



Universidad de las Ciencias Informáticas
Facultad 7

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

TÍTULO: MÓDULO ENFERMERÍA DEL SISTEMA DE
INFORMACIÓN HOSPITALARIA alas HIS

Autores: Elisa Sierra Prego
Gustavo Luis Fong Zurbano

Tutor: Ing. Mainoldis Fuentes Suárez

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 firmo la presente a los 24 días del mes de junio del año 2009.

Elisa Sierra Prego
Autora

Gustavo Luis Fong Zurbano
Autor

Ing. Mainoldis Fuentes Suárez
Tutor

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

DATOS DE CONTACTO

Tutor

Ing. Mainoldis Fuentes Suárez: Instructor recién graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad # 7. Ha impartido las asignaturas Introducción a la Programación, Programación I y II e Inteligencia Artificial. Se ha desempeñado como líder de módulo en el proyecto de gestión hospitalaria y como especialista principal de gestión de configuración.

Correo electrónico: mfuentes@uci.cu

AGRADECIMIENTOS

A todos aquellos que de una forma u otra han marcado una huella en nuestras vidas y han permitido la realización de este trabajo: muchísimas gracias de todo corazón.

Agradecerle a nuestros compañeros del proyecto, quienes supieron tendernos una mano amiga cuando la necesitábamos. A Juan Manuel, por las muchas molestias, a Reinel, Yasser, Francisco, Adriel, Aliuska, Carrandi, Omar, en fin a todos los que estuvieron presentes cuando más se necesitaba.

A los profes del proyecto, quienes nos brindaron todo el apoyo que necesitábamos, en especial a nuestro tutor y a las profesoras Nadieżka y Kenia por toda su ayuda.

A la Revolución Cubana y en especial al Comandante en Jefe Fidel Castro por hacer posible este sueño.

De Elisa:

A mis padres por brindarme todo su apoyo durante estos cinco años.

A mi abuelo Andrés quien siempre estuvo atento y muy preocupado por mí.

A mis abuelas queridas, mis tías y primas, especialmente a mi prima Lenabel.

A mis hermanos Elianys y Armín.

A mi compañero de Tesis, quien estuvo a mi lado durante los cinco años aguantándome, gracias por ser tan paciente.

A mis compañeros tanto del proyecto como los que han estado a mi lado durante el transcurso de la carrera.

De Gustavo:

A mis padres, quienes siempre han sabido darme su apoyo incondicional y han estado alentándome y dándome los consejos adecuados en cada momento no solo durante toda la carrera sino también durante mi vida, a mis abuelos queridos, a Yoly por encargarse de todo lo que me hacía falta para que mi estancia en la UCI fuese aun mejor, y bueno a la principal figura que fue la que me ha guiado hasta aquí desde primer año que es mi compañera de tesis, a ella que siempre ha estado para los momentos que todos quieren estar, los buenos y para los que casi nadie nunca está, que son los malos, al igual que su familia.

A todos muchas gracias.

DEDICATORIA

De Elisa:

Primeramente, dedicarle el trabajo con todo mi amor y cariño a mi abuelo, papi gracias por todo tu apoyo y por cuidar tanto de mí.

A mis padres quienes también han estado presentes en los momentos más duros de la carrera y de mi vida.

A mis hermanos, mis tías, especialmente a tía Ely. A mis primas. A toda mi familia por estar siempre ayudándome en todo lo que necesite.

A mis abuelas, que tanto quiero y a mi abuelo Armín que aunque ya no está, siempre espero este momento.

A Gustavo y a sus padres quienes me apoyaron en todo momento.

De Gustavo:

En primer lugar a mis padres que son mi guía en la vida.

A mis abuelos que me han cuidado desde siempre.

A mi compañera de tesis que se merece este trabajo y mucho más...

Y a su familia por acogerme en su seno como un miembro más...

RESUMEN

En las instituciones hospitalarias se gestionan grandes volúmenes de información. Usualmente se realiza de forma manual y en los casos en que se encuentra automatizado, no se satisfacen las necesidades requeridas, además de ser demasiado caros, basados en tecnologías y herramientas propietarias y no brindan una solución estándar que se adecúe a las necesidades del personal que labora en dichas instituciones.

Por estas razones se propone obtener una aplicación informática que facilite la gestión de la información en el área de Enfermería de las Instituciones Hospitalarias, basada en tecnologías libres, multiplataforma y guiada por la metodología de desarrollo Proceso Unificado de Desarrollo (RUP), utilizándose para el modelado el Visual Paradigm como herramienta CASE.

Como lenguaje de programación se utilizará Java, utilizando una arquitectura en capas, aplicándose específicamente el patrón, Modelo Vista Controlador y contando con el IDE de desarrollo Eclipse SDK. Para la gestión de datos se contará, con PostgreSQL como gestor de base de datos y como servidor de aplicaciones Jboss Server.

Con dicha aplicación se espera que el paciente reciba una atención de salud con calidad, al agilizar el trabajo diario de los enfermeros en las instituciones hospitalarias. Mediante el registro de las mediciones de los signos vitales del paciente, así como las notas de evolución para la Historia Clínica Electrónica (HCE) del paciente, se garantiza la seguridad y confiabilidad de la información del mismo. Además que, por cada evolución del paciente se podrá emitir un diagnóstico de enfermería, basado en el estándar para diagnósticos NANDA.

PALABRAS CLAVES: *Enfermería, Historia Clínica, Diagnóstico de Enfermería, NANDA, Indicaciones médicas.*

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Sistema de Salud Pública.....	5
1.2 Sistemas de Información Hospitalaria (HIS).....	6
1.3 Antecedentes	6
1.4 Conceptos básicos asociados al dominio del problema	9
1.5 Tendencias y tecnologías actuales a considerar.....	10
1.6 Lenguaje de programación.....	16
1.7 Servidor de aplicaciones.....	17
1.8 Sistemas gestores de bases de datos.....	17
1.9 Metodologías de desarrollo de software.....	18
1.10 Herramientas	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	22
2.1 Procesos de negocio de enfermería.....	22
2.2 Información manipulada.....	25
2.3 Modelo de Negocio.....	27
2.4 Requerimientos.....	35
CAPÍTULO 3: DISEÑO DEL SISTEMA	52
3.1 Descripción de la arquitectura	52
3.2 Modelo de Diseño.....	53
CAPÍTULO 4: IMPLEMENTACIÓN.....	73
4.1 Modelo de datos	73
4.2 Modelo de implementación.....	89
4.3 Tratamiento de errores.....	93
4.4 Seguridad.....	94
4.5 Estándares de codificación.....	94
CONCLUSIONES.....	98
RECOMENDACIONES.....	99
REFERENCIAS BIBLIOGRÁFICAS.....	100
BIBLIOGRAFÍA.....	105
ANEXOS	108
GLOSARIO.....	120

ÍNDICE DE TABLAS

Tabla 1.1.	Sistemas de información hospitalaria.	6
Tabla 2.1.	Actores involucrados a los procesos.	30
Tabla 2.1.	Descripción del proceso Atender paciente en hospitalización.	31
Tabla 2.2.	Descripción del proceso Atender paciente en consulta externa.	32
Tabla 2.1.	Descripción del proceso Atender paciente en emergencia.	33
Tabla 2.1.	Descripción del proceso Atender paciente en post-operatorio.	34
Tabla 2.2.	Descripción de actores del sistema	40
Tabla 2.3.	Descripción del caso de uso Crear evolución de enfermería	46
Tabla 2.4.	Descripción del caso de uso Buscar evolución de enfermería	46
Tabla 2.5.	Descripción del caso de uso Ver evolución de enfermería	47
Tabla 2.6.	Descripción del caso de uso Modificar evolución de enfermería	47
Tabla 2.7.	Descripción del caso de uso Ver diagnóstico de enfermería	47
Tabla 2.8.	Descripción del caso de uso Buscar diagnóstico de enfermería	48
Tabla 2.9.	Descripción del caso de uso Ver plan de atención de enfermería	48
Tabla 2.10.	Descripción del caso de uso Buscar plan de atención de enfermería	49
Tabla 2.11.	Descripción del caso de uso Crear control de tratamiento	49
Tabla 2.12.	Descripción del caso de uso Buscar control de tratamiento	49
Tabla 2.13.	Descripción del caso de uso Ver control de tratamiento	50
Tabla 2.14.	Descripción del caso de uso Crear signos vitales	50
Tabla 2.15.	Descripción del caso de uso Modificar signos vitales	50
Tabla 3.1.	Descripción de la clase PC_CrearEvoluciónEnfermería	65
Tabla 3.2.	Descripción de la clase PC_BuscarEvoluciónEnfermería	65
Tabla 3.3.	Descripción de la clase PC_VerEvoluciónEnfermería	66
Tabla 3.4.	Descripción de la clase PC_CrearSignos Vitales	66
Tabla 3.5.	Descripción de la clase PC_ModificarSignos Vitales	66
Tabla 3.6.	Descripción de la clase PC_VerSignos Vitales	66
Tabla 3.7.	Descripción de la clase PC_CrearControl Tratamiento	66
Tabla 3.8.	Descripción de la clase PC_VerControl Tratamiento	66
Tabla 3.9.	Descripción de la clase PS_CrearEvoluciónEnfermería	67
Tabla 3.10.	Descripción de la clase PS_VerEvoluciónEnfermería	67
Tabla 3.11.	Descripción de la clase PS_CrearSignos Vitales	67
Tabla 3.12.	Descripción de la clase PS_VerSignos Vitales	67

Tabla 3.13.	Descripción de la clase PS_ ModificarsignosVitales.....	68
Tabla 3.14.	Descripción de la clase PS_ CrearcontrolTratamiento.....	68
Tabla 3.15.	Descripción de la clase PS_ VercontrolTratamiento	68
Tabla 3.16.	Descripción de la clase CrearSignosVitalesControlador.....	69
Tabla 3.17.	Descripción de la clase ModificarSignos VitalesControlador	69
Tabla 3.18.	Descripción de la clase VerSignosVitalesControlador	69
Tabla 3.19.	Descripción de la clase GestionarEvolucionEnfermeriaControlador.....	70
Tabla 3.20.	Descripción de la clase VerEvolucionEnfermeria	71
Tabla 3.21.	Descripción de la clase CrearControlTratamientoEnfermeria.....	71
Tabla 4.1.	Descripción de la tabla SignosVitales.....	74
Tabla 4.2.	Descripción de la tabla SignosVitalesEnfermero	75
Tabla 4.3.	Descripción de la tabla HojaFrontal	77
Tabla 4.4.	Descripción de la tabla PlanAtencionEnfermero.....	77
Tabla 4.5.	Descripción de la tabla IngresoPostoperatorio	78
Tabla 4.6.	Descripción de la tabla DiagnosticoMedico.....	78
Tabla 4.7.	Descripción de la tabla EvolucionEnfermero.....	79
Tabla 4.8.	Descripción de la tabla DiagnosticoEnfermero	79
Tabla 4.9.	Descripción de la tabla Admision.....	80
Tabla 4.10.	Descripción de la tabla SolicitudIntervencion.....	82
Tabla 4.11.	Descripción de la tabla EgresoPostoperatorio	82
Tabla 4.12.	Descripción de la tabla LiquidosVenas.....	83
Tabla 4.13.	Descripción de la tabla LiquidosIngeridosOralmente.....	83
Tabla 4.14.	Descripción de la tabla LiquidosEliminados	84
Tabla 4.15.	Descripción de la tabla BalanceLiquidos	84
Tabla 4.16.	Descripción de la tabla NotaOperatoria	85
Tabla 4.17.	Descripción de la tabla NandaDiagnostico	86
Tabla 4.18.	Descripción de la tabla ControlIndMedicamentos.....	87
Tabla 4.19.	Descripción de la tabla ControlTratamientoMedicamentos	88
Tabla 4.20.	Descripción de la tabla ControlTratamientoObservacion	89
Tabla A 4.1.	Descripción del caso de uso Registrar ingreso del paciente	117
Tabla A 4.2.	Descripción del caso de uso Ver ingreso del paciente	117
Tabla A 4.3.	Descripción de la Tabla VerIngresoPostoperatorioControlador	118
Tabla A 4.4.	Descripción de la Tabla EliminarIngresoPostoperatorioControlador	118
Tabla A 4.5.	Descripción de la Tabla CrearIngresoPostoperatorioControlador	119

ÍNDICE DE FIGURAS

Figura 2.1	Diagrama de procesos del negocio	29
Figura 2.1	Diagrama de proceso Atender paciente en hospitalización	31
Figura 2.2	Diagrama de proceso Atender paciente en consulta externa	32
Figura 2.1	Diagrama de proceso Atender paciente en emergencia	33
Figura 2.2	Diagrama de proceso Atender paciente en post-operatorio.....	34
Figura 2.3	Actores del sistema	41
Figura 2.4	Diagrama de casos de uso Atender paciente en hospitalización.....	42
Figura 2.5	Diagrama de casos de uso Atender paciente en post-operatorio.....	43
Figura 2.6	Diagrama de casos de uso Atender paciente en consulta externa	44
Figura 2.7	Diagrama de casos de uso Casos de uso comunes dentro del módulo.....	45
Figura 3.1	Diagrama de Paquetes	54
Figura 3.2	Diagrama de clases del diseño Gestionar evolución de enfermería	56
Figura 3.3	Diagrama de secuencia escenario Crear evolución de enfermería.....	57
Figura 3.4	Diagrama de secuencia escenario Seleccionar evolución de enfermería	57
Figura 3.5	Diagrama de secuencia escenario Ver evolución de enfermería	58
Figura 3.6	Diagrama de secuencia escenario Modificar evolución de enfermería	58
Figura 3.7	Diagrama de secuencia escenario Ver diagnóstico de enfermería	59
Figura 3.8	Diagrama de secuencia escenario Ver plan de atención de enfermería	59
Figura 3.9	Diagrama de clases del diseño Gestionar signos vitales	60
Figura 3.10	Diagrama de secuencia escenario Crear signos vitales	61
Figura 3.11	Diagrama de secuencia escenario Modificar signos vitales	61
Figura 3.12	Diagrama de secuencia escenario Seleccionar signos vitales	62
Figura 3.13	Diagrama de secuencia escenario Ver signos vitales	62
Figura 3.14	Diagrama de clases del diseño Gestionar control de tratamiento	63
Figura 3.15	Diagrama de secuencia Crear control de tratamiento	64
Figura 3.16	Diagrama de secuencia Seleccionar control de tratamiento	64
Figura 3.17	Diagrama de secuencia ver control de tratamiento	65
Figura 3.18	Subsistemas de implementación	90
Figura 3.19	Subsistemas de implementación por capas.....	91
Figura 3.20	Diagrama de Despliegue	92
Figura A 1.1	Diagrama de clases del diseño Gestionar ingreso de pacientes.	108
Figura A 1.2	Diagrama de secuencia escenario Registrar ingreso de paciente	109

Figura A 1.3	Diagrama de secuencia escenario Buscar ingreso de paciente	110
Figura A 1.4	Diagrama de secuencia escenario Ver ingreso de paciente	110
Figura A 1.5	Diagrama de clases del diseño Gestionar egreso de pacientes	111
Figura A 1.6	Diagrama de secuencia escenario Buscar egreso del paciente.....	112

INTRODUCCIÓN

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) a escala mundial es una realidad y uno de los motores principales de la sociedad actual. La informática representa el punto clave de los sectores sociales, ya sea en el funcionamiento empresarial, al gestionar el comercio, o en la propia educación, con la ayuda de multimedias y otras herramientas que favorecen la enseñanza. Además aporta grandes beneficios como la rapidez en la obtención de resultados, el almacenamiento de grandes cantidades de información, así como el fácil y rápido acceso a la misma. Una sociedad que apareje la informatización junto al desarrollo de las demás esferas y procesos, será más eficaz y competitiva.

Uno de los campos de la actividad humana que más se ha visto beneficiado, con toda esta revolución informática, es el de la salud. La necesidad de mejorar los servicios de planificación sanitaria y los recursos hospitalarios, han hecho necesaria la integración de las nuevas tecnologías en este sector. La informática ha permitido desarrollar herramientas y tecnologías para el servicio hospitalario, con el objetivo de que se gestione el amplio flujo de información; de modo que sea un proceso ágil y eficiente que permita mejorar la calidad de los servicios ofrecidos al paciente.

Para la gestión de la información hospitalaria existen los llamados Sistemas de Información Hospitalaria (más conocidos por sus siglas en inglés, HIS). Su función es la de apoyar las actividades en los niveles operativos, tácticos y estratégicos dentro de un hospital. Para tal efecto se utilizan las computadoras para obtener, almacenar, procesar y comunicar información clínica y administrativa.

Existen diferentes tipos de sistemas de información dentro de un hospital, pero es importante considerar tres grandes grupos:

- Los que prestan apoyo a los servicios de salud.
- Los que soportan el funcionamiento de las diferentes áreas del hospital.
- Los que automáticamente obtienen datos clínicos del paciente.

Además de tener como objetivo fundamental sistematizar y optimizar la gestión de la información referente al paciente. Para una mayor organización de dichos sistemas, estos se dividen en varios módulos, debido a la gran cantidad de especialidades y servicios que se brindan en los hospitales.

Uno de estos departamentos es *enfermería*, donde se maneja una gran cantidad de registros en distintas áreas que poseen una información vital para cualquier evaluación de rendimiento y calidad en el área hospitalaria. Llevar registros de los signos vitales, datos de los pacientes y administración de fármacos e insumos, son datos que habitualmente tienen un control directo en este departamento. [1]

El departamento de enfermería se encuentra distribuido en todas las áreas del hospital. Los enfermeros atienden a los pacientes que se presentan por los diferentes servicios. Como es poco el personal que trabaja en los servicios, el trabajo de atención al paciente se incrementa, lo cual conlleva que la calidad de dicha atención se vea afectada, así como el rendimiento laboral de los trabajadores del departamento.

Si existe un duplicado de registro de cualquier información que provoca un mal control de la validación de datos y por consiguiente una gran cantidad de información confusa, puede influir de manera negativa a la hora de la toma de decisiones. Algo similar ocurre cuando dicha información se pierde por cualquier eventualidad, es entonces cuando se determina que la gestión manual de toda esta información que se gestiona en los hospitales, entorpece en gran medida el trabajo del personal que labora en estas instituciones, afectando la calidad de los servicios brindados al paciente.

Lo anterior puede ratificarse en los procesos que ocurren diariamente en un hospital. Por ejemplo, cuando un paciente es hospitalizado, los enfermeros registran las condiciones en las cuales ha ingresado dicho paciente, en la hoja de evolución del paciente, donde además aparece el diagnóstico emitido por el médico.

En dependencia del estado del paciente, se le miden y registran los signos vitales. Luego se guarda esta información en el libro destinado para esta tarea en cada cambio de turno, lo cual posibilita que el enfermero del turno siguiente no comprenda la caligrafía del documento, por lo que pudiera darle un tratamiento incorrecto al paciente, o en el mejor de los casos volver a realizar la misma operación y comprobar los signos vitales del paciente.

Debido a que la información se gestiona manualmente, en consulta externa no se lleva un correcto control de la cantidad y calidad de insumos, a veces por la pérdida ocasional del libro de reportes, o porque el libro de control de materiales contiene información repetida en diferentes páginas, conllevando a un mal entendimiento de la situación.

Otro ejemplo cuestionable ocurre cuando llega un paciente al cuerpo de guardia y su estado es muy delicado, se procede a otras actividades para salvar su vida y luego se miden sus signos vitales, los

cuales, se registran en la historia clínica del mismo. Este proceso muchas veces retrasa la atención del paciente debido a que se le hace una nueva historia clínica al llegar al hospital y no se le conocen qué estudios previos se le han realizado al mismo, obstaculizando así el propósito de las instituciones sanitarias, que no es otro que brindar una mejor atención.

Por los criterios expresados anteriormente se tiene como **problema a resolver**: ¿Cómo facilitar la gestión de la información relacionada con los procesos en el departamento de enfermería de las instituciones hospitalarias?

Este problema presenta como **objeto de estudio**: Proceso de gestión de información en la atención de salud de las instituciones hospitalarias. El **campo de acción** está centrado en el proceso de gestión de la información en la atención de salud en el departamento de enfermería de las instituciones hospitalarias.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar el módulo de Enfermería del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información de las instituciones hospitalarias.

Para darle cumplimiento al principal objetivo de la investigación se desarrollaron las **tareas** siguientes:

- Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria.
- Analizar los procesos de negocio asociados al Departamento de Enfermería de las instituciones hospitalarias.
- Asimilar la arquitectura definida por el Área Temática de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
- Obtener mediante el Proceso Unificado de Desarrollo, los flujos de trabajo de “Modelo de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
- Implementar los procesos Atender paciente, Atender paciente en post-operatorio y Atender paciente en consulta externa, del módulo de Enfermería.

En este sentido se puede destacar que el desarrollo del módulo de Enfermería, del sistema alas HIS, proporcionará un grupo de beneficios entre los que pueden ser mencionados los siguientes:

1. Los enfermeros podrán consultar las indicaciones médicas e inclusive la Historia Clínica Electrónica (HCE) del paciente y procesar las indicaciones de medicamentos conociendo detalles de cada toma.
2. Se podrá registrar los signos vitales del paciente y notas de evolución para la HCE del paciente.
3. Los enfermeros podrán registrar transferencias internas y el alta del paciente teniendo la posibilidad de marcar la cama, para que la misma quede libre.

El documento se encuentra estructurado de la siguiente forma:

En el **Capítulo 1** se realiza el estudio preliminar de sistemas existentes en el mundo, que gestionan los procesos hospitalarios y dentro de ellos el área de enfermería. Además de las tecnologías y metodologías más utilizadas hasta el momento.

En el **Capítulo 2** se analiza el marco conceptual asociado a la información que será manipulada por el sistema. Se logra un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización. Además de utilizarse como apoyo, la descripción de los procesos del negocio.

En el **Capítulo 3** se tratan aspectos acerca del modelado y construcción, de la estructura de la aplicación.

En el **Capítulo 4** aparece la representación de clases y subsistemas, en términos de componentes y 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

Este capítulo está dedicado a analizar conceptos que se deben conocer para tener un dominio básico del problema. Además de realizar un análisis detallado del estado del arte en cuanto a sistemas, tecnologías, metodologías y herramientas.

1.1 Sistema 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 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». [2]

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: [3]

1. *Atención Primaria de Salud (APS)*: Da solución aproximadamente al 80 % de los problemas de salud de la población y que correspondan con las acciones de promoción y protección de la salud. Aunque sus actividades se realizan en cualquier unidad del SNS, están relacionados 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, pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas. 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 en los mismos y el desarrollo de la investigación. A este nivel pertenecen los Institutos y hospitales especializados.

1.2 Sistemas de Información Hospitalaria (HIS)

Los HIS están orientados a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de la cualquier institución hospitalaria, permitiendo la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes burocráticos que enfrentan los pacientes.

Todo HIS genera reportes e informes dependiendo del área o servicio para el cual se requiera, dando lugar a una mayor calidad en la atención sanitaria. La función de un HIS es la de apoyar las actividades en los niveles operativos, tácticos y estratégicos dentro de un hospital. [4]

Es por todo esto que se hace tan imprescindible la utilización de un sistema de este tipo en la actualidad; para recabar, almacenar, procesar información clínica y administrativa, evitando la duplicidad o deterioro de la información del paciente y permitiendo que este reciba una mejor atención. Además haría el trabajo del personal de la institución hospitalaria, más flexible y fácil.

1.3 Antecedentes

En estos momentos a escala mundial, existen diversas soluciones informáticas en el sector hospitalario. Algunos de estos sistemas incluyen el departamento de enfermería como un subsistema de la aplicación, otros no lo conciben como parte de un HIS, pues enfocan al departamento de enfermería como sistema independiente. La tabla que se muestra a continuación refleja una comparación entre algunos de los sistemas existentes para la gestión hospitalaria a nivel mundial.

Sistemas para Hospitales				Tipo de Licencia		Tipo de Aplicación	
HIS	País	Multiplataforma	Enfermería	Libre	Propietario	Web	Windows
X- HIS	España	X	X	---	X	X	---
Galen Hospital	Cuba	---	---	---	X	---	X
HOSIX-V	España	---	X	---	X	X	---
Kewan-Cosmosalud	España	---	X	---	X	---	X

Tabla 1.1. Sistemas de información hospitalaria.

1.3.1 X-HIS

X-HIS es una solución global en un entorno abierto que, además de gestionar electrónicamente la historia clínica del paciente, permite la integración con otros sistemas de información. El software integra la información de un hospital o conjunto de hospitales(multicentro) con las áreas de gestión, hospitales de referencia, centros de especialidades, laboratorios, sistemas RIS y PACS, contabilidades y otros organismos o corporaciones que se desee, de forma que proporcione una Historia Clínica única y completa del paciente.

X-HIS es un sistema multiplataforma, capaz de soportar a sistemas operativos tales como Unix, Window y Linux. Posee independencia de base de datos: MS SQL Server, Oracle, Informix, Adaptive Server. Está implementado bajo la arquitectura cliente servidor y/o tres capas, además de ser un sistema multilingüaje. [5]

Dentro de las funcionalidades que contiene la estación de enfermería, para la gestión clínica le permite a los enfermeros:

Gestionar camas desde un puesto de enfermería. Además de conocer el estado actual de una planta o unidad de enfermería, de sus camas, datos de ingreso (diagnóstico, servicio y profesional responsable), información asistencial registrada e información personal de los pacientes. Pueden realizar diversas operaciones sobre los pacientes ingresados, tales como registro del curso clínico, petición de pruebas de laboratorio, implementación de su historia clínica, órdenes médicas, registro de constantes clínicas y registro de medicamento suministrado (comunicándose con farmacia para el control del stock). [6]

1.3.2 Galen Hospital

Es un producto de una empresa cubana desarrolladora de software, que está orientado hacia la informatización de la gestión de pacientes como elemento básico en la atención médica, optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos.

Esta solución integra diferentes módulos para conformar un Sistema de Gestión Hospitalaria que permite controlar la información del paciente e informatizar los diferentes servicios que brinda el hospital. Orientado hacia la informatización de la gestión de pacientes como elemento básico de control para mejorar la atención médica, optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos. [7]

Brinda la información requerida para la actividad gerencial a todos los niveles y la elaboración de reportes estadísticos. No contempla al departamento de enfermería como uno de sus módulos.

1.3.3 HOSIX-V

En un software de gestión hospitalaria diseñado en una arquitectura cliente - servidor accesible desde clientes web, lo que permite ser ejecutado desde cualquier navegador ya sea vía intranet o internet.

Está constituido por módulos que pueden funcionar de forma autónoma y se adapta con facilidad a todo tipo de organización. Además es también un sistema integrado, que relaciona directamente las diferentes áreas. Por eso es la herramienta indispensable para la organización con mayor eficacia del trabajo diario.

Dentro de las funcionalidades para enfermería permite: [8]

- La generación y mantenimiento de planes estandarizados y de cuidados, recogiendo su descripción, actividades, complicaciones, diagnósticos y objetos.
- Obtener informes de control de la calidad del trabajo realizado por la enfermera, así como los cuidados asistenciales recibidos por los pacientes.
- La generación de órdenes y peticiones a otros servicios y la obtención de documentación y listados.

1.3.4 Kewan- Cosmosalud

En un HIS, que posibilita generar documentos e imprimir etiquetas, de los procesos administrativos de admisión al paciente. Además se gestionan las áreas básicas, como admisión, archivo, facturación, entre otros y se gestionan los sistemas departamentales, como farmacia, rehabilitación, entre otros. También cubre la funcionalidad de atención primaria y permite la gestión de la información clínica, mediante la HCE del paciente. [9]

Dentro de las funcionalidades la estación de enfermería, le permite a los enfermeros gestionar camas; hojas de evolución; órdenes de enfermería; toma de constantes; diagnósticos de enfermería; resultados de pruebas; vacunas; pedidos de almacén.

Utiliza varios catálogos codificadores: CIE-9, CIE-10, NANDA; para médicos, enfermeros, trabajadores sociales, etc.; búsquedas por diferentes criterios; sinónimos; codificación automática; revisión de codificación. [10]

Después de haber analizado las características de diferentes HIS, se ha podido determinar que en su gran mayoría, son sistemas que presentan funcionalidades de relevancia para el campo de acción, algunos centrándose en un área específica del hospital, en la que se ofrecen los servicios de enfermería, pero presentan como desventajas que son sistemas de escritorio, lo cual eleva los costos y dificulta el proceso de despliegue, al tener que instalar dicho sistema en cada puesto de trabajo. Cuentan con la desventaja de no ser multiplataforma, lo que restringe al usuario a trabajar sobre un sistema operativo en específico.

1.4 Conceptos básicos asociados al dominio del problema

Para una mejor comprensión de la situación asociada al problema a solucionar, se explican a continuación diferentes términos propios del entorno vinculado al campo de acción.

Enfermería se encarga de realizar un conjunto de actividades profesionales destinadas a promover la adquisición, mantenimiento o restauración de un estado de salud óptimo del paciente.

El **enfermero**, que es aquella persona que ejerce la enfermería y sigue indicaciones médicas. Ayuda al individuo a llevar a cabo y mantener, por sí mismo, acciones de auto cuidado para conservar la salud y la vida, recuperarse de la enfermedad y afrontar las consecuencias de esta.

Un término que se maneja con bastante frecuencia por parte de los enfermeros, es **Stock**, lugar donde se tiene una cantidad limitada de los medicamentos que son de frecuente uso en una sala determinada.

Los enfermeros tienen acceso además a hojas de **indicaciones médicas**, donde generalmente se registra las indicaciones del médico hacia el paciente; puede tratarse de una prueba diagnóstica o de un tratamiento. En dichas hojas se registra además el **cumplimiento del tratamiento**, donde se establece un control de tratamiento, además se especifica si se ha cumplido o no, en el último caso, la causa por la cual no se le ha dado cumplimiento a dicho tratamiento.

La **evolución de enfermería**, término también de frecuente uso por el enfermero; es una sección de la historia clínica en la cual se especifica la incidencia de cada paciente cuando el enfermero realiza el recorrido habitual por las salas del hospital chequeando el estado del mismo.

1.4.1 Diagnóstico de enfermería y la taxonomía NANDA

Es el **diagnóstico de enfermería**: aquel juicio clínico sobre la respuesta de una persona, familia o comunidad a etapas de la vida/ problemas de salud reales o potenciales, que la enfermera identifica, valida y trata de forma independiente. [11]

Una **taxonomía** es un estudio teórico de la clasificación sistemática incluyendo sus bases, principios, reglas y procedimientos. Esta ciencia de cómo clasificar e identificar, ha permitido la codificación de los diagnósticos de enfermería, mediante la taxonomía **NANDA**, lenguaje enfermero reconocido, que cumple los criterios establecidos por el Comité for Nursing Practice Information Infrastructure (CNPII) de la American Nurses Association.

NANDA se divide en varias taxonomías, tales como: taxonomía de intervenciones de enfermería (NIC) y taxonomía de resultados de enfermería (NOC). [12]

1.5 Tendencias y tecnologías actuales a considerar

El desempeño de un proyecto está antecedido por la investigación de las tecnologías de punta que se utilizan, así como las ventajas y desventajas que trae su aplicación. Teniendo en cuenta las características del entorno, donde se desplegará el sistema a desarrollar y los antecedentes encontrados, se han definido un conjunto de tecnologías a utilizar, permitiéndole al mismo, que esté libre de costos adicionales relativos a pago de licencias de software, además de que sea multiplataforma.

A continuación se mencionarán las características de las tecnologías escogidas, así como sus ventajas.

1.5.1 Sistemas distribuidos. Modelo cliente servidor

A los sistemas distribuidos se les conoce como una colección de ordenadores autónomos, enlazados por una red de ordenadores y soportados por un software que hace que la colección actúe como un servicio integrado. Se podría ver a un sistema distribuido, como la unión de una tecnología de red y el uso de potentes máquinas multiprocesadores.

Una de las arquitecturas más utilizadas, en el desarrollo de los sistemas distribuidos, es la arquitectura **cliente servidor**. Esta es una arquitectura centralizada, ya que distribuye lógicamente los componentes en diferentes máquinas según sus funcionalidades.

Cliente servidor se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. [13]

Los sistemas que son construidos con esta arquitectura tienen una base de datos que reside en un equipo central, llamado servidor y es compartida entre varios usuarios.

El tener los datos almacenados y administrados en una ubicación central ofrece varias ventajas:

- Toda la información está almacenada en un mismo lugar, al cual todos los usuarios pueden acceder.
- Mediante el uso de restricciones y desencadenadores, se definen una sola vez las reglas de la organización y las reglas de seguridad.
- Como la información no se almacena en los clientes, no hay necesidad de dedicar espacio de disco a almacenar los datos, por lo que los gastos en hardware se pueden minimizar.

1.5.2 Patrones de arquitectura y diseño

Un patrón es un modelo que se puede seguir para realizar algo. Para el desarrollo de un software existen los patrones de arquitectura, que describen un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.

Por otra parte están los patrones de diseño, que son una abstracción de una solución en un nivel alto, proveen un esquema para refinar los subsistemas o los componentes de un sistema informático o las relaciones entre ellos.

El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que

provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista. [14]

Este modelo de arquitectura presenta varias ventajas:

- Soporta múltiples vistas: Como la vista se encuentra 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.
- 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 y como el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Si se diseña un sistema con este patrón, se puede construir por separado las piezas del programa y luego unirlos en tiempo de ejecución. Así mismo cuando alguna de las piezas se vea afectada, conllevando a que el funcionamiento de un componente no sea el más óptimo, esta puede cambiarse por otra.

1.5.3 Frameworks, librerías y componentes

Los frameworks se pueden considerar como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Aceleran el proceso de desarrollo, permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Java Server Faces (JSF)

El framework de desarrollo JSF trata la vista de interfaz de usuario, de una forma algo diferente a lo que comúnmente se utiliza en aplicaciones web, la programación de la interfaz se realiza a través de componentes, permitiendo el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. Permite crear interfaces de usuario (UI) para aplicaciones web, mediante componentes reutilizables.

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.

RichFaces

RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Con RichFaces se puede integrar fácilmente las capacidades JavaScript asíncrono y XML (AJAX), utilizando para ello el framework Ajax4jsf, dentro del desarrollo de aplicaciones de negocio de nivel empresarial.

RichFaces incluye ciclo de vida, validaciones, conversiones y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización del “look-and-feel” que puede ser fácilmente incorporado dentro de las aplicaciones JSF. [15]

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.

Ajax

El framework de extensión Ajax, que es el acrónimo para Asynchronous JavaScript + XML y el concepto es: Cargar y renderizar una página, luego mantenerse en esa página mientras los scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma. [16]

Ajax incorpora: [17]

- Presentación basada en estándares usando XHTML y CSS.
- Conversión y validación de campos.
- Establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y, combinado con Facelets, se

elimina la necesidad de motores de renderización, mejorando el rendimiento en general además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets.

Ajax4JSF

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, entre otras. [18]

Facelets

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF entre otras cosas. Permite crear plantillas para construir un árbol de componentes de forma que se puedan definirse como composición de otros.

JBoss Seam

JBoss Seam es un poderoso y moderno framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Una característica importante es que se pueden hacer validaciones en los POJOs (Plain Object Java) y presenta como una de sus principales innovaciones, 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. [19]

El principal contexto, es el contexto conversacional, así como el asociado a procesos del negocio, con estos se logra un uso más eficiente de la memoria evitando *memory-leaks*. Integra además el concepto de *workspaces* permitiendo que el usuario tenga en varios *tabs* del navegador procesos del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía JBoss JBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones (workflows) y complejas interacciones con el usuario (pageflows). [20]

Jboss Drools

Jboss Drools es un componente que permite gestionar, almacenar, clasificar, modificar y desplegar en caliente reglas de negocio, que podrán ser invocadas vía servicios web desde aplicaciones Java. Este proporciona un interfaz web, que facilita la definición de reglas a personas que no trabajan con habitualmente con IDEs de desarrollo, principalmente analista de negocio y expertos en reglas. [21]

Hibernate

Hibernate es un framework, que genera sentencias sql a partir de sentencias HQL. Provee herramientas de mapeo objeto/relacional y permite reducir significativamente el tiempo de desarrollo. A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos

Deben destacarse ciertas características de las colecciones en Hibernate: [22]

1. *Colecciones ordenadas*: Hibernate soporta la implementación de colecciones ordenadas a través de los interfaces `java.util.SortedMap` y `java.util.SortedSet`. Si se desea, se puede definir un comparador en la definición de la colección. Los valores permitidos son `natural`, `unsorted` y el nombre de la clase que implementa `java.util.Comparator`.
2. *El colector de basura de las colecciones*: Las colecciones son automáticamente persistidas cuando son referenciadas por un objeto persistente y también son borradas automáticamente cuando dejan de serlo.

Tecnologías

CSS, hojas de estilo en cascada (Cascading Style Sheets), que constituyen un mecanismo simple, que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos *HTML* y *XML*, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el

estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. [23]

HTML, lenguaje que hace posible presentar información (por ejemplo, investigaciones científicas) en internet. Lo que se ve al visualizar una página en internet es la interpretación que hace el navegador del código HTML.

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.

XML, lenguaje de etiquetado extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. [24]

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Extended EI, 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 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 Lenguaje de programación

Como lenguaje escogido se tiene a Java. Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

El lenguaje Java tiene como ventaja que es un lenguaje multiplataforma. Este es un lenguaje que se ha extendido, cobrando, cada día, más importancia tanto en el ámbito de Internet como en la informática

en general. Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. [25]

1.7 Servidor de aplicaciones

JBoss Application Server es el servidor de aplicaciones java más utilizado en el mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en tecnologías web. También soporta Enterprise Java Beans, arquitectura componente del lado del servidor para la plataforma Java, que permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos. [26]

Una de las facilidades que este servidor presenta es que puede ser instalado sobre varios ambientes, tales como Windows o GNU/Linux.

1.8 Sistemas gestores de bases de datos

Un sistema gestor de base de datos (SGBD), puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario.

Las principales funciones que debe cumplir un SGBD es relacionarse con la creación y mantenimiento de la base de datos (BD), el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

El **PostgreSQL**, es un SGBD, que constituye una mejora del sistema de base de datos Postgres, ya que mantiene los puntos fuertes de su antecesor como ser su poderoso modelo de datos y toda su riqueza en cuanto a tipos de datos, y reemplaza el lenguaje de consultas PostQuel con un subconjunto más extenso de SQL. Es un proyecto Open Source mucho más puro, ya que no tiene empresas comerciales detrás como es MySQL o Firebird (en un principio fue Borland). [27]

Dentro de los beneficios que aporta se encuentran: [28]

- Puede ser ejecutado sobre la mayoría de las plataformas.

- El tamaño máximo para las bases de datos PostgreSQL es ilimitado.
- *Alta concurrencia*: permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- *Disparadores (triggers)*: Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

1.9 Metodologías de desarrollo de software

El resultado final de un proyecto de software es un producto donde intervienen las personas a través de un proceso de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto, a través de plantillas, que explica los pasos necesarios para terminar el proyecto.

En un proceso de desarrollo de software la metodología define quien debe hacer que, cuando y como debe hacerlo. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, entre otros) exigen que el proceso sea configurable.

1.9.1 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo, es la metodología líder en el mundo de modelación visual, análisis de requerimientos y diseño de sistemas orientados a objetos.

RUP, se encuentra entre las metodologías más conocidas. Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. Agrupa las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. El ciclo de vida de RUP se caracteriza por ser dirigido por caso de uso, centrado en la arquitectura, iterativo e incremental. [29]

Algunas ventajas de la aplicación de esta metodología son las siguientes: [30]

1. Se pueden identificar riesgos desde el inicio del desarrollo del software.

2. Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan para obtener resultados claros a corto plazo.
3. Como en cada iteración se desarrollan todos los flujos de trabajo, se distribuye la carga de trabajo a lo largo del tiempo del proyecto.
4. Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño.

1.9.2 Lenguaje Unificado de Modelado (UML)

RUP y el Lenguaje Unificado de Modelado están estrechamente relacionados entre sí, pues mientras el primero establece las actividades y los criterios para conducir un sistema desde su máximo nivel de abstracción (la idea en la cabeza del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente), el segundo ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite modelar sistemas con tecnología orientada a objetos. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. [31]

Este lenguaje de modelado formal aporta las siguientes ventajas: [32]

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

1.9.3 Notación para el modelado de procesos de negocio (BPMN)

Antes de conocer para que se utiliza esta notación, es necesario resaltar en qué consiste un proceso de negocio; este es una colección de actividades que, tomando una o varias clases de entradas, crean

una salida que tiene valor para un cliente. Los procesos de negocio representan el flujo de trabajo y de información a través del negocio. [33]

El objetivo principal de esta notación, es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua.

A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos. La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de los mismos.

La administración de los procesos permite asegurar que los mismos se ejecuten eficientemente, y la obtención de información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que puede identificar posibles ineficiencias en los mismos, y actuar sobre ellas para optimizarlos. [34]

1.10 Herramientas

Dentro del conjunto de herramientas que conforman el ambiente de desarrollo del módulo de enfermería para el Sistema de Información Hospitalaria alas HIS, se encuentra el Visual Paradigm 6.0 como herramienta CASE (*Computer-Aided Software Engineering*) basada en UML, que permite crear los diagramas que se generen como parte de la documentación. Con la utilización de BPMN para la documentación del modelado de los procesos del negocio.

Para el desarrollo de la aplicación web, se tiene al Entorno de desarrollo integrado (IDE) Eclipse SDK en su versión 3.3.2, con la colección de plug-ins de JBoos Tools, necesarios para utilizar los frameworks de desarrollo que se proponen.

Como el lenguaje de programación es Java, se necesita un intermediario entre una aplicación programada en Java y el sistema operativo que se esté utilizando, para ello se escogió el JRE (Java Runtime Environment), una máquina virtual de Java y Java EE (Java Enterprise Edition) como plataforma de soporte, que mediante JPA (estándar para la persistencia y el mapeo objeto/relacional), permite utilizar el dominio Java para administrar las bases de datos relacionales.

Conclusiones

En el capítulo se realiza un estudio sobre la organización del sistema de salud a nivel internacional, se enmarca en la atención hospitalaria como lo más importante.

Dentro de esta atención se realizó un estudio de diferentes HIS a nivel global, obteniéndose como resultado que por lo general se encargan de gestionar actividades en diferentes áreas de los hospitales, pero tienen el inconveniente de ser propietarios o de simplemente no poder integrarse con otro sistema. Por lo que se hace necesario desarrollar un nuevo sistema, con la utilización de estándares como NANDA, para el diagnóstico médico y con las tecnologías seleccionadas.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Este capítulo está dedicado a describir las características generales del sistema. Además se describe el modelo de negocio estudiado, se especifican los requisitos funcionales y no funcionales y los casos de usos del sistema.

2.1 *Procesos de negocio de enfermería*

Los principales procesos que realizan los enfermeros en las instituciones hospitalarias, diariamente, son: la atención al paciente hospitalizado, a los pacientes que pasan a la sala de recuperación, luego de haber sido sometidos a una intervención quirúrgica, lo que comúnmente se le llama post-operatorio, además la atención en consulta externa y en el área de emergencia.

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

Para comprender mejor la panorámica que implica el campo de acción, los diferentes procesos que se han identificado, explicando las actividades realizadas por las enfermeras, en diferentes áreas o servicios dentro de las instituciones hospitalarias, se reflejan a continuación.

Atender paciente en hospitalización

En este proceso se recrean todas las operaciones que se realizan sobre el paciente hospitalizado. En el que se destacan actividades por parte del enfermero, como lo es entregar y recibir turnos y enfermos. En esta actividad el enfermero controla los materiales que existen en la sala por cada turno, controlando los insumos de la sala y el estado de ellos. También analiza alguna eventualidad ocurrida en el turno anterior, así como el estado de los pacientes y la conducta a seguir para su recuperación.

Se destaca además como actividad el realizar el protocolo de ingreso, donde el enfermero que recibe al paciente, se encarga de ubicarlo en la habitación y cama asignada, explicando a familiares y paciente sobre los cuidados pre-operatorio y post-operatorio y se deja por escrito la evolución del paciente, las condiciones en que ingreso así como el diagnóstico de enfermería. En este documento también es escrito, por parte del médico, el diagnóstico emitido por el mismo.

El proceso también contiene la recogida de los datos del paciente que están o no en la historia clínica (HC), para así facilitar el acceso a estos cuando la HC está ocupada o en otro servicio. Otra actividad es la medición de signos vitales, lo cual se hace en dependencia del estado del paciente y se actualiza

en cada turno o cada cuatro horas, dependiendo del estado del paciente. Esta información el enfermero la pasa a la HC, recogiendo además la firma de quien lo actualiza.

Se realizan solicitudes y recibos de medicamentos. En casos en el que el servicio no tenga acceso a la unidosis, la petición es realizada a farmacia por el enfermero, esta incluye solicitud, recípe y la orden médica que se encuentra en la HC, el enfermero se encarga de llevar esta solicitud y de buscar los medicamentos una vez que en farmacia son despachados.

Como último aspecto se tiene al hecho de que los enfermeros se encargan de darle cumplimiento al tratamiento médico, plasmando en el documento de control del tratamiento, la persona que cumplió el mismo y el turno en que lo realizó. Esta información se guarda en la HC.

Atender paciente en consulta externa

Este proceso se encarga de la atención al paciente por parte de los enfermeros en el área de consulta externa. Donde la primera acción del enfermero es recibir la unidad de trabajo, verificando que el material quirúrgico y demás insumos estén en correcto estado, registrando eventualidades en el libro de control de materiales.

Otra actividad que se realiza en el proceso, es la solicitud de materiales. El enfermero se encarga de realizar la solicitud al stock de servicio, en caso de no existan los materiales deseados, el coordinador se encarga de realizar lo solicitud al depósito general; cuando este recibe lo solicitado, despacha el pedido y actualiza el documento de control de materiales. El enfermero por su parte verifica que la cantidad insumos recibidos sea la correcta.

En la consulta externa, cuando el paciente requiere de atención de enfermería, el enfermero recoge sus datos y la indicación médica, registrándolo en el libro de registro de actividades diarias y mensualmente el enfermero reporta todas las actividades que realizó, tomándose los datos de dicho libro. En caso de suceder cualquier eventualidad en el servicio, el enfermero se encarga de escribirlo en el reporte de enfermería.

Una importante actividad de los enfermeros en esta área del hospital, radica a la hora de clasificar a los pacientes en primera consulta, esto es cuando asisten a consulta por primera vez y de control o consulta sucesiva cuando es para seguimiento. Luego de clasificarlos los organiza en la manera en que fueron llegando, para mantener organizada la fila y también pasa a los mismos a la consulta a

medida que el médico vaya concluyendo. La auxiliar de enfermería tiene asignada varias consultas y pasa a los pacientes a medida que las mismas se desocupan.

Atender paciente en emergencia

Este proceso se encarga de la atención al paciente por parte de los enfermeros en el área de emergencia. En esta área el enfermero se encarga de medir los signos vitales al paciente cuando el estado en el que llega es favorable, sino primeramente se realizan actividades para salvar su vida, de lo cual depende además la periodicidad con que se realice la medición. Dentro de esos signos están la temperatura, el pulso, la respiración, la tensión arterial y alguna observación del estado del paciente, los cuales se registran en la HC.

En esta área el enfermo también se encarga de plasmar en el documento de control de tratamientos, la persona que lo cumplió y el turno en que se realizó, registrándose en la HC. Además de que en cada turno, se describe la evolución del paciente y se escribe la observación hacia el mismo.

Cuando el paciente es hospitalizado en emergencia, el enfermero se encarga de llenar un documento para el plan de atención hacia el paciente.

Otra actividad es el registro de psicotrópicos, donde se controlan los medicamentos psicotrópicos que se reciben en cada turno, recogiéndose la fecha, turno, nombre de medicamento, cantidad con que se cuenta y si se le aplica a algún paciente, se recogen nombres y apellidos, dosis, turno y nombre de el enfermero que lo aplicó, así como el nombre del médico que lo indicó.

Dentro de cada turno se registran además eventualidades que surgen en su curso, en el caso de no ocurrir nada relevante se registra también como constancia.

Se pueden realizar además solicitudes de medicamentos a farmacia, acción realizada por un mensajero encargado de llevar dicha solicitud y el enfermero se encarga de buscar los medicamentos una vez que farmacia lo despacha.

Atender paciente en post-operatorio

Este proceso se encarga de la atención al paciente por parte de los enfermeros, cuando se han realizado intervenciones quirúrgicas, lo que se llama post-operatorio.

Cuando un paciente va a ingresar al post-operatorio, el enfermero escribe en el libro de ingreso los datos del paciente recién ingresado, nombre y apellidos del paciente, número de historia clínica, fecha hora que ingresa, diagnóstico médico, servicio al cual pertenece, la fecha de egreso, observación, la firma de quien lo recibió.

En esta área el enfermero se encarga de medir los signos vitales cuando el paciente ingresa a post-operatorio, realizándose dicha actividad cada una hora y utilizando una hoja de control de signos, balance y medicamentos, diferente en cada turno de trabajo. Después del médico indicar un tratamiento, el enfermero lo escribe en la hoja antes mencionada junto al cumplimiento del mismo.

En el caso del servicio no tener acceso a la unidosis, la solicitud y el recibo de medicamentos son realizados a la farmacia por el enfermero, esta incluye solicitud, recípe y la orden médica que se encuentra en la HC, el enfermero se encarga de llevar esta solicitud y de buscar los medicamentos una vez que en farmacia son despachados.

Otras actividades son el control de narcóticos y psicotrópicos donde el enfermero realiza un conteo de los medicamentos ubicados en el stock del servicio y la cantidad recibida del turno anterior, recogándose para el caso de los narcóticos la fecha, el nombre y apellidos del paciente al que se le ha indicado el medicamento, número de historia, nombre del medicamento, cantidad de ampollas, nombre y apellidos de médico que lo indicó, cédula de identidad o matrícula y firma de el enfermero. Para el caso de los psicotrópicos se recoge el nombre de medicamento, la fecha, el turno, la cantidad y la firma del coordinador.

Además del ingreso se notifica el egreso, ya sea por una mejoría o por defunción. En el área de post operatorio se encargan de avisar al servicio al cual pertenece el paciente para que este recoja la historia clínica del paciente.

2.2 Información manipulada

Como resultado de los procesos descritos anteriormente, se generan un gran número de documentos, comunes dentro de ellos. A continuación se presentan los principales documentos, que son manipulados por los enfermeros en sus actividades diarias dentro de las instituciones hospitalarias.

Hoja de evolución de enfermería: Este es un documento elaborado por el enfermero, tras realizar la revista en cada turno, describiendo la evolución del paciente y la observación ante cualquier eventualidad en la fecha y hora correspondiente. Este documento es único por cada paciente. Se

recoge el nombre y el estado en que se encuentra el paciente. En caso que el paciente no tenga identificación se le asigna una clave o se registra como paciente desconocido. Se registra además el número de historia clínica, número de cédula de identidad, servicio al que pertenece y ubicación física dentro del hospital.

Libro de signos vitales: Es un documento para registrar los signos vitales de los pacientes, donde se guardan datos como nombre y apellidos del paciente, servicio en el que está con su ubicación física, hora y fecha de la medición de los signos y el número de HC. Los signos vitales que en este documento se registran son la temperatura, pulso, presión arterial y frecuencia respiratoria.

Libro de ingreso: En este documento se registran los datos del ingreso de los pacientes a post-operatorio, guardándose datos como el nombre y apellidos del paciente, así como su número de HC, fecha y hora del ingreso, diagnóstico médico, el servicio al cual pertenece, observación y el nombre del enfermero que recibió al paciente. En este libro además se registran los datos del egreso, la hora y fecha, la observación y el nombre del enfermero que recibe el egreso así como el médico que realiza el egreso.

Registro diario de actividades de enfermería: En este documento se registra toda la información referente a las actividades que realizan los enfermeros diariamente en el hospital, como por ejemplo la fecha de la actividad, nombre y apellidos del enfermero, programa de salud al que pertenece, servicio en el que trabaja, así como nombre y apellidos del paciente que atiende, con su edad y sexo respectivo, de donde proviene y el procedimiento que realiza el mismo con respecto al paciente.

Hoja de indicaciones médicas. Cumplimiento de enfermería: En este documento se registra los datos de la indicación médica, con el nombre de médico, el nombre y apellidos del paciente, el servicio al que pertenece, la ubicación física dentro del hospital, la fecha y hora del cumplimiento del tratamiento. De los medicamentos indicados, se guardará el nombre, tipo, dosis, vía (oral, endovenosa), volumen en caso de ser líquido. Además se recoge por turno, el nombre del enfermero que le da cumplimiento a la indicación, con la fecha, hora y observaciones. En el caso de que no se haya cumplido algún tratamiento se debe especificar la causa. Toda la información que en este documento se guarda, posteriormente es registrada en la Historia Clínica.

2.3 Modelo de Negocio

La disciplina de modelado de negocio es la primera que propone RUP dentro del ciclo de desarrollo de un software, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema.

El modelado del negocio es una técnica para comprender los procesos del negocio de la organización. Los propósitos que se persiguen al realizarse este modelo son: entender la estructura y la dinámica de la organización, entender los problemas actuales e identificar mejoras potenciales, asegurarse de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga. [35]

Para realizar el modelado del negocio se utilizó la notación BPMN. La misma aporta una mayor visibilidad de las actividades que se realizan y ayuda a lograr un mejor entendimiento del flujo de trabajo existente entre las áreas hospitalarias, lo que permite definir con claridad las actividades innecesarias a la hora de automatizar el negocio.

2.3.1 Diagrama de procesos del negocio

Mediante los elementos gráficos que proporciona la notación BPMN se pueden construir los diagramas de procesos. Estos diagramas están basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización. Con el objetivo de obtener diagramas entendibles, tanto para audiencias de negocio como técnicas, los diagramas de procesos de negocio están estructurados con diferentes elementos gráficos, entre los que se pueden contar los objetos de flujo, los de conexión, los objetos carriles y los artefactos. [36]

Para un mejor entendimiento a continuación se explican los elementos gráficos utilizados en el modelado.

Los objetos de flujo pueden ser eventos, actividades o decisiones. Los eventos pueden ser de inicio, intermedio o fin, indicando una acción lógica en el flujo del proceso. Por su parte las actividades se refieren a un término genérico para el trabajo que una organización realiza. Pueden ser tareas o subprocesos. Por último se tiene a las decisiones, que se utilizan para controlar la divergencia de la secuencia de un flujo. Determina las “tradicionales” decisiones, tanto bifurcaciones, como uniones y acoplamientos de flujos. [37]

Entre los objetos de conexión que se utilizan para conectar los objetos de flujo se pueden encontrar los flujos de secuencia, de mensaje y de asociación. Los flujos de secuencia son utilizados para indicar orden lógico entre las actividades del proceso. Por su parte los flujos de mensajes se utilizan para mostrar el flujo de mensajes entre dos participantes de procesos separados. Por último la asociación se utiliza para asociar datos, textos u otros artefactos con flujos de objetos. Generalmente se utilizan para mostrar las entradas y salidas de las actividades. [38]

Los objetos carriles constituyen el mecanismo para organizar actividades en categorías separadas visualmente, para ilustrar diferentes capacidades funcionales o responsabilidades. BPMN soporta los objetos carriles con dos constructores principales: calles (pool) y pistas (lane). Las calles representan los participantes en el proceso. Pueden actuar como contenedor gráfico para separar al grupo de actividades realizadas por un participante de otras calles. Por su parte las pistas constituyen una partición dentro de las calles, utilizándose para organizar y categorizar actividades. [39]

Dentro de los artefactos se encuentran los objetos de datos (Data Object), que constituyen el mecanismo para mostrar como las actividades requieren o producen objetos. Ellos se conectan a las actividades a través de asociaciones.

A continuación se muestran los procesos de negocio identificados, así como la relación que existe entre ellos. El color rosa implica a los subprocessos que se van a automatizar.

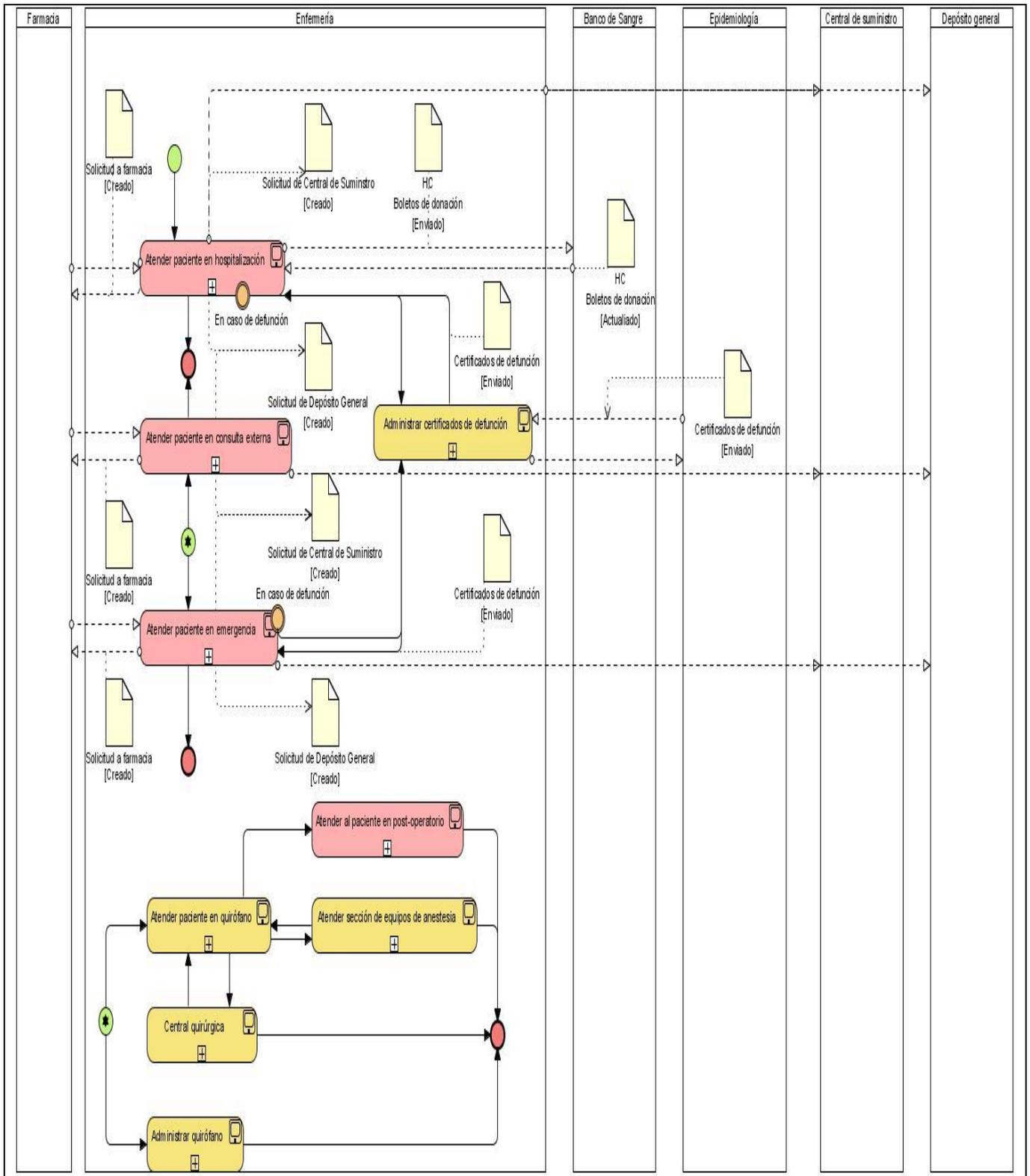


Figura 2.1 Diagrama de procesos del negocio

2.3.1.1. Diagrama de subprocesos

Para la comprensión de los diferentes procesos que se van a automatizar, se muestran imágenes, que contienen los diagramas de procesos. En dichos diagramas de procesos, se muestran las actividades que se automatizarán tanto por el sistema como por otro que se le integre, para esto se diferencian por colores: rosa para actividades automatizadas por el sistema, violeta para actividades automatizadas por el sistema integrado.

Por cada proceso se identifican actores involucrados. Estos pueden representar a un ser humano, software o hardware, institución o entidad, que desempeña un rol dentro de los procesos del negocio. Interactúa además con entidades y otros involucrados para que el negocio funcione. A continuación se describen dichos roles además de una breve descripción por cada proceso.

ACTORES INVOLUCRADOS	
Rol	Funciones
Coordinador de enfermería	Se encarga del buen funcionamiento de su sala, y supervisa las actividades de los enfermeros de su sala.
Enfermera	Se encarga de realizarles mediciones de signos vitales a los pacientes que le son asignados, así como chequear el cumplimiento de las indicaciones médicas.
Auxiliar de enfermería	Se encarga de la atención al paciente cuyo diagnóstico sea menos complejo, y de su aseo personal y de los cuidados de enfermería.
Secretaria	Se encarga de transcribir documentos en la sala, así como de dar información a los pacientes o familiares.
Director de enfermería	Dirige y guía todo el proceso de enfermería en el hospital.
Subdirector de enfermería	Es receptor del personal que ingresa a enfermería y que dependen del departamento.
Supervisor de enfermería	Se encarga y supervisa todas las actividades que realizan los coordinadores que le son asignadas.

Tabla 2.1. Actores involucrados a los procesos.

PROCESO:	Atender paciente en consulta externa.
Flujo de Información	
<p>En el diagrama se representan el flujo de eventos que ocurre diariamente en el área de consulta externa donde el enfermero actualiza el libro de control de materiales y el de reporte al llegar a la unidad de trabajo, al igual que actualiza el libro de reporte de enfermería en caso de ocurrir alguna eventualidad. Puede consultar la hoja de indicaciones médicas y darle cumplimiento a los tratamientos indicados. Por su parte el coordinador se encarga de despachar medicamentos cuando se realicen solicitudes para cumplir el tratamiento y el auxiliar de enfermería clasifica a los pacientes que asisten a la consulta.</p>	

Tabla 2.2. Descripción del proceso Atender paciente en consulta externa.

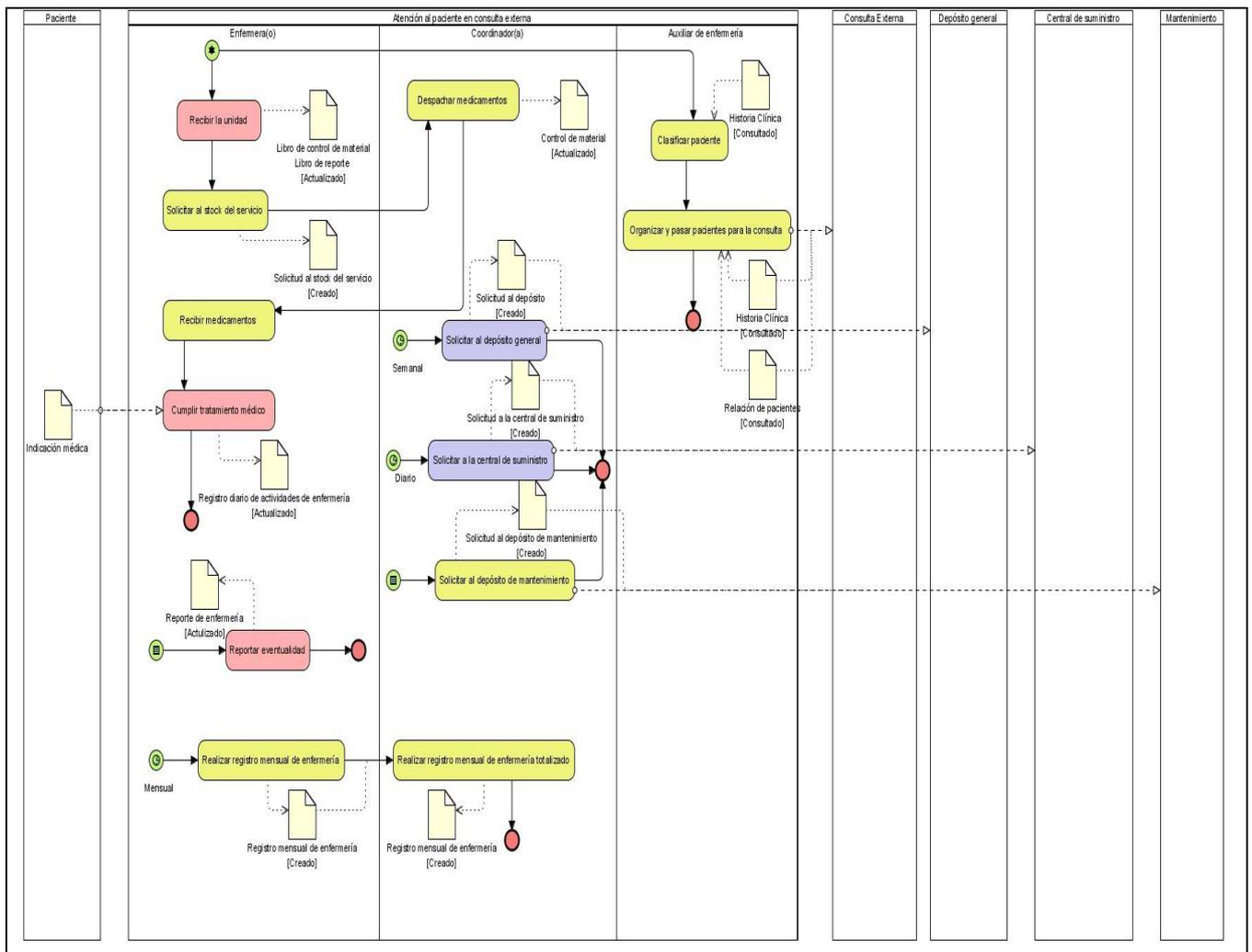


Figura 2.2 Diagrama de proceso Atender paciente en consulta externa

PROCESO:	Atender paciente en emergencia.
Flujo de Información	
<p>En el diagrama se representan el flujo de eventos que ocurre diariamente en el área de emergencia, donde el enfermero se encarga de actualizar la hoja de control de signos, balance y medicamentos, cuando se realicen las mediciones de signos vitales a los pacientes y al darle cumplimiento a las indicaciones médicas. Crea además el plan de atención de enfermería. Actualiza además el libro de emergencias cuando ocurra alguna eventualidad en la sala.</p>	

Tabla 2.1. Descripción del proceso Atender paciente en emergencia.

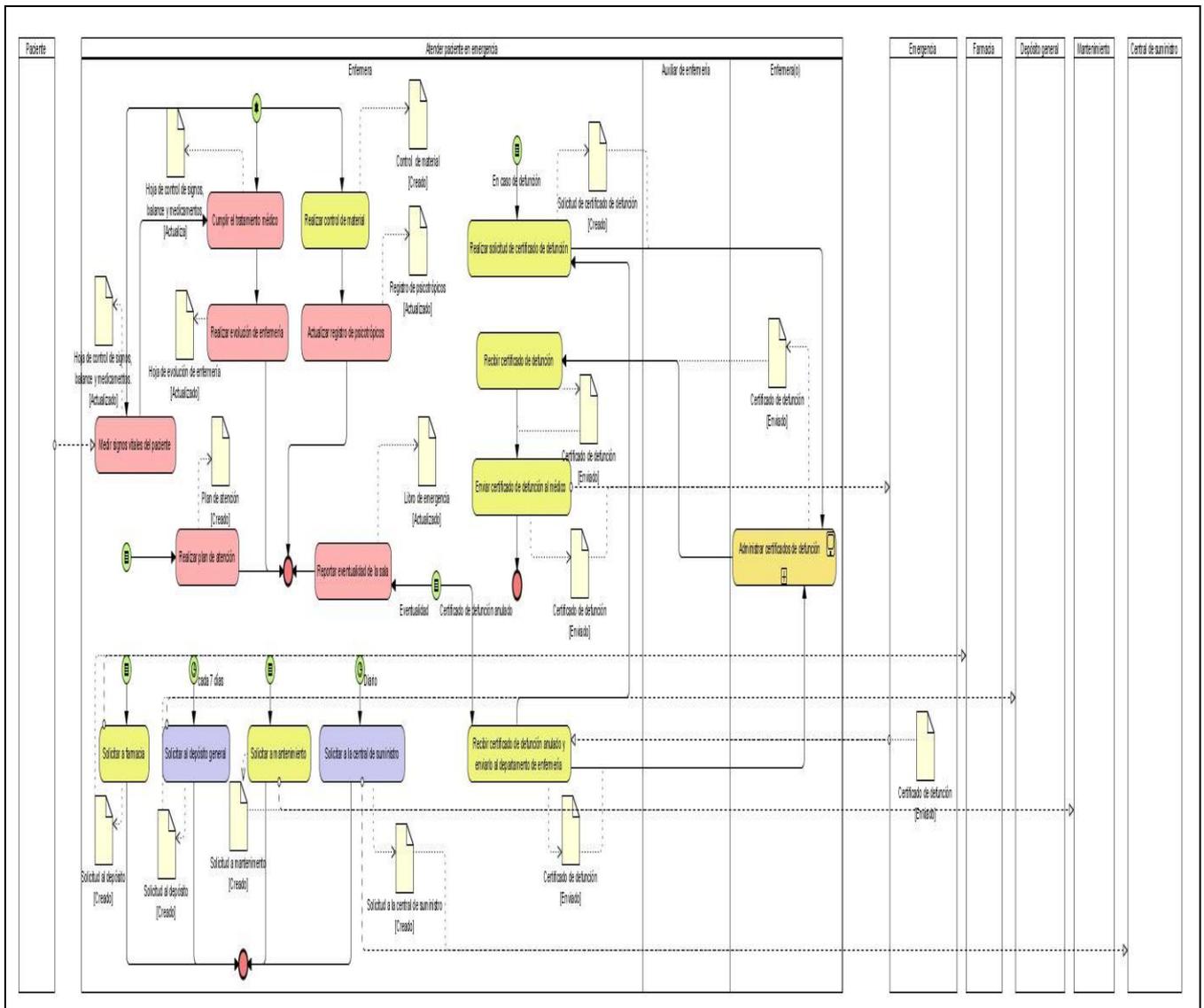


Figura 2.1 Diagrama de proceso Atender paciente en emergencia

PROCESO:	Atender paciente en post-operatorio.
Flujo de Información	
<p>En el diagrama se representan el flujo de eventos que ocurre diariamente en las salas de recuperación, donde se tiene un control sobre el ingreso y egreso a la misma. Las principales actividades radican en el registro de los signos vitales por los enfermeros, en la hoja de control de signos, balance y medicamentos. Además el enfermero puede realizar el control de tratamiento y escribir en la hoja de evolución del paciente las observaciones vistas. Estos tienen además un control sobre los pacientes En esta área el coordinador se encarga de actualizar el libro de reportes de enfermería cuando ocurra alguna incidencia, así como el de control de narcóticos.</p>	

Tabla 2.1. Descripción del proceso Atender paciente en post-operatorio.

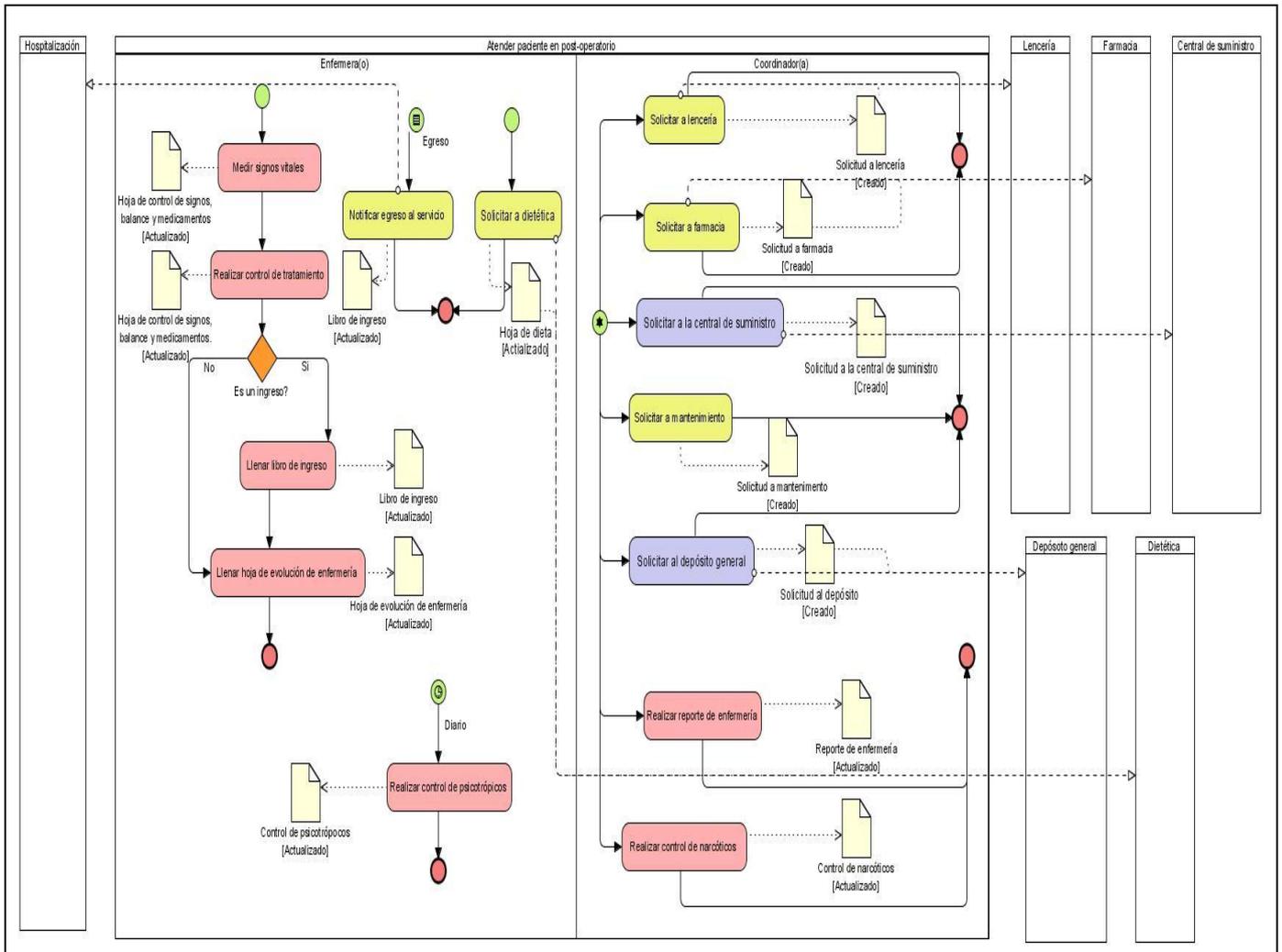


Figura 2.2 Diagrama de proceso Atender paciente en post-operatorio

2.4 Requerimientos

RUP propone como una de sus flujos de trabajo el de Gestión de Requerimientos. En este flujo tiene mayor peso en la fase de inicio, teniéndose como artefactos fundamentales del flujo, la especificación de requisitos, actores y casos de usos del sistema, que constituyen el modelo de casos de uso.

2.4.1 Especificación de los requerimientos de software

Los requerimientos de software, posibilitan determinar la solución tecnológica a algún problema existente, logrando como resultado, un informe detallado de los requerimientos necesarios para la solución a dicho problema, incluyendo las especificaciones técnicas, determinación de casos de uso y las especificaciones básicas del sistema que permite alcanzar los objetivos y solucionar los problemas existentes. [40]

Los requerimientos de software se clasifican en funcionales y no funcionales, donde la primera clasificación corresponde a las especificaciones del sistema, mientras que los últimos expresan las cualidades que el producto debe tener.

2.4.1.1. Requerimientos funcionales

Los requerimientos funcionales especifican capacidades o condiciones que el sistema debe cumplir, 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. [41]. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas, se han definido los siguientes requisitos funcionales:

RF_1 Listar pacientes	RF_9 Registrar ingreso del paciente
RF_2 Listar camas	RF_10 Eliminar ingreso del paciente
RF_3 Crear actividad diaria	RF_11 Ver ingreso del paciente
RF_4 Eliminar actividad diaria	RF_12 Buscar ingreso del paciente
RF_5 Ver actividad diaria	RF_13 Modificar ingreso del paciente
RF_6 Buscar actividad diaria	RF_14 Registrar egreso
RF_7 Modificar actividad diaria	RF_15 Eliminar egreso
RF_8 Generar reporte de actividades diarias	RF_16 Ver datos de egreso

RF_17 Buscar egreso de paciente	RF_31 Buscar signos vitales
RF_18 Listar pacientes en post-operatorio	RF_32 Modificar signos vitales
RF_19 Crear evolución de enfermería	RF_33 Crear balance de líquidos (BL)
RF_20 Ver evolución de enfermería	RF_34 Ver control de BL
RF_21 Buscar evolución de enfermería	RF_35 Buscar control de BL
RF_22 Modificar evolución de enfermería	RF_36 Modificar control de BL
RF_23 Crear observación del turno	RF_37 Generar reporte de BL
RF_24 Crear control de tratamiento	RF_38 Crear diagnóstico de enfermería
RF_25 Ver control de tratamiento	RF_39 Ver diagnóstico de enfermería(E)
RF_26 Buscar control de tratamiento	RF_40 Buscar diagnóstico de enfermería
RF_28 Registrar signos vitales	RF_41 Modificar diagnóstico de E.
RF_29 Graficar signos vitales	RF_42 Crear plan de atención E.
RF_30 Ver signos vitales	RF_43 Ver plan de atención de E.

2.4.1.2. Requerimientos no funcionales

Los requerimientos no funcionales expresan las propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. [42]

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

2.4.1.2.1. 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

2.4.1.2.2. Seguridad

Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.

Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo 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 en el sistema.

Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD. El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

2.4.1.2.3. 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.

2.4.1.2.4. 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.

2.4.1.2.5. Hardware

2.4.1.2.5.1. Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del sistema, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se Firefox 2.x).

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.

2.4.1.2.5.2. Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables.

- Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.

2.4.1.2.6. Software

El sistema debe correr en sistemas operativos Windows y Linux, utilizando la plataforma JAVA (Java virtual machine, JBoss AS y PostgreSQL).

El sistema deberá disponer del navegador web Firefox 2.x.

2.4.1.2.7. 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.

2.4.1.2.8. Interfaz

2.4.1.2.8.1. Interfaces de usuario

Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información. La interfaz contará con 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 aparecerán en idioma español.

2.4.1.2.9. Soportabilidad

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

2.4.2 Definición de actores del sistema

A continuación se describen las personas o roles, que han sido identificadas como actores del sistema.

Actores del sistema	Funciones
Usuario	Es el rol genérico que se utiliza para la autenticación en el sistema.
Enfermero	Es el rol genérico que se encarga de la atención al paciente en los diferentes servicios. Este se encarga de realizar mediciones de signos vitales, cumple el control de tratamiento, puede evolucionar al paciente y emitir un diagnóstico de enfermería, así como un plan de atención de enfermería para ese diagnóstico.
Enfermero de hospitalización y emergencia	Es el rol que se encarga de la atención al paciente en el área de hospitalización y emergencia.
Enfermero de consulta externa	Es el rol que se encarga de la atención al paciente en el área de consulta externa. Puede crear reportes de

	actividades diarias, cuando ocurra alguna eventualidad.
Enfermero en post – operatorio	Es el rol que se encarga de la atención al paciente en el área de post-operatorio. Registra ingresos y egresos de pacientes a la sala de recuperación.
Coordinador	Es el rol que se encarga de las gestiones administrativas en los servicios.
Coordinador en hospitalización	Es el rol que se encarga de las gestiones administrativas en hospitalización.
Coordinador en consulta externa	Es el rol que se encarga de las gestiones administrativas en consulta externa.
Coordinador en emergencia	Es el rol que se encarga de las gestiones administrativas en emergencia.

Tabla 2.2. Descripción de actores del sistema

2.4.2.1. 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 las diferentes áreas de las instituciones hospitalarias, en dependencia del lugar donde labore serán las actividades que podrá realizar sobre el sistema.

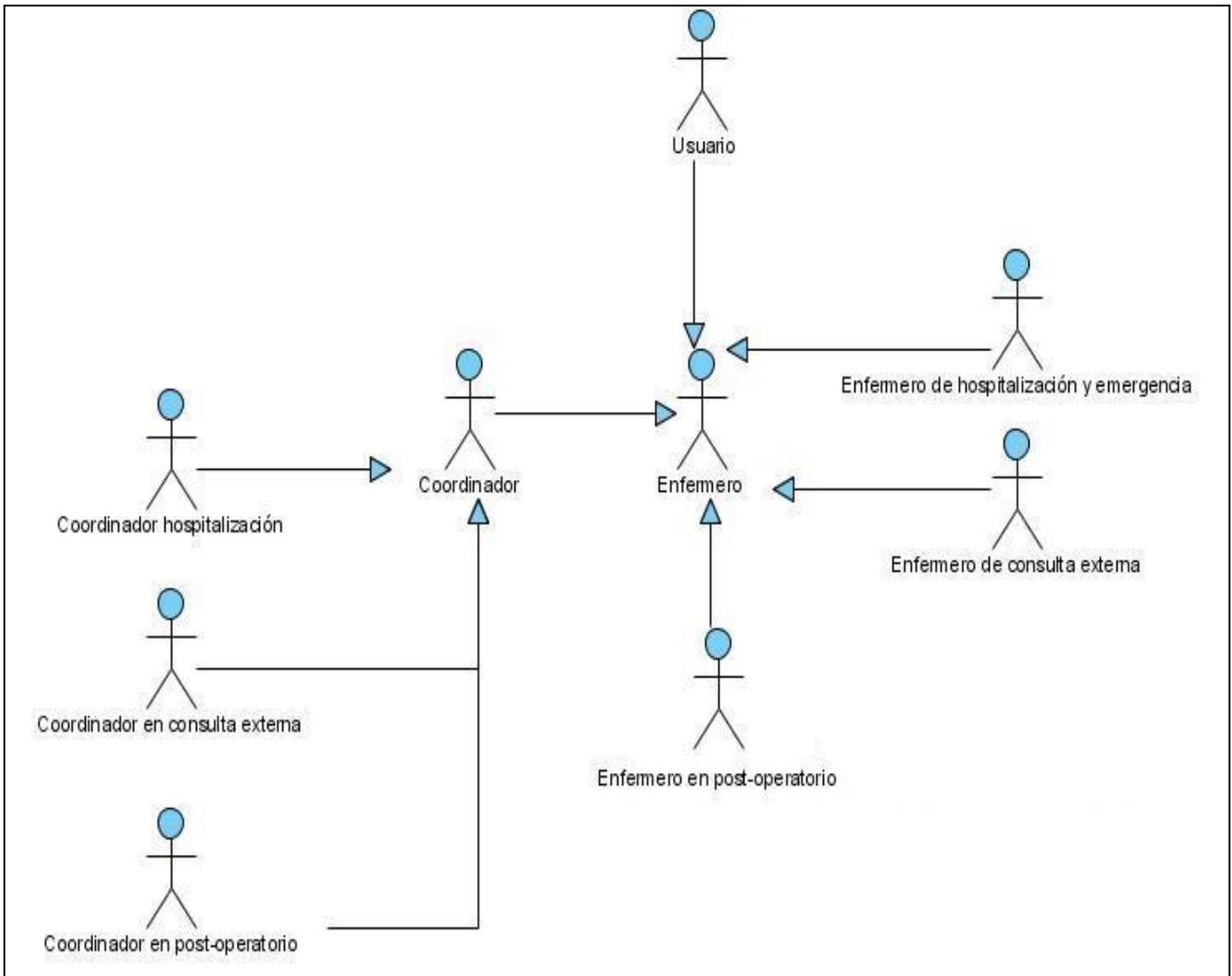


Figura 2.3 Actores del sistema

2.4.3 Diagrama de casos de uso del sistema

Para una mejor comprensión de la concepción del sistema, se han agrupado los casos de uso en procesos. En el caso de los procesos Atender paciente en hospitalización y emergencia, como ambos incluían actividades similares, se agruparon en un mismo proceso, en el modelado del sistema.

Diagrama de casos de uso Atender paciente en hospitalización

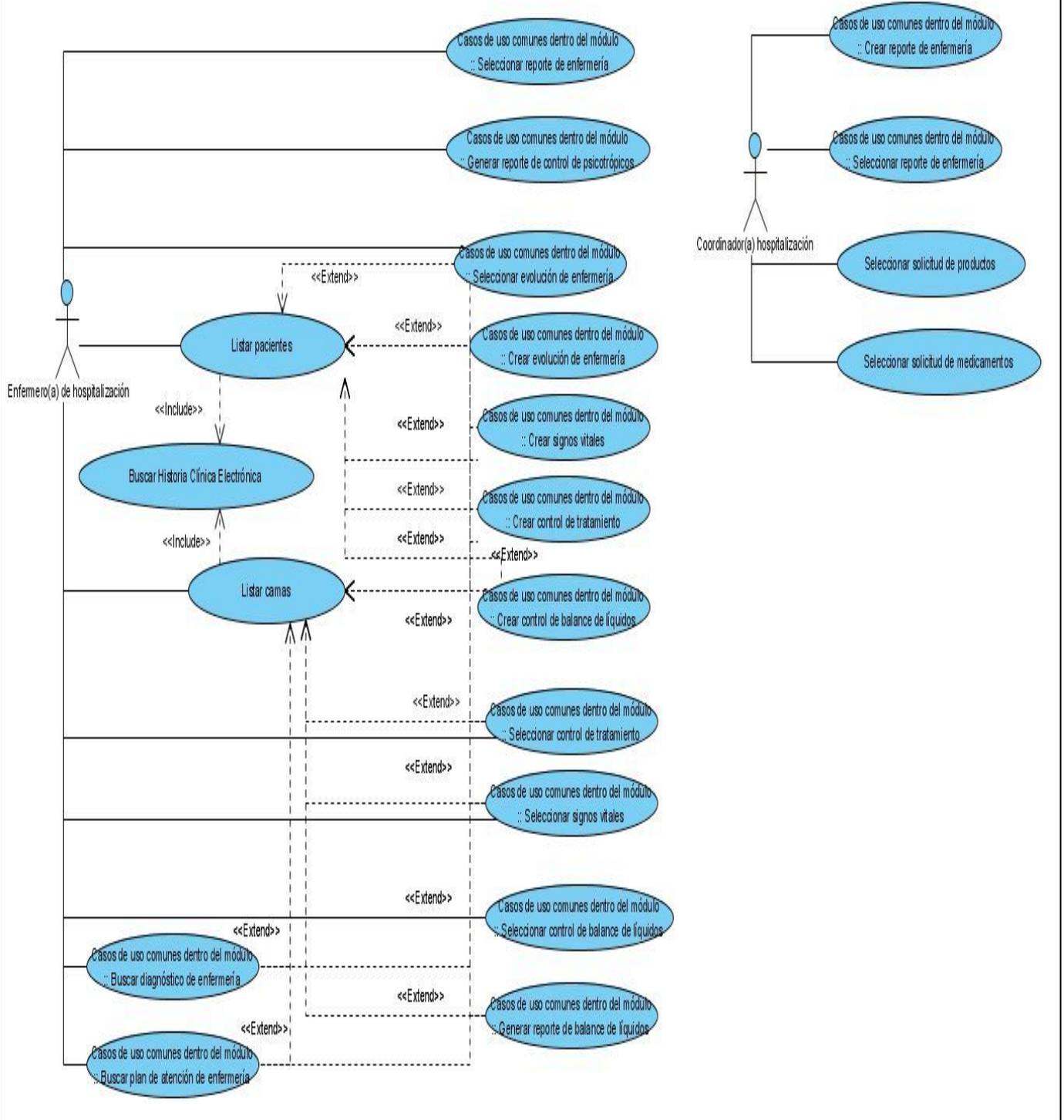


Figura 2.4 Diagrama de casos de uso Atender paciente en hospitalización

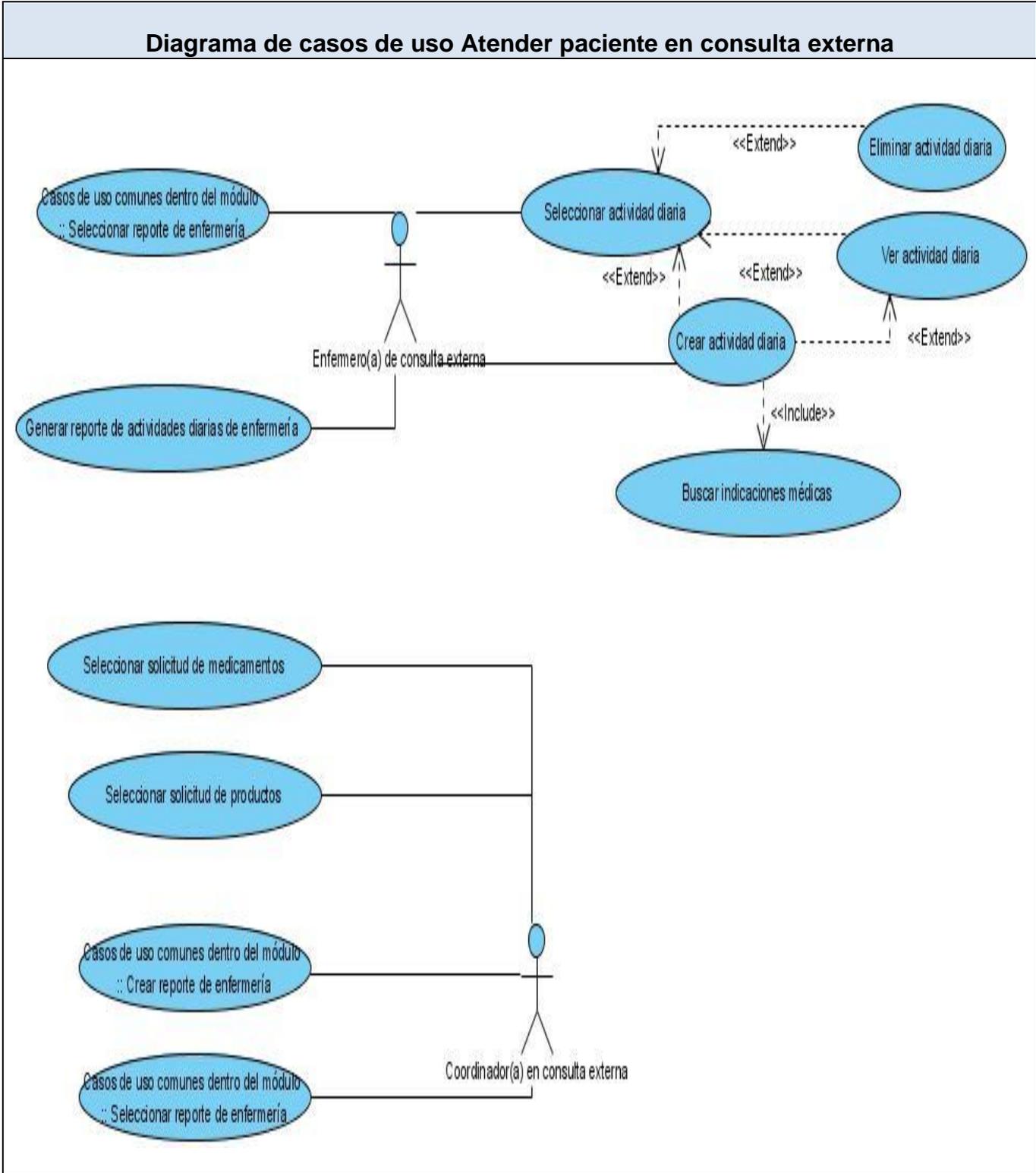


Figura 2.6 Diagrama de casos de uso Atender paciente en consulta externa

En los procesos de negocio identificados, existen actividades comunes, por lo que se decidió agrupar estas funcionalidades como casos de uso comunes dentro del módulo.

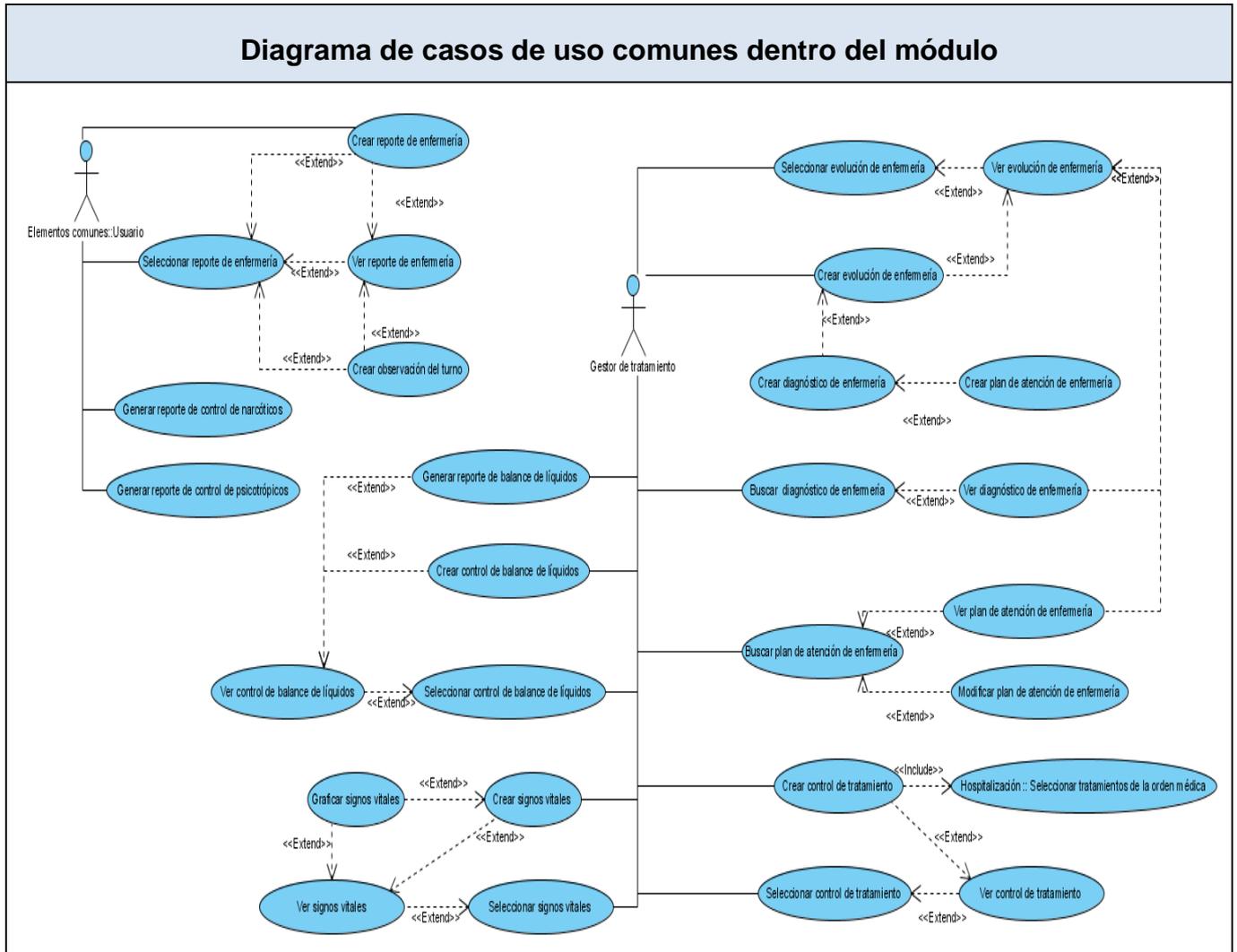


Figura 2.7 Diagrama de casos de uso Casos de uso comunes dentro del módulo

2.4.4 Descripción textual de los casos de uso

En los diferentes procesos identificados, la mayoría de las actividades que se realizan son las mismas, por lo que los casos de uso arquitectónicamente significativos son comunes en los diferentes procesos. A continuación se describen algunos de estos casos de uso.

CASO DE USO:	Crear evolución de enfermería
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear evolución de enfermería, el sistema brinda la posibilidad de introducir los datos para crear la evolución de enfermería, el actor introduce los datos, el sistema crea la evolución de enfermería, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	No existen
Postcondiciones:	Se creó una evolución de enfermería
REFERENCIAS	
Requisitos:	19

Tabla 2.3. Descripción del caso de uso Crear evolución de enfermería

CASO DE USO:	Buscar evolución de enfermería
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar evolución de enfermería, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la evolución de enfermería deseada. El actor introduce los datos que considera como criterios para buscar la evolución de enfermería, el sistema busca y muestra las evoluciones de enfermería que cumplen con los criterios de búsqueda, el actor selecciona la evolución de enfermería deseada, el caso de uso termina.
Complejidad:	Media
Precondiciones:	No existen
Postcondiciones:	Se buscó la evolución de enfermería dado criterios.
REFERENCIAS	
Requisitos:	21

Tabla 2.4. Descripción del caso de uso Buscar evolución de enfermería

CASO DE USO:	Ver evolución de enfermería
Resumen:	El caso de uso inicia cuando el actor selecciona una evolución de enfermería y accede a la opción de Ver evolución de enfermería, el sistema muestra los datos de la evolución de enfermería, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	Para ver los datos de una evolución de enfermería, esta debe estar

	seleccionada.
Postcondiciones:	Se vieron los datos de una evolución de enfermería.
REFERENCIAS	
Requisitos:	20

Tabla 2.5. Descripción del caso de uso Ver evolución de enfermería

CASO DE USO:	Modificar evolución de enfermería
Resumen:	El caso de uso inicia cuando el actor accede a la opción Modificar evolución de enfermería, el sistema brinda la posibilidad de introducir los campos a modificar de la evolución, el actor introduce nuevos datos, el sistema actualiza la evolución de enfermería, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	Se debe seleccionar un paciente para ver la evolución de enfermería.
Postcondiciones:	Se actualiza la evolución de enfermería
REFERENCIAS	
Requisitos:	22

Tabla 2.6. Descripción del caso de uso Modificar evolución de enfermería

CASO DE USO:	Ver diagnóstico de enfermería.
Resumen:	El caso de uso inicia cuando el actor selecciona un diagnóstico de enfermería y accede a la opción de Ver diagnóstico de enfermería, el sistema muestra los datos del diagnóstico de enfermería, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	Para ver los datos de un diagnóstico de enfermería, este debe estar seleccionado.
Postcondiciones:	Se vieron los datos de un diagnóstico de enfermería.
REFERENCIAS	
Requisitos:	39

Tabla 2.7. Descripción del caso de uso Ver diagnóstico de enfermería

CASO DE USO:	Buscar diagnóstico de enfermería.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar diagnóstico de enfermería, el sistema brinda la posibilidad de introducir

	<p>criterios de búsqueda para localizar diagnósticos de enfermería, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los diagnósticos de enfermería que cumplen con los criterios de búsqueda, el caso de uso termina.</p>
Complejidad:	Media
Precondiciones:	No existen
Postcondiciones:	Se buscó diagnóstico de enfermería dado criterios.
REFERENCIAS	
Requisitos:	40

Tabla 2.8. Descripción del caso de uso Buscar diagnóstico de enfermería

CASO DE USO:	Ver plan de atención de enfermería
Resumen:	<p>El caso de uso inicia cuando el actor selecciona un plan de atención de enfermería y accede a la opción de Ver plan de atención de enfermería, el sistema muestra los datos del plan de atención de enfermería, el caso de uso termina.</p>
Complejidad:	Baja
Precondiciones:	Para ver los datos de un plan de atención de enfermería, este debe estar seleccionado.
Postcondiciones:	Se vieron los datos de un plan de atención de enfermería.
REFERENCIAS	
Requisitos:	43

Tabla 2.9. Descripción del caso de uso Ver plan de atención de enfermería

CASO DE USO:	Buscar plan de atención de enfermería
Resumen:	<p>El caso de uso inicia cuando el actor accede a la opción Buscar plan de atención de enfermería, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el plan de atención de enfermería, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los planes de atención de enfermería que cumplen con los criterios de búsqueda, el caso de uso termina.</p>
Complejidad:	Media
Precondiciones:	No existen
Postcondiciones:	Se buscó plan de atención de enfermería dado criterios.

REFERENCIAS	
Requisitos:	40

Tabla 2.10. Descripción del caso de uso Buscar plan de atención de enfermería

CASO DE USO:	Crear control de tratamiento
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear control de tratamiento, el sistema brinda la posibilidad de introducir los datos para crear el control de tratamiento, el actor introduce los datos del control de tratamiento, el sistema crea el control de tratamiento, el caso de uso termina.
Complejidad:	Alta
Precondiciones:	No existen
Postcondiciones:	Se creó un control de tratamiento
REFERENCIAS	
Requisitos:	24

Tabla 2.11. Descripción del caso de uso Crear control de tratamiento

CASO DE USO:	Buscar control de tratamiento
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar control de tratamiento, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el control de tratamiento deseado, el actor introduce los datos que considera como criterios para buscar el control de tratamiento, el sistema busca y muestra los controles de tratamientos que cumplen con los criterios de búsqueda, el actor selecciona el control de tratamiento deseado, el caso de uso termina.
Complejidad:	Media
Precondiciones:	No existen
Postcondiciones:	Se creó un control de tratamiento
REFERENCIAS	
Requisitos:	26

Tabla 2.12. Descripción del caso de uso Buscar control de tratamiento

CASO DE USO:	Ver control de tratamiento
Resumen:	El caso de uso inicia cuando el actor selecciona un control de tratamiento

	y accede a la opción de Ver control de tratamiento, el sistema muestra los datos del control de tratamiento, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	Debe estar seleccionado el control de tratamiento.
Postcondiciones:	Se vieron los datos de un control de tratamiento.
REFERENCIAS	
Requisitos:	25

Tabla 2.13. Descripción del caso de uso Ver control de tratamiento

CASO DE USO:	Crear signos vitales
Resumen:	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, el caso de uso termina.
Complejidad:	Media
Precondiciones:	No existen
Postcondiciones:	Se crearon signos vitales
REFERENCIAS	
Requisitos:	28

Tabla 2.14. Descripción del caso de uso Crear signos vitales

CASO DE USO:	Modificar signos vitales
Resumen:	El caso de uso inicia cuando el actor selecciona, desde la vista signos vitales, la opción Modificar signos vitales, el sistema brinda la posibilidad de modificar los datos que se deseen, el actor introduce los datos de los signos vitales, el sistema guarda los cambios, el caso de uso termina.
Complejidad:	Media
Precondiciones:	Se debe seleccionar un paciente para ver los signos vitales.
Postcondiciones:	Se actualizan los signos vitales
REFERENCIAS	
Requisitos:	32

Tabla 2.15. Descripción del caso de uso Modificar signos vitales

Conclusiones

En este capítulo se realizó un estudio del flujo actual de los procesos de negocio asociados al campo de acción, determinándose cuales de las actividades de estos procesos son objeto de automatización y describiéndose por último los casos de uso que se identificaron como arquitectónicamente significativos.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En este capítulo se describen los elementos más importantes correspondientes a la etapa del diseño del sistema, utilizando para su modelado el Lenguaje Unificado de Modelación (UML), que permite representar la expansión de los casos de uso y el modelo conceptual. También se presentarán los diagramas de interacción y los diagramas de clases del diseño del sistema.

3.1 Descripción de la arquitectura

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. Representa básicamente la estructura de un sistema y comprende los principales componentes del mismo, sus propiedades externamente visibles y las relaciones entre ellos. [43]

Para el desarrollo del sistema se ha escogido una arquitectura basada en el patrón MVC, descrito en el capítulo 1. Este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos: el modelo, la vista y el controlador. Al aplicarse el patrón se obtiene un diseño que desacopla la vista del modelo y que permite la reusabilidad de los componentes, además de que las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

La vista está formada por páginas XHTML, contenedoras de formularios que mediante controles JSF y RichFaces recogen, validan y muestran los datos que el usuario provee y solicita en cada una de las operaciones que realiza. Para los controles JSF básicos se utiliza Ajax4Jsf. Todo esto combinado con Facelets, que elimina la necesidad de dos motores de renderización (uno para JSF y otro para JSP). Por otra parte los controles para interfaz de usuario de Seam, adicionan varias mejoras a JSF, desde validación, expresiones EL extendidas, integración de la navegación en la interfaz de usuario basada en pageflows o procesos del negocio. [44]

El controlador está integrado por clases controladoras, que encierran la lógica del negocio, a las cuales, mediante anotaciones que provee el framework Seam, se les especifica el contexto en que se encuentran (conversacional, evento, página), los cuales definen el estado de los datos y las entidades que manejan. Este framework integra transparentemente la administración de procesos del negocio vía jBoss JBPM y mediante Drools se manejan las reglas del negocio. [45]

En el modelo se utiliza para el acceso a datos, la implementación de JPA de Hibernate , minimizando por un lado las configuraciones en XML débilmente tipeadas y por otro lado usando los servicios del

contenedor de EJB3, se elimina gran parte del código “infraestructural” en cuanto a transacciones, la transmisión del contexto de persistencia, entre otros.

3.2 Modelo de Diseño

En este flujo se modela el sistema, con la vista arquitectónica definida, de forma tal que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen.

El eje de este flujo, es la elaboración de los diagramas de interacción. Dichos diagramas muestran gráficamente la comunicación entre objetos, con la finalidad de cumplir los requerimientos especificados. Los diagramas de interacción no son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Los diagramas de interacción, son además la realización de los diagramas de clases del diseño. Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones, que se utilizan para modelar la vista de diseño estática de un sistema. [46]

Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Para realizar un diseño con un grado alto, en la calidad del mismo, se aplicaron patrones de diseño durante la realización de los diagramas antes expuestos, permitiendo asignar responsabilidades a objetos y diseñar la colaboración entre ellos. [47]

Los patrones de diseño, no son más que la descripción de un problema y la solución del mismo, de forma que se pueda utilizar en diferentes contextos dando respuesta a interrogantes comunes.

Un importante papel en el diseño realizado, lo tuvieron los patrones generales de software para asignación de responsabilidades (GRASP, en inglés). De este grupo de patrones se tiene al patrón Experto y al Creador que al utilizarlos, se tuvo como resultado que a cada clase del diseño, se le asignaran tareas acorde a la información que dicha clase poseía, además de poder crear instancias acorde a la responsabilidad que estas tenían. Con esto se logró conservar el encapsulamiento, pues los objetos logran valerse de su propia información para realizar lo que se les pide.

Para lograr una mayor organización en el diseño del sistema, se han organizado las clases y diagramas por paquetes, obteniéndose el diagrama de paquetes que se muestra a continuación.

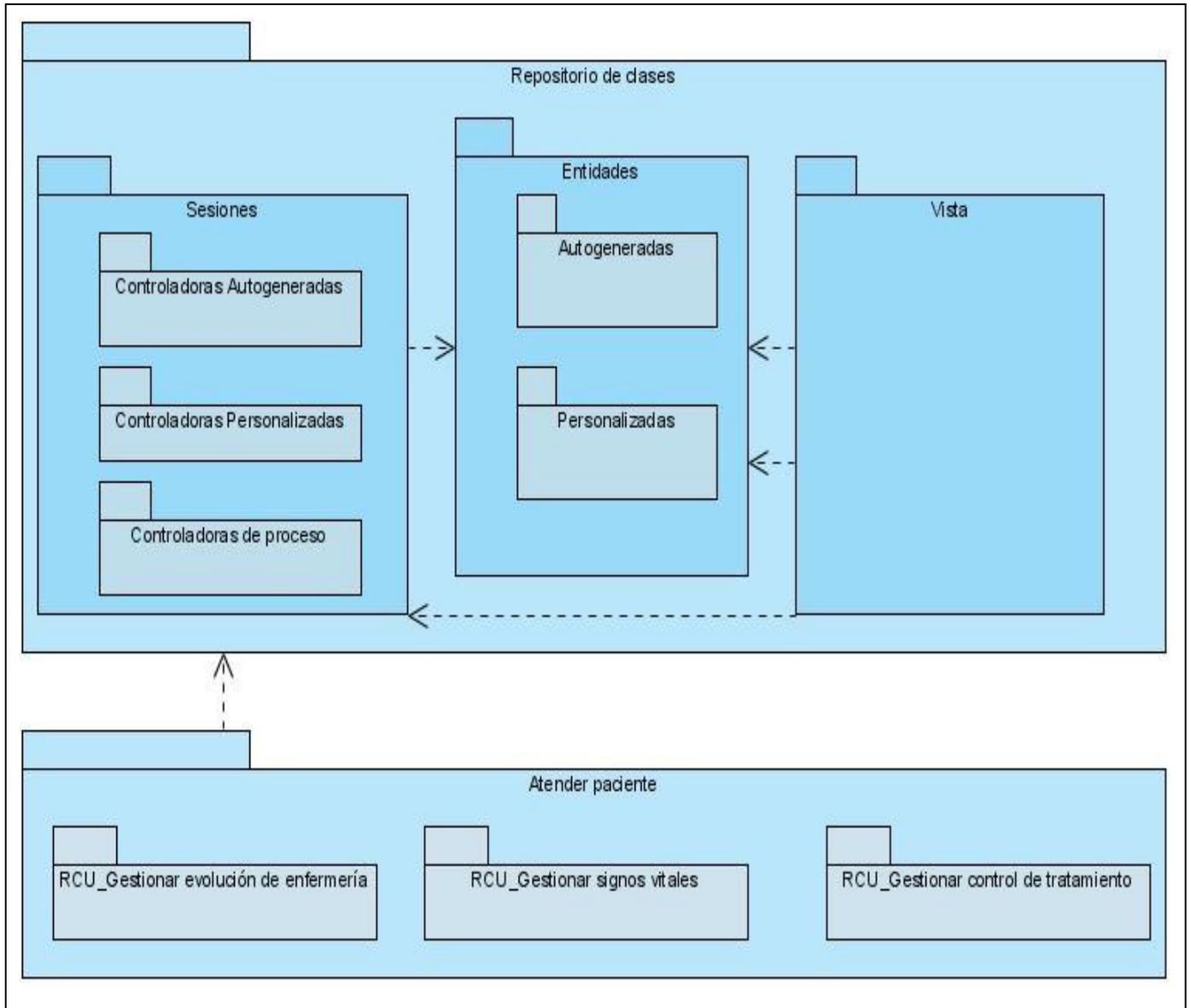


Figura 3.1 Diagrama de Paquetes

Para comprender el esquema de describen los paquetes:

Repositorio de clases: Contendrá todas las clases definidas en el diseño de acuerdo a la tecnología, que serán usadas en la implementación del módulo.

Dentro del repositorio se cuenta un paquete para las entidades, otro para las sesiones y otro para las vistas.

El paquete de **entidades**, se contienen las entidades autogeneradas de la base de datos, así como las que han sufrido cambios por la lógica del negocio, que son las personalizadas.

En el paquete de **sesiones**, se encuentran las clases controladoras autogeneradas por el entorno de desarrollo, las clases controladoras personalizadas que son modificadas y las clases controladoras propias del proceso.

Por último en el paquete de las **vistas** se encuentran distribuidas por procesos y casos de uso las páginas que conforman la interfaz de usuario.

Las vistas invocan a las sesiones enviándole eventos de entrada y datos, los cuales son gestionados en estas y modifican las entidades. Las entidades por su parte son consultadas desde las vistas, obteniendo los datos ya actualizados desde las entidades.

Se representa además el paquete correspondiente al proceso, y dentro del paquete se representan los paquetes que contienen las realizaciones de los casos de usos, agrupadas en gestionar, en dependencia de la entidad.

Para cada realización del caso de uso, se modela un diagrama de clases del diseño utilizándose 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. [48]

Las páginas servidoras se encargan de construir o generar el resultado XHTML que conforma las páginas cliente, mediante la relación <<build>>, estas contiene los formularios que envían sus datos a la página servidora para ser procesados los pedidos mediante una relación de <<submit>>. Entre las páginas clientes pueden existir vínculos (<<link>>) o redireccionamientos (<<redirect>>). Las páginas servidoras se relacionan con las clases controladoras mediante el <<include>>, permitiéndole a estas últimas realizar todas las operaciones sobre las entidades, las cuales le permiten a los formularios actualizarse cuando ocurra algún cambio.

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. Las entidades pueden reconocerse mediante los estereotipos <<Entity Bean>> y <<ORM Persistable>>.

Para cada escenario de dicho diagrama es modelado un diagrama de interacción. Como diagrama de interacción fue seleccionado el de secuencia. A continuación se muestran algunos de estos diagramas. Para consultar los restantes Ver Anexo 1.

Gestionar evolución de enfermería

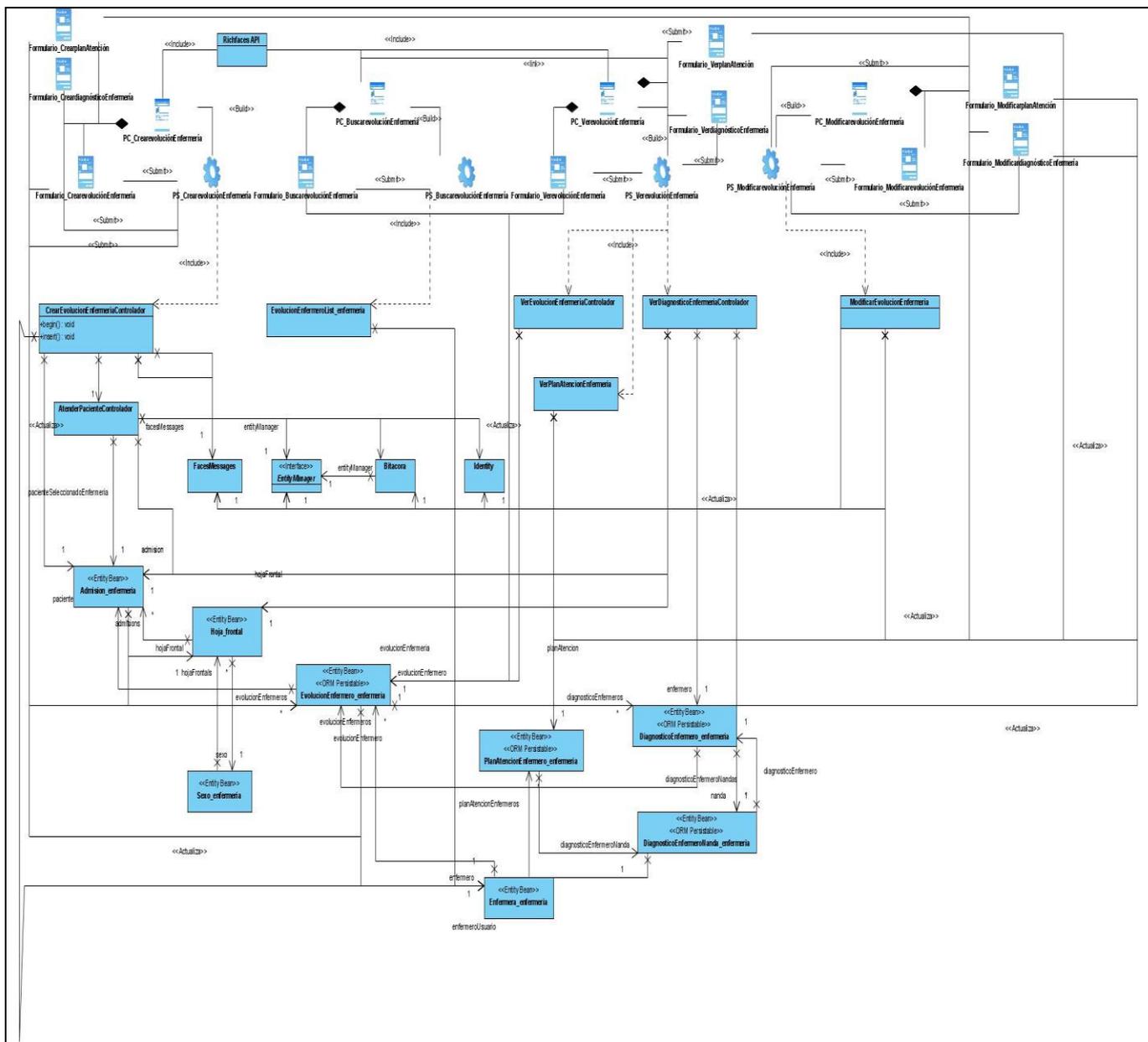


Figura 3.2 Diagrama de clases del diseño Gestionar evolución de enfermería

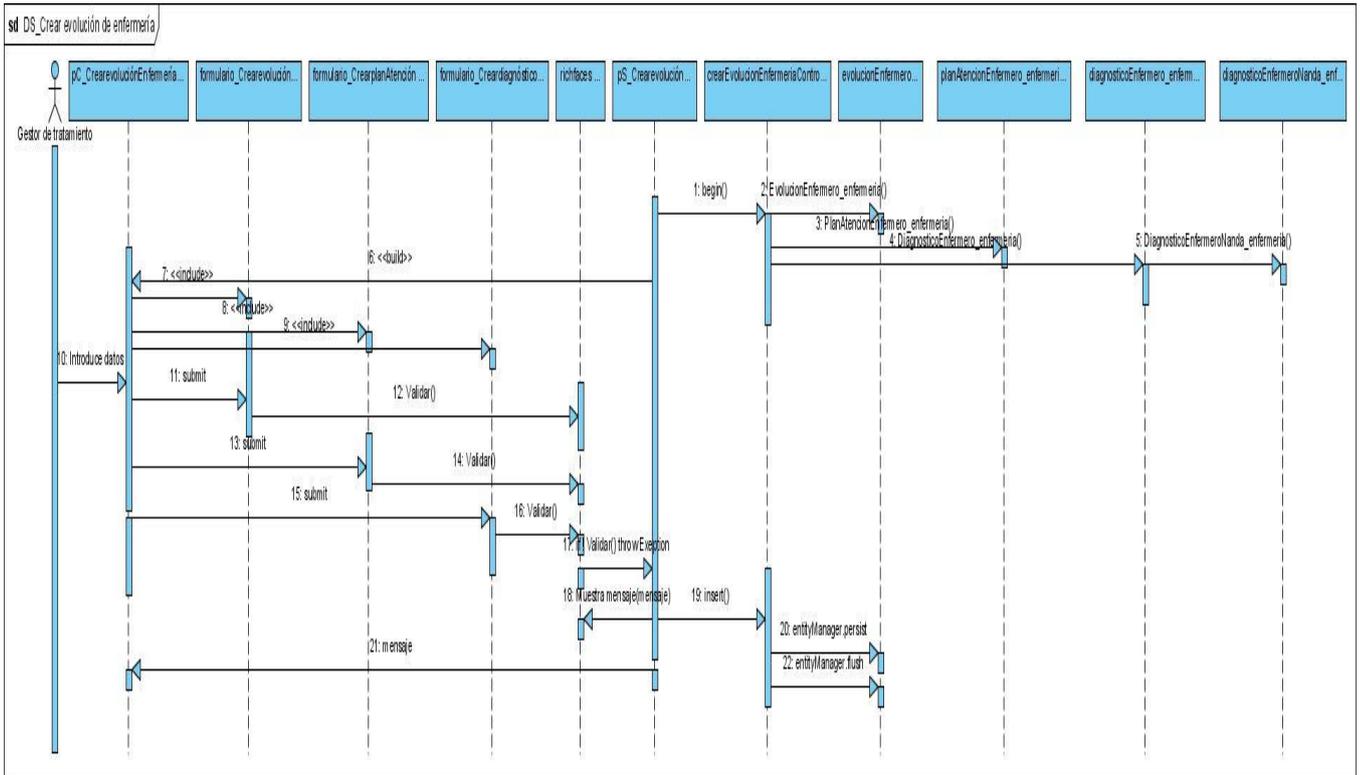


Figura 3.3 Diagrama de secuencia escenario Crear evolución de enfermería

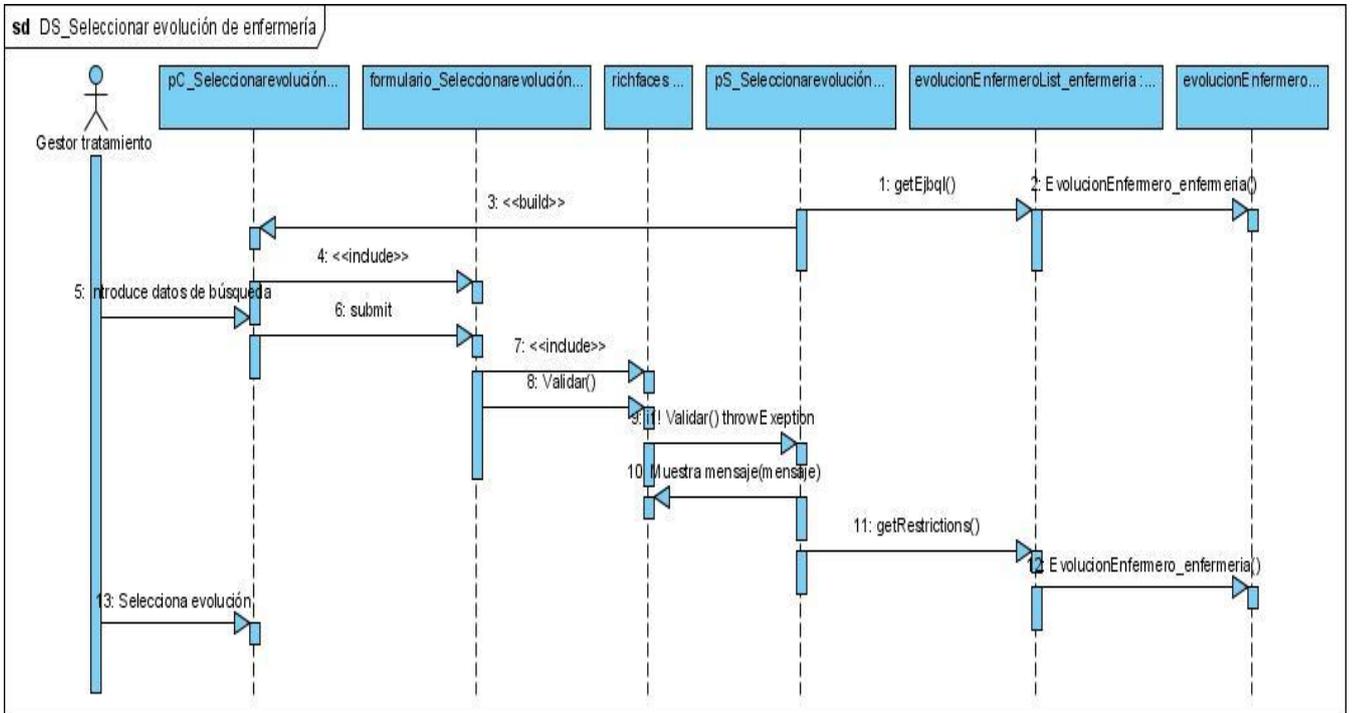


Figura 3.4 Diagrama de secuencia escenario Seleccionar evolución de enfermería

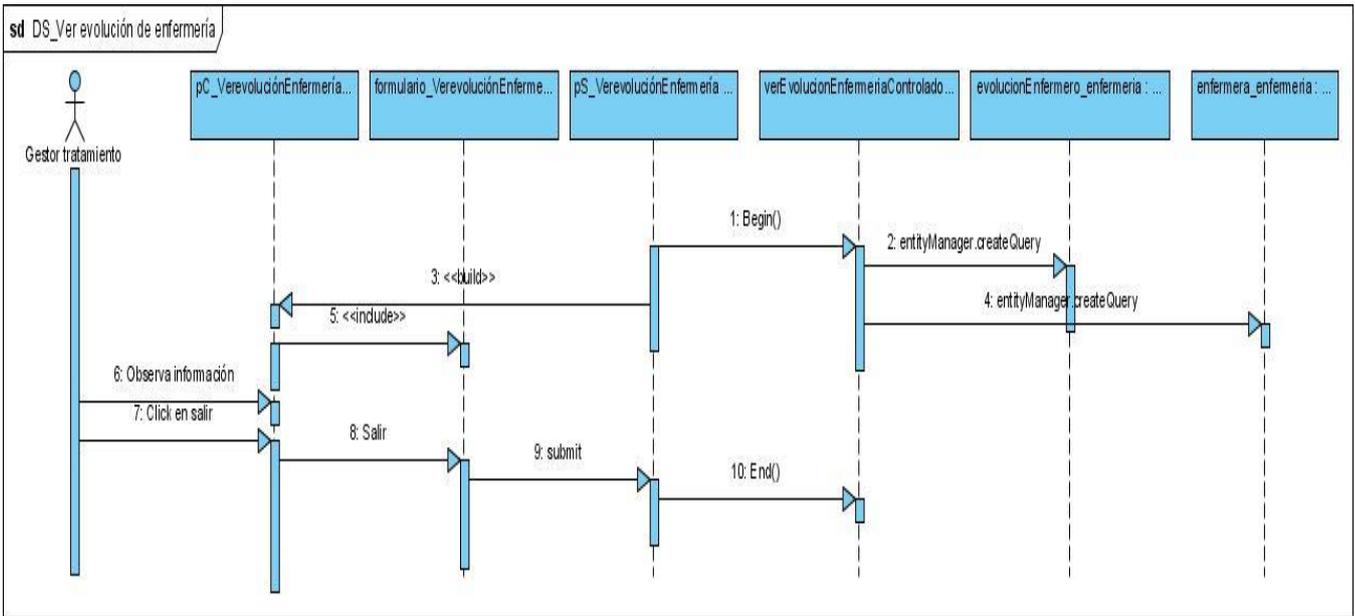


Figura 3.5 Diagrama de secuencia escenario Ver evolución de enfermería

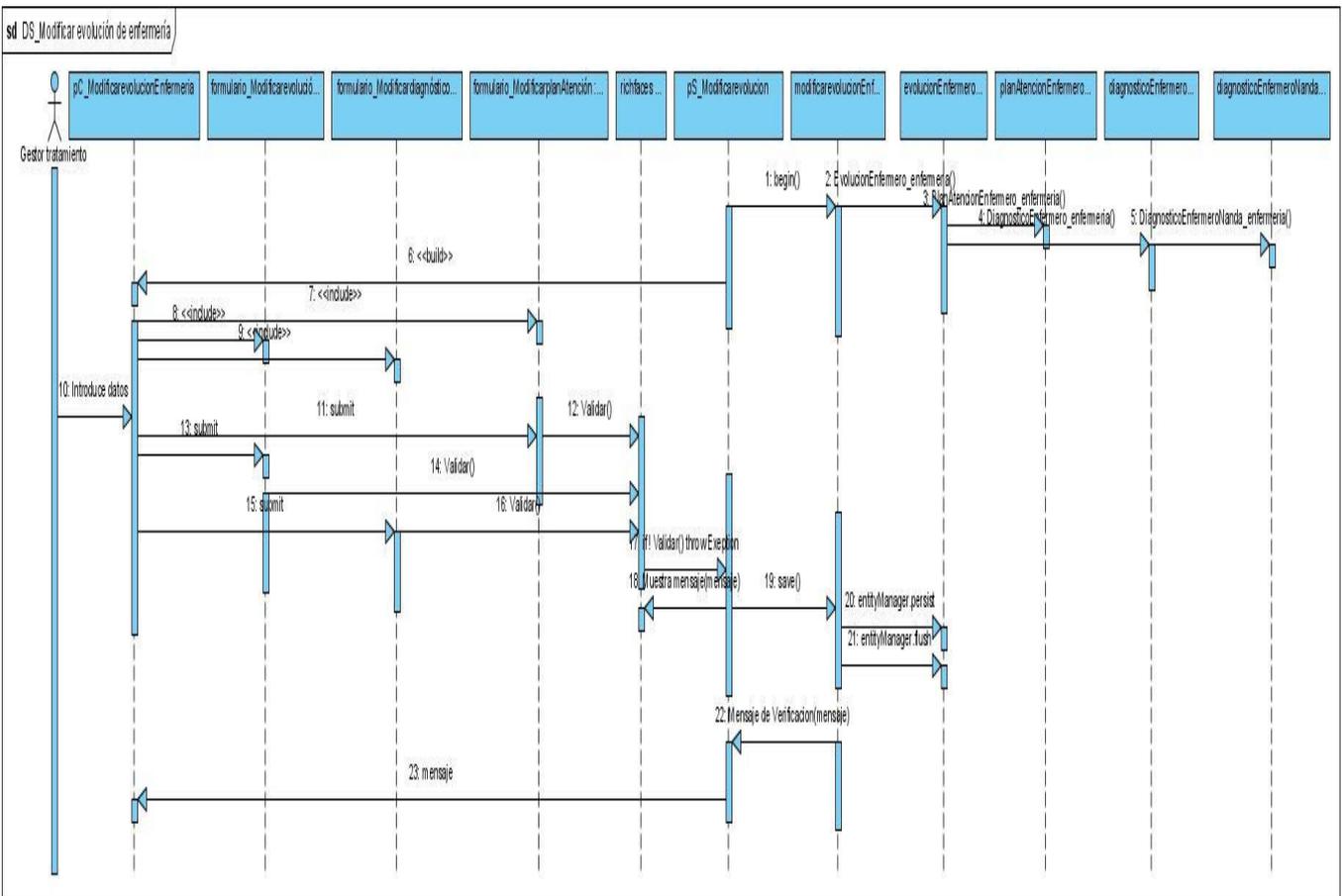


Figura 3.6 Diagrama de secuencia escenario Modificar evolución de enfermería

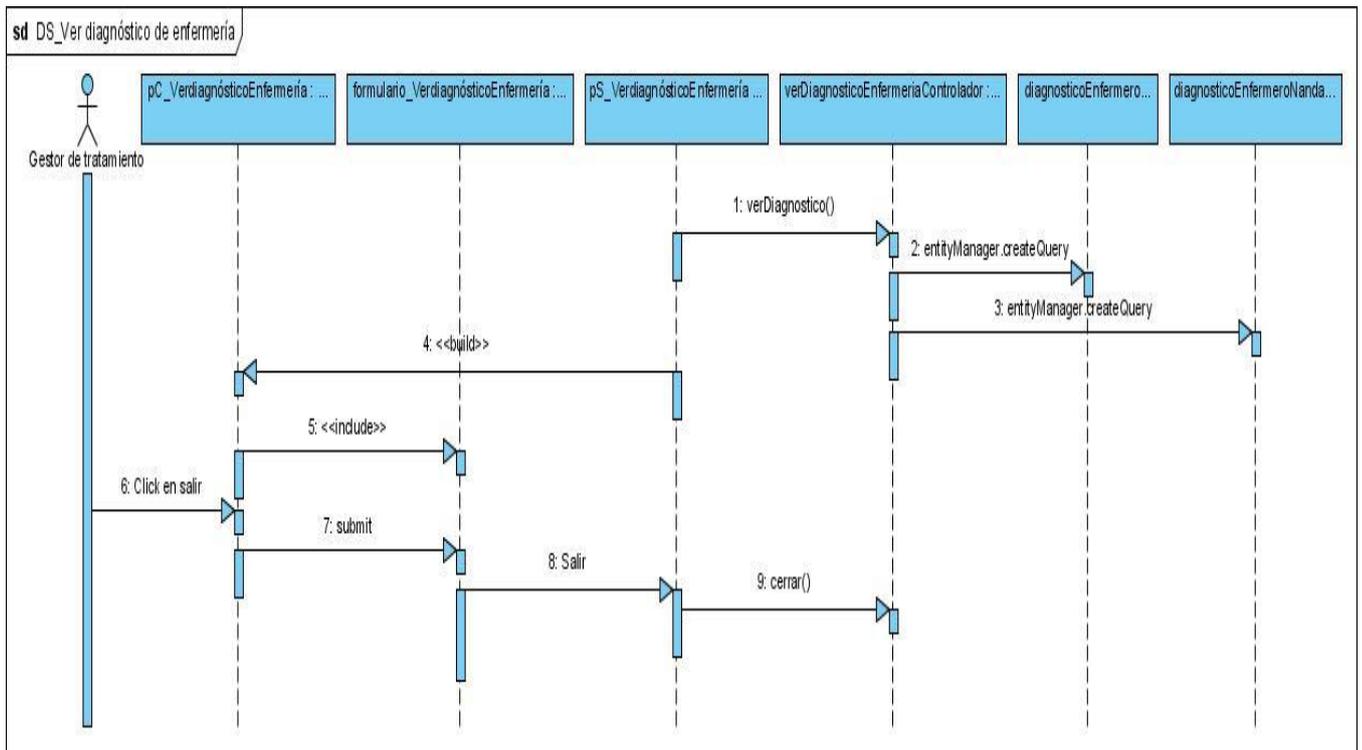


Figura 3.7 Diagrama de secuencia escenario Ver diagnóstico de enfermería

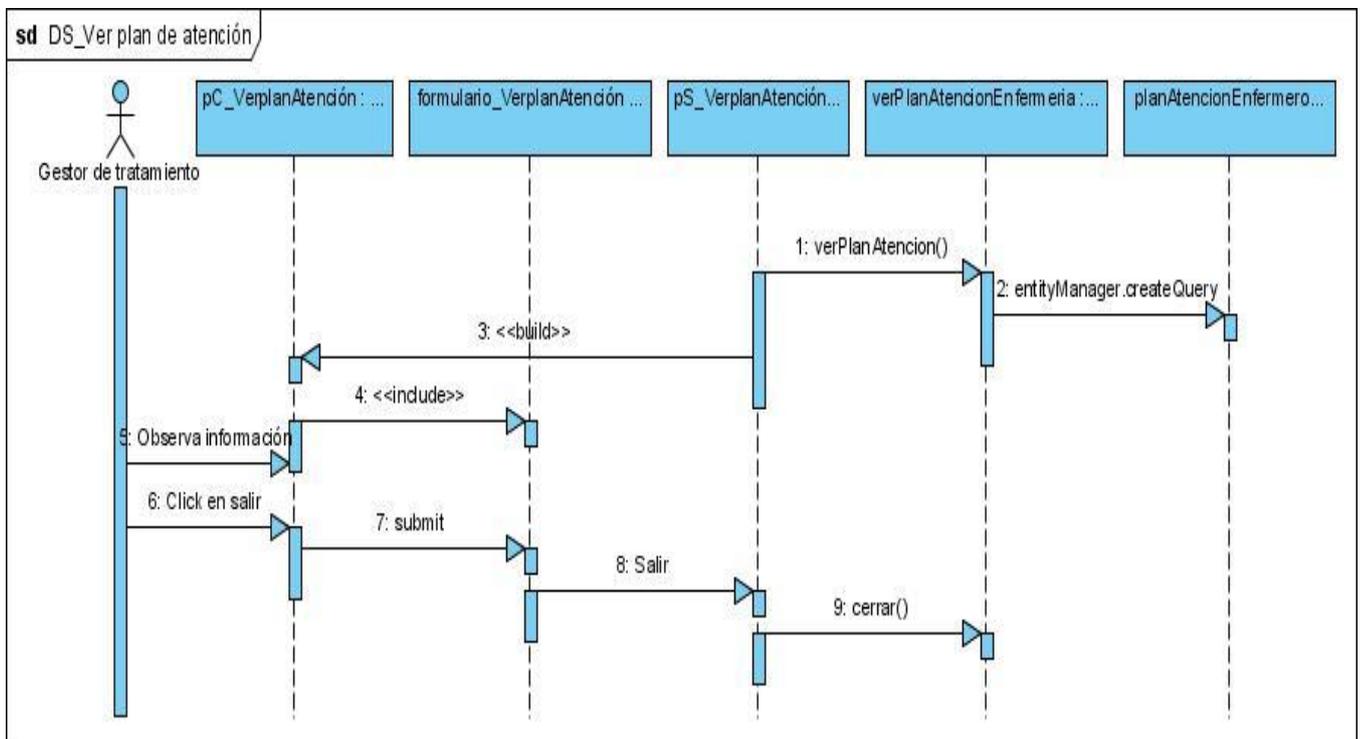


Figura 3.8 Diagrama de secuencia escenario Ver plan de atención de enfermería

Gestionar signos vitales

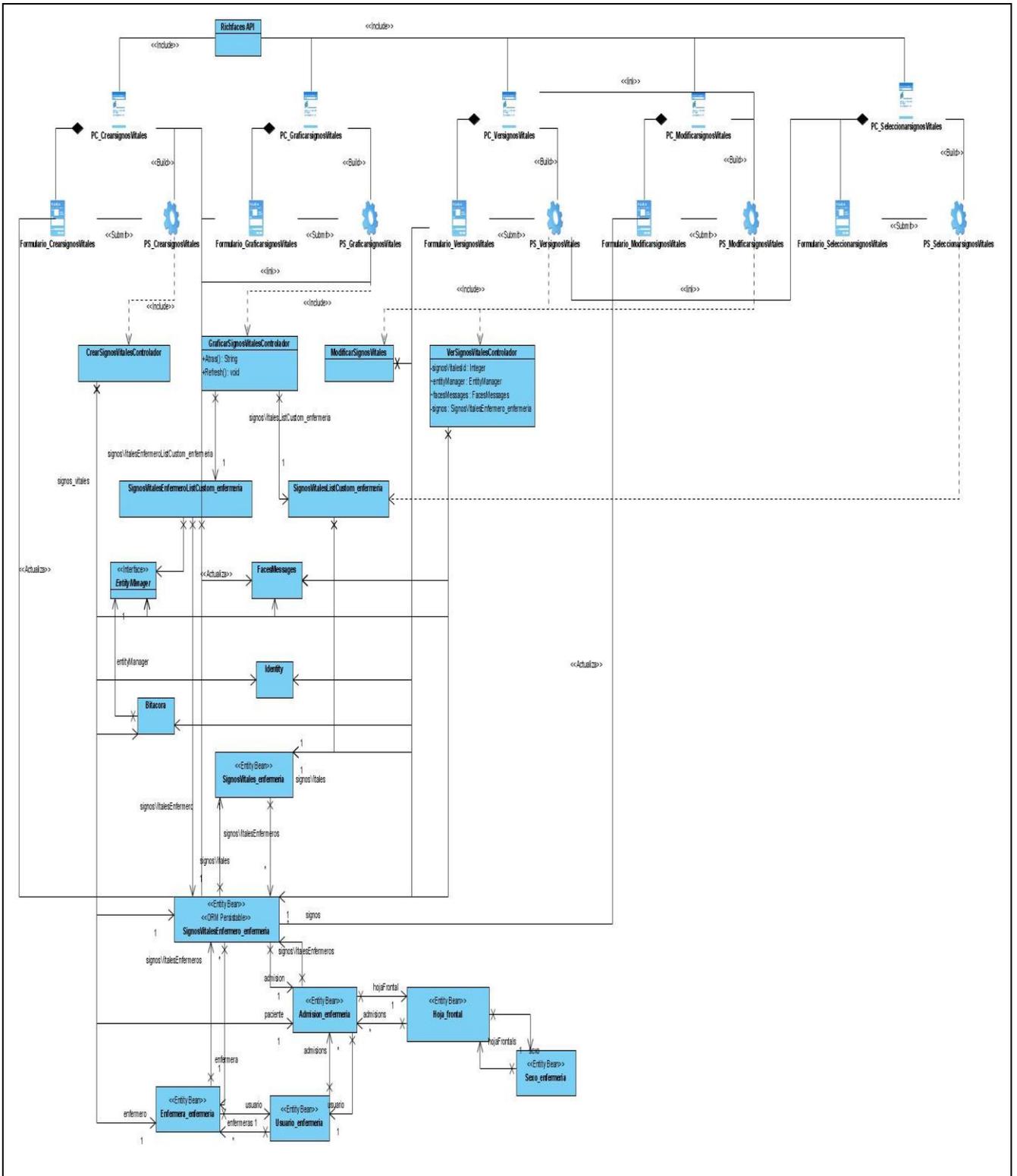


Figura 3.9 Diagrama de clases del diseño Gestionar signos vitales

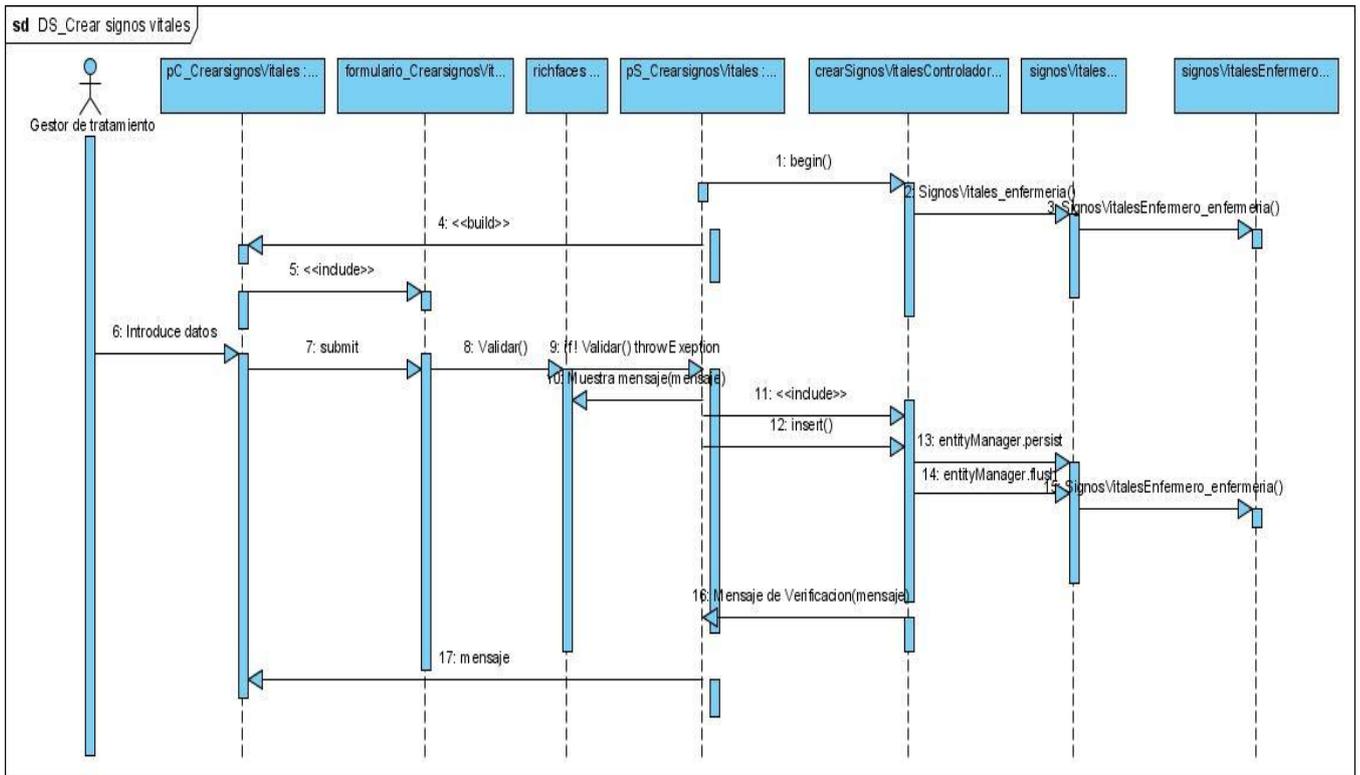


Figura 3.10 Diagrama de secuencia escenario Crear signos vitales

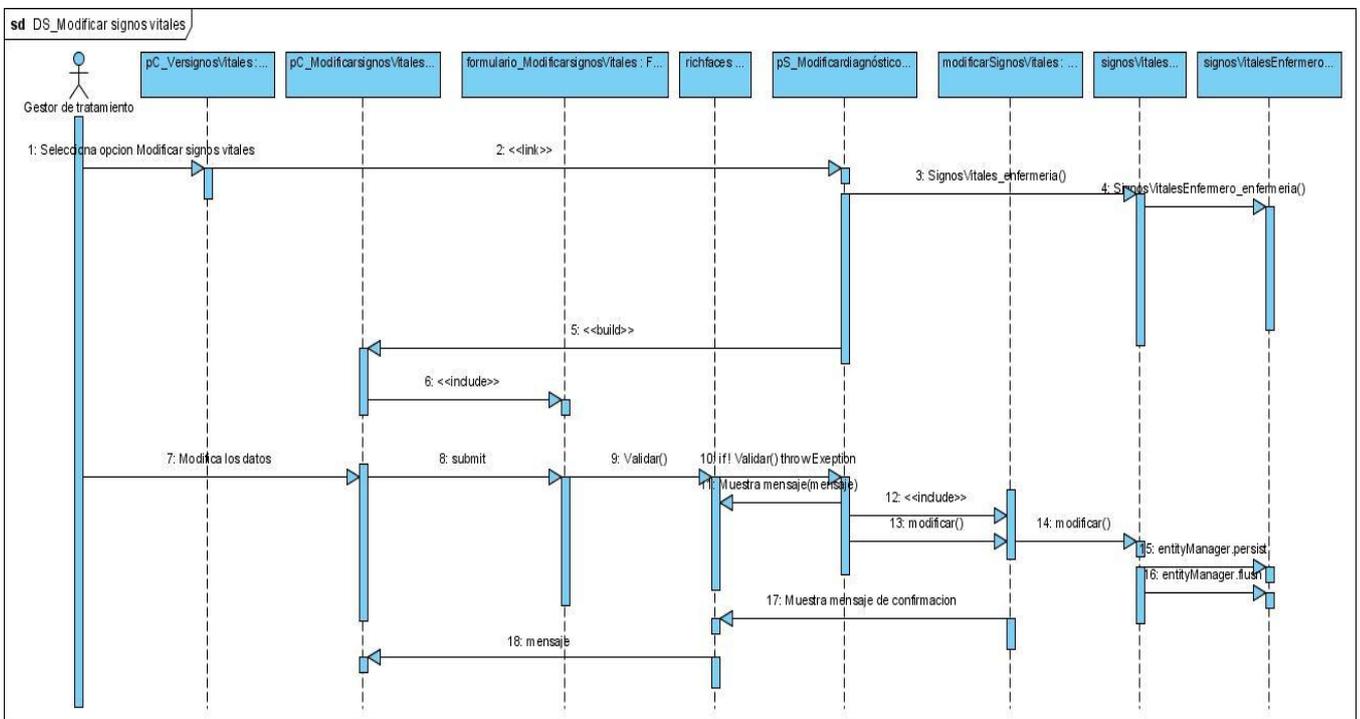


Figura 3.11 Diagrama de secuencia escenario Modificar signos vitales

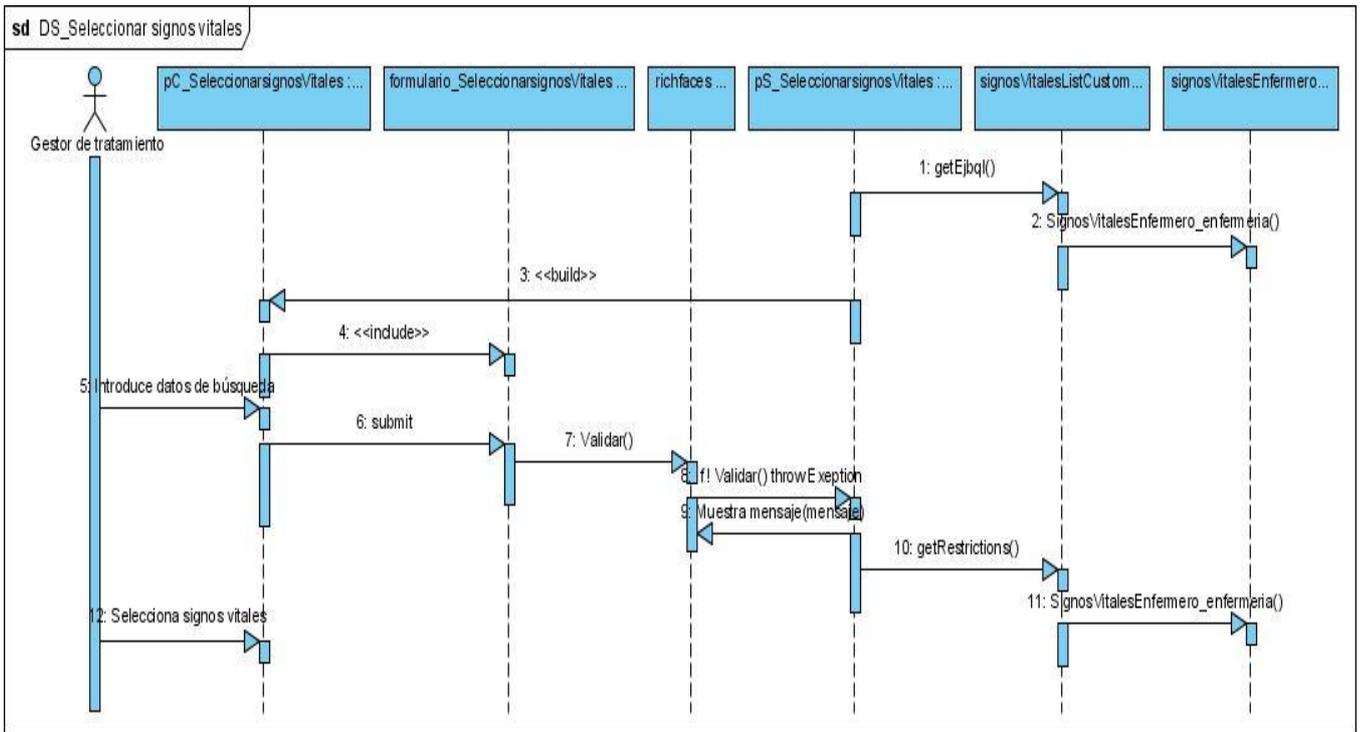


Figura 3.12 Diagrama de secuencia escenario Seleccionar signos vitales

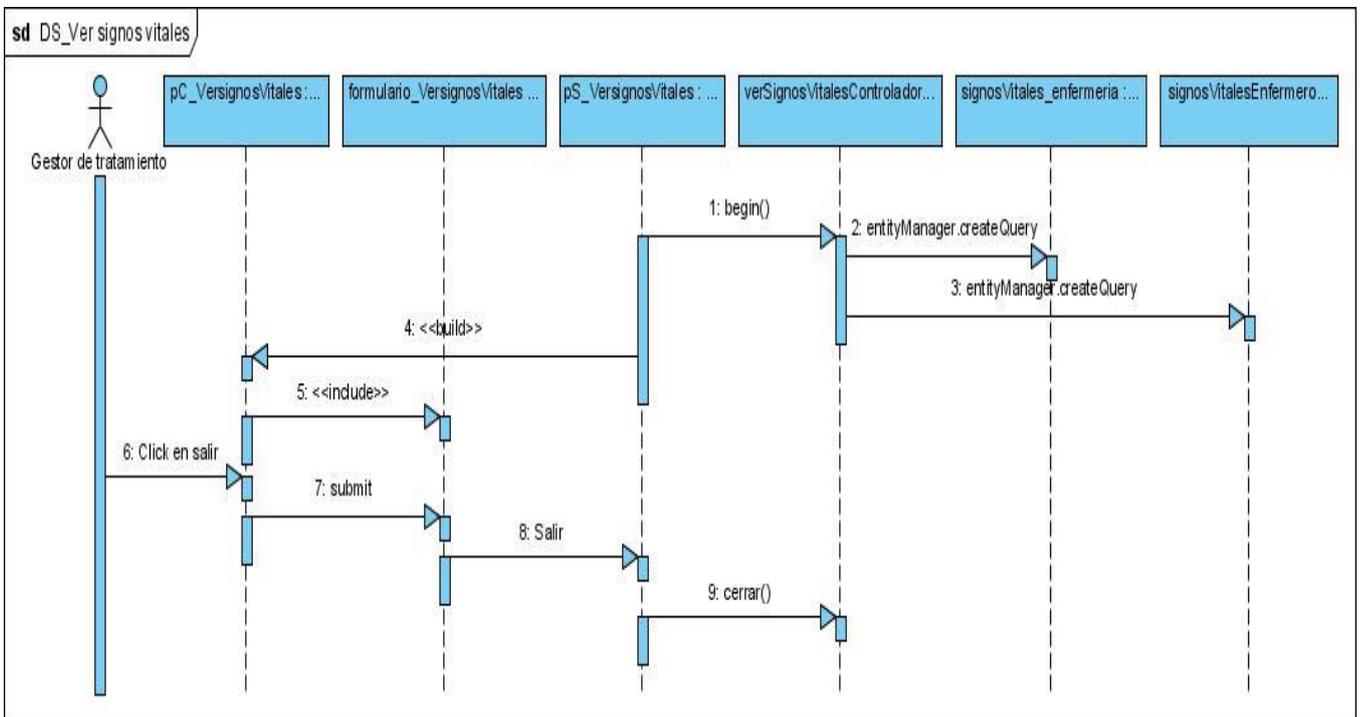


Figura 3.13 Diagrama de secuencia escenario Ver signos vitales

Gestionar Control de tratamiento

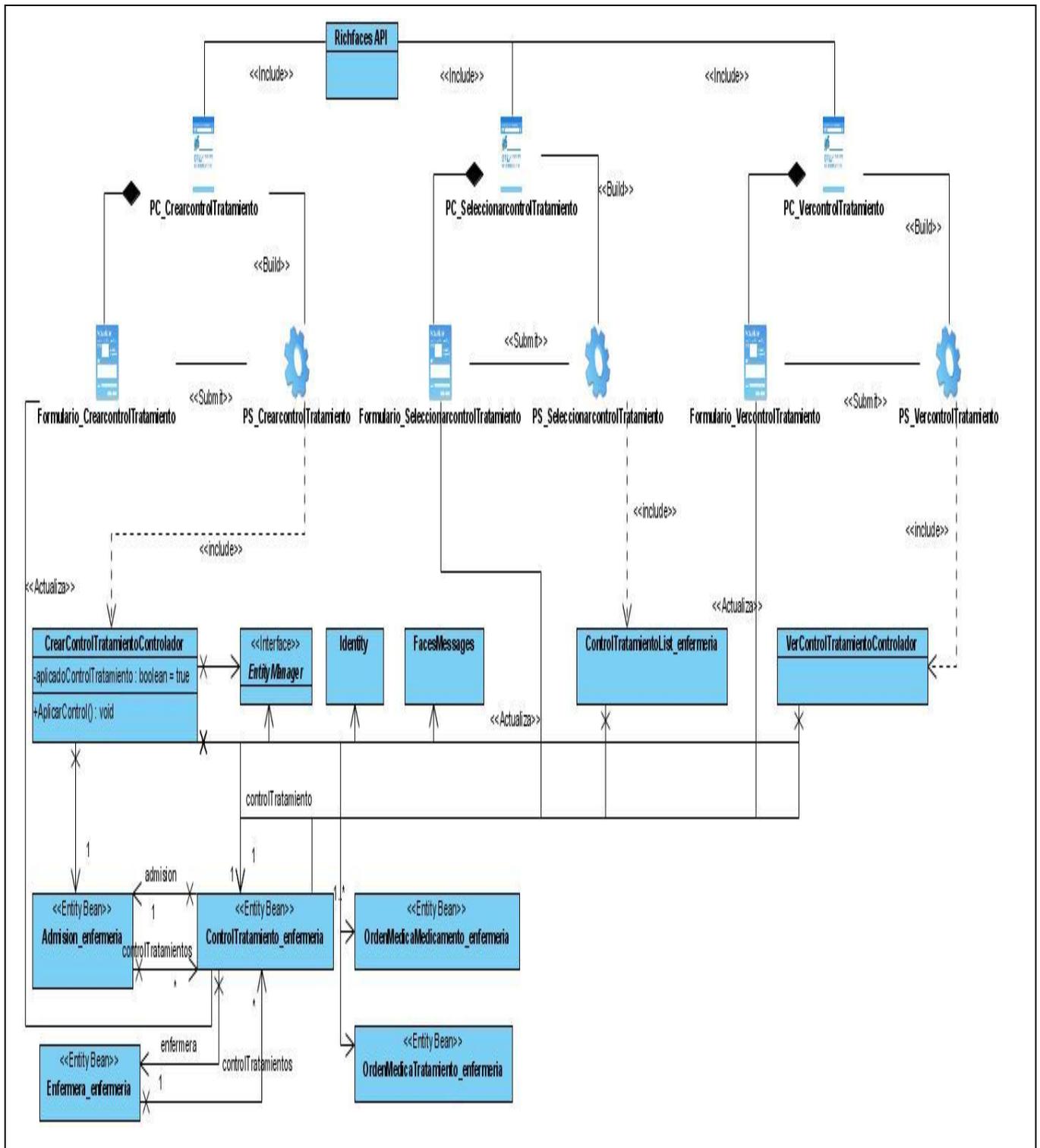


Figura 3.14 Diagrama de clases del diseño Gestionar control de tratamiento

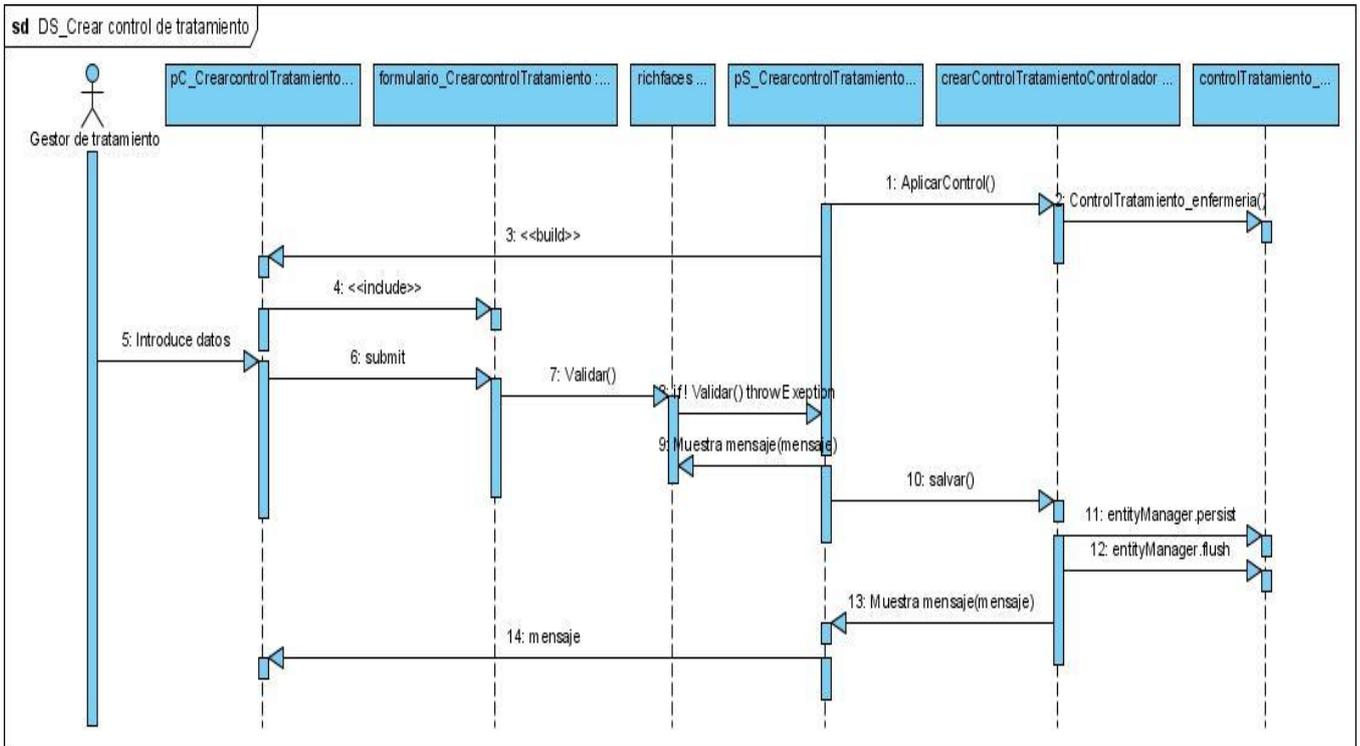


Figura 3.15 Diagrama de secuencia Crear control de tratamiento

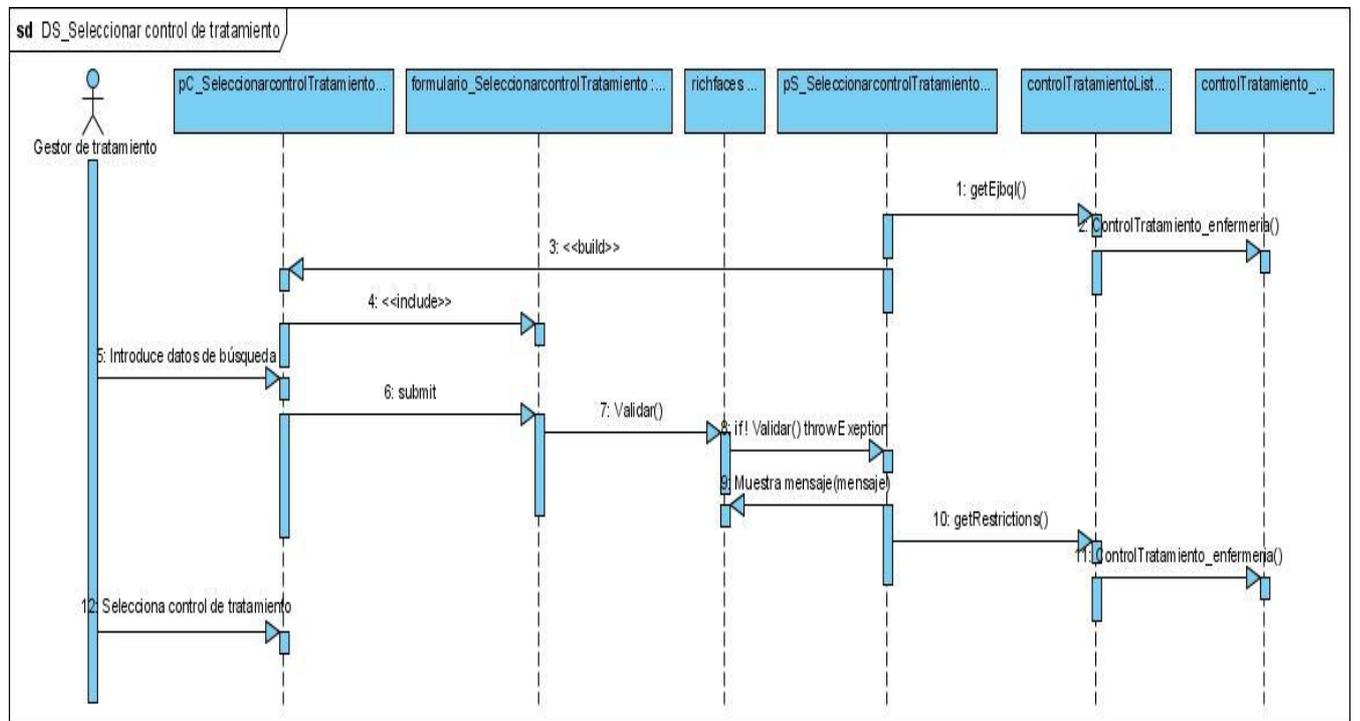


Figura 3.16 Diagrama de secuencia Seleccionar control de tratamiento

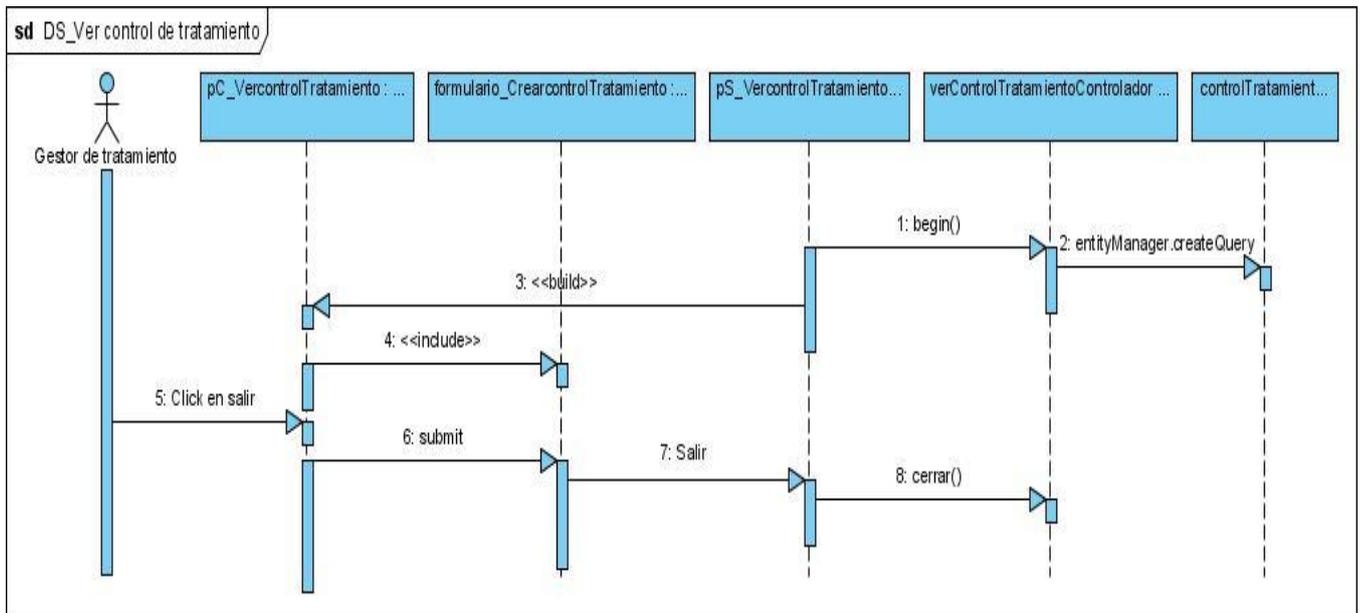


Figura 3.17 Diagrama de secuencia ver control de tratamiento

3.2.1 Descripción de las clases del diseño

A continuación serán explicadas algunas de las clases que han sido identificadas para la futura implementación del sistema. Del mismo modo serán descritas algunas de las responsabilidades que realizarán las páginas servidoras que responden a la Lógica de Negocio, lográndose una mayor comprensión mayor del funcionamiento que tendrá el sistema en desarrollo.

3.2.1.1. Descripción de las páginas clientes

Nombre: PC_CrearevoluciónEnfermería
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero la creación de una evolución de enfermería.

Tabla 3.1. Descripción de la clase PC_CrearevoluciónEnfermería

Nombre: PC_BuscarevoluciónEnfermería
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero la búsqueda de evoluciones de enfermería de un paciente.

Tabla 3.2. Descripción de la clase PC_BuscarevoluciónEnfermería

Nombre: PC_VerevoluciónEnfermería
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero la visualización de una evolución de enfermería de un paciente.

Tabla 3.3. Descripción de la clase PC_VerevoluciónEnfermería

Nombre: PC_CrearsignosVitales
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero crear el registro de las mediciones de signos vitales de un paciente

Tabla 3.4. Descripción de la clase PC_CrearsignosVitales

Nombre: PC_ModificarsignosVitales
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero modificar el registro de las mediciones de signos vitales de un paciente

Tabla 3.5. Descripción de la clase PC_ModificarsignosVitales

Nombre: PC_VersignosVitales
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero la visualización del registro de las mediciones de signos vitales de un paciente

Tabla 3.6. Descripción de la clase PC_VersignosVitales

Nombre: PC_CrearcontrolTratamiento
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero crear el registro de control de tratamiento de un paciente.

Tabla 3.7. Descripción de la clase PC_CrearcontrolTratamiento

Nombre: PC_VercontrolTratamiento
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al enfermero visualizar el registro de control de tratamiento de un paciente.

Tabla 3.8. Descripción de la clase PC_VercontrolTratamiento

3.2.1.2. Descripción de las páginas servidoras

Nombre: PS_ CrearevoluciónEnfermería
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ CrearevoluciónEnfermería 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_ CrearevoluciónEnfermería. 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 de la clase PS_ CrearevoluciónEnfermería

Nombre: PS_ VerevoluciónEnfermería
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ VerevoluciónEnfermería es una página que se ejecuta del lado del servidor en la Capa de Presentación. Esta recibe y valida los datos que se envían desde la página cliente PC_ VerevoluciónEnfermería. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para la búsqueda del objeto a visualizar.

Tabla 3.10. Descripción de la clase PS_ VerevoluciónEnfermería

Nombre: PS_ CrearsignosVitales
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ CrearsignosVitales 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_ CrearsignosVitales. 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 de la clase PS_ CrearsignosVitales

Nombre: PS_ VersignosVitales
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ VersignosVitales 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_ VersignosVitales. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para la búsqueda del objeto a visualizar.

Tabla 3.12. Descripción de la clase PS_ VersignosVitales

Nombre: PS_ ModificarsignosVitales
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ ModificarsignosVitales 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_ ModificarsignosVitales. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para la modificación de los datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.13. Descripción de la clase PS_ ModificarsignosVitales

Nombre: PS_ CrearcontrolTratamiento
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ CrearcontrolTratamiento 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_ CrearcontrolTratamiento. 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 3.14. Descripción de la clase PS_ CrearcontrolTratamiento

Nombre: PS_ VercontrolTratamiento
Tipo de Clase: Página Servidora
Descripción General: La clase PS_ VercontrolTratamiento 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_ VercontrolTratamiento. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para la búsqueda del objeto a visualizar.

Tabla 3.15. Descripción de la clase PS_ VercontrolTratamiento

3.2.1.3. Descripción de las páginas controladoras

Nombre: CrearSignosVitalesControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
+cid	Integer
+signos	SignosVitales_enfermeria
+signosFeto	SignosVitalesFeto_enfermeria
+caracteristicaPulso	String
+ubicacionPulso	String

+caracteristicaFR	String
+media	float
Responsabilidades:	
+ begin():void	Se guardan los datos de los signos vitales en la clase.
+ insert():void	Para darle persistencia a los datos.

Tabla 3.16. Descripción de la clase CrearSignosVitalesControlador

Nombre: ModificarSignosVitalesControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
+ signosId	Integer
+ signosFetold	Integer
+ signos	SignosVitales_enfermeria
+ signosFeto	SignosVitalesFeto_enfermeria
Responsabilidades:	
+ begin():void	Se guardan los datos de los signos vitales en la clase.
+ salvar():void	Para actualizar y darle persistencia a los datos.

Tabla 3.17. Descripción de la clase ModificarSignosVitalesControlador

Nombre: VerSignosVitalesControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
-signosVitalesId	Integer
-signosFetold	Integer
-signos	SignosVitales_enfermeria
-signosFeto	SignosVitalesFeto_enfermeria
Responsabilidades:	
+setSignosVitalesId(Integer signosVitalesId):void	Dado un identificador, se busca el signo vital para poder mostrarlo en la interfaz.

Tabla 3.18. Descripción de la clase VerSignosVitalesControlador

Nombre: GestionarHojaEvolucionEnfermeriaControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
+nanda	DiagnosticoEnfermeroNanda_enfermeria
+nandaLista	List<NandaDiagnosticos_enfermeria>
+seleccionado	NandaDiagnosticos_enfermeria
+aux	NandaDiagnosticos_enfermeria
+factoresEnfermeria	List<NandaFactoresInDiagnostico_enfermeria>
+caracteristicasEnfermeria	List<NandaCaracteristicasInDiagnostico_enfermeria>
+diagnostico1	DiagnosticoEnfermero_enfermeria
-cid	Integer
-paciente	Admision_enfermeria
-enfermero	Enfermera_enfermeria
-evolucion	EvolucionEnfermero_enfermeria
+prototipoAtencion	List<NandaPrototipoAtencion_enfermeria>
+prototipoAtencionaux	List<NandaPrototipoAtencion_enfermeria>
+planAtencion	List<PlanAtencionEnfermero_enfermeria>
+nandaListaplan	List<NandaDiagnosticos_enfermeria>
+nandaListaplanPersit	List<DiagnosticoEnfermeroNanda_enfermeria>
Responsabilidades:	
+save():void	Para darle persistencia a la evolución, ya que la clase es la que controla todos los datos que incluye la evolución.
+begin():void	Para obtener todos los datos referentes al Plan de atención, Evolución enfermería y diagnostico de enfermería

Tabla 3.19. Descripción de la clase GestionarEvolucionEnfermeriaControlador

Nombre: VerEvolucionEnfermeria	
Tipo de clase: Controladora	
Atributo:	Tipo:
-evolucionId	Integer
-evolucionIdBusc	Integer
-evolucionEnfermeria	EvolucionEnfermero_enfermeria
-diagnosticoNanda	List<DiagnosticoEnfermeroNanda_enfermeria>

-diagnostico	DiagnosticoEnfermero_enfermeria
-planAtencion	PlanAtencionEnfermero_enfermeria
-listPlanAtencion	List<PlanAtencionEnfermero_enfermeria>
+aux	DiagnosticoEnfermeroNanda_enfermeria
+verfactoresEnfermeria	List<NandaFactoresInDiagnostico_enfermeria>
+vercaracteristicasEnfermeria	List<NandaCaracteristicasInDiagnostico_enfermeria>
+verprototipoAtencion	List<NandaPrototipoAtencion_enfermeria>
Responsabilidades:	
+setEvolucionId(Integer evolucionId):void	Obtiene el identificador de la evolución par hacer la búsqueda de la Evolución ,Plan de atención y el Diagnostico de enfermería

Tabla 3.20. Descripción de la clase VerEvolucionEnfermeria

Nombre: CrearControltratamientoEnfermeria	
Tipo de clase: Controladora	
Atributo:	Tipo:
+admision	Admision_enfermeria
+enfermera	Enfermera_enfermeria
+ordenMedica	OrdenMedica_enfermeria
+controlTratamiento	ControlTratamiento_enfermeria
+observacion	ControlTratamientoObservaciones_enfermeria
+controlMedicamentosOrdenados	List<ControlTratamientoMedicamentos_enfermeria>
+controlTratamientosOrdenados	List<ControlTratamientoMedico_enfermeria>
+controlNutricionOrdenados	List<ControlTratamientoNp_enfermeria>
+controlPreparacionOrdenados	List<ControlTratamientoPi_enfermeria>
+controlDietaOrdenados;	List<ControlTratamientoDieta_enfermeria>
+controlFormulaOrdenados;	List<ControlTratamientoFormula_enfermeria>
+controlPrevio	boolean
-cid	Integer
-subCid	Integer
Responsabilidades:	
+begin():void	Para capturar todos los datos en la clase.
+insert():void	Para darle persistencia al control de tratamiento.

Tabla 3.21. Descripción de la clase CrearControlTratamientoEnfermeria

Conclusiones

El modelo de diseño obtenido contiene las clases del diseño, con los atributos y métodos que serán utilizados en la implementación y mediante los diagramas de interacción, se refleja la colaboración entre las clases a través de mensajes.

CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo se define el modelo de datos, a partir del diseño propuesto. Además de desarrolla el flujo de trabajo implementación, describiéndose cómo los elementos del modelo de diseño son implementados en términos de componentes y como se organizan en el modelo de despliegue. Se definen también elementos claves para la seguridad, el tratamiento de errores y los estilos de código a utilizar.

4.1 Modelo de datos

El diseño de la base de datos se basa en el diagrama de clases persistentes, este a su vez se basa en los diagramas de clases de diseño.

Los datos reflejados en las clases entidad, obtenidas en los diferentes diagramas de clases del diseño, son los que se capturan y almacenan durante el proceso que describe la aplicación, y estas a su vez son las que componen del diagrama de clases persistentes. Para consultar diagrama de clases persistentes. Ver Anexo 2.

A partir del diagrama de clases persistentes, se realizan transformaciones, derivadas de los procesos de normalización del diseño de la base de datos, y de la selección de una serie de tablas que formarán parte del diagrama entidad relación. Para consultar el diagrama entidad relación Ver Anexo 3.

4.1.1 Descripción de las tablas

Nombre: SignosVitales			
Descripción: Contiene los registros de las mediciones de los signos vitales.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de los signos vitales
version	Integer		Número de la versión de los signos vitales
PADiastolica	float	No	Valor a almacenar de la presión diastólica
PASistolica	float	No	Valor a almacenar de la presión sistólica
PAMedia	float	No	Valor a almacenar de la presión media
circunferenciaCefalica	Float		Valor a almacenar del tamaño de la circunferencia cefálica
saturacionO2	Float		Valor a almacenar de la cantidad de O2 que posee el cuerpo del paciente
diuresisHoraria	Float		Valor a almacenar de la cantidad de diuresis

			orinaria
presionVenosaCentral	Float		Valor a almacenar de la presión venosa central
peso	Float		Valor a almacenar del peso corporal del paciente
talla	Float		Valor a almacenar de la talla del paciente
observacion	String		Valor a almacenar de la información que el enfermero desee registrar
eliminado	Boolean		Estado de los signos vitales registrados al paciente, eliminada o no
cid	Integer		Identificador de la última conversación relacionada a la tabla.
fecha	Date	No	Fecha de creación de los signos vitales
hora	Date	No	Hora en que se crearon los signos vitales
id_p_caracteristica	CaracteristicaPulso_enfermeria	No	Identificador de la tabla que contiene los valores del pulso (FK)
id_p_ubicacion	UbicacionPulso_enfermeria	No	Identificador de la tabla que contiene la ubicación del pulso(FK)
p_valor	float	No	Valor del pulso
id_f_r_caracteristicas	CaracteristicaFR_enfermeria	No	Identificador de la tabla características de la frecuencia respiratoria(FK)
f_r_valor	float	No	Valor de la frecuencia respiratoria
temperatura	float	No	Valor de la temperatura
id_signos_vitales_feto	SignosVitalesFeto_enfermeria		Identificador de la tabla que contiene los signos vitales de feto(FK)
id_hoja_frontal	HojaFrontal_enfermeria	No	Identificador de la tabla que contiene la información de la hoja frontal(FK)
id_usuario	Usuario_enfermeria	No	Identificador de la tabla que contiene los datos del usuario(FK)

Tabla 4.1. Descripción de la tabla SignosVitales

Nombre: SignosVitalesEnfermero

Descripción: Contiene la persona que realizó una medición de signos vitales en un momento

determinado.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de los signos vitales que registra el enfermero
version	Integer		Número de la versión registrada de los signos vitales
eliminado	Boolean		Estado de los signos vitales
cid	Integer		Identificador de la última conversación relacionada a la tabla. (FK)

Tabla 4.2. Descripción de la tabla SignosVitalesEnfermero

Nombre: HojaFrontal			
Descripción: Contiene todos los datos del paciente.			
Atributo:	Tipo:	Nulo:	Descripción:
numeroHc	String	No	Identificador de la historia clínica del paciente(FK)
id	Integer	No	Identificador de la hoja frontal de enfermería
version	Integer		Número de la versión de la hoja frontal de enfermería
foto	String		Foto del paciente registrado
nombres	String		Nombre del paciente
apellido1	String		Primer apellido del paciente
Apellido2	String		Segundo apellido del paciente
cedula	String		Número de identidad del paciente
idGrupoSanguineo	Integer		Grupo sanguíneo del paciente
idRaza	Integer		Raza del paciente
idEtnia	Integer		Etnia a la que pertenece el paciente
idNacion	Integer		Nación a la que pertenece el paciente
idEstadoCivil	Integer		Estado civil en que se encuentra el paciente
correoElectronico	String		Correo electrónico del paciente
avisarEnEmergencia	String		Nombre del familiar o tutor que se desee avisar en caso de emergencias
telfEmergencia	String		Número telefónico al que se debe llamar en caso de emergencia
idLugarNacimiento	Integer		Lugar en que nació el paciente
idDireccionParticular	Integer		Dirección particular del paciente
idLugarResidencia	Integer		Lugar en que reside el paciente

telfFijo	String		Número telefónico o del lugar de residencia del paciente
telfCelular	String		Número telefónico del celular del paciente
idPeriodoNeonatal	Integer		Identificador del periodo neonatal del paciente
idAlimentacion	Integer		Identificador de la alimentación que necesita el paciente(FK)
idAntSexReprod	Integer		Identificador de los antecedentes sexuales del paciente(FK)
idDatosMadre	Integer		Identificador de los datos de la madre del paciente(FK)
idDatosPadre	Integer		Identificador de los datos del padre del paciente(FK)
idDatosRepresentante	Integer		Identificador de los datos del representante del paciente(FK)
idFactorSanguineo	Integer		Identificador del factor sanguínea del paciente(FK)
idAntGinecoObstetrico	Integer		Identificador de los antecedentes gineco obstetricia del paciente(FK)
profesion	String		Profesión del paciente
idTipoHojaFrontal	Integer		Identificador del tipo de hoja frontal del paciente(FK)
idAntHematoOnc	Integer		Identificador de los antecedentes hemato oncológicos del paciente(FK)
eliminado	Boolean		Estado de la hoja frontal de enfermería
cid	Integer		Identificador de la última conversación realizada a la tabla(FK)
fechaNacimiento	Date		Fecha de nacimiento del paciente
idAntPrenatObst	Integer		Identificador de la tabla que contienen los datos de los antecedentes prenatales Obstetricios(FK)
idDesarrollo	Integer		Identificador de la tabla que contiene los datos del desarrollo(FK)
idAntSexReprod	Integer		Identificador de la tabla que contiene los antecedentes sexuales de reproducción(FK)
estaEmbarazada	Boolean		Muestra si el paciente está o no embarazada
descripcionDesconocido	String		Muestra la descripción que emite el enfermero del paciente si este es desconocido
nombresFoneticos	String		Nombre por el que llaman al paciente

apellido1Fonetico	String		Apellido por el que llaman al paciente
apellido2Fonetico	String		Apellido por el que llaman al paciente

Tabla 4.3. Descripción de la tabla HojaFrontal

Nombre: PlanAtencionEnfermero			
Descripción: Contiene los datos del plan de atención de enfermería para el paciente, dado un diagnóstico de enfermería.			
Atributo:	Tipo:	Nulo:	Descripción:
idDiagnostico	Integer	No	Identificador del plan de atención de enfermería
nandaPrototipoAtencion	NandaPrototipoAtencion_enfermeria	No	Identificador de la tabla que contiene los prototipos de atención(FK)
diagnosticoEnfermeroNanda	DiagnosticoEnfermeroNanda_enfermeria		Identificador de la tabla que contiene los diagnósticos nanda(FK)
version	Integer		Número de la versión del plan de atención de enfermería
objetivosEsperados	String		Breve descripción de los objetivos que se esperan, para realizar el plan de atención de enfermería
accionEnfermero	String		Acción que realice el enfermero
resultado	String		Resultado que se obtiene del plan de atención de enfermería
eliminado	Boolean		Estado del plan de atención de enfermería
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.4. Descripción de la tabla PlanAtencionEnfermero

Nombre: IngresoPostoperatorio			
Descripción: Contiene los datos del ingreso del paciente a post-operatorio.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del ingreso a post operatorio
version	Integer		Número de la versión del ingreso a post

			operatorio
fecha	Date	No	Fecha en la que se realizó el ingreso a post operatorio
hora	Date	No	Hora en la que se realizó el ingreso a post operatorio
eliminado	Boolean		Estado del ingreso
cid	Integer		Identificador de la última conversación realizada a la tabla
servicioQuirurgico	ServicioQuirurgico_enfermeria	No	Identificador de la tabla que contiene los datos del servicio quirúrgico(FK)
hojaFrontal	HojaFrontal_enfermeria	No	Identificador de la tabla que contiene los datos de la hoja frontal(FK)

Tabla 4.5. Descripción de la tabla IngresoPostoperatorio

Nombre: DiagnosticoMedico			
Descripción: Contiene los datos del diagnóstico emitido por el médico al paciente.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del diagnóstico médico
version	Integer		Número de la versión del diagnóstico médico
eliminado	Boolean		Estado del diagnóstico médico
cid	Integer		Identificador de la última conversación realizada a la tabla
fechaEmision	Date	No	Fecha en la que se emitió el diagnóstico médico
horaEmision;	Date	No	Hora en la que se emitió el diagnóstico médico

Tabla 4.6. Descripción de la tabla DiagnosticoMedico

Nombre: EvolucionEnfermero			
Descripción: Contiene los datos de la evolución.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de la evolución de enfermería
version	Integer		Número de la versión de la evolución de enfermería
fecha	Date	No	Fecha en la que se realiza la evolución de enfermería
hora	Date	No	Hora en la que se realiza la evolución de enfermería

observacion	String		Observación realizada por el enfermero
eliminado	Boolean		Estado de la evolución de enfermería
cid	Integer		Identificador de la última conversación realizada a la tabla
admision	Admision_enfermeria	No	Identificador de la tabla que contiene los datos de la admisión(FK)
enfermera	Enfermera_enfermeria	No	Identificador de la tabla que contiene los datos de la enfermera(FK)

Tabla 4.7. Descripción de la tabla EvolucionEnfermero

Nombre: DiagnosticoEnfermero			
Descripción: Contiene el registro de quien hizo el diagnóstico según el estándar de diagnósticos de enfermería.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del diagnostico de enfermería
version	Integer		Número de la versión del diagnóstico de enfermería
fecha	Date	No	Fecha en la que se realizó el diagnóstico de enfermería
hora	Date	No	Hora en la que se realizó el diagnóstico de enfermería
observacion	String		Observación realizada por el enfermero
eliminado	Boolean		Estado del diagnóstico de enfermería
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.8. Descripción de la tabla DiagnosticoEnfermero

Nombre: Admision			
Descripción: Contiene los datos del paciente con el que se está trabajando, o sea los que no han sido egresados.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de la admisión de enfermería
version	Integer		Número de la version de la admisión de enfermería
fecha	Date	No	Fecha en la que se realice la admisión de enfermería

idCama	Integer		Identificador de la entidad cama (FK)
observaciones	String		Observaciones realizadas en la admisión de enfermería
eliminado	Boolean		Estado de la admisión de enfermería
idEstadoEvolucionPaciente	Integer		Identificador de la entidad evolución del paciente(FK)
cid	Integer		Identificador de la última conversación realizada a la tabla
ordenAdmision	OrdenAdmision_enfermeria	No	Identificador de la tabla que contiene los datos de la orden de admisión del paciente(FK)
usuario	Usuario_enfermeria	No	Identificador de la tabla que contiene los datos del usuario(FK)
estadoEvolucionPaciente;	EstadoEvolucionPaciente_enfermeria		Identificador de la tabla que contiene los datos del estado de evolución del paciente(FK)
hojaFrontal	HojaFrontal_enfermeria	No	Identificador de la tabla que contiene los datos de la hoja frontal(FK)
pesoIngreso;	BigDecimal		Valor del peso de paciente
numeroCaso	Integer		Valor del caso
hora	Date		Hora en que se realiza la admision

Tabla 4.9. Descripción de la tabla Admision

Nombre: SolicitudIntervencion			
Descripción: Contiene los datos de las solicitudes que son aprobadas, las cuales constituyen el plan de cirugías.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de la solicitud de intervención
version	Integer		Número de la versión de la solicitud de intervención
fechaSolicitud	Date	No	Fecha en la que fue realizada la solicitud de intervención

hora	Date	No	Hora en la que fue realizada la solicitud de intervención
electiva	Boolean		Identifica si un paciente es de emergencia o no
duracionAproximada	Date		Tiempo que dura la intervención
fechaFijada	Date	No	Fecha en la que se realizará la intervención
eliminado	Boolean		Estado de la solicitud de intervención
cid	Integer		Identificador de la última conversación realizada a la tabla
aprobada	Boolean		Muestra si la solicitud de intervención fue aprobada o no
emergencia	Boolean		
causaModificacionQx	String		Describe las causas por las cuales o por la cual ha sido modificada la cirugía
otrosAnestesiologos	String		Nombre de los anestesiólogos que intervendrán en la solicitud de intervención
otrasEnfermeras	String		Nombre de las enfermeras que intervendrán en la solicitud de intervención
id_hoja_frontal	HojaFrontal_enfermeria	No	Identificador de la tabla que contiene los datos de la hoja frontal(FK)
causa_rechazo_solicitud	String		Descripción del porque ha sido rechazada la solicitud
id_procedimiento	CipProcedimiento_enfermeria	No	Identificador de la tabla que contiene los datos del procedimiento(FK)
id_diagnostico	DiagnosticoMedico_enfermeria		Identificador de la tabla que contiene los datos del diagnostico(FK)
id_servicio_quirurgico	ServicioQuirurgico_enfermeria	No	Identificador de la tabla que contiene los datos del servicio quirúrgico(FK)
realizada	Boolean		Valor de la solicitud si fue realizada o no
usuario	Usuario_enfermeria	No	Identificador de la tabla que contiene los datos del usuario(FK)

Tabla 4.10. Descripción de la tabla SolicitudIntervencion

Nombre: EgresoPostoperatorio			
Descripción: Contiene los datos de los egresos realizados en las salas de pos-operatorio			
Atributo:	Tipo;	Nulo:	Descripción:
idEgreso	Integer	No	Identificador del egreso pos-operatorio
version	Integer		Número de la versión de los líquidos ingresados al paciente en venas
ingresoPostoperatorio	IngresoPostoperatorio_enfermeria	No	Identificador de la tabla que contiene los datos del ingreso pos-operatorio médico (FK)
enfermera	Enfermera_enfermeria	No	Identificador de la tabla que contiene los datos de la enfermera (FK)
medico	Medico_enfermeria	No	Identificador de la tabla que contiene los datos del médico (FK)
fecha	Date	No	Fecha en la que se realiza el egreso pos-operatorio
hora	Date	No	Hora en la que se realiza el egreso pos-operatorio
observacion	String		Observación que se le realice al paciente
eliminado	Boolean		Estado del egreso pos-operatorio
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.11. Descripción de la tabla EgresoPostoperatorio

Nombre: LiquidosVenas			
Descripción: Contiene los datos de los líquidos que serán ingeridos en venas.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de los líquidos ingresados al paciente en venas
version	Integer		Número de la versión de los líquidos ingresados al paciente en venas
coloides;	Integer		Cantidad en cc de la muestra

sangre;	integer		Cantidad en cc de la muestra
nutricionParental;	integer		Cantidad en cc de la muestra
eliminado	Boolean		Estado de los líquidos ingresados al paciente en venas
cid	Integer		Identificador de la última conversación realizada a la tabla
balanceLiquidos;	BalanceLiquidos_enfermeria	No	Identificador de la tabla que contiene los datos del balance de líquidos (FK)
tratamiento	Integer		Cantidad en cc de la muestra
hidratacionParental	Integer		Cantidad en cc de la muestra

Tabla 4.12. Descripción de la tabla LiquidosVenas

Nombre: LiquidosIngeridosOralmente			
Descripción: Contiene los datos de los líquidos que son ingeridos de forma oral.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de los líquidos ingresados al paciente vía oral
version	Integer		Número de la versión de los líquidos ingresados al paciente vía oral
eliminado	Boolean		Estado de los líquidos ingresados al paciente vía oral
cid	Integer		Identificador de la última conversación realizada a la tabla
valor	String		Cantidad en cc de la muestra
balanceLiquidos	BalanceLiquidos_enfermeria		Identificador de la tabla que contiene los datos del balance de líquidos(FK)
tipoLiquido	TipoLiquido_enfermeria		Nombre de liquido ingerido

Tabla 4.13. Descripción de la tabla LiquidosIngeridosOralmente

Nombre: LiquidosEliminados	
Descripción: Contiene los datos de los líquidos de un paciente, que son eliminados.	

Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de los líquidos eliminados del paciente
version	Integer		Número de la version de los líquidos eliminados del paciente
diuresis	Integer		Cantidad en cc de la muestra
evacuaciones	integer		Cantidad en cc de la muestra
emesis	Integer		Cantidad en cc de la muestra
sondasNasogastricas	integer		Cantidad en cc de la muestra
otros	integer		Cantidad en cc de la muestra
drenajes	integer		Cantidad en cc de la muestra
eliminado	Boolean		Estado de los líquidos eliminados del paciente
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.14. Descripción de la tabla LiquidosEliminados

Nombre: BalanceLiquidos			
Descripción: Contiene los datos del balance de líquidos de un paciente.			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del balance de líquidos
version	Integer		Número de la versión del balance de líquidos
pesoActual	float	No	Peso que tiene el paciente en el momento de realizar el balance de líquidos
Hora	Date	No	Hora en la que se realizó el balance de líquidos
Fecha	Date	No	Fecha en la que se realizó el balance de líquidos
eliminado	Boolean		Estado del balance de líquidos
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.15. Descripción de la tabla BalanceLiquidos

Nombre: NotaOperatoria			
Descripción: Contiene los datos de las notas operatorias			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador de la nota operatoria

version	Integer		Número de la versión d la nota operatoria
cipProcedimiento	CipProcedimiento_enfermeria		
diagnosticoMedicoById DiagnosticoPreoperatorio	DiagnosticoMedico_enfermeria	No	Identificador de la tabla que contiene los datos del diagnostico realizado por el médico antes de que el paciente sea operado(FK)
medico	Medico_enfermeria		Identificador de la tabla que contiene los datos del médico (FK)
solicitudIntervencion	SolicitudIntervencion_enfermeria		Identificador de la tabla que contiene los datos de la solicitud de intervención (FK)
diagnosticoMedicoById DiagnosticoPosoperatorio	DiagnosticoMedico_enfermeria	No	Identificador de la tabla que contiene los datos del diagnostico realizado por el médico después de que el paciente sea operado(FK)
fecha	Date		Fecha en la que se realiza la nota operatoria
hallazgos	String		Hora en la que se realiza la nota operatoria
procedimiento	String		Descripción del procedimiento realizado
cid	Integer		Identificador de la última conversación realizada a la tabla
eliminado	Boolean		Estado de la nota operatoria

Tabla 4.16. Descripción de la tabla NotaOperatoria

Nombre: NandaDiagnostico			
Descripción: Contiene todos los diagnósticos referentes al estándar internacional Nanda			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del Nanda diagnóstico
version	Integer		Número de la versión del diagnóstico Nanda
nandaTaxonomias	NandaTaxonomias_enfermeria		Identificador de la tabla que contiene los datos de las Nanda taxonomías(FK)

nandaTipoDiagnostico	NandaTipoDiagnostico_enfermeria	No	Identificador de la tabla que contiene los datos de los tipos de diagnósticos Nanda(FK)
nandaCategoriaValoracion	NandaCategoriaValoracion_enfermeria		Identificador de la tabla que contiene los datos de la categoría de valoración Nanda(FK)
nandaPatronesFuncionalesSalud	NandaPatronesFuncionalesSalud_enfermeria	No	Identificador de la tabla que contiene los datos de los patrones funcionales Nanda(FK)
codigo	String		Valor del código que pertenece a cada diagnóstico Nanda
valor	String		Nombre del diagnóstico
eliminado	Boolean		Estado del Nanda diagnóstico
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.17. Descripción de la tabla NandaDiagnostico

Nombre: ControlIndMedicamentos			
Descripción: Contiene todo el control de las indicaciones de medicamentos ordenados a un paciente determinado			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del control de indicaciones de medicamentos
id_registro	Integer	No	Identificador de la tabla que contiene los datos del registro(FK)
fecha	date		Fecha en la que se realizo el control de las indicaciones de medicamentos
hora	date		Hora en la que se realizo el control de las indicaciones de medicamentos
id_enfermero	Enfermera_enfermeria	No	Identificador de la tabla que contiene los datos del enfermero(FK)
id_ind_medicamento	IndMedicaMedicamento_enfermeria	No	Identificador de la tabla que contiene los

			datos indicaciones de medicamentos(FK)
dosis_suministrada	Integer		Valor que se le suministrara al paciente
id_via_administracion	Integer	No	Identificador de la tabla que contiene los datos de las vías de administración de los medicamentos (FK)
cumplido	boolean		Si se cumplió con el control de indicaciones de medicamentos
pendiente	boolean		Si la indicación del medicamento esta o no pendiente
version	Integer		Número de la versión del control de indicaciones de medicamentos
eliminado	boolean		Estado del control de las indicaciones de medicamentos
cid	Integer		Identificador de la última conversación realizada a la tabla
causa	String		Causa por la cual se le indico el medicamento

Tabla 4.18. Descripción de la tabla ControlIndMedicamentos

Nombre: ControlTratamientoMedicamentos			
Descripción: Contiene los datos para hacer el control de tratamiento de los medicamentos que serán suministrados a los pacientes			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del Control de tratamiento de enfermería
version	Integer		Número de la versión del control de tratamiento de enfermería
controlTratamiento	ControlTratamiento_enfermeria	No	Identificador de la tabla que contiene los datos del control de tratamiento(FK)
viaAdministracionMedicamento	ViaAdministracionMedicamento_enfermeria	No	Identificador de la tabla que contiene los datos de la vía de administración de medicamentos al paciente(FK)

origenMedicamento	OrigenMedicamento_enfermeria	No	Identificador de la tabla que contiene los datos del origen de los medicamentos(FK)
enfermera	Enfermera_enfermeria	No	Identificador de la tabla que contiene los datos de la enfermero(FK)
ordenMedicaMedicamento	OrdenMedicaMedicamento_enfermeria	No	Identificador de la tabla que contiene los datos de la orden medica de los medicamentos(FK)
fecha	Date		Fecha en la que se realizo el control de tratamiento de medicamentos
hora	Date		Hora en la que se realizo el control de tratamiento de medicamentos
dosis	int		Valor de la dosis a suministrar
cumplido	boolean		Si el control de tratamiento de medicamentos se cumplió o no
causa	String		Causa por la cual se cumplió o no el control de tratamiento de medicamentos
pendiente	boolean		Si se encuentra pendiente o no el control de tratamiento de medicamentos
eliminado	Boolean		Estado del control de tratamiento de medicamentos
cid	Integer		Identificador de la última conversación realizada a la tabla

Tabla 4.19. Descripción de la tabla ControlTratamientoMedicamentos

Nombre: ControlTratamientoObservacion			
Descripción: Contiene los datos de la observación de un control de tratamiento realizado a un paciente determinado			
Atributo:	Tipo:	Nulo:	Descripción:
id	Integer	No	Identificador del control de tratamiento de observación
version	Integer		Número de la versión del control de tratamiento de observación

controlTratamiento	ControlTratamiento_enfermeria	No	Identificador de la tabla que contiene los datos(FK)
enfermera	Enfermera_enfermeria	No	Identificador de la tabla que contiene los datos(FK)
observacion	String		Valor de la observación realizada al paciente
eliminado	Boolean		Estado del control de tratamiento de observación
cid	Integer		Identificador de la última conversación realizada a la tabla
fecha	Date		Fecha en la que se realizo el control de tratamiento de observación
hora	Date		Hora en la que se realizo el control de tratamiento de observación
pendiente	boolean		Si el control de tratamiento de observación se encuentra pendiente o no

Tabla 4.20. Descripción de la tabla ControlTratamientoObservacion

4.2 Modelo de implementación

En este modelo se diseña el sistema a implementar en términos de componentes, ficheros de código fuente, ficheros de códigos binarios, ejecutables, entre otros.

Con la implementación se propone distribuir el sistema, asignando componentes ejecutables a nodos, en el diagrama de despliegue. Además de implementar las clases y subsistemas encontrados durante el diseño. Las clases se implementan como componentes de ficheros que contienen código fuente. [49]

4.2.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutables. Los componentes a su vez representan todos los tipos de elementos software que entran en la

fabricación de aplicaciones informáticas. Estos pueden ser simples archivo, paquetes, bibliotecas cargadas dinámicamente, entre otros. [50]

Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación. Estos paquetes son estereotipados como <<subsistemas>>. Cada subsistema puede contener componentes y otros subsistemas.

A continuación serán expuestos los diagramas de componentes asociados a varios de los subsistemas de implementación identificados. Siguiendo la arquitectura en capas, la estructuración en subsistemas de implementación es la siguiente.

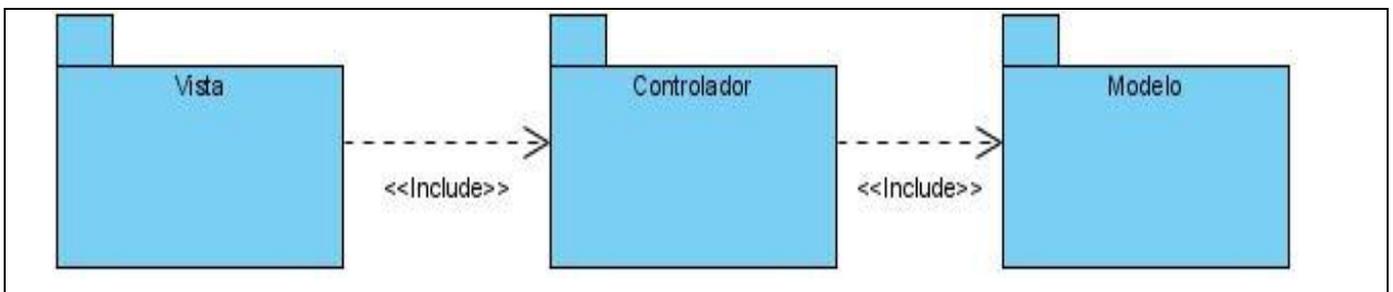


Figura 3.18 Subsistemas de implementación

Existen diferentes tipos de componentes. Para el modelado del diagrama de subsistemas se han utilizado <<file>>, para representar ficheros existente y <<library>> para representar todos las .dll o .jar utilizadas.

En el subsistema Vista, se contemplan todos componentes necesarios para el trabajo con las páginas XHTML, sus ficheros .properties, .XML y .HTML. Los ficheros .XML son los encargados de la navegación de las páginas. Las páginas.XHTML importan librerías tales como Ajax4JSF, Richfaces ó Facelets, donde esta último permite crear plantillas para construir un árbol de componentes, de forma que se puedan definirse como composición de otros.

Las páginas.XHTML importan los componentes y paquetes que encierran la lógica del negocio, tales como Jboss Drools para manejar las reglas del negocio, o Jboss Seam, que permite una correcta integración de la vista y el modelo. Importa además, del subsistema Modelo, el paquete contenedor de las entidades y las librerías de hibernate, para el mapeo de dichas entidades. A continuación se presenta el diagrama de subsistemas de implementación por capas.

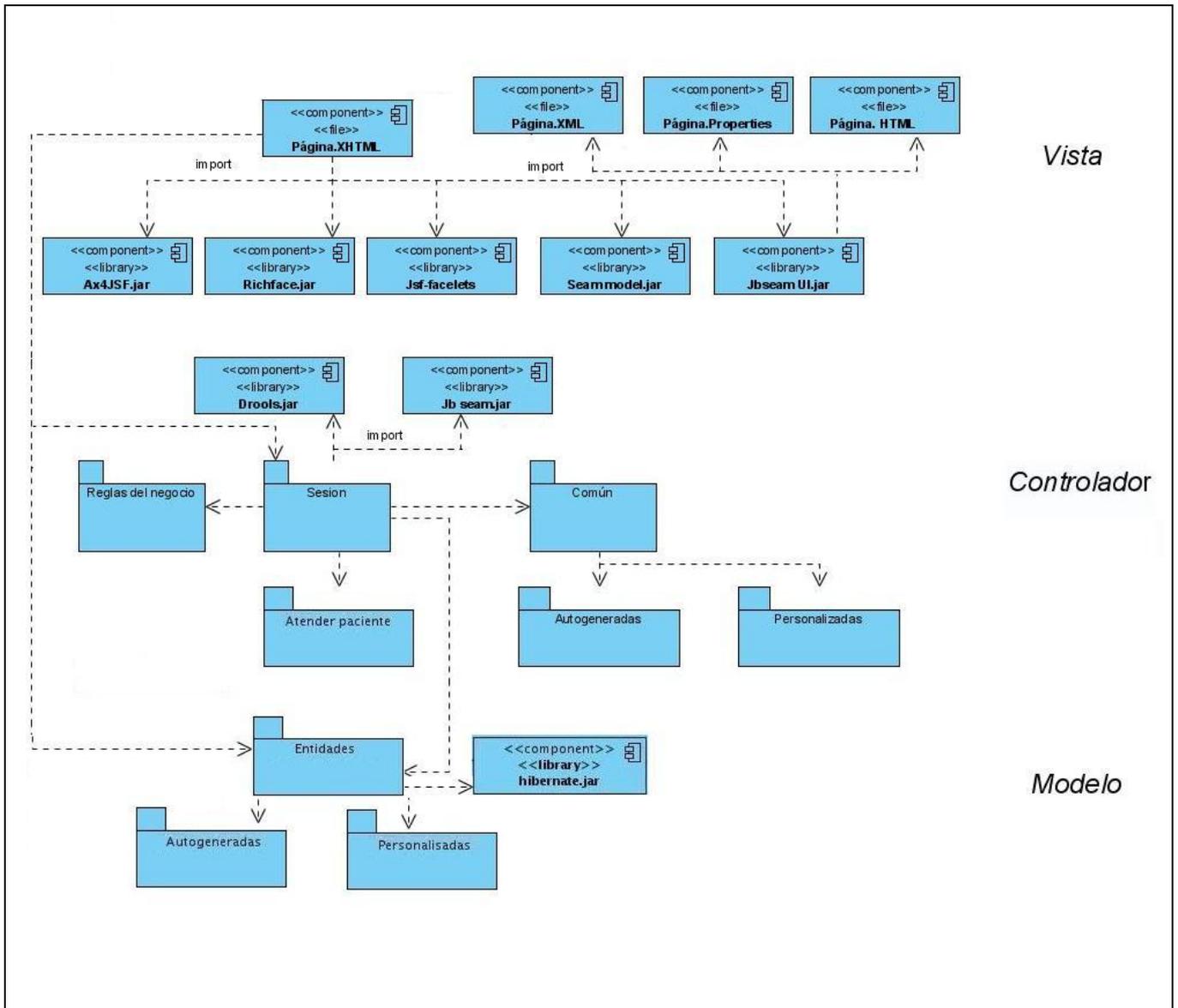


Figura 3.19 Subsistemas de implementación por capas.

4.2.2 Diagrama de despliegue

Los diagramas de despliegue representan la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Los nodos son elementos físicos que existen en tiempo de ejecución y representan recursos computacionales. [51]

Mediante el siguiente modelo, se representa a través de nodos y relaciones entre ellos, el conjunto de dispositivos que interactuarían entre sí al desplegar el sistema propuesto, por ejemplo la comunicación entre páginas cliente y servidores de bases de datos y de aplicaciones.

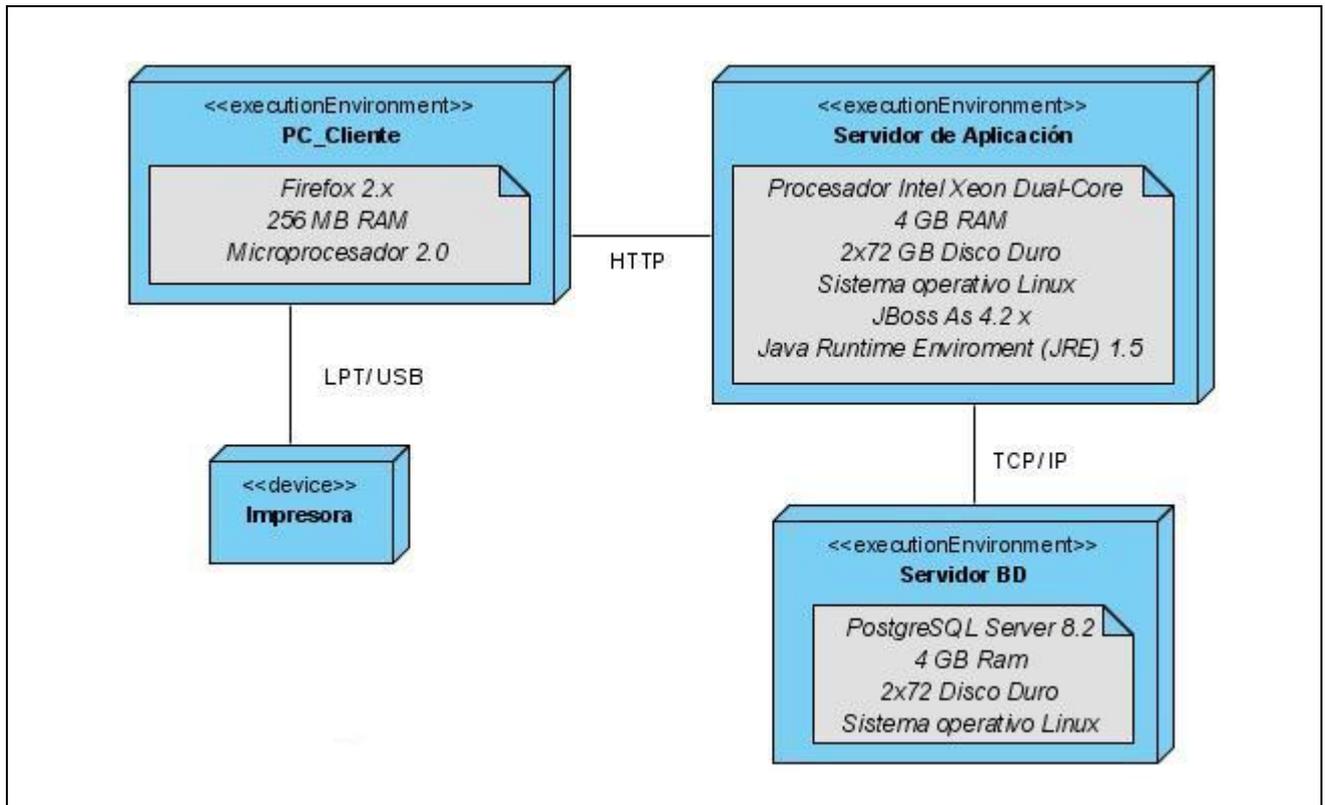


Figura 3.20 Diagrama de Despliegue

A continuación se describen cada uno de los nodos presentes en el diagrama anterior y la comunicación entre ellos:

Servidor de aplicación: Representa el servidor sobre el cual se encuentra corriendo la aplicación web. Este accede al servidor BD para el manejo de información mediante el protocolo TCP/IP. Debe tener el JRE y Jboss Server, para que corra la aplicación.

PC_Cliente: Representa la computadora del cliente que se conecta al servidor de aplicación, mediante Firefox 2.x. La misma se comunica con el servidor por el protocolo HTTP.

Servidor BD: Es la representación del servidor de base de datos PostgreSQL Server, en el cual los datos pertenecientes a la aplicación son almacenados.

Impresora: Dispositivo que se conecta con la PC cliente mediante los puertos LPT ó USB, para cuando se vaya a imprimir algún reporte.

4.3 Tratamiento de errores

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas, es a esto a lo que se comúnmente se le llama excepción. Mediante el tratamiento de excepciones se restaura un estado en el que la rutina pueda seguir la ejecución, obteniéndose un sistema más robusto y fiable.

El lenguaje de programación que se utiliza para el desarrollo del sistema, proporciona una solución elegante al problema de las excepciones, permitiendo escribir el flujo principal de su código y tratar los casos excepcionales en otro lugar. Esto lo hace mediante el bloque **try**, para detectar cuando ocurra algún fallo y el **catch** para manejar dichas excepciones, mediante mensajes que se muestran en la interfaz de usuario, por las facilidades que brinda el FacesMessages, componente del framework Seam.

Ejemplo:

```
public String insert() {  
  
    try {  
        entityManager.persist(ingreso);  
    }  
    catch (RuntimeException e) {  
        facesMessages.add(e.getMessage());  
    }  
}
```

En el sistema propuesto, el control de las excepciones se lleva a cabo a toda porción de código, donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos, aunque también se controlan los errores que pueden surgir en la validación de datos provenientes de la interfaz de usuario; utilizando para ello controles de conversión y validación; 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.

Mediante 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.

Existe además un archivo XML, que engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, así como la página a la que el sistema redirecciona en caso de la aparición de un error sorpresivo.

4.4 Seguridad

Para lograr un alto grado en cuanto a la integridad, confidencialidad, disponibilidad y de no repudio en el sistema propuesto, es que se han tomado varias medidas en cuanto a la seguridad del mismo.

Para acceder al sistema, primeramente se deberán haber establecido permisos. Estos permisos se darán teniendo en cuenta el rol que desempeña el actor en la institución hospitalaria, por un administrador encargado de esta tarea. Cuando se desee iniciar una sesión de de trabajo, se solicitará: nombre de usuario y contraseña. El actor deberá introducir los datos solicitados, el sistema deberá verificar que los datos introducidos sean válidos, si es así, el sistema mostrará al actor solamente las funcionalidades a las que tiene acceso.

Para lograr que el usuario solo pueda acceder a las opciones permitidas se utilizan reglas, que restringen el acceso a directorios, páginas, controles, opciones del menú y servicios del negocio. Esto puede llevarse a cabo por la posibilidad de integración con el motor de reglas JBoss Drools, que brinda el framework de Seguridad de JBoss Seam.

Se podrá además llevar un registro de trazas, para tener control de que hace cada usuario. Una traza se registra por cada acción que realiza un usuario sobre el sistema.

El administrador del sistema, se encargará de configurar funcionalidades propias del sistema, tales como administrar roles, usuarios, seguridad, entre otros.

4.5 Estándares de codificación

Los estándares de codificación son reglas específicas a una lengua, que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, permitiendo obtener un código de alta calidad.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, con vistas a generar un código de alta calidad, es de gran importancia para la calidad del software y para obtener un buen rendimiento.

Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y posteriormente, se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de *software* fácil de comprender y de mantener.

Para la solución del problema tratado en este trabajo se han utilizado el estándar de la SUN (Stanford University Network) para Java, con la utilización la **Notación Camello**, para denotar variables y parámetros. En esta notación, si el identificador es una palabra simple se escribe todo con minúscula, pero si es compuesta, la primera palabra comienza con minúscula y las palabras a continuación con mayúscula. Ejemplo: `int cantidad` o `int cantidadReal`.

Utiliza además la **Notación Pascal**, para las clases y métodos. En esta notación los identificadores de clases y métodos se escriben con mayúscula en todo momento. Ejemplo: `public class Paciente`.

A continuación se especifican algunas restricciones para la nomenclatura, basada en el estándar a utilizar.

Indentación

Inicio y bloque de fin: Se deben dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque `{}`. Lo mismo sucede para el caso de las instrucciones `if`, `else`, `for`, `while`, `do while`, `switch`, `foreach`.

Aspectos generales: El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla.

- Los inicios (`{`) y cierre (`}`) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.
- Nunca colocar `{` en la línea de un código cualquiera, esto requiere una línea propia.

Comentarios, separadores, líneas, espacios en blanco y márgenes

Ubicación de comentarios: Se debe comentar al inicio de la clase o función especificando el objetivo de la misma, así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.

Líneas en blanco: Se debe dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco: Se deben usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto.

Variables y constantes

Apariencia de variables: El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación Camello.

Ejemplo: sNombrePaciente.

Apariencia de constantes: Se deben declarar las constantes con todas sus letras en mayúscula.

Clases y objetos

Apariencia de clases y objetos: Se utilizara la Notación Pascal. Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato.

Apariencia de atributos: El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación Camello.

Apariencia de las funciones: Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Ejemplo: function BuscarUnidad().

Entidades

Se define que el nombre de cada entidad con solo leerlo se conozca el propósito de la misma, además debe seguir la siguiente estructura: [nombre de la entidad]_enfermería.

Conclusiones

En este capítulo fue presentada la base de datos que se obtuvo del diseño propuesto. Además la construcción del sistema, a partir del diagrama de componentes, el cual representa las dependencias entre los componentes software. Mediante el diagrama de despliegue, se ilustran los nodos que serán usados para la implantación de la aplicación y para cada uno de éstos el protocolo de comunicación.

Se pudo ver también como tratar las posibles excepciones que el sistema pueda tener, así como implementar un sistema seguro, confiable y más legible, mediante los estándares de codificación.

CONCLUSIONES

En aras de darle cumplimiento al problema propuesto, se ha podido arribar a las siguientes conclusiones:

- Los Sistemas de Información Hospitalaria estudiados, no satisfacen las necesidades requeridas para lograr un óptimo funcionamiento en el área de Enfermería de las instituciones hospitalarias, además de ser sistemas propietarios en su gran mayoría.
- La gestión manual de toda la información manipulada en los procesos de negocio definidos, dificulta el trabajo del personal de enfermería.
- La arquitectura definida es robusta y flexible. Capaz de responder a los requerimientos no funcionales del sistema.
- El sistema informático obtenido permite optimizar la gestión de hojas de evolución, así como las mediciones de signos vitales y el control de tratamiento.

RECOMENDACIONES

Se recomienda al Área Temática de Gestión Hospitalaria:

- Incluir un sistema de alerta para que el personal de enfermería realice el control de tratamiento a la hora que le corresponde.
- Adecuar las interfaces del sistema para facilitar el uso de las estaciones móviles.
- Incluir en el sistema la planificación de enfermería para tener un mejor control sobre el personal que interactúa con el paciente.

REFERENCIAS BIBLIOGRÁFICAS

1. Aumentar la productividad: Rol de la enfermería en el Gerenciamiento Hospitalario Informatizado y su impacto en los costos. 2008. Disponible en:
 - a. http://www.biocom.com/informatica_medica/enfermeria.html
2. Marín Díaz, Miguel E. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. Cuba. 2006. p 3.
3. 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:
 - a. http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc
4. Sistemas de información. Disponible en:
 - a. http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n
5. ISOFT. X-HIS. Disponible en:
 - a. http://www.isoftware.es/dummy/xHIS_080108.pdf
6. Igual a [5].
7. Softel. Productos desarrollados: Galen Hospital. 2007. Disponible en:
 - a. <http://www.softel.cu/productos.htm>
8. SIVSA. Hosix-v: El producto y Módulos. Disponible en: <http://www.sivsa.com/>
9. Herrero Santoja, Joaquín. La experiencia de Kewan- Cosmosalud. Disponible en:
 - a. <http://www.informatica2007.sld.cu/Members/jherreroypf/la-experiencia-kewan-cosmosalud/2006-11-15.1968340577/download>
10. Igual a [9].
11. Taxonomías enfermeras. Glosario de términos. Disponible en:
 - a. <http://www.taxonomiaenfermera.com/pag/glosario.html>
12. Igual a [11].

13. Arquitectura Cliente- Servidor: Características principales de este tipo de arquitectura de cara a base de datos. 2007. Disponible en:
 - a. <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
14. Microsoft patterns and practices. Model-View-Controller. MSDN. 2009. Disponible en: <http://msdn.microsoft.com/en-us/library/ms978748.aspx>
15. Guía de desarrollo del RichFaces. 2008. Disponible en:
 - a. http://jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html_single/index.html
16. Amarino, Mariano. Ajax un nuevo acercamiento a aplicaciones *Web*. 2005.
 - a. Disponible en: <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
17. Igual a [16].
18. JBoss Community. (2007). Recuperado el 28 de Diciembre de 2008, de JBoss Ajax4jsf. Introducción. Disponible en:
 - a. <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/Introduction.html>
19. Framework JBoss Seam. Febrero 2008. Disponible en:
 - a. <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>
20. IH-SW-DR-091 ALAS-HIS_Documento de Arquitectura del Sistema. 2008. p69, p70.
21. Fuentes Pérez, Hector. Guia Rapida de JBoss Drools BRMS 4.0. Junio, 2008. Disponible en: <http://www.fuenteperez.es/blog/guia-rapida-de-jboss-drools-brms-40>
22. Suárez González, Hector. Manual Hibernate. 2003. Disponible en:
 - a. <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>
23. Guías Breves de Tecnologías W3C. Guía Breve de CSS. Disponible en:
 - a. <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>

24. HTML con XHTML y CSS. Todo el código para crear sitios web efectivos y originales. Elizabeth Castro.p22, p23.
25. L 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.
26. JBoss Application Server 4.2. Disponible en: <http://www.osalt.com/es/jboss>
27. Caraballo, Daniel; Madera, Mario; Odin, Marcelo. Estudio de GNU/Linux y el software libre como solución integral para empresas. 2004. p74. Disponible en:
 - a. <http://www.fing.edu.uy/~asabigue/prgrado/2004eofgl/contenido/archivos/Anexo-II.pdf>
28. PostgreSQL: Otras características. Disponible en:
 - a. <http://es.wikipedia.org/wiki/PostgreSQL>
29. Metodologías de desarrollo. 2008. p2. Disponible en:
 - a. http://teleformacion.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_1
30. Metodologías de desarrollo. 2008. p2, p3. Disponible en:
 - a. http://teleformacion.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_1
31. Rumbaugh James, Jacobson Ivar, Booch Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Education. 2007. p18.
32. Hernández Orallo, Enrique. El lenguaje unificado de modelado (RUP).
33. Notación de Modelado de Procesos de Negocio. 2007. Disponible en:
 - a. http://www.economicasunp.edu.ar/02-EGrado/materias/trelew/analisis_sistemas%20I/info/BPD.ppt
34. Owen Martin, Raj Jog. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. p4. Disponible en: www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
35. Presman, Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

36. Owen Martin, Raj Jog. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. Disponible en: www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
37. Teoría de sistemas. Universidad de Valparaíso. 2005
38. Igual a [37].
39. Igual a [37].
40. Igual a [35].
41. Igual a [35].
42. Igual a [35].
43. Igual a [35].
44. IH-SW-DR-091 ALAS-HIS_Documento de Arquitectura del Sistema. 2008. p60.
45. Igual a [44]
46. Igual a [35].
47. Igual a [35].
48. Navarro Franco, Ángel José. *UML en acción. Modelando Aplicaciones Web*. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.
49. Igual a [35].
50. Igual a [35].
51. Igual a [35].
52. Sistemas numerativos de clasificación. La CIE y el MeSH. Disponible en:
 - a. <http://elies.rediris.es/elies14/cap441.htm>
53. <http://www.babylon.com/definition/Dose/Spanish>

54. Diccionario de cáncer. Disponible en:

http://www.cancer.gov/templates/db_alpha.aspx?lang=spanish&CdrID=44691

BIBLIOGRAFÍA

1. ALBET.SA. 2008. IH-SW-DR-091 ALAS-HIS_Documento de Arquitectura del Sistema. 2008. [En línea] 2008.
2. Biocom. 2008. Aumentar la productividad: Rol de la enfermería en el Gerenciamiento Hospitalario Informatizado y su impacto en los costos. [En línea] 2008. [Citado el: 23 de enero de 2009.] http://www.biocom.com/informatica_medica/enfermeria.html.
3. Caraballo, Daniel; Madera, Mario; Odin, Marcelo. 2004. Estudio de GNU/Linux y el software libre como solución integral para empresas. [En línea] 2004. [Citado el: 12 de marzo de 2009.] <http://www.fing.edu.uy/~asabigue/prgrado/2004eofgl/contenido/archivos/Anexo-II.pdf>.
4. Castro, Elizabeth. HTML con XHTML y CSS. Todo el código para crear sitios web efectivos y originales. [En línea]
5. Cerritos, Antonio, Fernández Puerto, Francisco y Gatica Lara, Florina. 2003. Sistema de Información Hospitalaria". México, D.F. [En línea] 2003. [Citado el: 6 de marzo de 2009.] <http://educacion.salud.gob.mx/cursos/informatica/HIS/his.pdf> .
6. Chamba, Wilman. 2008. Framework JBoss Seam. [En línea] febrero de 2008. [Citado el: 23 de febrero de 2009.] <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.
7. Cumberlidge, Matt. Business Process Management with JBoss jBPM. [En línea]
8. Dan, Allen. 2008. Seam in Action. [En línea] febrero de 2008.
9. Desarrolloweb. 2007. Arquitectura Cliente- Servidor: Características principales de este tipo de arquitectura de cara a base de datos. [En línea] 2007. [Citado el: 15 de 12 de 2008.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
10. Fuentes Pérez, Hector. 2008. Guía Rápida de JBoss Drools BRMS 4.0. [En línea] junio de 2008. [Citado el: 7 de marzo de 2009.] <http://www.fuenteperez.es/blog/guia-rapida-de-jboss-drools-brms-40>.
11. Guía Práctica de Workflow y BPM. JBossJBPM 3.1. [En línea] [Citado el: 9 de febrero de 2009.] http://www.metalogo.org/typo3conf/ext/naw_securedl/secure.php?u=0&file=uploads/media/JBP M-user-guide-spanish.pdf&t=1201712070&hash=6557d3634f229b2ae558349a4165c430.
12. Guías Breves de Tecnologías W3C. Guía Breve de CSS. [En línea] [Citado el: 6 de marzo de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.

13. Hernández Orallo, Enrique. El lenguaje unificado de modelado (RUP). [En línea]
14. Herrero Santoja, Joaquín. 2006. La experiencia de Kewan- Cosmosalud. [En línea] 2006. [Citado el: 8 de febrero de 2009.] <http://www.informatica2007.sld.cu/Members/jherreroypf/la-experiencia-kewan-cosmosalud/2006-11-15.1968340577/download>.
15. ISOFT. X-HIS. [En línea] [Citado el: 16 de enero de 2009.]
http://www.isoftware.es/dummy/xHIS_080108.pdf.
16. Jboss.org. 2008. Guía de desarrollo del RichFaces. [En línea] 2008. [Citado el: 25 de mayo de 2009.] http://jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html_single/index.html.
17. Katz, Max. Practical Richfaces. [En línea] [Citado el: 18 de diciembre de 2008.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
18. García de Jalón, Javier; Alfonso Brazález, Aitor; Imaz Larzabal, Alberto; Calleja, Jesús; García, Jon. Aprenda Java como si estuviera en primero. . [En línea] Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. [Citado el: 9 de febrero de 2009.] <http://www.osalt.com/es/jboss>.
19. Marín Díaz, Miguel E. 2006. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. [En línea] 2006.
20. Medisist. Enlace del médico moderno. [En línea] [Citado el: 3 de enero de 2009.] <http://www.medisist.com.mx/enlace-del-medico-moderno.htm>.
21. 2008. Metodologías de desarrollo. [En línea] 2008. [Citado el: 13 de marzo de 2009.] http://teleformacion.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_1.
22. 2008. Metodologías de desarrollo. [En línea] 2008. [Citado el: 24 de febrero de 2009.] http://teleformacion.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_1.
23. 2009. Microsoft patterns and practices. Model-View-Controller. MSDN. [En línea] 2009. [Citado el: 14 de enero de 2009.] <http://msdn.microsoft.com/en-us/library/ms978748.aspx>.
24. 2007. Notación de Modelado de Procesos de Negocio. [En línea] 2007. [Citado el: 18 de febrero de 2009.]

- http://www.economicasunp.edu.ar/02-EGrado/materias/trelew/analisis_sistemas%20l/info/BPD.ppt.
25. Owen Martin, Raj Jog. 2003. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. [En línea] 2003. www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf.
26. Presman, Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
27. Proyectos HIS. [En línea] [Citado el: 6 de febrero de 2009.] <http://cim.uag.mx/his.html>.
28. Ramírez Márquez, Abelardo Dr, Castell-Florit Serrate, Pastor Dr y Mesa, Guillermo Dr. 2003. El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP). [En línea] 2003. [Citado el: 12 de marzo de 2009.] http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc.
29. Rumbaugh James, Jacobson Ivar, Booch Grady. 2007. El Lenguaje Unificado de Modelado. Manual de Referencia. [En línea] Pearson Education., 2007.
30. SIVSA. Hosix-v: El producto y Módulos. . [En línea] [Citado el: 23 de febrero de 2009.] <http://www.sivsa.com/>
31. Softel. Care2x- The proyect. [En línea] [Citado el: 13 de febrero de 2009.] Disponible en: <http://www.care2x.org/>.
- 32.—. 2007. Productos desarrollados: Galen Hospital. [En línea] 2007. [Citado el: 28 de febrero de 2009.] <http://www.softel.cu/productos.htm>.
33. Suárez González, Hector. 2003. Manual Hibernate. [En línea] 2003. [Citado el: 18 de diciembre de 2008.] <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>.

ANEXOS

Anexo 1 Diagramas de clases del diseño. Diagramas de secuencia

Gestionar ingreso

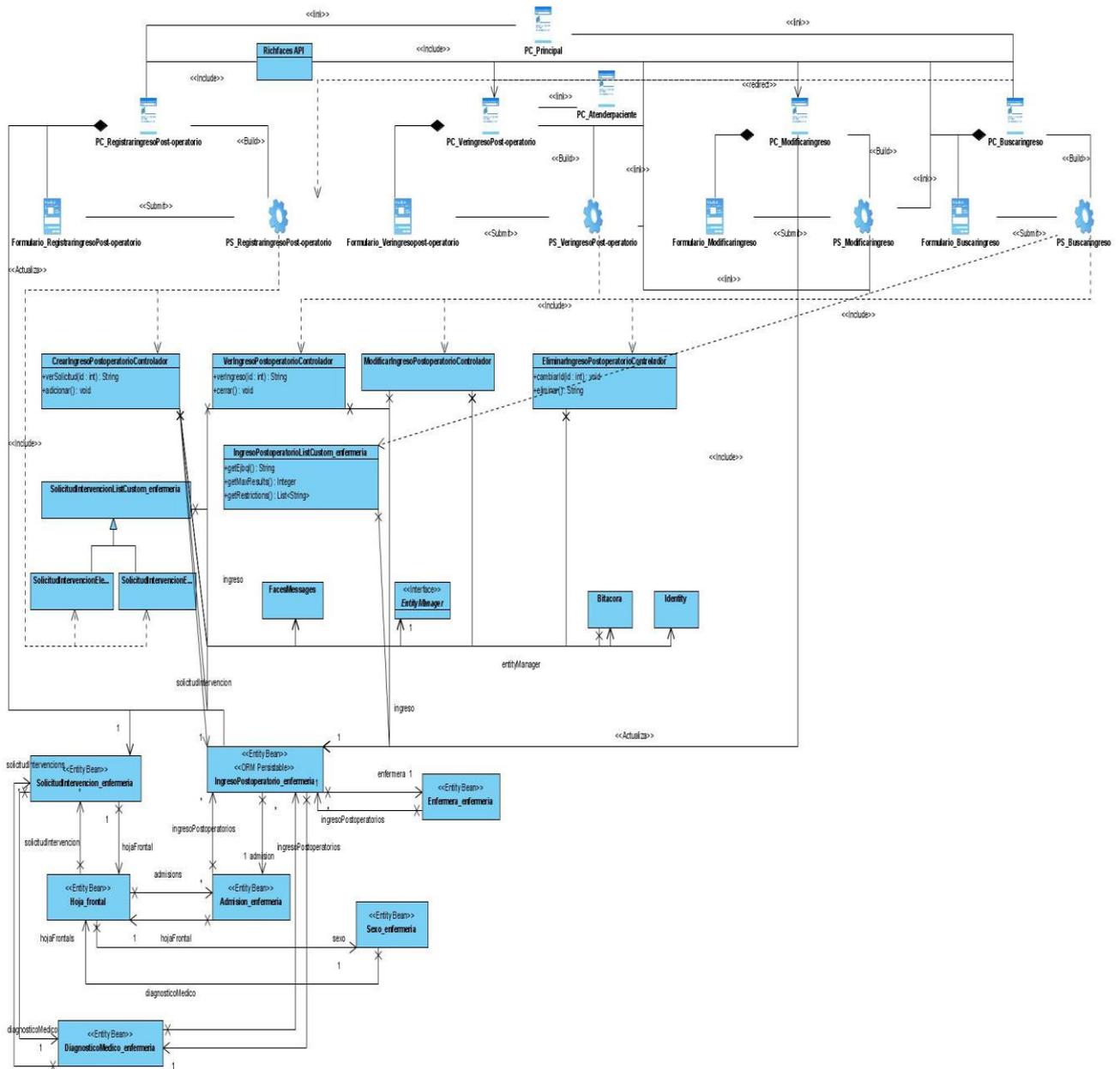


Figura A 1.1 Diagrama de clases del diseño Gestionar ingreso de pacientes.

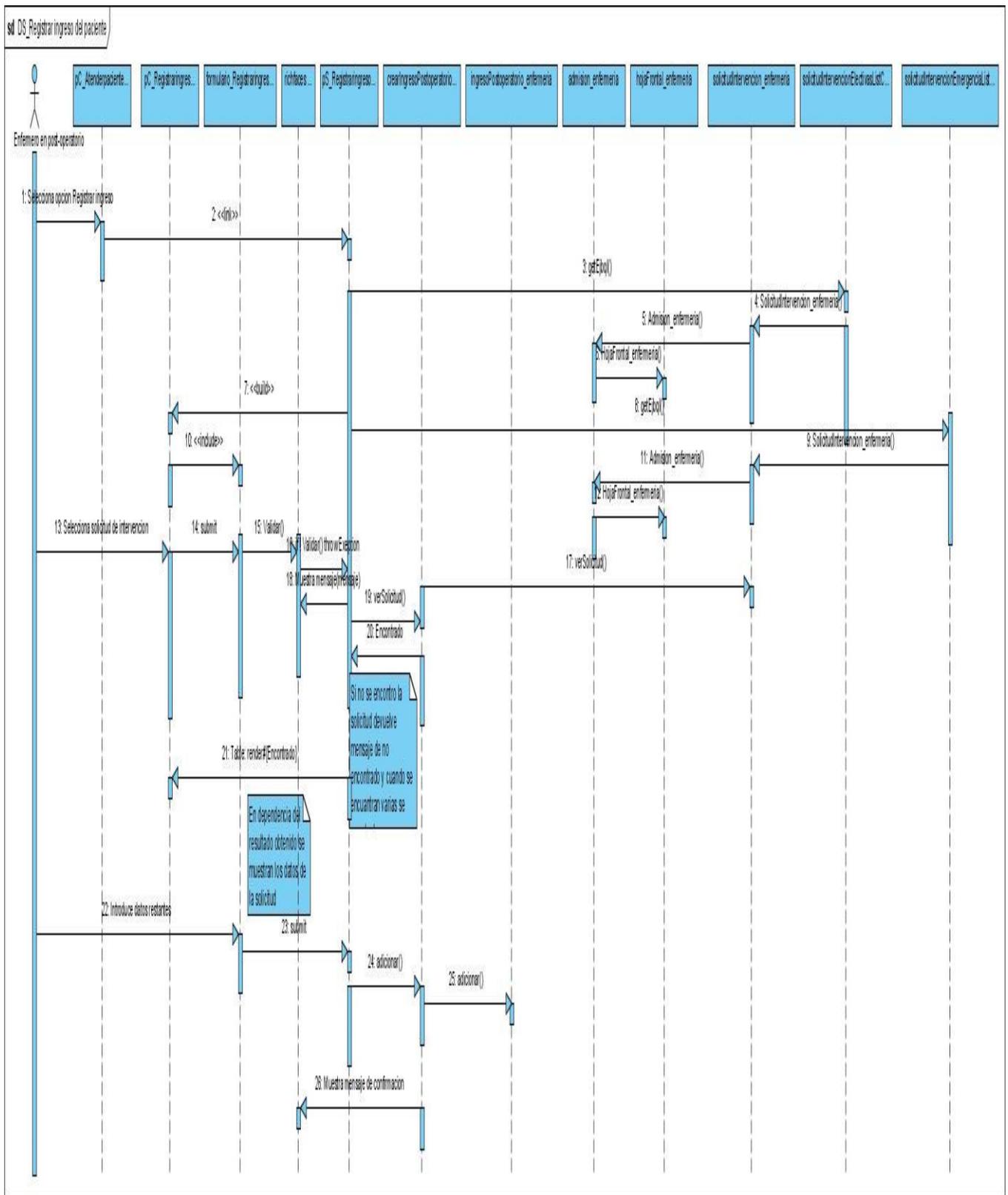


Figura A 1.2 Diagrama de secuencia escenario Registrar ingreso de paciente

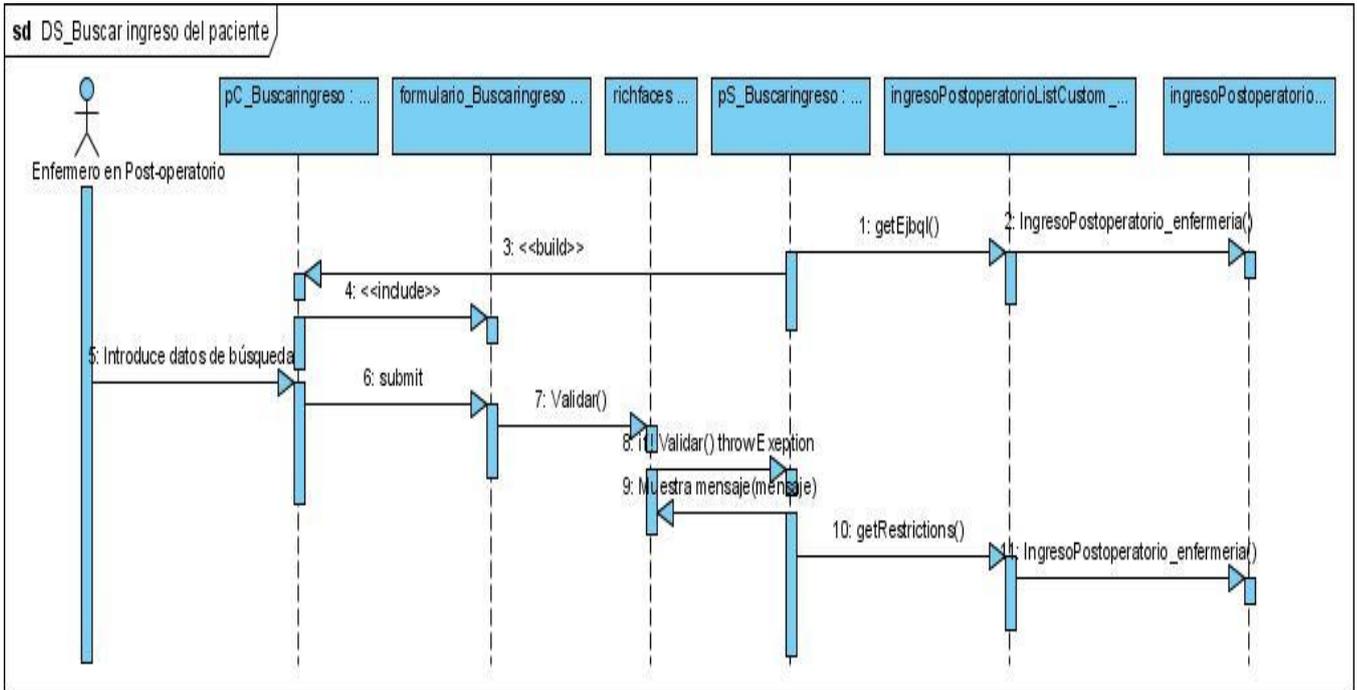


Figura A 1.3 Diagrama de secuencia escenario Buscar ingreso de paciente

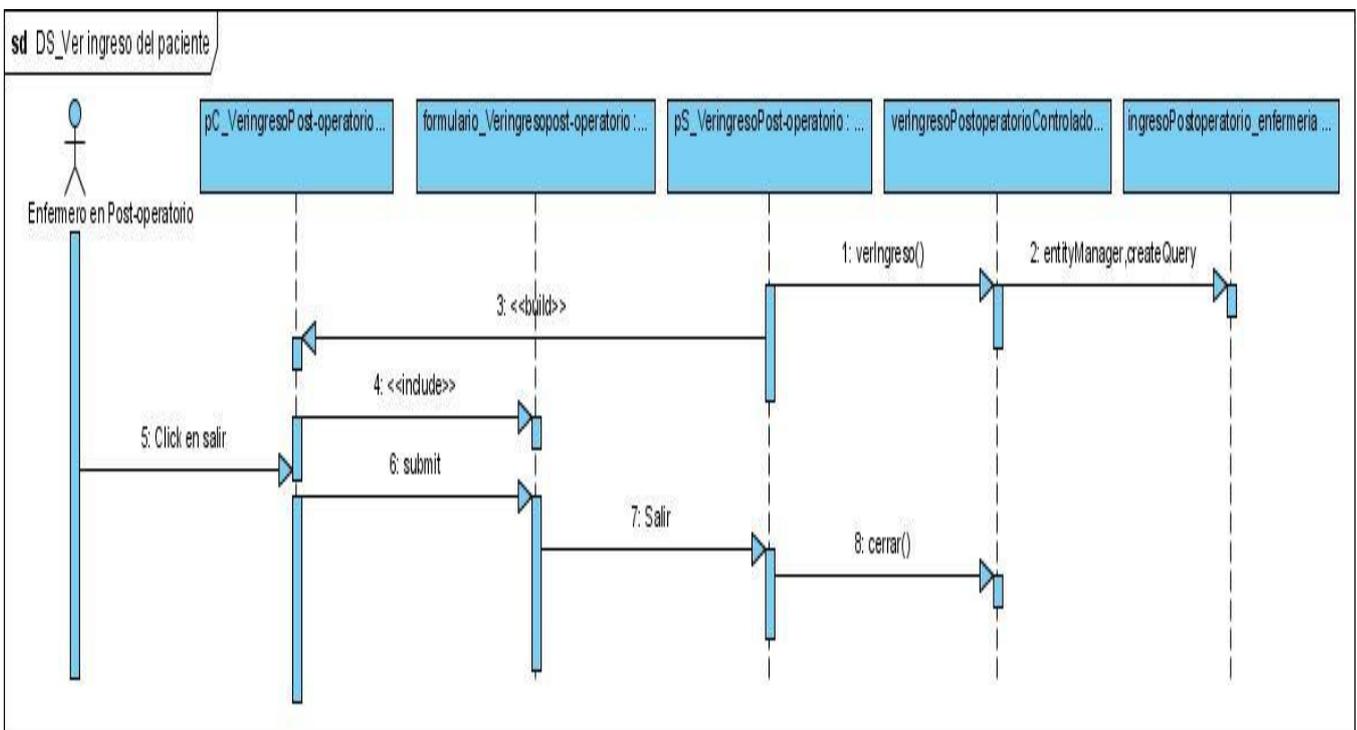


Figura A 1.4 Diagrama de secuencia escenario Ver ingreso de paciente

Gestionar egreso de pacientes

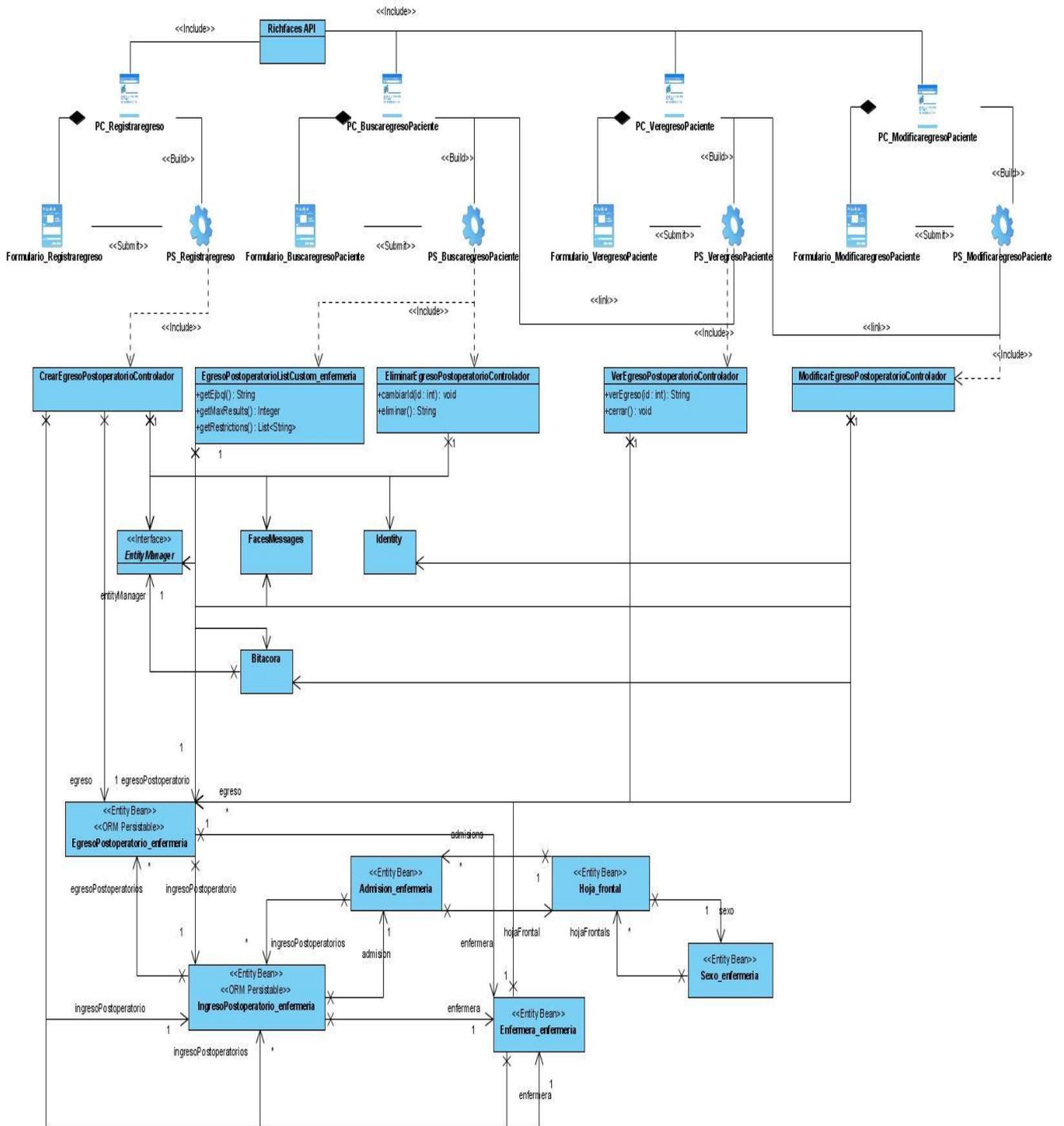


Figura A 1.5 Diagrama de clases del diseño Gestionar egreso de pacientes

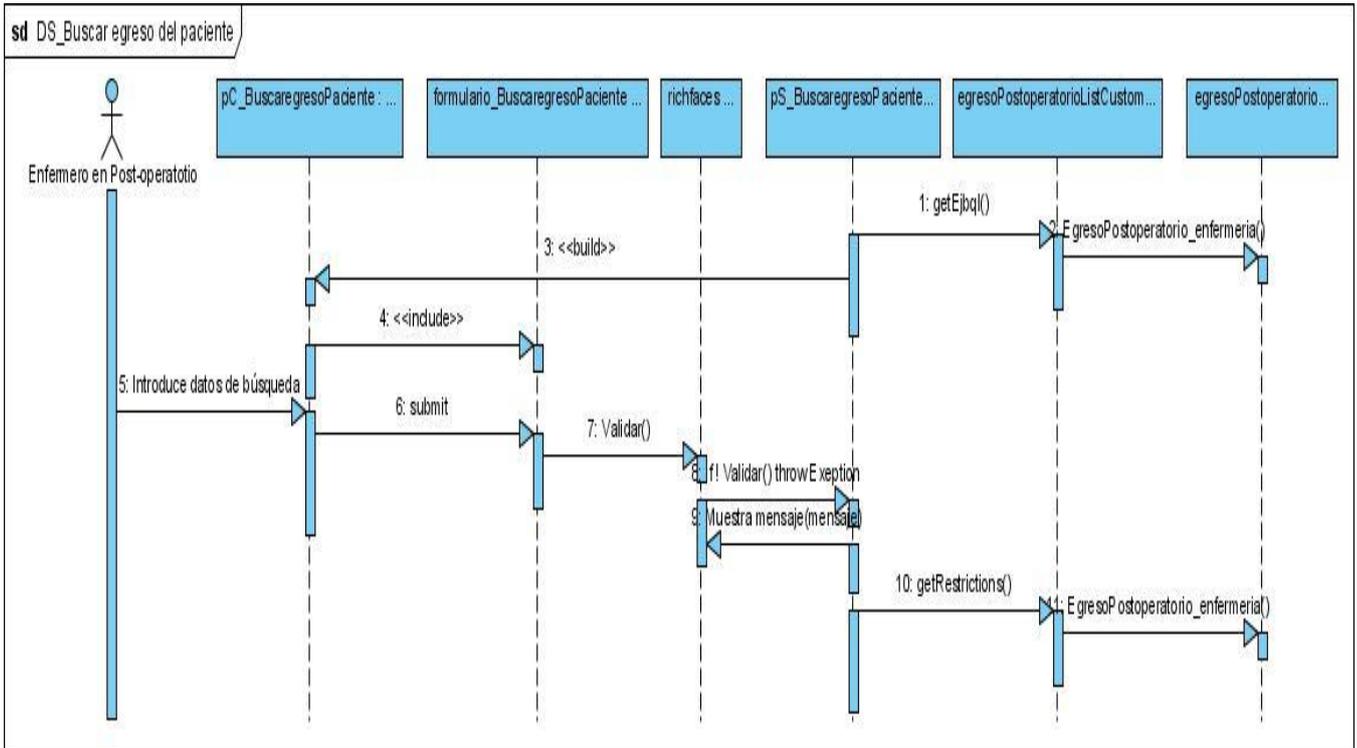


Figura A 1.6 Diagrama de secuencia escenario Buscar egreso del paciente

Anexo 3 Diagrama entidad-relación

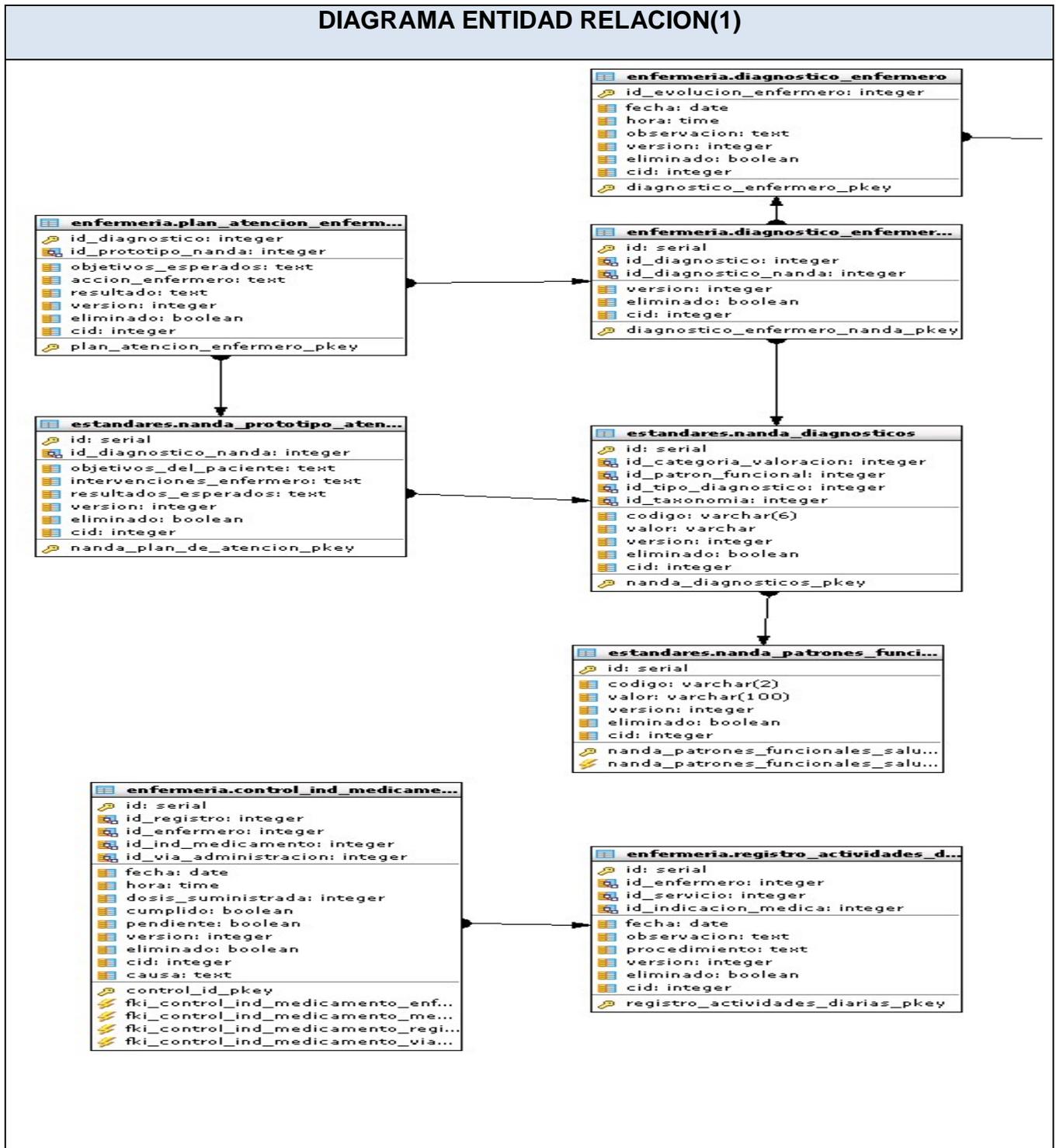


Figura A 3.1 Diagrama Entidad Relación parte 1

DIAGRAMA ENTIDAD RELACION(2)

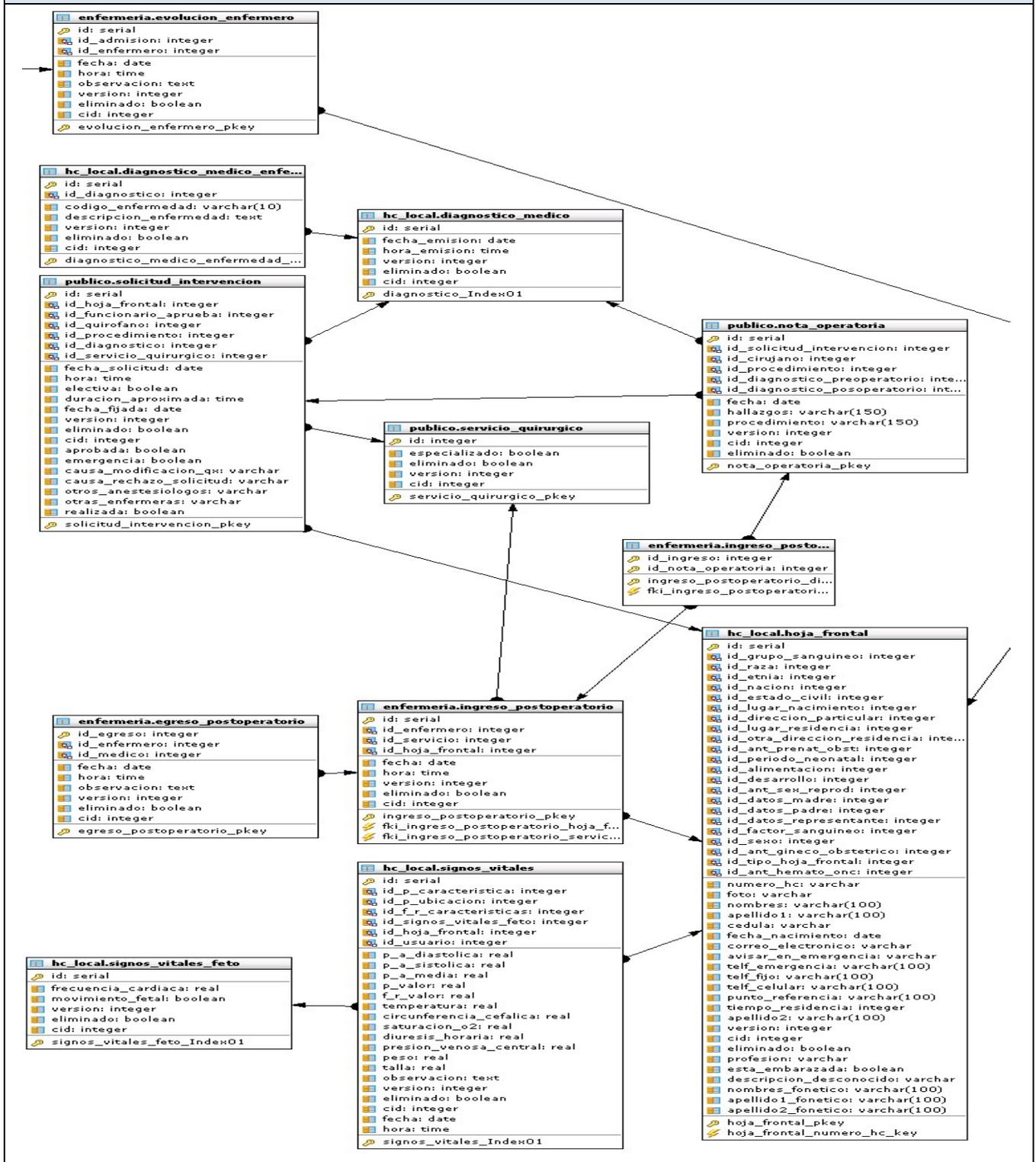


Figura A 3.2 Diagrama Entidad Relación parte 2

DIAGRAMA ENTIDAD RELACION(3)

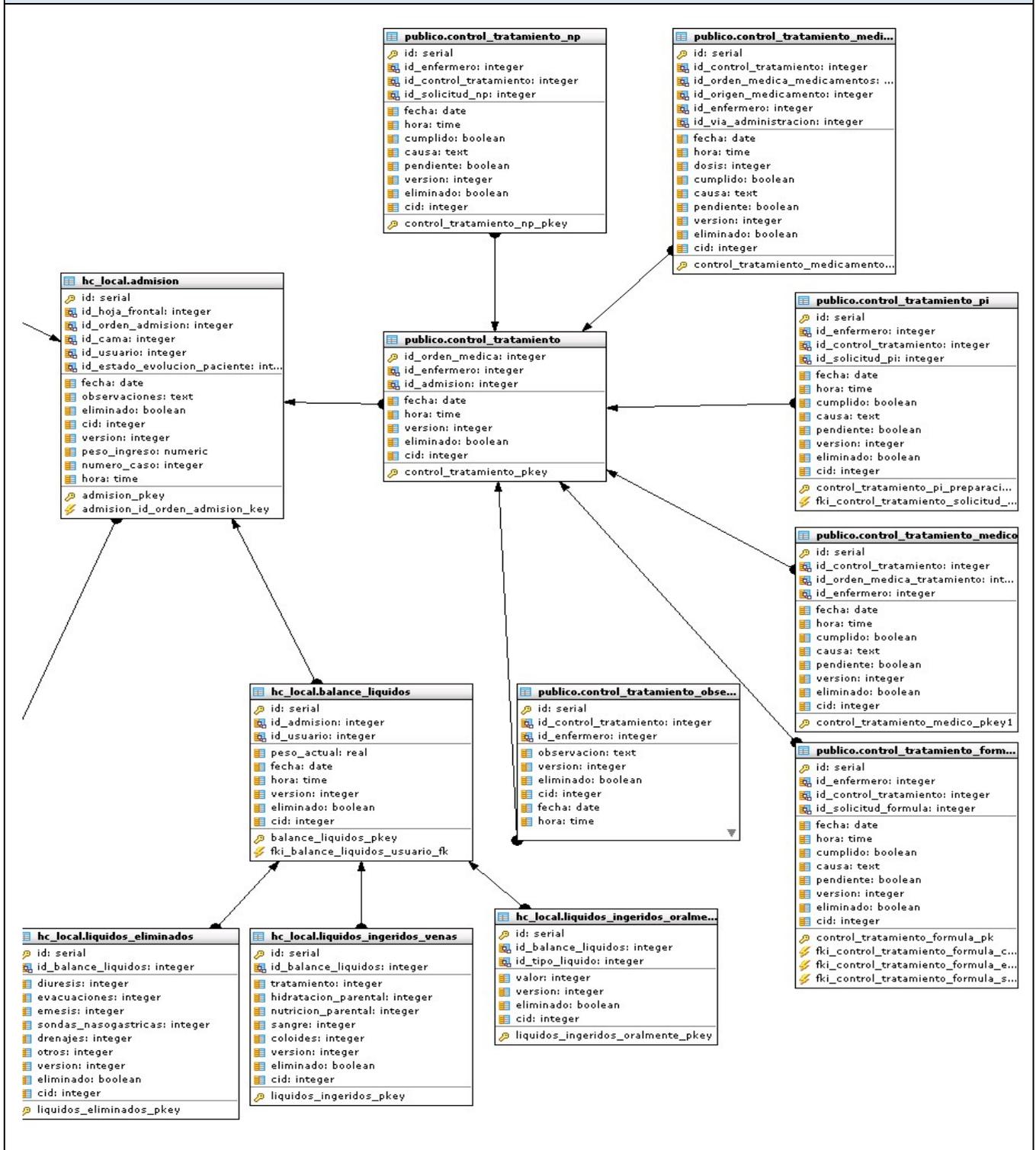


Figura A 3.3 Diagrama Entidad Relación parte 3

Anexo 4 Tablas

CASO DE USO:	Registrar ingreso del paciente
Resumen:	El caso de uso inicia cuando el actor accede a la opción Registrar ingreso del paciente, el sistema brinda la posibilidad de introducir los datos para crear el ingreso del paciente, el actor introduce los datos del ingreso del paciente, el sistema registra el ingreso del paciente, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	No existen
Postcondiciones:	Se registró un ingreso de paciente.
REFERENCIAS	
Requisitos:	9

Tabla A 4.1. Descripción del caso de uso Registrar ingreso del paciente

CASO DE USO:	Ver ingreso del paciente.
Resumen:	El caso de uso inicia cuando el actor selecciona un ingreso del paciente y accede a la opción de Ver ingreso del paciente, el sistema muestra los datos del ingreso del paciente, el caso de uso termina.
Complejidad:	Baja
Precondiciones:	Para ver los datos de un ingreso del paciente, este debe estar seleccionado.
Postcondiciones:	Se vieron los datos de un ingreso del paciente.
REFERENCIAS	
Requisitos:	11

Tabla A 4.2. Descripción del caso de uso Ver ingreso del paciente

Nombre: VerIngresoPostoperatorioControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
+ingreso	IngresoPostoperatorio_enfermeria
+ingresold	Integer
+enfermedades;	List<DiagnosticoMedicoEnfermedad_enfermeria>

+paciente	Admision_enfermeria
- notaOperatoria	List<NotaOperatoria_enfermeria>
+descripciones	List<String>
+ingreso	IngresoPostoperatorio_enfermeria
Responsabilidades:	
+ verPaciente():void	Se busca el ingreso para poder mostrarlo en la interfaz.
+cerrar():void	Para terminar la conversación.

Tabla A 4.3. Descripción de la Tabla VerIngresoPostoperatorioControlador

Nombre: EliminarIngresoPostoperatorioControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
id	int
Responsabilidades:	
+eliminar():void	Dado el id que tengo en la clase, hago una consulta, buscándolo en la base de datos y entonces modifico el campo eliminado, poniéndolo en true, debido a que no se va a realizar la eliminación física del ingreso.

Tabla A 4.4. Descripción de la Tabla EliminarIngresoPostoperatorioControlador

Nombre: CrearIngresoPostoperatorioControlador	
Tipo de clase: Controladora	
Atributo:	Tipo:
-cid	Integer
+enfermera	Enfermera_enfermeria
+diagnostico	DiagnosticoMedico_enfermeria
+descripcion	List<String>
+lista	List<DiagnosticoMedico_enfermeria>
+ingreso	IngresoPostoperatorio_enfermeria
+servicio	Servicio_enfermeria
+admision	Admision_enfermeria
+solicitudIntervencion	SolicitudIntervencion_enfermeria

+seleccionado	Boolean
+notasOperatoria	List<NotaOperatoria_enfermeria>
-id	Integer
+seleccionado	Boolean
+solicitudIntervencion	SolicitudIntervencion_enfermeria
+attribute	SolicitudIntervencion_enfermeria
+ingreso	IngresoPostoperatorio_enfermeria
Responsabilidades:	
+ verSolicitud(Integer id):void	Dado un identificador realizar una consulta para guardar en la clase la solicitud que se selecciono en la interfaz.
+insert():void	Para darle persistencia a los datos.

Tabla A 4.5. Descripción de la Tabla CrearIngresoPostoperatorioControlador

GLOSARIO

API: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CIE 9, CIE 10: La CIE-9, es una clasificación numérica de términos con una estructura estática y cuyas relaciones, bastante limitadas. Ha sido diseñada para ser la clasificación nuclear de cada una de las familias de enfermedades y de circunstancias relacionadas con la sanidad. Mientras que la CIE-9 abarca casi todos los trastornos, la CIE-10 se centra en los mentales, con alguna que otra renovación en la enumeración de los códigos. [52]

Defunción: Muerte de una persona. Desaparición de todo signo de vida, después del nacimiento.

GNU/Linux: (*GNU con Linux* o *GNU+Linux*) es la denominación defendida por Richard Stallman y otros para el sistema operativo que utiliza el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU y de varios otros proyectos/grupos de software. Comúnmente este sistema operativo es denominado como Linux, aunque Stallman sostiene que esta denominación no es correcta

IDE: Un entorno de desarrollo integrado o en inglés *Integrated Development Environment* ('*IDE*') es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Insumos: Son los bienes y servicios que incorporan al proceso productivo las unidades económicas y que, con el trabajo de los obreros y empleados y el apoyo de las máquinas, son transformados en otros bienes o servicios con un valor agregado mayor.

JavaScript: Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Unidosis: Los medicamentos se pueden presentar en forma de multidosis o unidosis. En la unidosis, cada unidad de medicamento es una toma y viene identificada con su lote y caducidad. Para su uso

hospitalario, se emplea la unidosis y cada vez comienza a ser más habitual que también se emplee en la medicación de las oficinas de farmacia [53]

Narcótico: Sustancia que se usa para tratar el dolor que va de moderado a grave. Los narcóticos son como los opiáceos, tales como la morfina y la codeína, pero no se elaboran con opio. Se unen a los receptores de opioides del sistema nervioso central. [54]

NIC: Organización sistemática de las denominaciones de las intervenciones en función de las similitudes de lo que puede considerarse como estructura conceptual.

NOC: Organización sistemática de resultados en grupos o categorías basada en semejanzas, diferencias y relaciones entre los resultados.

PACS: Picture Archiving and Communication System: sistema computarizado para el archivo digital de imágenes médicas.

Psicotrópico: Una sustancia psicotrópica o psicótropo (del griego psyche, "mente" y tropein, "tornar") es un agente químico que actúa sobre el sistema nervioso central, lo cual trae como consecuencia cambios temporales en la percepción, ánimo, estado de conciencia y comportamiento.

Récipe: El récipe es un pedido escrito que hace el médico al farmacéutico para que éste expendiera fármacos al paciente. Es un documento de carácter legal, debe escribirse con tinta, en LETRA LEGIBLE y de ser posible, en una computadora o máquina de escribir. La claridad de la letra es un requisito imprescindible para la elaboración de la receta

RIS: (Sistema de Información Radiológica) Transmite exámenes a distancia y en una red interna. Pueden gestionar y archivar en forma simultánea tanto las imágenes médicas (rayos X, por ejemplo) como la información respectiva de cada examen y el historial médico del paciente.

Signos vitales: Conjunto de variables fisiológicas que son la presión arterial, la frecuencia cardíaca, la frecuencia respiratoria y la temperatura corporal

Stock de servicio: En todo el mundo se ha generalizado el uso de esta palabra inglesa para indicar existencia o acumulación de productos o mercancías o el acopio o provisión que se hace de ellas.