
“Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 30 días del mes de junio del año 2009.

Arturo Iván Morffe Zaldívar

Leonardo García Ferrer

Lic. Dainerys Castañero Rodríguez

DATOS DE CONTACTO

Graduada de Licenciatura en Ciencias de la Computación en el año 2004 en la universidad Central de las Villas. Actualmente se desempeña como profesor universitario en la Universidad de las Ciencias Informáticas (UCI) donde ha impartido asignaturas como Sistemas de Bases de Datos, Ingeniería de Software y Gestión de Software. Posee categoría docente de Instructor y cursa la maestría de Informática Aplicada. Desde que se incorporó en la universidad ha estado vinculada a actividades productivas en el área hospitalaria, actualmente se encuentra desempeñando el rol de analista en los módulos Citas y Consulta Externa del Sistema de Información Hospitalaria (HIS).

Correo electrónico: dainerysc@uci.cu

AGRADECIMIENTOS

De Arturo:

A mis padres y familiares que me han apoyado en todo momento y han confiado en mí sin importar las condiciones ni las inclemencias y han sabido guiarme por el camino correcto.

A Marisnelys mi primita que cada vez que la necesito aparece de la nada para salvarme.

A los compañeros de grupo que nos han brindado siempre una mano cuando nos ha hecho falta.

Al los profe Angel Fabra y Jublar que nos han apoyado y nos han brindado su ayuda siempre de forma incondicional.

A los muchachones del piquete de Moa que por 5 años hemos compartidos buenos y malos momentos y nos hemos convertido en una gran familia.

A Yanet, que estuvo a mi lado por 4 largos años y estuvo allí en los momentos en que más lo necesitaba.

A todas aquellas personas que de una forma u otra siempre han estado en el momento que mas falta hacen.

De Leonardo:

Primero que todo quisiera agradecerle a mi mamá por lo especial que es para mí, por estar siempre a mi lado y brindarme su amor y apoyo a lo largo de toda mi vida y en especial en estos cinco años de mi carrera, para ella este gran regalo, "Mi Título".

A mi papá y hermanita por estar siempre a mi lado y ser partes importantes en mi vida y en general a toda mi familia que de una forma u otra siempre me apoyaron y ayudaron.

A mi tata Ori que desde que la conocí cambió mi vida totalmente, me apoyó, me guió, me dio su amor, su felicidad y rápidamente formó parte esencial de este sueño logrado.

A todos mis amigos que son tantos que si los menciono a todos nunca acabaría, pero a todos mil gracias especialmente a Darian, William, Franco, Juanma, Yoel,...

DEDICATORIA

De Arturo:

A mis abuelos que no están conmigo en estos momentos, pero siempre me enseñaron a seguir adelante y a saber decidir en la vida.

A mis padres que nunca me han abandonado ni en las buenas ni en las malas y me han apoyado en todo y han confiado en mí siempre.

A mi hermano que me ha servido de ejemplo toda la vida, mi compañero incondicional.

A mis tías Beatriz y Yamilia que me han dado todo el apoyo del mundo.

A Jose Manuel, Yanelis y Marisnelys los primos que más quiero y me hacen sentirme orgulloso de ellos cada día.

De Leonardo:

Dedico este trabajo a toda mi familia especialmente a mi mamá y mi papá por hacer realidad este sueño, a mi novia Ori por estar siempre conmigo, a todos mis amigos por su apoyo y a todos los que confiaron y creyeron en mí,

RESUMEN

El presente trabajo tiene como objetivo desarrollar el módulo Consulta Externa del Sistema de Información Hospitalaria alas-HIS, para facilitar la gestión de información en esta área de las instituciones hospitalarias.

Para la realización del mismo se hizo uso de las herramientas definidas en la arquitectura establecida por el Área Temática Hospitales. El desarrollo estuvo guiado por las especificaciones que propone la metodología RUP, obteniendo los artefactos de los diferentes flujos de trabajo, como el modelado de negocio, análisis y diseño e implementación.

Para la implementación se emplearon herramientas libres y de código abierto. Se utilizó Eclipse como Entorno de Desarrollo Integrado, Java como lenguaje de programación orientado a objetos del lado del servidor. Además, PostgreSQL como Sistema Gestor de Bases de Datos, Hibernate como herramienta ORM para la persistencia de los datos y el framework Seam para la lógica del negocio.

La aplicación permitirá eliminar el procesamiento manual de la información garantizándose con ello la calidad de los documentos generados. Por otra parte se agilizará la búsqueda de información registrada en consultas anteriores, facilitando los procesos de emisión de tratamiento, y gestión de solicitudes asociadas a los servicios de apoyo al diagnóstico.

PALABRAS CLAVE

Consulta externa

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 <i>Conceptos básicos asociados al dominio del problema.....</i>	<i>5</i>
1.2 <i>Sistemas automatizados existentes vinculados al campo de acción.....</i>	<i>6</i>
1.3 <i>Tecnologías y herramienta actuales a considerar</i>	<i>9</i>
1.4 <i>Lenguaje de programación</i>	<i>14</i>
1.5 <i>Metodologías de desarrollo.....</i>	<i>14</i>
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....	17
2.1 <i>Flujo actual de los procesos involucrados en el campo de acción</i>	<i>17</i>
2.2 <i>Objeto de automatización</i>	<i>19</i>
2.3 <i>Modelo de Negocio.....</i>	<i>20</i>
2.4 <i>Especificación de los requerimientos de software</i>	<i>26</i>
2.5 <i>Modelo de casos de uso del sistema</i>	<i>36</i>
2.6 <i>Descripción textual de los casos de uso.....</i>	<i>41</i>
CAPÍTULO 3. DISEÑO DEL SISTEMA	45
3.1 <i>Descripción de la arquitectura.....</i>	<i>45</i>
3.2 <i>Modelo de diseño</i>	<i>46</i>
3.3 <i>Descripción de las clases del diseño.....</i>	<i>59</i>
CAPÍTULO 4. IMPLEMENTACIÓN	64
4.1 <i>Modelo de datos.....</i>	<i>64</i>
4.2 <i>Descripción de las tablas.....</i>	<i>66</i>
4.3 <i>Modelo de implementación</i>	<i>77</i>
4.4 <i>Tratamiento de excepciones.....</i>	<i>80</i>
4.5 <i>Seguridad.....</i>	<i>80</i>
4.6 <i>Estrategias de Codificación. Estándares y estilos a utilizar.....</i>	<i>81</i>
CONCLUSIONES.....	86
RECOMENDACIONES.....	87
REFERENCIAS BIBLIOGRÁFICAS	88
BIBLIOGRAFÍA.....	Error! Bookmark not defined.
ANEXOS	93
GLOSARIO DE TÉRMINOS	94

ÍNDICE DE FIGURAS

Figura 2.1	Diagrama global de procesos del negocio	21
Figura 2.2	Diagrama de procesos del negocio: Atender paciente	22
Figura 2.3	Diagrama de procesos de negocio: Generar información estadística	23
Figura 2.4	Diagrama de procesos de negocio: Generar información para la Sección de estadística.....	24
Figura 2.5	Actores del sistema	39
Figura 2.1	Diagrama de casos de uso: Atender paciente.....	40
Figura 2.2	Diagrama de casos de uso: Generar estadísticas.....	41
Figura 3.1	Diagrama de paquetes.....	48
Figura 3.2	Diagrama de clases del diseño: CU_Consultar relación de pacientes.....	49
Figura 3.3	Diagrama de secuencia: CU_Consultar relación de pacientes.....	49
Figura 3.4	Diagrama de clases del diseño: CU_Crear Hoja general de consulta	50
Figura 3.5	Diagrama de secuencia: CU_Crear Hoja general de consulta	51
Figura 3.6	Diagrama de clases del diseño: CU_Buscar hojas de consulta	52
Figura 3.7	Diagrama de secuencia: CU_Buscar hojas de consulta	52
Figura 3.8	Diagrama de clases del diseño: CU_Ver datos de Hoja general de consulta	53
Figura 3.9	Diagrama de secuencia: CU_Ver datos de Hoja general de consulta	54
Figura 3.10	Diagrama de clases del diseño: CU_Crear solicitud de consulta.....	55
Figura 3.11	Diagrama de secuencia: CU_Crear solicitud de consulta.....	56
Figura 3.12	Diagrama de clases del diseño: CU_Consultar solicitud de consulta.....	57
Figura 3.13	Diagrama de secuencia: CU_Consultar solicitud de consulta.....	58
Figura 3.14	Diagrama de clases del diseño: CU_Crear indicaciones médicas	58
Figura 3.15	Diagrama de clases del diseño: CU_Crear indicaciones médicas	59
Figura 4.1	Modelo de datos	65
Figura 4.2	Diagrama de despliegue.....	78
Figura 4.3	Diagrama de componentes.....	79

INDICE DE TABLAS

Tabla 2.1	Actores involucrados	26
Tabla 2.2	Actores del sistema.....	38
Tabla 2.3	CU_ Consultar relación de pacientes	41
Tabla 2.4	Crear Hoja general de consulta.....	42
Tabla 2.5	Buscar hojas de consulta.....	42
Tabla 2.6	Ver datos de Hoja general de consulta.....	42
Tabla 2.7	Crear solicitud de interconsulta.....	43
Tabla 2.8	Consultar solicitud de interconsulta	43
Tabla 2.9	Crear indicaciones médicas.....	44
Tabla 3.1	Clase controladora: CrearIndicacionesMedicasControlador	59
Tabla 3.2	Clase controladora: ConsultarRelacionPacientesControlador_consulta	60
Tabla 3.3	Clase controladora: CrearHojaGeneralControlador_consulta.....	60
Tabla 3.4	Clase controladora: VerHojaGeneralControlador_consulta	61
Tabla 3.5	Clasecontroladora: ConsultarSolicitudInterconsultaControlador_consulta	61
Tabla 3.6	Clase controladora: CrearSolicitudInterconsultaControlador_consulta.....	62
Tabla 3.7	Clase interfaz: CrearIndicacionesMedicas	62
Tabla 3.8	Clase interfaz: ConsultarRelacionPacientes	62
Tabla 3.9	Clase interfaz: CrearHojaGeneral	62
Tabla 3.10	Clase interfaz: Ver HojaGeneral	63
Tabla 3.11	Clase interfaz: CrearSolicitudInterconsulta.....	63
Tabla 3.12	Clase interfaz: ConsultarSolicitudInterconsulta.....	63
Tabla 4.1	Hoja_consulta.....	68
Tabla 4.2	Cita	69
Tabla 4.3	General.....	69
Tabla 4.4	General_exam_fis.....	71
Tabla 4.5	General_exam_fun.....	73
Tabla 4.6	Solicitud_interconsulta	73
Tabla 4.7	Ind_medica_medicamento	74
Tabla 4.8	Indicacion_medica.....	75
Tabla 4.9	Unidad_tiempo.....	76
Tabla 4.10	Unidad_pres_medida.....	76
Tabla 4.11	Via_administracion	77

INTRODUCCIÓN

En estos últimos tiempos, se ha hecho necesario el uso de las Tecnologías de la Información y las Comunicaciones (TIC) para la obtención de conocimiento y la creación de herramientas que permitan gestionar de manera eficiente la información en cualquier proceso de la sociedad. La automatización de estos procesos viene dada por el acelerado incremento de los volúmenes de datos generados en diferentes sectores y la obtención de reportes estadísticos con un mayor grado de complejidad.

La rama de la salud es uno de los sectores que ha proyectado incorporar en su funcionamiento soluciones informáticas. Pues tiene como reto lograr la calidad y el control de la información generada en los diferentes niveles de atención. Con el objetivo de agilizar las tareas médicas, disminuir la espera del paciente y elevar la calidad en los servicios prestados

Estas soluciones han evolucionado, obteniéndose sistemas para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier centro asistencial, ejemplo los sistemas de información hospitalaria (SIH o HIS por sus siglas en inglés). Los que además de lograr la optimización de los recursos humanos y materiales, minimizan los inconvenientes burocráticos que puede enfrentar el paciente. También generan reportes e informes, y permiten la retroalimentación elevando la calidad de la atención médica en los servicios o áreas de las instituciones hospitalarias.

Los HIS en su gran mayoría se han convertido en productos de software que responden a las necesidades específicas de las distintas áreas de una institución hospitalaria, las cuales mantienen una estrecha relación entre sí. Una de las áreas donde se presta atención médica es la de consulta externa, en esta se agrupan un conjunto de servicios médicos destinados a ofrecer atención especializada a pacientes cuya patología no sea urgente.

El proceso de mantener la información actualizada y organizada en esta área se hace cada día más difícil, debido al incremento de pacientes atendidos y a la interrelación que tiene con las áreas destinadas a apoyar el diagnóstico final. Actualmente la mayoría de las instituciones hospitalarias no cuentan con una aplicación informática que se encargue de automatizar algunos de los procesos que se llevan a cabo en el

área de consulta externa, por lo que no se garantiza el almacenamiento de toda la información que se genera durante la atención del paciente, debido a que se realiza de forma manual.

Una de las consecuencias que trae la ausencia de un sistema que automatice estos procesos, se evidencia en la Historia Clínica (HC) -documento donde se registran cada uno de los episodios médicos en los que interviene el paciente- la cual se encuentra en formato duro. El deterioro de la misma -por las condiciones de los locales donde se almacena y la manipulación del personal autorizado a la hora de consultarlas-, puede afectar la posible toma de decisiones.

Por otra parte, el resultado de una consulta médica se registra en formatos de papel que se anexan a la HC. Estos en la mayoría de los casos tienen varios años de creado y debido a su constante reproducción se pueden tornar ilegibles, afectando el llenado de la información por parte del personal médico. Las indicaciones médicas y resultados de estudios complementarios sólo son válidos en formato duro, lo que trae consigo que la pérdida o destrucción de los mismos implique rehacer las solicitudes o los exámenes. Así mismo, la búsqueda en la HC de información generada en diferentes consultas, por parte de los médicos, se torna tediosa y difícil debido que esta se incrementa gradualmente a medida que el paciente es atendido en diferentes especialidades.

Otro de los problemas que enfrenta el área de consulta externa está condicionado por la falta de herramientas para la obtención de estadísticas diarias y mensuales. La información que se genera en cada servicio clínico crece vertiginosamente. Lo que provoca que el proceso de generación de reportes estadísticos sea cada vez más engorroso y se necesite mayor tiempo para obtener un resultado, que en la mayoría de los casos no es el más acercado, ni garantiza un nivel de desglose y especificidad adecuado.

Las situaciones descritas anteriormente han dado pie a la creación de sistemas que agilicen y mejoren los principales procesos que se llevan a cabo en el área de consulta externa, donde su principal misión será proporcionar atención médica de óptima calidad a los pacientes que demanden de los servicios de consulta externa y que puedan ser atendidos con eficiencia de manera oportuna.

Considerando lo analizado anteriormente, se define el siguiente **problema científico** a resolver: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Consulta Externa de las instituciones hospitalarias?

Este problema se enmarca en el **objeto de estudio** concerniente a los procesos de gestión de la información en las instituciones hospitalarias y **campo de acción** que comprende los procesos de gestión de información en el área de Consulta Externa de las instituciones hospitalarias.

Basado en esa idea se define el **objetivo general** del presente trabajo:

Desarrollar el módulo de Consulta Externa del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **tareas de la investigación**:

1. Evaluar las tendencias actuales en el mundo de los Sistemas de información Hospitalaria.
2. Identificar los principales procesos asociados al área de Consulta Externa de las instituciones hospitalarias.
3. Definir las funcionalidades del módulo de Consulta Externa del Sistema de Información Hospitalaria.
4. Asimilar la arquitectura definida por el Área Temática Gestión Hospitalaria para el desarrollo de sus aplicaciones.
5. Obtener mediante el Proceso Unificado de Desarrollo, los flujos de trabajo de “Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
6. Implementar los procesos relacionados con la atención al paciente y la generación de estadísticas del módulo de Consulta Externa.

En este sentido se puede destacar que el desarrollo del módulo Consulta Externa del Sistema de Información Hospitalaria alas HIS, proporcionará un grupo de beneficios entre los que pueden ser mencionados los siguientes:

1. Mejorar la organización de los procesos que se llevan a cabo en el área de Consulta Externa.

2. Disponer y acceder a información única e integrada del paciente para facilitar los procesos de diagnóstico, tratamiento y seguimiento.
3. Garantizar la calidad en los informes médicos que se les entregan a los pacientes.
4. Permitir la revisión de diagnóstico de pacientes atendidos con anterioridad.
5. Obtener datos estadísticos que se generan en esta área.

El presente documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **FUNDAMENTACIÓN TEÓRICA**, ubica al lector en el ambiente de desarrollo del módulo de Consulta Externa, justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo del mismo. Seguidamente el capítulo, **CARACTERÍSTICAS DEL SISTEMA**, contiene los diferentes procesos de negocio asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **DISEÑO DEL SISTEMA** se centra en la modelación detallada y la construcción de la estructura de la aplicación. En el cuarto y último, **IMPLEMENTACIÓN**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área en cuestión.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan conceptos básicos que posibilitan un mejor entendimiento de los procesos vinculados al campo de acción. También son analizados los sistemas de información hospitalaria existentes y se fundamenta la selección de tecnologías, metodologías y un grupo de herramientas utilizadas en el desarrollo de la investigación.

1.1 Conceptos básicos asociados al dominio del problema

Prestarle atención especializada a cualquier paciente que lo necesite es la principal función que tienen los médicos que laboran en los servicios comprendidos en el área de *Consulta Externa*. Es aquí donde se valora, diagnostica y de ser necesario, se remite a un paciente para otras áreas de cualquier centro asistencial.

Los pacientes pueden ser atendidos en una *consulta de triaje especializado, triaje general, primera, control o interconsulta* y en la mayoría de los casos deben asistir con una cita previa, excepto en la consulta de triaje general, donde son vistos y diagnosticados por un médico especialista el mismo día que acuden al hospital.

En la *consulta de triaje especializado* el especialista a partir de un interrogatorio y examen físico general, hace una valoración de la situación presentada por el paciente y decide si el mismo amerita ser atendido en servicio. A partir del resultado de esta decisión es que comienza el trámite para la creación de la HC.

La *Historia Clínica* es el conjunto de documentos que se genera tanto a pacientes ambulatorios como a hospitalizados. Su principal función es asistencial, ya que permite reflejar el seguimiento de los pacientes durante todo su paso por el hospital. Otras funciones son: la docencia, al permitir la realización de estudios de investigación y epidemiología, la evaluación de la calidad asistencial, la planificación y gestión sanitaria y su utilización en casos legales en aquellas situaciones jurídicas en que se requiera.

Un paciente asiste a una *consulta de primera* cuando comienza a ser tratado por determinada patología, aquí al igual que en el resto de las consultas el médico especialista puede orientar un conjunto de exámenes que lo ayuden a contribuir con la emisión de un diagnóstico definitivo. En la *consulta de control*, se atienden pacientes por una patología anteriormente diagnosticada en el propio servicio, es de vital

importancia, pues es donde se da un seguimiento a la evolución del paciente según el tratamiento indicado por el especialista.

Durante el transcurso de una consulta se puede emitir por el médico tratante una solicitud para *interconsulta*, este es el procedimiento que permite la participación de otro profesional de la salud a fin de proporcionar atención integral al paciente.

1.2 Sistemas automatizados existentes vinculados al campo de acción

A nivel mundial existen un gran número de empresas especializadas en el desarrollo e implantación de soluciones informáticas vinculadas al sector de la salud, con el objetivo de optimizar recursos y mejorar la calidad de los servicios prestados. Muchas de estas soluciones van encaminadas a gestionar información que se genera en departamentos y áreas de las instituciones hospitalarias y específicamente en el área de consulta externa.

Sistema de Información para la Gerencia Hospitalaria (SIGHO)

Software basado en la Norma Oficial Mexicana (NOM-168-SSA1-1998) referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital. Está integrado por 14 módulos entre los que incluye el de consulta externa que permite al médico tener acceso de manera sencilla al historial del paciente; incluye los distintos diagnósticos que le han emitido, el tratamiento recibido, así como evoluciones y estudios auxiliares realizados.

Entre las principales funcionalidades del módulo se encuentran el registro de la historia clínica y datos derivados de la atención médica, como son solicitudes de estudios a servicios auxiliares de diagnóstico, solicitudes de interconsulta, registro de alergias a medicamentos, prescripción de medicamentos y emisión de recetas.

Se apoya en estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9.

Entre las salidas de información que presenta el módulo se encuentran:

1. Reporte para el Sistema Único de Información para la Vigilancia Epidemiológica (SUIVE) que incluye información semanal de nuevos casos de enfermedades.
2. Reporte de productividad por médico.
3. Impresión de la historia clínica del paciente.

Para su desarrollo se utilizó Microsoft Asp.net y Framework Microsoft.NET y SQL Server 2000 como gestor de base de datos. [1]

HIS CNT PACIENTES

Sistema de información integrada, totalmente modular, escalable, flexible y de fácil puesta en marcha. Está diseñado para integrar el ciclo de atención del paciente frente a la prestación de servicios médicos, terapéuticos y diagnósticos. Tiene como eje central para efectuar sus consultas, una HC única, dinámica, digital, integrada y adaptable a todas las especialidades médicas y ambientes de atención en la consulta externa.

Es una aplicación de escritorio que además brinda la funcionalidad de definir el horario de los médicos para la asignación de citas, logrando la optimización del tiempo, y ofreciendo a sus pacientes una mejor oportunidad del servicio. [2]

@Clinic

Software compuesto por varios módulos dentro de los que se encuentran: Agenda, Admisión, Pruebas, Facturación, Imagen, Farmacia y Consulta; este último donde recae el mayor peso.

El módulo de Consulta consta de un menú Archivo: donde se encuentran los datos propios de la consulta médica y desde el cual se puede acceder a diferentes secciones. Dentro de las secciones están: PREVIO: se especifican las impresiones iniciales captadas por el médico, EXPLORACIÓN: se redactan las conclusiones obtenidas tras la exploración y se almacenan radiografías o pruebas gráficas, DIAGNOSIS: se plasma el resultado final del proceso realizado en la consulta médica y REVISIONES: se registran las evoluciones del paciente y se anotarán las posibles incidencias y cambios de medicación, permitiendo además imprimir informes personalizados a partir de la utilización de la información almacenada.

Su desarrollo se llevó a cabo utilizando software propietario, es una aplicación de escritorio que puede utilizar cualquier tipo de gestor de base de datos, y es compatible MacOs, Linux y versiones superiores a Windows 95. [3]

x-HIS

Software desarrollado por la empresa española ISOFT, elimina la navegación por menús e incorpora una interfaz que facilita el uso de la aplicación a cualquier tipo de usuario. Está compuesto por varios módulos en los cuales se gestiona la información de las diferentes áreas del hospital. Uno de estos es el asociado a la consulta externa, que permite la gestión de las agendas de los profesionales y las especialidades asociadas, en un entorno mono o multihospitalario.

Este módulo abarca tres áreas claramente diferenciadas, que se corresponden con el ciclo lógico de trabajo asociados a las tareas: gestión de agendas, citación y confirmación de citas. Permite crear un archivo de historias clínicas de forma digital para gestionar de manera informatizada el historial de los pacientes, manteniendo los informes clínicos ordenados y clasificados en expedientes clínicos electrónicos.

X-HIS fue desarrollado bajo los estándares HL7, NOC, NANDA, es multiplataforma, con independencia de base de datos: MS SQL Server, ORACLE, SYBASE. [4]

CONSULTA MÉDICA

Consulta Médica v1.21 es un software que gestiona la información generada en la consulta médica. Permite crear fichas de pacientes atendidos (todos sus datos personales) y un historial clínico de los mismos. Brinda la opción de programar citas médicas para un día determinado y permite imprimir los documentos generados.

Está preparado para funcionar en una red de área local, de manera que si varios médicos de un mismo centro quieren compartir la misma base de datos de pacientes pueden hacerlo. Incluye además la gestión de pacientes centralizada. Tiene como limitación que permite atender un máximo de 10 pacientes en consulta simultáneamente. Su desarrollo se basa en software propietario pues utiliza como lenguaje de programación ASP.NET y sus datos son organizados con el gestor de bases de datos MySQL. [5]

1.3 Tecnologías y herramienta actuales a considerar

Eclipse

Eclipse es un entorno de desarrollo integrado (IDE siglas en inglés) de código abierto, portable y multiplataforma. Brinda una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins.

La arquitectura de plug-ins permite integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones. Hay plugins para el desarrollo de Java (JDT Java Development Tools) así como para el desarrollo en C/C++, COBOL, estos conservan el registro de las versiones, generan y mantienen la documentación de cada etapa del proyecto.

Dentro de las características que presenta este lenguaje se destacan: poseer un editor visual con sintaxis coloreada, permitir la compilación incremental de código, modificar e inspeccionar valores de variables, avisar de los errores cometidos mediante una ventana secundaria y depurar el código que reside en una máquina remota. Es soportado por los sistemas operativos:

- Linux
- Windows
- Solaris 8 (SPARC/GTK 2)
- Mac OSX –Mac/Carbon. [6]

JBoss Application Server

JBoss Server es un servidor de aplicaciones J2EE de código abierto implementado en Java. Puede ser utilizado en cualquier sistema operativo que soporte el lenguaje Java. Por ser una plataforma certificada, J2EE, soporta todas las funcionalidades de J2EE-1.4, incluyendo servicios adicionales como: clustering, caching y persistencia. Es ideal para aplicaciones Java y aplicaciones basadas en la web. Soporta Enterprise Java Beans (EJB) 3.0, es incrustable y orientado a arquitectura de servicios, además, posee servicios del middleware para cualquier objeto de Java. [7]

JBoss Seam

JBoss Seam es un framework OpenSource desarrollado por la empresa JBoss con el fin de unir diferentes tecnologías y estándares de Java, JSF, EJB3.0, JPA, además de BPM (Business Process Management) en un solo framework, a la vez que añade algunas funcionalidades no contempladas por ellos. Elimina la capa artificial que existe entre EJB3.0 y JSF y provee un consistente sistema de anotaciones para integrar estos dos frameworks. En Jboss Seam los datos de los formularios se almacenan en entity beans y los eventos se manejan con session beans.

Permite a los desarrolladores usar anotaciones POJOs (Plain Object Java) para todos los componentes de la aplicación. Los componentes Seam pueden ser definidos completamente mediante el uso de anotaciones, incluidas las biyecciones, validadores, observers. Añade además, 3 nuevos contextos a los tradicionales (request, session y application) que son las pages, conversation y busines.

Comparada con aplicaciones desarrolladas en otros framework, las aplicaciones Seam son conceptualmente simples y requieren significativamente menos código (en Java y en XML) para obtener las mismas funcionalidades. [8]

Una de las principales ventajas del uso de Seam es permitir el control de sus componentes mediante anotaciones, lo cual reduce la cantidad de archivos XML de configuración.

JBoss Tools

Jboss tools es una colección de plugins que se le añaden al IDE, los cuales le incorporan una serie de funcionalidades, además de poder ser añadidos a diferentes servidores de aplicación. Posee un editor gráfico para la configuración de archivos Seam y soporta la realización de pruebas de integración de Seam desde el Eclipse. [9]

Dentro de los plugins están:

- Seam tools: Incluye soporte para la integración de los componentes del *framework* Seam.
- RichFaces VE: Editor visual para componentes HTML, JSF y RichFaces

- Hibernate tools: Sirve de apoyo para la utilización de los componentes del *framework* Hibernate y para el mapeo con la base de datos.

JSF

Java Server Faces (JSF) es un framework Java que permite crear interfaces de usuario (UI) para aplicaciones web, mediante componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. [10]

RichFaces

RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Con Richfaces se puede integrar fácilmente las capacidades JavaScript asíncrono y XML (AJAX), utilizando para ello el framework Ajax4jsf, dentro del desarrollo de aplicaciones de negocio de nivel empresarial. Provee facilidades de validación y conversión de los datos proporcionados por el usuario y administración avanzada de recursos como imágenes, código Java Script y hojas de estilo en cascada (CSS). Se integra completamente dentro del ciclo de vida JSF. Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora la experiencia de usuario. [11][12][13]

FACELETS

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. [14]

Ajax4JSF

Ajax4jsf es una librería Open Source que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Con este se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo y realizar peticiones automáticas al servidor. Ajax4jsf permite dotar a cada aplicación JSF de contenido mucho más profesional con muy poco esfuerzo. [15]

PostgreSQL

PostgreSQL es un sistema de base de datos relacional de código abierto, que se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Este cuenta con diversas versiones para sistemas operativos tales como: Linux, Windows, Unix, Mac OS X, Solaris, BSD, Tru64 y otros.

Soporta vistas, uniones, claves extranjeras, triggers, incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99) y presenta soporte de protocolo de comunicación encriptado por SSL, extensiones para alta disponibilidad, nuevos tipos de índices y minería de datos.

Permite crear, editar, copiar, extraer y bajar todo objeto de las bases de datos tales como esquemas, tablas, vistas, funciones, dominios, reglas, secuencias, idiomas, operadores, etc., construye consultas visualmente, ejecuta consultas y scripts SQL, visualiza y edita datos, representa datos como diagramas, exporta e importa datos desde y hacia los formatos de archivos de uso más popular. Además administra roles, usuarios, grupos y sus privilegios, y usa una serie de herramientas adicionales diseñadas para una fácil y eficiente operación con el Servidor PostgreSQL.

Dentro de las características que se destacan de PostgreSQL y que lo convierten en una herramienta eficiente para el trabajo en las base de datos están la atomicidad, la consistencia, el aislamiento y la durabilidad, juntas aseguran que solo empieza aquello que se puede acabar, garantizan que una operación no puede afectar a otras y una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. [16]

Hibernate

Hibernate es un framework que provee herramientas de mapeo objeto/relacional y permite reducir significativamente el tiempo de desarrollo. Con su API nativa es el servicio base para la persistencia de datos. Posee un lenguaje de consultas llamado HQL (parecido al lenguaje de consultas SQL). Sus herramientas soportan distintos tipos de base de datos, lo que confiere cierto nivel de portabilidad a las aplicaciones que lo utilizan. A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos. [17]

EJB

EJB son componentes del contexto de servidor que cubren la necesidad de intermediar entre la capa web y diversos sistemas empresariales. Los cuales nacen para encapsular la lógica de negocio de una forma integrada. De esta manera no queda dispersa su representación en un grupo de sistemas empresariales. Están especialmente pensados para integrar la lógica de la empresa que se encuentra en sistemas distribuidos, de tal forma que el desarrollador no tenga que preocuparse por la programación a nivel de sistema, sino que se centre en la representación de entidades y reglas de negocio. [18]

JPA

Java Persistence API (JPA) proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes.

El mapeo objeto/relacional, es decir, la relación entre entidades Java y tablas de la base de datos, se realiza mediante anotaciones en las propias clases de entidad, por lo que no se requieren archivos descriptores XML. También pueden definirse transacciones como anotaciones JPA. [19]

Visual Paradigm

Visual Paradigm es una herramienta profesional CASE (Computer-Aided Software Engineering) que utiliza el Lenguaje de Modelado Unificado (UML, como sus siglas en inglés lo indican). Soporta el ciclo de vida

completo del desarrollo de software: modelado de negocio, análisis y diseño orientados a objetos, implementación, pruebas y despliegue.

Proporciona tutoriales de UML, soporta UML versión 2.1, así como demostraciones interactivas de dicho lenguaje. Se integra a la plataforma Java, la cual funciona en sistemas operativos Windows, Linux y Mac OS X y con varios IDEs, entre los que se encuentran Eclipse y NetBeans. Posibilita la generación de código desde diagramas y la generación de documentos en varios formatos como HTML, Microsoft Word y PDF. [20]

1.4 Lenguaje de programación

Java

Java es un lenguaje de programación desarrollado por la compañía Sun Microsystems, fue diseñado desde un principio orientado a objetos. Muchas de sus sintaxis las toma de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores. No es necesario preocuparse por liberar memoria ya que permite a través del reciclador liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la misma.

Una de las principales características por las que Java se ha hecho famoso es permitir la ejecución de un mismo programa en múltiples sistemas operativos y ordenadores. Es un lenguaje compilado, pues genera ficheros de clases compiladas, las cuales son interpretadas por la máquina virtual de java que mantiene el control sobre las que se estén ejecutando.

Permite programar aplicaciones web dinámicas, con acceso a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. Se utiliza de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. Soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. [21]

1.5 Metodologías de desarrollo

Proceso Unificado Racional (RUP)

RUP es un proceso de desarrollo de software y junto con el Lenguaje unificado de modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser una forma disciplinada de asignación de tareas y responsabilidades (quién hace qué, cuándo y cómo) y por pretender implementar las mejores prácticas en Ingeniería de Software a través de desarrollo iterativo, administración de requisitos, uso de arquitectura basada en componentes, control de cambios, modelado visual del software y verificación de calidad del mismo.

Es iterativo e incremental, está centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles [22].

Divide el proceso de desarrollo en ciclos, teniendo un producto al culminar de cada uno, estos se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo su alcance con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- **Construcción:** Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Lenguaje Unificado del Modelado (UML)

UML es un lenguaje usado para especificar, visualizar, construir y documentar los componentes de un sistema orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está apoyado en gran manera por el OMG (Object Management Group).

Además ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Permite modelar sistemas utilizando técnicas orientadas a objetos, es independiente aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. [23]

Conclusiones

En este capítulo se hizo un estudio de los sistemas existentes relacionados fundamentalmente con el área de consulta externa, llegando a la conclusión, de que la mayoría comparten la característica de estar desarrollados sobre software propietario con altos costos de licencia. Además son aplicaciones de escritorio, las cuales aumentan los gastos, por la necesidad de equipos de mayor potencia en relación al uso de memoria y espacio en disco. Por otra parte, no garantizan la gestión de la información relacionada con diferentes servicios de apoyo de una institución hospitalaria.

De igual forma fueron analizadas un conjunto de herramientas, metodologías y tecnologías que constituyen la propuesta tecnológica, para el desarrollo del Módulo Consulta Externa para el Sistema de Información Hospitalaria alas HIS.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se describen los principales procesos asociados al área de consulta externa y se representan gráficamente a través de diagramas de procesos para lograr una mejor comprensión de los mismos. Además se especifican los actores involucrados en cada uno y se definen los requerimientos funcionales y no funcionales, a partir de los cuales se llega a los casos de uso del sistema.

2.1 Flujo actual de los procesos involucrados en el campo de acción

Atender paciente

La atención al paciente es uno de los procesos más importantes que se llevan a cabo en los servicios del área de consulta externa. En ella se involucran el paciente, que necesita asistencia médica; el médico que se encarga de brindarla, y por último, no menos importante, la enfermera y el técnico de registros y estadísticas de salud que tienen la responsabilidad de entregar antes de la consulta, las historias clínicas de los pacientes citados y un documento donde se deben registrar los datos generales de los pacientes que asistieron.

Un paciente llega a la consulta especializada cuando es remitido por un médico del hospital u otro centro asistencial, si requiere seguimiento por una patología anteriormente diagnosticada en el servicio o presenta determinada dolencia que necesita ser tratada. En dependencia de estas situaciones puede ser visto en una consulta de primera, control, triaje o interconsulta.

Aunque hay distinción de consultas, el procedimiento a la hora de atender un paciente es parecido, siempre se interroga y examina para llegar a un diagnóstico y a la prescripción de un tratamiento. Con el interrogatorio se conoce el motivo de la consulta, los antecedentes personales y familiares así como los hábitos psicobiológicos que presenta el paciente y que pueden influir en el comportamiento de la patología presentada.

Los pacientes con una remisión o solicitud de interconsulta, enriquecen el interrogatorio, pues estos documentos justifican la necesidad de requerir atención especializada a partir de resultados de exámenes realizados, que sirven de punto de partida para llegar a un diagnóstico definitivo en un tiempo prudencial.

La realización del examen funcional y el físico son otras de las actividades que complementan la atención del paciente, se basan en la descripción de la historia actual de la enfermedad y en métodos prácticos como la inspección, palpación, percusión y auscultación que les sirven al médico para indagar sobre aspectos negativos y positivos, de interés a la hora de emitir un diagnóstico.

El paciente puede presentar resultados de exámenes indicados antes de la consulta, estos son evaluados por el médico, que decide si cumplen con un período de validez y son suficientes a la hora de la toma de decisiones.

El médico durante la consulta puede indicar análisis de laboratorio, estudios radiológicos e imagenológicos, solicitudes de biopsia, citologías que lo ayudarán a conocer o corroborar la causa de la enfermedad.

A partir del interrogatorio y los exámenes practicados el médico emite un diagnóstico presuntivo o definitivo y determina si el paciente necesita tratamiento. Los tratamientos pueden ser ambulatorios o de hospitalización, estos se materializan en las indicaciones médicas, que tienen los medicamentos que debe consumir el paciente, además de la vía, la dosis e indicaciones complementarias que debe seguir el paciente para su pronta recuperación. Por cada uno de los medicamentos indicados se crea un recípe, que debe ser presentado en la farmacia para adquirir los medicamentos. En el caso de los tratamientos de hospitalización se genera además una orden de admisión la cual debe ser presentada en el área de admisión del hospital para formalizar el ingreso.

Antes de culminar la consulta el médico decide si el paciente necesita una interconsulta, hospitalización, una nueva cita, reposo, entre otras. Cada opción cuenta con un modelo que se debe llenar y entregar al paciente, para garantizar la continuidad de la atención en el hospital o en otro centro asistencial.

En cada espacio de la atención al paciente se genera información valiosa que se registra en la HC, la solicitud de interconsulta u hoja de evolución, teniendo en cuenta el tipo de consulta. Esto se hace con el objetivo de mantener actualizado el historial de consultas para posteriores análisis médicos y estadísticos.

Generar información estadística

El proceso de Generar información estadística se lleva a cabo en todos los servicios del área de consulta externa, con el objetivo de tener el control de la cantidad de pacientes atendidos. La enfermera y el

médico comparten dentro de este proceso la responsabilidad de entregar la relación de pacientes una vez culminado el horario de consulta y el técnico de estadísticas de salud se encarga de procesar la información.

Diariamente el técnico de registros de estadísticas de salud apoyándose del libro de control de citas y de la relación de pacientes, confecciona las estadísticas que reflejan la cantidad de pacientes atendidos, desglosada por tipo de consulta. Mensualmente el propio técnico basado en la información recopilada diariamente, genera un resumen que se entrega al área de coordinación de consulta externa.

Generar información para la Sección de estadísticas

El proceso de Generar información para la Sección de Estadísticas se realiza en el área de coordinación de consulta externa. Los técnicos de registros y estadísticas de salud y la coordinadora del área son los principales involucrados en cada una de sus actividades. Mensualmente la coordinadora recibe la información recopilada por cada servicio, confecciona las estadísticas generales y las entrega a la sección de Estadísticas del hospital.

2.2 Objeto de automatización

En los procesos de negocio que se llevan en acabo en el área de consulta externa se identificaron un conjunto de actividades y registros asociados a la HC, lo cuales serán el principal objeto a automatizar durante el desarrollo de la investigación. La informatización de actividades y registros traerá consigo que todos los procesos que se realizan de forma manual y generan modelos en formato duro se eliminen, con el desarrollo de un HIS encargado de informatizar las actividades del área de consulta externa.

Para llevar a cabo el proceso de atender paciente el médico especialista tendrá que acceder al listado de pacientes citados para el día. Una vez seleccionado el paciente a consultar, el sistema mostrará la hoja de consulta correspondiente a la especialidad del médico y servicio donde se encuentre trabajando.

Cada hoja de consulta estará compuesta por datos generales y datos específicos, estos últimos dependen de la especialidad o servicio. Dentro de los datos generales se encuentra información sobre antecedentes tanto personales como familiares, hábitos psicobiológicos, inmunizaciones, entre otros, que podrán ser modificados en cualquier consulta, quedando siempre una traza de las acciones realizadas sobre cada entidad.

En caso de que el paciente se haya realizado exámenes previos a la consulta, el sistema permitirá realizar una búsqueda de los resultados asociados a estos, teniendo en cuenta determinados criterios, que agilizarán el proceso.

Durante la consulta al paciente se le pueden indicar diferentes exámenes complementarios con el objetivo de llegar a un diagnóstico definitivo, con el sistema se garantizará de forma rápida la creación de cada solicitud.

El sistema permitirá registrar en cada hoja de consulta el diagnóstico final o impresión diagnóstica a partir del codificador internacional de enfermedades (CIE) en su décima edición.

La asignación de citas para estudios radiológicos e imagenológicos, exámenes de laboratorio y consultas de seguimiento o control serán otras de las opciones que se podrán realizar a través de las hojas de consulta. De esta manera el paciente solo tendrá que presentarse una vez al lugar donde debe ser atendido.

La informatización de los registros que se generan en cada consulta, servirá de apoyo a médicos y técnicos de registros y estadísticas de salud, para llevar el control y recopilar la información, lo que permitirá la generación de estadísticas de forma rápida y confiable.

2.3 Modelo de Negocio

La creación de modelos que organicen y representen los detalles de situaciones reales vinculadas con sistemas a desarrollar, son necesarios para entender y analizar la complejidad del problema a resolver.

Un modelo previo al desarrollo de un sistema, es el Modelo de negocio, que tiene como propósitos comprender los problemas actuales de la organización e identificar mejoras potenciales de estos, además de asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

A continuación se representa el diagrama de procesos de negocio, que engloba los tres procesos asociados al objeto de estudio de la investigación:

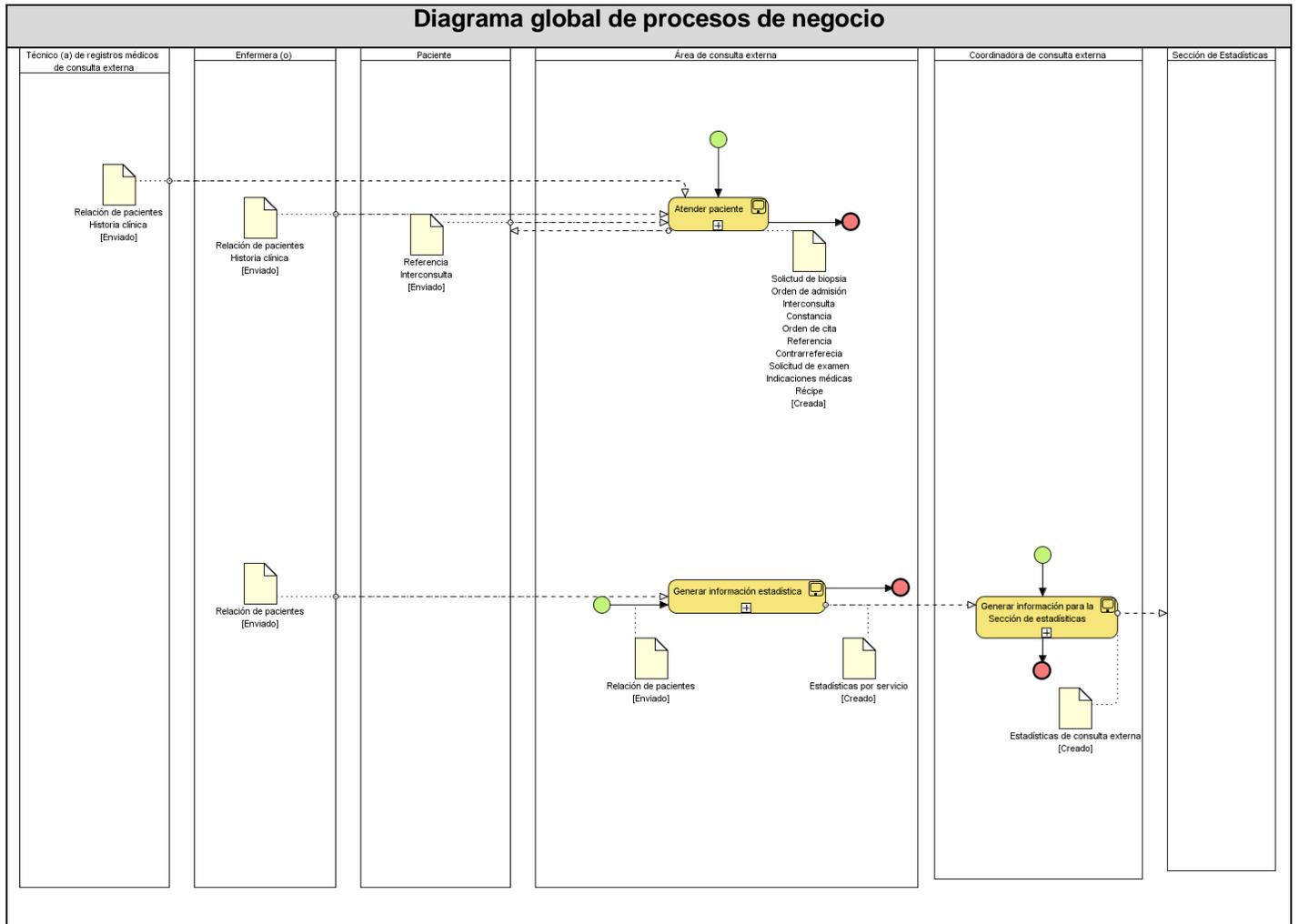


Figura 2.1 Diagrama global de procesos del negocio

Por cada proceso representado en el diagrama anterior se realiza un diagrama de procesos de negocio que facilita la comprensión de los mismos.

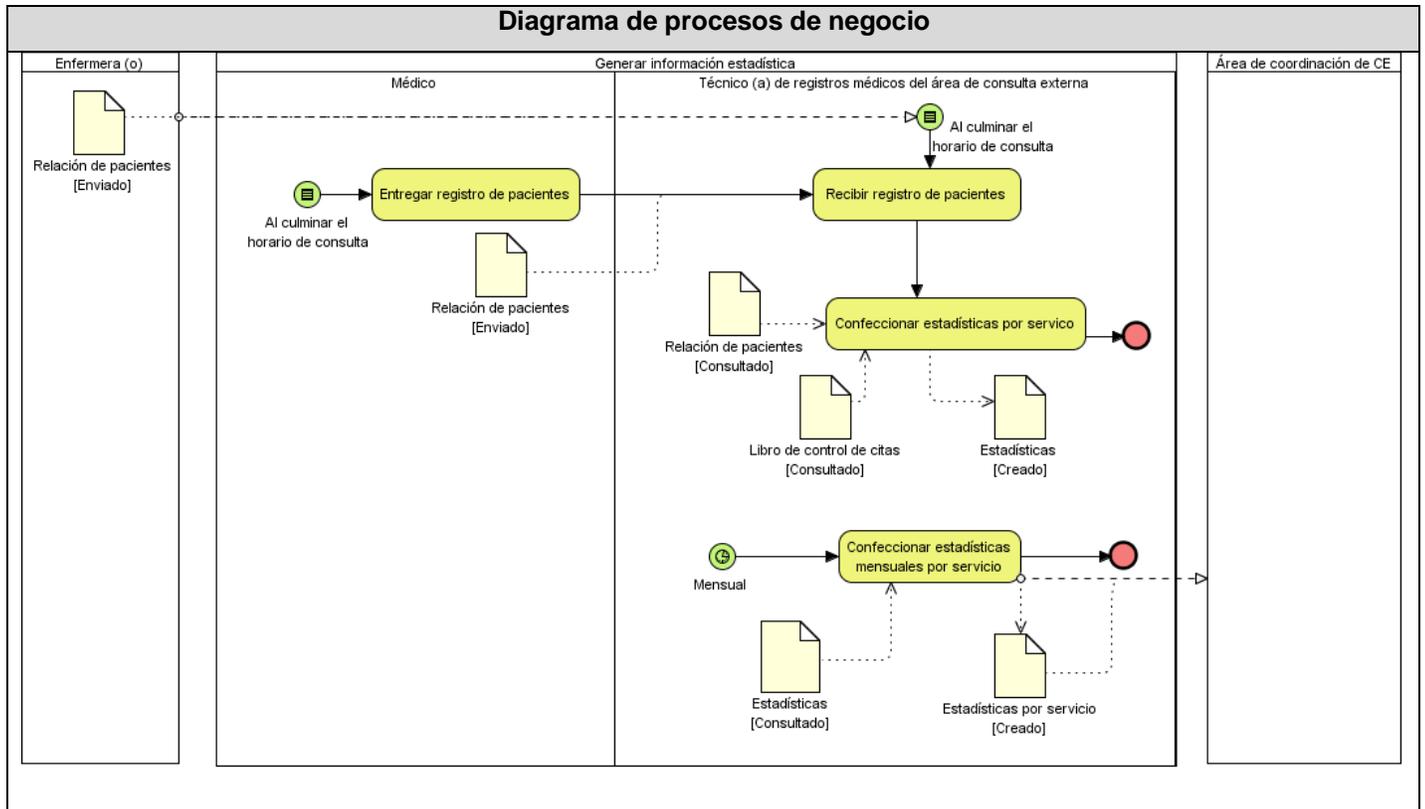


Figura 2.3 Diagrama de procesos de negocio: Generar información estadística

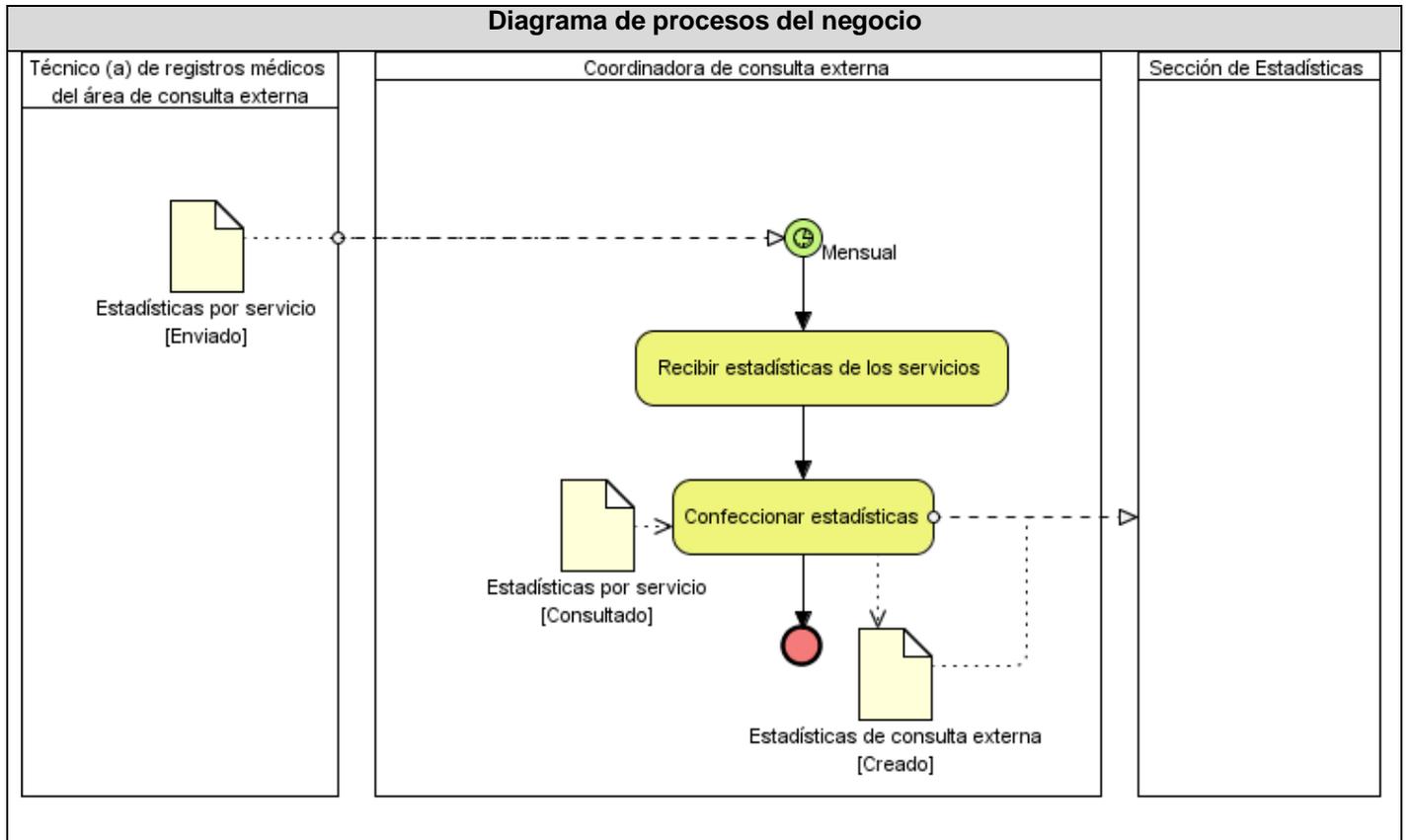


Figura 2.4 Diagrama de procesos de negocio: Generar información para la Sección de estadística

La ficha de procesos es otro de los artefactos que se incluye en el modelado de negocio. Se estructura por acápite, que garantizan la descripción textual de cada proceso identificado.

La misión del proceso, la identificación del especialista funcional encargado de validar la información recopilada y los actores involucrados - individuos, grupos de personas o departamentos organizacionales que desempeñan determinados roles dentro del proceso-, son datos que encabezan la ficha.

También contiene la especificación del flujo de información de cada actividad involucrada en el proceso e incluye los artefactos de entrada y salida, especificando formato y frecuencia de emisión o recepción.

Las reglas que constituyen restricciones en el proceso y procedimientos que sigue determinada organización son otros de los aspectos que se tienen en cuenta en cada ficha.

Actores involucrados en los procesos

Rol	Funciones
Paciente	Recibir atención médica
Médico	Interrogar, examinar, diagnosticar, indicar complementarios y tratamiento, a pacientes que solicitan asistencia médica. Entregar a el (la) técnico (a) de registros y estadísticas de salud la Relación de pacientes.
Enfermera	Entregar al médico las historias clínicas de los pacientes citados y la hoja de Relación de pacientes antes del horario de consulta. Entregar a el (la) técnico (a) de registros y estadísticas de salud la Relación de pacientes
Técnico (a) de registros y estadísticas de salud del área de consulta externa	Entregar al médico las historias clínicas de los pacientes citados y la hoja de Relación de pacientes antes del horario de consulta. Confeccionar diariamente las estadísticas generadas en el servicio. Entregar las estadísticas mensuales, generadas en cada servicio.

Área de coordinación de consulta externa	Recibir las estadísticas generadas mensualmente por cada servicio.
Coordinadora de consulta externa	Generar las estadísticas mensuales de consulta externa y entregarlas a la Sección de estadísticas del hospital.
Sección de estadísticas	Recibir las estadísticas mensuales del área consulta externa.

Tabla 2.1 Actores involucrados

A partir de las actividades representadas en los diagramas de procesos de negocio y apoyados en los flujos de información, descritos en las fichas, se definen las principales funcionalidades y características del sistema propuesto.

2.4 Especificación de los requerimientos de software

Requerimientos funcionales

Los requerimientos funcionales especifican una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas, definiendo un comportamiento de entrada y salida del mismo. Dentro de las acciones identificadas como requisitos funcionales para el desarrollo del sistema se pueden mencionar las que se listan a continuación:

1. Consultar relación de pacientes
2. Consultar relación de pacientes para triaje especializado
3. Consultar relación de pacientes para triaje general
4. Crear hoja general de consulta
5. Ver datos de hoja general de consulta
6. Crear hoja de traumatología y ortopedia
7. Ver datos de la hoja traumatología y ortopedia
8. Crear hoja neurológica

9. Ver datos de hoja neurológica
10. Crear hoja ginecológica
11. Ver datos de hoja ginecológica
12. Crear hoja oftalmológica
13. Ver datos de hoja oftalmológica
14. Crear hoja uro-genital
15. Ver datos de hoja uro-genital
16. Crear hoja ORL
17. Ver datos de hoja ORL
18. Crear hoja dermatológica
19. Ver datos de hoja dermatológica
20. Crear hoja obstétrica
21. Ver datos de hoja obstétrica
22. Crear hoja de gastroenterología
23. Ver datos de hoja de gastroenterología
24. Crear hoja de radioterapia y medicina nuclear
25. Ver datos de hoja de radioterapia y medicina nuclear
26. Crear hoja de cirugía colo-rectal
27. Ver datos de hoja de cirugía colo-rectal
28. Crear hoja hemato-oncológica
29. Ver datos de hoja hemato-oncológica

30. Crear hoja hemato-oncológica pediátrica
31. Ver datos de hoja hemato-oncológica pediátrica
32. Crear hoja para triaje general
33. Ver datos de hoja para triaje general
34. Crear cita de primera en la consulta
35. Modificar cita de primera
36. Eliminar cita de primera
37. Crear cita de control en la consulta
38. Modificar cita para control
39. Eliminar cita para control
40. Ver datos de cita para consulta
41. Crear cita para laboratorio desde la consulta
42. Crear cita para estudio radiológico e imagenológico
43. Modificar datos de cita para estudio radiológico e imagenológico
44. Eliminar cita para estudio radiológico e imagenológico
45. Ver datos de cita para estudio radiológico e imagenológico
46. Buscar cupo disponible para cita
47. Crear tarjeta de citas
48. Crear solicitud de análisis de laboratorio para usuarios internos
49. Modificar solicitud de análisis de laboratorio

- 50. Eliminar solicitud de análisis de laboratorio
- 51. Ver datos de solicitud de análisis de laboratorio
- 52. Ver datos de informe de resultados de laboratorio
- 53. Emitir constancia
- 54. Modificar datos de una constancia
- 55. Eliminar constancia
- 56. Ver datos de una constancia
- 57. Crear orden de admisión
- 58. Modificar orden de admisión
- 59. Eliminar orden de admisión
- 60. Ver orden de admisión
- 61. Consultar solicitud de interconsulta
- 62. Crear referencia
- 63. Modificar datos de una referencia
- 64. Eliminar referencia
- 65. Ver datos de una referencia
- 66. Crear contrarreferencia
- 67. Modificar datos de una contrarreferencia
- 68. Eliminar contrarreferencia
- 69. Ver datos de contrarreferencia

- 70. Consultar referencia
- 71. Crear solicitud de transfusión
- 72. Modificar datos de solicitud de transfusión
- 73. Eliminar datos de solicitud de transfusión
- 74. Ver datos de solicitud de transfusión
- 75. Crear solicitud de examen de tipiaje
- 76. Eliminar datos de solicitud de examen de tipiaje
- 77. Ver datos de solicitud de examen de tipiaje
- 78. Crear solicitud de intervención quirúrgica
- 79. Modificar solicitud de intervención quirúrgica
- 80. Eliminar solicitud de intervención quirúrgica
- 81. Ver datos de la solicitud de intervención quirúrgica
- 82. Consultar resultado de estudios radiológicos e imagenológicos
- 83. Crear indicaciones médicas
- 84. Modificar datos de indicaciones médicas
- 85. Eliminar indicaciones médicas
- 86. Ver datos de indicación médica
- 87. Seleccionar medicamentos
- 88. Buscar Indicaciones médicas
- 89. Crear solicitud de interconsulta

90. Modificar datos de solicitud de interconsulta
91. Eliminar solicitud de interconsulta
92. Ver datos de solicitud de interconsulta
93. Crear solicitud de biopsia
94. Modificar solicitud de biopsia
95. Eliminar solicitud de citología
96. Ver datos de solicitud de biopsia
97. Crear solicitud de citología
98. Modificar solicitud de citología
99. Eliminar solicitud de citología
100. Ver datos de solicitud de citología
101. Crear solicitud de citología ginecológica
102. Modificar solicitud de citología ginecológica
103. Eliminar solicitud de citología ginecológica
104. Ver datos de solicitud de citología ginecológica
105. Modificar antecedentes personales
106. Modificar antecedentes familiares
107. Modificar hábitos psicobiológicos
108. Buscar informe final de anatomía patológica
109. Ver datos de informe final de biopsia

- 110. Ver datos de informe final de citología
- 111. Ver datos de informe final de microscopía electrónica
- 112. Ver datos de informe final de inmunohistoquímica
- 113. Actualizar inmunizaciones
- 114. Buscar hoja de consulta
- 115. Buscar resultado de interconsulta
- 116. Consultar acciones realizadas
- 117. Consultar acciones realizadas hasta el momento
- 118. Seleccionar enfermedad
- 119. Notificar enfermedades con formatos para FIE
- 120. Generar estadísticas del servicio
- 121. Generar estadísticas de consulta externa

Requerimientos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema, como restricciones del entorno o implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad; de manera que definen las restricciones físicas sobre los requisitos funcionales.

A continuación se desglosan los requisitos no funcionales definidos para el producto, dentro de los cuales se especifican los de usabilidad, restricciones de diseño e implementación, entre otros.

Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- Usuarios normales: 20 días
- Usuarios avanzados: 30 días

Seguridad

- Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- El sistema proporcionará un registro de actividades (log) de cada usuario en el sistema.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvallas realizadas.

Rendimiento

- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
- El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

Soporte

- Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos.

- Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.
- Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Hardware

- **Estaciones de trabajo**

- En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda Internet Explorer 7 o superior o Firefox 2.x o superior).
- Por lo que se escogieron estaciones de trabajo de 256Mb de memoria RAM y un microprocesador de 2.0Hz con Sistema operativo Linux.

- **Servidores**

- La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables.
- Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

Software

- El servidor debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java virtual machine, JBoss AS y PostgreSql).
- Los clientes deberán disponer de un navegador web, estos pueden ser IE 7 o superior, Opera 9 superior, Google chrome 1 o superior y Firefox 2 o superior.

Restricciones de diseño

- La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio.
- La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.
- La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación de éste.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

- Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

Interfaz

• Interfaces de usuario

- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.
- La interfaz contará con accesos directos y menús desplegados que faciliten y aceleren su utilización.
- La entrada de datos incorrecta será detectada claramente e informada al usuario.

- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- **Interfaces de comunicación**
 - Para el intercambio electrónico de datos entre aplicaciones se usará el estándar HL7 (Health Level Seven).
 - El sistema usará el formato estándar WSDL para la descripción de los servicios web.
 - El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.
 - El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

Portabilidad

- El producto podrá ser utilizado bajo los sistemas operativos Linux o Windows.

Una vez definidas las funcionales del sistema y propiedades del producto a desarrollar, se define el Modelo de Casos de Uso del Sistema, compuesto por diagramas de casos de uso así como las especificaciones de cada caso de uso representado.

2.5 Modelo de casos de uso del sistema

Actores del sistema

Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él.

Actor	Descripción
Médico de consulta externa	Se encarga de agrupar el comportamiento común de los actores: gestor de la Hoja de traumatología y ortopedia, gestor de la Hoja neurológica, gestor de la Hoja ginecológica, gestor de la Hoja oftalmológica, gestor de la Hoja general de consulta, gestor de la Hoja uro-genital, gestor de la Hoja ORL, gestor de la Hoja dermatológica, gestor de la Hoja obstétrica, gestor de la Hoja de gastroenterología, gestor de la Hoja de radioterapia y medicina nuclear, gestor de la Hoja para triaje general, gestor de la Hoja hemato-oncológica, gestor de la Hoja hemato-oncológica pediátrica, gestor de la Hoja de cirugía Colo-Rectal. Se encarga además, de buscar resultados de análisis emitidos por el laboratorio y anatomía patológica, crear indicaciones médicas, referencias, contrarreferencia, solicitudes de interconsulta, solicitudes de análisis de laboratorio, órdenes de admisión, entre otras.
Gestor de la Hoja de traumatología y ortopedia	Es el encargado de crear la Hoja de traumatología y ortopedia.
Gestor de la Hoja neurológica	Es el encargado de crear la Hoja neurológica.
Gestor de la Hoja ginecológica	Es el encargado de crear la Hoja ginecológica.
Gestor de la Hoja oftalmológica	Es el encargado de crear la Hoja oftalmológica.
Gestor de la Hoja general de consulta	Es el encargado de crear la Hoja general de consulta.
Gestor de la Hoja uro-genital	Es el encargado de crear la Hoja uro- genital.
Gestor de la Hoja ORL	Es el encargado de crear la Hoja ORL.
Gestor de la Hoja dermatológica	Es el encargado de crear la Hoja dermatológica.
Gestor de la Hoja obstétrica	Es el encargado de crear la Hoja obstétrica.
Gestor de la Hoja de gastroenterología	Es el encargado de crear la Hoja de gastroenterología.
Gestor de la Hoja de radioterapia	Es el encargado de crear la Hoja de radioterapia y medicina nuclear.

y medicina nuclear	
Gestor de la Hoja hemato-oncológica	Es el encargado de crear la Hoja hemato-oncológica.
Gestor de la Hoja hemato-oncológica pediátrica	Es el encargado de crear la Hoja hemato-oncológica pediátrica.
Gestor de la Hoja de cirugía Colo-Rectal	Es el encargado de crear la Hoja de cirugía Colo-Rectal.
Gestor de la Hoja para triaje general	Es el encargado de crear la Hoja para triaje general.
Jefe (a) de servicio	Encargado (a) de generar las estadísticas del servicio y definir el fondo de tiempo para las consultas de cada médico del servicio.
Coordinadora de consulta externa	Encargada de generar las estadísticas de consulta externa.

Tabla 2.2 Actores del sistema

Vista global de los actores

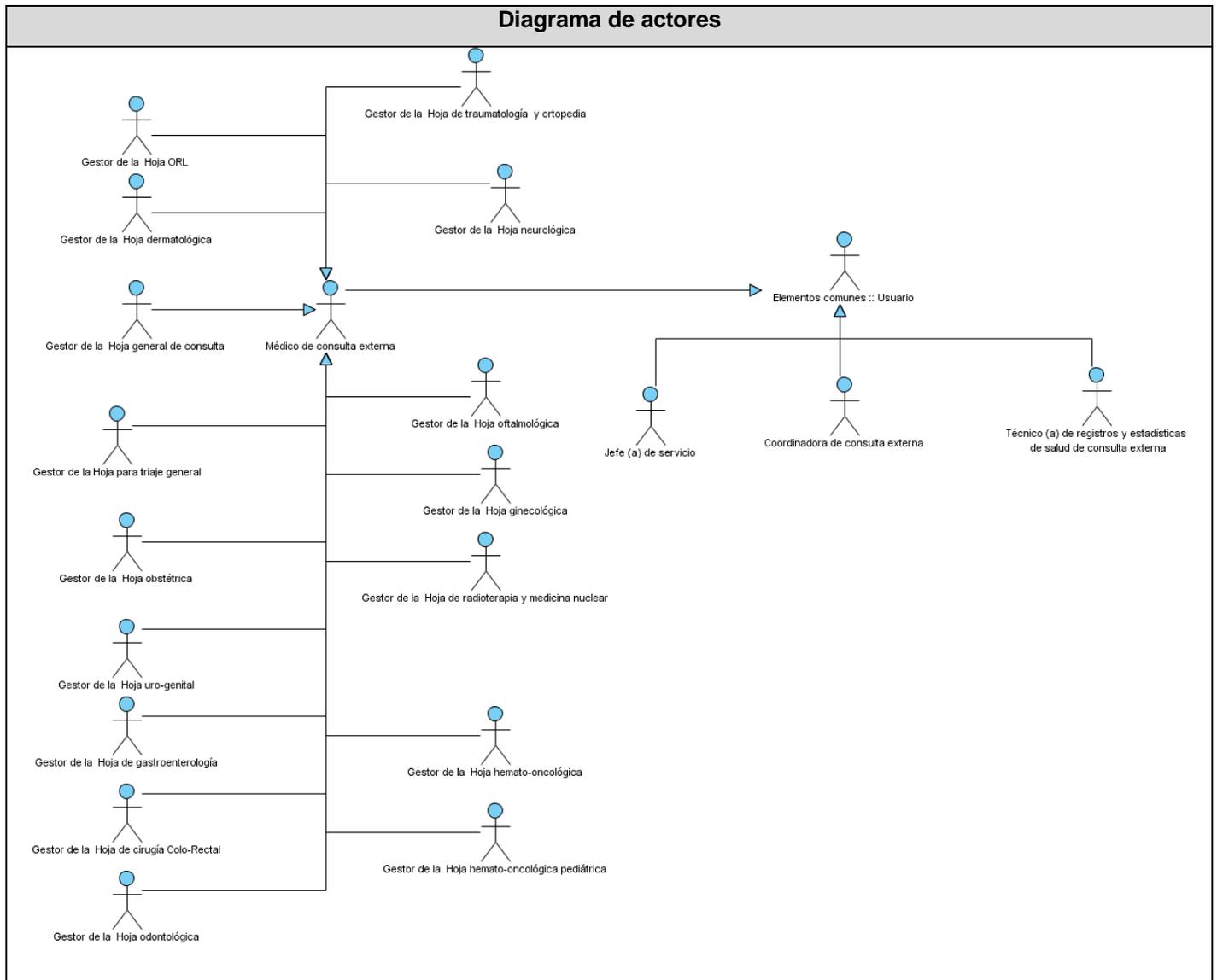


Figura 2.5 Actores del sistema

Diagramas de casos de uso

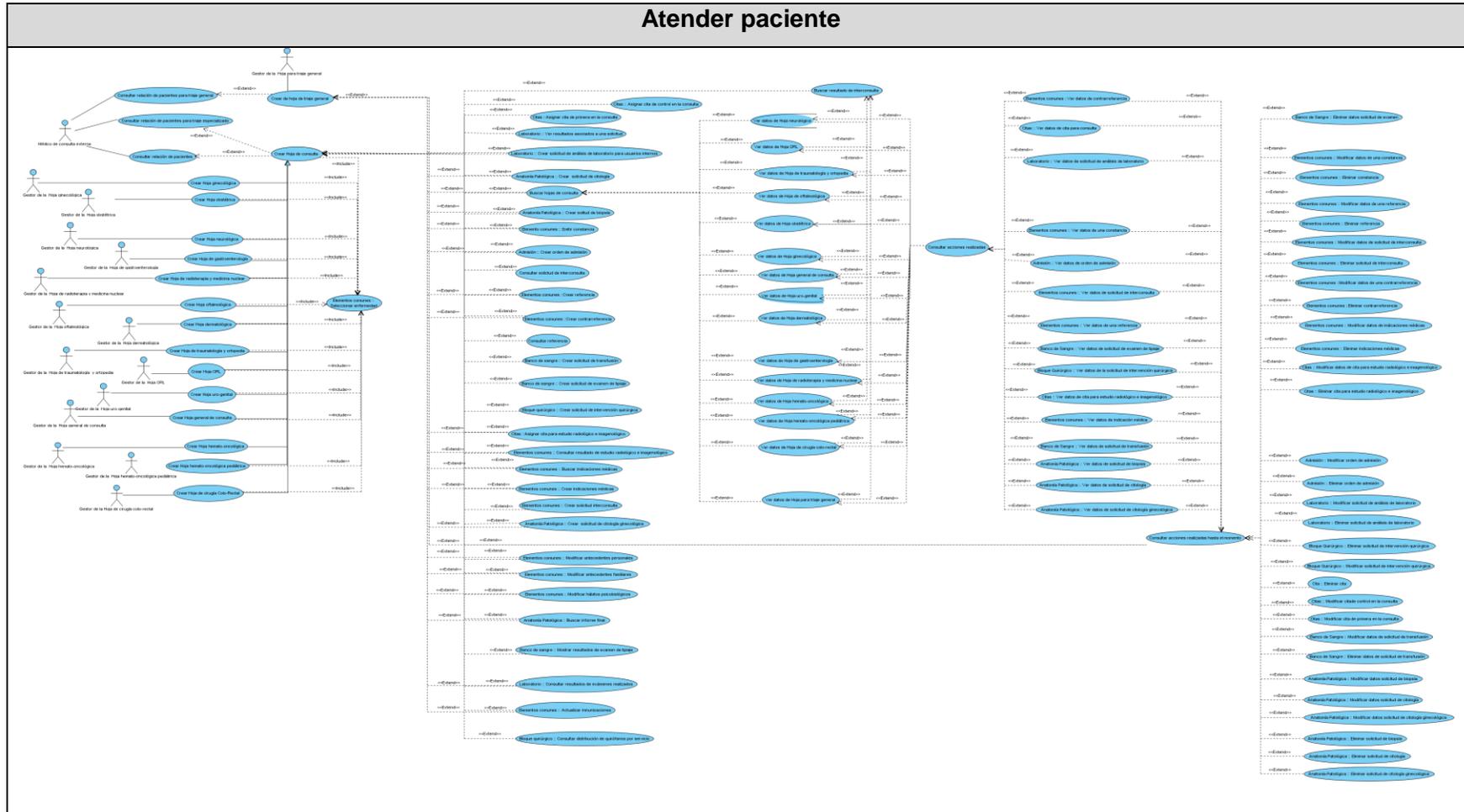


Figura 2.1 Diagrama de casos de uso: Atender paciente

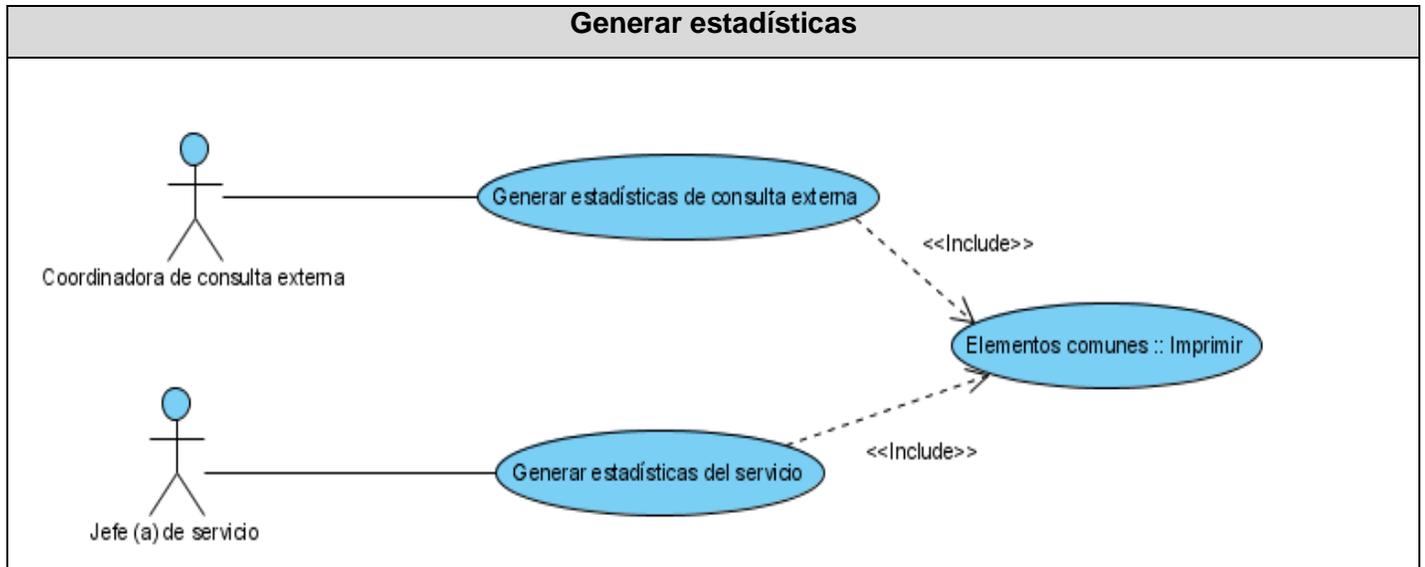


Figura 2.2 Diagrama de casos de uso: Generar estadísticas

2.6 Descripción textual de los casos de uso

CASO DE USO:	Consultar relación de pacientes
Resumen:	El caso de uso inicia cuando el actor accede a la opción Consultar pacientes, el sistema busca en Cita para consulta, los pacientes citados para el médico en la fecha actual y los muestra por tipo de consulta, el sistema permite seleccionar el paciente a consultar, el caso de uso termina.
Precondiciones:	No existen
REFERENCIAS	
Actores:	Médico de consulta externa
Requisitos:	2.1.1

Tabla 2.3 CU_ Consultar relación de pacientes

CASO DE USO:	Crear Hoja general de consulta
Resumen:	El caso de uso inicia cuando el actor accede a la opción Consultar paciente, el sistema brinda la posibilidad de introducir y seleccionar los datos para crear la Hoja general de

	consulta, el actor introduce y selecciona los datos de la Hoja general de consulta, el sistema crea la Hoja general de consulta, el caso de uso termina.
Precondiciones:	Debe haberse citado el paciente.
REFERENCIAS	
Actores:	Gestor de la Hoja general de consulta
Requisitos:	2.1.9

Tabla 2.4 CU_Crear Hoja general de consulta

CASO DE USO:	Buscar hojas de consulta
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar hojas de consulta, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la(s) hoja(s) de consulta(s), el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra la(s) hoja(s) de consulta(s) que cumple con los criterios de búsqueda, el caso de uso termina.
Precondiciones:	No existen
REFERENCIAS	
Actores:	
Requisitos:	2.1.22

Tabla 2.5 CU_Buscar hojas de consulta

CASO DE USO:	Ver datos de Hoja general de consulta
Resumen:	El caso de uso inicia cuando el actor selecciona una hoja de consulta y accede a la opción de Ver hoja de consulta, el sistema muestra los datos de la Hoja general de consulta, el caso de uso termina.
Precondiciones:	Para ver los datos de una Hoja general de consulta, esta debe estar seleccionada.
REFERENCIAS	
Actores:	
Requisitos:	2.1.31

Tabla 2.6 CU_Ver datos de Hoja general de consulta

CASO DE USO:	Crear solicitud de interconsulta
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear Solicitud de interconsulta, el sistema brinda la posibilidad de introducir los datos para crear la solicitud, el actor introduce los datos, el sistema crea la solicitud de interconsulta, el caso de uso termina.
Precondiciones:	No existe
REFERENCIAS	
Actores:	
Requisitos:	2.4.1

Tabla 2.7 CU_Crear solicitud de interconsulta

CASO DE USO:	Consultar solicitud de interconsulta
Resumen:	El caso de uso inicia cuando el actor selecciona la opción Consultar solicitud de interconsulta, el sistema muestra los datos de la Solicitud de interconsulta, el caso de uso termina.
Precondiciones:	Para consultar los datos de la Solicitud de interconsulta, antes debe haberse creado e iniciado una consulta.
REFERENCIAS	
Actores:	
Requisitos:	2.1.20

Tabla 2.8 CU_Consultar solicitud de interconsulta

CASO DE USO:	Crear indicaciones médicas
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear indicaciones médicas, el sistema brinda la posibilidad de introducir los datos para crear las indicaciones médicas, el actor introduce los datos de las indicaciones médicas, el sistema crea las indicaciones médicas, el caso de uso termina.
Precondiciones:	No existe

REFERENCIAS	
Actores:	
Requisitos:	2.5.1

Tabla 2.9 CU_Crear indicaciones médicas

Conclusiones

En este capítulo se especificaron los principales procesos asociados al área de consulta externa. Con el apoyo de diagramas y fichas se logró hacer una representación detallada de cada proceso, que permitió un mejor entendimiento del problema y una rápida identificación de las funcionalidades y propiedades del sistema a desarrollar. A partir de estas funcionalidades se definió el modelo de casos de uso del sistema, con sus diagramas y especificaciones de casos de uso.

CAPÍTULO 3. DISEÑO DEL SISTEMA

En el presente capítulo se describe la concepción arquitectónica definida por el Área temática de hospitales. Además se definen los artefactos concernientes al modelo de diseño: diagramas de clases y de interacción, que son las entradas del modelo de implementación y se describen las clases asociadas a cada diagrama representado.

3.1 Descripción de la arquitectura

La arquitectura de software define al sistema en términos de componentes computacionales e interacciones entre ellos, envuelve un conjunto de decisiones estratégicas de diseño, lineamientos, reglas y patrones que restringen el diseño y la implementación de un software.

Para el desarrollo de este sistema se aplica el patrón Modelo-Vista-Controlador, que separa los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos. La modularidad y el diseño independiente de cada componente, permiten realizar cambios en cualquiera de estos, sin que afecten a otras capas. A partir de este patrón los componentes del sistema se dividen en tres capas: presentación, negocio y acceso a datos.

La capa de presentación está formada por páginas XHTML que contienen formularios. Se desarrolla básicamente con JSF, haciendo uso de la librería de componentes Richfaces, la cual se integra fácilmente con el framework Seam. Además incorpora controles out-of-the-box AJAX-ready y el framework de extensión AJAX para los controles JSF básicos Ajax4Jsf.

La capa de negocio está compuesta por clases controladoras y contiene reglas de gestión de eventos. Usa como framework a Seam, el cual integra todas las tecnologías estándares JSF, EJB3, JPA, además de BPM (Business Process Management) garantizando crear una capa consistente que impide todo tipo de errores.

Para el acceso a datos se usa la implementación de JPA de Hibernate 3.3, minimizando las configuraciones en XML sin chequeo de tipos y usando los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam. Además se pueden establecer validaciones end-to-end gracias a los Hibernate Validators.

El framework Seam es quien rige la comunicación entre los elementos de cada una de las capas, pues permite a través de anotaciones, que las páginas referencien las funcionalidades especificadas en las clases controladoras, y que estas a su vez puedan usar componentes tanto de la capa de acceso a datos como de la capa de negocio.

Una vez definida la concepción arquitectónica del sistema a desarrollar, se pasa al diseño de los casos de uso del sistema.

3.2 Modelo de diseño

El modelo de diseño se realiza para crear especificaciones adicionales que enriquecen el modelo de análisis con detalles próximos a la implementación. Es una solución lógica, que describe como los requisitos funcionales y no funcionales, tienen impacto en el sistema. Para mejorar el diseño y lograr un sistema robusto se aplica durante la realización de diagramas los patrones de diseño permitiendo asignar las responsabilidades de los objetos y diseñar las interacciones entre ellos. Un patrón de diseño no es más que una solución a un problema de diseño no trivial que es efectiva y reusable, aplicándose en diferentes problemas de diseño en distintas circunstancias. [24]

Dentro de los más conocidos están los GRASP (General Responsibility Assignment Software Patterns) que en su traducción al español significa patrones generales de software para asignar responsabilidades y los GOF(Gang of four) abreviatura que proviene del inglés y que significa pandilla de los cuatro, llamado de esta forma debido a cuatro autores que lo definieron.

Los patrones GRASP, tienen una importante utilidad en el diseño realizado ya que se le asigna a cada clase la responsabilidad de realizar tareas que puedan según la información que poseen, además de crear las instancias de otras clases en correspondencia con la responsabilidad dada, poniéndose de manifiesto los patrones Experto y Creador. De esta manera los objetos logran valerse de la información que contienen para realizar las tareas que se les pide, además de utilizar el patrón Creador para definir quién será el responsable de crear una nueva instancia de una clase. Este diseño obtenido cumple con los patrones de Bajo acoplamiento y de Alta cohesión facilitando así la centralización de actividades de cada elemento, y logrando que estos realicen una única tarea dentro del sistema. [25]

Existen varios elementos que componen al modelo de diseño entre los que se destacan los *diagramas de clases de diseño* y los *diagramas de interacción*.

Los diagramas de clases ilustran las relaciones entre las clases e interfaces y se utilizan con la perspectiva de obtener una vista estática del sistema.

Los diagramas de interacción representan las relaciones entre objetos y se utilizan para modelar aspectos dinámicos. Existe dos tipos de diagramas de interacción, el de colaboración y el de secuencia, este último es el más usado pues maneja objetos de diferentes clases y muestra una secuencia lógica de llamadas entre los mismos.

A partir de los principales aspectos a tener en cuenta para la elaboración del modelo de diseño, se define una estructura de paquetes que permita dividir el sistema en fracciones manejables para su futura implementación.

Se utilizan los criterios de empaquetamiento: por procesos y clases. Cada paquete referente a un proceso, está conformado por subpaquetes que responden a las realizaciones de casos de uso, donde cada subpaquete contiene un diagrama de clases del diseño y los respectivos diagramas de secuencia según los escenarios. Cada diagrama de clases se realiza utilizando estereotipos web.

Se concibe un paquete repositorio de clases que contiene tres subpaquetes, el de las vistas con los contenidos web referentes a las páginas clientes y los formularios que las componen, el de las sesiones o controladoras conformado por las clases controladoras autogeneradas, personalizadas y del proceso, y el de las entidades que contiene las clases autogeneradas y personalizadas.

Lo mencionado anteriormente se evidencia en el diagrama de paquetes que se muestra a continuación:

Diagrama de paquetes

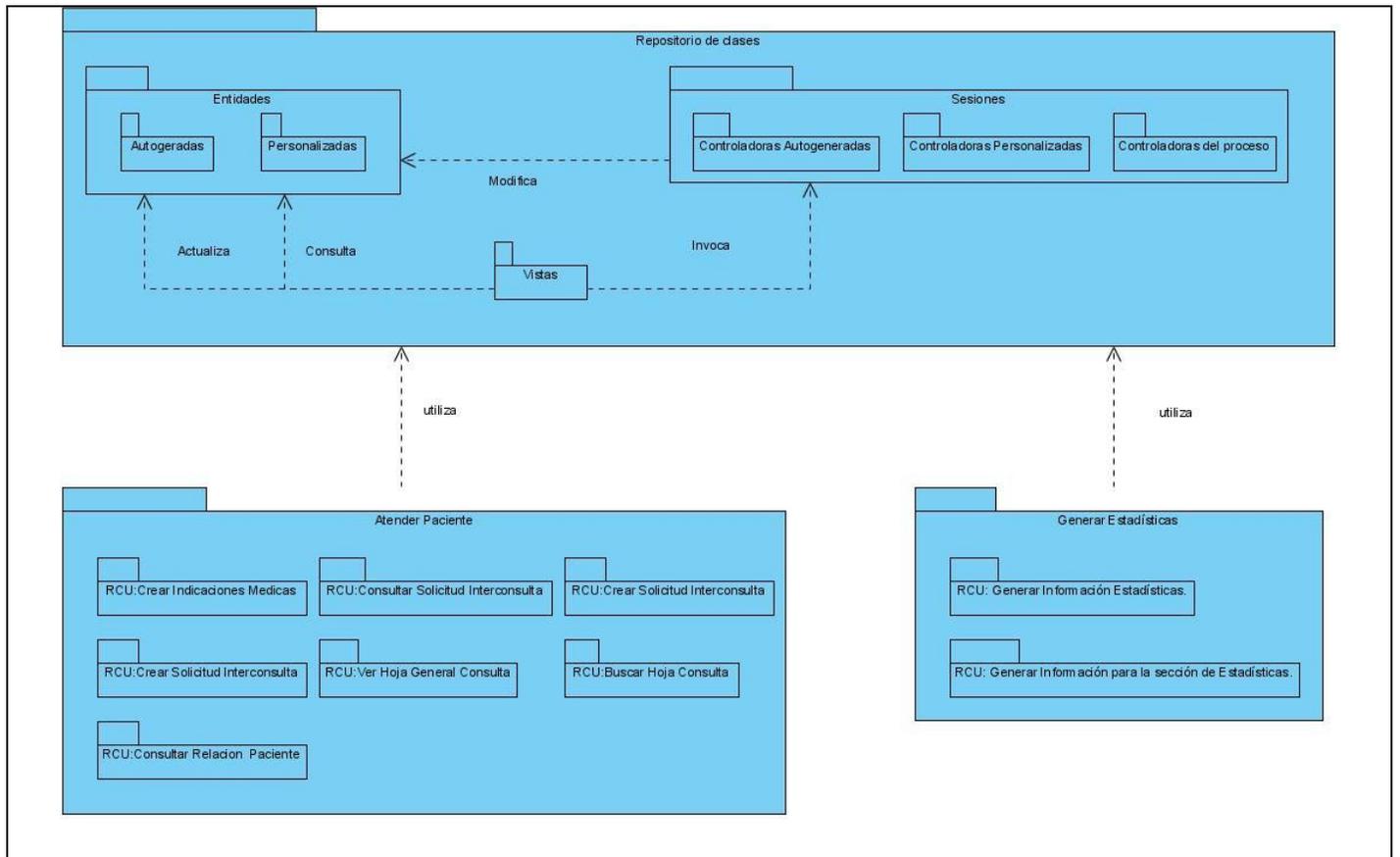


Figura 3.1 Diagrama de paquetes

A continuación se muestran algunas de las realizaciones de los casos de usos del sistema propuesto, relacionados con el proceso de atender paciente:

DCD_Consultar relación de pacientes

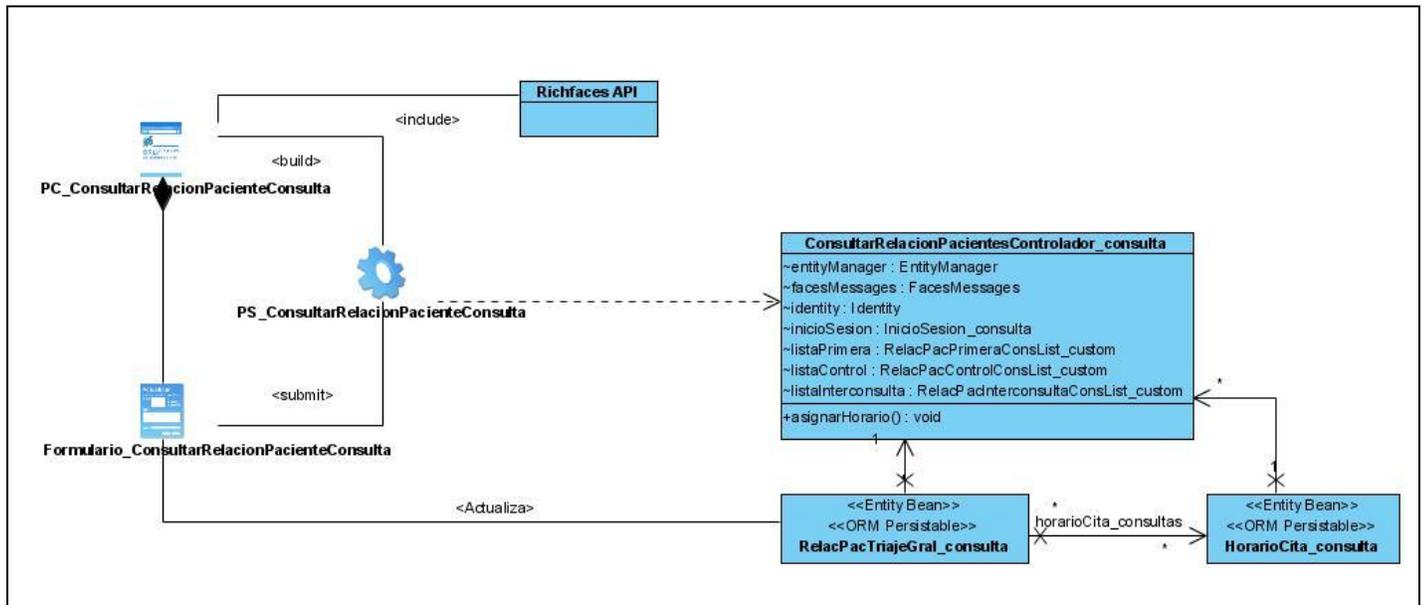


Figura 3.2 Diagrama de clases del diseño: CU_Consultar relación de pacientes

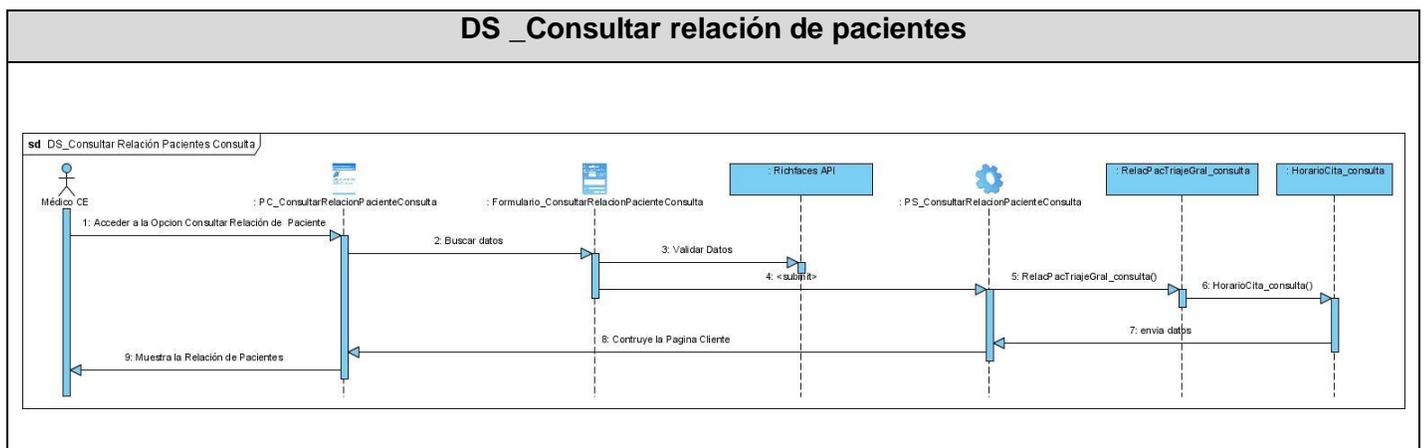


Figura 3.3 Diagrama de secuencia: CU_Consultar relación de pacientes

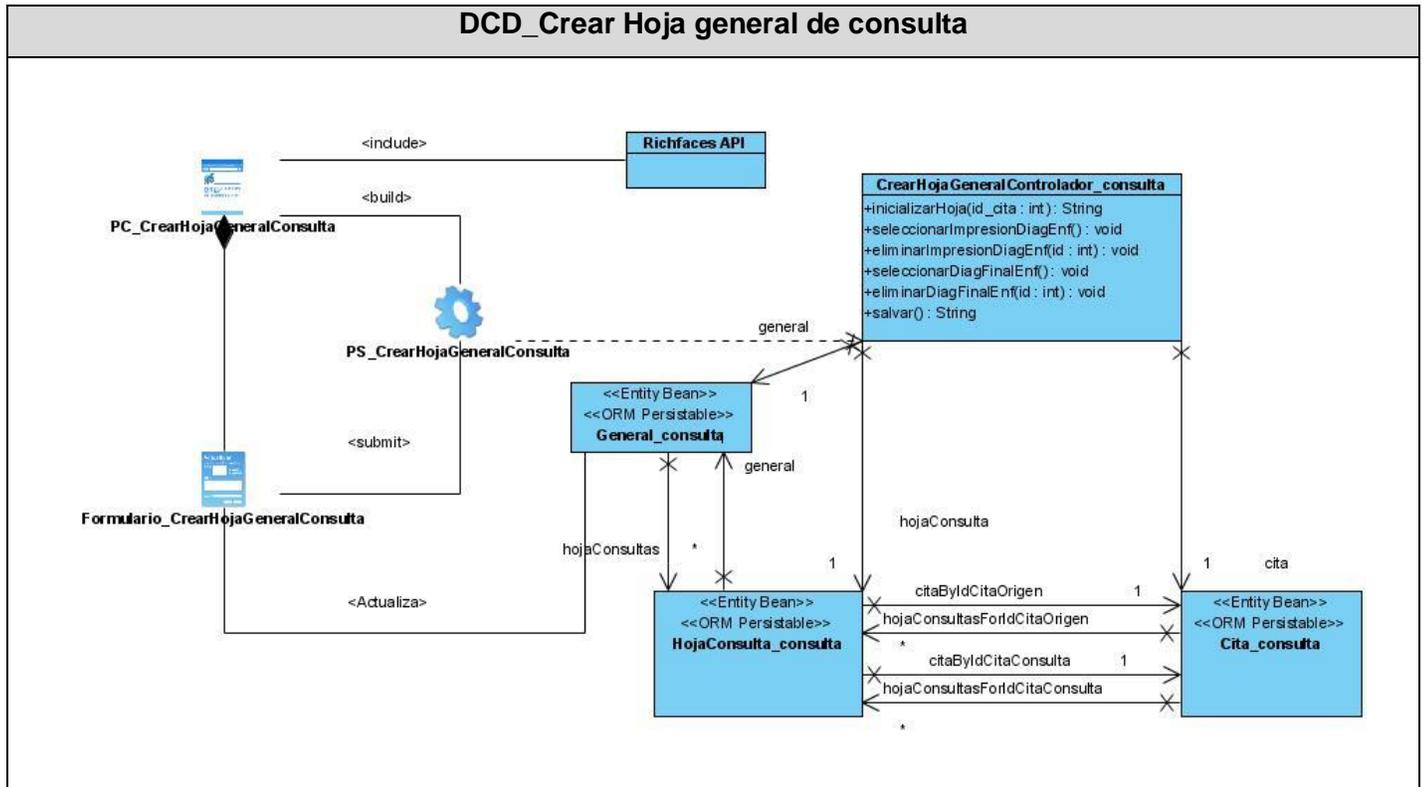


Figura 3.4 Diagrama de clases del diseño: CU_Crear Hoja general de consulta

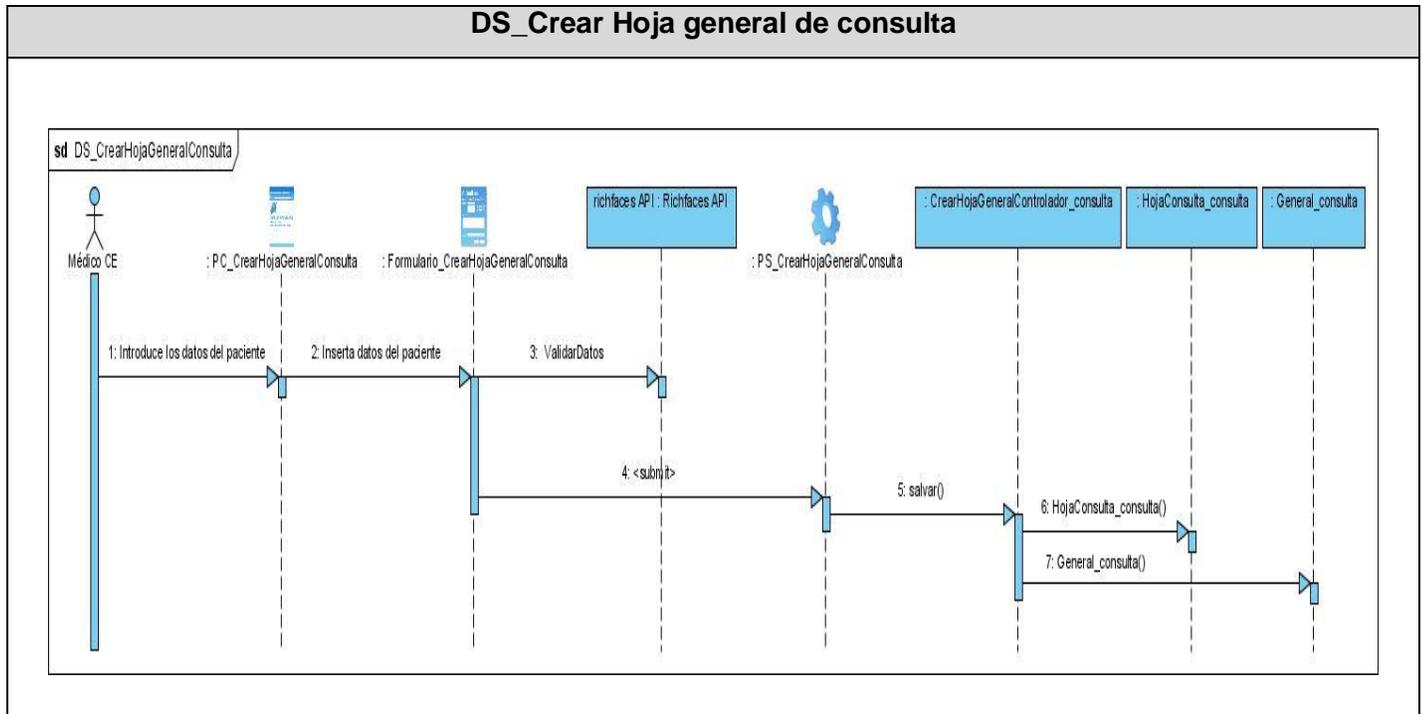
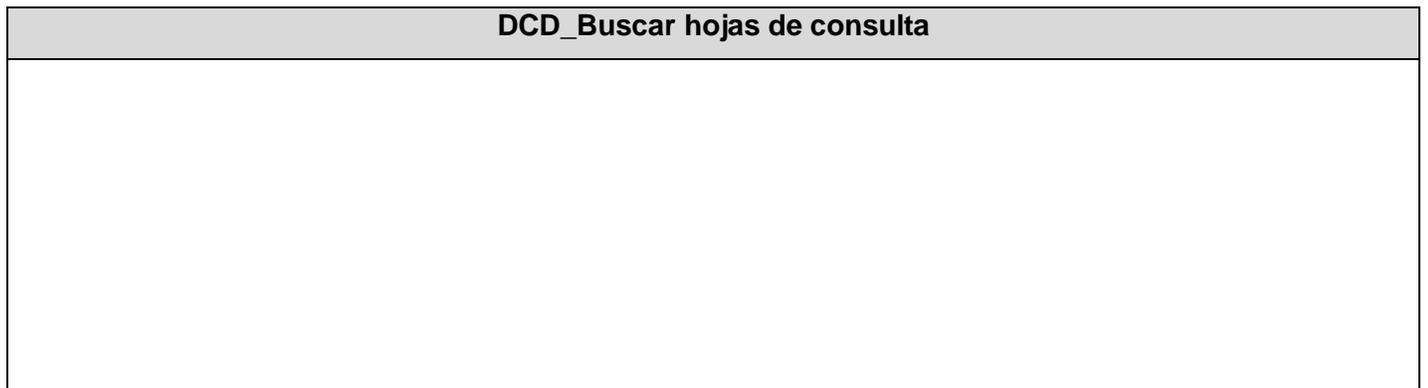


Figura 3.5 Diagrama de secuencia: CU_Crear Hoja general de consulta



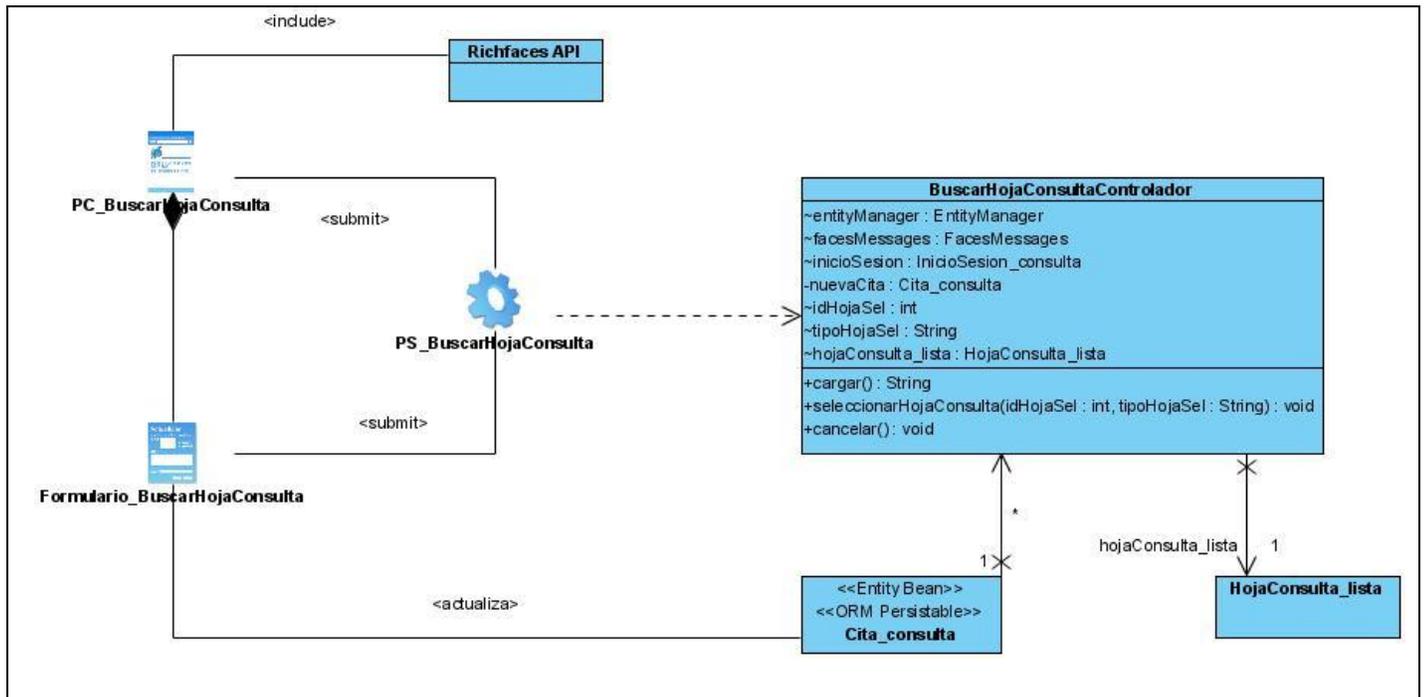


Figura 3.6 Diagrama de clases del diseño: CU_Buscar hojas de consulta

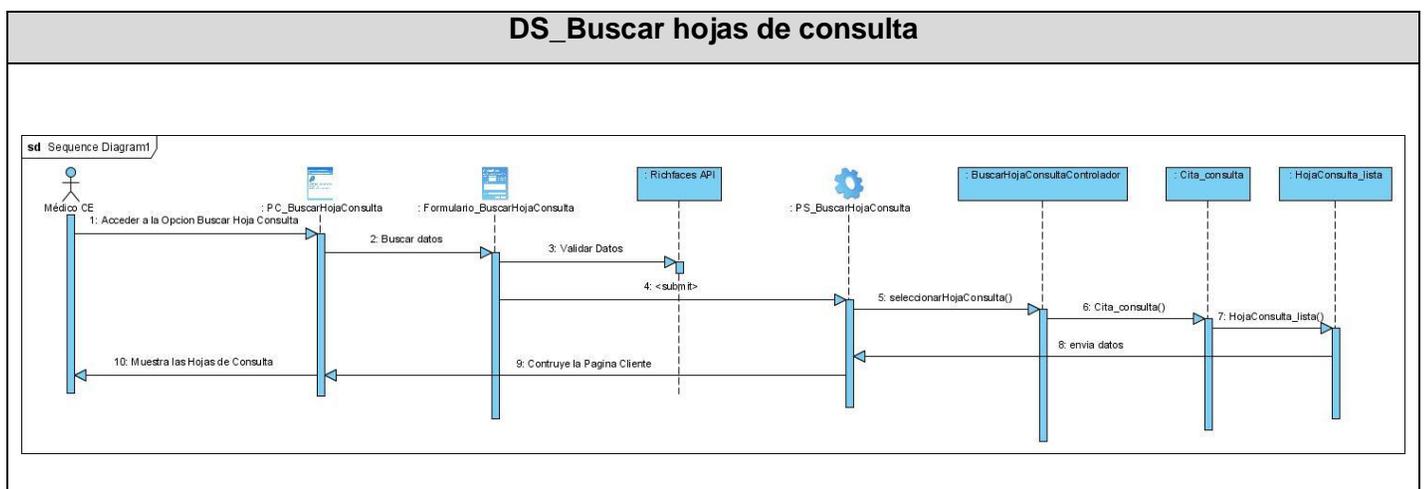


Figura 3.7 Diagrama de secuencia: CU_Buscar hojas de consulta

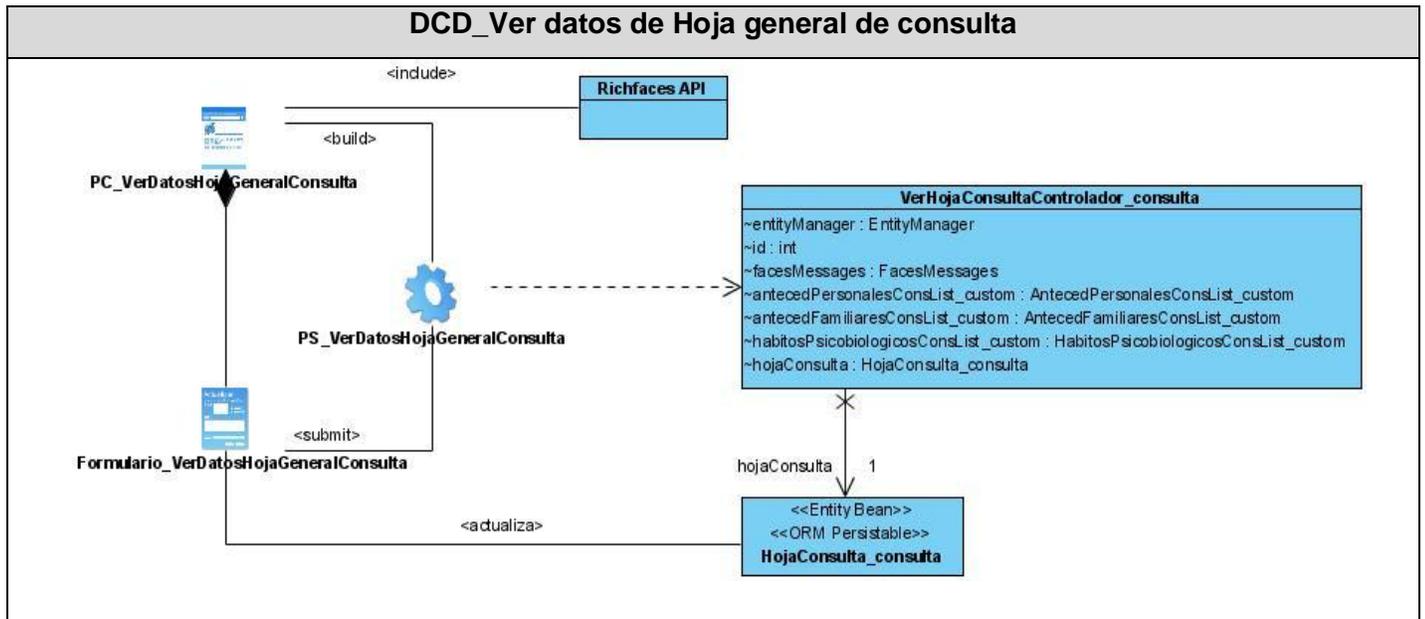
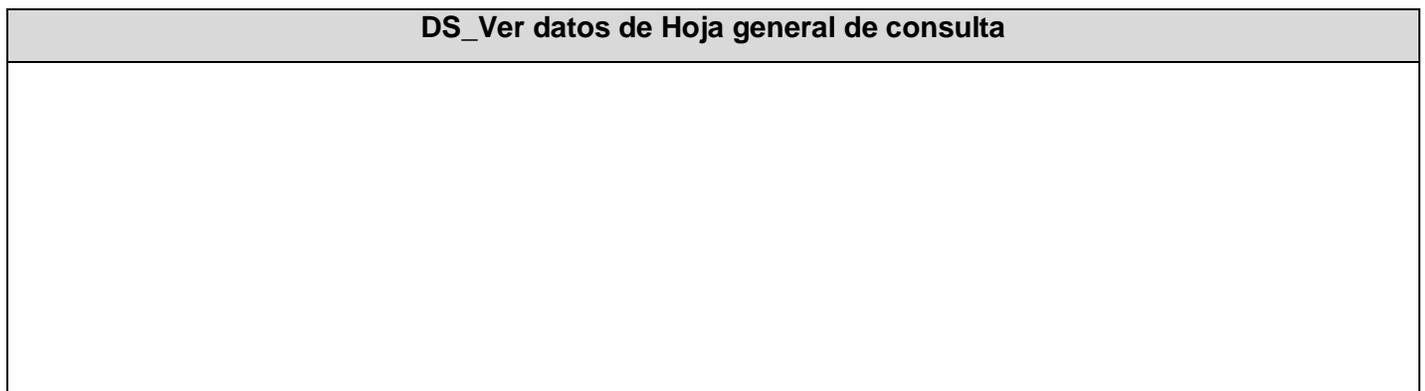


Figura 3.8 Diagrama de clases del diseño: CU_Ver datos de Hoja general de consulta



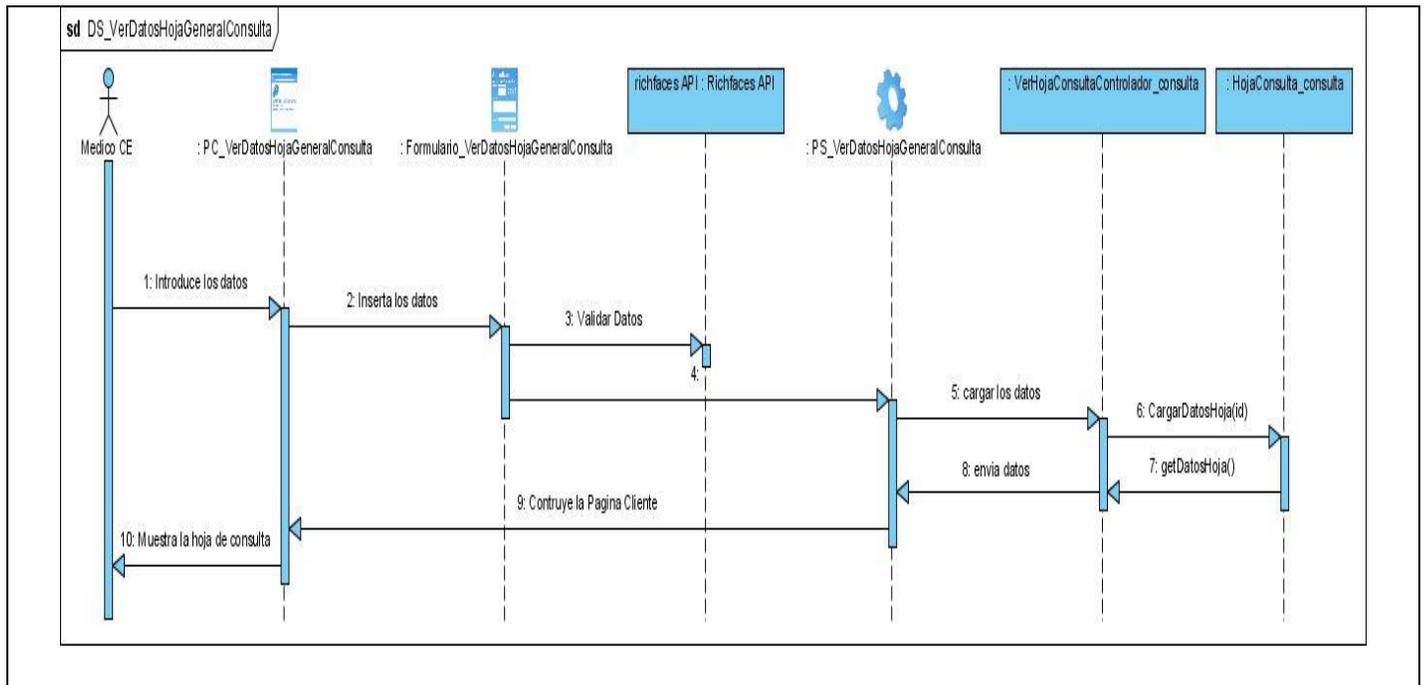


Figura 3.9 Diagrama de secuencia: CU_Ver datos de Hoja general de consulta

DCD_Crear solicitud de interconsulta

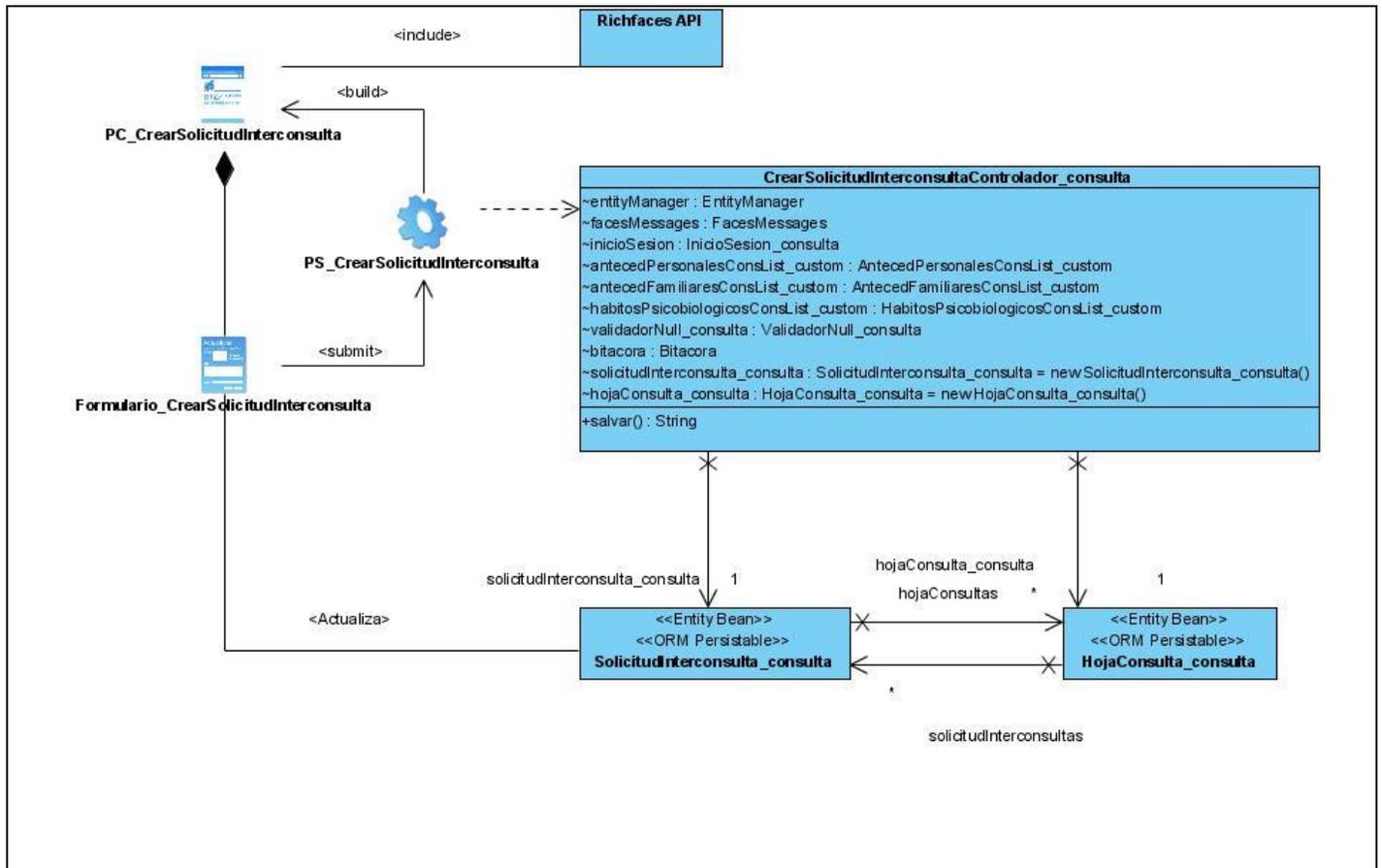


Figura 3.10 Diagrama de clases del diseño: CU_Crear solicitud de interconsulta

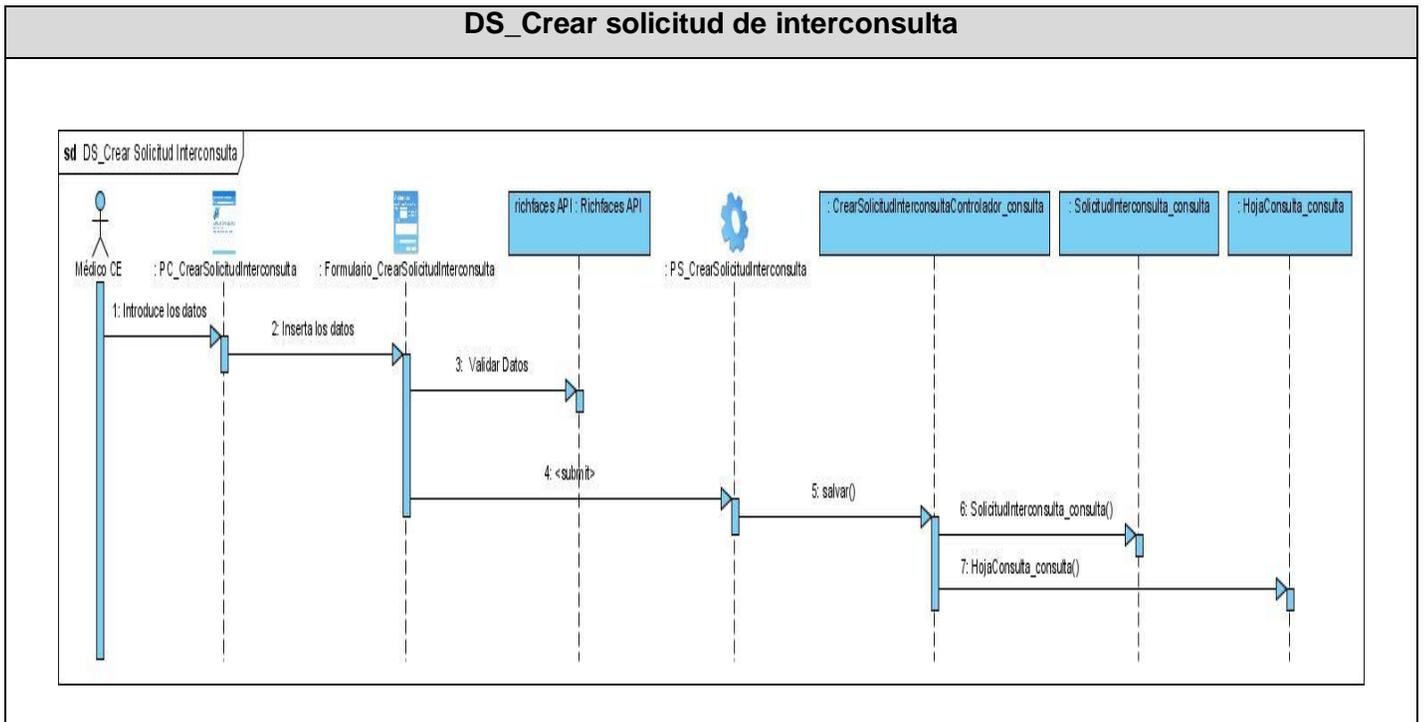


Figura 3.11 Diagrama de secuencia: CU_Crear solicitud de interconsulta

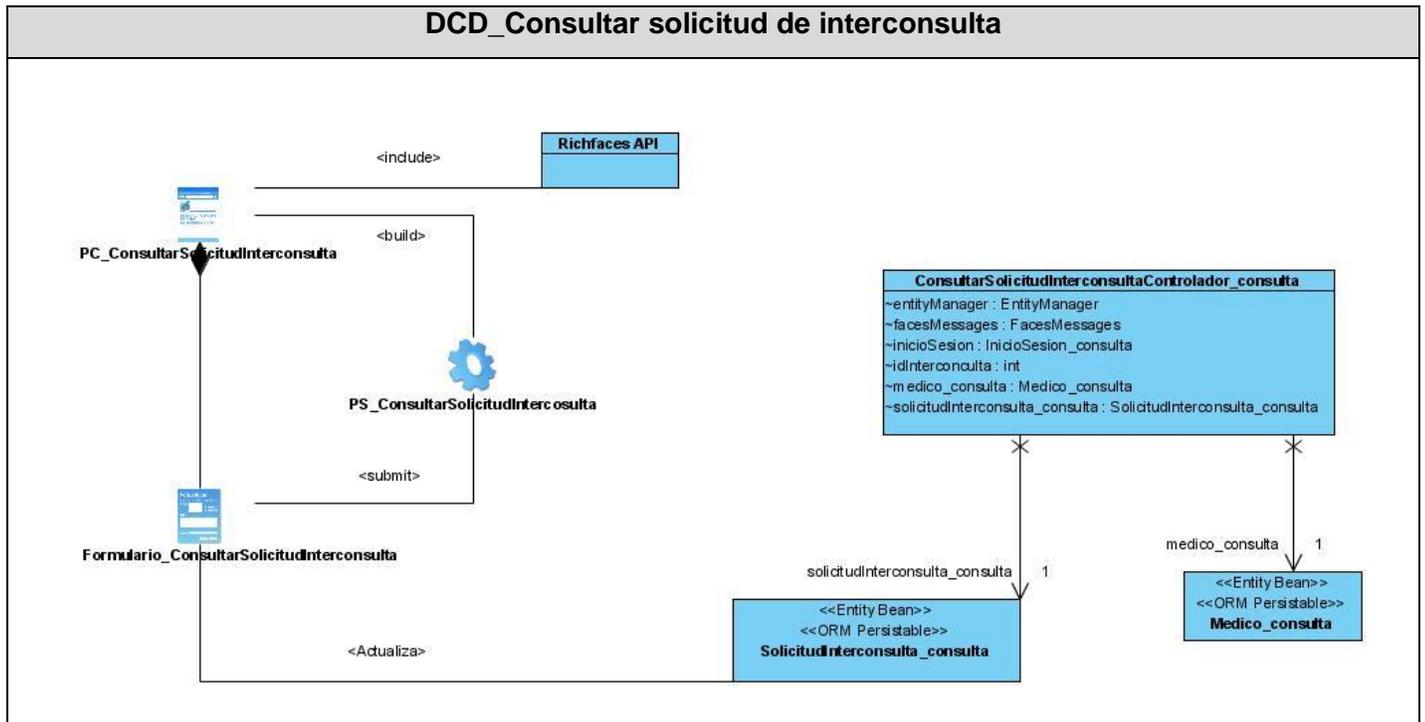


Figura 3.12 Diagrama de clases del diseño: CU_Consultar solicitud de interconsulta

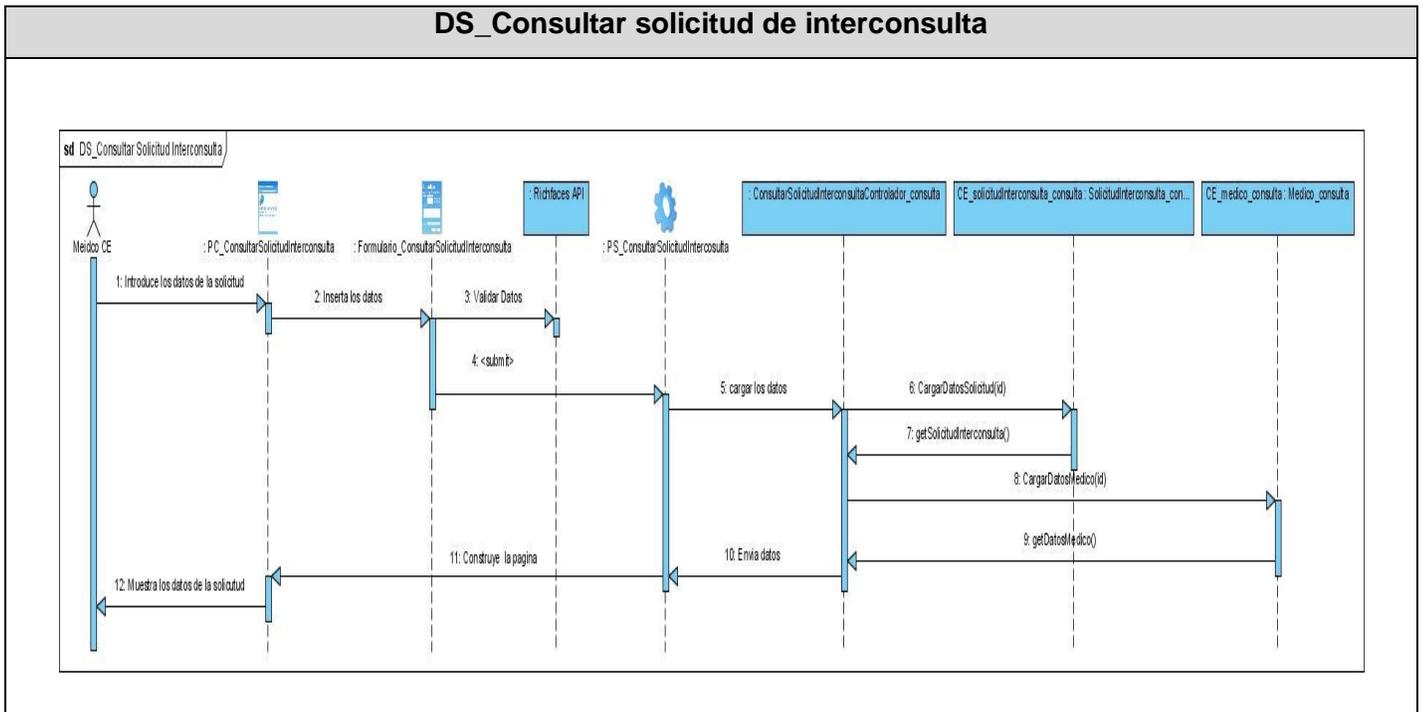


Figura 3.13 Diagrama de secuencia: CU_Consultar solicitud de interconsulta

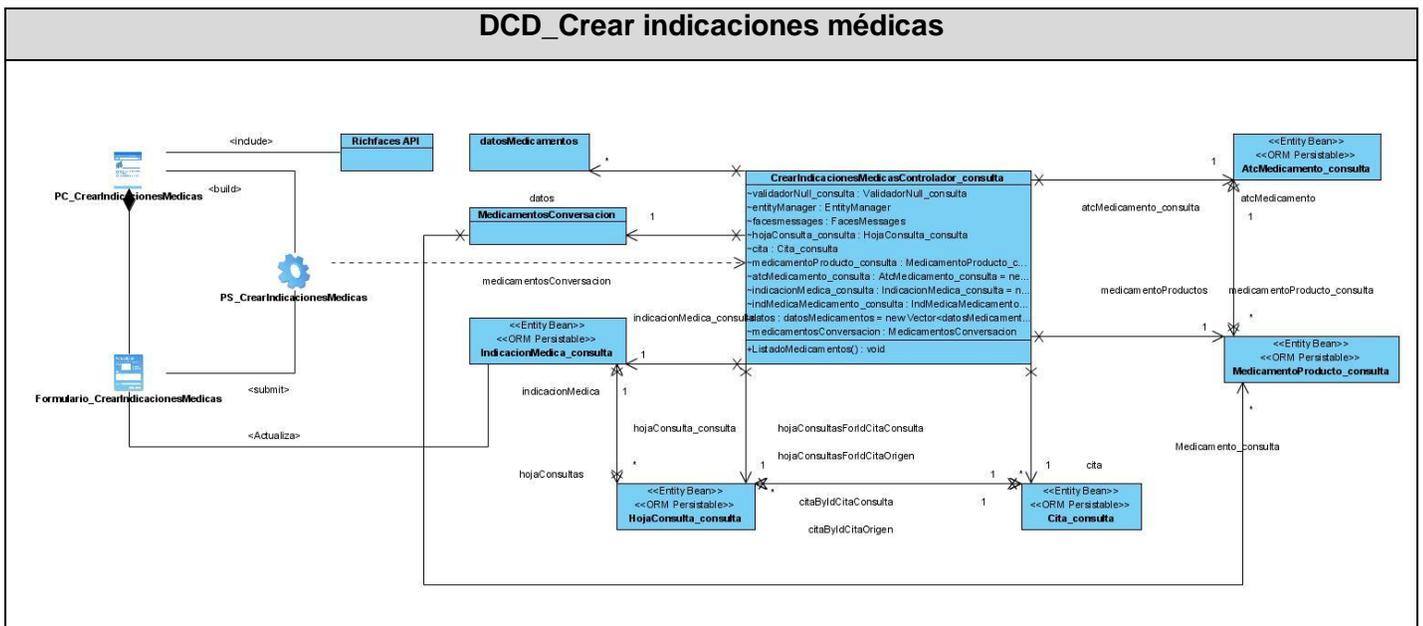


Figura 3.14 Diagrama de clases del diseño: CU_Crear indicaciones médicas

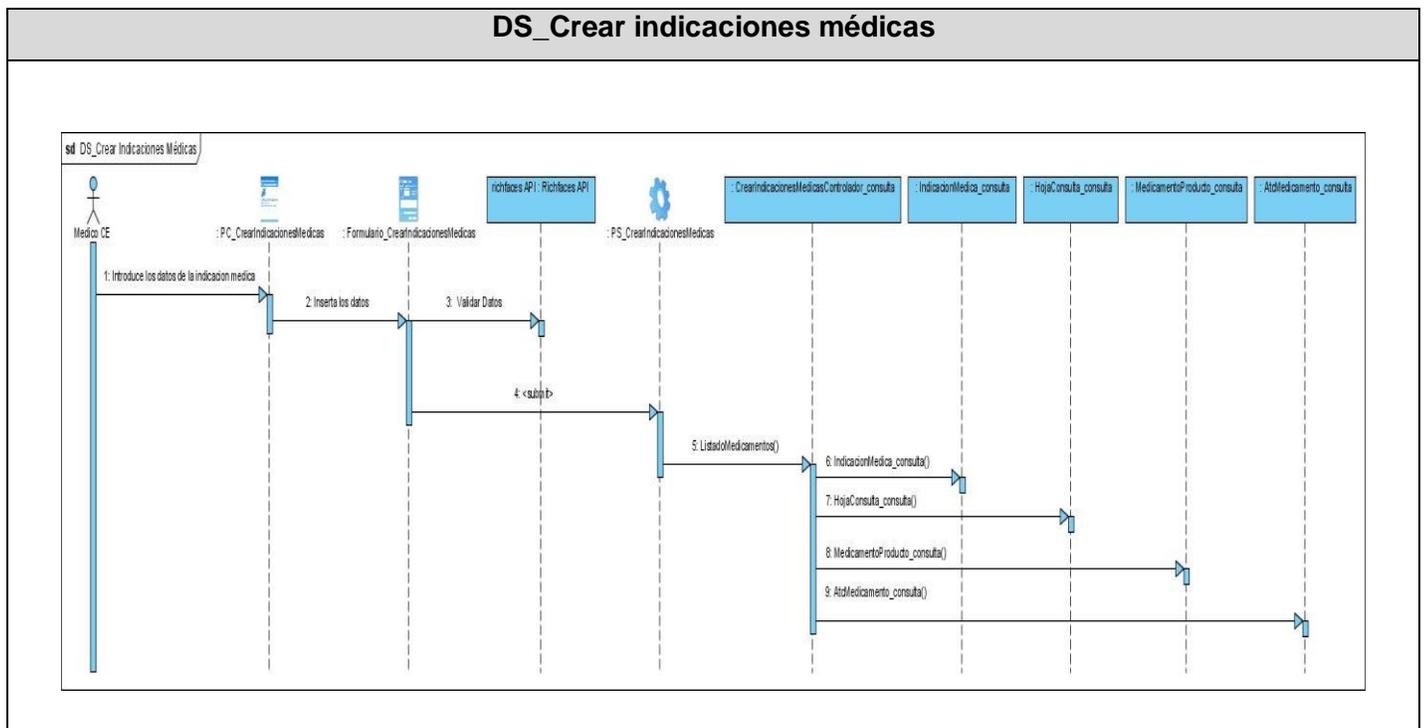


Figura 3.15 Diagrama de secuencia: CU_Crear indicaciones médicas

3.3 Descripción de las clases del diseño

Nombre: CrearIndicacionesMedicasControlador	
Tipo de clase : Controladora	
Atributo	Tipo
idCita	Int

Para cada responsabilidad:	
Nombre:	ListadoMedicamentos()
Descripción:	Se llena los datos que es el listado de los medicamentos seleccionados por el médico.
Nombre:	Salvar ()
Descripción:	Se persisten los objetos medicamentoProducto_consulta, atcMedicamento_consulta, indicacionMedica_consulta, indMedicaMedicamento_consulta, cita, hojaConsulta_consulta, para así salvar los datos en la base de datos y crear la indicación médica.

Tabla 2.10 Clase controladora: CrearIndicacionesMedicasControlador

Nombre: ConsultarRelacionPacientesControlador_consulta	
Tipo de clase : Controladora	
Para cada responsabilidad:	
Nombre:	asignarHorario()
Descripción:	Dado el usuario del médico y su especialidad se hacen tres consultas a la base de datos para asignarle el horario de la cita y la fecha a las listaPrimera, listaControl y listaInterconsulta del médico.

Tabla 2.11 Clase controladora: ConsultarRelacionPacientesControlador_consulta

Nombre: CrearHojaGeneralControlador_consulta	
Tipo de clase: Controladora	
Atributo	Tipo
idCita	int
Para cada responsabilidad:	
Nombre:	inicializarHoja()
Descripción:	Dada la cita se obtienen los antecedentes personales, familiares y hábitos psicológicos del paciente.
Nombre:	seleccionarImpresionDiagEnf()
Descripción:	Se busca la impresión diagnóstica de la enfermedad seleccionada en el BuscadorImpresionDiagnosticaCIE_consulta e insertarla en el listado de impresión diagnóstica impresionDiagEnf.
Nombre:	eliminarImpresionDiagEnf(int id)
Descripción:	Dado el id de la impresión diagnóstica, se elimina del listado de impresión

	diagnóstica impresionDiagEnf.
Nombre:	seleccionarDiagFinalEnf()
Descripción:	Se busca el diagnóstico final de la enfermedad seleccionada en el BuscadorDiagnosticoFinalCIE_consulta e insertarla en el listado de diagnóstico diagFinalEnf.
Nombre:	eliminarDiagFinalEnf()
Descripción:	Dado el id del diagnóstico final, se elimina del listado diagnóstico final diagFinalEnf.
Nombre:	salvar()
Descripción:	Se persisten todos los objetos de examen general declarados así como la Impresión diagnóstica y diagnóstico final para insertarlos en la base de datos y así crear la hoja general.

Tabla 2.12 Clase controladora: CrearHojaGeneralControlador_consulta

Nombre: VerHojaGeneralControlador_consulta	
Tipo de clase: Controladora	
Atributo	Tipo
idConsulta	int
Para cada responsabilidad:	
Nombre:	setIdConsulta()
Descripción:	Dado el id de la consulta que se desea ver se hace una consulta para asignarle a hojaConsulta_consulta dicha consulta y así acceder a sus datos.

Tabla 2.13 Clase controladora: VerHojaGeneralControlador_consulta

Nombre: ConsultarSolicitudInterconsultaControlador_consulta	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	idSolicitud()
Descripción:	Dado el id de la solicitud que se desea consultar se hace una consulta para asignarle a solicitudInterconsulta_consulta dicha solicitud y así acceder a sus datos.

Tabla 2.14 Clase controladora: ConsultarSolicitudInterconsultaControlador_consulta

Nombre: CrearSolicitudInterconsultaControlador_consulta
--

Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	salvar()
Descripción:	Se persisten los objetos declarados para insertarlos en la base de datos y así crear la solicitud de interconsulta.

Tabla 2.15 Clase controladora: CrearSolicitudInterconsultaControlador_consulta

Nombre: CrearIndicacionesMedicas
Tipo de clase: Interfaz
Descripción: Se muestran los datos generales del paciente y le brinda al médico la posibilidad de seleccionar los medicamentos, así como escribir las observaciones y seleccionar la fecha de culminación del tratamiento.

Tabla 2.16 Clase interfaz: CrearIndicacionesMedicas

Nombre: ConsultarRelacionPacientes
Tipo de clase: Interfaz
Descripción: Se muestran los listados de pacientes de primera, de control y de interconsultas.

Tabla 2.17 Clase interfaz: ConsultarRelacionPacientes

Nombre: CrearHojaGeneral
Tipo de clase: Interfaz
Descripción: Esta interfaz está compuesta por tabs en los cuales se registran los síntomas, signos físicos y vitales que presenta el paciente, así como los datos generales del paciente.

Tabla 2.18 Clase interfaz: CrearHojaGeneral

Nombre: VerHojaGeneral

Tipo de clase : Interfaz
Descripción: Esta interfaz muestra los datos generales del médico que ha realizado la consulta así como los datos generales del paciente, está compuesta por tabs en los cuales se muestran los síntomas, signos físicos y vitales que presenta el paciente.

Tabla 2.19 Clase interfaz: Ver-HojaGeneral

Nombre: CrearSolicitudInterconsulta
Tipo de clase: Interfaz
Descripción: Esta interfaz muestra los datos generales del paciente que está siendo atendido y permite registrar los datos referentes a la solicitud de una interconsulta.

Tabla 2.20 Clase interfaz: CrearSolicitudInterconsulta

Nombre: ConsultarSolicitudInterconsulta
Tipo de clase (interfaz)
Descripción: Esta interfaz muestra los datos generales del paciente y del médico que ha realizado la solicitud de interconsulta y se muestra los datos referentes a la solicitud de la interconsulta.

Tabla 2.21 Clase interfaz: ConsultarSolicitudInterconsulta

Conclusiones

En este capítulo fue descrita la arquitectura del sistema propuesto. Se especificó la estructura del diseño, teniendo en cuenta las realizaciones de los casos de usos con sus clases correspondientes y de cada clase se realizó una breve descripción especificando los atributos y métodos asociados.

CAPÍTULO 4. IMPLEMENTACIÓN

En el presente capítulo se introduce el flujo de trabajo de implementación. A partir del resultado del diseño se obtiene el modelo de datos, diagrama de componente y despliegue; estos últimos, con la idea de mostrar las dependencias entre las partes del código y la estructura física del sistema en ejecución. Además se especifica la estrategia de codificación a seguir, así como, elementos de seguridad y captura de errores.

4.1 Modelo de datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. Son una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Un modelo de datos se distingue de otro por el tratamiento que da a sus categorías, ya sea estática, dinámica o de integridad sobre las entidades y las operaciones. [26]

A continuación se muestra parte del modelo de datos propuesto:

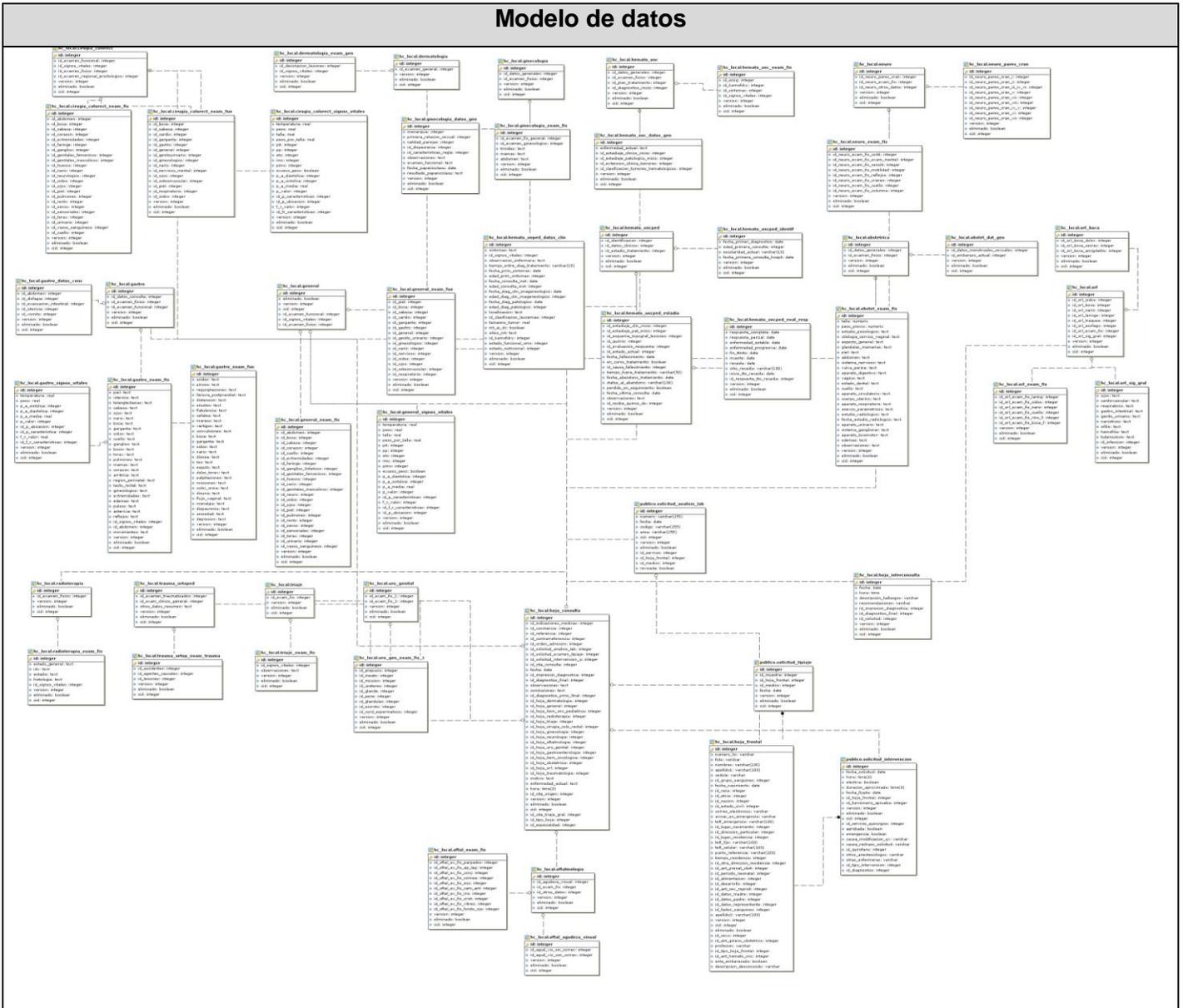


Figura 4.1 Modelo de datos

4.2 Descripción de las tablas

Nombre: hoja_consulta		
Descripción: Contiene los identificadores de las tablas que almacenan los datos referentes a la consulta médica, con el objetivo de acceder a los resultados de las distintas consultas que se le puede realizar al paciente, además almacena la fecha y hora en que es creada, la enfermedad que tiene el paciente en el momento que va hacer consultado y el motivo de la consulta.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_indicaciones_medicas	integer	Identificador de la tabla indicacion_medica.
id_constancia	integer	Identificador de la tabla constancia.
id_referencia	integer	Identificador de la tabla referencia.
id_contrarreferencia	integer	Identificador de la tabla contrarreferencia
id_orden_admision	integer	Identificador de la tabla orden_admision.
id_solicitud_analisis_lab	integer	Identificador de la tabla solicitud_analisis_lab.
id_solicitud_examen_tipaje	integer	Identificador de la tabla solicitud_examen_tipaje.
id_solicitud_intervencion_q	integer	Identificador de la tabla solicitud_intervencion_q.
id_cita_consulta	integer	Identificador de la tabla cita.
fecha	date	Fecha en que es creada la hoja de consulta.
id_impresion_diagnostica	integer	Identificador de la tabla impresion_diagnostica.
id_diagnostico_final	integer	Identificador de la tabla diagnostico_final.
id_observaciones	integer	Identificador de la tabla observaciones, donde se almacenan las observaciones realizadas

		por el médico en la consulta.
id_conclusiones	integer	Identificador de la tabla conclusiones, donde se almacena las conclusiones de la consulta médica.
id_diagnostico_princ_final	integer	Identificador de la tabla diagnostico_princ_final.
id_hoja_dermatologica	integer	Identificador de la tabla dermatología, que es donde se almacenan los datos referentes de la consulta dermatológica.
id_hoja_general	integer	Identificador de la tabla general, que es donde se almacenan los datos referentes a una consulta general.
id_hoja_hem_onc_pediatica	integer	Identificador de la tabla hemato_oncped.
id_hoja_radioterapia	integer	Identificador de la tabla radioterapia, que es donde se almacenan los datos referentes a una radioterapia.
id_hoja_triaje	integer	Identificador de la tabla triaje, la cual almacena los datos referentes a la consulta de triaje.
id_hoja_cirujia_colo_rectal	integer	Identificador de la tabla cirugia_colorect, la cual almacena los datos referentes a esta cirugía.
id_hoja_ginecologia	integer	Identificador de la tabla ginecología, la cual almacena los datos referentes a la consulta ginecológica.
id_hoja_neurologia	integer	Identificador de la tabla neuro, la cual almacena los datos referentes a la consulta neurológica.
id_hoja_ofthalmologia	integer	Identificador de la tabla oftalmología, la cual almacena los datos referentes a la consulta oftalmológica.
id_hoja_uro_genital	integer	Identificador de la tabla uro_genital, la cual almacena los datos referentes a la consulta urológica.
id_hoja_gastroentorologia	integer	Identificador de la tabla gastro, la cual almacena los datos referentes a la consulta

		de gastroenterología.
id_hoja_hem_oncologia	integer	Identificador de la tabla hemato_onc, la cual almacena los datos referentes a la consulta de hemato oncológica.
id_hoja_obstetrica	integer	Identificador de la tabla obstétrica, la cual almacena los datos referentes a la consulta de obstetricia.
id_hoja_orl	integer	Identificador de la tabla orl, la cual almacena los datos referentes a la consulta otorrinolaringología.
id_hoja_traumatologia	integer	Identificador de la tabla trauma_ortoped, la cual almacena los datos referentes a la consulta traumatólogica.
motivo	text	Motivo por la que el paciente acude a la consulta médica.
enfermedad_actual	text	Enfermedad que presenta el paciente.
hora	time	Hora en la que se crea la hoja de consulta.
id_cita_origen	integer	Identificador de la tabla cita, la cual almacena los datos referentes a la cita reservada al paciente para la consulta.

Tabla 3.1 Hoja_consulta

Nombre: cita		
Descripción: Almacenar las citas existentes que han sido planificadas a los médicos.		
Atributo	Tipo	Descripción
id	int	Id de la cita.
eliminado	boolean	Si la cita está eliminada o no.
cid	integer	Para trabajar con la bitácora.
fecha	date	Fecha en que se crea la cita.
hora_entrada	time	Hora de entrada de la cita.

Tabla 3.2 Cita

Nombre: general		
Descripción: Contiene los identificadores de las tablas donde se registran los síntomas, signos físicos y vitales que padece el paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_examen_funcional	integer	Identificador de la tabla general_exam_fun, la cual almacena los síntomas del paciente.
id_signos_vitales	integer	Identificador de la tabla general_signos_vitales, la cual almacena los signos vitales del paciente.
id_examen_fisico	integer	Identificador de la tabla general_exam_fis, la cual almacena los signos físicos del paciente.

Tabla 3.3 General

Nombre: general_exam_fis		
Descripción: Contiene los identificadores de las tablas donde se registran los signos físicos que padece el paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_abdomen	integer	Identificador de la tabla donde se registran los signos físicos del abdomen.
id_boca	integer	Identificador de la tabla donde se registran los signos físicos la boca.
id_cabeza	integer	Identificador de la tabla donde se registran los signos físicos de la cabeza.
id_corazon	integer	Identificador de la tabla donde se registran los signos físicos del corazón.
id_cuello	integer	Identificador de la tabla donde se registran los signos físicos del cuello.
id_extremidades	integer	Identificador de la tabla donde se registran los signos físicos de las extremidades.
id_faringe	integer	Identificador de la tabla donde se registran los signos físicos de la faringe.
id_ganglios_linfaticos	integer	Identificador de la tabla donde se registran los signos físicos de los ganglios linfáticos.
id_genitales_femenino	integer	Identificador de la tabla donde se registran los signos físicos de los genitales femeninos.
id_huesos	integer	Identificador de la tabla donde se registran los signos físicos de los huesos.
id_nariz	integer	Identificador de la tabla donde se registran los signos físicos la nariz.
id_genitales_masculino	integer	Identificador de la tabla donde se registran los signos físicos de los genitales masculinos.

id_neuro	integer	Identificador de la tabla donde se registran los signos neurológicos.
id_oidos	integer	Identificador de la tabla donde se registran los signos físicos relacionados con los oídos.
id_ojos	integer	Identificador de la tabla donde se registran los signos físicos relacionados con los ojos.
id_piel	integer	Identificador de la tabla donde se registran los signos físicos de la piel.
id_pulmones	integer	Identificador de la tabla donde se registran los signos físicos de los pulmones.
id_recto	integer	Identificador de la tabla donde se registran los signos físicos del recto.
id_senos	integer	Identificador de la tabla donde se registran los signos físicos de los senos.
id_sensoriales	integer	Identificador de la tabla donde se registran los signos físicos del estado de los sensoriales.
id_torax	integer	Identificador de la tabla donde se registran los signos físicos del tórax.
id_urinario	integer	Identificador de la tabla donde se registran los signos físicos del sistema urinario.
id_vasos_sanguineos	integer	Identificador de la tabla donde se registran los signos físicos de los vasos sanguíneos.

Tabla 3.4 General_exam_fis

Nombre: general_exam_fun		
Descripción: Contiene los identificadores de las tablas donde se registran los síntomas que padece el paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_piel	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto

		a la piel.
id_boca	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a la boca.
id_cabeza	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a la cabeza.
id_cardio	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema cardiovascular.
id_garganta	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a la garganta.
id_gastro	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema gastrointestinal.
id_general	integer	Identificador de la tabla donde se registran los síntomas generales asociados a la enfermedad actual del paciente.
id_genito_urinario	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema genitourinario.
id_ginecológico	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema ginecológico.
id_nariz	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a la nariz.
id_nervioso	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema nervioso.
id_oidos	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a los oídos.
id_ojos	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto a los ojos.
id_osteomuscular	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto

		al sistema osteomuscular.
id_respiratorio	integer	Identificador de la tabla donde se registran los síntomas que presenta el paciente con respecto al sistema respiratorio.

Tabla 3.5 General_exam_fun

Nombre: solicitud_interconsulta		
Descripción: Se almacenan los datos referentes a la solicitud de interconsulta realizada a un paciente, además de los id de la tabla de los servicios, tanto solicitado como solicitante.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_servicio_solicitante	integer	Identificador de la tabla Servicio, representa al servicio que hace la solicitud.
id_servicio_solicitado	integer	Identificador de la tabla Servicio, representa al servicio que es solicitado.
motivo_interconsulta	varchar	Se almacena el motivo de la interconsulta.
resumen_caso	varchar	Se almacena el resumen médico del caso que es atendido en la interconsulta.
fecha	date	Se almacena la fecha en que es creada la solicitud de interconsulta.
hora	time	Se almacena la hora en que se crea la solicitud de interconsulta.
ubicación_paciente	varchar	Número que muestra el servicio, la cama y otros datos de donde se encuentra el paciente.
aceptada	boolean	Si la solicitud es aceptada.

Tabla 3.6 Solicitud_interconsulta

Nombre: ind_medica_med medicamento		
Descripción: Se almacenan los datos referentes a los medicamentos mandados por el médico al paciente y contiene el identificador de la indicación médica que ha sido creada.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
id_indicacion_médica	integer	Identificador de la Indicación Médica que ha sido creada.
id_med medicamento	integer	Identificador del medicamento que el médico le ha mandado al paciente.
dosis	integer	Dosis del medicamento que el paciente debe ingerir.
id_unidad_tiempo	integer	Identificador del nomenclador unidad_tiempo, el cual almacena la cantidad de tiempo que el paciente se tomará el medicamento.
valor_tiempo	integer	La cantidad de medicamento que el paciente debe ingerir en un tiempo determinado.
id_via_administracion	integer	Identificador del nomenclador via_administracion, el cual almacena la vía por la cual el paciente debe ingerir el medicamento.
id_unidad_medida_medic	integer	Identificador del nomenclador unidad_pres_medida, el cual almacena la unidad de medida de algunos medicamentos.

Tabla 3.7 Ind_medica_med medicamento

Nombre: indicacion_medica

Descripción: Se almacenan los datos referentes a la indicación médica y contiene el identificador del médico que realiza dicha indicación.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
fecha	date	Se almacena la Fecha en que es creada la indicación médica.
Observaciones	text	Se almacenan las observaciones que desee emitir el médico.
id_medico	integer	Identificador del médico que realiza la indicación médica.

Tabla 3.8 Indicacion_medica

Nombre: unidad_tiempo		
Descripción: Almacena la cantidad de tiempo que el paciente se tomará el medicamento.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
valor	varchar	Cantidad de tiempo que el paciente se tomará el medicamento.

Tabla 3.9 Unidad_tiempo

Nombre: unidad_pres_medida		
Descripción: Identificador del nomenclador unidad_pres_medida, el cual almacena la unidad de medida de algunos medicamentos.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.
cid	integer	Identificador único de la conversación.
valor	varchar	Unidad de medida de medicamentos.

Tabla 3.10 Unidad_pres_medida

Nombre: via_administracion		
Descripción: Almacena la vía por la cual el paciente debe ingerir el medicamento.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Si la información está eliminada.

cid	integer	Identificador único de la conversación.
valor	varchar	Vía por la cual el paciente debe ingerir el medicamento.

Tabla 3.11 Via_administracion

4.3 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, se implementan en términos de componentes. Además representa la organización de los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, así como la dependencia que se establece entre cada componente.

Diagrama de despliegue

Un diagrama de despliegue representa la distribución física del sistema en términos de cómo se distribuirán las funcionalidades entre los nodos, cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitarán para el despliegue del sistema.

La arquitectura en tiempo de ejecución de la solución propuesta se modela haciendo usos de tres elementos de hardware: computadora cliente, servidor de aplicaciones y servidor de base de datos, además de contar con un dispositivo de conexión, en este caso la impresora.

La comunicación entre los nodos se rige por los protocolos HTTP para la asociación entre la computadora cliente y el servidor de aplicaciones y TCP/IP para la conexión entre los dos servidores. La computadora cliente y la impresora se conectan a través de los puertos USB o LPT.

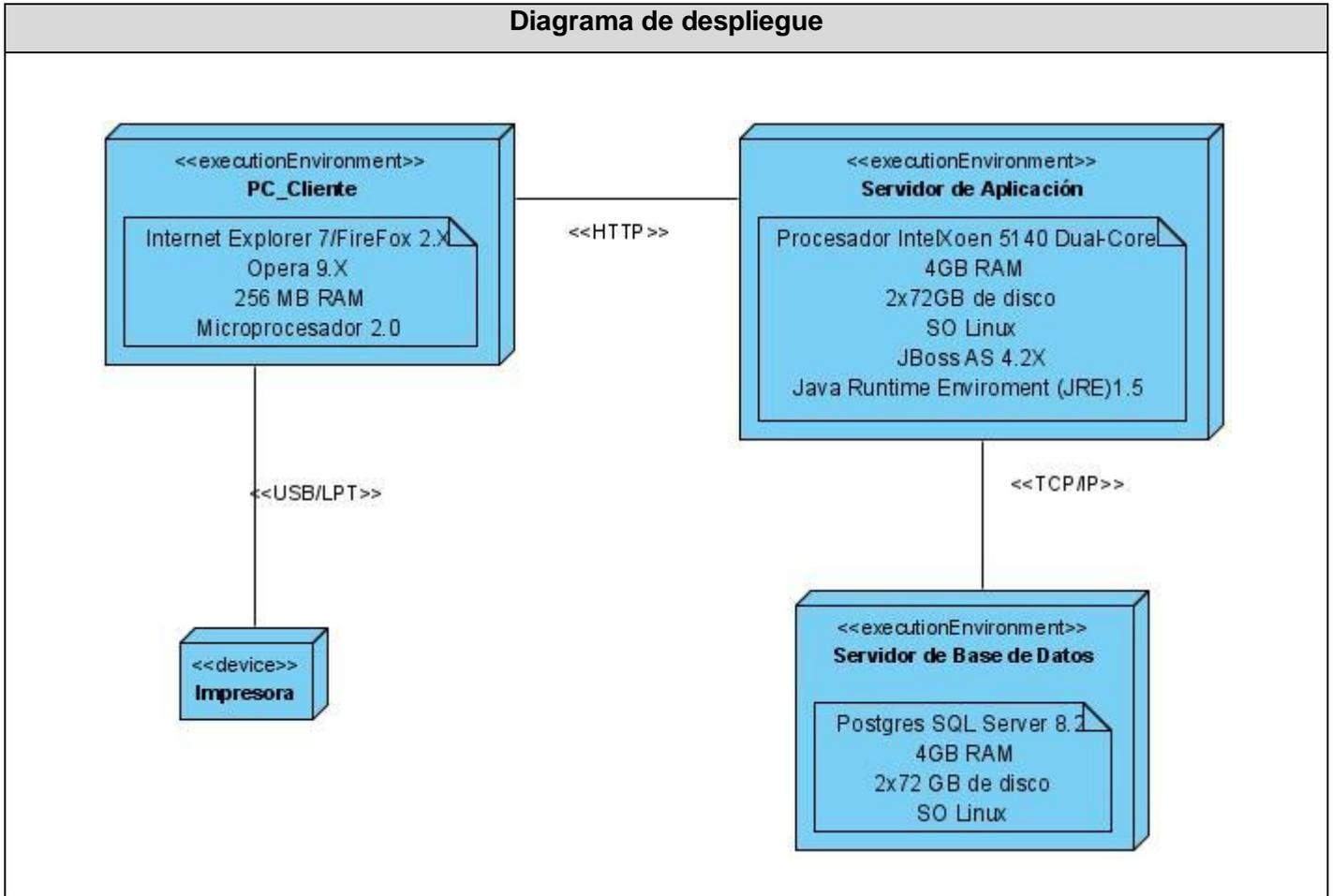


Figura 4.2 Diagrama de despliegue

4.4 Tratamiento de excepciones

El proceso de ejecución del sistema puede verse interrumpido por un error u otro evento, que impide el correcto funcionamiento de la aplicación, este suceso se conoce como excepción. El tratamiento de estas excepciones permite que el programa continúe funcionando aún teniendo un error, pues se transfiere el control, al manipulador adecuado donde va a ser tratado.

En el sistema propuesto, el tratamiento de excepciones se realiza mediante la sentencia try-catch-finally, try para generar la excepción en caso de que alguna sentencia que se encuentra dentro del bloque lo requiera y catch para atraparla.

El tratamiento de las excepciones se realiza en todas las porciones del código fuente donde pueda ocurrir un error, fundamentalmente en aquellas donde se realice inserción y selección de datos en una base s de datos, con el fin de evitar que los datos manipulados no estén en el formato correcto.

El sistema cuenta con un archivo XML (page.xml) que contiene la configuración de todos los mensajes que se deben mostrar según el tipo de excepción, así como la página a la que se redirecciona en caso de la aparición de algún error inesperado. También se hace uso del componente Seam FacesMessages, proveniente del framework Seam. Este se encarga de capturar los mensajes de error o de notificación que surgen en el código de las clases controladoras, y permite mostrarlos en vistas o interfaces de usuario.

4.5 Seguridad

Para obtener un sistema seguro se debe garantizar la integridad, confidencialidad y disponibilidad de la información. La mezcla de estas características trae consigo que la información sea modificada y vista por el personal autorizado; de forma controlable y que esté disponible cuando se necesite.

A continuación se listan un conjunto de acciones llevadas a cabo con el objetivo de que el usuario final tenga la oportunidad de contar con un software seguro:

- Se mantendrá un control a nivel de usuario y contraseñas, permitiendo solamente el acceso a los datos y áreas a aquellas personas con la autoridad establecida de acuerdo a las funciones que realizan y los privilegios que poseen.

- Las contraseñas solo podrán ser cambiadas por el administrador del sistema al igual que los permisos para acceder a los datos.
- Se define un segundo nivel de seguridad en las estaciones de trabajo, posibilitando solo la ejecución de aplicaciones que hayan sido definidas por la estación en cuestión.
- El sistema permitirá registrar las trazas de todas las actividades y accesos a la información realizados en todo momento por cada usuario que accede a la aplicación.
- La información que se intercambia entre el sistema y otros sistemas externos será cifrada, evitando las posibilidades de acceso y modificación a la misma.

4.6 Estrategias de Codificación. Estándares y estilos a utilizar

Un estándar de codificación define un conjunto de reglas establecidas para la escritura del código fuente. Estas reglas son utilizadas como modelo de trabajo para garantizar un estilo de programación homogéneo durante la implementación del sistema y para entender el código por parte de otros programadores. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código adecuado es de gran importancia para la calidad del software y para obtener un buen rendimiento.

Para el desarrollo del sistema presentado, se utiliza un estándar de codificación del cual se describen a continuación un conjunto de elementos asociados:

Como idioma se utiliza el español y las palabras no se acentúan.

La indentación tiene como objetivo lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento, para esta se definieron los siguientes aspectos:

- Inicio y fin de bloque: se dejan dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Sucede lo mismo para las instrucciones if, else, for, while, do while, switch, foreach.
- Aspectos generales: no se debe usar el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla. Los inicios ({) y cierre (}) de ámbito deben estar alineados

debajo de la declaración a la que pertenecen y en caso de existir una sola instrucción deben evitarse. No se coloca ({) en la línea de un código cualquiera, este tiene su línea propia.

También se tuvo en cuenta los comentarios, separadores, líneas, espacios en blanco y márgenes que permiten establecer una forma común de comentar el código, para que sea comprensible a la hora de leerlo. Relacionados a los mismos se encuentran los siguientes aspectos.

- Ubicación de comentarios: estos pueden estar al inicio de cada clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro), y al final de cada bloque de código.
- Líneas en blanco: se emplean antes y después de la declaración de métodos, clases y estructuras.
- Espacios en blanco: se establecen entre operadores lógicos y aritméticos para lograr una mayor legibilidad del código.
- Aspectos generales:
 - Se evita comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.
 - No se usan espacios en blanco después del corchete abierto y antes del cerrado, después del paréntesis abierto y antes del cerrado y antes de un punto y coma.

Sobre el trabajo con variables y constantes se tuvo en cuenta los siguientes aspectos:

- Apariencia de variables:
 - El nombre de las variables comienzan con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará la notación CamellCasing.¹

¹ **Notación CamellCasing:** Especifica que la palabra de inicio del identificador comienza con minúscula. Si el identificador está compuesto por más de una palabra entonces éstas deben comenzar con mayúsculas.

- Apariencia de constantes: se declaran con todas sus letras en mayúscula.
- Aspectos generales: se debe emplear un nombre para variables y constantes que con solo leerlo se conozca el propósito de las mismas.

Para el uso de clases y objetos se definieron un conjunto de reglas que garantizan nombrar las clases e instancias de la misma forma para toda la aplicación:

- Apariencia de clases y objetos:
 - Nombre de clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará la notación PascalCasing.²
- Apariencia de atributos: el nombre de estos en las clases comienzan con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará la notación CamellCasing.
- Apariencia de las funciones
 - Primera letra en mayúscula.
 - El nombre de las funciones con verbos que denoten la acción que hace la función y se empleará la notación PascalCasing.
 - Las funciones que obtienen algún dato emplean el prefijo get y si fijan algún valor se emplea el prefijo set.
- Declaración de parámetros en funciones: se declaran agrupados por tipos, poniendo primero los string, después los numéricos, además de agruparlos según valores por defecto
- Aspectos generales: el nombre que se emplea para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

² **Notación PascalCasing:** Especifica que la palabra de inicio del identificador comienza con mayúscula. Si el identificador está compuesto por más de una palabra entonces éstas deben comenzar con mayúsculas.

También se tuvo en cuenta elementos para la base de datos, tablas, esquemas y campos que son mencionados a continuación:

- Apariencia de la base de datos: el nombre comienza con el prefijo `bd` -que representa el tipo- a continuación `underscoard`³ y luego el nombre completamente en minúscula. Los nombres son cortos y descriptivos.
- Apariencia de las tablas: el nombre comienza con el prefijo `tb` seguido de `underscore` y luego se escriben todas las letras en minúscula, en caso de ser un nombre compuesto se utiliza `underscore` para separarlo.
- Tablas que representen relaciones: el nombre empleado para estas tablas comienza con el prefijo `tr` seguido de `underscore` y la concatenación del nombre de las dos tablas que la generaron separados por `underscore`, todo en minúscula.
- Tablas que representen nomencladores: el nombre a emplear para estas tablas de relación debe comenzar con el prefijo `tn` seguido de `underscoard`, debe ser corto, descriptivo y todo en minúscula.
- Apariencia de los campos: el nombre de los campos se escribe con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.
- Nombre de los campos: todos los campos: los campos identificadores comienzan con el identificador `id` seguido de `underscore` y posteriormente el nombre del campo.
- Aspectos generales: el nombre empleado para las bases de datos, tablas y campos, con sólo leerlos se conoce el propósito de los mismos.

Además se tuvo en cuenta aspectos relacionados con los controles, los cuales se muestran a continuación:

- Apariencia de los controles :

³ **Underscoard:** Carácter ASCII representado como “_”.

- El nombre que se le asigna a los controles comienza con las primeras letras en minúscula, las cuales identifican el tipo de datos al que se refiere. Para los nombres compuestos se emplea la notación CamellCasing.

Conclusiones

En este capítulo se obtuvo el modelo de datos del sistema presentado. Se elaboraron los diagramas de componente y despliegue estrechamente relacionados con el modelo de diseño. De igual manera fue descrita la concepción del tratamiento de errores, la seguridad del sistema y la estrategia de codificación enmarcada en el estándar a utilizar.

CONCLUSIONES

Una vez finalizada la presente investigación, se ha cumplido con el objetivo y las tareas propuestas, por lo que se concluye:

- Los sistemas consultados gestionan la información de las actividades desarrolladas en el área de consulta externa. Sin embargo en su mayoría, no cuentan con hojas de consulta especializadas ni permiten realizar solicitudes asociadas a los servicios de apoyo al diagnóstico. Por otra parte, no generan estadísticas propias del área estudiada y están desarrollados haciendo uso de software propietario.
- El estudio de los procesos del negocio permitió definir las funcionalidades del sistema donde las de mayor relevancia están asociadas a la gestión de las hojas de consulta.
- Se realizó el diseño de la solución en correspondencia con la arquitectura propuesta, obteniéndose un sistema robusto y flexible capaz de gestionar la información relacionada con la atención del paciente en el área de consulta externa.
- Se desarrolló el módulo de Consulta Externa del Sistema de Información Hospitalaria alas HIS, que facilita la gestión de información en esta área de las instituciones hospitalarias.

RECOMENDACIONES

Se recomienda:

- Desarrollar las funcionalidades para gestionar la relación de pacientes que se presentan a la consulta sin cita previa, así como la recodificación del diagnóstico final emitido por el médico.
- Realizar un estudio de la información correspondiente a las hojas de consulta, que permita definir un estándar en su estructura, sin obviar los aspectos propios de las mismas.
- Realizar un levantamiento en instituciones hospitalarias con el objetivo de incluir nuevos formatos de hojas de consulta.

REFERENCIAS BIBLIOGRÁFICAS

1. SIGHO MANUAL TÉCNICO 1.0 ID: FMT-A5-007 Instalador Inicial Guía de Instalación Versión del Módulo 3.0.0 Introducción al SIGHO Sesión II Lic. Aarón García López
2. CNT Sistemas de Información S.A. (s.f.). Recuperado el marzo de 2009, de <http://cnt.com.co/PAGINA/index.asp?id=178>
3. Club ReM. (s.f.). Recuperado el marzo de 2009, de <http://www.e-rem.net/clinic.html>
4. iSoft. (s.f.). Recuperado el febrero de 2009, de <http://www.isoftsanidad.es/corporate/productos/2582.asp>
5. Puente, R. G. (s.f.). *Consulta Médica el software para gestionar Consultas Médicas*. Recuperado el marzo de 2009, de <http://www.galeon.com/rgpuente/consulta.htm>
6. Laura Bermejo Sanz, E. G. *Eclipse como IDE*.
7. *El Servidor de Aplicaciones JBOSS*. (16 de Septiembre de 2004). Recuperado el marzo de 2009, de http://laurel.datsi.fi.upm.es/~ssoo/DAW/Trabajos/2003-2004/Septiembre/19/Contenido_archivos/resource1/r1conten2-1.htm
8. Alejandro. (s.f.). *Blog del programador*. Recuperado el abril de 2009, de <http://blogdeprogramacion.blogspot.com/search/label/Jboss%20Seam>
9. *SeamFramework.org*. (s.f.). Recuperado el abril de 2009, de <http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbosstools.html>
10. Barello, R. M. (s.f.). *Endesa Framework AME*. Recuperado el junio de 2009, de <http://ame.endesa.es/confluence/display/AMEBASE/JSF>
11. Kegan, J. (agosto de 2008). *10 RichFaces Proof of Concept*. Recuperado el mayo de 2009, de <http://www.labor.state.ny.us/cioshares/richFacesPOC.pdf>
12. *Richfaces Developer Guide*. Red Hat. (2008).
13. *12 Katz Max. Practical Richfaces*. Apress. p 12. (2008).
14. *Facelets*. (29 de agosto de 2007). Recuperado el junio de 2009, de <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>
15. ramos, J. A. (s.f.). *14 Adictos al Trabajo*. Recuperado el marzo de 2009, de http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf#6._Conclusi%C3%B3n
16. *Up to Down. Descarga, descubre y comparte*. (s.f.). Recuperado el marzo de 2009, de <http://postgresql.uptodown.com/>
17. Bauer Christian, K. G. (2007). *17 Java Persistence with Hibernate*. Manning Publications. p 67.

18. *Enterprise JavaBeans(EJB)*. (enero de 2005). Recuperado el abril de 2009, de <http://www.proactiva-calidad.com/java/ejb/introduccion.html>
19. *Marco de Desarrollo de la Junta de Andalucía*. (s.f.). Recuperado el mayo de 2009, de <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/JPA>
20. *Visual Paradigm*. (s.f.). Recuperado el marzo de 2009, de <http://www.visual-paradigm.com/product/vpumil/>
21. *Curso de Java*. (s.f.). Recuperado el abril de 2009, de <http://tikal.cifn.unam.mx/~jsegura/LCGII/java2.htm>
22. *Universidad de Palmero*. (s.f.). Recuperado el junio de 2009, de <https://writer.zoho.com/public/faguero/Adm---Ciclo-vida-RUP/fullpage>
23. *NEXOS Software S.A.* (s.f.). Recuperado el junio de 2009, de http://www.nexos-software.com.co/Articulo_22.htm
24. *Milestone Consulting. UML Value Added Trainig Center*. (s.f.). Recuperado el marzo de 2009, de <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>
25. *Patrones GRAPS*. (s.f.). Recuperado el mayo de 2009, de http://sophia.javeriana.edu.co/~lcdiaz/ADOO2006-3/grasp_cpatermostro-lvargas-jviafara.pdf
26. *Bases de Datos. Modelos de Datos*. (n.d.). Retrieved junio 2009, from <http://elies.rediris.es/elies9/4-2.htm>

BIBLIOGRAFÍA

1. Alejandro. (s.f.). *Blog del programador*. Recuperado el abril de 2009, de <http://blogdeprogramacion.blogspot.com/search/label/Jboss%20Seam>
2. Allen, D. *Seam in Action*.
3. Barelo, R. M. (s.f.). *Endesa Framework AME*. Recuperado el junio de 2009, de <http://ame.endesa.es/confluence/display/AMEBASE/JSF>
4. *Bases de Datos. Modelos de Datos*. (s.f.). Recuperado el junio de 2009, de <http://elies.rediris.es/elies9/4-2.htm>
5. Bauer Christian, K. G. (2007). *Java Persistence with Hibernate. Manning Publications*. p 67.
6. *Business Process Modeling Notation*. (2006).
7. Carlos Reynoso, N. K. (2004). *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*.
8. Christian Baver, G. K. *Java Persistence with Hibernate*.
9. *Club ReM*. (s.f.). Recuperado el marzo de 2009, de <http://www.e-rem.net/clinic.html>
10. *Curso de Java*. (s.f.). Recuperado el abril de 2009, de <http://tikal.cifn.unam.mx/~jsegura/LCGII/java2.htm>
11. *CNT Sistemas de Información S.A.* (s.f.). Recuperado el marzo de 2009, de <http://cnt.com.co/PAGINA/index.asp?id=178>
12. *El Servidor de Aplicaciones JBOSS*. (16 de Septiembre de 2004). Recuperado el marzo de 2009, de http://laurel.datsi.fi.upm.es/~ssoo/DAW/Trabajos/2003-2004/Septiembre/19/Contenido_archivos/resource1/r1conten2-1.htm
13. *Enterprise JavaBeans(EJB)*. (enero de 2005). Recuperado el abril de 2009, de <http://www.proactiva-calidad.com/java/ejb/introduccion.html>
14. *Facelets*. (29 de agosto de 2007). Recuperado el junio de 2009, de <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>
15. Healt, i. U. (s.f.). *iSoft*. Obtenido de http://www.isoftware.es/dummy/xHIS_080108.pdf
16. Ivar Jacobson, G. B. *El Proceso Unificado de Desarrollo de Software*.
17. *iSoft*. (s.f.). Recuperado el febrero de 2009, de <http://www.isoftware.es/corporate/productos/2582.asp>

18. *java for live*. (abril de 2008). Recuperado el marzo de 2009, de <http://java4life.blogspot.com/2008/04/richfaces-320.html>
19. *Java en castellano*. (3 de junio de 2005). Recuperado el junio de 2009, de <http://www.programacion.com/java/noticia/1342/>
20. Juan Diego Pérez Jiménez, A. D. (2007). *¿Por qué OMG ha elegido BPMN para modelar de Procesos de*.
21. Katz Max. *Practical Richfaces*. Apress. p 12. (2008).
22. Kegan, J. (agosto de 2008). *RichFaces Proof of Concept*. Recuperado el mayo de 2009, de <http://www.labor.state.ny.us/cioshares/richFacesPOC.pdf>
23. Larman, G. *UML y Patrones*.
24. Laura Bermejo Sanz, E. G. *Eclipse como IDE*.
25. Milestone Consulting. *UML Value Added Trainig Center*. (s.f.). Recuperado el marzo de 2009, de <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>
26. *Marco de Desarrollo de la Junta de Andalucía*. (s.f.). Recuperado el mayo de 2009, de <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/JPA>
27. NEXOS Software S.A. (s.f.). Recuperado el junio de 2009, de http://www.nexos-software.com.co/Articulo_22.htm
28. Oliván, J. A. *Sistemas de informacion hospitalarios*.
29. Ortigosa, A. (2008). *Interfaces de Usuario Avanzadas*.
30. *Patrones GRAPS*. (s.f.). Recuperado el mayo de 2009, de http://sophia.javeriana.edu.co/~lcdiaz/ADOO2006-3/grasp_cpaternostro-lvargas-jviafara.pdf
31. Pressman, R. S. *Ingeniería de Software. Un enfoque práctico*.
32. Puente, R. G. (s.f.). *Consulta Médica el software para gestionar Consultas Médicas*. Recuperado el marzo de 2009, de <http://www.galeon.com/rgpuente/consulta.htm>
33. Ramos, J. A. (9 de Abril de 2007). Recuperado el 1 de Junio de 2009, de <http://www.adictosaltrabajo.com/tutoriales/retornapdf.php?pdf=Ajax4Jsf>
34. Ramos, J. A. (s.f.). *Adictos al Trabajo*. Recuperado el marzo de 2009, de http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf#6._Conclusi%C3%B3n
35. *RichFaces*. (2008).
36. *Richfaces Developer Guide*. Red Hat. (2008).

37. *SeamFramework.org*. (s.f.). Recuperado el abril de 2009, de <http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbossstools.html>
38. Torres, M. L. (s.f.). *Introducción al diseño con patrones*. Recuperado el marzo de 2009, de <http://www.elrincondelprogramador.com/default.asp?id=29&pag=articulos/leer.asp>
39. *Universidad de Palermo*. (s.f.). Recuperado el junio de 2009, de <https://writer.zoho.com/public/faguero/Adm---Ciclo-vida-RUP/fullpage>
40. *Up to Down*. *Descarga, descubre y comparte*. (s.f.). Recuperado el marzo de 2009, de <http://postgresql.uptodown.com/>
41. *Visual Paradigm*. (s.f.). Recuperado el marzo de 2009, de <http://www.visual-paradigm.com/product/vpuml/>

ANEXOS

GLOSARIO DE TÉRMINOS

Constancia

Escrito que se emite cuando el paciente necesita reposo físico o cuando necesita justificar en el centro laboral su ausencia.

Contrarreferencia

Documento que se emite después que el paciente es valorado por el médico y pueda ser remitido para otro lugar fuera del hospital.

Interconsulta

Documento que se emite cuando un paciente de un servicio del hospital necesita ser valorado por un médico de otro servicio del hospital.

Referencia

Documento que se emite cada vez que un paciente necesite ser valorado por un especialista de la misma institución o de otra.