

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Módulo Hospitalización del Sistema de
Información Hospitalaria alas HIS**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores: Arianna Devesa Navarro

Reinel Muñoz Pérez

Tutor: Ing. Karel Gómez Velázquez

Ciudad de La Habana

Junio de 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 26 días del mes de junio del año 2009:

Arianna Devesa Navarro
Autora

Reinel Muñoz Pérez
Autor

Ing. Karel Gómez Velázquez
Tutor

DATOS DE CONTACTO

Ing. Karel Gómez Velázquez (kgomez@uci.cu): Ingeniero en Ciencias Informáticas, graduado en la UCI, en el curso 2006-07. Posee la Categoría Docente de Profesor Instructor. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Actualmente imparte asignaturas de Práctica Profesional. Además, se desempeña como Jefe de Área Temática Gestión Hospitalaria.

*El éxito no se logra sólo con cualidades especiales.
Es sobre todo un trabajo de constancia, de método y de organización.*

J. P. Sergent

AGRADECIMIENTOS

Agradecemos a todos aquellos que contribuyeron a nuestra formación durante todos estos años.

A todo el personal del Hospital Universitario de Caracas, que nos brindó toda la información necesaria para la realización de este Trabajo de Diploma, especialmente a Mercedes, Dora y Magali.

A todos nuestros profesores, los que nos han guiado y enseñado lo mejor de sí, en especial a Maykell Sánchez, pues fue quien nos impulsó en este camino.

A los amigos de siempre, gracias por estar ahí.

DEDICATORIA

De Arianna

A mis padres, que me han apoyado en todo momento y siempre han confiado en mí.

A mi hermano, por ser mi fuente de inspiración y mi ejemplo a seguir.

A mis amigas y amigos de todos los tiempos Odelkis, Yaremis y Elis que estuvieron siempre a mi lado en los buenos y malos momentos. A Joannis, Yojan y Rey por comprenderme y aconsejarme, nunca olvidaré nuestros momentos juntos. A mis amigos de Cienfuegos que siempre los tengo presente Maite, Gabriel, Aymara y Leonardo.

A mis compañeros de grupo y a todas aquellas personas que durante estos años contribuyeron a mi formación.

De Reinel

A mi madre, por apoyarme y guiarme, por traerme a este mundo, por compartir tristezas y alegrías. Todos mis logros son tuyos también.

A mi papá y mis hermanos, que nos vemos poco pero siempre los tengo presente.

A todos mis tíos y tías, en especial a Milaida y Nibita.

A mis abuelos y abuelas.

A mi padrastro y mi hermanastro, por ser mi familia, por aconsejarme en los momentos difíciles.

A los amigos de siempre, Roberto, Adrian, Yuliet, Arianna, Tury y Yojan.

A todo aquel que de una forma u otra ha tenido que ver con mi formación como profesional y como ser humano.

RESUMEN

En el área de hospitalización en las instituciones hospitalarias se gestionan grandes volúmenes de información. Su procesamiento usualmente se realiza de forma manual y en los casos en que se encuentra automatizado, los productos de software existentes son demasiado caros, basados en tecnologías y herramientas propietarias y no brindan una solución estándar que resuelva las necesidades del personal que allí labora. Por estas razones, el objetivo de la investigación es desarrollar el módulo Hospitalización del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

En el desarrollo del sistema se utilizó el Proceso Unificado de Desarrollo y se basa en tecnologías libres, multiplataforma y sobre una arquitectura en capas. Utiliza Java como lenguaje de programación e implementa el patrón de arquitectura Modelo Vista Controlador. Como Sistema de Gestión de Bases de Datos se utiliza PostgreSQL y como servidor de aplicaciones el JBoss Server. Para obtener una interfaz visual moderna y realizar eficientemente las peticiones al servidor, se utilizan las librerías JBoss RichFaces. Para la administración de las reglas y procesos del negocio se utilizan Drools y JBoss jBPM respectivamente.

Entre los beneficios que aporta la aplicación se encuentran proveer una atención de salud con calidad, que garantice la seguridad y confiabilidad de la información médica y su registro en una Historia Clínica Electrónica Única, así como disponer de una herramienta de gestión clínica y administrativa que facilite la toma de decisiones al personal que labora en la hospitalización.

Palabras claves:

Sistema de Información Hospitalaria, Hospitalización, Historia Clínica, Evolución médica, Diagnóstico final, Clasificación Internacional de Enfermedades (CIE)

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 <i>Sistemas de salud pública</i>	7
1.2 <i>Sistemas de Información Hospitalaria (HIS)</i>	8
1.3 <i>Antecedentes</i>	8
1.4 <i>Conceptos relacionados con el campo de acción</i>	12
1.5 <i>Tendencias y tecnologías actuales a considerar</i>	14
1.6 <i>Tecnologías utilizadas en el proceso de desarrollo</i>	18
1.7 <i>Herramientas</i>	25
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	27
2.1 <i>Flujo actual de los procesos involucrados en el campo de acción</i>	27
2.2 <i>Objeto de automatización</i>	29
2.3 <i>Modelo de Negocio</i>	29
2.4 <i>Propuesta del sistema</i>	34
2.5 <i>Modelo de casos de uso del sistema</i>	40
CAPÍTULO 3: DISEÑO DEL SISTEMA	49
3.1 <i>Modelo de Diseño</i>	49
3.2 <i>Diagramas de clases del diseño</i>	53
3.3 <i>Descripción de clases del diseño</i>	60
CAPÍTULO 4: IMPLEMENTACIÓN	66
4.1 <i>Modelo de Datos</i>	66
4.2 <i>Modelo de Implementación</i>	75
4.3 <i>Tratamiento de excepciones</i>	81
4.4 <i>Seguridad</i>	82
4.5 <i>Estrategias de codificación. Estándares y estilos</i>	82
CONCLUSIONES	84
RECOMENDACIONES	85
REFERENCIAS BIBLIOGRÁFICAS	86
BIBLIOGRAFÍA	91
ANEXOS	95
GLOSARIO DE TÉRMINOS	109

INTRODUCCIÓN

Con el auge y desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), a nivel mundial se ha evidenciado en los últimos tiempos, un gran interés por vincularlas con las diferentes esferas y áreas del conocimiento de la sociedad actual. Su uso se ha hecho imprescindible no solo para la obtención de conocimientos, sino también para la creación de instrumentos que permitan gestionar de manera eficiente la información generada por cualquier proceso socio económico o productivo. [1]

Las necesidades de automatización de estas áreas, están basadas en la complejidad de los sistemas de información manipulados, directamente proporcional a la cantidad de datos generados y la obtención de reportes estadísticos asociados, cada vez con mayor nivel de desglose y de especificidad. Entiéndase como un sistema de información al conjunto de instrucciones organizadas, sistematizadas y lógicas que se relacionan entre sí con el fin de obtener y generar información, analizarla y relacionarla para satisfacer las necesidades de las áreas administrativas u operativas de una organización en general.[2]

Las ciencias médicas son una de las esferas cuya proyección actual se ha ido orientando hacia una incorporación progresiva y sistemática de estas tecnologías, como una vía factible para aumentar la calidad de los servicios de salud que son prestados a los pacientes. Se cuenta, de esta forma, con métodos novedosos para la gestión administrativa en consultas, en los hospitales y centros de investigación biomédica. De esta manera se dispone de sistemas automatizados que apoyen al diagnóstico, tratamiento y rehabilitación de los problemas de salud.

La complejidad de las acciones preventivas, curativas y de rehabilitación, así como el grado de especialización de los servicios, la organización de los sistemas de salud responden a los niveles de Atención Primaria, Atención Secundaria u Hospitalaria y Atención Terciaria o Especializada de Salud. Los Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés), son aquellos sistemas de información orientados a satisfacer las necesidades de almacenamiento, procesamiento e interpretación de los datos médico-administrativos generados. Además, constituyen un apoyo para las actividades en los niveles operativos, estratégicos y tácticos de cualquier institución hospitalaria.

Este tipo de sistema permite también, la optimización de los recursos humanos y materiales y minimiza los inconvenientes y morosidades que enfrentan los pacientes en el proceso de obtener las acciones de salud correspondientes. A partir de él se pueden obtener reportes e informes estadísticos, en dependencia del área o servicio que los requieran. Esto da lugar a la retroalimentación en el desempeño de la atención de salud y como consecuencia aumenta o mejora la calidad de vida de los pacientes y de los servicios de salud que se prestan. [3]

Varios países en el mundo han optado por utilizar dichos sistemas. En Europa y Estados Unidos, se pueden encontrar gran número de centros de atención de salud, que brindan sus servicios y gestionan la información mediante los HIS. Asimismo existen empresas o compañías que se dedican a desarrollarlos y comercializarlos a precios excesivamente altos.

En la región latinoamericana la informatización de la salud ha sido pilar fundamental en los objetivos de varios países. Para llevarla a cabo se necesita modernizar y actualizar tecnológicamente las infraestructuras ya existentes. Además se deben construir nuevas instituciones, lo cual lleva implícito el equipamiento de acuerdo a la complejidad de los procedimientos clínicos. Así como, disponibilidad de servicios en el área de influencia y pertinencia tecnológica en relación al grado de complejidad de atención que brindan los hospitales.

Para la explotación de los recursos tecnológicos existentes se necesita disponer de recursos humanos capacitados para su operatividad. A su vez se debe contar con soluciones informáticas que posibiliten la gestión de la información de los procesos asistenciales con la calidad, eficiencia y eficacia que estos demandan y que sean adaptables a cualquier institución de salud de cualquier nivel. Como parte de las estrategias de negocio identificadas por la Facultad - la cual se dedica a la producción de sistemas de salud- uno de los desarrollos estratégicos identificados lo constituye el desarrollo del producto de software alas HIS.

Una de las áreas de atención al paciente en las instituciones hospitalarias lo constituye la Hospitalización, en la cual se ingresa al paciente cuando su condición de salud requiere de atención médica permanente. Durante su estancia en esta área, se lleva a cabo la coordinación entre las distintas especialidades médicas que atienden al paciente a través de las interconsultas. También, se programan las pruebas diagnósticas en los servicios de apoyo y se lleva el registro de cada episodio hospitalario en la Historia Clínica del paciente. [4]

El personal de enfermería en esta área es el encargado de llevar a cabo el cumplimiento de los tratamientos médicos, registrar y controlar sus signos vitales, así como tramitar las órdenes de dieta para su alimentación. También labora en esta área el personal de registros y estadísticas, quienes son los encargados de gestionar la información relacionada con los pacientes recién admitidos y de realizar traslados y egresos en los casos requeridos. Del mismo modo organizan la documentación clínica del paciente y recolectan la información para la elaboración de los censos y estadísticas del movimiento e indicadores hospitalarios.

Actualmente, en cualquier institución hospitalaria que no cuente con un HIS o alguna aplicación informática, no se garantiza todo el almacenamiento de la información generada en la atención al paciente, pues los procesos asociados se realizan de forma manual. La Historia Clínica del paciente, no se encuentra centralizada, impidiendo que se pueda contar con este importante documento médico - legal en cada institución de salud en la que este ha recibido atención.

Como este documento se encuentra en formato duro es susceptible al deterioro provocado por la manipulación de la información que contiene. Así como, a la influencia de las condiciones físicas y ambientales de los locales en donde se almacena. En muchos casos, en estos lugares el espacio no es suficiente o presentan problemas constructivos lo que provoca la destrucción del documento.

El volumen de las historias clínicas se incrementa gradualmente según la cantidad de interconsultas entre las distintas especialidades, informes, órdenes médicas, evoluciones y estudios complementarios que son indicados o realizados por el personal médico. Otro factor que influye en el aumento de estas, es el tiempo de estancia del paciente en el servicio; mientras más prolongado sea, mayor será el volumen de información que se genere.

De esta manera, el proceso de organización y consulta de los distintos episodios de la Historia Clínica del paciente se convierte en un proceso tedioso y difícil de realizar. Además se dificulta enormemente la toma de decisiones, pues no se pueden llevar a cabo en el tiempo requerido, ni de la forma correcta. Los médicos que laboran en el área, en la mayoría de los casos, no pueden consultar diagnósticos realizados anteriormente a los pacientes. Por la carencia de información estadística en tiempo real no se pueden obtener informes con datos estadísticos que permitan hacer estudios y análisis de la población atendida, ni de las enfermedades más comunes con que se presentan los pacientes.

Por todo lo descrito anteriormente, se determina que el **Problema a resolver** es: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Hospitalización de las instituciones hospitalarias?

El **Objeto de estudio** de la presente investigación es el Proceso de gestión de información en las instituciones hospitalarias. A su vez, el **Campo de acción** se enmarca en el Proceso de gestión de información en el área de Hospitalización de las instituciones hospitalarias.

Para resolver el problema identificado se propone el siguiente **Objetivo general**: Desarrollar el módulo Hospitalización 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. Analizar los procesos de negocio asociados al área de Hospitalización de las instituciones hospitalarias.
2. Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria.
3. Asimilar la arquitectura definida por el Área Temática Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Obtener mediante el Proceso Unificado de Desarrollo, los artefactos correspondientes a los flujos de trabajo “Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
5. Implementar el proceso de negocio relacionado con la atención al paciente del módulo de Hospitalización.

El desarrollo del módulo de Hospitalización del Sistema de Información Hospitalaria alas HIS proporcionará un grupo de beneficios. Entre ellos se encuentran:

Para el paciente:

1. Recibir una atención de salud con calidad, que garantice la seguridad y confiabilidad de su información médica.
2. Registro histórico en una Historia Clínica Electrónica Única de todos los procedimientos médicos, pruebas diagnósticas, evoluciones e informes ordenados o realizados como parte de la atención de salud que recibe.
3. Información de salud centralizada, viabilizando su acceso en cualquier centro de salud al que se presente, necesitado de cuidados médicos.
4. Mayor calidad y confiabilidad de los informes que se recibe del personal que labora en el área.

Para el personal que labora en el área:

1. Disponer de un sistema que tenga como centro de referencia al paciente dando respuesta a sus necesidades de salud pública y de asistencia médica. Lo que facilita su acceso a los recursos de salud.
2. Disponer de una herramienta de gestión clínica y gestión administrativa que permite dar respuesta a sus necesidades reales.
3. Posibilidad de revisión de diagnósticos de pacientes atendidos con anterioridad.
4. Contar con servicios informáticos orientados a la organización de la Agenda Médica.

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 Hospitalización del Sistema de Información Hospitalaria. En él se enuncian las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para su desarrollo. Seguidamente el capítulo, **CARACTERÍSTICAS DEL SISTEMA**, contiene un marco conceptual asociado a la información que

será manipulada por el sistema. Se llega 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 brinda la base teórica y conceptual para el desarrollo del Módulo de Hospitalización. Como aspectos esenciales se abordan los principales conceptos relacionados con el dominio del problema y el campo de acción, los antecedentes existentes de sistemas o subsistemas similares. Así como, un análisis de las tecnologías, metodologías y herramientas a considerar en el proceso de desarrollo de software con las que se realiza el proceso de desarrollo.

1.1 Sistemas de salud pública

La forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado, es lo que se conoce como Sistema Nacional de Salud (SNS). La Organización Mundial de la Salud (OMS) lo define como: “Un complejo de elementos interrelacionados que contribuyen a la salud en los hogares, los lugares de trabajo, los lugares públicos y las comunidades, así como el medio ambiente físico y psicosocial en el sector de salud y otros sectores afines. Además es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la salud de una población”. [5]

De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como la especialización de los servicios de salud brindados, los diferentes niveles de atención médica se han organizado en: [6]

1. **Atención Primaria de Salud (APS):** Da solución aproximadamente al 80 % de los problemas de salud de la población y a lo que corresponda con las acciones de promoción y protección de la salud. Aunque sus actividades se realizan en cualquier unidad del SNS, están relacionadas fundamentalmente con las que se realizan en clínicas Urbanas o Rurales, Dispensarios y Postas Médicas.
2. **Atención Médica Secundaria:** Este nivel da cobertura a cerca del 15 % de los problemas de salud, su función fundamental es tratar al hombre ya enfermo, tanto desde el punto de vista individual como colectivo. También desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y

especializadas (Especialidades). Comprende la atención médica brindada en los distintos Hospitales.

3. **Atención Médica Terciaria:** El nivel terciario debe abarcar alrededor del 5 % de los problemas de salud, relacionados con secuelas o aumento de las complicaciones de determinadas dolencias. Se brindan servicios de muy alta complejidad, con la óptima utilización de los recursos y medios existentes y el desarrollo de la investigación. A este nivel pertenecen los Institutos y Hospitales especializados.

1.2 Sistemas de Información Hospitalaria (HIS)

Para la informatización de las acciones de salud correspondientes al Nivel de Atención Médica Secundaria se utilizan los sistemas de información. Un sistema de información es aquel que está constituido por un conjunto de instrucciones organizadas, sistematizadas y lógicas que se relacionan entre sí por medio de un lenguaje informático. Persigue el fin de obtener información, analizarla, relacionarla y generar nueva información para satisfacer las necesidades de las áreas administrativas, operativas de una organización en general.

Dentro de ellos se encuentran los Sistemas de Información Hospitalaria, que están orientados a satisfacer las necesidades de la generación de información mediante la elaboración de reportes, que dependen del área o servicio para el cual se requiera. Esto da lugar a la retroalimentación de la calidad de la atención de los servicios de salud. Un HIS es un sistema medular, porque permite tener una base de datos de tipo demográfico, información de asegurados así como datos clínicos y estadísticos. Se caracteriza por utilizar una identificación numérica para cada uno de los pacientes. Además actúan como sistemas bases para todos los demás sistemas, ya sean clínicos, financieros o administrativos, relacionados con el paciente. [7]

1.3 Antecedentes

Un HIS posee varios módulos, cada uno está orientado a un área específica en las instituciones hospitalarias. Dentro de las distintas áreas que poseen dichas instituciones se encuentra la Hospitalización; en ella el paciente recibe los cuidados médicos y de enfermería durante todo el período de su estancia en el hospital y es donde se gestiona la información relacionada a los

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

movimientos hospitalarios y a su Historia Clínica. Como parte de la investigación sobre los antecedentes a nivel mundial de los HIS que poseen módulo de Hospitalización o gestionan la información relacionada con la Historia Clínica del paciente, se obtuvieron los resultados que se muestran a continuación:

Sistemas de Información Hospitalaria			Tipo de Licencia		Tipo de Aplicación	
	País	Multiplataforma	Libre	Propietario	Web	Escritorio
CNT Pacientes	Colombia	---	---	X	---	X
Comlogik HIMS	Filipinas	---	---	X	---	X
3M HIS	USA	---	---	X	---	X
Microsoft Amalga	USA	---	---	X	X	X
UNITYHIS	España	---	---	X	---	X
XO HMS	India	X	---	X	---	X
Care2X	Alemania	X	X	---	X	---
Siemens Soarian Integrated Care	Alemania	X	---	X	---	X
CGI Sovera HIM	USA	---	---	X	X	---
SIGHO	México	---	---	X	---	X
Galen Hospital	Cuba	---	---	X	---	X

Tabla 1.1 Sistemas de Información Hospitalaria existentes. *Características*

A continuación se presentan las principales características de los HIS que se consideraron más significativos para el objeto de estudio de la investigación.

UNITYHIS

Es uno de los principales productos de la empresa española TCMED, que cuenta con diez años de experiencia en el desarrollo de aplicaciones informáticas para el sector de la salud. UNITYHIS permite interrelacionar las distintas áreas de atención al paciente como una sola organización asistencial. El sistema está diseñado como una aplicación de escritorio, lo que hace más engorroso el proceso de despliegue en las diferentes estaciones de trabajo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Como parte de los módulos que lo conforman se encuentra el de Hospitalización, que brinda la posibilidad de controlar y gestionar los ingresos, traslados y altas. Este se encarga de la solicitud de información al archivo de Historias Clínicas y del registro de episodios hospitalarios. Obtiene información sobre el censo de pacientes y camas, estadísticas de admisiones y elaboración de listados oficiales. También permite obtener una visión global de la Historia Clínica y codificar diagnósticos a través del Codificador Internacional de Enfermedades en su 9na revisión.

HIS CNT Pacientes

Es el principal producto de la empresa CNT Sistemas de Información, establecida en Colombia con veinticinco años de experiencia, especializada para realizar soluciones integradas en el área de la salud. HIS CNT Pacientes está diseñado para integrar todo el ciclo de atención del paciente frente a la prestación de servicios médicos, terapéuticos y diagnósticos.

Tiene como eje principal la Historia Clínica Electrónica, adaptable a todas las especialidades médicas y ambientes de atención como Hospitalización. A través de este módulo, el usuario puede llevar a cabo el censo hospitalario, traslado y liquidación automática de estancias, así como la gestión de los documentos médicos que conforman la Historia Clínica del paciente. Entre ellos se encuentran órdenes médicas, interconsultas, control de signos vitales y evoluciones.

HIMS

Es un producto de la compañía Comlogik Business Systems, desarrollado en una suite de soluciones de software integradas para la administración de la información hospitalaria, de forma escalable y flexible. En 1999 desarrolló la primera versión del HIMS, un HIS para la plataforma Windows. Esta aplicación está basada en una arquitectura Cliente-Servidor.

Dentro de los distintos módulos que conforman este sistema se encuentran los relacionados con la administración del paciente: RADT (Registro, Admisión y Transferencia) y EMR (Historia Clínica Electrónica). Estos permiten al usuario realizar búsquedas de pacientes, llevar el control sobre las camas y las habitaciones, así como la obtención de toda la documentación médica incluida en la Historia Clínica.

Amalga HIS

Es un producto desarrollado por la compañía Microsoft, diseñado para responder a las necesidades del mercado emergente en el sector de la salud. Está construido sobre una arquitectura simple y de código

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

base como cualquier otro sistema conformado por múltiples productos. Responde a las necesidades de algunos de los departamentos del hospital y tiene interfaz de comunicación con otros sistemas, para lo cual utiliza el estándar de comunicación HL7.

Sus principales módulos son Gestión de Medicación que permite al usuario realizar las órdenes médicas, así como llevar el control de su cumplimiento y la Historia Clínica Electrónica. En ella se pueden consultar todos los datos clínicos del estadio del paciente en el hospital, así como su condición y progreso. Su principal desventaja consiste en ser un software propietario y se necesita gran cantidad de recursos para su implantación debido a la tecnología que utiliza, además de ser un producto que no es multiplataforma. No cuenta con módulos como Banco de Sangre, Epidemiología, Bloque quirúrgico y Anatomía Patológica, lo cual reduce su valor agregado en cuanto a la atención de salud del paciente.

Galen Hospital

Galen Hospital es el único HIS desarrollado hasta el momento en Cuba. Es una solución para los centros médicos que brindan servicios de Consulta Externa, Medios Diagnósticos y Hospitalización. Está orientado a la gestión de pacientes como elemento básico de control para mejorar la atención médica. Este sistema informatiza los procesos relacionados con la hospitalización, donde se tienen en cuenta los ingresos, traslados y egresos del paciente y se controla el uso de las camas y los servicios de hospitalización. También cuenta con Control de Historia Clínica, a través del cual se gestiona el movimiento del documento de Historia Clínica desde el momento de su confección, solicitudes de acceso a la información clínica almacenada y los movimientos hospitalarios. [8]

Luego de haber realizado un análisis con profundidad sobre las principales características de los HIS encontrados como parte de los resultados de la investigación realizada, se puede concluir que a pesar de que algunos de estos presentan funcionalidades de relevancia para el campo de acción, cuentan con las desventajas de ser aplicaciones de escritorio. Esto dificulta el proceso de despliegue y eleva su costo, al tener que ser instalado el sistema en cada una de las estaciones de trabajo. No son sistemas portables o multiplataforma, lo que obliga al usuario a depender de la plataforma sobre la cual se ejecuta el sistema. Son demasiado caros, basados en tecnologías y herramientas propietarias y no brindan una solución estándar que se adecúe a las necesidades del personal que allí labora.

1.4 Conceptos relacionados con el campo de acción

Cuando el estado de salud del paciente es tal, que necesita atención médica de manera constante, se procede a su admisión en un servicio de Hospitalización. Una vez ahí, le son realizados todos los procedimientos médicos necesarios para el restablecimiento de su salud. Toda la información que es generada pasa a formar parte de la Historia Clínica, documento único, integrado, acumulativo y legal que surge en el contacto entre el equipo de salud y el paciente. Para lograr que la obtención de la información y la atención al paciente se efectúen con la mayor rapidez y calidad posible, se propone la creación de un sistema de información o gestión hospitalaria que esté orientado a satisfacer las necesidades de generación de información para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria.

Para una mejor comprensión del negocio asociado al problema a solucionar, se relacionan a continuación un grupo de conceptos propios del entorno asociados con el dominio del campo de acción:

Hospitalización

Área del hospital donde el paciente será atendido una vez ingresado y durante su estancia en él. Tanto los médicos del área como enfermeros estarán a cargo de su salud durante el tiempo que se encuentre hospitalizado.

Historia Clínica

Documento que se crea para almacenar el comportamiento evolutivo de un paciente durante su estancia en el hospital. No se limita a ser una narración o exposición de hechos simplemente, sino que incluye juicios, documentos y procedimientos; es un documento que se va haciendo en el tiempo, documentando fundamentalmente la relación médico - paciente. [9]

Médico tratante

Profesional de la salud que se encarga de la atención al paciente. Tiene las facultades para examinar y medicar al paciente, así como consultar a otros especialistas si el caso lo requiere.

Servicio tratante

La hospitalización cuenta con diferentes áreas especializadas llamadas servicios, donde es ubicado el paciente según su patología.

Movimientos hospitalarios

Es el conjunto de ingresos, egresos y traslados asociados a un paciente.

Impresión diagnóstica

Conjunto de diagnósticos que describen la evolución del paciente durante su estancia en el hospital. Puede o no llegar a ser el diagnóstico final.

Diagnóstico final

Es el diagnóstico que emite el médico tratante una vez que se comprueba la enfermedad o conjunto de enfermedades que presenta el paciente.

Interconsulta

Todos los médicos, independientemente de su especialidad necesitan, en ciertos casos, los consejos de otros sobre la atención a determinados pacientes. Esta acción es llamada interconsulta.

Evolución médica

Durante la consulta el médico tratante se encarga de plasmar en el formato de la Evolución, el estado clínico y además todos los análisis complementarios que pueda necesitar el paciente. También registra los signos vitales, la impresión diagnóstica y emite además órdenes médicas y de dieta.

Clasificación Internacional de Enfermedades (CIE)

La Clasificación Internacional de Enfermedades (CIE), es el estándar que utiliza la Organización Mundial de la Salud como referencia para las estadísticas médicas y sanitarias. El CIE brinda una clasificación estándar para las enfermedades, las afecciones y las causas externas de enfermedades y traumatismos, con objeto de recopilar información sanitaria útil relacionada con las defunciones, las enfermedades, la morbilidad y los traumatismos. La clasificación diagnóstica internacional es estándar para todos los fines epidemiológicos generales y muchos fines de gestión sanitaria.

Las categorías establecidas por el CIE son útiles para respaldar la adopción de decisiones, establecer los sistemas de reembolso y disponer de documentación independiente sobre información médica. También se utiliza para clasificar las enfermedades y otros problemas de salud que figuran en muchos tipos de registros sanitarios y demográficos, como los certificados de defunción y los registros de los hospitales. Además de permitir almacenar y consultar información diagnóstica para fines clínicos y epidemiológicos, esos registros sirven de base para la compilación de las estadísticas nacionales de la mortalidad y la morbilidad. [10]

1.5 Tendencias y tecnologías actuales a considerar

En este epígrafe se exponen los elementos asociados al ambiente de desarrollo del sistema y para ello en primer lugar, es preciso definir qué es la Arquitectura de Software, que según Roger Pressman representa, “... *la descripción de los subsistemas o componentes de un sistema informático y las relaciones entre ellos...*” [11]

La Arquitectura de Software es el conjunto de decisiones significativas sobre la organización de un sistema, la selección de los elementos estructurales y sus interfaces de los cuales el sistema está compuesto junto con su comportamiento. Describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. Como un edificio, un sistema de software es una única entidad, pero al arquitecto del software y a los desarrolladores les resulta útil presentar el sistema desde diferentes perspectivas para comprender mejor el diseño. Estas perspectivas son vistas del modelo del sistema y de una manera conjunta representan la arquitectura. Seguidamente se mencionan algunas decisiones importantes sobre la Arquitectura de Software: [12]

Una de las tareas más importantes en el desarrollo de cualquier sistema de información es la definición de su ambiente de desarrollo. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos para la creación de un producto de software. Selecciona y diseña con base a objetivos prefijados para el sistema de información, que pueden ser de tipo funcional, de mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información, teniendo en cuenta las limitaciones derivadas de las tecnologías disponibles para implementarlos. [13]

Al tener en cuenta las características del entorno donde se desplegará el producto a desarrollar y el estado del arte de los HIS en el mundo, se define que las tecnologías a utilizar en el proceso de

desarrollo del sistema deben ser libres, multiplataforma y que faciliten u optimicen el mismo. A continuación se mencionan las definiciones de las tecnologías escogidas, así como sus ventajas.

1.5.1 Sistemas distribuidos. Modelo Cliente - Servidor

Un sistema de cómputo distribuido, es una colección de sistemas de cómputo autónomos, llamados nodos. Estos están interconectados a través de una red y de software de comunicaciones, capaces de cooperar para la realización de una tarea común; mientras que la computación distribuida es el conjunto de modelos, técnicas y herramientas computacionales. Las mismas ayudan a resolver los problemas planteados por los sistemas distribuidos en cuanto a la generación, el procesamiento y la utilización de la información requerida para su funcionamiento. [14]

Modelo Cliente - Servidor

Desde el punto de vista funcional, se puede definir el Modelo Cliente - Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. La arquitectura Cliente - Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento. Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente, lo que en computación distribuida afecta directamente el tráfico de la red, reduciéndolo grandemente. [15]

1.5.2 Patrones de arquitectura y diseño

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Describen interacciones amplias de elementos abstractos de diseño que permiten al arquitecto o diseñador pensar en un problema complejo mediante una abreviatura intuitiva.

El logro del mejoramiento y la calidad del diseño de un sistema, depende en gran medida de la utilización de patrones que proporcionen esquemas para refinar subsistemas o componentes de un sistema. Estos patrones son conocidos como patrones de diseño. Un patrón de diseño nombra, abstrae e identifica los aspectos claves de una estructura de diseño común, que lo hace útil para la creación de un diseño orientado a objetos reutilizables. Un patrón de diseño es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, una manera más práctica de describir la organización de un sistema.

Son los patrones los que permitirán hablar posteriormente de la arquitectura, en términos de abarcar también las decisiones más importantes en relación con el sistema a desarrollar. Los patrones arquitectónicos ofrecen un conjunto de subsistemas predefinidos, en los que se especifica sus responsabilidades y se incluyen reglas para organizar las relaciones entre ellos. [16]

Arquitectura en tres capas

En la arquitectura en capas, una capa representa una separación lógica del software, una separación básica de preocupaciones en el nivel del desarrollador, de modo que se pueden dividir más fácilmente las responsabilidades con respecto al sistema. La arquitectura en tres capas define como su propio nombre lo indica, tres elementos esenciales: la capa de presentación, que reúne todos los aspectos del software que tienen que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos y que típicamente incluyen el manejo y la apariencia de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general; la capa de negocio, que posee todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios, incluyendo las tareas que forman parte de los procesos, las reglas y restricciones que aplican y la capa de datos, que reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de las Bases de Datos. [17]

Modelo Vista Controlador (MVC)

Este patrón permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Esto proporciona múltiples vistas sobre un mismo modelo de datos. Este patrón se usa frecuentemente en aplicaciones web donde se utilicen diferentes interfaces de usuario y el código que provee los datos a la página es dinámico. Los tres elementos esenciales de este patrón son los siguientes: [18]

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Modelo:** Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio para cambiar el estado de estos datos, habitualmente desde el controlador.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Tanto la vista como el controlador dependen del modelo, el cual no depende de otros conceptos o clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. En aplicaciones web por ejemplo, la separación entre la vista (navegador) y el controlador que constituye los componentes del lado del servidor que manejan los requerimientos a través del Protocolo para la Transferencia de Hipertexto (HTTP) está muy claramente definida. Los requerimientos no funcionales de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas del negocio. Los clientes pueden preferir distintas opciones de representación pero dado que el modelo no depende de la vista, agregar nuevas opciones o modificar las ya existentes generalmente no afecta al modelo.

Además permite definir cómo organizar el modelo de diseño en capas, que pueden estar físicas y/o lógicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. De igual modo es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Entre las ventajas del patrón MVC están las siguientes:

- **Soporte de múltiples vistas:** Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.
- **Adaptación al cambio:** Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización. [19]

La arquitectura en tres capas y el patrón MVC, pueden relacionarse lógicamente mediante cada uno de sus elementos, donde la capa de presentación podría corresponderse con la Vista, la capa de negocio con el Controlador y la capa de datos con el Modelo.

1.6 Tecnologías utilizadas en el proceso de desarrollo

A continuación se realiza una descripción de las tecnologías utilizadas en el proceso de desarrollo agrupadas cada una en las capas donde son utilizadas.

1.6.1 Capa de Presentación

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, la especificación de la navegación del usuario.

RichFaces

JBoss 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, administración avanzada de recursos como imágenes, código JavaScript 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 grandemente la experiencia de usuario. [20] [21]

Seam UI

Serie de controles JSF altamente integrables con JBoss Seam. Están dirigidos a complementar los controles JSF incorporados, y los controles de otras bibliotecas externas.

Facelets

Facelets es un framework simplificado de presentación, que permite diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al desarrollador, pues permite definir vistas JSF utilizando plantillas del tipo HTML, reduciendo el código innecesario para agregar componentes en la vista y que no necesariamente sea un contenedor web.

Lenguaje de Marcado de Hipertexto Extensible (XHTML)

XHTML es una versión más estricta y limpia del Lenguaje de Marcado de Hipertexto (HTML), que nace precisamente con el objetivo de reemplazarlo ante su limitación de uso con las cada vez más abundantes herramientas basadas en el Lenguaje de Marcado Extensible (XML), un formato de datos universal altamente extensible, que define una manera estándar de estructurar el marcado de documentos. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. Es un lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella.

Extended EL

Extended EL provee una extensión al estándar Lenguaje de Expresión Unificado (EL) aumentando su expresividad y su poder. Este estándar adopta las características ofrecidas por el lenguaje de expresión de JSF. Mediante su utilización se puede reducir drásticamente la cantidad de código en las páginas creadas, aumentando la productividad, haciendo más fácil el mantenimiento y más pequeña la curva de aprendizaje en términos del desarrollo.

1.6.2 Capa de Negocio

JBoss Seam

JBoss Seam 2.0 (Seam) es un framework para el desarrollo de aplicaciones web en Java, que define un modelo de componentes uniforme para toda la lógica de negocio de las aplicaciones que sean desarrolladas mediante su utilización. Revitaliza el estándar Java EE poniéndole fin a sus divergencias y unificando sus componentes, haciéndolos más accesibles y funcionales. Integra fácilmente tecnologías estándares como Java Server Faces (JSF), modelo de componentes para la capa de presentación; Enterprise JavaBeans (EJB3), modelo de componentes para la lógica de negocio y persistencia del lado del servidor; Java Persistence API (JPA), y de Business Process Management (BPM). Integra además librerías de controles de código abierto basadas en JSF como RichFaces e ICEFaces. [22][23][24]

Realiza nuevos aportes en el campo de la administración de estado. Mientras que en los frameworks tradicionales todo el estado es administrado básicamente en la sesión HTTP, Seam provee una mayor granularidad de contextos de estado y aporta un nuevo concepto, la administración de espacios de trabajo. Esta permite al usuario tener en varias pestañas o ventanas del navegador actividades del negocio con contextos completamente aislados. [25]

JBoss jBPM

JBoss jBPM posibilita la automatización a un conjunto más amplio de problemas de negocio, desde flujos de trabajo embebidos hasta la orquestación de procesos de negocio empresariales. Es un motor de procesos de negocio flexible, escalable y modular. Se puede integrar fácilmente con Seam y provee un modelo de programación orientada a procesos que mezcla lo mejor de Java y las técnicas de programación declarativa.

Drools

Drools es un sistema de administración de reglas de negocio que se vale de la programación declarativa para decir “qué hacer” y no “cómo hacerlo”. Reduce la brecha entre el negocio y las Tecnologías de la Información y las Comunicaciones (TIC), aumentando considerablemente el mantenimiento y la calidad del código escrito por los desarrolladores. Su principal beneficio es que permite separar la lógica de negocio del código de la aplicación, de esta manera un cambio en el flujo del negocio no implica la recompilación ni la aplicación de modificaciones al sistema.

1.6.3 Capa de Datos

Java Persistence API (JPA)

Es la Interfaz de Programación de Aplicaciones (API) estándar para la persistencia y el mapeo objeto/relacional para la plataforma Java EE, y permite utilizar un modelo de dominio Java para administrar bases de datos relacionales. [26]

Enterprise JavaBeans (EJB3)

Arquitectura componente del lado del servidor para la plataforma Java. Permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos. Incorpora el estándar JPA como el principal API de persistencia para aplicaciones EJB3.

Hibernate

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 (bastante 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. [27]

1.6.4 Tecnologías y metodologías horizontales

PostgreSQL

Es un Sistema Gestor de Base de Datos (SGBD, por sus siglas en inglés) relacional de código abierto, muy poderoso, con una arquitectura probada. Puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día. Posee protección de transacciones u operaciones de Atomicidad, Consistencia, Aislamiento, Durabilidad (ACID, por sus siglas en inglés). Es un gestor de base de datos empresarial, que posee características sofisticadas como Control de Concurrencia Multi - Versión (MVCC, por sus siglas en inglés), replicación asíncrona, transacciones anidadas, realización de respaldo de datos en línea, optimizador o planificador de consultas, soporta internacionalización. Es altamente escalable en cuanto a la cantidad de información que puede manejar y al número de usuarios concurrentes que puede alojar. [28]

El tamaño máximo para las bases de datos PostgreSQL es ilimitado. Es totalmente compatible con el estándar ANSI-SQL 92/99. Permite realizar relaciones de herencia entre tablas, definir sistemas de reglas y eventos. PostgreSQL ejecuta procedimientos almacenados en más de una docena de lenguajes de programación tales como Java, Perl, Python, Ruby, Tcl, C/C++, y su propio lenguaje PL/pgSQL. De esto se deriva que existan muchas librerías para realizar la conexión a PostgreSQL desde varios lenguajes, compilados o interpretados, entre ellos ODBC, Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme, para nombrar algunos. Desde el año 1999 hasta la fecha ha ganado un gran número de premios que ratifican a este SGBD como el más completo dentro del mundo del código abierto. [29]

Java

Lenguaje de programación desarrollado por la compañía Sun Microsystems, que utiliza el paradigma de la Programación Orientada a Objetos (POO). Es un lenguaje robusto pues no permite el manejo directo del hardware ni de la memoria. Dentro de sus principales ventajas se encuentra la de ser multiplataforma. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. [30]

JBoss Server

Servidor de aplicaciones Java más utilizado actualmente en el mercado. Cientos de profesionales y desarrolladores de código abierto han contribuido a su creación y desarrollo. Es una plataforma certificada que cumple con las especificaciones de J2EE para el despliegue y desarrollo de aplicaciones empresariales Java, aplicaciones web, y portales. Provee servicios extendidos almacenamiento de datos en memoria y de manera persistente. Permite la integración de todas las tecnologías y herramientas utilizadas por JBoss Seam. Es actualizado e integrado constantemente con lo último del estado del arte de las aplicaciones web. [31][32]

Proceso Unificado de Desarrollo (RUP)

Es un proceso de desarrollo de software que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado. Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
 - Desarrollo iterativo
 - Administración de requisitos
 - Uso de arquitectura basada en componentes
 - Control de cambios
 - Modelado visual del software
 - Verificación de la calidad del software

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos, que son los productos tangibles obtenidos mediante su aplicación y trabajadores, personas que juegan un rol determinado para la obtención de estos artefactos.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al culminar de cada ciclo, estos se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante: [33]

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances 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 de Modelado (UML)

Lenguaje utilizado para especificar, visualizar y documentar los componentes de un sistema en desarrollo orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el Grupo de Administración de Objetos (OMG, por sus siglas en inglés). UML ofrece una variante para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos, componentes de software reutilizables.

RUP define los componentes que se utilizarán para construir el sistema y las interfaces que los conectarán, mediante la utilización de una combinación del desarrollo iterativo e incremental. Define además la función del sistema aplicando un enfoque basado en escenarios, es decir desde el punto de vista del usuario. Entonces acopla la función con un marco de trabajo arquitectónico que identifica la forma que tomará el software.

Algunas de las propiedades de UML como lenguaje de modelado estándar son: [34][35]

- **Concurrencia,** es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.

- Ampliamente utilizado por la industria desde su adopción por OMG.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

Notación para el Modelado de Procesos de Negocio (BPMN)

Es un nuevo estándar para modelar flujos de procesos de negocio y servicios web. Su meta principal es definir una notación entendible para todos los usuarios del negocio - desde los analistas del negocio que modelan los procesos hasta los desarrolladores técnicos responsables de implementarlos y finalmente a los usuarios del negocio- quienes monitorearán y administrarán dichos procesos. BPMN crea un puente estándar entre el diseño del proceso de negocio y su implementación.

BPMN es un facilitador de fondo para una nueva iniciativa en el mundo de la arquitectura empresarial, la Administración de Procesos de Negocio (BPM, por sus siglas en inglés). Esta permite gestionar el cambio para mejorar los procesos de negocio y unifica las disciplinas de Modelación de Procesos, Simulación, Flujo de trabajo, Integración de Aplicaciones Empresariales (EAI, por sus siglas en inglés) y la integración Business to Business (B2B) en un único estándar. [36] [37]

1.7 Herramientas

Con los elementos expuestos anteriormente se pueden definir las herramientas que conforman el Ambiente de Desarrollo del Módulo de Hospitalización para el Sistema de Información Hospitalaria alas HIS.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En primer lugar se utilizará el Visual Paradigm 6.0 como herramienta CASE (Computer - Aided Software Engineering), esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Utilizando la notación BPMN para generar la documentación del negocio y el lenguaje UML 2.0 para crear los diagramas de casos de uso del flujo de requerimientos, así como los correspondientes a los de los flujos de diseño e implementación. Para administrar la base de datos pgAdmin III, debido a que es un ambiente gratuito, relacional, muy poderoso y puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día.

Finalmente se escoge el Eclipse, un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) que cuenta con la posibilidad de añadir nuevas funcionalidades al editor, a través de nuevos módulos, lo que es beneficioso para la utilización de JBoss Tools, conjunto de herramientas que posibilitan el desarrollo desde el IDE Eclipse con RichFaces, Seam e Hibernate, además de realizar la administración y configuración del servidor JBoss AS.

En este capítulo se valoraron los principales conceptos relacionados con los procesos que se desarrollan en el área de hospitalización, lo que permitió una mejor comprensión de los mismos. Se realizó un estudio sobre los principales sistemas de información hospitalaria existentes a nivel nacional e internacional y se determinó que estos no brindan una solución estándar para la realización de los procesos que allí se ejecutan. Además, se describieron las tecnologías escogidas para el desarrollo del producto y las ventajas de cada una de ellas.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El Sistema Nacional de Salud (SNS) incluye, dentro de sus niveles de atención, como parte de la atención secundaria, a los hospitales. Dentro de ellos, el área de Hospitalización se encarga de la atención directa al paciente admitido, lo que genera gran cantidad de información. Esta pasa a formar parte de la Historia Clínica, documento médico legal de donde se obtienen gran parte de las estadísticas y toda la información médica durante la estadía del paciente en el hospital. Es por esto que en el presente capítulo se propone la elaboración de un sistema que informatice la gran mayoría de estos registros y se documentan los artefactos de desarrollo que son elaborados como resultado de esta tarea.

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

En un hospital se llevan a cabo numerosos procesos, dentro de los cuales se encuentran la gestión de los movimientos hospitalarios y la atención al paciente, siendo este último uno de los más importantes a realizar en la hospitalización. Los procesos mencionados involucran una serie de documentos necesarios para cada paciente en específico. La atención al paciente es iniciada una vez que este es admitido, momento en el cual se le realiza por parte del médico un examen físico, del cual se obtiene una impresión diagnóstica y su estado clínico, los que son registrados en la Historia Clínica. En caso de que el ingreso sea por el área de emergencia, el médico es el encargado de firmar la orden de admisión.

Es deber diario del médico registrar la evolución del paciente, dejando plasmado en la Historia Clínica cómo ha progresado el paciente entre una consulta y otra, además de consultar los resultados obtenidos de los exámenes complementarios que se le han realizado. Luego de cumplimentar esta actividad, el médico puede realizar las indicaciones médicas lo que cuenta con los medicamentos y procedimientos que se deben aplicársele al paciente para su recuperación.

Pueden ser realizadas también solicitudes de interconsulta, las que están dirigidas a especialistas de otros servicios médicos con el objetivo de evaluar el caso para contar con otro criterio especializado. Como respuesta a la misma, el médico interconsultante realiza una evolución al paciente donde deja reflejada su opinión sobre el caso que se está valorando. Los servicios de apoyo juegan un papel importante en la atención al paciente, pues brindan los medios de diagnóstico para apoyar la

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

determinación y corroboración del diagnóstico. Entre las distintas solicitudes que el médico puede realizar a estos servicios, se encuentran los exámenes de laboratorio, tipiaje, transfusión, intervención quirúrgica, citología, biopsia, autopsia y citas para la realización de estudios radiológicos e imagenológicos.

La gestión de los movimientos hospitalarios en la Hospitalización, se evidencian con la realización de la transferencia y el egreso, los que son ordenados por el médico y procesados por Técnicos de Estadísticas de Salud en el servicio. La transferencia, es llevada a cabo cuando el paciente presenta una patología propia de otro servicio y la disponibilidad de camas de este permite el movimiento. Por otra parte el egreso se realiza cuando el paciente ha mejorado su estado de salud considerablemente y no requiere observación periódica, cuando ha fallecido o contra indicación médica. En todos los casos el médico debe realizar un resumen de la estancia del paciente en el servicio de hospitalización, con aquellos aspectos que considera son significativos, momento en el que además muestra el diagnóstico clínico final.

Cuando el egreso realizado es por defunción, el médico debe confeccionar el Certificado de Defunción correspondiente, en el cual debe especificarse la necesidad o no de la realización de autopsia pudiendo ser estas clínicas o forenses. Esta última se hace necesaria en aquellos casos en que el paciente está involucrado en un caso médico legal y no es necesario contar con el consentimiento de los familiares para su realización. Por el contrario, para la realización de una clínica, se precisa del consentimiento de los familiares para que pueda efectuarse.

Normalmente estos procesos son efectuados de forma manual, afectando de gran manera la administración de los hospitales, pues como no se puede tener una información actualizada de todos los insumos necesarios se gastan más recursos. En el área de hospitalización se genera una gran cantidad de información, lo que hace que la Historia Clínica alcance un gran volumen, dificultando en cierta medida su manipulación. Así como también la búsqueda, de forma rápida y eficiente de información clínica de pacientes necesaria para la toma de decisiones y obtención de estadísticas de salud.

2.2 Objeto de automatización

Como resultado de estos procesos se generan un grupo de documentos que formarán parte de la Historia Clínica del paciente. Los que se presentan y explican a continuación han sido escogidos para su automatización por la importancia que estos tienen en el funcionamiento básico del área de hospitalización.

Se desean automatizar todos los procesos que se encuentran enmarcados en la atención al paciente. Entre estos el proceso de creación de las hojas de hospitalización, que corresponden a las hojas de cada especialidad médica, la información referente a la evolución médica que se le realiza a diario al paciente, la realización de las impresiones diagnósticas y diagnósticos finales utilizando para ello la Clasificación Internacional de Enfermedades (CIE) en su décima edición; así como la medición y gráfica de los signos vitales y la conformación de las órdenes médicas. Se pretende automatizar además el conjunto de solicitudes a los servicios de apoyo, siendo estas solicitudes de interconsulta, transfusión, tipiaje, biopsia, citologías, autopsia, análisis a laboratorios, intervención quirúrgica y cita para estudios radiológicos e imagenológicos.

A raíz de la identificación de los documentos que se manejan en los procesos de la Hospitalización surge la necesidad de la creación de un sistema para automatizar estos procesos a partir de una aplicación a fin con los conocimientos del personal de esta área. Para esto el software debe ser entendible, de fácil manejo y con una interfaz gráfica amigable.

2.3 Modelo de Negocio

Dentro del ciclo de desarrollo de software propuesto por RUP, el primer flujo de trabajo es el de modelado de negocio. Este tiene mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Es en este flujo de trabajo donde se conocen a fondo cómo son iniciadas, ejecutadas y terminadas cada una de las actividades de un proceso determinado, teniéndose en cuenta claramente la precedencia que se establece entre estas, obteniéndose a través del modelado de estos procesos una visión más amplia del negocio existente. [38]

2.3.1 Actores del negocio

Un actor del negocio es el principal beneficiado de los procesos existentes, es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [39]

ACTOR	DESCRIPCIÓN
Paciente	Principal beneficiado de los procesos de negocio, pues recibe toda la atención clínica en el área de hospitalización.

2.3.2 Trabajadores del negocio

Un trabajador del negocio es la abstracción de una persona o sistema software que representa un rol desempeñado en las realizaciones de las actividades identificadas en un proceso de negocio. Un trabajador del negocio colabora con otros trabajadores, es notificado de los eventos y manipula las entidades para realizar sus responsabilidades. [40]

TRABAJADOR	DESCRIPCIÓN
Médico	Principal protagonista de los procesos de negocio en hospitalización, es el encargado de brindarle la atención al paciente y emitir y actualizar la mayoría de los documentos clínicos que se generan en el área.

2.3.3 Diagrama de Procesos de Negocio

BPMN define diagramas de procesos de negocios basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización. Se compone de un conjunto de elementos gráficos que facilitan un diagrama entendible tanto por audiencias de negocios

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

como técnicas. BPMN define los objetos de flujo, objetos conectores, artefactos y swimlanes como las cuatro categorías básicas de estos elementos.

Dentro de los objetos de flujo se encuentran las actividades, las cuales representan las acciones que se realizan dentro de un proceso. Las mismas pueden ser atómicas o compuestas, presentándose en dos tipos, tareas y subprocesos. Un proceso es una actividad que se realiza dentro de una compañía. El cual puede contener subprocesos, que tienen dentro las agrupaciones de flujos de actividades.

Las tareas son las actividades atómicas que no pueden ser descompuestas en otras. Las decisiones (Gateway) se usan para controlar la divergencia o convergencia de la secuencia de flujo. Así, esto determina las tradicionales decisiones, así como la creación de nuevos caminos, la fusión de estos o la unión. Por último en este grupo se encuentran los eventos, siendo estos sucesos que ocurren durante el curso del proceso de negocio. Estos eventos afectan al flujo del proceso y suelen tener una causa o un impacto en él.

Los objetos de flujo se conectan entre ellos en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. Hay tres objetos conectores que hacen esta función. Estos conectores son asociaciones, flujos de secuencia y de mensaje. Una asociación se usa para asociar datos, texto, y otros artefactos con los objetos de flujo. Las asociaciones se usan para mostrar entradas y salidas de las actividades. El flujo de mensaje es utilizado para mostrar el flujo de mensajes entre dos participantes del proceso separados y flujo de secuencia muestra el orden en el que las diferentes actividades se ejecutarán en el Proceso.

Los swimlanes son un mecanismo para organizar actividades en categorías separadas visualmente para ilustrar diferentes capacidades funcionales o responsabilidades. BPMN soporta los swimlanes con dos constructores principales. Estos tipos de objetos son llamados calles (pool) y pistas (lane), el primero se caracteriza por ser calles independientes y el segundo son sub particiones de las mismas. Por otra parte se incluyen los objetos que se generan como parte de las actividades, los cuales a su vez sirven de entrada para la realización de las mismas. A continuación se representa el diagrama de procesos de negocio.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

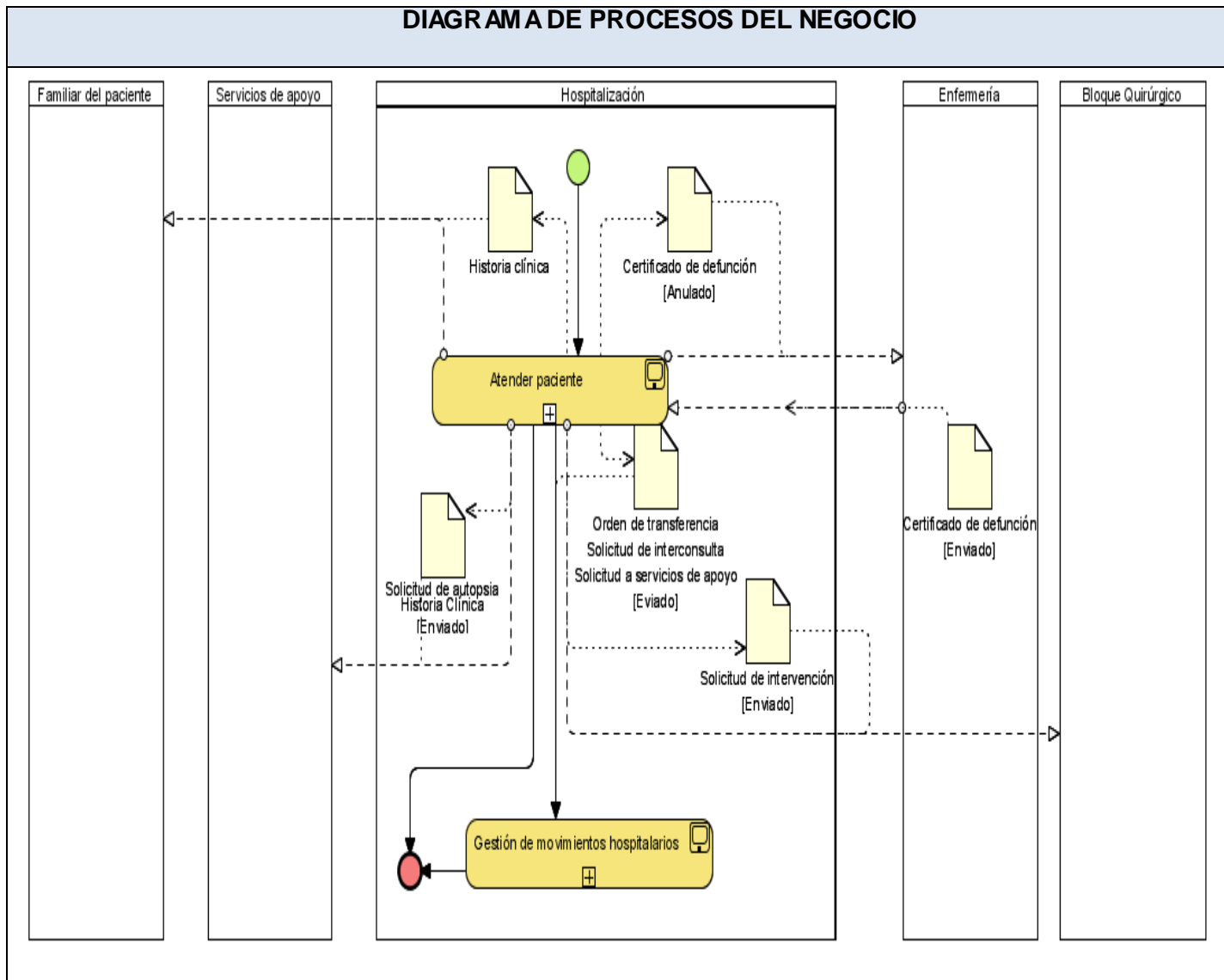


Figura 2.1 Diagrama global de procesos del negocio.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.3.3.1 Atender Paciente

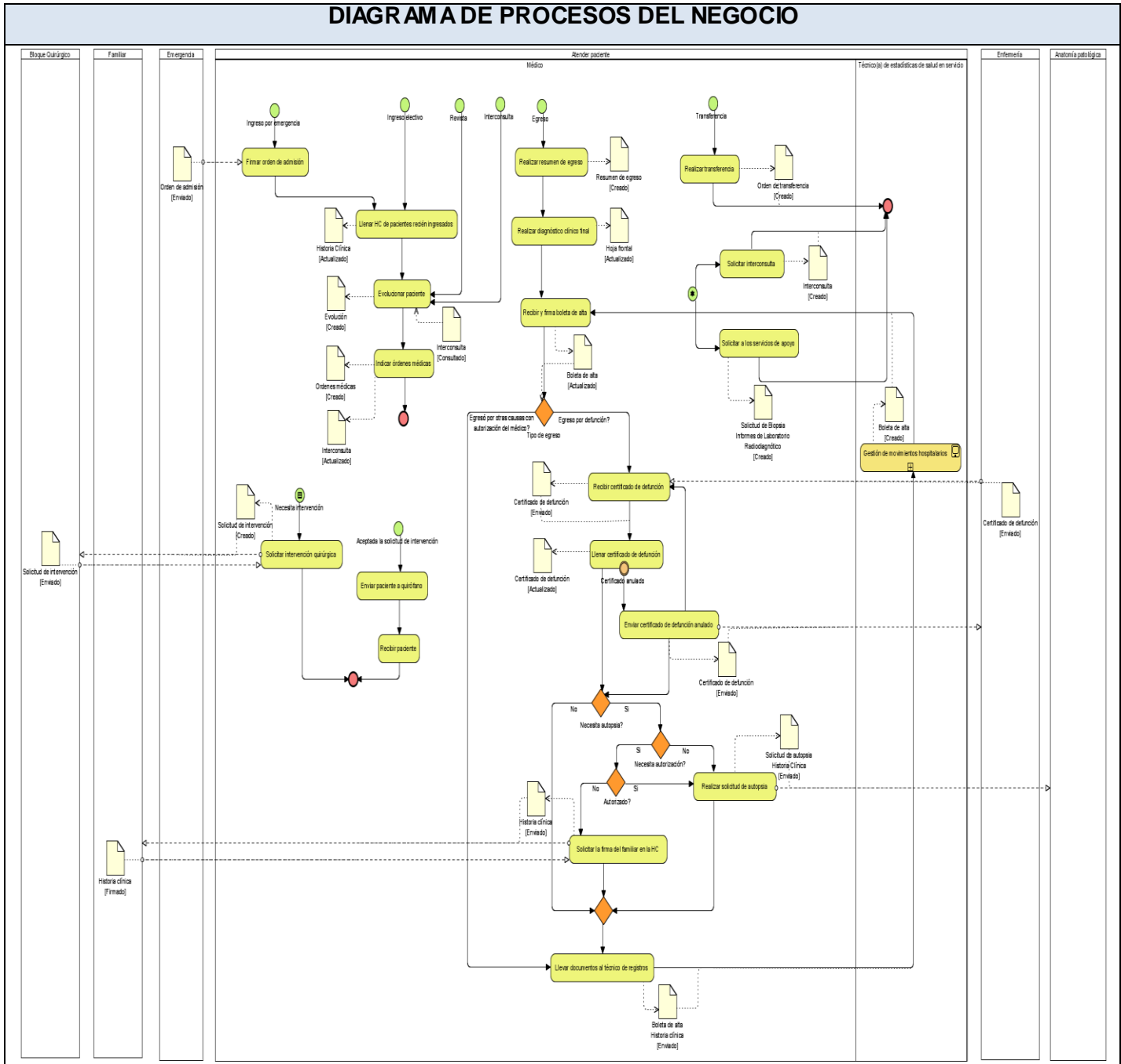


Figura 2.2 Diagrama de proceso del negocio *Atender paciente*

2.4 Propuesta del sistema

RUP propone, dentro de sus disciplinas, que a continuación de la elaboración del Modelado de Negocio, continúe el ciclo de desarrollo con el flujo de trabajo Gestión de Requerimientos, que se encarga de extraer los requisitos que debe tener el sistema a partir de los cuales se construye el Diagrama de Casos de Uso del Sistema y se identifican aquellos que son arquitectónicamente significativos.

2.4.1 Especificación de requerimientos de software

La especificación de los requerimientos de software constituye un elemento de vital importancia para la elaboración de un software de calidad superior. El Glosario Estándar de Terminologías de Ingeniería de Software de la IEEE define los requerimientos de software como condiciones o capacidades que deben estar presentes en un sistema o componentes de este, para satisfacer un contrato, estándar, especificación u otro documento formal. Es necesario hacer énfasis en la precisión con que se debe realizar esta tarea por cumplir un papel primordial en el proceso de producción de software, pues se enfoca en un área fundamental: la definición de lo que se desea producir, mediante una descripción más clara del comportamiento del sistema, minimizando los problemas derivados de su desarrollo.

2.4.2 Requerimientos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales: [41]

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

REQUERIMIENTOS FUNCIONALES	
RF1 - Listar pacientes	RF53 - Buscar guardia médica
RF2 - Crear hoja general de hospitalización	RF54 - Crear orden médica
RF3 - Ver hoja general de hospitalización	RF55 - Ver orden médica
RF4 - Crear hoja neurológica de hospitalización	RF56 - Modificar orden médica
RF5 - Ver hoja neurológica de hospitalización	RF57 - Eliminar orden médica
RF6 - Crear hoja ORL de hospitalización	RF58 - Buscar orden médica
RF7 - Ver hoja ORL de hospitalización	RF59 - Crear solicitud de interconsulta
RF8 - Crear hoja dermatológica de hospitalización	RF60 - Ver solicitud de interconsulta
RF9 - Ver hoja dermatológica de hospitalización	RF61 - Modificar solicitud de interconsulta
RF10 - Crear hoja ginecológica de hospitalización	RF62 - Eliminar solicitud de interconsulta
RF11 - Ver hoja ginecológica de hospitalización	RF63 - Buscar solicitudes de interconsulta recibidas
RF12 - Crear hoja oftalmológica de hospitalización	RF64 - Aceptar solicitud de interconsulta
RF13 - Ver hoja oftalmológica de hospitalización	RF65 - Crear hoja de interconsulta
RF14 - Crear hoja uro-genital de hospitalización	RF66 - Ver hoja de interconsulta
RF15 - Ver hoja uro-genital de hospitalización	RF67 - Buscar hoja de interconsulta
RF16 - Crear hoja de traumatología y ortopedia	RF68 - Seleccionar solicitudes de interconsulta aceptadas
RF17 - Ver hoja de traumatología y ortopedia	RF69 - Crear solicitud de análisis de laboratorio para pacientes internos
RF18 - Crear hoja obstétrica de hospitalización	RF70 - Buscar solicitudes con resultados aprobados
RF19 - Ver hoja obstétrica de hospitalización	RF71 - Ver datos de solicitud de análisis de laboratorio
RF20 - Crear hoja reumatológica de hospitalización	RF72 - Consultar solicitud de análisis de laboratorio
RF21 - Ver hoja reumatológica de hospitalización	RF73 - Eliminar solicitud de análisis de laboratorio
RF22 - Crear hoja de gastroenterología de hospitalización	RF74 - Crear solicitud de transfusión
RF23 - Ver hoja de gastroenterología de hospitalización	RF75 - Registrar datos de la transfusión
RF24 - Crear hoja de radioterapia y medicina nuclear de hospitalización	RF76 - Mostrar donaciones por paciente
RF25 - Ver hoja de radioterapia y medicina nuclear de hospitalización	RF77 - Ver datos de solicitud de transfusión
RF26 - Crear hoja hemato-oncológica de	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

hospitalización	RF78 - Eliminar datos de solicitud de transfusión
RF27 - Ver hoja hemato-oncológica de hospitalización	RF79 - Modificar datos de solicitud de transfusión
RF28 - Crear hoja hemato-oncológica pediátrica de hospitalización	RF80 - Crear solicitud de examen de tipiaje
RF29 - Ver hoja hemato-oncológica pediátrica de hospitalización	RF81 - Mostrar resultados de examen de tipiaje
RF30 - Crear hoja de cirugía colo-rectal de hospitalización	RF82 - Ver datos de solicitud de examen de tipiaje
RF31 - Ver hoja de cirugía colo-rectal de hospitalización	RF83 - Eliminar datos de solicitud de examen de tipiaje
RF32 - Buscar hoja de hospitalización	RF84 - Modificar datos de solicitud de examen de tipiaje
RF33 - Consultar acciones realizadas	RF85 - Asignar cita para estudio radiológico e imagenológico
RF34 - Consultar acciones realizadas hasta el momento	RF86 - Ver datos de cita para estudio radiológico e imagenológico
RF35 - Crear evolución médica	RF87 - Modificar datos de cita para estudio radiológico e imagenológico
RF36 - Ver evolución médica	RF88 - Eliminar cita para estudio radiológico e imagenológico
RF37 - Buscar evolución médica	RF89 - Seleccionar enfermedad
RF38 - Crear signos vitales	RF90 - Modificar antecedentes personales
RF39 - Modificar signos vitales	RF91 - Modificar antecedentes familiares
RF40 - Eliminar signos vitales	RF92 - Modificar hábitos psicobiológicos
RF41 - Ver signos vitales	RF93 - Crear solicitud de citología ginecológica
RF42 - Graficar signos vitales	RF94 - Crear solicitud de citología
RF43 - Buscar signos vitales	RF95 - Crear solicitud de biopsia
RF44 - Crear procedimientos médicos pendientes	RF96 - Ver datos de solicitud de análisis
RF45 - Ver procedimientos médicos pendientes	RF97 - Eliminar solicitud de análisis
RF46 - Modificar procedimientos médicos pendientes	RF98 - Modificar solicitud de análisis
RF47 - Eliminar procedimientos médicos pendientes	RF99 - Buscar solicitud de análisis
RF48 - Buscar procedimientos médicos pendientes	RF100 - Buscar informe final
RF49 - Crear guardia médica	RF101 - Realizar solicitud de intervención
	RF102 - Ver datos de solicitud de intervención quirúrgica
	RF103 - Modificar datos de solicitud de

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

RF50 - Ver guardia médica RF51 - Modificar guardia médica RF52 - Eliminar guardia médica	intervención quirúrgica RF104 - Eliminar datos de solicitud de intervención quirúrgica RF105 - Consultar Plan quirúrgico
---	---

2.4.3 Requerimientos no funcionales

Los requerimientos no funcionales son cualidades o propiedades que el producto debe tener, es decir, restricciones en el producto que está siendo desarrollado. Estos no describen lo que el software hará, sino cómo lo hará. Son muy importantes debido a que permiten a los clientes valorar las características no funcionales del producto como: usabilidad, rendimiento y portabilidad, que junto a las funcionalidades esperadas del software ayudarán a marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Es preciso tener en cuenta que la definición de los requerimientos no funcionales cobra un valor adicional porque los errores que puedan existir en ellos son difíciles y caros de resolver. [42]

- **Requerimientos de 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.

- **Requerimientos de seguridad**

Se mantendrá seguridad y control a nivel de usuario, garantizando su acceso sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema. Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento. Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. El sistema proporcionará un registro de actividades (log) de cada usuario. Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos. El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

- **Requerimientos de 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.

- **Requerimientos de soporte**

Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP. Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación. Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados. 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.

- **Requerimientos de 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 IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz 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

CAPÍTULO 2: CARACTERÍSTICAS DEL 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.

- **Requerimientos de software**

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

- **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 del motor de persistencia Hibernate.

- **Requerimientos 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.

- **Requerimientos de Interfaz**

- **Interfaces de usuario**

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegables 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 software**

Se interactuará con el sistema alas RIS para realizar solicitudes y obtener resultados de estudios radiológicos e imagenológicos.

- **Interfaces de comunicación**

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven). El sistema usará el formato estándar WSDL (Web Services Description Language) 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.

2.5 Modelo de casos de uso del sistema

El Modelo de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. De esta forma permitiendo de el establecimiento de un acuerdo entre clientes y desarrolladores sobre las condiciones y requerimientos que debe cumplir el sistema. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos. Es decir representa gráficamente a los procesos y su interacción con los actores y constituye una entrada de gran valor para las siguientes fases de construcción de un software.

2.5.1 Definición de actores

Un actor del sistema es una persona o la abstracción de un software que interactúa de alguna manera con el sistema: puede intercambiar información con él, ser un recipiente pasivo de información,

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

representar el rol que juegan una o varias personas, un equipo o un sistema automatizado. A continuación se definen los actores del sistema en desarrollo: [43]

ACTORES	JUSTIFICACION
Médico de hospitalización	Actor encargado de registrar la atención al paciente, 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 pediátrica, gestor de la hoja obstétrica, gestor de la hoja reumatológica, gestor de la hoja de gastroenterología, gestor de la hoja de radioterapia y medicina nuclear, gestor de la hoja hemato-oncológica, gestor de la hoja hemato-oncológica pediátrica y gestor de la hoja de cirugía colo-rectal. Se encarga también de buscar los resultados de los análisis emitidos por el laboratorio y anatomía patológica, crear indicaciones médicas, solicitudes de interconsulta, solicitudes de análisis de laboratorio, crear orden de admisión y demás solicitudes que realiza el médico.
Gestor de la Hoja general de hospitalización	Es el actor encargado de crear la hoja general de hospitalización a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja de traumatología y ortopedia	Es el actor encargado de crear la hoja de traumatología y ortopedia a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja neurológica	Es el actor encargado de crear la hoja neurológica a cada paciente que se encuentra hospitalizado
Gestor de la Hoja ginecológica	Es el actor encargado de crear la hoja ginecológica a cada paciente que se encuentra hospitalizado.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Gestor de la Hoja oftalmológica	Es el actor encargado de crear la hoja oftalmológica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja uro-genital	Es el actor encargado de crear la hoja uro- genital a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja ORL	Es el actor encargado de crear la hoja ORL a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja dermatológica	Es el actor encargado de crear la hoja dermatológica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja obstétrica	Es el actor encargado de crear la hoja obstétrica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja reumatológica	Es el actor encargado de crear la hoja reumatológica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja de gastroenterología	Es el actor encargado de crear la hoja de gastroenterología a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja de radioterapia y medicina nuclear	Es el actor encargado de crear la hoja de radioterapia y medicina nuclear a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja hemato-oncológica	Es el actor encargado de crear la hoja hemato-oncológica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja hemato-oncológica pediátrica	Es el actor encargado de crear la hoja hemato-oncológica pediátrica a cada paciente que se encuentra hospitalizado.
Gestor de la Hoja de cirugía colo-rectal	Es el actor encargado de crear la hoja de cirugía colo-rectal a cada paciente que se encuentra hospitalizado.

Tabla 2.1 Justificación de Actores del Sistema.

2.5.2 Vista global de actores

En el siguiente diagrama se puede observar cómo han sido separados mediante la relación de generalización-especialización los diferentes roles asociados a la atención del paciente en el módulo. Esta propuesta tiene como objetivo separar las labores de creación de las hojas de hospitalización en

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

actores diferentes, para facilitar que cada actor pueda crear la misma según la especialidad del servicio en que se encuentre.

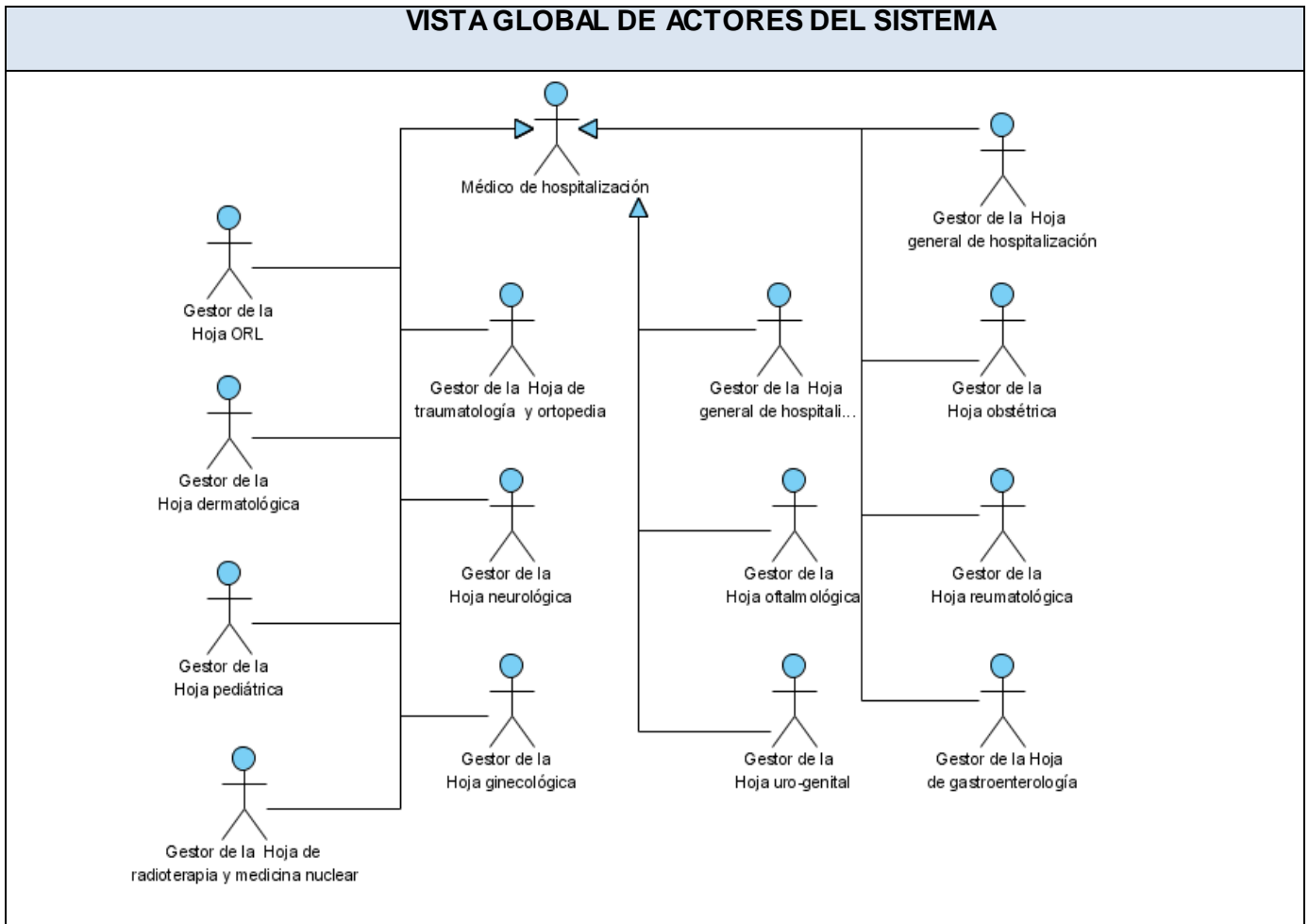


Figura 2.3 Vista Global Actores del Sistema

2.5.3 Diagrama de casos de uso

En este importante diagrama se representan las relaciones entre actores y casos de uso del sistema. A continuación se muestra el Diagrama de Casos de Uso del Módulo de Hospitalización:

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

DIAGRAMA DE CASOS DE USO DEL SISTEMA

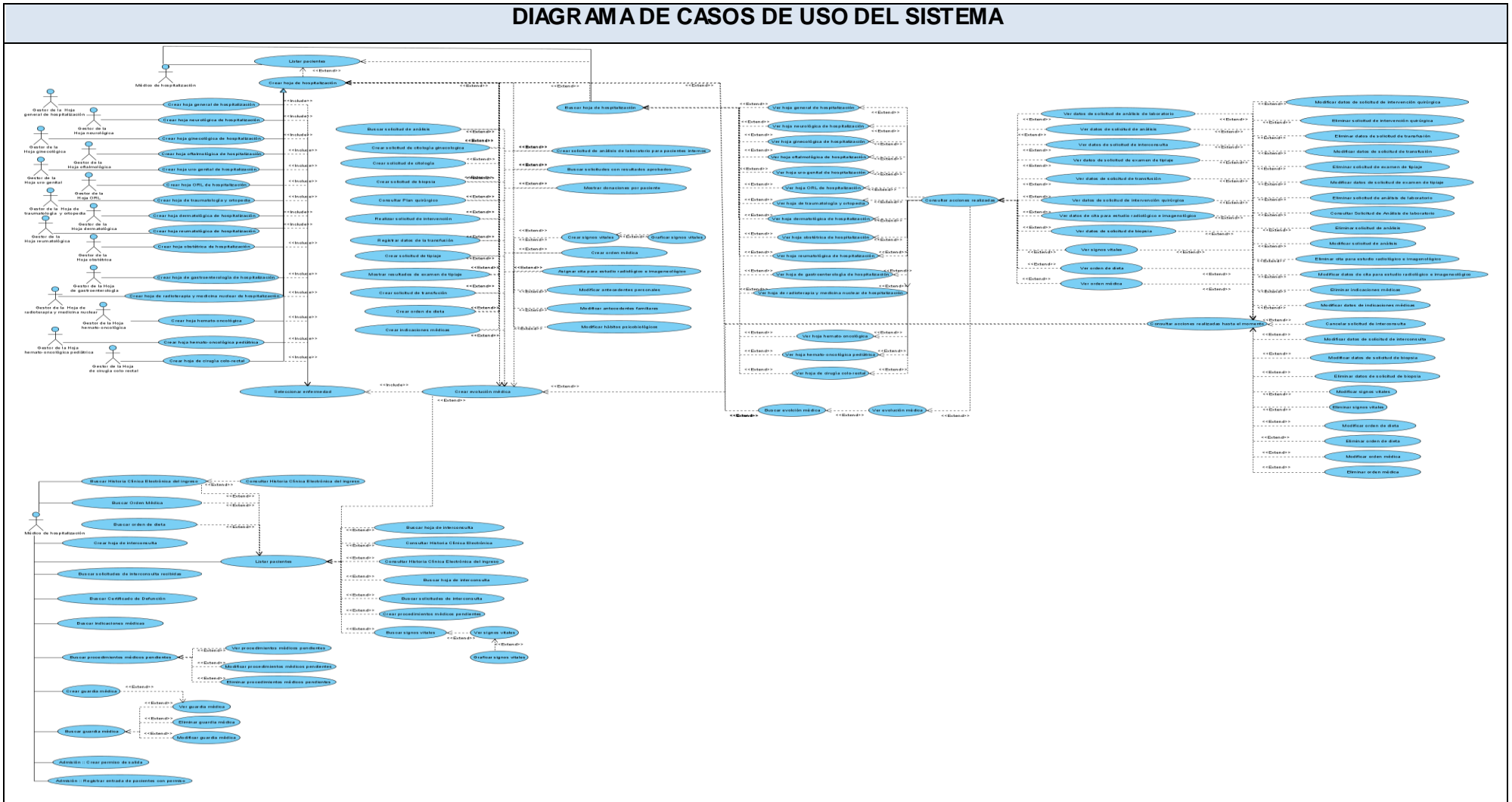


Figura 2.4 Diagrama de Casos de Uso *Proceso Atender paciente*.

2.5.4 Descripción Textual de los Casos de Uso

A continuación se describen los casos de uso identificados, mediante los cuales se satisfacen los requerimientos funcionales del sistema en desarrollo:

CUS_1	Listar pacientes
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a las opciones Ver por camas, Ver por pacientes o Ver por habitación. El sistema muestra los pacientes que se encuentran hospitalizados en el servicio al cual el médico accedió por el criterio seleccionado, terminando así el caso de uso.
Referencias:	RF1

Tabla 2.3 Descripción textual *CUS1_Listar pacientes*

CUS_2	Crear hoja general de hospitalización
Actores:	Gestor de la Hoja general de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción Crear hoja general de hospitalización. El sistema brinda la posibilidad de introducir y seleccionar los datos para crear la hoja general de hospitalización y permite acceder a varias opciones. El actor introduce y selecciona los datos de la hoja general de hospitalización y selecciona la o las opciones a realizar. El sistema crea la hoja general de hospitalización, terminando así el caso de uso.
Referencias:	RF2

Tabla 2.4 Descripción textual *CUS2_Crear hoja general de hospitalización*

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

CUS_3	Crear evolución médica
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción Evolucionar pacientes. El sistema muestra los datos generales relacionados con la historia clínica del paciente. El sistema brinda la posibilidad de introducir una observación y seleccionar el estado del paciente y el diagnóstico médico. Además permite acceder a varias opciones. El actor selecciona la o las opciones a realizar. El sistema crea la evolución médica, terminando así el caso de uso.
Referencias:	RF35

Tabla 2.5 Descripción textual *CUS3_Crear evolución médica*

CUS_4	Crear orden médica
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción Crear orden médica. El sistema brinda la posibilidad de introducir los datos de los medicamentos, tratamientos, dietas, preparaciones intravenosas y nutriciones parenterales. El actor introduce los datos de la orden médica. El sistema crea la orden médica terminando así el caso de uso.
Referencias:	RF54

Tabla 2.6 Descripción textual *CUS4_Crear orden médica*

CUS_5	Crear signos vitales
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción Crear signos vitales. El sistema brinda la posibilidad de introducir los datos para crear los signos vitales. El actor introduce los datos de los signos vitales. El sistema crea los signos vitales, terminando así el caso de uso.
Referencias:	RF38

Tabla 2.7 Descripción textual *CUS5_Crear signos vitales*

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

CUS_6	Crear solicitud de interconsulta
Actores:	Médico de hospitalización
Descripción:	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, terminando así el caso de uso.
Referencias:	RF58

Tabla 2.8 Descripción textual *CUS6_Crear solicitud de interconsulta*

CUS_7	Crear hoja de interconsulta
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción Crear hoja de interconsulta. El sistema permite seleccionar una de las solicitudes de interconsulta aceptadas y brinda la posibilidad de introducir los datos para crear la hoja de interconsulta. El actor introduce los datos. El sistema crea la hoja de interconsulta, terminando así el caso de uso.
Referencias:	RF64

Tabla 2.9 Descripción textual *CUS7_Crear hoja de interconsulta*

CUS_8	Seleccionar enfermedad
Actores:	Médico de hospitalización
Descripción:	El caso de uso inicia cuando el actor accede a la opción seleccionar enfermedad. El sistema brinda la posibilidad de introducir criterios de búsqueda para localizar enfermedades. El actor introduce el código o la descripción para realizar la búsqueda. El sistema busca y muestra las enfermedades que cumplen con los criterios de búsqueda. El actor selecciona o descarta las enfermedades terminando así el caso de uso.
Referencias:	RF89

Tabla 2.10 Descripción textual *CUS8_Seleccionar enfermedad*

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El desarrollo de este capítulo ha permitido alcanzar un mejor entendimiento del sistema y las restricciones que deben existir para satisfacer las necesidades de los clientes. Se especificaron todos los requerimientos funcionales y no funcionales del sistema, se identificaron los actores que intervienen. Así como todos los casos de uso, que fueron descritos de forma detallada y reflejan las funcionalidades recogidas previamente en los requerimientos. El desarrollo de este flujo de trabajo y los artefactos obtenidos a partir de él permiten comenzar con el Flujo de Trabajo de Diseño, que requiere de mayor esfuerzo por parte del equipo de desarrollo para la construcción de la solución de software propuesta.

CAPÍTULO 3: DISEÑO DEL SISTEMA

El objetivo del diseño es traducir los requisitos a una especificación que describe cómo implementar el sistema, definiendo cómo cumple sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. El principal objetivo de este capítulo está encaminado a comprender los aspectos relacionados con los requerimientos, lenguajes de programación, componentes reutilizables, tecnologías de distribución y concurrencia. Entre los artefactos que son mostrados se encuentran: Modelo de Diseño -especificándose la estructura y la definición de los elementos que este posee-, Diagramas de Clases y descripción de las clases del diseño.

3.1 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema. Además describe las clases más importantes, su organización en paquetes y subsistemas, sirviendo como abstracción de la implementación y utilizándose como entrada fundamental de las actividades de implementación. Es representado por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es una forma de organización de este artefacto, en porciones más manejables. Los casos de uso son realizados por las clases del diseño y sus objetos mediante colaboraciones en el Modelo de Diseño, denotando la realización de casos de uso del diseño. [44]

3.1.1 Estructuración

Para la conformación del Modelo de Diseño se hace necesario su organización en piezas más manejables: subsistemas o paquetes, que contengan sus interfaces y las dependencias entre cada una de ellas, conformando de esta manera la estructura principal del sistema. A continuación se muestra el diagrama con los paquetes del diseño, estructurados según el paradigma de la Arquitectura en tres capas.

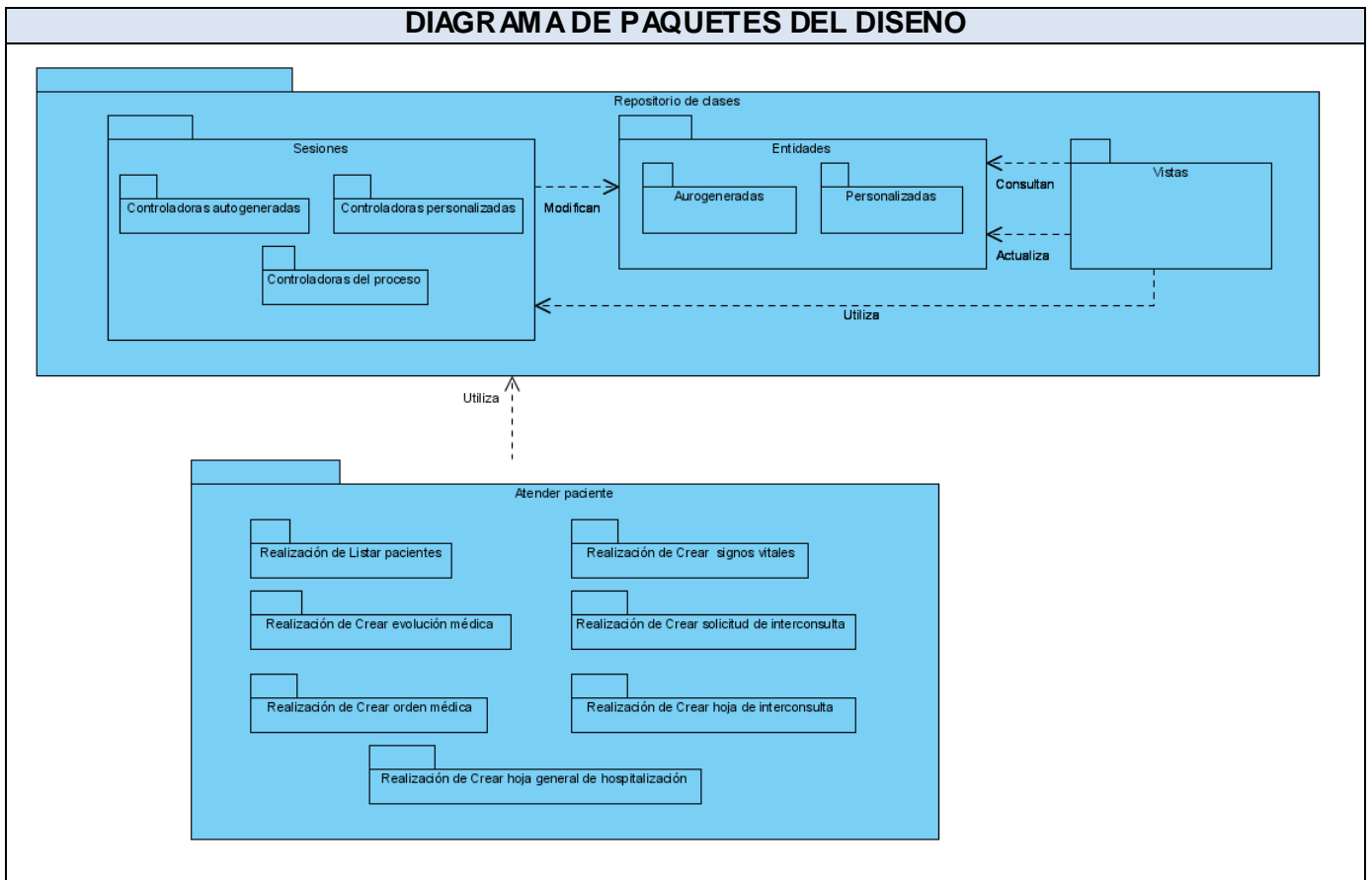


Figura 3.2 Diagrama de paquetes del diseño *Atender paciente*

Para posibilitar una mejor comprensión del Diagrama de paquetes del Diseño, se describe a continuación el contenido de cada uno de los paquetes contenidos en él.

El paquete Repositorio de clases contiene todas las clases definidas en el diseño de acuerdo a las tecnologías, que serán utilizadas en la implementación del módulo, estructuradas en tres paquetes:

Entidades: Contiene las entidades Autogeneradas y Personalizadas de la base de datos. Las Autogeneradas son aquellas obtenidas mediante el proceso de ingeniería inversa de la base de datos, utilizando para ello las herramientas ORM de Hibernate. Las Personalizadas son aquellas que son modificadas y, en algunos casos, pueden presentar relaciones de herencia o composición con las Autogeneradas.

Vistas: Contiene el conjunto de clases que conforman o representan la interfaz de usuario.

Sesiones: Contiene las clases Controladoras autogeneradas, las Controladoras personalizadas y las Controladoras propias del proceso. Las primeras son autogeneradas por el entorno de desarrollo, las segundas son aquellas que son modificadas y las terceras son propias del proceso.

En el diagrama se representa además, el paquete correspondiente al proceso Atender paciente, dentro del cual se representan los paquetes que contienen las realizaciones de los casos de uso identificados.

3.1.2 Definición de elementos de diseño

Para la realización de los Diagramas de Clases del Diseño se utiliza la extensión de UML para la utilización de estereotipos web. Esta extensión presenta como elementos más significativos tres clases UML “Server Page”, “Client Page” y “Form” empleadas para el código servidor, código cliente y formularios respectivamente.

El código servidor se encarga de construir o generar el resultado XHTML que conforma el código cliente (<<build>>), los formularios envían sus datos al código servidor para ser procesados los pedidos (<<submit>>), además forman parte del código cliente o resultado XHTML. Es por esto que la relación entre la clase empleada para el código cliente y la clase empleada para el formulario es de agregación. Entre las páginas clientes pueden existir vínculos (<<link>>) o redireccionamientos (<<redirect>>). Es importante destacar que una página cliente es construida por una sola página servidora; esta a su vez, puede completar su funcionamiento incluyendo código existente en otra página de este mismo tipo, utilizando la relación de inclusión (<<include>>), que aunque no es propia de la extensión de UML, las herramientas de modelado la consideran para representar todas las relaciones existentes en el modelo. [45]

Para la nomenclatura de estas clases en los Diagramas de Clases se siguió la siguiente estructura: PC_<NombreClaseCliente>, FR_<NombreFormulario>, PS_<NombreClaseServidora> y <NombreClaseControladora>, para las páginas clientes, formularios, clases servidoras ubicadas en la Capa de Presentación y clases de la Lógica de Negocio respectivamente. En los diagramas también aparecen las clases entidades y sus relaciones con las clases controladoras. Las mismas pueden ser reconocidas mediante los estereotipos <<Entity Bean>> y <<ORM Persistable>>. Pueden apreciarse

además dos clases muy importantes, RichFaces API y Entity Manager. RichFaces API representa la integración de los componentes de interfaz de usuario, validación y conversión de datos de RichFaces con las páginas clientes. La clase Entity Manager está estereotipada como <<Interface>> y representa la clase con las que se relacionan las controladoras para llevar a cabo las operaciones de persistencia de datos.

En la estructura de los Diagramas de Clases del Diseño se manifiesta la aplicación de los patrones de diseño, así como las restricciones que establecen sobre la arquitectura definida. Estos diagramas están estructurados siguiendo el patrón MVC, el cual permite la separación en capas de los objetos que componen el diseño del sistema, e introduce la necesidad de crear tres categorías de objetos principales: vistas, controladores y modelos.

Los modelos encapsulan el estado y exponen la funcionalidad de la aplicación, responden a consultas y notifican a las vistas sobre los cambios en su estado actual. Las vistas muestran y actualizan los modelos y le envían las acciones realizadas por el usuario a los controladores. Estos últimos por su parte, definen el comportamiento de la aplicación convirtiendo las acciones del usuario a actualizaciones de los modelos, seleccionando las vistas para la respuesta, una para cada responsabilidad. En los diagramas, el conjunto de clases clientes, servidoras y formularios representan las vistas, las clases entidades representan los modelos y las clases controladoras a los controladores.

La capa de persistencia o datos en el patrón MVC también forma parte del modelo. En esta capa se utilizan los patrones de diseño Abstract Factory, Singleton y los patrones de arquitectura de aplicaciones empresariales; más específicamente los patrones de persistencia o mapeo objeto relacional, descritos por Martin Fowler, en su libro "Patterns of Enterprise Application Architecture". Entre estos se encuentran los patrones Active Record, Identity map, Identity field, Foreign key mapping, Association table mapping, Lazy load y Query object.

Otros de los patrones de diseño que se utilizan son los patrones GRASP, los cuales tienen como objetivo la descripción de los principios fundamentales de diseño de objetos para la asignación de responsabilidades y dentro de estos, los patrones Experto, Creador, Alta cohesión y Bajo acoplamiento. Mediante la asignación a cada clase de las tareas o responsabilidades que estas pueden realizar en dependencia de la información que contienen se evidencia el uso de los patrones Experto y Creador. Estos conservan el encapsulamiento y definen quién será el responsable de crear

una instancia de una clase respectivamente. Al utilizar los patrones Alta cohesión y Bajo acoplamiento se permite la colaboración entre clases o elementos del diseño sin que se afecte su reutilización y entendimiento cuando se encuentren aislados.

3.2 Diagramas de clases del diseño

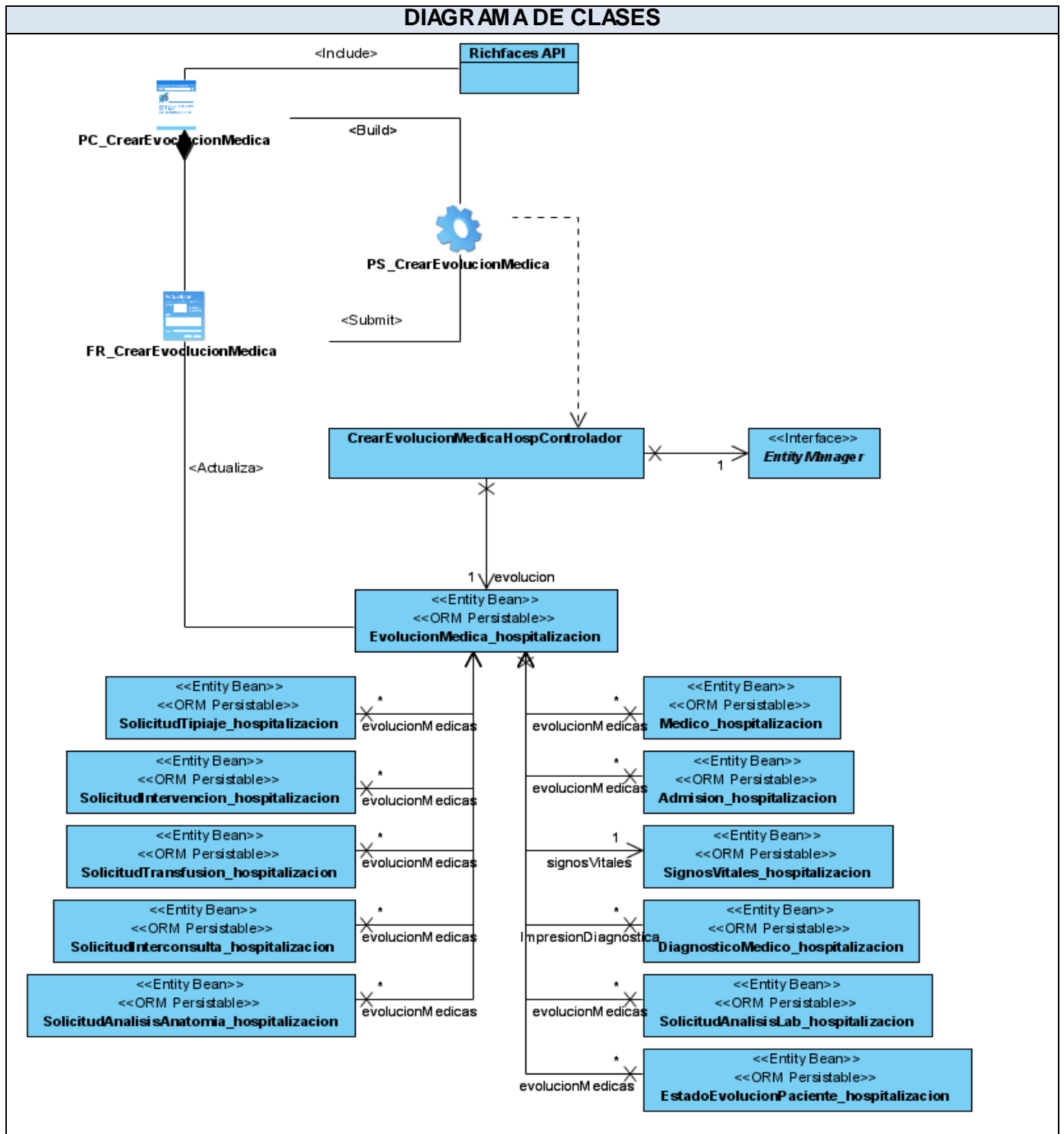


Figura 3.2 Diagrama de Clases del Diseño *CUS3_Crear evolución médica*

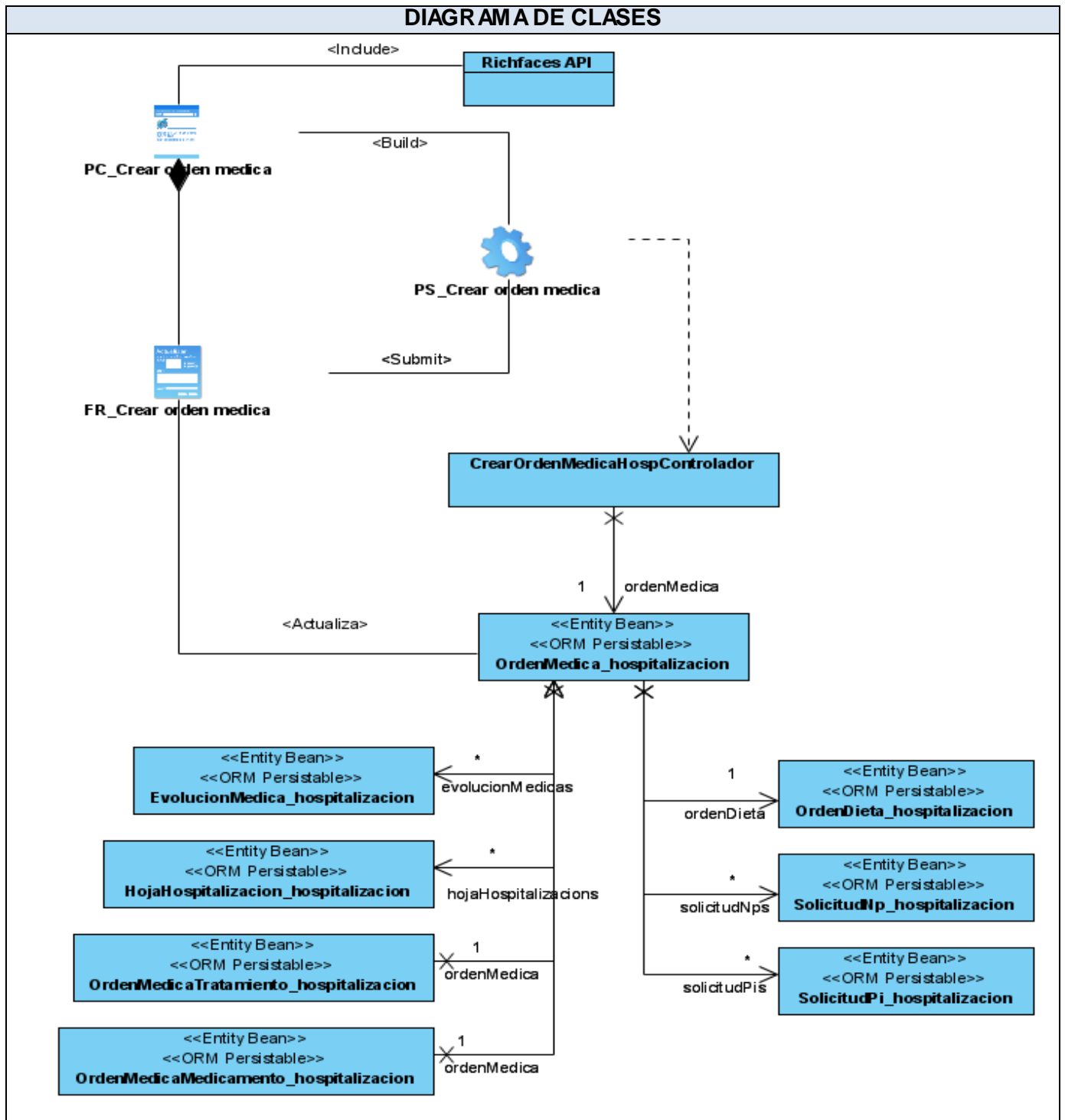


Figura 3.3 Diagrama de Clases del Diseño *CUS4_Crear orden médica*

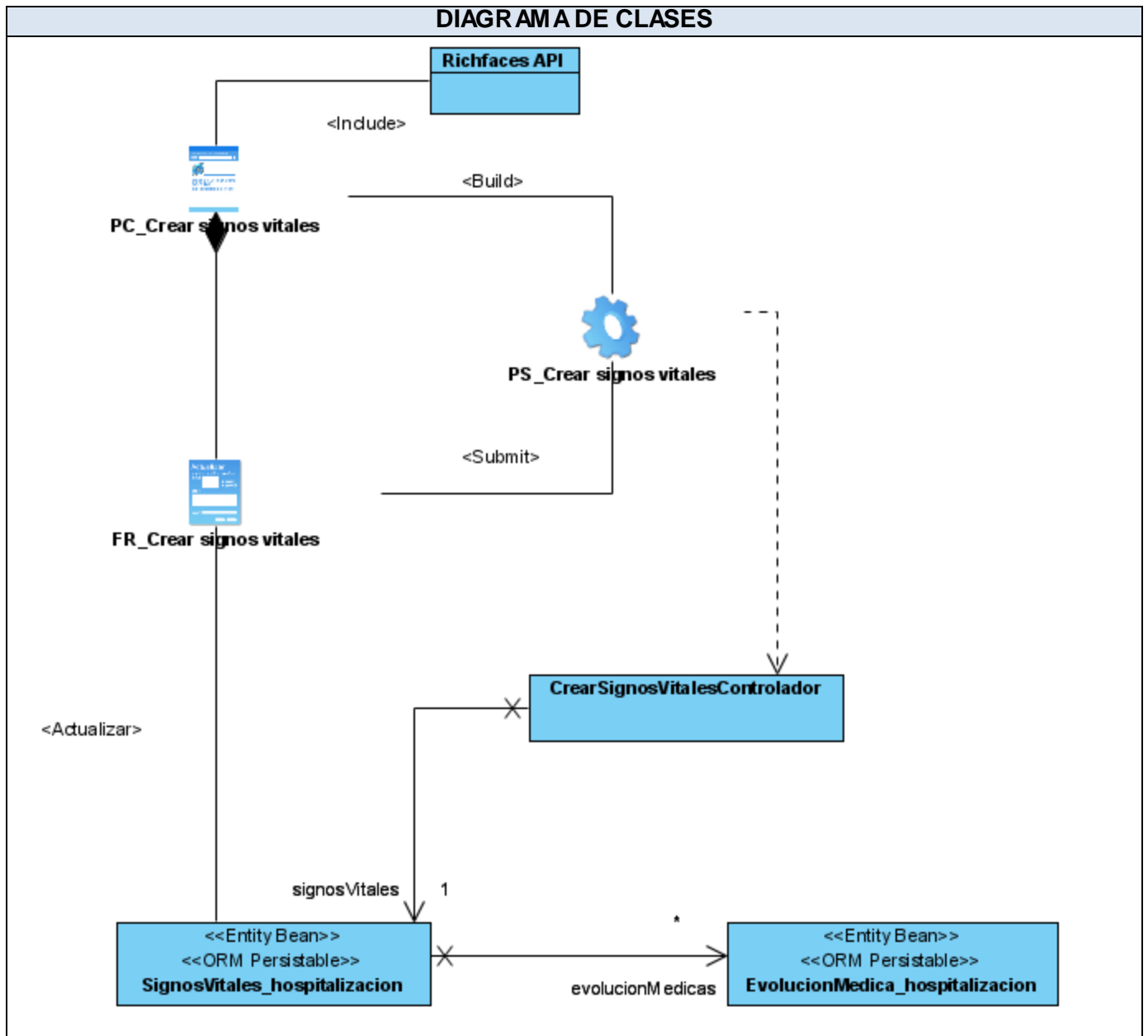


Figura 3.4 Diagrama de Clases del Diseño *CUS5_Crear signos vitales*

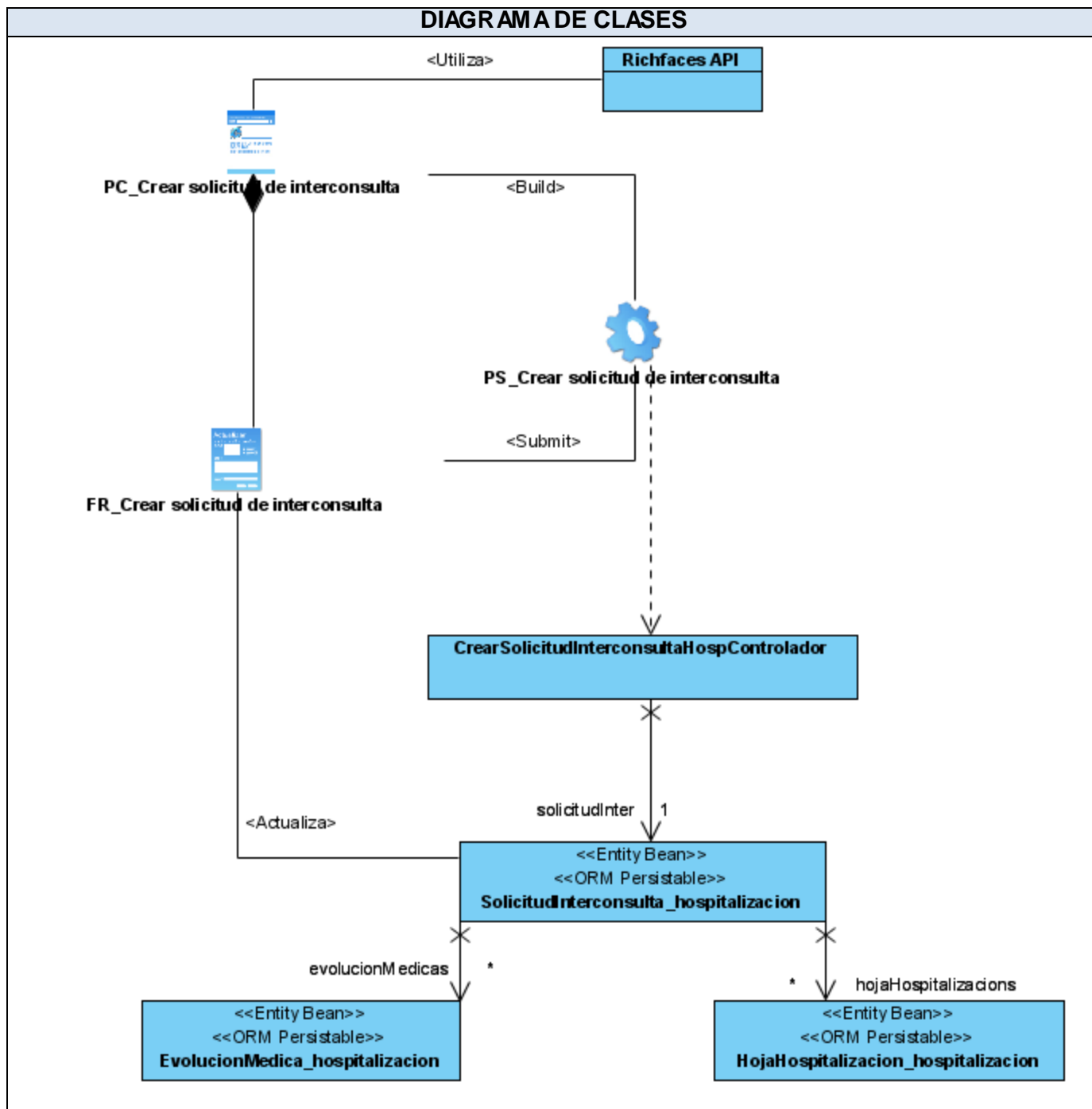


Figura 3.5 Diagrama de Clases del Diseño *CUS6_Crear solicitud de interconsulta*

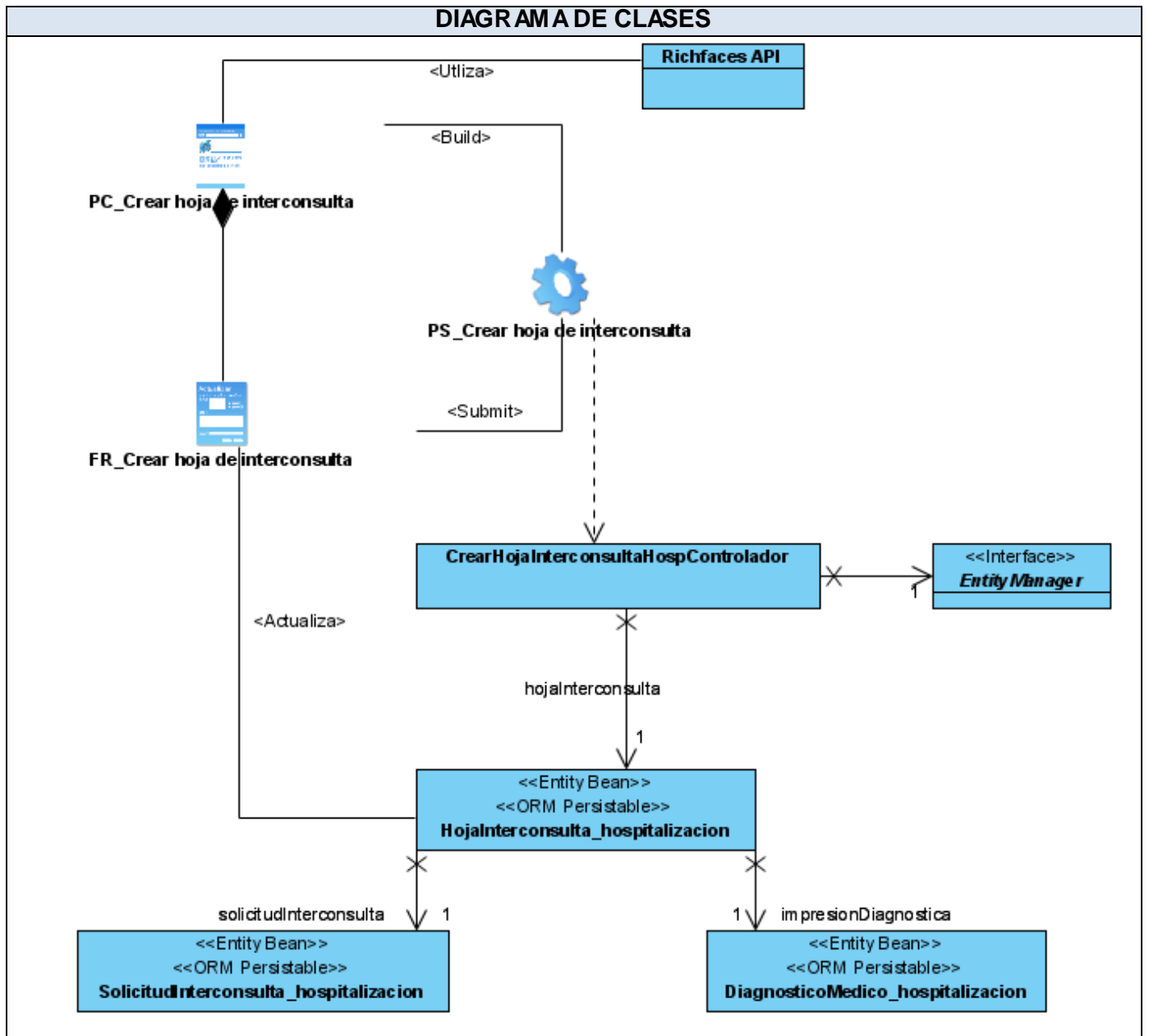


Figura 3.6 Diagrama de Clases del Diseño *CUST7_Crear hoja de interconsulta*

3.3 Descripción de clases del diseño

Seguidamente serán explicadas algunas de las clases que han sido identificadas para su futura implementación. Del mismo modo serán descritas algunas de las responsabilidades que realizarán las páginas servidoras que responden a la Lógica de Negocio. De esta manera se tendrá una comprensión mayor del funcionamiento que tendrá el sistema en desarrollo.

3.3.1 Descripción de páginas clientes

Nombre: PC_Crear evolución médica
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al médico de hospitalización la creación de una evolución médica.

Tabla 3.1 Descripción página cliente *PC_Crear evolución médica*

Nombre: PC_Crear hoja general de hospitalización
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente. Le permite al médico de hospitalización la creación de la hoja general de hospitalización, la que cuenta con tres agrupaciones por las siguientes categorías: examen funcional, examen físico y signos vitales.

Tabla 3.2 Descripción página cliente *PC_Crear hoja general de hospitalización*

Nombre: PC_Crear orden médica
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente. Le permite al médico de hospitalización la creación de una orden médica, la que consta de cinco agrupaciones por las siguientes categorías: medicamentos, tratamientos, dieta, nutrición parenteral y preparación

Tabla 3.3 Descripción página cliente *PC_Crear orden médica*

Nombre: PC_Crear signos vitales
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente. Le permite al médico de hospitalización la creación de de signos vitales.

Tabla 3.4 Descripción página cliente *PC_Crear signos vitales*

Nombre: PC_Crear solicitud de interconsulta
Tipo de Clase: Página Cliente
Descripción General: Es una página web que se ejecuta del lado del cliente. Le permite al médico de hospitalización la creación de una orden médica, que consta de cinco grupos: medicamentos, tratamientos, dieta, nutrición parenteral y preparación. intravenosa.

Tabla 3.5 Descripción página cliente *PC_Crear solicitud de interconsulta*

Nombre: PC_Crear hoja de interconsulta
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente. Le permite al médico de hospitalización la creación de la hoja de interconsulta.

Tabla 3.6 Descripción página cliente *PC_Crear hoja de interconsulta*

3.3.2 Descripción de páginas servidoras

Nombre: PS_Crear hoja general de hospitalización
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear hoja general de hospitalización es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear hoja general de hospitalización. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.7 Descripción página cliente *PS_Crear hoja general de hospitalización*

Nombre: PS_Crear evolución médica
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear evolución médica es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear evolución médica. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.8 Descripción página cliente *PS_Crear evolución médica*

Nombre: PS_Crear orden médica
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear orden médica es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear orden médica. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.9 Descripción página cliente *PS_Crear orden médica*

Nombre: PS_Crear signos vitales
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear signos vitales es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear signos vitales. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.10 Descripción página cliente *PS_Crear signos vitales*

Nombre: PS_Crear solicitud de interconsulta
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear solicitud de interconsulta es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear solicitud de interconsulta. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.11 Descripción página cliente *PS_Crear solicitud de interconsulta*

Nombre: PS_Crear hoja de interconsulta
Tipo de Clase: Página Servidora
Descripción General: La clase PS_Crear hoja de interconsulta es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Crear hoja de interconsulta. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.12 Descripción página cliente *PS_Crear hoja de interconsulta*

3.3.3 Descripción de clases del negocio

Nombre: CrearHojaGeneralHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void asignarCampos()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización
+ void seleccionarImpresionDiagEnf()	Permite seleccionar una enfermedad para conformar la impresión diagnóstica
+ void eliminarImpresionDiagEnf(Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar la impresión diagnóstica.
+ void seleccionarDiagFinalEnf ()	Permite seleccionar una enfermedad para conformar diagnóstico final
+ void eliminarDiagFinalEnf (Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar el diagnóstico final.
+ void save()	Persiste en la base de datos la Hoja general de hospitalización

Tabla 3.13 Descripción clase controladora *CrearHojaGeneralHospControlador*

Nombre: CrearEvolucionMedicaHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void asignarCampos()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización.
+ void seleccionarImpresionDiagEnf()	Permite seleccionar una enfermedad para conformar la impresión diagnóstica.
+ void eliminarImpresionDiagEnf(Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar la impresión diagnóstica.
+ void seleccionarDiagFinalEnf ()	Permite seleccionar una enfermedad para conformar diagnóstico final.
+ void eliminarDiagFinalEnf (Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar el diagnóstico final.
+ void save()	Persiste en la base de datos la Evolución médica.

Tabla 3.14 Descripción página cliente *CrearEvolucionMedicaHospControlador*

Nombre: CrearOrdenMedicaHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void asignarCampos()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización.
+ void seleccionarImpresionDiagEnf()	Permite seleccionar una enfermedad para conformar la impresión diagnóstica.
+ void eliminarImpresionDiagEnf(Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar la impresión diagnóstica.
+ void seleccionarDiagFinalEnf()	Permite seleccionar una enfermedad para conformar diagnóstico final.
+ void eliminarDiagFinalEnf (Integer id)	Permite eliminar una enfermedad dada del conjunto de enfermedades seleccionadas para conformar el diagnóstico final.
+ void save()	Persiste en la base de datos la Evolución médica.

Tabla 3.15 Descripción página cliente *CrearOrdenMedicaHospControlador*

Nombre: CrearSignosVitalesHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void asignarCampos()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización.
+ void save()	Persiste en la base de datos los Signos vitales.

Tabla 3.16 Descripción página cliente *CrearSignosVitalesHospControlador*

Nombre: CrearSolicitudInterconsultaHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void asignarCampos()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización.
+ void save()	Persiste en la base de datos los Signos vitales.

Tabla 3.17 Descripción página cliente *CrearSolicitudInterconsultaHospControlador*

Nombre: CrearHojaInterconsultaHospControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
+ void crear()	Inicializa todos los atributos necesarios para la creación de la Hoja general de hospitalización.
+ void save()	Persiste en la base de datos los Signos vitales.

Tabla 3.17 Descripción página cliente *CrearHojaInterconsultaHospControlador*

Al concluir este capítulo donde fue obtenido el Modelo de Diseño y se describió la realización física de los casos de uso. Centrándose en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a desarrollar. Lo que constituyó una abstracción para la implementación y es utilizado como una entrada fundamental en las actividades que se realizan en el Flujo de Trabajo de Implementación.

CAPÍTULO 4: IMPLEMENTACIÓN

Este capítulo describe cómo los elementos del Diseño son implementados en términos de componentes, así como su organización física de acuerdo a los nodos específicos en el Diagrama de Despliegue. Este artefacto junto con el Diagrama de Componentes conforma el Modelo de Implementación. Además se aborda sobre el tratamiento de excepciones, la seguridad del sistema, así como las estrategias de codificación, estándares y estilos que se utilizan.

4.1 Modelo de Datos

Es el modelo que describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. El Modelo de Datos tiene gran importancia en el ciclo de desarrollo de software, y de manera particular para la fase de implementación, pues define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; y atributos: características básicas de dichos objetos y relaciones: forma en que se enlazan los objetos entre sí. [46]

4.1.1 Descripción de las tablas de la Base de Datos

A continuación se realiza la descripción de las tablas de la Base de Datos especificando el nombre de cada atributo, su tipo, si no admite valores nulos y una breve descripción del mismo. En las descripciones se especificará además si el atributo es llave primaria (PK) o llave foránea (FK).

Nombre: admisión			
Descripción: Almacena la admisión de un paciente al hospital, ya sea por consulta externa o por emergencia.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la admisión (PK).
id_hoja_frontal	integer	no	Identificador único para la tabla hoja_frontal (FK).
id_orden_admision	integer	no	Identificador único para la tabla orden_admision (FK).
fecha	date	no	Fecha en que se realiza la admisión.
id_cama	integer	no	Identificador único para la tabla cama (FK).
observaciones	text		Observaciones emitidas en la admisión.
id_usuario	integer	no	Identificador único para la tabla usuario (FK).
peso_ingreso	numeric		Peso del paciente al ingresar.
id_estado_evolucion_paciente	integer		Identificador único para la tabla estado_evolucion_paciente (FK).

Tabla 4.1 Descripción de la tabla de la Base de Datos: *admisión*.

Nombre: hoja_hospitalización			
Descripción: Almacena los datos de la hoja de hospitalización de un paciente en cada en cada ingreso o traslado a un servicio.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único para la hoja de hospitalización (PK).
fecha_creada	date	no	Fecha en que es creada la hoja.
hora_creada	time	no	Hora en que es creada la hoja.
id_medico	integer	no	Identificador único para la tabla medico (FK).
id_admision	integer	no	Identificador único para la tabla admision (FK).
id_diagnostico_final	integer		Identificador único para la tabla diagnostico_final (FK).
id_impresion_diagnostica	integer		Identificador único para la tabla impresion_diagnostica (FK).

CAPÍTULO 4: IMPLEMENTACIÓN

id_diagnostico_princ_final	integer		Identificador único para la tabla diagnostico_princ_final (FK).
observaciones	text		Observaciones que se emitidas.
id_orden_medica	integer		Identificador único para la tabla orden_medica (FK).
id_hoja_general	integer		Identificador único para la tabla hoja_general (FK).
id_hoja_dermatologia	integer		Identificador único para la tabla hoja_dermatologia (FK).
id_hoja_gastroenterologia	integer		Identificador único para la tabla hoja_gastroenterologia (FK).
id_hoja_ginecologia	integer		Identificador único para la tabla hoja_ginecologia (FK).
id_hoja_cirugia_colo_rectal	integer		Identificador único para la tabla hoja_cirugia_colo_rectal (FK).
id_hoja_hemato_onc	integer		Identificador único para la tabla hoja_hemato-onc (FK).
id_hoja_hemato_onc_pediatica	integer		Identificador único para la tabla hoja_hemato_onc_pedriatica (FK).
id_hoja_obstetrica	integer		Identificador único para la tabla hoja_obstetrica (FK).
id_hoja_radioterapia	integer		Identificador único para la tabla hoja_radioterapia (FK).
id_hoja_ofthalmologia	integer		Identificador único para la tabla hoja_ofthalmologia (FK).
id_hoja_orl	integer		Identificador único para la tabla hoja_orl (FK).
id_hoja_urogenital	integer		Identificador único para la tabla hoja_urogenital (FK).
id_hoja_traumatologia_ortopedia	integer		Identificador único de la tabla hoja_traumatologia_ortopedia (FK).
id_hoja_neurologia	integer		Identificador único para la tabla hoja_neurologia (FK).
id_solic_exam_tipeaje	integer		Identificador único para la tablaolic_exam_tipiaje (FK).
id_solic_interconsulta	integer		Identificador único para la tablaolic_interconsulta (FK).
id_solic_intervencion	integer		Identificador único para la tablaolic_intervencion (FK).
id_solic_analisis_lab	integer		Identificador único para la tablaolic_analisis_lab (FK).
id_estado_evolucion_paciente	integer	no	Identificador único para la tabla estado_evolucion_paciente (FK).
conclusiones	text		Conclusiones que se emiten en la hoja.
id_impresion_diag_princ_final	integer		Identificador único para la tabla impresion_diag_princ_final (FK).
id_tecnico_recodifica_diagnostico	integer		Identificador único para la tabla tecnico_recodifica_diagnostico (FK).

Tabla 4.2 Descripción de la tabla de la Base de Datos: *hoja_hospitalización*.

Nombre: general			
Descripción: Almacena los datos de la hoja general de hospitalización de un paciente en cada en cada ingreso o traslado a un servicio.			
Atributo	Tipo	Nulo	Descripción
id	integer		Identificador único de la hoja general. (PK)
id_examen_funcional	integer		Identificador único para la tabla examen_funcional. (FK)
id_signos_vitales	integer		Identificador único para la tabla signos_vitales. (FK)
id_examen_fisico	integer		Identificador único para la tabla examen_fisico. (FK)

Tabla 4.3 Descripción de la tabla de la Base de Datos: *general*.

Nombre: evolución_médica			
Descripción: Almacena los datos de la evolución médica de un paciente.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la evolución médica (PK).
fecha	date	no	Fecha en que se realiza la evolución.
hora	time	no	Hora en que se realiza la evolución.
id_medico	integer	no	Identificador único para la tabla medico (FK).
id_diagnostico_final	integer		Identificador único para la tabla diagnostico_final (FK).
id_impression_diagnostica	integer		Identificador único para la tabla impresión_diagnostica (FK).
observaciones	varchar	no	Observaciones emitidas.
id_orden_medica	integer		Identificador único para la tabla orden_medica (FK).
id_signos_vitales	integer		Identificador único para la tabla signos_vitales (FK).
id_solicitud_intervencion	integer		Identificador único para la tabla solicitud_intervencion (FK).
id_solic_interconsulta	integer		Identificador único para la tabla solic_interconsulta (FK).
id_sol_analisis_lab	integer		Identificador único para la tabla sol_analisis_lab (FK).
id_sol_exam_tipeaje	integer		Identificador único para la tabla sol_exam_tipeaje (FK).
id_estado_evolucion_paciente	integer	no	Identificador único para la tabla estado_evolucion_paciente (FK).
id_diagnostico_princ_final	integer		Identificador del diagnóstico final principal (FK).
id_tecnico_codifica_diag_final	integer		Identificador del técnico que realiza la re codificación (FK).

Tabla 4.4 Descripción de la tabla de la Base de Datos: *evolución_médica*.

Nombre: orden_médica			
Descripción: Almacena la orden médica realizada al paciente.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la orden médica.
fecha	date	no	Fecha en que se realiza la orden médica.
hora	time	no	Hora en que se realiza la orden médica.
observaciones	varchar		Observaciones que se emiten.
id_medico	integer	no	Identificador único para la tabla medico (FK).
id_hoja_frontal	integer	no	Identificador único para la tabla hoja_frontal (FK).
vigente	boolean		La orden médica vigente es la última que se haya creado.
id_orden_dieta	integer		Identificador único para la tabla orden_dieta (FK).
id_solicitud_np	integer		Identificador único para la tabla solicitud_np (FK).
id_solicitud_pi	integer		Identificador único para la tabla solicitud_pi (FK).

Tabla 4.5 Descripción de la tabla de la Base de Datos: *orden_médica*.

Nombre: orden_medica_medicamentos			
Descripción: Almacena los medicamentos asociados a una orden médica.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la orden_medica_medicamentos (PK).
id_orden_medica	integer	no	Identificador único para la tabla orden_medica (FK).
id_medicamento	integer	no	Identificador único para la tabla medicamento (FK).
dosis	integer	no	Dosis que se indica para la administración del medicamento.
id_via_administracion	integer	no	Identificador único para la tabla via_administracion (FK).
id_unidad_tiempo	integer	no	Identificador único para la tabla unidad_tiempo (FK).
valor_tiempo	integer	no	Valor que indica cada cuanto tiempo se le debe administrar el medicamento al paciente.
opcional	boolean	no	Indica si el medicamento indicado es opcional o no.
desc_med_opcional	text		Descripción de la conducta a seguir para el medicamento opcional.
id_estado	integer		Identificador único para la tabla estado (FK).
fecha_suspension	integer	no	Fecha en que debe suspenderse la administración del medicamento al paciente.

Tabla 4.6 Descripción de la tabla de la Base de Datos: *orden_médica_medicamentos*.

Nombre: orden_médica_tratamiento			
Descripción: Almacena los tratamientos asociados a una orden médica.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la orden médica tratamiento (PK).
id_orden_medica	integer	no	Identificador único para la tabla orden_medica (FK).
id_tratamiento_medico	integer	no	Identificador único para la tabla tratamiento_medico (FK).
fecha_suspension	date		Fecha en que debe suspenderse la administración del tratamiento al paciente.
valor_unidad_tiempo	integer	no	Valor que indica cada cuanto tiempo se le debe administrar el tratamiento al paciente.
id_unidad_tiempo	integer	no	Identificador único para la tabla unidad_tiempo (FK).

Tabla 4.7 Descripción de la tabla de la Base de Datos: *orden_médica_tratamiento*.

Nombre: orden_dieta			
Descripción: Almacena la orden de dieta asociada a una orden médica.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la orden de dieta (PK).
id_tipo_dieta	integer	no	Identificador único para la tabla tipo_dieta (FK).
observaciones	varchar		Observaciones que se emiten.
fecha	date	no	Fecha de creación de la orden de dieta.
hora	time	no	Hora de creación de la orden de dieta.
id_medico	integer	no	Identificador único para la tabla medico (FK).

Tabla 4.8 Descripción de la tabla de la Base de Datos: *orden_dieta*.

Nombre: solicitud_np			
Descripción: Almacena la solicitud de nutrición parenteral asociada a una orden médica.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la solicitud de nutrición parenteral (PK).
volumen_total	integer		Volumen total de la solución.
vel_inf_ml_h	real		Velocidad de infusión ml/h.
vel_inf_gt_min	real		Velocidad de infusión gotas/min.
calorias_total	integer		Total de calorías a administrar.
osmolaridad	real		Osmolaridad de la nutrición parenteral.

CAPÍTULO 4: IMPLEMENTACIÓN

id_estado	integer		Identificador único para la tabla estado (FK).
fecha_inicio	date	no	Fecha en que se debe comenzar a administrar.
fecha_fin	date	no	Fecha en que debe culminar su administración al paciente.
id_unidad_tiempo	integer	no	Identificador único para la tabla unidad_tiempo (FK).
valor_tiempo	integer	no	Valor que indica cada cuanto tiempo se le debe administrar al paciente la nutrición parenteral.

Tabla 4.9 Descripción de la tabla de la Base de Datos: *solicitud_np*.

Nombre: solicitud_pi			
Descripción: Almacena la solicitud de preparación intravenosa asociada a una orden médica.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la solicitud de preparación intravenosa (PK).
fecha_solicitud	date	no	Fecha en que se crea.
solucion	varchar	no	Nombre de la solución acuosa.
cantidad	integer	no	Cantidad que se va a administrar.
id_estado	integer	no	Identificador único para la tabla estado (FK).
fecha_culminacion	date	no	Fecha en que debe dejarse de administrar al paciente.
id_unidad_tiempo	integer	no	Identificador único para la tabla unidad_tiempo (FK).
valor_tiempo	integer	no	Valor que indica cada cuanto tiempo se le debe administrar al paciente la preparación intravenosa.

Tabla 4.10 Descripción de la tabla de la Base de Datos: *solicitud_pi*.

Nombre: signos_vitales			
Descripción: Almacena los signos vitales de un paciente.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la medición de signos vitales (PK).
p_a_diastolica	real	no	Valor de la presión arterial diastólica.
p_a_sistolica	real	no	Valor de la presión arterial sistólica.
p_a_media	real	no	Valor de la presión arterial media.
p_valor	real	no	Valor del pulso.
id_p_caracteristica	integer	no	Identificador único para la tabla de p_caracteristica (FK).
id_p_ubicacion	integer	no	Identificador único para la tabla p_ubicacion (FK).
id_f_r_caracteristicas	integer	no	Identificador único para la tabla de f_r_caracteristicas

CAPÍTULO 4: IMPLEMENTACIÓN

			(FK).
f_r_valor	real	no	Valor de la frecuencia respiratoria.
temperatura	real	no	Valor de la temperatura.
circunferencia_cefalica	real		Valor de la circunferencia cefálica.
saturacion_o2	real		Valor de la saturación de oxígeno.
diuresis_horaria	real		Valor de la diuresis horaria.
presion_venosa_central	real		Valor de la presión venosa central.
peso	real		Valor del peso.
talla	real		Valor de la talla.
id_signos_vitales_feto	integer		Identificador único para la tabla de signos_vitales_feto (FK).
observacion	text		Observaciones que se emiten.
fecha	date	no	Fecha en que se emiten.
hora	time	no	Hora en que se emiten.

Tabla 4.11 Descripción de la tabla de la Base de Datos: *signos_vitales*.

Nombre: solicitud_interconsulta			
Descripción: Almacena los datos de una solicitud de interconsulta de un paciente.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la solicitud de interconsulta (PK).
id_servicio_solicitante	integer	no	Identificador único para la tabla servicio, representando el servicio solicitante (FK).
id_servicio_solicitado	integer	no	Identificador único para la tabla servicio, representando el servicio solicitado (FK).
motivo_interconsulta	varchar	no	Resumen del motivo de la interconsulta.
resumen_caso	varchar		Resumen del caso.
fecha	date	no	Fecha en que se emite.
hora	time	no	Hora en que se emite.
ubicacion_paciente	varchar		Ubicación del paciente.
aceptada	boolean		Es aceptada por el médico del servicio receptor.

Tabla 4.12 Descripción de la tabla de la Base de Datos: *solicitud_interconsulta*.

Nombre: hoja_interconsulta			
Descripción: Almacena los datos de la hoja de interconsulta.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único de la hoja de interconsulta (PK).
fecha	date	no	Fecha en que se emite.
hora	time	no	Hora en que se emite.
descripcion_hallazgos	varchar		Descripción de los hallazgos encontrados.
recomendaciones	varchar		Recomendaciones que se emiten durante la consulta.
id_impresion_diagnostica	integer		Identificador único para la tabla de impresion_diagnostica (FK).
id_diagnostico_final	integer		Identificador único para la tabla de diagnostico_final (FK).
id_solicitud	integer	no	Identificador único para la tabla de solicitud_interconsulta (FK).

Tabla 4.13 Descripción de la tabla de la Base de Datos: *hoja_interconsulta*.

Nombre: estado_evolucion_paciente (Nomenclador)			
Descripción: Almacena el estado del paciente.			
Atributo	Tipo	Nulo	Descripción
id	integer	no	Identificador único del estado del paciente (PK).
valor	varchar	no	Valor del estado (Grave, De cuidado, En estudio y mejorado).
prioridad	integer	no	Prioridad para la atención según el estado del paciente. (4, 3, 2, 1)

Tabla 4.14 Descripción de la tabla de la Base de Datos: *estado_evolucion_paciente*.

4.2 Modelo de Implementación

Está conformado por los Diagramas de Componentes y de Despliegue, describiendo cómo los elementos del Modelo de Diseño se implementan en términos de componentes, archivos de código fuente y ejecutables. El Modelo de Implementación describe además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. [47].

4.2.1 Diagrama de Componentes

Es una representación gráfica que muestra un conjunto de elementos del modelo tales como componentes, subsistemas o paquetes de implementación y sus relaciones. Permite modelar la vista estática del sistema mostrando las dependencias lógicas entre un conjunto de componentes de software. Este diagrama se estructura en paquetes, que son divisiones físicas del sistema. Los paquetes están organizados en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Los componentes pueden ser de código fuente, librerías, binarios o ejecutables y tienen relaciones de traza con los elementos del modelo que implementan. En el Diagrama de Componentes elaborado se evidencian las relaciones existentes entre los paquetes y los componentes. Dichas relaciones son de inclusión y pueden ser identificadas por el estereotipado <<import>>. [48]

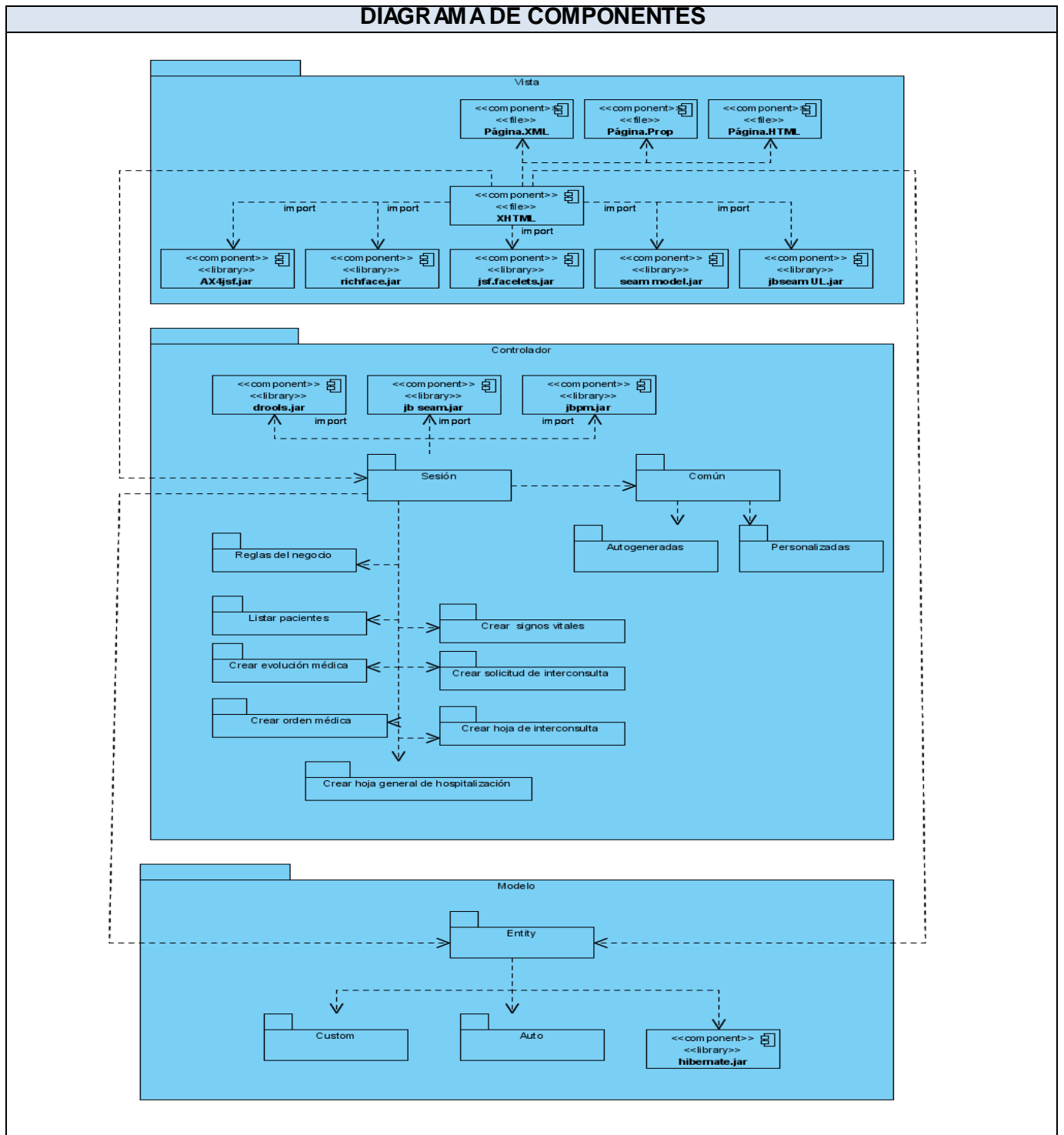


Figura 4.2 Diagrama de componentes

El paquete Vista está conformado por los archivos XHTML, los page.xml, los archivos properties y las librerías RichFaces, Ajax4Jsf, JSF Facelets y Seam UI. Los archivos XHTML contienen el código fuente de las páginas que conforman la interfaz visual del módulo e importan las librerías mencionadas para utilizar los componentes de interfaz de usuario, capacidad AJAX y la utilización de plantillas que estas brindan. Los archivos page.xml contienen las reglas de navegación de la página y los archivos properties permiten la internacionalización de la interfaz de usuario y definen los mensajes a mostrar por cada cultura idiomática a utilizar.

Cada archivo XHTML utiliza los archivos java del paquete Controlador, que contienen la lógica del negocio de la aplicación. Estos a su vez importan las librerías Seam, jBPM y Drools, las cuales brindan las utilidades del framework definido para la implementación del módulo y permiten administrar los procesos y las reglas del negocio respectivamente. Además pueden utilizar los archivos java contenidos en el paquete Modelo, los que mediante la librería Hibernate permiten la representación del Modelo de Datos utilizando el mapeo objeto relacional. A continuación se representa el Diagrama de Componentes correspondiente a un caso de uso significativo

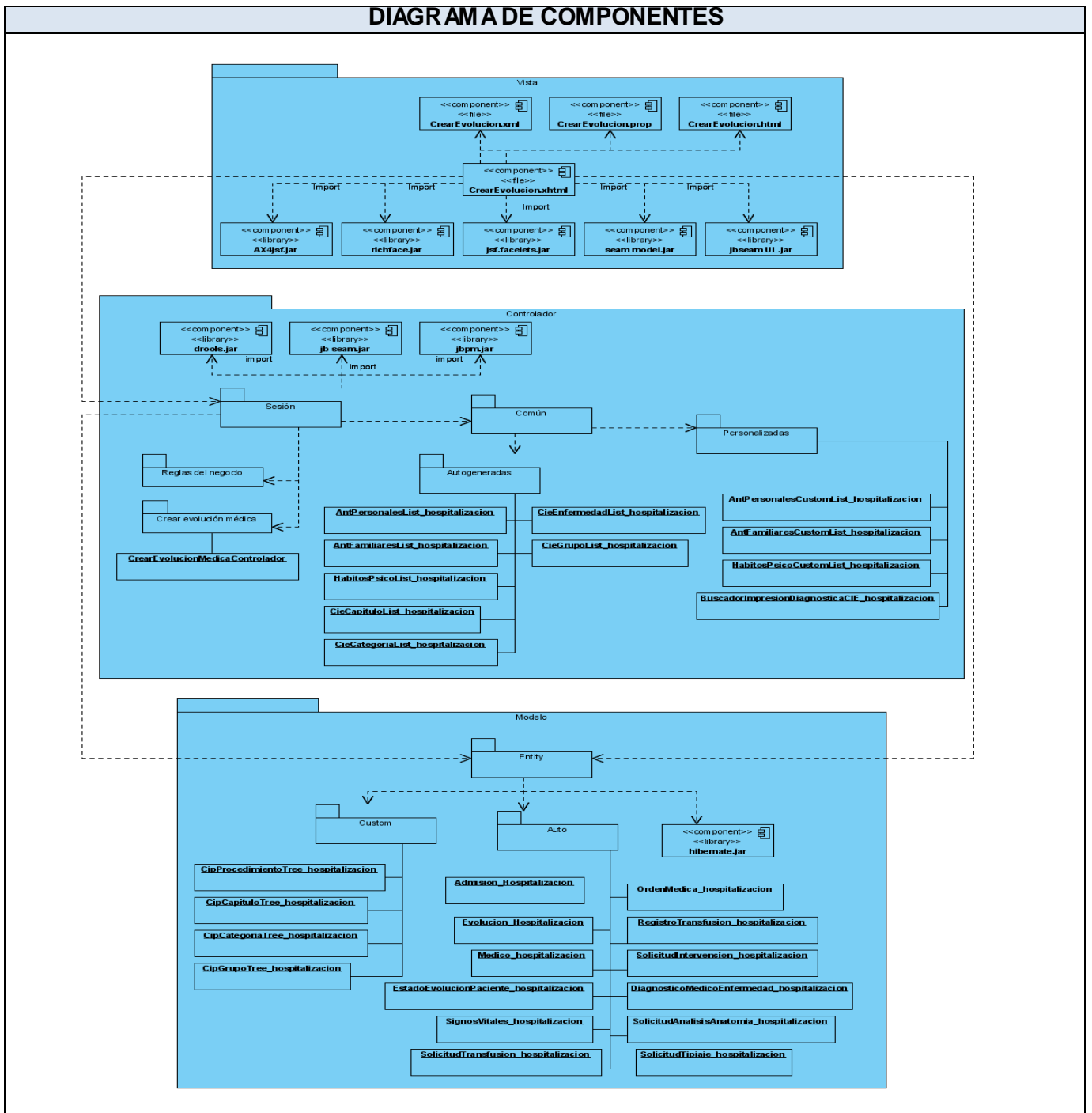


Figura 4.3 Diagrama de componentes CU Crear evolución médica.

4.2.2 Diagrama de Despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Por esta razón, tal distribución tiene una influencia principal para las actividades de implementación. Cada nodo representa un recurso de cómputo.

El módulo de Hospitalización contará con dos servidores, uno de aplicaciones (web) y otro de base de datos, conectados entre sí mediante el Protocolo de Control de Transmisión/ Protocolo de Internet (TCP/IP, por sus siglas en inglés). Además contará con las estaciones clientes que se comunicarán con el servidor web mediante el Protocolo de Transferencia de Híper Texto (HTTP, por sus siglas en inglés). A continuación se muestra el Diagrama de Despliegue que evidencia las características de software y de hardware de cada uno de los nodos físicos.

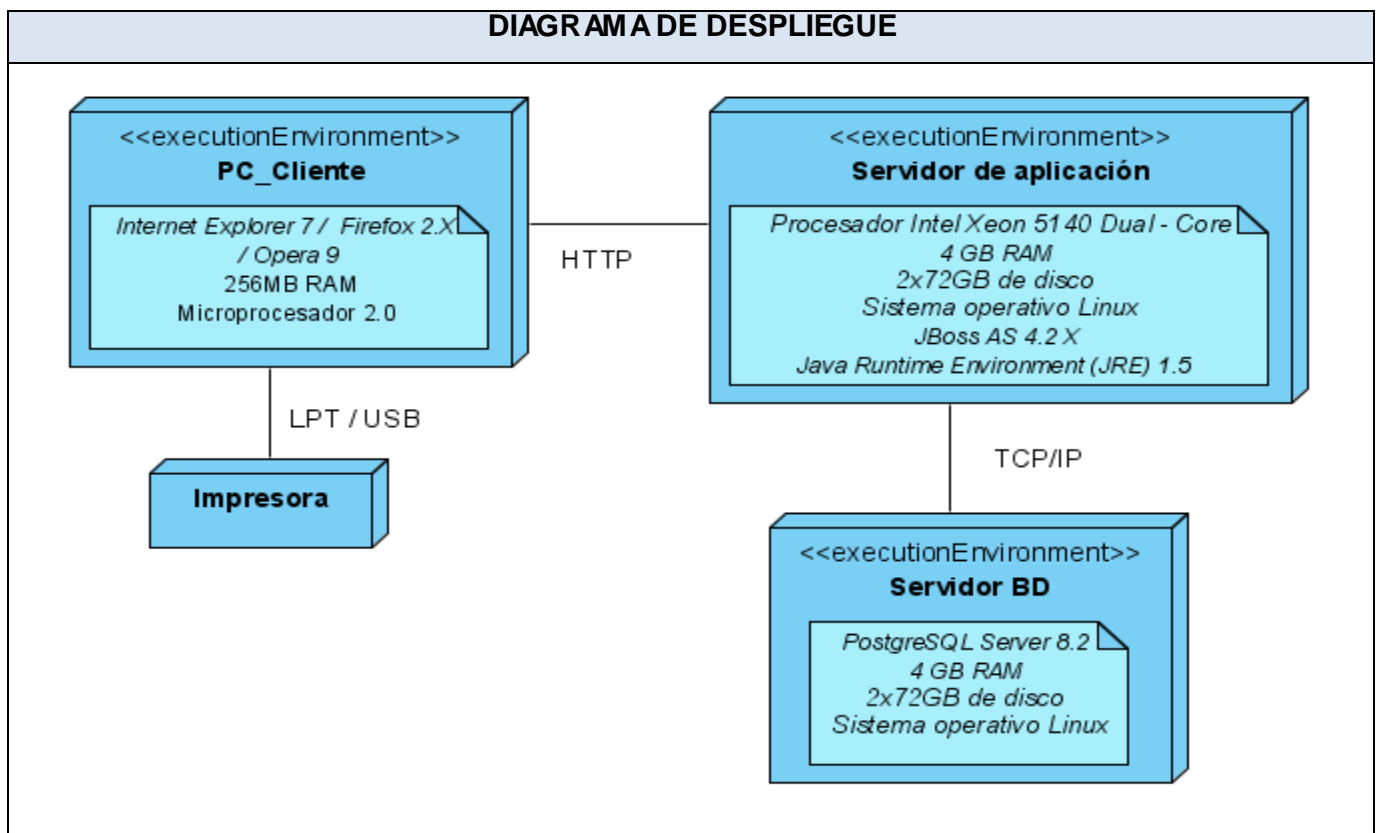


Figura 4.4 Diagrama de despliegue

4.3 Tratamiento de excepciones

Una excepción es un evento que ocurre durante la ejecución de un programa interrumpiendo el flujo normal de las sentencias. Dicho evento puede ser desde serios problemas de hardware, como la avería de un disco duro, hasta los simples errores de programación y pueden ser tratados mediante una estructura de control que poseen los lenguajes de programación de alto nivel, diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla. A esta estructura de control se le conoce como tratamiento de excepciones.

Un buen tratamiento de excepciones garantiza un aumento considerable en la calidad de las aplicaciones que se desarrollen. El lenguaje de programación que se utiliza para el desarrollo del módulo, proporciona una solución elegante al problema, lo que permite escribir el flujo principal de su código y tratar los casos excepcionales en otro lugar, como se muestra en la Figura 4.4. Con esta separación se diferencia la conducta a seguir en dependencia del tipo de excepción ocurrida.

EJEMPLO DE TRATAMIENTO DE EXCEPCIONES
<pre>public static void main(String[] args) { try { // Se ejecuta el flujo principal del código, que puede producir una // excepción. } catch (IOException e) { // Tratamiento de una excepción de entrada/salida. } catch (Exception e) { // Tratamiento de una excepción cualquiera. } finally { // Código a ejecutar haya ocurrido o no una excepción. } }</pre>

Figura 4.5 *Tratamiento de excepciones con el lenguaje de programación Java*

En el módulo se realiza el tratamiento de excepciones de persistencia de datos, de bloqueo optimista, así como de validación y conversión de valores de entrada proporcionados por el usuario. Para ello utiliza controles de conversión y validación que proveen algunas de las tecnologías que se utilizan en el desarrollo; como el validador que proporciona el framework Hibernate, el cual implementa la validación de datos multicapa, donde las restricciones y constantes de validación son expresadas mediante anotaciones en el Modelo (Entidades) y chequeadas en distintas capas de la aplicación. Con

JBoss Seam y JSF se puede iniciar el proceso de validación desde la capa de presentación de la aplicación, permitiendo que las constantes sean expresadas y procesadas en el Modelo y el Controlador, y que los mensajes correspondientes a cada excepción capturada sean presentados en la Vista.

Para garantizar la utilización óptima de recursos y aprovechar las capacidades de internacionalización que brinda JSF, los mensajes de respuesta a cada una de las excepciones son definidos en archivos o ficheros de propiedades para cada una de los idiomas que soportará el sistema, lo cual permite una mejor comunicación con el usuario, sin tener que crear versiones de la aplicación para cada idioma.

4.4 Seguridad

La seguridad es un tema de gran importancia para cualquier Sistema de Información y toma mayor relevancia cuando se gestiona información relacionada con la salud de las personas. Para garantizar la seguridad del Módulo Hospitalización toda la autorización está basada en reglas que restringen el acceso a directorios, páginas, controles, opciones del menú y servicios del negocio. Ninguna de estas reglas del negocio está embebida en el código de la aplicación, posibilitando que en caso de ocurrir un cambio en alguna de estas, no se requiera el cambio ni la respectiva recompilación del código, sino solo en la definición de la propia regla en su fichero de configuración. Esto puede llevarse a cabo por la posibilidad de integración con el motor de reglas JBoss Rules, que brinda el Framework de Seguridad de JBoss Seam.

Se realiza además el control a nivel de usuarios y contraseñas, garantizando el acceso sólo a los niveles establecidos de acuerdo a la función que realiza cada usuario. Las contraseñas sólo pueden ser cambiadas por el propio usuario o por el administrador del sistema. Todas las acciones realizadas por los usuarios quedan registradas a cada momento en una especie de bitácora, almacenándose la fecha, la hora, el usuario y la actividad que realizó.

4.5 Estrategias de codificación. Estándares y estilos

Las estrategias de codificación son muy importantes para la implementación de cualquier producto de software, en especial para las personas que los desarrollan. Un producto de software puede estar más del 80% de su vida útil en mantenimiento y en la mayoría de los casos, difícilmente dicho

mantenimiento es llevado a cabo por su autor original. Las estrategias de codificación hacen al software más legible, permitiendo que los desarrolladores puedan entender su código a fondo y más rápido.

Como estrategia de codificación para el desarrollo del sistema se utiliza el estándar de codificación del lenguaje de programación Java (Java Code Conventions), definido en las especificaciones del propio lenguaje. Todo el código se escribirá en idioma español. Para el nombre de las clases se utilizará la notación o estilo Pascal Casing, con la cual los nombres quedan compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Los identificadores de los atributos y métodos de la clase se escribirán en notación Camel Casing: nombres compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula.

En el capítulo se ha concluido el proceso de implementación y se obtuvieron los artefactos que conforman el Modelo de Implementación, Diagrama de Despliegue y Componentes. También se desarrolló un sistema con la totalidad de las funcionalidades previstas para el correcto funcionamiento del proceso de atención al paciente del módulo de Hospitalización. Esto facilita la gestión de información en el área de hospitalización de las instituciones hospitalarias y beneficia directamente a los pacientes atendidos y al personal que allí labora.

CONCLUSIONES

- El proceso de gestión de información en el área de Hospitalización de las instituciones hospitalarias es complejo y los sistemas existentes que lo informatizan no responden a las necesidades del personal que allí labora.
- La guía del Proceso Unificado de Desarrollo es muy importante para la comprensión del negocio, el levantamiento de requerimientos, y la correcta realización del diseño e implementación de cualquier sistema de información.
- El almacenamiento de toda la información relacionada con los procedimientos médicos, pruebas diagnósticas, evoluciones y movimientos hospitalarios realizados al paciente en una Historia Clínica Electrónica única contribuye a garantizar una mayor calidad y confiabilidad de la misma.
- La utilización del patrón de arquitectura MVC, conjuntamente con el empleo de la administración de procesos y reglas del negocio permitió obtener un sistema flexible, robusto y con alta capacidad de realizar modificaciones en tiempo de desarrollo.

RECOMENDACIONES

- Mejorar el diseño de la Base de Datos del sistema para lograr una mayor optimización del almacenamiento y consulta de los datos correspondientes a las distintas hojas de la Historia Clínica.
- Realizar un levantamiento de información que posibilite la incorporación al sistema de las hojas de la Historia Clínica correspondiente a los servicios médicos que no están comprendidos en el alcance de la investigación.
- Realizar una investigación sobre la posibilidad de hacer extensivo el sistema para su utilización en dispositivos móviles.
- Continuar el estudio de los procesos de negocio identificados en el área de hospitalización con el objetivo de hallar posibles nuevas mejoras e incorporar al módulo los procesos relacionados con los movimientos hospitalarios y la administración del área.

REFERENCIAS BIBLIOGRÁFICAS

[1] Bommel JH Van. Medical Informatics. Art or Science? Meth Inform Me 1996; 35: 157-172.

[2] Mandirola Brioux Humberto Fernán. La crisis en el Área Salud en la era de la información. Buenos Aires: BIOCUM. Disponible en:

http://www.biocom.com/informatica_medica/La%20crisis%20en%20la%20%20era%20de%20la%20informaci%C3%B3n%20en%20el%20Area%20Salud.pdf

[3] Sistema de Información Hospitalaria. México D.F.: Universidad Autónoma de México. D. R. Facultad de Medicina, 2003. p 7. Disponible en:

<http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>

[4] Tamayo Canillas Ma. Luisa. Admisión y gestión de pacientes en la atención especializada. Gijón, España: Servicio de Admisión y Documentación Clínica Hospital de Cabueñes.

[5] Marín Díaz, Miguel E. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. Cuba. 2006. p 3.

[6] Ramírez Márquez, Abelardo Dr.; Castell-Florit Serrate, Pastor Dr.; Mesa, Guillermo Dr. El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP). La Habana, Cuba. 2003. Disponible en:

http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc

[7] Sistema de Información Hospitalaria. México D.F.: Universidad Autónoma de México. D. R. Facultad de Medicina, 2003. p 7. Disponible en:

<http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>

[8] Propuesta para la informatización de Centros de Atención Médica. Softel. La Habana, Cuba. 2007. p 4 Disponible en: <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>

[9] IH-SW-DE-024 ALAS-HIS_Consulta Externa y Citas_Glosario de términos. ALBET. La Habana, Cuba. 2008.

- [10] La OMS utiliza Internet para revisar la clasificación internacional de enfermedades. OMS. 2007. Disponible en: <http://www.who.int/mediacentre/news/releases/2007/pr13/es/>
- [11] Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Parte 1. La Habana, Cuba. Editorial Félix Varela. 2005
- [12] Reynoso, Carlos Billy. Introducción a la Arquitectura de Software. Universidad de Buenos Aires. Versión 1.0. Junio 2006. Disponible en:
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp#E1E
- [13] Documento de arquitectura de Software alas HIS. ALBET. 2008. p 7.
- [14] Gudiño Fleites, Pedro. Tutorial de Sistemas Distribuidos I. Departamento de Sistemas y Computación. Instituto Tecnológico de Colima. México. 2004. Disponible en:
http://www.itcolima.edu.mx/profesores/tutoriales/sistemas_distribuidos_l/sd_u1_1.htm
- [15] Márquez Avendaño, B. M., Zulaica Rugarcía. Implementación de un reconocedor de voz gratuito al sistema de ayuda a invidentes Dos-Vox en español. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Enero 2004. Disponible en:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/
- [16] Departamento de Lenguajes y Sistemas Informáticos. Tema 1: Patrones Arquitectónicos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Disponible en:
<http://www.lsi.us.es/docencia/get.php?id=1891>
- [17] Pattern-Oriented Software Architecture, Volume 1: A System of Patterns by Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal John Wiley & Son Ltd; ISBN 0471958697 (alternate, search); 1 edition (August 8, 1996)
- [18] Reynoso, Carlos; Kicillof, Nicolás. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Universidad de Buenos Aires. 2004. Disponible en:

http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10

[19] Microsoft patterns and practices. Model-View-Controller. MSDN. 2009. Disponible en:
<http://msdn.microsoft.com/en-us/library/ms978748.aspx>

[20] Richfaces Developer Guide. Red Hat. 2008. p 17. Disponible en:
<http://www.jboss.org/jbossrichfaces/docs/>

[21] Katz Max. Practical Richfaces. Apress. 2008. p 12.

[22] Seam - Contextual Components. Introduction to JBoss Seam. Disponible en:
<http://www.seamframework.org/Documentation>

[23] Allen Dan. Seam in action. Manning Early Access Program. Manning Publications Co. 2008. p 4, p 5, p 6.

[24] Documento de arquitectura de Software alas HIS. ALBET. 2008. p 69.

[25] Documento de arquitectura de Software alas HIS. ALBET. 2008. p 69, p 70.

[26] Java Persistence API FAQ. Sun Microsystems. Disponible en:
<http://java.sun.com/javaee/overview/faq/persistence.jsp>

[27] Bauer Christian, King Gavin. Java Persistence with Hibernate. Manning Publications Co. 2007. p 67.

[28] PostgreSQL About. Disponible en: <http://www.postgresql.org/about>

[29] PostgreSQL Awards. Disponible en: <http://www.postgresql.org/about/awards>

[30] García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. Enero 2000.

- [31] Dr. Juntao Yuan Michael. On the road to simplicity. Disponible en: <http://www.javaworld.com/javaworld/jw-02-2005/jw-0221-jboss4.html>
- [32] JBoss Server Manager Reference Guide. Red Hat 2008. Disponible en: <http://www.jboss.org/jbossas/docs/>
- [33] Canós José H., Letelier Patricio, Penadés M^a Carmen. Metodologías Ágiles en el Desarrollo de Software. DSIC. Universidad Politécnica de Valencia. Enero 2007. Disponible en <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- [34] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [35] Rumbaugh James, Jacobson Ivar, Booch Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Education. 2007.
- [36] Business Process Modeling Notation Specification. Object Management Group. Febrero 2006. p 1. Disponible en: <http://www.omg.org/technology/agreement.htm>
- [37] Owen Martin, Raj Jog. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. p 4. Disponible en: http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
- [38] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [39] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [40] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [41] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [42] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

- [43] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [44] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [45] Navarro Franco, Ángel José. UML en acción. Modelando Aplicaciones Web. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.
- [46] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [47] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [48] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [49] Disponible en: <http://www.hl7spain.org/VerPagina.asp?IDPage=0>

BIBLIOGRAFÍA

- Allen Dan. Seam in action. Manning Early Access Program. Manning Publications Co. 2008.
- Bauer Christian, King Gavin. Java Persistence with Hibernate. Manning Publications Co. 2007.
- Bommel JH Van. Medical Informatics. Art or Science? Meth Inform Me 1996.
- Business Process Modeling Notation Specification. Object Management Group. Febrero 2006. Disponible en: <http://www.omg.org/technology/agreement.htm>
- Canós José H., Letelier Patricio, Penadés M^a Carmen. Metodologías Ágiles en el Desarrollo de Software. DSIC. Universidad Politécnica de Valencia. Enero 2007. Disponible en <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- Conceptos generales de Sistemas Distribuidos. Departamento de Arquitectura de Computadoras. Universidad Politécnica de Catalunya. España. 2007. Disponible en: <http://studies.ac.upc.edu/EPSC/FSD/FSD-ConceptosGenerales.pdf>
- Departamento de Lenguajes y Sistemas Informáticos. Tema 1: Patrones Arquitectónicos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=1891>
- Documento de arquitectura de Software alas HIS. ALBET. La Habana, Cuba. 2008.
- Dr. Juntao Yuan Michael. On the road to simplicity. Disponible en: <http://www.javaworld.com/javaworld/jw-02-2005/jw-0221-jboss4.html>
- García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprenda Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. Enero 2000.
- Health Level Seven Spain. Disponible en: <http://www.hl7spain.org/VerPagina.asp?IDPage=0>

- IH-SW-DE-029 ALAS-HIS_Hospitalización_Glosario de términos. ALBET. La Habana, Cuba. 2008.
- IH-SW-DR-063 ALAS-HIS_Hospitalizacion_Especificacion de requisitos. La Habana, Cuba. 2008.
- IH-SW-DR-078 ALAS-HIS_Hospitalizacion_Modelo de casos de uso del sistema. La Habana, Cuba. 2008.
- IH-SW-DR-030 ALAS-HIS_Hospitalización_Procesos Elementales del Negocio_Procesos actuales. La Habana, Cuba. 2008.
- IH-SW-DE-046 ALAS-HIS_Hospitalización_Glosario de entidades. La Habana, Cuba. 2008.
- IH-SW-DE-024 ALAS-HIS_Consulta Externa y Citas_Glosario de términos. ALBET. La Habana, Cuba. 2008.
- Java Code Conventions. Sun Microsystems. California, Estados Unidos de América. 1997.
- Java Persistence API FAQ. Sun Microsystems. Disponible en:
<http://java.sun.com/javase/overview/faq/persistence.jsp>
- JBoss Server Manager Reference Guide. Red Hat 2008. Disponible en:
<http://www.jboss.org/jbossas/docs/>
- Katz Max. Practical Richfaces. Apress. 2008.
- La OMS utiliza Internet para revisar la clasificación internacional de enfermedades. OMS. 2007. Disponible en: <http://www.who.int/mediacentre/news/releases/2007/pr13/es/>
- Marín Díaz, Miguel E. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. Cuba. 2006.
- Márquez Avendaño, B. M., Zulaica Rugarcía. Implementación de un reconocedor de voz gratuito al sistema de ayuda a invidentes Dos-Vox en español. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de

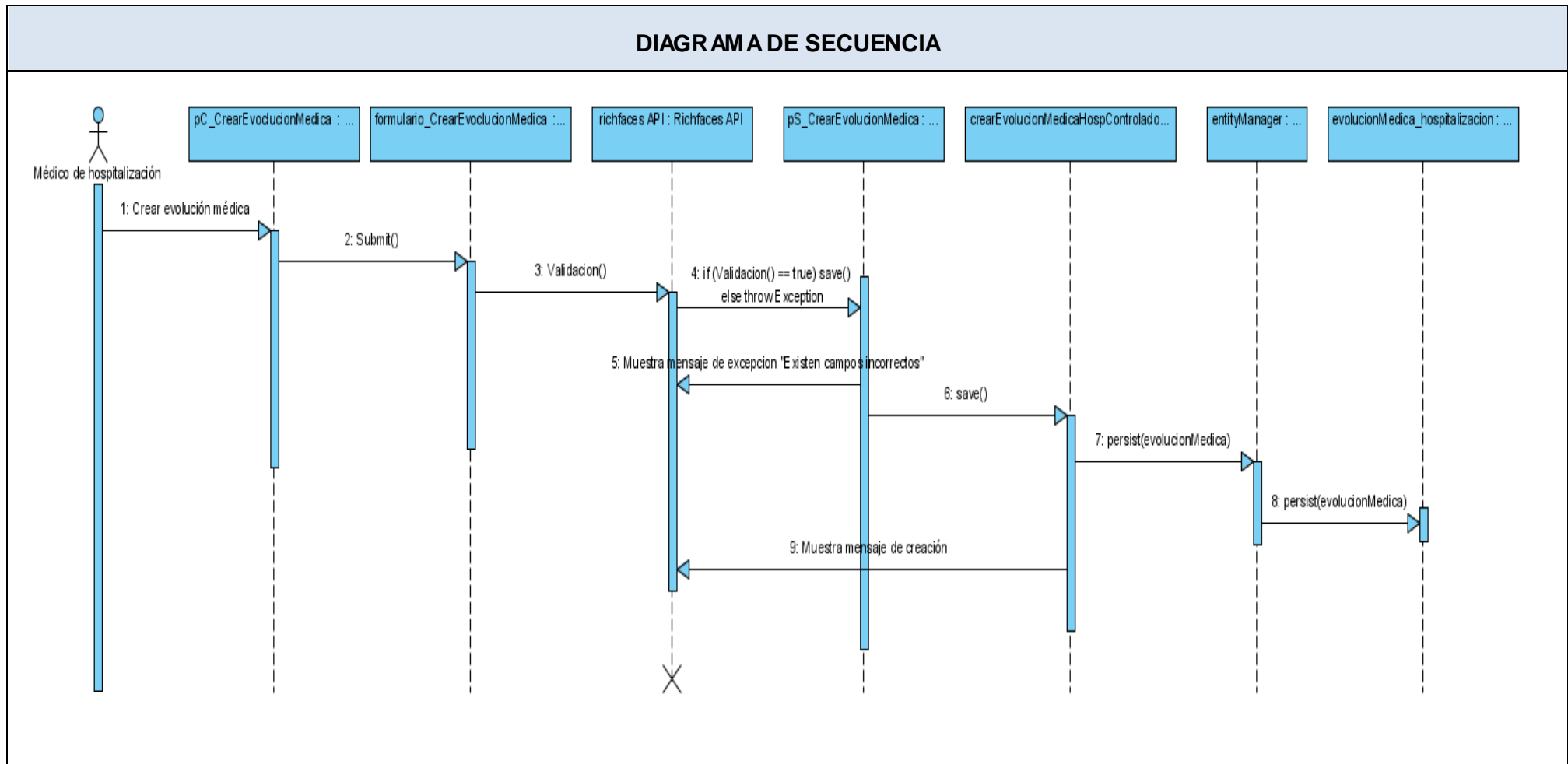
Ingeniería, Universidad de las Américas Puebla. Enero 2004. Disponible en:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/

- Mandirola Brieux Humberto Fernán. La crisis en el Área Salud en la era de la información. Buenos Aires: BIOCUM. Disponible en:
http://www.biocom.com/informatica_medica/La%20crisis%20en%20la%20era%20de%20la%20informaci%C3%B3n%20en%20el%20Area%20Salud.pdf
- Microsoft patterns and practices. Model-View-Controller. MSDN. 2009. Disponible en:
<http://msdn.microsoft.com/en-us/library/ms978748.aspx>
- Navarro Franco, Ángel José. UML en acción. Modelando Aplicaciones Web. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.
- Owen Martin, Raj Jog. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. Disponible en:
http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
- Pattern-Oriented Software Architecture, Volume 1: A System of Patterns by Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal John Wiley & Son Ltd; ISBN 0471958697 (alternate, search); 1 edition (August 8, 1996)
- Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Parte 1. La Habana, Cuba. Editorial Félix Varela. 2005
- PostgreSQL About. Disponible en: <http://www.postgresql.org/about>
- PostgreSQL Awards. Disponible en: <http://www.postgresql.org/about/awards>
- Propuesta para la informatización de Centros de Atención Médica. Softel. La Habana, Cuba. 2007. Disponible en: <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>

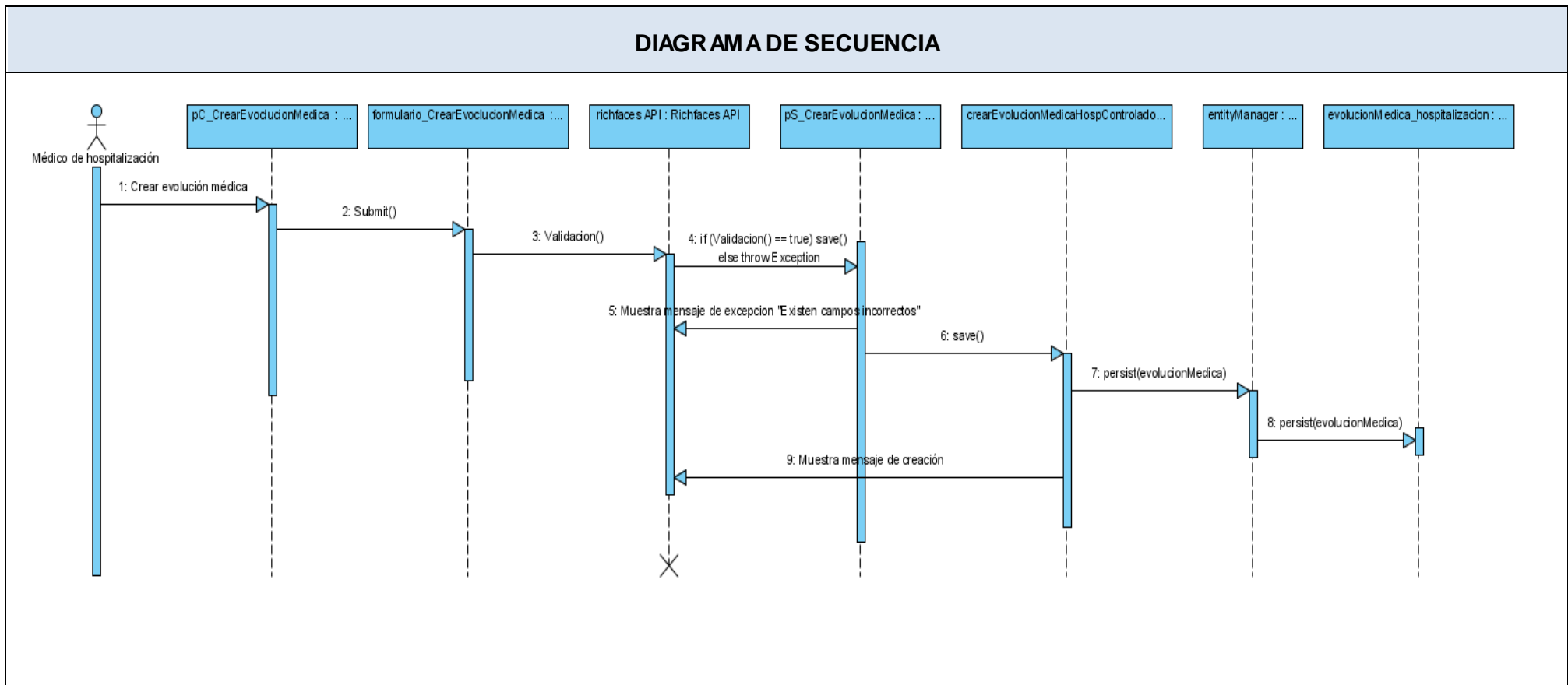
- Ramírez Márquez, Abelardo Dr.; Castell-Florit Serrate, Pastor Dr.; Mesa, Guillermo Dr. El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP). La Habana, Cuba. 2003. Disponible en: http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc
- Richfaces Developer Guide. Red Hat. 2008. Disponible en: <http://www.jboss.org/jbossrichfaces/docs/>
- Reynoso, Carlos Billy. Introducción a la Arquitectura de Software. Universidad de Buenos Aires. Versión 1.0. Junio 2006. Disponible en: <http://www.willydev.net/descargas/prev/IntroArq.pdf>
- Reynoso, Carlos; Kicillof, Nicolás. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Universidad de Buenos Aires. 2004. Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/estiloypatron.pdf>
- Rumbaugh James, Jacobson Ivar, Booch Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Education. 2007.
- Seam - Contextual Components. Introduction to JBoss Seam. Disponible en: <http://www.seamframework.org/Documentation>
- Sistema de Información Hospitalaria. México D.F.: Universidad Autónoma de México. D. R. Facultad de Medicina, 2003. Disponible en: <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>
- Tamayo Canillas Ma. Luisa. Admisión y gestión de pacientes en la atención especializada. Gijón, España: Servicio de Admisión y Documentación Clínica Hospital de Cabueñes.

ANEXOS

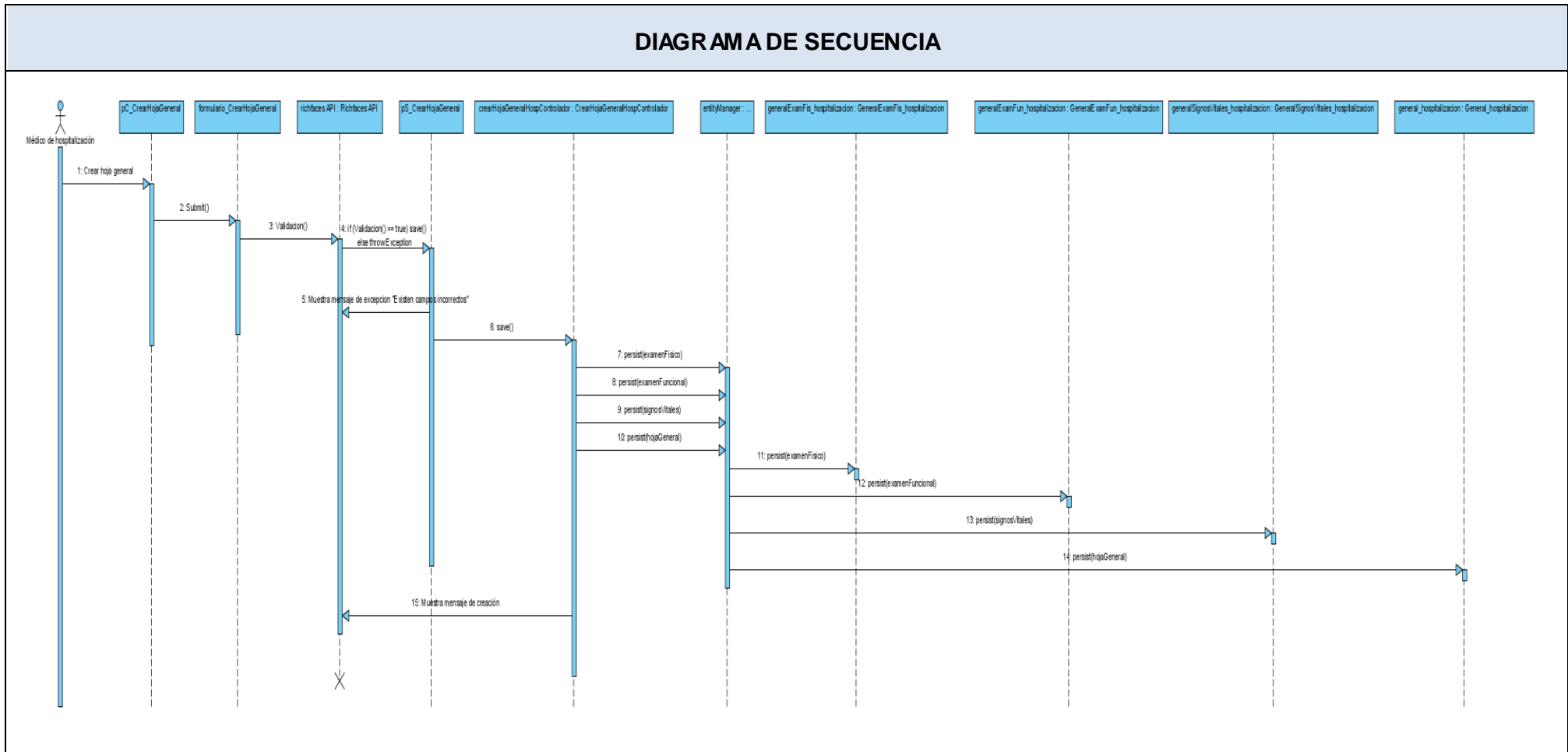
Anexo 1: Diagrama de secuencia del caso de uso Crear evolución médica



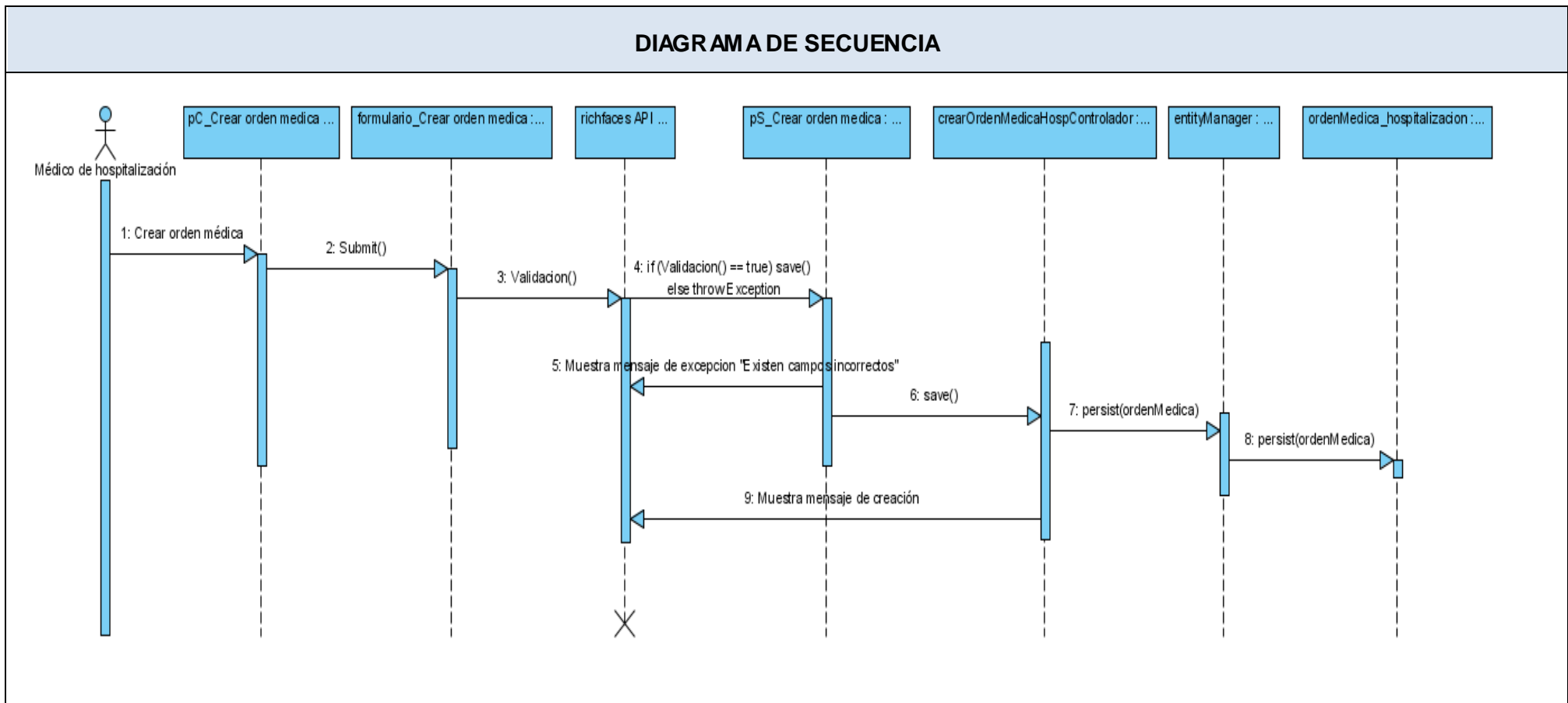
Anexo 2: Diagrama de secuencia del caso de uso Crear evolución médica



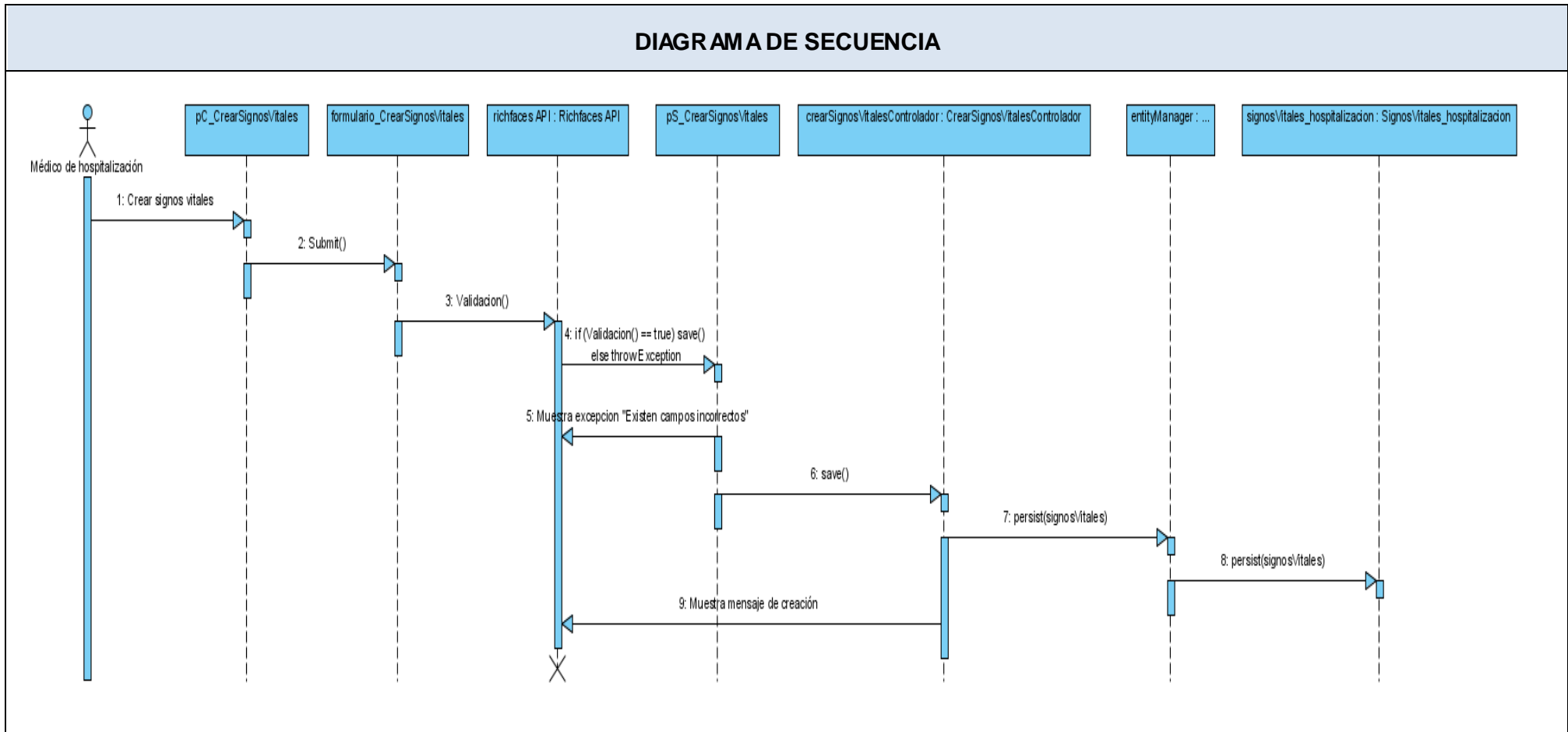
Anexo 3: Diagrama de secuencia del caso de uso Crear hoja general de hospitalización



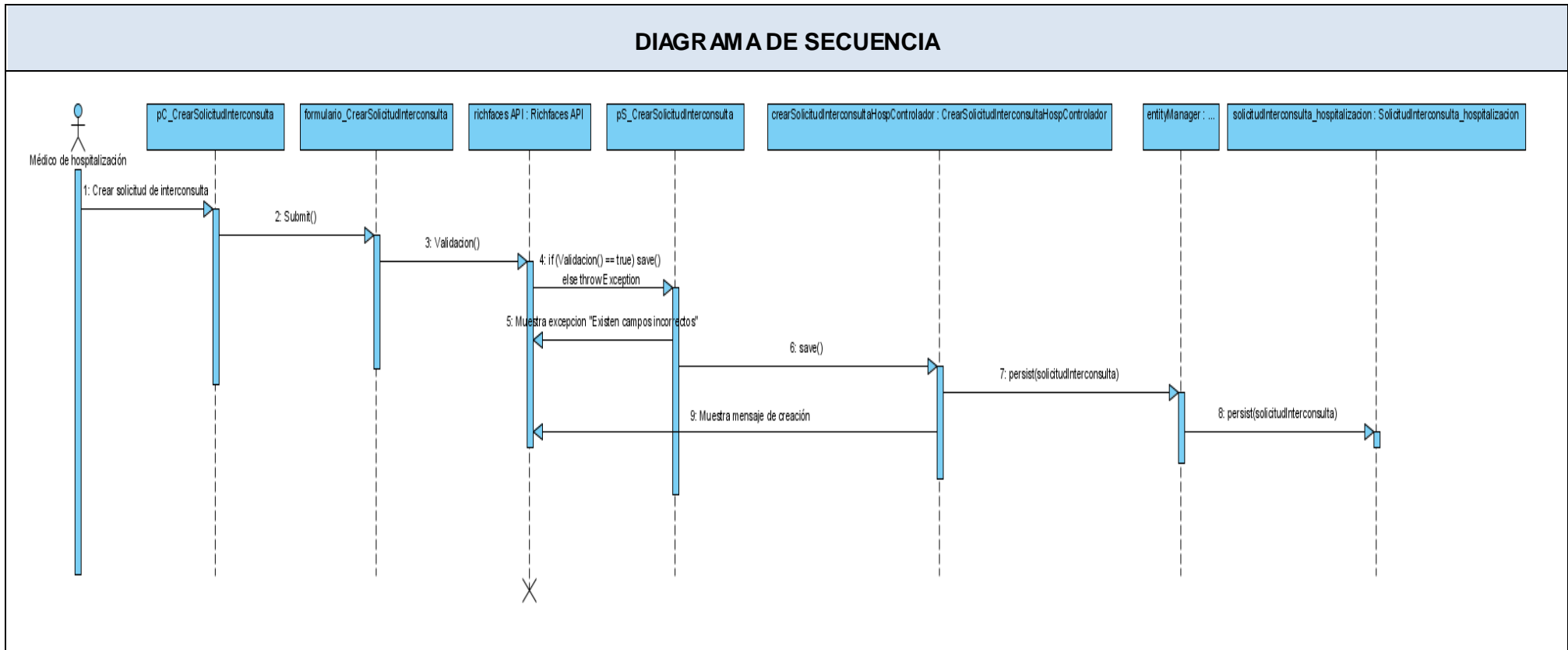
Anexo 4: Diagrama de secuencia del caso de uso Crear orden médica



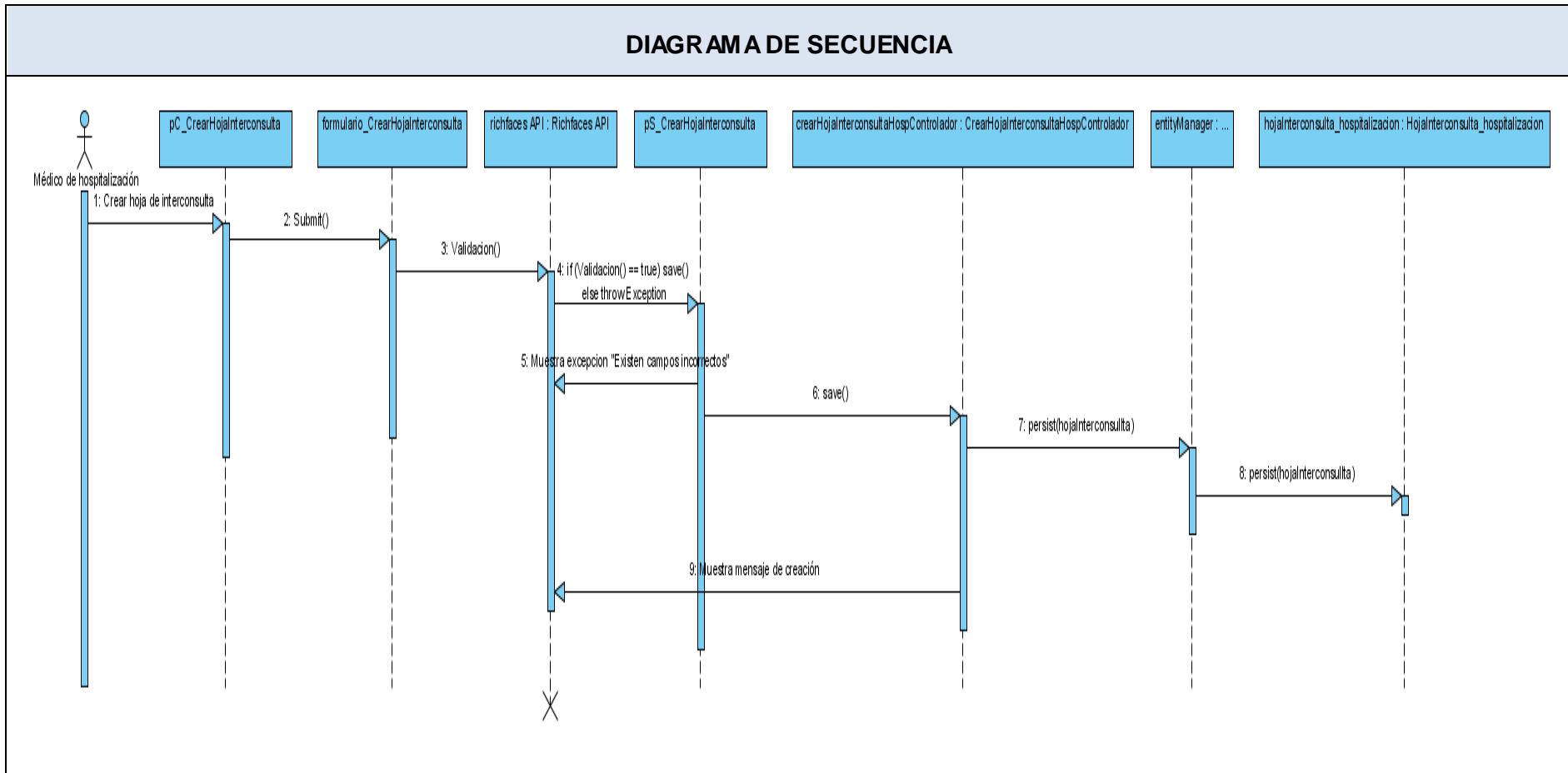
Anexo 5: Diagrama de secuencia del caso de uso Crear signos vitales



Anexo 6: Diagrama de secuencia del caso de uso Crear solicitud de interconsulta




Anexo 7: Diagrama de secuencia del caso de uso Crear hoja de interconsulta




Anexo 8: Interfaz caso de uso Listar pacientes


Listar pacientes


Datos generales del paciente No.H.C.:148523


	Nombre:	Reisel	Cédula:	89010632985
	Primer Apellido:	De la Rosa	Fecha de nacimiento:	06/01/1989
	Segundo apellido:	Ge	Sexo:	Masculino


Servicio: Traumatología Ver por: [Ver opciones](#)



S3H2-C2-Z



S3H2-C4-C



S3H2-C3-G



S3H2-C1-H



S3H2-C1-F


S3H2-C4-B



S3H2-C2-I


S3H2-C2-J


S3H2-C3-E


S3H2-C1-D

Anexo 9: Interfaz caso de uso Crear evolución médica

Crear evolución médica		Buscar...
		Fecha: 29/04/2009
Datos generales del paciente		No.H.C.:148523
	Nombre: Reisel	Cédula: 89010632985
	Primer apellido: De la Rosa	Fecha de nacimiento: 06/01/1989
	Segundo apellido: Ge	Sexo: Masculino
Ver opciones		Consultar acciones realizadas hasta el momento
Antecedentes personales		»
Antecedentes familiares		»
Hábitos psicobiológicos		»
Antecedentes prenatales y obstétricos		»
Período neonatal		»
Alimentación		»
Desarrollo		»
Inmunizaciones		»
Datos de la evolución		
Estado del paciente:		
seleccione <input type="button" value="v"/>		
Observaciones:		
<div style="border: 1px solid gray; height: 40px;"></div>		
Impresión diagnóstica		»
Diagnóstico final		»
<input type="button" value="Aceptar"/>		<input type="button" value="Cancelar"/>

Anexo 10: Interfaz caso de uso Seleccionar enfermedad

Datos de la evolución

Estado del paciente:
 ▼

Observaciones:

Impresión diagnóstica >>

Diagnóstico final <<

Código: Descripción:


[Búsqueda avanzada](#)

<div style="background-color: #e6f2ff; padding: 2px;">CIE</div> <ul style="list-style-type: none"> ▶ Grupos (1) ▶ Categorías (6) ▶ Subcategorías (51) ▼ Enfermedades (492) <ul style="list-style-type: none"> 📄 C50.0a - Tumor maligno del pezón 📄 N02.0a - Hematuria recurrente y persistente por 📄 A16.9a - Tuberculosis SAI 📄 C34.0a - Tumor maligno de carina 📄 A02.2+b - Enfermedad renal tubulointersticial d 📄 A05.1a - Intoxicación alimentaria clásica debida 📄 A08.1a - Enteritis debida a virus pequeño de es 📄 A15.0a - Bronquiectasia tuberculosa confirmada 	<input type="button" value=">>"/>	<div style="background-color: #e6f2ff; padding: 2px;">Diagnóstico</div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">C34.0a</td> <td style="padding: 2px;">Tumor maligno de carina</td> <td style="text-align: right; padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">A08.1a</td> <td style="padding: 2px;">Enteritis debida a virus pequeño de estructura redonda</td> <td style="text-align: right; padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">A15.0a</td> <td style="padding: 2px;">Bronquiectasia tuberculosa confirmada por hallazgo microscópico del bacilo tuberculoso en esputo, con o sin cultivo</td> <td style="text-align: right; padding: 2px;"></td> </tr> </table>	C34.0a	Tumor maligno de carina		A08.1a	Enteritis debida a virus pequeño de estructura redonda		A15.0a	Bronquiectasia tuberculosa confirmada por hallazgo microscópico del bacilo tuberculoso en esputo, con o sin cultivo	
C34.0a	Tumor maligno de carina										
A08.1a	Enteritis debida a virus pequeño de estructura redonda										
A15.0a	Bronquiectasia tuberculosa confirmada por hallazgo microscópico del bacilo tuberculoso en esputo, con o sin cultivo										

Anexo 11: Interfaz caso de uso Crear orden médica

Crear orden médica Buscar...

Datos generales del paciente No.H.C.:148523



Nombre:	Reisel	Cédula:	89010632985
Primer apellido:	De la Rosa	Fecha de nacimiento:	06/01/1989
Segundo apellido:	Ge	Sexo:	Masculino

Ubicación

Servicio: Traumatología Habitación: AAA Cama: C

Datos de la orden médica

Observaciones:

Indicación de medicamentos «

Medicamento	Dosis	Vía	Horario	Fecha de suspensión		
DICIGEL SUSPENSIN	1	Oral ▼	12 provisional ▼	30/04/2009		

Medicamentos seleccionados como opcionales

DICIGEL SUSPENSIN »

Adicionar

Orden de dieta »


Solicitud de nutrición parenteral »

Solicitud de preparación intravenosa »

Indicación de otros tratamientos »

Aceptar
Cancelar


Anexo 12: Interfaz caso de uso Crear solicitud de transfusión

Crear solicitud de transfusión		Buscar...
Datos generales del paciente		No.H.C.:148523
	Nombre: Reisel	Cédula: 89010632985
	Primer apellido: De la Rosa	Fecha de nacimiento: 06/01/1989
	Segundo apellido: Ge	Sexo: Masculino
Datos de la transfusión		
Prioridad Urgente ▼	Componente: Concentrado globular ▼	Cantidad: 2
Hemoglobina: 13.5		
Motivo: Hemorragia		
		Aceptar Cancelar

Anexo 13: Interfaz caso de uso Crear solicitud de análisis de laboratorio

Crear solicitud de análisis de laboratorio para pacientes internos Buscar...

Datos generales del paciente No.H.C.:148523



Nombre: Reisel Cédula: 89010632985

Primer apellido: De la Rosa Fecha de nacimiento: 06/01/1989

Segundo apellido: Ge Sexo: Masculino

Datos de la solicitud

Fecha de emisión: 17/11/2008 Secciones

Servicio solicitante:
Traumatología
Secciones

Inmunología
 Hematología
 Bioquímica
 Bacteriología

▶▶
▶
◀
◀◀

Listado de exámenes seleccionados

Examen	Sección	Urgente	Antibiograma	
ggg	Bioquímica	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▶
Urea	Bioquímica	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▶
Alcapone	Bioquímica	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▶
HIV	Inmunología	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▶
PTT	Hematología	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▶

Listado de exámenes

	Examen	Sección
◀	Creatinina	Bacteriología
◀	Cultivo	Bacteriología
◀	Chagas	Bacteriología
◀	Lala	Bacteriología
◀	dsvvdafsbv	Bacteriología

Aspectos de la sección de Bioquímica

»»

Aspectos de la sección de Inmunología

»»

Aspectos de la sección de Hematología

»»

Aceptar
Cancelar


Anexo 14: Interfaz caso de uso Crear solicitud de intervención quirúrgica

Solicitar intervención quirúrgica Q Buscar...

Datos generales del paciente No.H.C.:148523

Nombre: Reisel Cédula: 89010632985

Primer apellido: De la Rosa Fecha de nacimiento: 06/01/1989

Segundo apellido: Ge Sexo: Masculino

Datos de la solicitud

Servicio Quirúrgico: Traumatología Duración aproximada (horas): hrs. min. Emergencia:

<< < mayo, 2009 > >>

lun	mar	mié	jue	vie	sáb	dom	
18	27	28	29	30	1	2	3
19	4	5	6	7	8	9	10
20	11	12	13	14	15	16	17
21	18	19	20	21	22	23	24
22	25	26	27	28	29	30	31
23	1	2	3	4	5	6	7

21/05/2009 Clean Today

Procedimientos quirúrgicos solicitados para el 21/05/2009

Procedimiento	Duración aproximada (horas)
Puncion cisternal	2

Diagnóstico >>

Procedimiento quirúrgico >>

Cirujanos

Ayudantes

Equipos especiales

Dispositivos

Listado de Cirujanos

Nombre	Primer apellido	Segundo apellido	
Raúl Ferrer Hoz			+
Reinel Muñoz Perez	Muñoz	Pérez	+

Cirujanos seleccionados

Nombre	Primer apellido	Segundo apellido	
Reinel Muñoz Perez	Muñoz	Pérez	-

Aceptar

Cancelar

GLOSARIO DE TÉRMINOS

Egreso: movimiento hospitalario que consiste en dar salida al paciente del hospital.

Framework: es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

HL7: es una organización internacional, iniciada en los Estados Unidos en 1987, que pretende promover el desarrollo y evolución del estándar HL7 (Health Level Seven) para el formato de datos e intercambio de información entre diferentes Sistemas de Información de Salud. [49]

IEEE: corresponde a las siglas de The Institute of Electrical and Electronics Engineers, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización.

LPT: Terminal de conexión entre una computadora y la impresora u otro dispositivo.

Pruebas diagnósticas: aquellas pruebas o exámenes que brindan un resultado que permiten la emisión de un diagnóstico médico.

Servicios de apoyo: son servicios especializados que brindan los medios de diagnóstico para apoyar la determinación o corroboración del diagnóstico médico.

Tipiaje: examen utilizado para determinar el grupo sanguíneo y el factor Rh de una persona.

Transferencia: movimiento hospitalario que permite el traslado de un paciente hacia otra cama en otro servicio o dentro del mismo.

WSDL: es un lenguaje que está basado en XML y que permite la descripción de los servicios web desplegados. WSDL se utiliza también para la localización y ubicación de estos servicios en Internet.