

Universidad de las Ciencias Informáticas
Facultad 7



Título: Módulo Farmacia del Sistema de
Información Hospitalaria alas HIS

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Isnel Leyva Herbella

Reynier Soto Góngora

Tutor: Ing. Alejandro Mario Velázquez Carralero

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



“Seamos realistas y hagamos lo imposible.”

A handwritten signature in black ink, which appears to be the name 'Che' written in a stylized, cursive script.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 24 días del mes de junio del año 2009.

Isnel Leyva Herbella

Firma del Autor

Reynier Soto Gongora

Firma del Autor

Alejandro Mario Velazquez Carralero

Firma del Tutor

DATOS DE CONTACTO

Ing. Alejandro Mario Velázquez Carralero (amvelazquez@uci.cu): Ingeniero en Ciencias Informáticas, graduado en la UCI, en el curso 2006-07. Posee la Categoría Docente de Profesor Instructor. Actualmente imparte asignaturas de Práctica Profesional. Además, se desempeña como Arquitecto Principal de Área Temática Gestión Hospitalaria.

AGRADECIMIENTOS

De Reynier:

A mis padres, por darme todo la dedicación, el amor y comprensión del mundo.

A mi hermana, por ser tan dulce e inigualable.

A mis abuelos, por la infinita confianza y orgullo.

A mis tios, por apostar por mi futuro.

A mi novia, por ayudarme a transformar para siempre este “yo” en “nosotros”.

A mi hermano Daniel, por esta perfecta amistad de tantos años.

A mi hermano Felito, por haber sido el mejor ejemplo de mis mayores aspiraciones.

A mi hermano Juan Carlos, por ser un gran amigo.

A los “Clásicos” por ser amigos tan especiales.

A Maykell Sánchez por su ejemplo diario.

A Alejandro Mario por guiarme y confiar en mí.

A todos mis amigos del proyecto.

De Isnel:

A mis padres, por haberme apoyado en todo momento incondicionalmente, por su preocupación y los años de sacrificio...

A mis hermanos y toda mi familia en general, por no dudar ni un instante en que podría lograr mis propósitos...

A mi novia, por estar siempre a mi lado.

A todos mis compañeros y amigos de estudio, que han compartido conmigo buenos y malos momentos en especial al Bichito, Geraldine, el Yeti, Omar, el Niche.

A Alejandro Mario por guiarme y confiar en mí.

A todos mis profesores desde mis primeros estudios hasta ahora, en especial a Aidé Matamoros, por iniciarme en el mundo de la informática y a Maykell Sánchez por enseñarme que, "lo que sabemos es una gota de agua; lo que ignoramos es el océano."

A mis viejos amigos, Enmanuel, Karina, Sandra, por su amistad sincera.

DEDICATORIA

De Reynier:

A mi prima Geylin.

De Isnel:

A mi papá.

RESUMEN

En el área de farmacia en las instituciones hospitalarias se gestionan grandes volúmenes de información. Su procesamiento, usualmente se realiza de forma manual. Actualmente los productos de software para la gestión farmacéutica son demasiado caros, basados en tecnologías y herramientas propietarias y no satisfacen las necesidades del personal que allí labora. El objetivo del presente trabajo de diploma es desarrollar el módulo Farmacia del Sistema de Información Hospitalaria alas HIS, para facilitar la gestión de información en esta área de las instituciones hospitalarias.

El desarrollo del sistema está guiado por el Proceso Unificado de Desarrollo y se basa en tecnologías libres, multiplataforma y sobre una arquitectura en capas, utilizando Java como lenguaje de programación e implementando el patrón de arquitectura Modelo Vista Controlador. Como Sistema de Gestión de Bases de Datos se utiliza PostgreSQL y como servidor de aplicaciones el JBoss Server. Son utilizadas las librerías de componentes web JBoss UI y RichFaces. Para la administración de las reglas y procesos del negocio se utilizan Drools y JBoss jBPM.

Entre los beneficios que aporta se encuentran proveer un control estricto del consumo de los medicamentos, desglosado por áreas, permitiendo generar reportes de consumo por cada uno de los servicios del hospital, así como por pacientes hospitalizados. El Sistema se rige además por la codificación Internacional de Medicamentos avalada por la Organización Mundial de la Salud. Lo que pone a disposición del profesional médico una herramienta de gestión clínica y administrativa que responde a sus necesidades reales.

Palabras claves:

Sistema de Información Hospitalaria, Farmacia, Herramientas de gestión clínica, Clasificación Internacional de Medicamentos (ATC)

TABLA DE CONTENIDOS

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Sistemas de salud pública	6
1.2 Sistemas de información hospitalaria.....	7
1.3 Descripción específica del módulo.....	8
1.4 Sistemas automatizados existentes vinculados al campo de acción	8
1.5 Tecnologías actuales a considerar.....	10
1.5.1 Java EE	11
1.5.2 Seam.....	11
1.5.3 JBoss AS como servidor de aplicaciones.....	12
1.5.4 JBoss Tools	12
1.5.5 JSF.....	13
1.5.6 Rich Faces 3.2 como librería de componentes JSF	13
1.5.7 Ajax4jsf.....	14
1.5.8 PosgreSQL 8.3 como servidor de base de datos	14
1.5.9 Framework Hibernate para acceso a los datos	16
1.5.10 JPA (Java Persistence API)	16
1.6 Lenguajes de programación.....	16
1.6.1 Java.....	16
1.7 Sistemas distribuidos	17
1.7.1 Modelo Cliente-Servidor	17
1.7.2 Estilo arquitectónico.....	18
1.8 Metodologías de desarrollo.....	19
1.8.1 RUP.....	19
1.8.2 BPMN.....	20
1.8.3 UML.....	21
1.9 Herramientas de desarrollo.....	22
1.9.1 Eclipse como herramienta de desarrollo	22

1.9.2	PgAdmin como aplicación cliente para manejar la Base de datos.....	23
1.9.3	Visual Paradigm como herramienta de modelado.....	23
1.9.4	JRE.....	24
Capítulo 2: Características del Sistema		26
2.1	Flujo actual de los procesos involucrados en el campo de acción.....	26
2.2	Modelo de negocio.....	29
2.3	Objeto de automatización	32
2.4	Propuesta del sistema	33
2.4.1	Requerimientos funcionales.....	33
2.4.2	Requerimientos no funcionales.....	37
2.4.3	Modelo de casos de uso del sistema	40
Capítulo 3: Diseño del Sistema.....		51
3.1	Descripción de la arquitectura.....	51
3.2	Modelo de diseño	52
3.3	Patrones de diseño.....	52
3.3.1	Patrón MVC + JSF.....	54
3.4	Estructuración.....	54
3.5	Diagramas de clases del diseño	56
3.6	Diagramas de secuencias.....	60
3.7	Descripción de clases y atributos.....	63
Capítulo 4: Implementación		72
4.1	Modelo de datos	72
4.1.1	Descripción de las tablas	74
4.2	Modelo de clases persistentes.....	82
4.3	Modelo de despliegue.....	84
4.4	Diagrama de componentes	85
4.5	Tratamiento de errores	86
4.6	Seguridad	86
Conclusiones Generales		89
Recomendaciones		90

Referencias Bibliográficas.....	91
Bibliografía.....	93
Anexos.....	96
Glosario	104

ÍNDICE DE TABLAS

Tabla 1 Actores del negocio.....	30
Tabla 2 Trabajadores del negocio.....	31
Tabla 3 Actores del sistema.....	41
Tabla 4 Descripción textual CU-1: Ver estándar ATC	45
Tabla 5 Descripción textual CU-2: Crear nivel ATC.....	46
Tabla 6 Descripción textual CU-3: Modificar nivel ATC	46
Tabla 7 Descripción textual CU-4: Eliminar nivel ATC.....	46
Tabla 8 Descripción textual CU-5: Crear medicamento ATC.....	47
Tabla 9 Descripción textual CU-6: Modificar medicamento ATC	47
Tabla 10 Descripción textual CU-7: Eliminar medicamento ATC.....	48
Tabla 11 Descripción textual CU-8: Autorizar solicitud de medicamentos	48
Tabla 12 Descripción textual CU-9: Rechazar solicitud de medicamentos	48
Tabla 13 Descripción textual CU-10: Despachar medicamentos.....	49
Tabla 14 Descripción textual clase GestionarATCControlador.....	64
Tabla 15 Descripción textual clase RechazarSolicitudMedControlador	65
Tabla 16 Descripción textual clase RechazarSolicitudMedControlador	66
Tabla 17 Descripción textual clase DespacharSolicitudMedControlador	66
Tabla 18 Descripción textual clase ErrorMedControlador.....	67
Tabla 19 Descripción textual clase PerfilFarmacoControlador	68
Tabla 20 Descripción textual clase AdmisionPerfilListCustom_farmacia	68
Tabla 21 Descripción textual clase ListarPerfilNutricionControlador.....	69
Tabla 22 Descripción textual clase VerPerfilNutricionControlador.....	70
Tabla 23 Tablas del modelo de datos.	74
Tabla 24 Campos comunes a todas las tablas.....	75
Tabla 25 Tabla solicitud_formula_farmacia.....	75
Tabla 26 Campos de la tabla solicitud_formula_farmacia.	75

Tabla 27 Tabla med_in_solicitud_farmacia.	75
Tabla 28 campos de la tabla med_in_solicitud_farmacia.	76
Tabla 29 Tabla estado_solicitud_formula.	76
Tabla 30 Campos de la tabla estado_solicitud_formula.	76
Tabla 31 Tabla formula.	76
Tabla 32 Campos de la tabla formula.	76
Tabla 33 Tabla farmacia_satelite.	77
Tabla 34 Campos de la tabla farmacia_satelite.	77
Tabla 35 Tabla solicitud_med_farmacia.	77
Tabla 36 Campos de la tabla solicitud_med_farmacia.	77
Tabla 37 Tabla presentacion_medicamento.	77
Tabla 38 Campos de la tabla presentacion_medicamento.	78
Tabla 39 Tabla producto_in_formula.	78
Tabla 40 Campos de la tabla producto_in_formula.	78
Tabla 41 Tabla producto.	78
Tabla 42 Campos de la tabla producto.	78
Tabla 43 Tabla medicamento_rechazado.	79
Tabla 44 Campo de la tabla medicamento_rechazado.	79
Tabla 45 Tabla medicamento_producto.	79
Tabla 46 Campo de la tabla medicamento_producto.	79
Tabla 47 Tabla unidad_pres_med.	79
Tabla 48 Campo de la tabla unidad_pres_med.	80
Tabla 49 Tabla med_in_bandeja_induccion.	80
Tabla 50 Campos de la tabla med_in_bandeja_induccion.	80
Tabla 51 Tabla medicamento_cont.	80
Tabla 52 Campos de la tabla medicamento_cont.	81
Tabla 53 Tabla med_in_stock_emergencia.	81
Tabla 54 Campos de la tabla med_in_stock_emergencia.	81

Tabla 55 Tabla med_in_carro_paro.	81
Tabla 56 Campos de la tabla med_in_carro_paro.....	81
Tabla 57 Tabla tipo_envase_farmacia.	82
Tabla 58 Campo de la tabla tipo_envase_farmacia.....	82

ÍNDICE DE FIGURAS

Figura 2.1 Actores del sistema.....	41
Figura 2.2 Diagrama de casos de uso del sistema. Proceso “Farmacia central” .	42
Figura 2.3 Diagrama de casos de uso del sistema. Proceso “Dispensario” .	43
Figura 2.4 Diagrama de casos de uso del sistema. Proceso “Unidosis” .	43
Figura 2.5 Diagrama de casos de uso del sistema. Proceso “Unidad de mezcla” .	44
Figura 2.6 Diagrama de casos de uso del sistema. Proceso “Unidad de nutrición parenteral” .	44
Figura 2.7 Diagrama de casos de uso del sistema. Proceso “Unidad de anestesiología” .	45
Figura 3.1 Modelo de diseño.....	55
Figura 3.2 Diagrama de clases del diseño ATC.	56
Figura 3.3 Diagrama de clases del diseño “Solicitud de medicamentos”.....	57
Figura 3.4 Diagrama de clases del diseño “Perfil farmacoterapéutico”.....	58
Figura 3.5 Diagrama de clases del diseño “Perfil de nutrición parenteral” .	59
Figura 3.6 Diagrama de secuencia ver ATC.....	60
Figura 3.7 Diagrama de secuencia crear medicamento ATC	60
Figura 3.8 Diagrama de secuencia modificar medicamento ATC	61
Figura 3.9 Diagrama de secuencia eliminar medicamento ATC	61
Figura 3.10 Diagrama de secuencia crear nivel ATC	62
Figura 3.11 Diagrama de secuencia modificar nivel ATC	62
Figura 3.12 Diagrama de secuencia eliminar nivel ATC.....	62
Figura 4.1. Modelo de datos.....	73
Figura 4.2. Modelo de clases persistentes.	83
Figura 4.3 Modelo de despliegue.	84
Figura 4.4 Diagrama de componentes.	85

INTRODUCCIÓN

El progreso de la humanidad, en gran medida, ha sido determinado por la capacidad del ser humano de adaptarse y transformar el medio que lo rodea. Una parte fundamental de su ciclo de desarrollo, es el perfeccionamiento de técnicas y herramientas necesarias para el trabajo en sociedad. La clave del éxito está en saber emplearlas racionalmente, optimizando el uso de sus potencialidades en función del bienestar colectivo. Este desarrollo al servicio de la humanidad permite equilibrar las diferencias y mejorar las ramas que proporcionan el sustento primordial para la estabilidad de un país.

En la actualidad, la situación económica global, impone aumentar la superación profesional, la automatización de los servicios, la investigación y el apoyo en la toma de decisiones en todas las actividades del hombre. Por lo que la informática se ha convertido en una rama de gran importancia para el desarrollo por las ventajas que brinda en casi todas las esferas de la sociedad. Entre ellas, la salud como columna vertebral de la calidad de vida de la sociedad contemporánea.

La informática al servicio de la salud representa el eslabón fundamental de la asistencia en la gestión hospitalaria. La gran cantidad de información que se genera, la necesidad de persistencia de los datos clínicos de un paciente y la administración de los medios de un ente hospitalario, constituyen el principal problema hoy en día en un gran número de países en todo el mundo. El crecimiento acelerado de la población mundial genera un consumo excesivo de recursos (papeles, etc.) que con el tiempo, o por algún otro factor como la humedad, sufren deterioro o se pierden informaciones valiosas. La carencia de soluciones costeables para la informatización de la salud, es una necesidad inmediata que conlleva a la búsqueda de soluciones que tengan en cuenta todos los procesos que se desarrollan en un hospital.

La Farmacia Hospitalaria es una especialización profesional que surge con el desarrollo de los grandes hospitales en la segunda mitad del siglo XX. Las necesidades de atención farmacéutica y de gestión de medicamentos en los modernos hospitales precisan claramente de una formación específica. (1) Representa el servicio de apoyo responsable de la adquisición, distribución, dispensación y control de los productos farmacéuticos circulantes. Dicha entidad provee al hospital de una asistencia esencial, pues es necesario asegurar un uso adecuado, racional y económico de los medicamentos.

Las farmacias hospitalarias manejan gran cantidad de información referente a los productos farmacéuticos que en estas se gestionan, los medios utilizados para el procesamiento de estos datos

actualmente son de forma manual, lo que conlleva a demoras en la recepción de los datos y el nivel de error en estos sea considerable.

Existen tarjetas de estantería, donde se registra toda la información necesaria del producto, pero esto dificulta la utilización y manipulación en general de los datos. La función de estas tarjetas es fundamentalmente archivar toda la información referente a la tramitación de los medicamentos, ya sean entregados, vencidos, solicitados o cualquier otra gestión que se necesite realizar. Esta información no tiene seguridad al ser almacenada de esta forma pues el deterioro por antigüedad, o negligencia, puede conllevar a la pérdida de dicha información. Por tal motivo se adoptan medidas de preservación creando copias de seguridad que pueden no implicar una desactualización en los datos replicados y pero sí un gasto excesivo de recursos (papel, tinta, etc.).

Para llevar a cabo las solicitudes de productos farmacéuticos de una sub-farmacia perteneciente a un servicio, se utilizan documentos de papel. También los pedidos que se realizan de la farmacia central a otras unidades y las devoluciones de medicamentos y productos entregados se archivan a través de planillas impresas. Esto provoca que existan dificultades cuando se necesita buscar información referente a la entrega de los mismos.

Para tener control sobre los consumos de cada producto en la farmacia por período de tiempo se realizan los cálculos de stock, estos cálculos se realizan con el objetivo de tener un estricto control de los valores máximos y mínimos de consumo para saber el comportamiento de estos y poder dar un tratamiento diferenciado. Todo este proceso se realiza de forma manual corriendo el riesgo de que existan errores generándose de esta forma información incorrecta.

El perfil farmacoterapéutico es un registro que se encarga esencialmente de controlar los medicamentos administrados a cada paciente y de llevar el cumplimiento de las indicaciones médicas. Dicha ficha se lleva manualmente, con riesgos de que pueda estar desactualizada y con peligro de deterioro físico. Es uno de los documentos más importantes, a la hora de sacar estadísticas o de consultar el estado de estabilidad de un paciente.

Las tarjetas, documentos y libros anteriormente mencionados han sido una solución parcial a la necesidad de gestión de los diferentes datos que se manipulan en las farmacias hospitalarias. Obviamente el servicio farmacéutico se ve afectado en gran medida debido a que gran parte de la gestión se hace de forma manual. Esto puede traer consigo la utilización ilícita de los medicamentos destinados a la atención médica de la población.

El control de medicamentos de la farmacia, en grandes centros en la actualidad, está orientado a departamentos, existe una entidad central, pero por sí misma no puede llevar el control de todo lo que sucede en las ramificaciones concernientes a los servicios del hospital. Esto puede provocar la detección tardía de errores internos que pueden traer graves consecuencias. Generalmente no existen mecanismos estadísticos automatizados que permitan al personal de farmacia detectar errores de gestión a tiempo o que sugieran posibles soluciones para normalizar y agilizar el funcionamiento de dicha entidad.

La incorporación de un sistema computarizado en el área de Farmacia de las instituciones hospitalarias, constituiría una herramienta para la mitigación de algunos de los problemas actuales del sistema de salud a nivel hospitalario y nacional como son: el retraso en los procesos que conduce a una demora en la atención al paciente, la falta de una historia clínica única, la carencia de información estadística en tiempo real a nivel de departamentos, la comunicación deficitaria y ausencia de controles de gestión automática.

En este sentido, el **Problema a resolver** es: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Farmacia de las instituciones hospitalarias?

El **Objeto de estudio** lo constituye el proceso de gestión de información en las instituciones hospitalarias, enmarcado en el **Campo de acción**, proceso de gestión de información en la atención de salud en el área de Farmacia de las instituciones hospitalarias.

Para resolver el problema identificado se propone el siguiente **Objetivo general**: Desarrollar el módulo Farmacia del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **Tareas a desarrollar**:

1. Evaluar las tendencias actuales en el mundo de los HIS.
2. Analizar los procesos de negocio asociados al área de Farmacia de las instituciones hospitalarias.
3. Asimilar la arquitectura definida por el Área Temática Gestión Hospitalaria para el desarrollo de sus aplicaciones.

4. Obtener mediante el Proceso Unificado de Desarrollo, los flujos de trabajo de “Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
5. Implementar los procesos: Unidad de nutrición parenteral, Unidad de anestesiología, Despachar medicamentos, Elaborar preparados y Farmacia satélite del módulo Farmacia del Sistema de Información Hospitalaria alas HIS.

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

1. Organización de los diferentes procesos que se llevan a cabo en el área de Farmacia.
2. Gestión y utilización óptima de recursos farmacéuticos.
3. Seguimiento continuo y actualizado de las fechas de vencimiento de los productos farmacéuticos.
4. Control estricto del consumo de los medicamentos, desglosado por áreas, permitiendo generar reportes de consumo por cada uno de los servicios del hospital, así como por pacientes hospitalizados.
5. Centralización y rápido acceso a toda la información.
6. Generación automática de la solicitud de los mismos de acuerdo a las necesidades.
7. Disponer de un sistema que tenga como centro de referencia al paciente, dando respuesta a sus necesidades de salud pública y de asistencia médica, facilitando su acceso a los recursos de salud.
8. Considerar a cada paciente como un ente común y único en todo el Sistema de Salud de Nacional.
9. Calidad del servicio médico.
10. Poner a disposición del profesional médico herramientas de gestión clínica y gestión administrativa que den respuesta a sus necesidades reales.

El presente documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **“FUNDAMENTACIÓN TEÓRICA”**, donde se realiza un estudio del estado del arte en cuando a sistemas de información hospitalarios, tecnologías, metodologías y herramientas existentes en el mundo.

El siguiente capítulo, **“CARACTERÍSTICAS DEL SISTEMA”**, contiene un marco conceptual asociado a la información que será manipulada por el sistema, arribando a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **“DISEÑO DEL SISTEMA”** se centra en la modelación detallada y la construcción de la estructura de la aplicación.

En el cuarto y último, **“IMPLEMENTACIÓN”**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los requerimientos de seguridad de los proyectos que pertenecen al Área Temática Gestión Hospitalaria.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza un estudio detallado de los principales conceptos y tecnologías utilizadas en el desarrollo del módulo Farmacia del Sistema de Información Hospitalaria alas HIS. Se abordan temas relacionados con la gestión de las farmacias hospitalarias y Sistemas de Gestión de Bases de Datos (SGBD).

1.1 Sistemas de salud pública

La forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado, es lo que se conoce como Sistema Nacional de Salud. 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”. Además es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la salud de una población. (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:

- I. 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.
- II. 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 (Especialidades). Comprende la atención médica brindada en los distintos Hospitales.

- III. 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. (3)

1.2 Sistemas de información hospitalaria

Es un sistema de información orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Permitiendo la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes burocráticos que enfrentan los pacientes.

Es una solución diseñada de manera modular, para articular de forma interdependiente las áreas de trabajo e integrar así al paciente, al personal médico, al personal de apoyo a la actividad médica, al personal administrativo y al personal directivo, empleando para esto las siguientes funcionalidades aplicadas a cada uno de los actores:

- I. Al Paciente: mediante la organización de su historia clínica, reducción de trámites y formatos.
- II. Al Personal Médico: mediante la organización de la información de las historias clínicas, notas médicas, tratamientos, indicaciones, generación de órdenes médicas, interconsultas, referencia y contra referencia, en línea y de manera centralizada.
- III. Al Personal de Apoyo: mediante la simplificación de los formatos que deben ser llenados, evitando la repetición de estudios, volver de nuevo a capturar la misma información, ya que toda esta se captura una sola vez y se registra donde ésta es generada inicialmente. Esto reduce las actividades de mensajería.
- IV. Al Personal Administrativo: el sistema actúa desde el momento en que la información de alguna actividad en la clínica u hospital se registra, genera a su vez, los movimientos contables correspondientes a las cuentas de ingresos y de egresos. Minimizando con ello el efecto replicable y optimizando el manejo de la logística de suministros.

- V. Al Personal Directivo: proporciona información resumida y detallada de las actividades vitales del hospital, como productividad de la planta médica, morbilidad en la consulta externa y en hospitalización, productividad de los servicios auxiliares de diagnóstico y situación financiera del hospital, entre otras. (4)

1.3 Descripción específica del módulo

La Farmacia Hospitalaria es el departamento dentro de la Institución de Salud responsable de la implantación del Mecanismo Especial de Protección (SSM, siglas en inglés), basado en la selección, planificación, adquisición, almacenamiento, distribución, control, producción, dispensación y provisión de información sobre productos farmacéuticos y afines, utilizados en el hospital. Es también partícipe del seguimiento al uso terapéutico que se les da a los medicamentos a través de actividades de fármaco vigilancia y de otra índole relacionada con los medicamentos. Es por tanto una unidad de apoyo asistencial.

1.4 Sistemas automatizados existentes vinculados al campo de acción

Se realizó un análisis de los principales sistemas informáticos que existen en el mundo vinculados a la gestión farmacéutica en general, con el objetivo de encontrar soluciones factibles a los problemas planteados de esta área. Se describieron las principales ventajas y desventajas de cada uno, así como las tecnologías y herramientas utilizadas para su desarrollo y posterior funcionamiento.

Los principales sistemas informáticos estudiados son los siguientes:

FARHOS (Gestión de farmacias hospitalarias)

Es un sistema de información para la gestión integral de la farmacia hospitalaria. Entre las principales funcionalidades que brinda están: la gestión de compras y almacén, la prescripción asistida desde planta, la distribución de medicamentos, la dispensación individualizada tanto a pacientes ingresados como ambulatorios. Está desarrollada con el modelo Cliente-Servidor. Como cliente utiliza cualquier máquina con el sistema operativo Windows 95, 98, NT, W2000 etc. El servidor puede estar sobre sistema operativo Windows o Unix. Como sistema gestor de base de datos posee Interbase 6, y como herramientas de desarrollo cliente MS-Access 2000.

TiCares-Dis

TiCares-Dis es un sistema de farmacia hospitalaria desarrollado por la empresa Telvent, que permite mantener un control de la distribución de los fármacos a las unidades asistenciales en la base de las prescripciones activas de los pacientes. Contempla distintos mecanismo de distribución de los fármacos: unidosis o multidosis intermedia u optimizada. Permite la consulta de tratamientos activos y del historial de tratamientos así como la gestión de devoluciones de fármacos. Está integrado con el sistema de información hospitalaria (HIS), TiCares, desarrollado por la misma empresa. (5)

HIS CNT Pacientes

La empresa CNT Sistemas de Información, establecida en Colombia con 25 años de experiencia, especializado para realizar soluciones integradas en el área de la salud. Presenta como su principal producto el HIS CNT Pacientes, el cual está diseñado para integrar todo el ciclo de atención del paciente frente a la prestación de servicios médicos, terapéuticos y de diagnósticos.

HIS CNT Pacientes, presenta el módulo Farmacia-Almacén. Dicho módulo permite optimizar la gestión de inventarios controlando la administración de depósitos de medicamentos. Provee funcionalidades para el registro de cotizaciones, órdenes de compra, entradas a almacén y demás movimientos. Posibilita crear diferentes niveles de Sub Stock o dependencias, así como el respectivo procesamiento de los pedidos de requisición y despachos a los diferentes Sub Stocks definidos. El módulo cuenta con funcionalidades para administrar el stock del depósito, permitiendo ajustes e informando qué ítems se encuentran por debajo del nivel mínimo del punto de reposición. Como desventaja presenta, que no posee el sistema de dispensación por dosis unitaria. (6)

SIGHO

SIGHO es un Sistema de Información para la Gerencia Hospitalaria, basado en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital. Se apoya de estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC.

Permite realizar registros individuales alrededor de la historia clínica electrónica, en cada uno de los módulos relacionados con la atención al paciente. Cuenta con 14 módulos dentro de los cuales se encuentra el módulo de Farmacia-Almacén. Este módulo permite administrar y tener un buen control de los medicamentos de la farmacia, entregados a pacientes atendidos en la unidad médica. Como desventaja presenta que es una aplicación de escritorio poco portable. No utiliza los estándares internacionales de codificación de medicamentos (como ATC) y sólo realiza despacho de

medicamentos a los stocks y recetas médicas sin crear un perfil farmacoterapéutico, utilizando el sistema de dispensación por dosis unitaria. (7)

Resumen

La principal desventaja que presentan estos sistemas radica en que ninguno da solución a todos los problemas planteados anteriormente como la dispensación por dosis unitaria, además agregar que todos son software propietario¹. Excepto el **TiCares-Dis**, todos los demás son aplicaciones de escritorio, careciendo así de multifuncionalidad con otros Sistemas Operativos (SO). La mayoría no utiliza estándares internacionales de codificación de medicamentos.

La solución informática debe reunir varias de las características que hacen innovadoras a cada una de las anteriormente expuestas. Debe estar desarrollada sobre software libre. Debe ser una aplicación web (modelo Cliente-Servidor), para así brindar gran versatilidad y escalabilidad en la gestión de la información independientemente del SO usado. Debe estar integrada al HIS (alás HIS) comunicándose directamente con los módulos de hospitalización y almacén, brindando un sistema de prescripción directa desde la orden médica emitida por el médico, al mismo tiempo que un control constante del consumo y existencia de cada uno de los medicamentos.

Además, debe regirse por el estándar internacional de ATC vigente desde 1996. El mismo define como codificar los medicamentos avalados por la OMS. El sistema debe contemplar las dos formas de dispensación de medicamentos: por dosis unitaria (sustentada por las farmacias satélites) y por reposición y paciente. La solución debe ser válida tanto: para pequeños hospitales (sin grandes procesamientos de información), como para grandes hospitales donde exista una farmacia central compuesta por varias farmacias satélites (en dependencia de las necesidades de la institución).

1.5 Tecnologías actuales a considerar

En todo proceso de desarrollo de software, es de vital importancia que exista una estrecha comunicación y entendimiento entre los diferentes entes que intervienen en la creación de un producto. Dicha cohesión dota al software de robustez, portabilidad y capacidad de asimilar cambios necesarios

¹ Software sin autorización para que cualquiera pueda usarlo, copiarlo, distribuirlo y modificarlo, gratuitamente o mediante una retribución. En particular, esto significa que el código fuente no está disponible.

sin la realización de profundas transformaciones. En este sentido los patrones arquitectónicos son de gran ayuda, así como una arquitectura claramente definida.

La arquitectura de software se define como:

“...conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software...”. (8)

Después de un profundo análisis de las características del software que se desea obtener, así como la experiencia y capacitación del equipo de desarrollo con que se cuenta, se definieron las siguientes tecnologías y herramientas a utilizar en el desarrollo del sistema a fin de lograr un producto robusto, costeable y eficiente.

1.5.1 **Java EE**

Java Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process.

1.5.2 **Seam**

JBoss Seam es una nueva y potente infraestructura para desarrollar aplicaciones Web 2.0 de próxima generación, al unificar e integrar tecnologías como AJAX, JavaServer Faces (JSF), Enterprise Java Beans (EJB3), Java Portlets, Business Process Management (BPM), Drools, Hibernate y JPA en una única solución.

Elimina la complejidad a nivel desde arquitectura hasta API, permitiendo la creación de complejas aplicaciones web basadas en Plain Old Java Objects (POJO), componentes de UI y el mínimo y solamente necesario XML. Se integra con librerías de controles de código abierto basadas en JSF como RichFaces y ICEFaces.

En la administración de estado, *Seam* provee una mayor granularidad de contextos. El principal, quizás, 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* o ventanas del navegador actividades del negocio con contextos completamente aislados. Además 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*). También a través de ficheros de reglas (*Drools*) define las posibles bifurcaciones del negocio permitiendo el fácil manejo de las condiciones sin tener que modificar el código fuente. (9)

1.5.3 JBoss AS como servidor de aplicaciones

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Está licenciado bajo la LGPL, por lo que puede libremente usarse sin costo alguno en cualquier aplicación comercial o ser redistribuido. Por ser una plataforma certificada JEE 5, soporta todas las especificaciones correspondientes, incluyendo servicios adicionales como *clustering*, *caching* y persistencia. *JBoss* es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta *Enterprise Java Beans* (EJB) 3.0. (10)

Los componentes claves son: *JBoss AS* 4.2, *Hibernate* 3.2.4, *Seam* 2.0.

1.5.4 JBoss Tools

JBoss Tools es un conjunto de *plug-ins* de Eclipse que tiene como objetivo ayudar a los desarrolladores a crear aplicaciones webs de forma rápida y sencilla.

Los módulos de *JBoss Tools* son:

- **RichFaces VE:** El editor visual aportado por Exadel proporciona el apoyo para la edición visual de páginas HTML, JSF, JSP y Facelets. También incluye soporte visual para las librerías de componentes JSF incluyendo *JBoss RichFaces*.
- **Seam Tools:** Incluye soporte para *seam-gen*, *RichFaces VE*.

- **Hibernate Tools:** Soporta el mapeo de archivos, anotaciones y JPA con la ingeniería inversa, completamiento de código, asistentes de proyecto, refactorización, ejecución interactiva de HQL/JPA-QL/Criteria.
- **JBoss AS Tools:** Fácil de iniciar, detener y debuguear al estar integrado con Eclipse. También incluye funciones para el despliegue eficaz de cualquier tipo de proyecto en el IDE.
- **Drools IDE:** Editor de ficheros de reglas, debugueo e inspección de reglas.
- **jBPM Tools:** Edición del flujo de trabajo del jBPM, motor de procesos BPM.
- **JBossWS Tools:** Desarrollo, invocación, inspección y pruebas de *webservices* sobre http con la adición y soporte de características JBossWS.

1.5.5 JSF

JavaServer Faces (JSF) es un *framework* para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Facelets como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL. (11)

1.5.6 Rich Faces 3.2 como librería de componentes JSF

RichFaces es una rica biblioteca de componentes para JSF que posee un avanzado marco para integrar fácilmente capacidades AJAX en el desarrollo de aplicaciones de negocios. Permite a los desarrolladores ahorrar tiempo y aprovechar las características de los componentes para crear aplicaciones Web, ricas en interfaz. Proporciona componentes fáciles de utilizar con etiquetas predefinidas, y brinda capacidades AJAX (Ajax4jsf). (12)

1.5.7 Ajax4jsf

Ajax4jsf es una librería *open source* que se integra totalmente en la implementación de JSF usada, 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 recargar por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. Estas características de la librería mencionada dotan a la aplicación JSF de contenido mucho más profesional con muy poco esfuerzo. (13)

1.5.8 PostgreSQL 8.3 como servidor de base de datos

PostgreSQL es un sistema de base de datos relacional que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Pertenece al ámbito del software libre, está bajo la licencia BSD (*Berkeley Software Distribution*). Esta licencia tiene menos restricciones en comparación con otras como la GPL (de la Fundación del Software Libre). La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Bajo esta licencia, el autor mantiene la protección de *copyright* únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación. A su vez, el usuario tiene libertad ilimitada con respecto al software, puede decidir incluso redistribuirlo como no libre. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros más.

Soporta ACID, o lo que es lo mismo, la realización de transacciones seguras; también, vistas, uniones, claves extranjeras, procedimientos almacenados, *triggers*, implementa internamente lenguajes de consulta de muy alto nivel como son el plpgsql (muy similar al plsql de Oracle), el plperl, el plpython y el c; además adicionar, que es un gestor preparado para incluirle lenguajes de consultas adicionales, habiendo la industria desarrollado alrededor de 20 lenguajes que este gestor soporta. Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp.

Para el control de réplicas en la actualidad existen variadas soluciones como es el Slony I (que permite réplicas asíncronas entre servidores de tipo *Master-Slave*) y el PgCluster (que permite la

creación de sistema de réplicas síncronos entre servidores de tipo *Master-Master*). Otras características interesantes de **PostgreSQL** son las siguientes:

- Alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella.
- Copias de seguridad en línea.
- Replicación asíncrona.
- Transacciones anidadas.
- Optimizador de consultas.
- Integridad referencial.

El tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla. El número de índices por tabla es también ilimitado.

Entre las ventajas que proporciona la versión 8.3 se encuentran:

- Soporte SQL/XML de acuerdo al estándar ANSI, incluyendo exportación en formato XML.
- Búsqueda en texto: ha sido incorporada la herramienta, TSearch2, en la distribución central, con mejor manejo y nuevos diccionarios e idiomas.
- Soporte de autenticación GSSAPI y SSPI.
- Nuevos tipos de datos: UUIDs, ENUMs y arreglos de tipos compuestos.

La versión 8.3 del gestor de base de datos ofrece mayor rendimiento que las versiones anteriores (entre un 5% y un 30% más dependiendo de la carga de trabajo). (14)

1.5.9 Framework Hibernate para acceso a los datos

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL. (15)

1.5.10 JPA (Java Persistence API)

Java Persistence API, más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs).

1.6 Lenguajes de programación

1.6.1 Java

Java es una plataforma de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

La plataforma Java consta de las siguientes partes:

- El lenguaje de programación, Java.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

El lenguaje mismo se inspira en la sintaxis de C++, pero su funcionamiento es más similar al de Smalltalk que a éste.

Java es un lenguaje orientado a objeto que 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. Tiene una arquitectura neutral, portátil y robusta. Posee las estructuras mínimas de un lenguaje de programación tradicional, sin añadir ninguna estructura más.

Es un lenguaje distribuido proporcionando una colección de clases para su uso en aplicaciones de red, que le permitan establecer conexiones con servidores o clientes remotos. Es compilado e interpretado a la vez: compilado al generar un código intermedio (*bytecodes*) que es interpretado por la máquina virtual. Proporciona numerosas comprobaciones en compilación y tiempo de ejecución y elimina la necesidad de liberación explícita de memoria, convirtiéndolo en un lenguaje robusto. Posee implementadas barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real. Está diseñado para soportar aplicaciones que serán ejecutadas en varios entornos de red: Unix, Windows, Mac. El *bytecodes* generado es indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas. Soporta sincronización de múltiples hilos de ejecución (*multithreading*).

1.7 Sistemas distribuidos

1.7.1 Modelo Cliente-Servidor

Esta arquitectura consiste básicamente en un programa cliente que realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la

misma. Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

1.7.2 Estilo arquitectónico

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción.

Modelo Vista Controlador (MVC)

Un propósito común en numerosos sistemas es el de tomar datos de un almacenamiento y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento. Dado que el flujo de información ocurre entre el almacenamiento y la interfaz, una tentación común, (hoy se llamaría un anti-patrón) es unir ambas piezas para reducir la cantidad de código y optimizar el rendimiento.

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- **Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** Maneja la visualización de la información.
- **Controlador:** Controla el flujo entre la vista y el modelo (los datos).

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

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

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

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes:

- Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una gran cantidad de requerimientos de actualización. (16)

1.8 Metodologías de desarrollo

1.8.1 RUP

Rational Unified Process (RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Utiliza UML para definir los modelos de software y se divide en 4 fases: Inicio, Elaboración, Construcción y Transición.

- Inicio: Tiene como objetivo determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: El objetivo es obtener la capacidad operacional inicial.
- Transición: Obtener el *release* del producto.

En esta metodología se han agrupado 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 flujos de apoyo. Tiene tres características esenciales: dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental. (17)

1.8.2 BPMN

BPMN (*Business Process Management Notation*) es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

BPMN nace debido al alto interés en las técnicas de modelado de procesos de negocio existente en junio del 2005, cuando la BPMI (*Business Process Management Initiative*) entra a formar parte de la OMG (*Object Management Group*) y fruto de esta unión, se publicó en febrero del 2006, la versión 1.0 final de BPMN, de la cual añadieron sus autores:

“BPMN se centra en los procesos de negocio y los Diagramas de Actividad de UML se centran en el diseño de software y por tanto no son competidoras, sino diferentes puntos de vista sobre un sistema.” (18)

Para modelar los procesos del negocio pueden utilizarse también los diagramas de actividades definidos por UML, pero BPMN brinda un conjunto de ventajas sobre los diagramas de actividades del estándar UML, como son:

- BPMN es más expresivo dando soporte (total o parcial) a 48 patrones de *workflow*, frente a los 42 de los diagramas de actividades.
- Es soportado por numerosas herramientas (al igual que los diagramas de actividades).
- Es más rico gráficamente y sus diagramas son más sencillos de comprender.
- Tiene el apoyo de la WfMC, una de las organizaciones más importante en el campo del *workflow*, que además de miembro de la OMG ha modificado una de sus especificaciones, XPD, para dar cobertura total a BPMN.

- BPMN puede transformarse directamente en BPEL, un lenguaje de orquestación de servicios web que se está consolidando como un estándar. (19)

1.8.3 UML

Lenguaje Unificado de Modelado (UML, por su nombre en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Dentro de sus características se pueden mencionar algunas tales como:

- Tecnología orientada a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo iterativo e incremental.
- Participación del cliente en todas las etapas del proyecto.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

Este puede ser utilizado en sistemas desarrollados en varios lenguajes de implementación y plataformas, incluyendo lenguajes de programación, bases de datos. Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos y está especialmente pensado para apoyar un estilo de desarrollo iterativo e incremental.

1.9 Herramientas de desarrollo

1.9.1 Eclipse como herramienta de desarrollo

Eclipse es un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment), multiplataforma, abierto y extensible. Eclipse es un IDE Java, aunque da soporte a otros lenguajes de programación, como son: C/C++, Cobol, Fortran, PHP o Python. Este entorno de desarrollo fue creado inicialmente por la IBM y actualmente es desarrollado por la Fundación Eclipse. Se le puede añadir a este IDE soporte de lenguajes adicionales mediante pluggins. Además la arquitectura plugin que presenta permite escribir cualquier extensión deseada en el ambiente.

Como características principales presenta:

- Ha sido diseñado de forma que pueda ejecutarse en cualquier plataforma. La última versión estable se encuentra disponible para los sistemas operativos Windows, Linux, Solaris, AIX, HP-UX y Mac OSX.
- Emplea un diseño basado en módulos (plug-in) los cuales se le pueden añadir para extender sus funcionalidades.
- Todas las versiones de Eclipse necesitan tener instalado en el sistema una máquina virtual Java (JVM), preferiblemente JRE (Java Runtime Environment) o JDK (Java Developer Kit) de Sun.
- Eclipse se distribuye bajo licencia EPL (Eclipse Public License). Esta licencia es considerada como libre por la FSF y por la OSI. La licencia EPL permite usar, modificar, copiar y distribuir nuevas versiones del producto licenciado. El antecesor de EPL es CPL (Common Public License) escrita por IBM.

1.9.2 PgAdmin como aplicación cliente para manejar la Base de datos

PgAdmin es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.

1.9.3 Visual Paradigm como herramienta de modelado

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos.

Provee soporte para la generación de código, tiene integración con diversos IDE's como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la

posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate.

Tiene dentro de sus características que es portable y posee gran facilidad de uso. Su diseño se centra en casos de uso y se enfoca al negocio que genera un software de mayor calidad. También tiene disponibilidad en múltiples plataformas. Soporta BPMN (Business Process Modeling Notation), modelado colaborativo con CVS y Subversion.

1.9.4 JRE

JRE es el acrónimo de Java Runtime Environment (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.

JVM (máquina virtual Java) es una instancia de JRE en tiempo de ejecución, este es el programa que interpreta el código Java y además por las librerías de clases estándar que implementan el API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto.

Un usuario sólo necesita el JRE para ejecutar las aplicaciones desarrolladas en lenguaje Java, mientras que para desarrollar nuevas aplicaciones en dicho lenguaje es necesario un entorno de desarrollo, denominado JDK, que además del JRE (mínimo imprescindible) incluye, entre otros, un compilador para Java.

Conclusiones

En este capítulo se realizó un estudio de los principales sistemas que existen a nivel nacional e internacional para la gestión farmacéutica en las instituciones hospitalarias. Los mismos no satisfacen las necesidades actuales de los procesos desarrollados en esta área, por lo que se decide implementar un sistema capaz de mitigar las principales deficiencias presentes en los sistemas analizados. Además se caracterizaron las herramientas y las tecnologías que son utilizadas en la solución propuesta. Estas dan medida de la robustez y potencia del software libre para el desarrollo de grandes proyectos. Siguiendo esta línea puede asegurarse el soporte y fiabilidad del producto a desarrollar.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se presentan los objetivos estratégicos de la organización, el flujo actual de los procesos involucrados en el campo de acción, descripción de los procesos que son objeto de automatización. Se plantean los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso del sistema.

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

Los procesos del negocio identificados en el modulo de farmacia están agrupados en 3 grandes grupos: farmacia central, dosis unitaria y dosis unitaria parenteral.

En estos 3 grupos se crearon atendiendo al lugar en donde se realizaban los procesos y los resultados que se debían obtener al realizar estos. Cada uno de estos grupos está compuesto por los siguientes procesos:

- Farmacia central
 - ✓ Trasladar medicamentos
 - ✓ Despachar medicamentos
 - ✓ Elaborar formulas magistrales y oficinales
- Dosis Unitaria
 - ✓ Reponer stock
 - ✓ Dispensar a sala
- Dosis unitaria parenteral
 - ✓ Unidad de mezcla intravenosa
 - ✓ Unidad de nutrición parenteral
 - ✓ Unidad de anestesiología

Trasladar medicamentos

El proceso comienza cuando el farmacéutico chequea todas las salidas de cada uno de los medicamentos, consultando la existencia de cada uno de estos en el stock de la farmacia central. Atendiendo el consumo promedio calculado para cada uno de los medicamentos en el tiempo que se haya establecido para las reposiciones del stock, se solicita el traslado de las cantidades necesarias de cada uno de los fármacos, pasando a formar parte del fondo fijo de la farmacia central. Posteriormente los mismos se distribuyen a los servicios y áreas del hospital que los soliciten.

Cada farmacéutico es responsable de realizar un conteo de las existencias del fondo fijo de la farmacia y dejar plasmadas dichas cantidades en la tarjeta de estantería de cada uno de los medicamentos, como inventario.

Despachar medicamentos

El proceso da comienzo cuando se reciben todas las solicitudes de medicamentos de los distintos servicios y áreas del hospital, que no posean una farmacia satélite asignada. Dichas solicitudes son verificadas por los asistentes de farmacia en la historia clínica de cada paciente y el récipe emitido por el médico. En el caso de que alguno de los medicamentos indicados este dentro de los medicamentos controlados, psicotrópicos, estupefacientes o productos biológicos, se verifica: la realización de la interconsulta con infectología para el caso de los controlados, la indicación en los rícpes blancos y morados en el caso de los psicotrópicos y estupefacientes respectivamente y el modelo de solicitud de producto biológico.

Después de verificados cada uno de los fármacos indicados el farmacéutico procede a autorizar las cantidades a despachar de cada uno de los mismos, cantidad que no tiene necesariamente que coincidir con la cantidad solicitada. Se finaliza totalizando las cantidades a despachar por servicio y área, procediendo al despacho de las mismas por el asistente de farmacia y el almacenista.

Elaborar formulas magistrales y oficinales

El proceso comienza al recepcionar las solicitudes de fórmulas Magistrales u Oficinales a realizar. En caso de ser una solicitud de fórmula Magistral, se realiza la solicitud de los productos necesarios para su elaboración. En caso de ser una solicitud de fórmula Ofical, se verifica la existencia de los productos para elaborarla, y de ser necesario, se realiza la solicitud de productos. En ambos casos después de tener las materias primas necesarias, se verifica la fórmula a preparar y se realizan los cálculos necesarios para su elaboración. Luego se registra en el control de preparación de formulas magistrales y oficinales en cada caso. Finalmente se rotula la fórmula creada. Si la misma es magistral, se procede a su entrega al paciente, de lo contrario se almacena hasta que se solicite.

Reponer stock

El proceso comienza al verificar la existencia de los medicamentos en la farmacia satélite (esta operación se realiza cada cierto tiempo determinado por el farmacéutico a cargo de la farmacia central del hospital). Atendiendo a su consumo y existencia, se procede a solicitar el traslado de los medicamentos necesarios desde la farmacia central a la farmacia satélite. Posteriormente se procede a dar entrada a cada uno y se registran en la tarjeta de estantería.

Diariamente el farmacéutico de la farmacia satélite debe verificar la existencia de los medicamentos del carro de paro y el stock de emergencia de cada uno de los servicios que atiende. En caso de ser necesario debe reponer los medicamentos ausentes o que no se encuentren en las cantidades requeridas. Además debe llevar un control de todos los estupefacientes y psicotrópicos despachados en el día. Mensualmente realiza la salida de todos los medicamentos despachados a los pacientes y envía dichas cantidades a la farmacia central, aunque debe realizar un inventario semanal que queda registrado en las tarjetas de estantería.

Dispensar a sala

El proceso comienza con la verificación por parte del farmacéutico de las órdenes médicas en la historia clínica de los pacientes. En caso que existiese algún error en las indicaciones dadas por el médico se le notifica el error detectado. Después de la verificación de las indicaciones se procede a la creación del perfil farmacoterapéutico en caso que el paciente sea de nuevo ingreso, de lo contrario, se efectúa a la actualización del mismo en caso de ser necesario.

El perfil almacena todos los medicamentos indicados al paciente en el transcurso de su hospitalización así como el cumplimiento de los mismos. Atendiendo a la información existente en los perfiles de cada paciente, se preparan las cajas de tratamiento con los medicamentos indicados para esa fecha. Estas cajas son entregadas a las enfermeras al frente de las salas y retiradas al siguiente día antes de entregar el nuevo tratamiento. Si queda algún medicamento sin cumplir a la hora de retirar las cajas, éste se registra en el perfil como una devolución.

Unidad de nutrición parenteral

El proceso comienza con la recepción de las solicitudes de elaboración de nutrición por parte del farmacéutico. Este evalúa dichas solicitudes y las acepta o devuelve con las sugerencias necesarias. De las solicitudes aprobadas se procede a crear el perfil de nutrición parenteral al paciente, en el caso de que no lo posea aún. Posterior a la elaboración de las nutriciones aprobadas, se crea el rótulo de

cada una de ellas, luego se registra en el control de reparto y se entrega en la sala donde se encuentra el paciente hospitalizado.

Unidad de anestesiología

El proceso comienza con la revisión diaria del estado de los stocks de quirófano por parte del farmacéutico, realizando las reposiciones de los medicamentos en donde sea necesario. Además realiza la revisión del plan de actividades quirúrgicas electiva, con el objetivo de crear el cronograma de preparaciones intravenosa. Por cada una de las preparaciones se realizan los cálculos pertinentes para su elaboración, se crean los rótulos y almacenan, para luego entregar las bandejas de inducción de cada intervención quirúrgica, registrando su salida.

Unidad de mezclas intravenosas

El proceso comienza recepcionando las prescripciones de mezclas intravenosas. El farmacéutico procede a la verificación de la indicación. En caso de detectar algún error, se consulta con el médico proscriptor y se corrige. Luego se registran las mezclas a preparar creando un control de preparación. Posterior a su preparación se crean los rótulos de cada una, se actualiza el control de reparto y se entregan dichas mezclas al paciente para su administración final. Al cierre del día se realiza un recorrido de verificación de la administración de las mezclas intravenosa. En caso de detectarse algún incumplimiento, se registran las causas por las cuales no se le administró.

2.2 Modelo de negocio

El modelado del negocio es la primera disciplina 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 actuales existentes de la entidad a la cual se le va a desarrollar el sistema. En este flujo de trabajo es donde se conocen a fondo cómo son iniciadas cada una de las actividades de un sector determinado. A través del modelado de estos procesos se obtiene una visión más amplia del negocio existente.

Actores del negocio

Un actor del negocio es el principal beneficiado de los procesos existentes. Es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa. El actor que se modela no es más que un rol que se juega cuando se interactúa con el negocio.

Actor del negocio	Descripción
Solicitante.(Medico, Servicio, Unidad de soporte nutricional, Enfermera)	Es el encargado de solicitar productos farmacéuticos a la farmacia según las necesidades que presenta el servicio hospitalario que representa, además de devolver los productos no utilizados.
Proveedor.	Es el encargado de hacer las distribuciones de productos de las unidades proveedoras a las farmacias.

Tabla 1 Actores del negocio.

Trabajadores del negocio

Un trabajador del negocio es la abstracción de una persona o sistema que representa un rol desempeñado en las realizaciones de CUN. Colabora con otros trabajadores, es notificado de los eventos del negocio y manipula las entidades del negocio para realizar sus responsabilidades.

Trabajador	Descripción
Farmacéutico	Principal protagonista de los procesos de negocio en farmacia, es el encargado de interactuar con todas las solicitudes y servicios que brinda la farmacia hospitalaria, mayormente autorizando estos.

Asistente de farmacia	Es el encargado de despachar las solicitudes autorizadas por el farmacéutico así como registrara las salidas de todos los productos farmacéuticos que se dispensen al hospital. Elabora las fórmulas, mezclas intravenosas y nutriciones parenterales solicitadas a la farmacia.
Almacenista	Es el encargado de darle salida y entrada a todos los medicamentos del almacén de farmacia, así como realizar inventario periódicamente a los mismos.

Tabla 2 Trabajadores del negocio.

Diagramas de procesos del negocio

Los modelos de procesos son una representación abstracta de los proceso de una organización. Muestran como y quien efectúa las actividades que generan valor para la organización, los actores involucrados en el proceso, los recursos consumidos y los eventos que dirigen a estos. De las actividades exponen: cuales son las operativas, cuales son ejecutables y por quien, sus estradas y salidas y la secuencia de todas las actividades dentro del proceso.

Los diagramas de procesos del negocio (Ver anexo 1) son basados en las técnicas de diagramas de flujos, adaptados para graficar las operaciones de los procesos. Se componen de un conjunto de elementos gráficos que facilitan un diagrama entendible y de fácil comprensión tanto por audiencias de negocio como técnicas.

2.3 Objeto de automatización

La automatización de los siguientes procesos optimizará en gran medida la gestión hospitalaria elevando progresivamente la calidad y eficiencia del servicio farmacéutico. Los cinco procesos a automatizar son los siguientes:

Farmacia central

Se desean automatizar la dispensación, distribución, y control de medicamentos a toda el área, los reportes estadísticos sobre el estado y consumo de productos farmacéuticos, la recepción de las solicitudes de medicamentos de los distintos servicios y áreas del hospital que no posean Unidosis y la verificación y autorización de las cantidades a despachar por cada solicitud.

Unidosis

Se desean automatizar todas las actividades concernientes de la dispensación por dosis unitaria y registro del destino final de los productos farmacéuticos, así como el perfil farmacoterapéutico de cada uno de los pacientes.

Dispensario

Se desea automatizar la recepción de las solicitudes de fórmulas magistrales y oficinales así como el registro y dispensación de las mismas.

Unidad de nutrición parenteral

Se desea automatizar la recepción de las solicitudes de nutriciones parenterales así como el registro, dispensación y cumplimiento de las mismas.

Unidad de mezcla

Se desea automatizar la recepción de las solicitudes de mezcla intravenosa así como dispensación y control de elaboración.

Unidad de anestesiología

Se desea automatizar las solicitudes y la dispensación de las bandejas de inducción.

2.4 Propuesta del sistema

Esta aplicación está diseñada con el objetivo de optimizar la gestión de la farmacia hospitalaria del sistema alas HIS automatizando la dispensación y control de productos farmacéuticos de una manera cómoda y eficiente, además brinda la posibilidad de acceder a toda la información que se genera producto a todo el flujo de los medicamentos dentro del ente hospitalario.

El *software* se desarrollará sobre la base de una aplicación Web posibilitándoles a los usuarios los consecuentes beneficios que estas aplicaciones reportan como son: el acceso al sistema por medio de un servidor Web, así como las actualizaciones y los mantenimientos del software sin necesidad de redistribuir e instalar. La persistencia de la información se realizará a través de una base de datos localizada en el servidor central asegurando mediante mecanismos de seguridad la fidelidad de la misma en consecuencia con las actualizaciones que pueda tener desde otro módulo o sistema.

2.4.1 Requerimientos funcionales

RF1 Autorizar solicitud de medicamento.

RF2 Despachar solicitud de medicamento.

RF3 Rechazar solicitud de medicamento.

RF4 Generar reporte de consumo de psicotrópicos y estupefacientes.

RF5 Generar registro de salida de medicamentos.

RF6 Generar control de hemoderivados.

RF7 Registrar psicotrópico con devolución de envase.

RF8 Registrar devolución de envase de psicotrópico.

RF9 Crear fórmula magistral.

RF10 Modificar fórmula magistral.

RF11 Ver datos de la fórmula magistral.

RF12 Buscar fórmula magistral.

RF13 Eliminar fórmula magistral.

RF14 Crear fórmula oficial.

RF15 Modificar fórmula oficial.

RF16 Ver datos de la fórmula oficial.

RF17 Buscar fórmula oficial.

RF18 Eliminar fórmula oficial.

RF19 Listar solicitudes de fórmulas magistrales.

RF20 Listar solicitudes de fórmulas oficiales.

RF21 Imprimir etiqueta de fórmula.

RF22 Rechazar solicitud de fórmula.

RF23 Generar reporte de preparación de fórmulas magistrales.

RF24 Generar reporte de preparación de fórmulas oficiales.

RF25 Generar control de entrega de fórmulas magistrales.

RF26 Ver perfil farmacoterapéutico.

RF27 Actualizar perfil farmacoterapéutico.

RF28 Eliminar perfil farmacoterapéutico.

RF29 Buscar perfil farmacoterapéutico.

RF30 Ver cumplimiento de los medicamentos indicados.

RF31 Registrar detección de error de medicación o sugerencia sobre el medicamento indicado.

RF32 Listar medicamentos por paciente para un día determinado.

RF33 Registrar despacho de medicamentos del día.

RF34 Generar reporte de consumo de medicamentos mensual por paciente.

RF35 Generar reporte de consumo de medicamentos mensual por medicamento.

RF36 Registrar reposición de stock de emergencia.

RF37 Registrar reposición de carro de paro.

RF38 Generar reporte de control de estupefacientes en la unidosis.

RF39 Generar reporte de control de psicotrópicos en la unidosis.

RF40 Registrar reempaque de dosis orales.

RF41 Imprimir etiquetas de reempaque.

RF42 Buscar control de reempaque.

RF43 Ver control de reempaque.

RF44 Autorizar solicitud de mezcla intravenosa.

RF45 Rechazar solicitud de mezcla intravenosa.

RF46 Ver cronograma de preparación de mezclas intravenosas.

RF47 Imprimir etiqueta de mezcla intravenosa.

RF48 Autorizar solicitud de nutrición parenteral.

RF49 Rechazar solicitud de nutrición parenteral.

RF50 Ver perfil de nutrición parenteral.

RF51 Eliminar perfil de nutrición parenteral.

RF52 Buscar perfil de nutrición parenteral.

RF53 Ver cumplimiento de nutrición.

RF54 Ver control de reparto de nutriciones parenterales.

RF55 Listar solicitudes de bandejas de inducción.

RF56 Ver solicitud de bandeja de inducción.

RF57 Dispensar solicitud de bandeja de inducción.

RF58 Registrar segunda solicitud.

RF59 Registrar devolución de medicamentos de la bandeja de inducción.

RF60 Ver codificación ATC.

RF61 Modificar nivel de ATC.

RF62 Insertar nivel al ATC.

RF63 Insertar medicamento al ATC.

RF64 Eliminar nivel al ATC.

RF65 Eliminar medicamento de ATC.

RF66 Seleccionar medicamento.

RF67 Crear solicitud de nutrición parenteral.

RF68 Modificar solicitud de nutrición parenteral.

RF69 Eliminar solicitud de nutrición parenteral.

RF70 Crear solicitud de mezcla intravenosa.

RF71 Modificar solicitud de mezcla intravenosa.

RF72 Eliminar solicitud de mezcla intravenosa.

RF73 Crear solicitud de bandeja de inducción.

RF74 Modificar solicitud de bandeja de inducción.

RF75 Eliminar solicitud de bandeja de inducción.

RF76 Crear solicitud de fórmula magistral.

RF77 Modificar solicitud de fórmula magistral.

RF78 Eliminar solicitud de fórmula magistral.

RF79 Crear solicitud de fórmula oficial.

RF80 Modificar solicitud de fórmula oficial.

RF81 Eliminar solicitud de fórmula oficial.

2.4.2 Requerimientos no funcionales

2.4.2.1 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

2.4.2.2 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.

2.4.2.3 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.

2.4.2.4 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.

2.4.2.5 Hardware

Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 7 o superior o Firefox 2.x).

Por lo que se escogieron estaciones de trabajo de 256Mb de memoria RAM y un microprocesador de 2.0Hz con Sistema operativo Linux.

Servidores

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

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

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

2.4.2.6 **Software**

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

El sistema deberá disponer de un navegador web, estos pueden ser IE 7 o superior, Opera 9 superior, Google chrome 1 o superior y Firefox 2 o superior.

2.4.2.7 **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.

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

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

2.4.2.9 **Interfaz**

Interfaces de usuario

Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.

La interfaz contará con teclas de función y menús desplegados que faciliten y aceleren su utilización.

La entrada de datos incorrecta será detectada claramente e informada al usuario.

Todos los textos y mensajes en pantalla aparecerán en idioma español.

Interfaces de comunicación

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven).

El sistema usará el formato estándar WSDL para la descripción de los servicios web.

El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.

El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

2.4.2.10 Portabilidad

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

2.4.3 Modelo de casos de uso del sistema

2.4.3.1 Definición de actores

Actor	Descripción
Usuario	Usuario global que permite la autenticación en el sistema y que valida al mismo dándole un rol.
Asistente de farmacia	Es el encargado de despachar los medicamentos solicitados por los servicios. Realizar las distintas transacciones de entradas (devolución de despachos) y salidas (despachos, transferencia a almacenes auxiliares) del inventario diariamente, las preparación de fórmulas magistrales y oficinales que sean requeridas para los pacientes hospitalizados, el inventario físico de los medicamentos al finalizar la jornada de trabajo y hacer la entrega al turno siguiente. Registrar en el control de sustancias estupefacientes y psicotrópicas, los medicamentos que hayan sido despachados durante la jornada de trabajo, siguiendo los procedimientos establecidos para tal fin. Totalizar todos los medicamentos del stock.

<p>Farmacéutico</p>	<p>Encargado de llevar estadísticas de consumo y determinar el consumo mensual de cada uno de los productos que se dispensan a través de la farmacia central.</p> <p>Supervisar las labores realizadas por el asistente de farmacia. Realiza los pedidos semanales para la reposición de las existencias de los medicamentos y material médico-quirúrgico a los diferentes depósitos de la Institución, los inventarios semanales de los medicamentos almacenados en la unidosis. Autoriza el despacho de las solicitudes de medicamentos, nutriciones parenterales y mezclas intravenosas. Lleva un control de las salidas de los medicamentos, la relación de pacientes que reciben sustancias estupefacientes y psicotrópicas. Supervisar el reempaque de sólidos y líquidos. Lleva estadísticas de consumo de medicamentos por paciente.</p>
---------------------	--

Tabla 3 Actores del sistema.

2.4.3.2 Vista global de actores

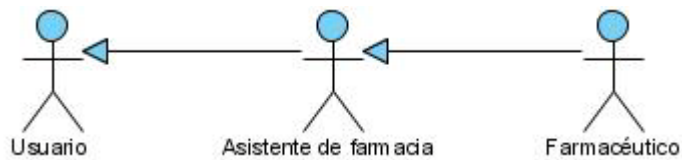


Figura 2.1 Actores del sistema.

2.4.3.3 Diagrama de casos de uso

Farmacia central

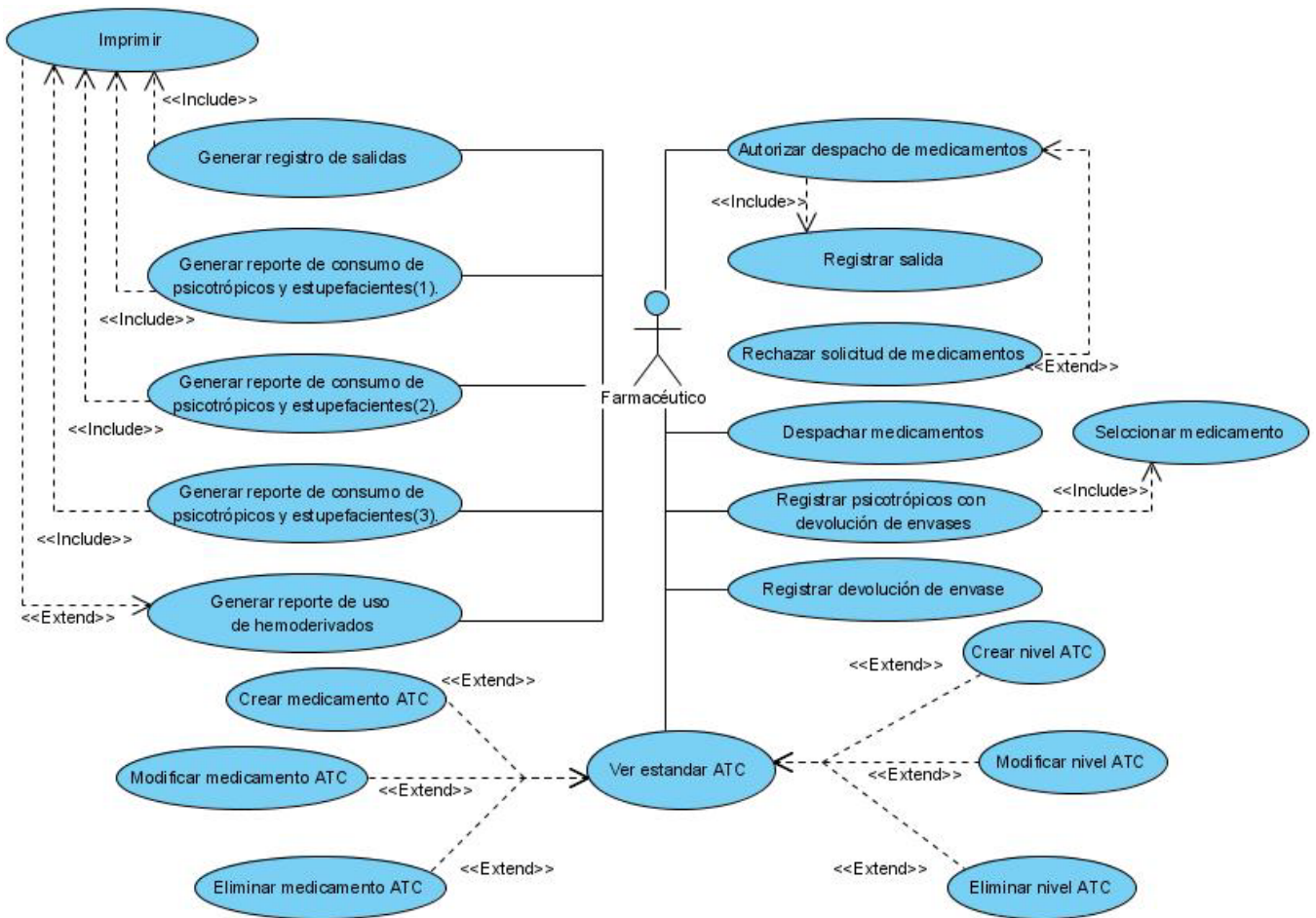


Figura 2.2 Diagrama de casos de uso del sistema. Proceso “Farmacia central”.

Unidad de mezcla

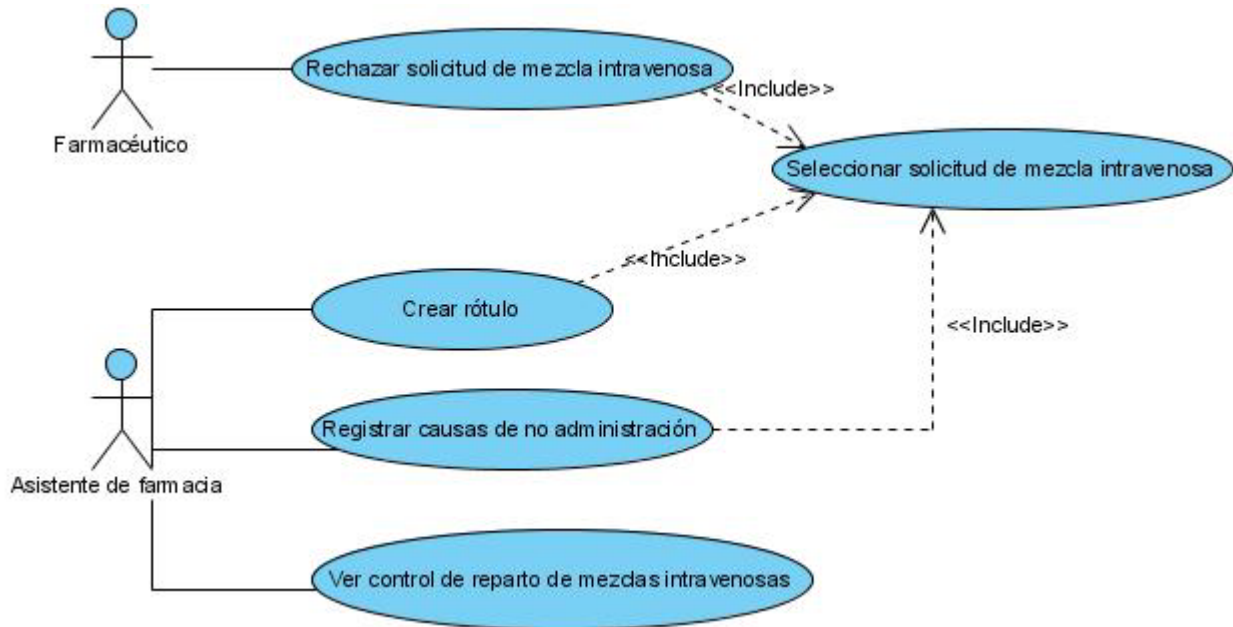


Figura 2.5 Diagrama de casos de uso del sistema. Proceso “Unidad de mezcla”.

Unidad de nutrición parenteral

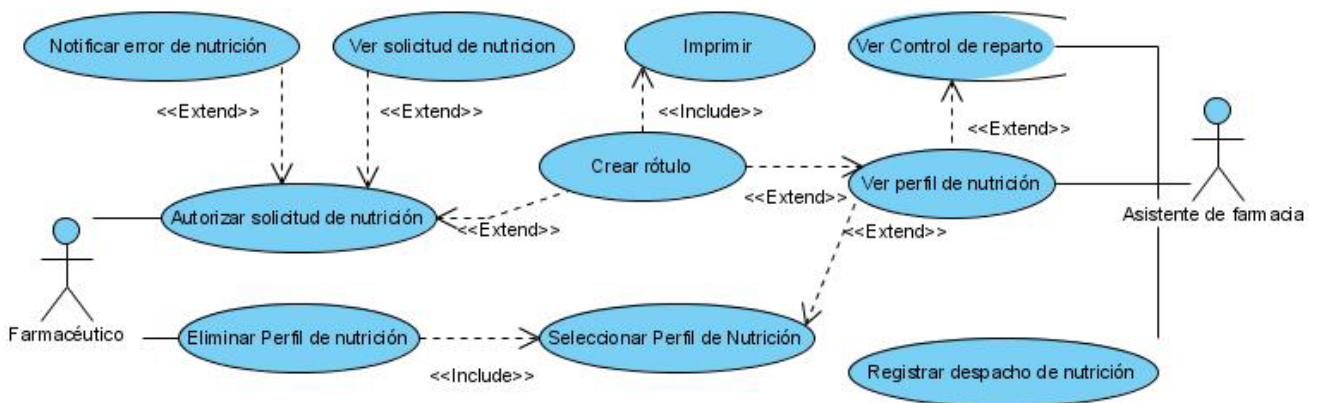


Figura 2.6 Diagrama de casos de uso del sistema. Proceso “Unidad de nutrición parenteral”.

Unidad de anestesiología

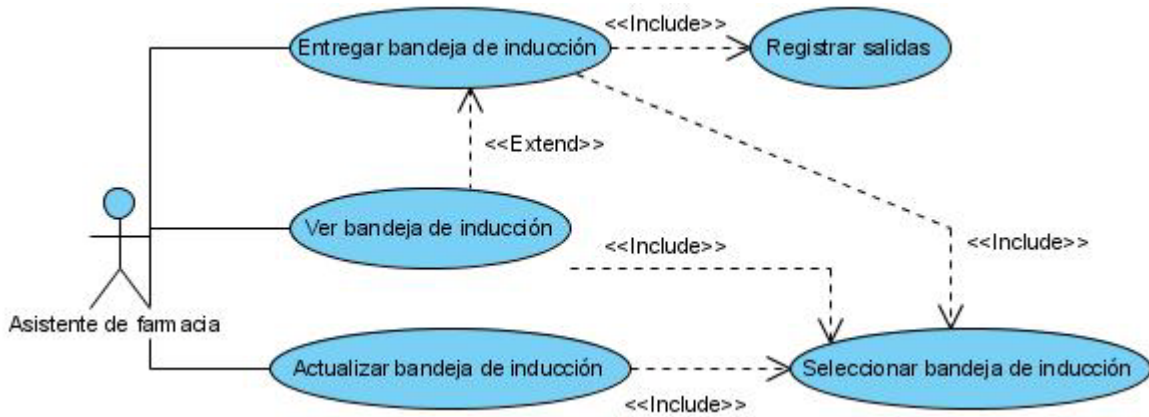


Figura 2.7 Diagrama de casos de uso del sistema. Proceso “Unidad de anestesiología”.

2.4.3.4 Descripción textual de los caso de uso.

CU-1	Ver estándar ATC
Actor	Farmacéutico
Propósito	Brindarle al usuario el estándar internacional ATC con toda su estructura de niveles.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Ver estándar ATC. El sistema muestra la estructura del estándar ATC, organizada en forma de un árbol, permitiéndole al usuario navegar por cada uno de sus niveles.
Referencia	RF 60
Precondiciones	No existen.
Poscondiciones	El sistema mostró toda la estructura del estándar ATC.

Tabla 4 Descripción textual CU-1: Ver estándar ATC

CU-2	Crear nivel ATC: [Extendido CU-1: Ver estándar ATC]
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de agregar un nuevo nivel al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Agregar nivel al

	estándar ATC. El sistema le brinda la posibilidad de introducir los datos del nuevo nivel y su localización, el usuario introduce los datos, el sistema crea el nuevo nivel. El caso de uso termina.
Referencia	RF 62
Precondiciones	No existen.
Poscondiciones	El sistema creó un nivel al estándar ATC

Tabla 5 Descripción textual CU-2: Crear nivel ATC

CU-3	Modificar nivel ATC: [Extendido CU-1: Ver estándar ATC]
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de modificar un nivel al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Actualizar nivel. El sistema le muestra los datos del nivel y le permite cambiar los mismos, el usuario reemplaza los datos que desee, el sistema modifica el nivel. El caso de uso termina.
Referencia	RF 61
Precondiciones	Seleccionar nivel ATC.
Poscondiciones	El sistema modificó un nivel al estándar ATC.

Tabla 6 Descripción textual CU-3: Modificar nivel ATC

CU-4	Eliminar nivel ATC: [Extendido CU-1: Ver estándar ATC]
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de eliminar un nivel al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Eliminar nivel. El sistema elimina el nivel. El caso de uso termina.
Referencia	RF 64
Precondiciones	Seleccionar nivel ATC.
Poscondiciones	El sistema eliminó un nivel al estándar ATC.

Tabla 7 Descripción textual CU-4: Eliminar nivel ATC

CU-5	Crear medicamento ATC: [Extendido CU-1: Ver estándar ATC]
-------------	--

Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de agregar un nuevo medicamento al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Agregar medicamento al estándar ATC. El sistema le brinda la posibilidad de introducir los datos del nuevo medicamento y su localización, el usuario introduce los datos, el sistema crea el nuevo medicamento. El caso de uso termina.
Referencia	RF 63
Precondiciones	No existen.
Poscondiciones	El sistema añadió un medicamento al estándar ATC

Tabla 8 Descripción textual CU-5: Crear medicamento ATC

CU-6	Modificar medicamento ATC: [Extendido CU-1: Ver estándar ATC]
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de modificar un medicamento al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Actualizar medicamento. El sistema le muestra los datos del medicamento y le permite cambiar los mismos, el usuario reemplaza los datos que desee, el sistema modifica el medicamento. El caso de uso termina.
Referencia	RF 61
Precondiciones	Seleccionar medicamento ATC.
Poscondiciones	El sistema modificó un medicamento al estándar ATC.

Tabla 9 Descripción textual CU-6: Modificar medicamento ATC

CU-7	Eliminar medicamento ATC: [Extendido CU-1: Ver estándar ATC]
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de eliminar un medicamento al estándar ATC.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Eliminar medicamento. El sistema elimina el medicamento. El caso de uso termina.
Referencia	RF 65

Precondiciones	Seleccionar medicamento ATC.
Poscondiciones	El sistema eliminó un medicamento al estándar ATC.

Tabla 10 Descripción textual CU-7: Eliminar medicamento ATC

CU-8	Autorizar solicitud de medicamentos
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de autorizar la cantidad a despachar en una solicitud de medicamento realizada a la farmacia.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Autorizar solicitud de medicamentos. El sistema lista las solicitudes sin autorizar existentes y le brinda la posibilidad al actor de introducir la cantidad deseada, el actor entra las cantidades autorizadas, el sistema actualiza las solicitudes. El caso de uso termina.
Referencia	RF 1
Precondiciones	Existir solicitudes de medicamentos por autorizar
Poscondiciones	El sistema actualiza las solicitudes de medicamentos.

Tabla 11 Descripción textual CU-8: Autorizar solicitud de medicamentos

CU-9	Rechazar solicitud de medicamento
Actor	Farmacéutico
Propósito	Brindarle al usuario la posibilidad de rechazar la solicitud de medicamentos a la farmacia, reflejando las causas.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Rechazar solicitud de medicamentos. El sistema permite entrar las causas por las cuales se rechaza la solicitud del o los medicamentos, el usuario introduce las causas, el sistema registra las mismas. El caso de uso termina.
Referencia	RF 3
Precondiciones	Seleccionar una solicitud de medicamento.
Poscondiciones	El sistema registró una solicitud de medicamento como rechazada.

Tabla 12 Descripción textual CU-9: Rechazar solicitud de medicamentos

CU-9	Despachar solicitud de medicamento
Actor	Asistente de farmacia
Propósito	Brindarle al usuario la posibilidad de registrar como despacha una solicitud de medicamentos previamente autorizada.
Resumen	El Caso de Uso comienza cuando el actor accede a la opción Despachar solicitud de medicamentos. El sistema permite que se entren los criterios para buscar la solicitud a despachar, luego permite que el funcionario que retira los medicamentos introduzca su usuario y contraseña del sistema. El sistema registra la solicitud como despachada y rebaja de las existencias de los medicamentos las cantidades sustraídas. El caso de uso termina.
Referencia	RF 2
Precondiciones	Existan solicitudes de medicamentos autorizadas.
Poscondiciones	El sistema registró el despacho de medicamentos.

Tabla 13 Descripción textual CU-10: Despachar medicamentos

Conclusiones

Al finalizar este capítulo se detalló el funcionamiento de cada uno de los procesos de la farmacia hospitalaria, así como los principales problemas existentes que dificultan el trabajo farmacéutico. Como resultado de este análisis se deriva el diseño de una herramienta automatizada que mejore el funcionamiento de dichos procesos. Se realizó el modelo del negocio y se identificó los actores, trabajadores y casos de uso del negocio, así como los requerimientos funcionales y no funcionales del sistema, basados en los requisitos de desarrollo que propone la metodología RUP con UML.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En el presente capítulo se expone el diseño propuesto para la solución de la aplicación, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. Se muestran los principales diagramas de clases y secuencia del modelo de diseño.

3.1 Descripción de la arquitectura

La Arquitectura de Software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. Su objetivo principal es aportar elementos que ayuden a la toma de decisiones y al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto.

Para el desarrollo del sistema y en base a las herramientas, tecnologías y metodologías propuestas se define como parte de la línea base de la Arquitectura la implementación del patrón de diseño MVC. Es una arquitectura en N capas definida por tres capas principales: Capa de presentación, capa de negocio y capa de acceso a datos.

La capa de presentación está conformada principalmente por paginas XHTML y controles JSF, Seam UI y Richfaces los cuales se encargan de validar y capturar los datos que introduzca el usuario. Estos componentes enriquecen la interfaz de usuario dotándola de un agradable diseño y vistosidad así como la optimización del envío y carga de datos mediante los componentes ajax4jsf.

La capa de negocio está constituida por clases controladoras que se encargan de definir la lógica del negocio así como del manejo y validación de los datos capturados en la capa de presentación además. Estas clases se pueden ubicar mediante anotaciones seam, en distintos contextos que les permiten mantener el estado de los datos que manejan. En esta capa se manejan también las reglas del negocio a través de Drools esto le da mayor dinamismo y funcionalidad al sistema.

La capa de acceso a datos se encarga principalmente de la carga, modificación, eliminación y persistencia de la información en la base de datos. Esta capa valida también los datos antes de persistirlos. Todo este manejo de datos es mediante Hibernate que abstrae al desarrollador del gestor

de base de datos utilizado a través del mapeo de tablas, esto permite llevar las consultas a un lenguaje de objetos. (20)

3.2 Modelo de diseño

Un modelo de Diseño es un artefacto que describe la realización de los casos de uso, centrándose gráficamente en la lógica de los requerimientos no funcionales y en el dominio de la solución aplicada al entorno de implementación. En el diseño se modela el sistema y encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los funcionales y las restricciones que se le suponen.

3.3 Patrones de diseño

"Cada patrón de diseño describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces."

Christopher Alexander²

Los patrones de diseño son soluciones a problemas comunes en el diseño de aplicaciones, es de buenas prácticas de programación implementarlos, pues ahorran tiempo y mejoran el software haciéndolo más eficiente, dinámico y seguro. No es más que la solución efectiva a un problema en un momento dado y puede ser reusable aplicándose a diferentes problemas de diseño en distintas circunstancias.

La etapa de implementación requiere de un buen diseño de clases basado en patrones que permitan aumentar la reutilización de código y contribuir al óptimo manejo de los sus instancias. *Seam* asegura

² Arquitecto austriaco, reconocido por sus diseños destacados de edificios en California, Japón y México. Partiendo de la premisa de que los usuarios de los espacios arquitectónicos saben más que los arquitectos sobre el tipo de edificios que necesitan, creó y validó (junto a Sarah Ishikawa y Murray Silverstein) el término lenguaje de patrón, un método estructurado que pone la arquitectura al alcance de personas no especializadas profesionalmente en la materia, y que popularizó en su libro *A Pattern Language*.

la persistencia de datos mediante la abstracción que brinda *Hibernate* del gestor de bases de datos a utilizar y al mapeo de las entidades del sistema.

EntityManagerFactory que es uno de los componentes principales de la Arquitectura JPA (Java Persistence API), pues es el encargado de crear objetos de *EntityManager* cuando el mismo se inyecte en algún contexto de la aplicación. Dicho componente implementa el patrón de diseño **Abstract Factory** permitiendo el acceso a la base de datos. *EntityManager* es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones, cada instancia puede realizar operaciones como inserción, lectura, modificación y eliminación (CRUD por sus siglas en inglés) sobre un conjunto de objetos persistentes.

Hibernate implementa también el patrón **Active record** donde una fila en la tabla de la base de datos se envuelve en una clase, de manera que se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado. También implementa los patrones de mapeo: **Identity field**, **Foreign Key Mapping** y **Association Table Mapping**. El primero centra su atención en la forma en que los objetos se relacionan en memoria, consiste básicamente en la generación de un único ID para definir la identidad de una entidad. El segundo es utilizado para mapear relaciones de uno a muchos y el tercero para mapear relaciones de muchos a muchos.

Hibernate implementa además los patrones de comportamiento **Identity map** y **Lazy Load**. El primero es utilizado para evitar tener en memoria dos representaciones distintas del mismo objeto en una transacción de negocio; funciona como una caché de objetos de negocio. El segundo resuelve problemas de carga desmesurada y dependencias circulares o sea optimiza la carga de datos de la base de datos manteniendo en memoria sólo los que se invoquen en cada momento.

Uno de los patrones más importantes que implementa *Hibernate* es el patrón **Query object** que se encarga de interpretar la estructura de objetos y traducirlos a consultas (queries) SQL. Mediante este patrón se pueden crear estas consultas, haciendo referencia a las clases y sus campos, en lugar de tablas y columnas independientemente del esquema en que puedan estar.

Los patrones **GRASP** (Patrones para asignar responsabilidades) tuvieron una importante utilidad en el diseño realizado. A cada clase le fueron asignadas las tareas que podían realizar según la información que poseía, además de crear las instancias de otras clases en correspondencia con la responsabilidad dada, poniéndose de manifiesto los patrones **Experto** y **Creador** siendo el primero uno de los más

usados. Con esto se logró conservar el encapsulamiento pues los objetos logran valerse de su propia información para realizar lo que se les pide.

El diseño obtenido cumple con los patrones de **Bajo acoplamiento** y **Alta cohesión** permitiendo la colaboración entre los elementos del diseño (clases), sin verse afectados la reutilización de los mismos y el entendimiento de estos cuando se encuentran aislados. La creación de clases controladoras facilitó realizar las operaciones del sistema, debido a que estas operaciones reflejan los procesos de la empresa o dominio y no es factible manejarse en la capa de interfaz o presentación.

3.3.1 Patrón MVC + JSF

JSF encaja bien en la arquitectura de la capa de presentación basada en MVC. Ofrece una clara separación entre el comportamiento y la presentación. Une los familiares componentes UI con los conceptos de la capa-Web sin limitarnos a una tecnología de script o lenguaje de marcas particular.

Los beans a los que se vinculan los formularios JSF son el modelo. También contienen acciones, que son una extensión de la capa del controlador y delegan las peticiones del usuario a la capa de la *lógica de negocio*. Desde la perspectiva de la arquitectura general de la aplicación, también se puede referir a la capa de *lógica de negocio* como la capa del modelo. Las páginas Facelets con etiquetas JSF personalizadas son la capa de la vista. El Servlet Faces proporciona la funcionalidad del controlador.

3.4 Estructuración

Para la realización del Modelo de Diseño, se organizó su estructura en fragmentos manejables, llamados paquetes, que encapsulan las clases: entidades, controladoras e interfaces. Estos paquetes guardan relaciones entre sí, estableciendo las dependencias entre las distintas clases que contienen. A continuación se presenta el diagrama de paquetes del diseño, con la estructura que estos presentan en el módulo. (Figura 3.1)

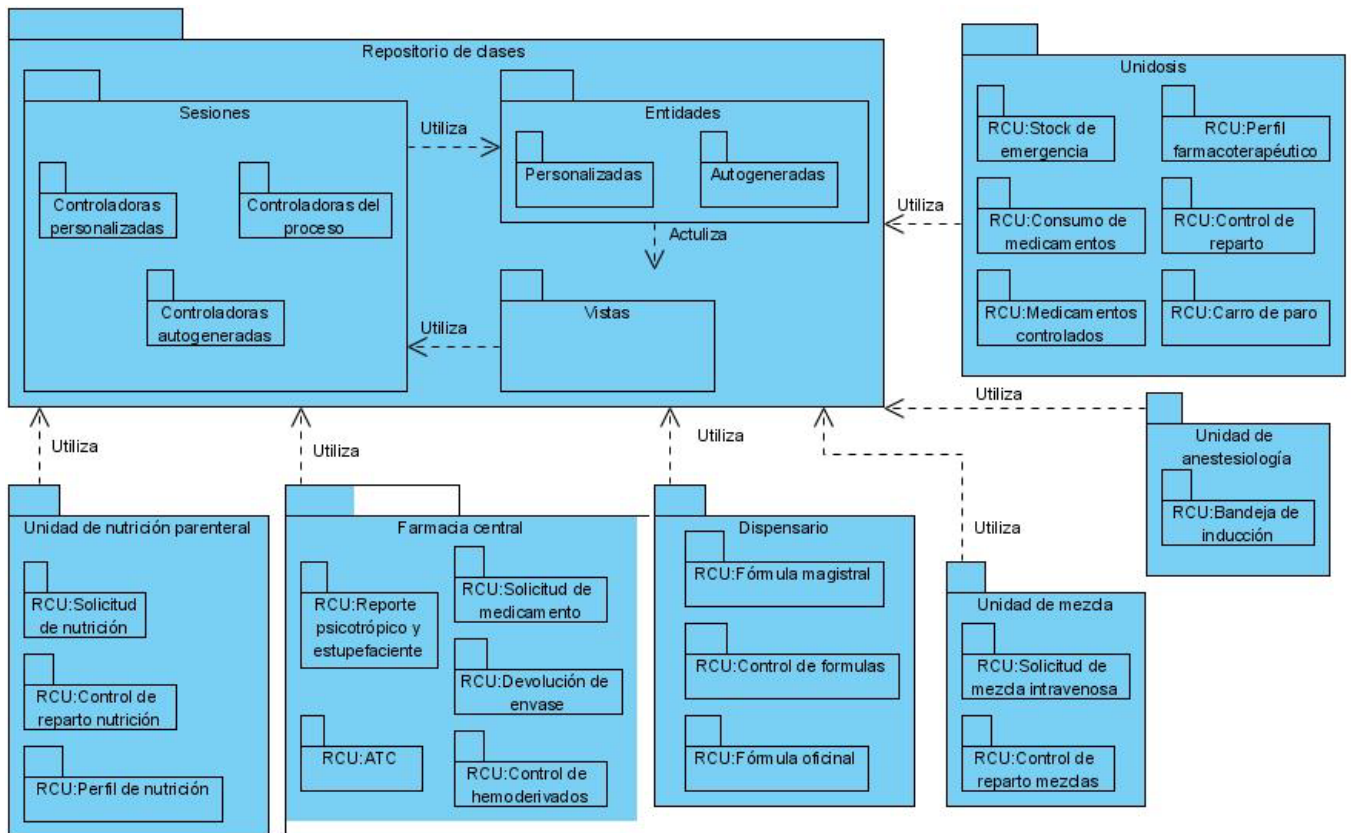


Figura 3.1 Modelo de diseño.

El paquete *Repositorio de clases*, contiene las clases del diseño de acuerdo a la tecnología que serán usadas en la implementación del módulo. Este engloba dos paquetes: *Entidades* y *Sesiones*. En el primero se encuentran, en el paquete *Autogeneradas*, todas las entidades que se autogeneran desde la base de datos utilizando el ORM Hibernate y en el paquete *Personalizadas*, la especialización realizadas de las clases autogeneradas, ya sea utilizando las relaciones de composición o herencia. El segundo incluye, el paquete *Controladoras autogeneradas*, en donde se encuentran las clases controladoras generadas (utilizando también la herramienta ORM), el paquete *Controladoras personalizadas*, con las especializaciones de las autogeneradas mediante la relación de herencia y el paquete *Controladoras del proceso*, en donde se encuentran todas las clases controladoras de cada uno de los procesos automatizados.

Además está definido un paquete por cada uno de los procesos del módulo. Este contiene un conjunto de paquetes en dependencia de las agrupaciones de los casos de usos que se hayan realizado en el proceso.

3.5 Diagramas de clases del diseño

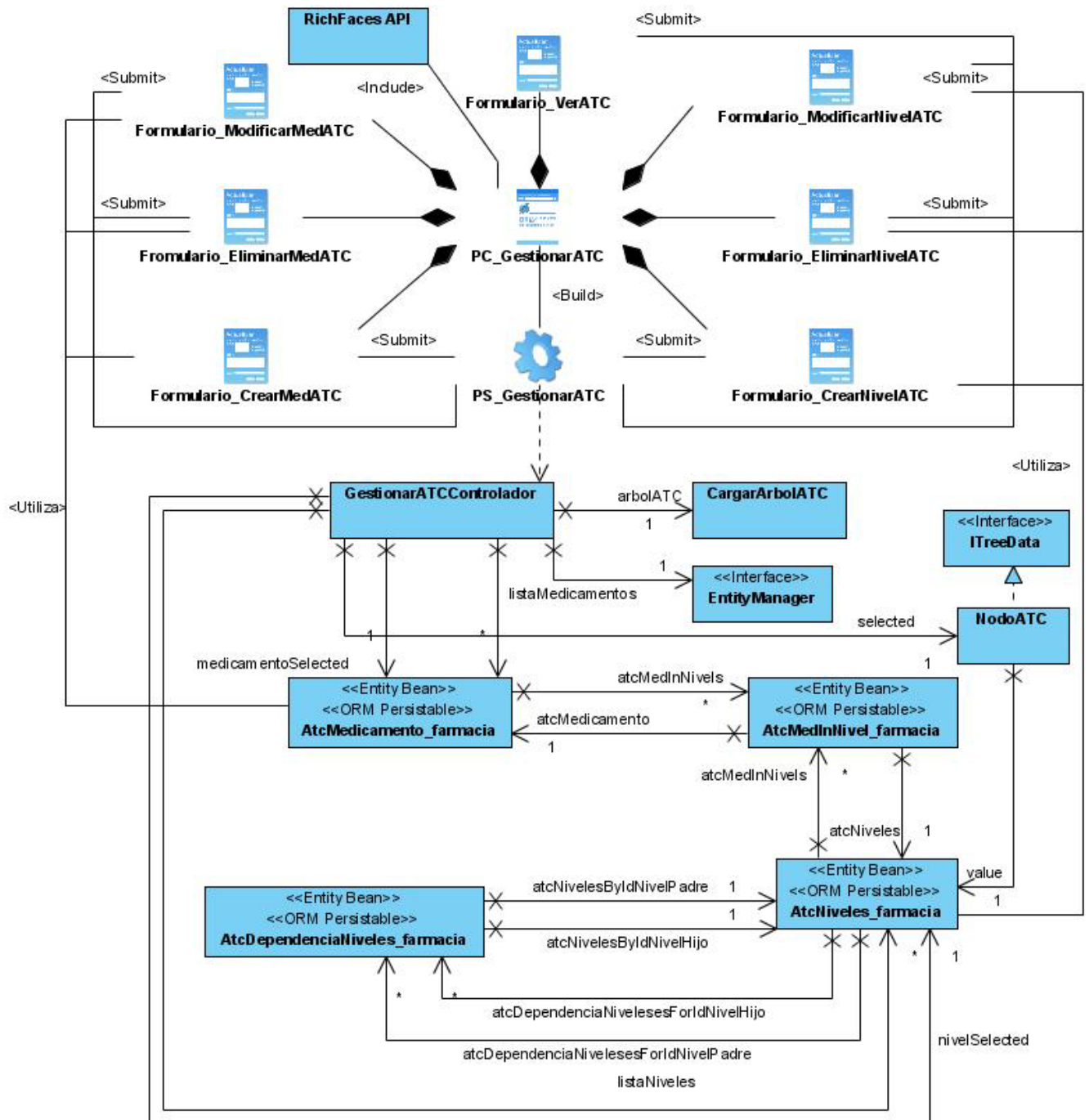


Figura 3.2 Diagrama de clases del diseño ATC.

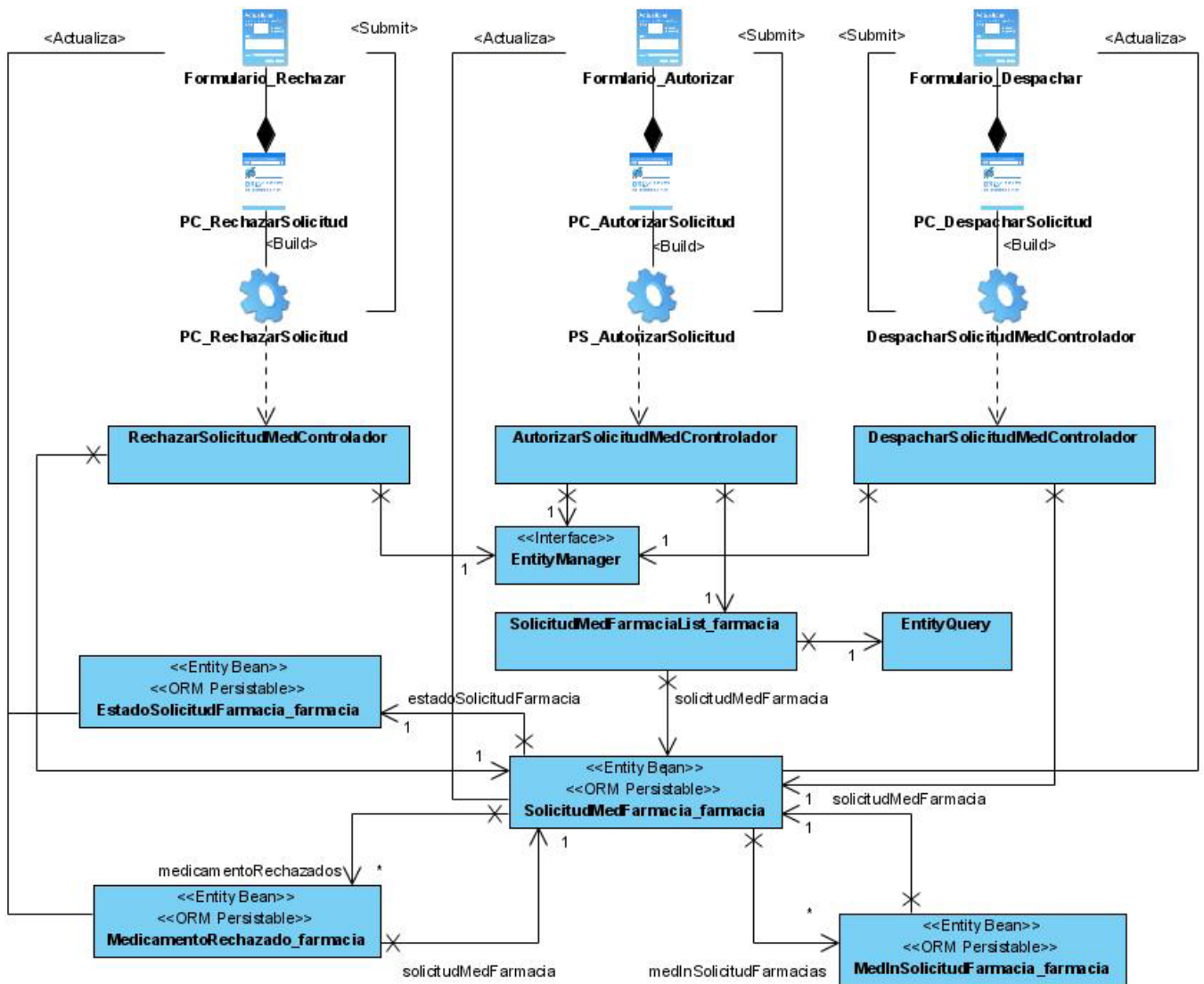


Figura 3.3 Diagrama de clases del diseño “Solicitud de medicamentos”.

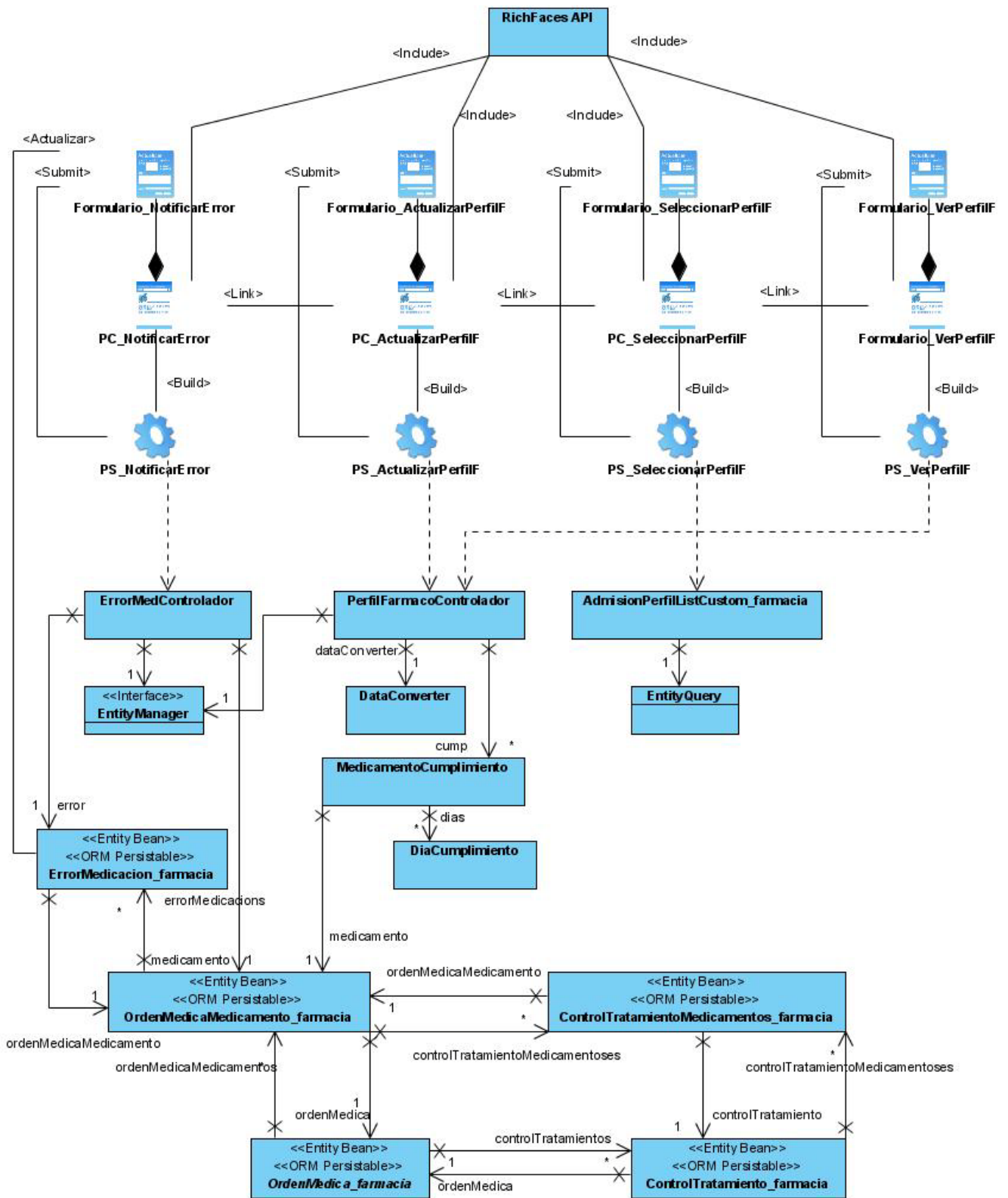


Figura 3.4 Diagrama de clases del diseño "Perfil farmacoterapéutico".

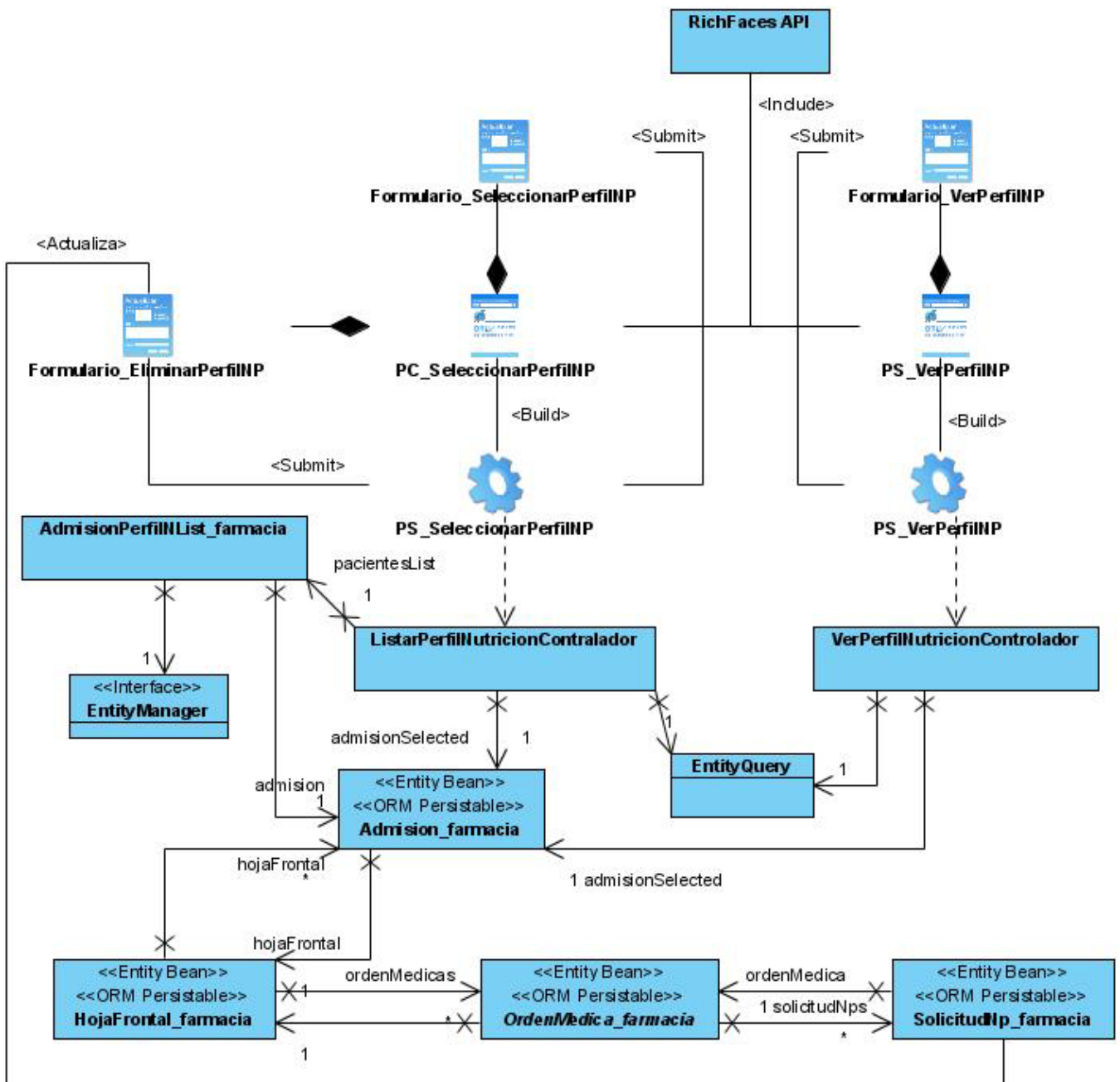


Figura 3.5 Diagrama de clases del diseño "Perfil de nutrición parenteral".

3.6 Diagramas de secuencias

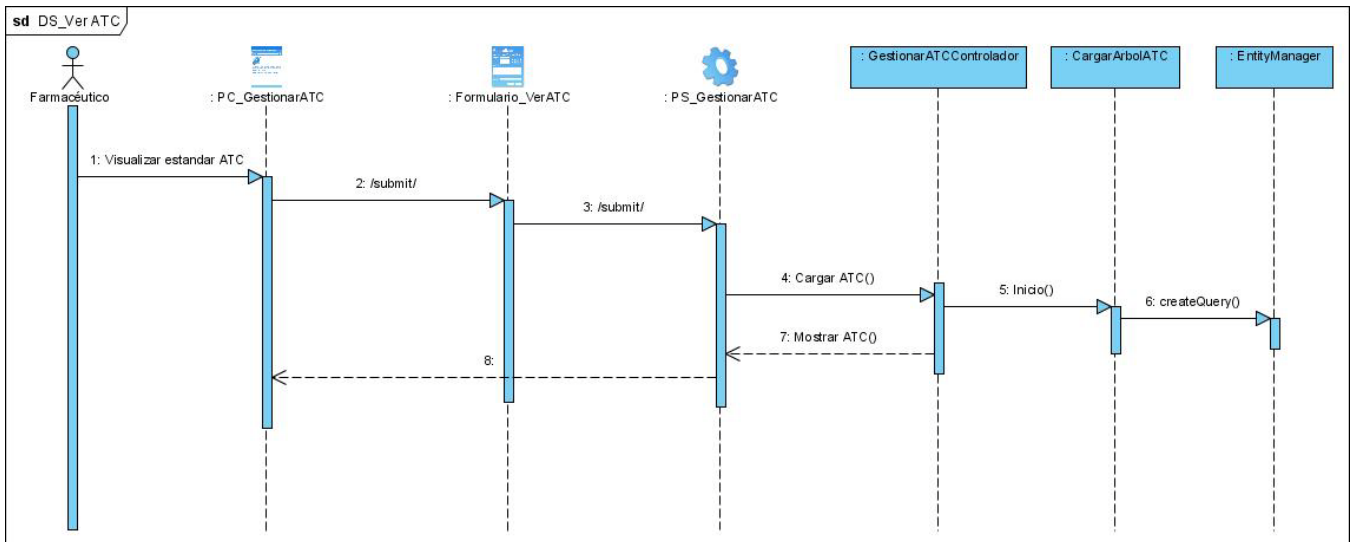


Figura 3.6 Diagrama de secuencia ver ATC

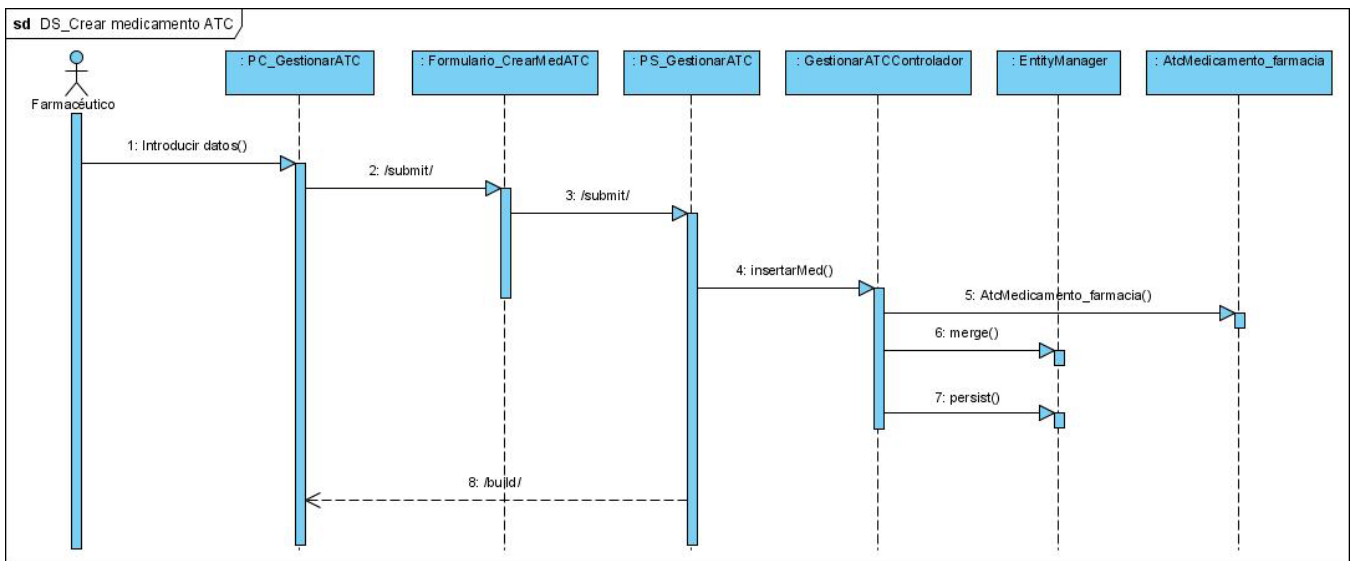


Figura 3.7 Diagrama de secuencia crear medicamento ATC

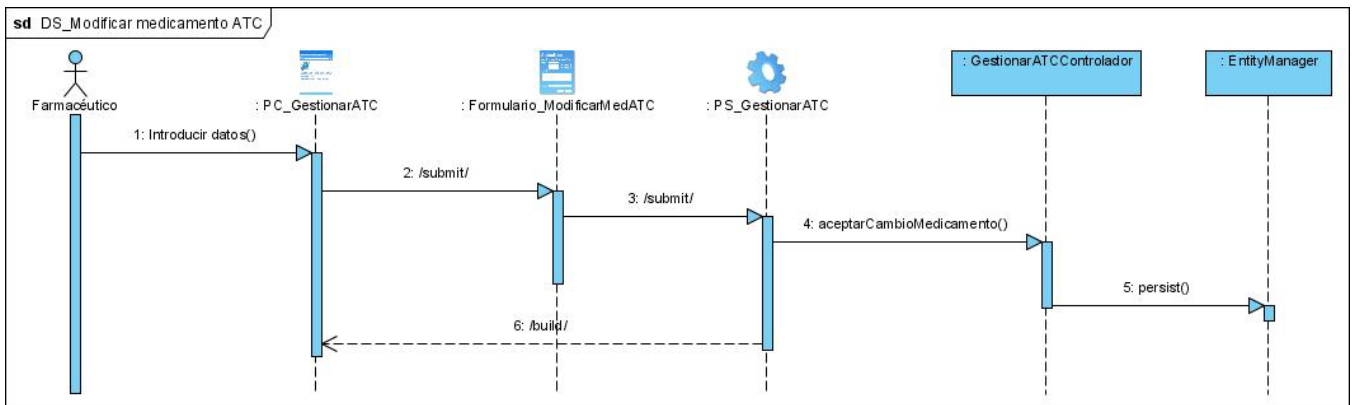


Figura 3.8 Diagrama de secuencia modificar medicamento ATC

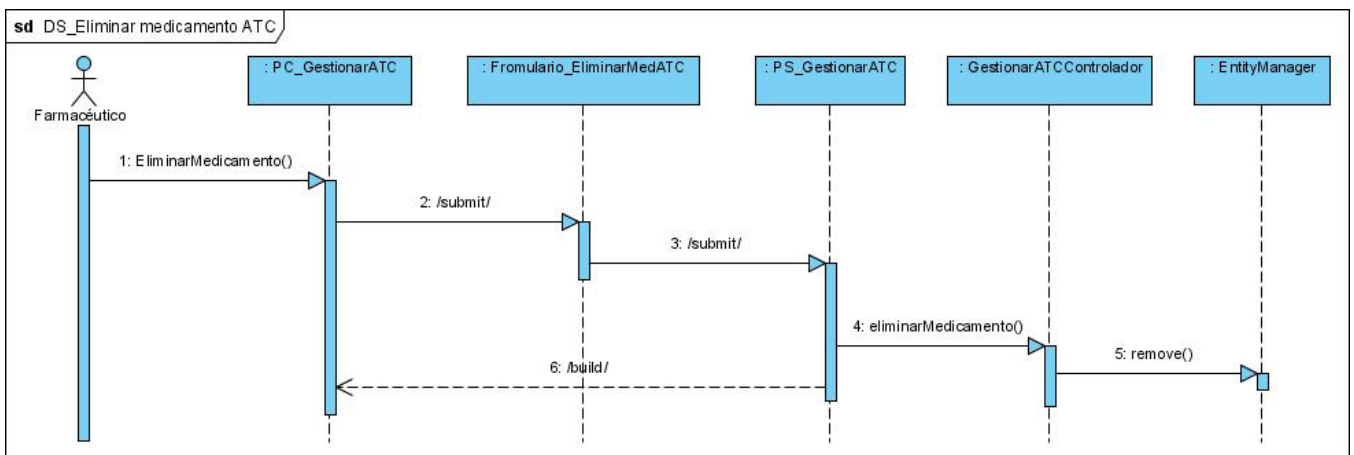


Figura 3.9 Diagrama de secuencia eliminar medicamento ATC

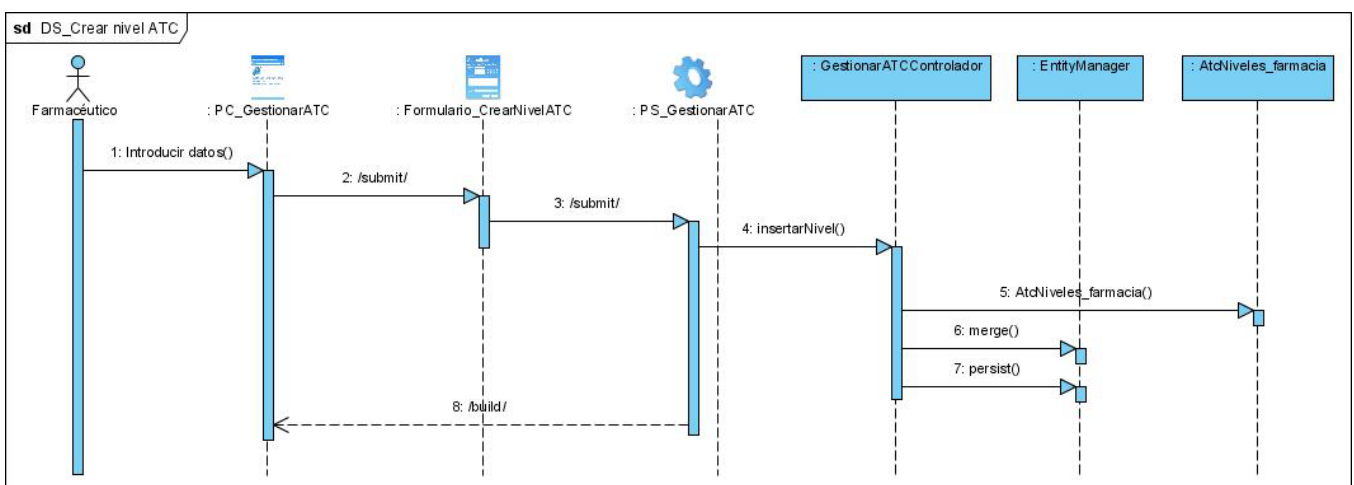


Figura 3.10 Diagrama de secuencia crear nivel ATC

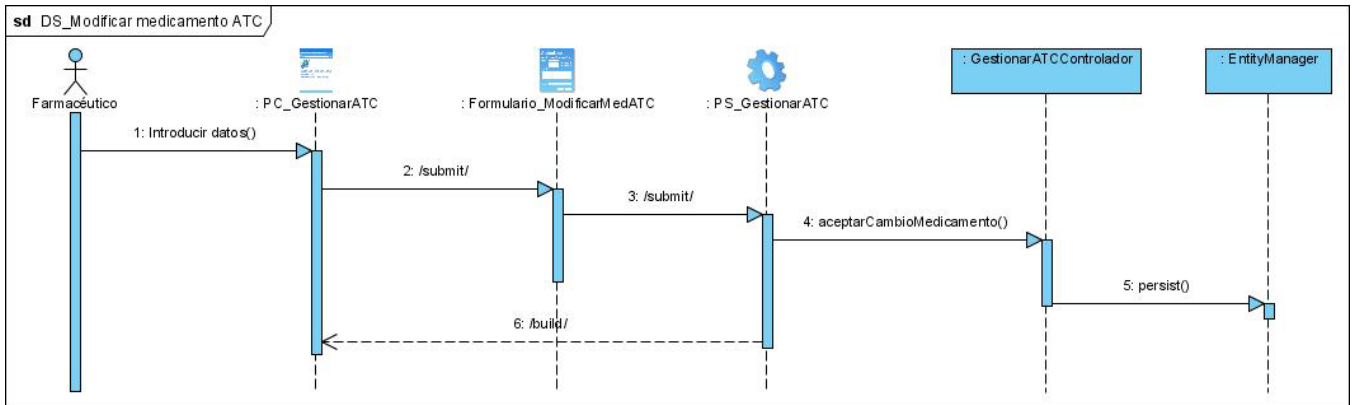


Figura 3.11 Diagrama de secuencia modificar nivel ATC

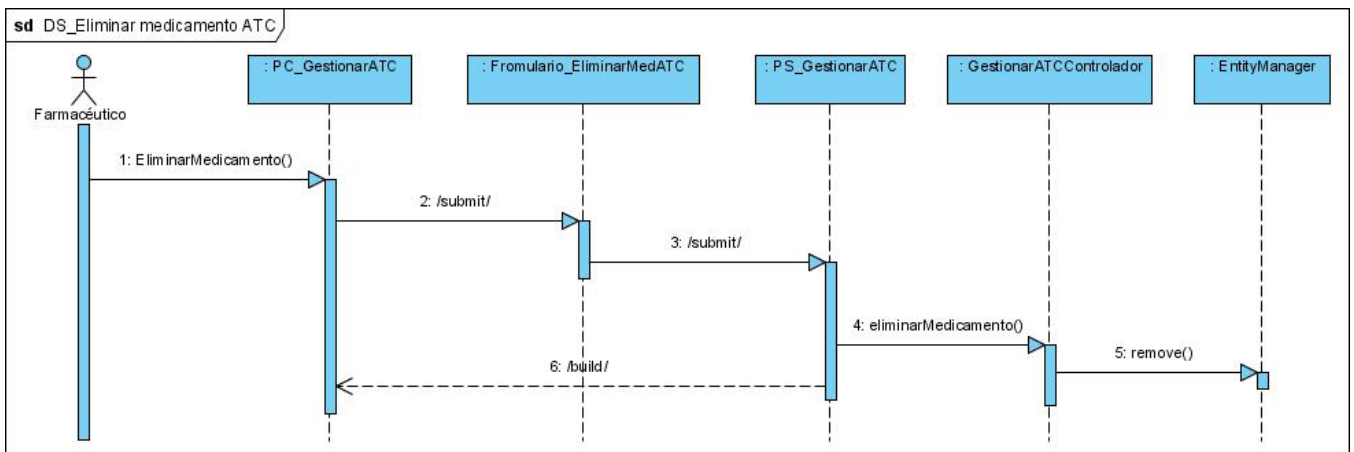


Figura 3.12 Diagrama de secuencia eliminar nivel ATC

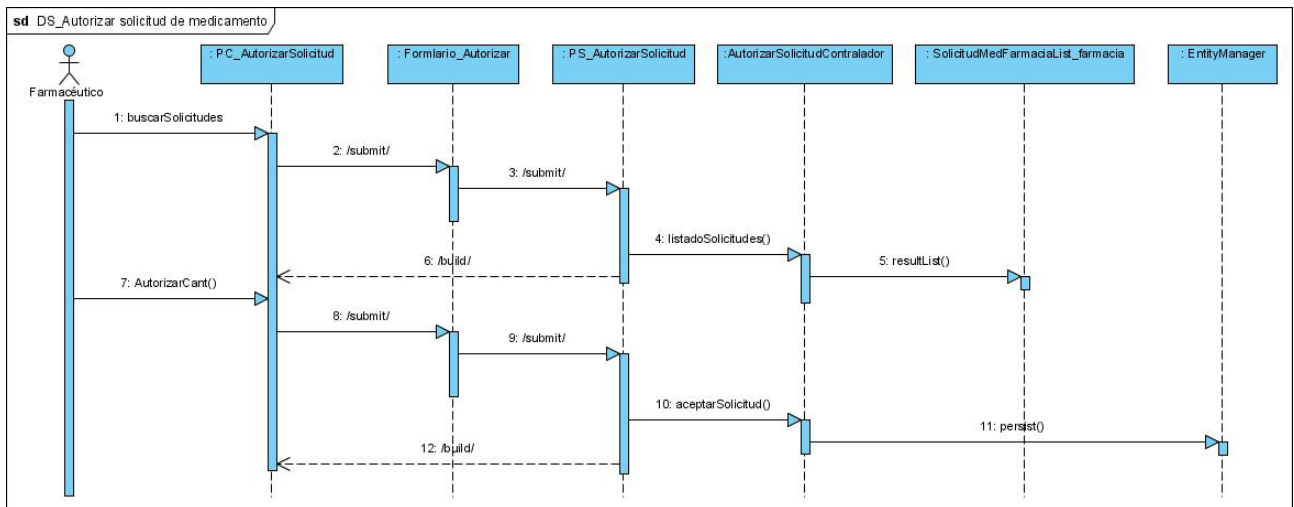


Figura 3.13 Diagrama de secuencia autorizar solicitud de medicamento

3.7 Descripción de clases y atributos

A continuación se describen algunas de las principales clases que han sido identificadas en el diseño para su futura implementación con el fin de obtener una mayor comprensión del sistema en desarrollo.

Nombre:	GestionarATCControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
entityManager	EntityManager	
arbolATC	CargarArbolATC	
selected	NodoATC	
node	TreeNode	
niveles	boolean	
medicamentos	boolean	
listaNiveles	List<AtcNiveles_farmacia>	
listaMedicamentos	List<AtcMedicamento_farmacia>	
nivelSelected	AtcNiveles_farmacia	
medicamentoSelected	AtcMedicamento_farmacia	
Para cada responsabilidad:		

Nombre:	Begin()
Descripción:	Comienza la conversación.
Nombre:	nodoSeleccionado(NodoATC selected, TreeNode node)
Descripción:	Asigna el nodo seleccionado y actualiza las listas de niveles y medicamentos contenidas dentro del nodo
Nombre:	List<AtcNiveles_farmacia> nivelesSeleccionados()
Descripción:	Devuelve la lista de niveles hijos del nivel seleccionado
Nombre:	List<AtcMedicamento_farmacia> medicamentosSeleccionados()
Descripción:	Devuelve la lista de medicamentos contenidos dentro del nivel seleccionado
Nombre:	nivelSelected(AtcNiveles_farmacia nivel)
Descripción:	Asigna el nivel seleccionado y lo introduce en el contexto
Nombre:	medicamentoSelected(AtcMedicamento_farmacia med)
Descripción:	Asigna el medicamento seleccionado y lo introduce en el contexto
Nombre:	aceptarCambioNivel()
Descripción:	Persiste los cambios realizados sobre la el nivel seleccionado
Nombre:	aceptarCambioMedicamento()
Descripción:	Persiste los cambios realizados sobre la el medicamento seleccionado

Tabla 14 Descripción textual clase GestionarATCControlador

Nombre:	RechazarSolicitudMedControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
entityManager	EntityManager	
listSolicitud	SolicitudMedFarmaciaList_farmacia	
solicitudSelected	SolicitudMedFarmacia_farmacia	
estado	EstadoSolicitudFarmacia_farmacia	
listMedSolicitud	List<MedInSolicitudFarmacia_farmacia>	
rechazo	MedicamentoRechazado_farmacia	
Para cada responsabilidad:		

Nombre:	List<SolicitudMedicamento_farmacia> solicitudesList()
Descripción:	Devuelve una lista de todas las solicitudes de medicamentos realizadas a la farmacia.
Nombre:	medicamentoSelected()
Descripción:	Medicamento seleccionado dentro de la solicitud.
Nombre:	solicitudSelected()
Descripción:	Solicitud seleccionado
Nombre:	aceptarCambiosSolicitud()
Descripción:	Persisten los cambios realizados a una solicitud.
Nombre:	rechazar()
Descripción:	Persiste en las solicitudes rechazadas.

Tabla 15 Descripción textual clase RechazarSolicitudMedControlador

Nombre:	AutorizarSolicitudMedControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
entityManager	EntityManager	
listSolicitudAutorizadas	SolicitudMedFarmaciaList_farmacia	
solicitudSelected	SolicitudMedFarmacia_farmacia	
estado	EstadoSolicitudFarmacia_farmacia	
listMedSolicitud	List<MedInSolicitudFarmacia_farmacia>	
Para cada responsabilidad:		
Nombre:	List<SolicitudMedicamento_farmacia> solicitudesList()	
Descripción:	Devuelve una lista de todas las solicitudes de medicamentos realizadas a la farmacia.	
Nombre:	medicamentoSelected()	
Descripción:	Medicamento seleccionado dentro de la solicitud.	
Nombre:	solicitudSelected()	
Descripción:	Solicitud seleccionado	
Nombre:	aceptarCambiosSolicitud()	

Descripción:	Persisten los cambios realizados a una solicitud.
Nombre:	autorizar()
Descripción:	Persiste en las solicitudes autorizadas.

Tabla 16 Descripción textual clase RechazarSolicitudMedControlador

Nombre:	DespacharSolicitudMedControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
entityManager	EntityManager	
listSolicitudAutorizadas	SolicitudMedFarmaciaList_farmacia	
solicitudSelected	SolicitudMedFarmacia_farmacia	
estado	EstadoSolicitudFarmacia_farmacia	
listMedSolicitud	List<MedInSolicitudFarmacia_farmacia>	
autorizado	List<boolean>	
Para cada responsabilidad:		
Nombre:	despachar(SolicitudMedFarmacia_farmacia solicitud)	
Descripción:	Registra la solicitud como despachada	
Nombre:	registrarUser(String user, String pass)	
Descripción:	Registra funcionario al que se le entregan los medicamentos.	
Nombre:	rebajaCantidades()	
Descripción:	Registra las cantidades despachadas de cada medicamento	

Tabla 17 Descripción textual clase DespacharSolicitudMedControlador

Nombre:	ErrorMedControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
paciente	Admision_farmacia	
medicamento	OrdenMedicaMedicamento_farmacia	

error	ErrorMedicacion_farmacia
fecha	Date
idFamacia	int
idMedicamento	String
idPaciente	String
entityManager	EntityManager
Para cada responsabilidad:	
Nombre:	validadorId()
Descripción:	Valida el id del paciente.
Nombre:	crearErrorMed()
Descripción:	Persiste la notificación del error de medicación.

Tabla 18 Descripción textual clase ErrorMedControlador

Nombre:	PerfilFarmacoControlador	
Tipo de clase:	Controladora	
Atributo	Tipo	
idPaciente	String	
mes	String	
medicamento	String	
medSelected	String	
anno	String	
paciente	Admision_farmacia	
idFamacia	int	
cump	List<MedicamentoCumplimiento>	
meses	List<String>	
dataConverter	DataConverter	
entityManager	EntityManager	
listadoOrdenMed	OrdenMedicaMedicamentoCustomList_farmacia	

listadoSOS	OrdenMedicaMedicamentoSOSCustomList_farmacia
Para cada responsabilidad:	
Nombre:	validadorId()
Descripción:	Valida el id del paciente.
Nombre:	errorMedicacion(Integer id)
Descripción:	Dado un id retorna si tiene o no notificación de error de medicación.
Nombre:	cumplimiento()
Descripción:	Retorna la lista de cumplimiento por medicamento.
Nombre:	toStringList()
Descripción:	Retorna la lista de nombres de los medicamentos.
Nombre:	annosList()
Descripción:	Retorna la lista de años.
Nombre:	update(Boolean cond)
Descripción:	Actualiza el filtrado del cumplimiento en dependencia de la condición.
Nombre:	medicamentoParam(int id)
Descripción:	Actualiza el id del medicamento seleccionado.

Tabla 19 Descripción textual clase PerfilFarmacoControlador

Nombre:	AdmisionPerfilListCustom_farmacia	
Tipo de clase:	Controladora	
Atributo	Tipo	
admision	Admision_farmacia	
idFarmacia	int	
idPaciente	int	
Para cada responsabilidad:		
Nombre:	getResultList()	
Descripción:	Retorna la lista de pacientes que tienen perfil farmacoterapéutico para la farmacia con idFarmacia.	

Tabla 20 Descripción textual clase AdmisionPerfilListCustom_farmacia

Nombre:	ListarPerfilNutricionControlador
Tipo de clase:	Controladora
Atributo	Tipo
entityManager	EntityManager
pacientesList	AdmisionPerfilNList_farmacia
listServicios	ServicioList_farmacia
listaServicios	List<Servicio_farmacia>
listaNombreServicio	List<String>
listaCamas	List<Cama_farmacia>
listaNombreCamas	List<String>
mostrar	boolean
admisionSelected	Admision_farmacia
Para cada responsabilidad:	
Nombre:	begin()
Descripción:	Inicia la conversación. Busca todos los servicios existentes en el hospital
Nombre:	selectedServicio()
Descripción:	Construye una lista con todas las camas existente en el servicio seleccionado
Nombre:	buscar()
Descripción:	Boolean que devuelve true si la búsqueda arrojo algún valor
Nombre:	selectedOne(Admision_farmacia selected)
Descripción:	Registra la admisión del paciente al cual se selecciono el perfil
Nombre:	eliminar()
Descripción:	Elimina el perfil de nutrición selccionado

Tabla 21 Descripción textual clase ListarPerfilNutricionControlador

Nombre:	VerPerfilNutricionControlador
----------------	-------------------------------

Tipo de clase:	Controladora	
Atributo	Tipo	
entityManager	EntityManager	
listaSolicitudNP	SolicitudNPListCustom_farmacia	
admisionSelected	Admision_farmacia	
idAdmision	int	
listComponentesReq	List<SolicitudNpCompNp_farmacia>	
listComponentes	List<SolicitudNpCompNp_farmacia>	
Para cada responsabilidad:		
Nombre:	boolean validatorId()	
Descripción:	Valida que el id que llega por parámetro sea un id valido.	
Nombre:	String actualizarListas(SolicitudNp_farmacia solicitudSelected)	
Descripción:	Llena la lista de componentes y de requerimiento de la nutrición.	
Nombre:	requerimientosC(SolicitudNp_farmacia solicitudSelected)	
Descripción:	Realiza una consulta a la base de datos para buscar los requerimientos calóricos de la nutrición	
Nombre:	componentes(SolicitudNp_farmacia solicitudSelected)	
Descripción:	Realiza una consulta a la base de datos para buscar los componentes de la nutrición	
Nombre:	List<ControlTratamientoNp_farmacia> controlTratamiento(SolicitudNp_farmacia solicitudSelected)	
Descripción:	Devuelve le control de la aplicación del tratamiento referente a la nutrición	

Tabla 22 Descripción textual clase VerPerfilNutricionControlador

Conclusiones

En este capítulo se realizó la modelación en términos del diseño de los casos de uso del sistema por escenarios, como parte de la solución propuesta. Se identificaron las clases interfaces, de control y de entidad de este flujo de trabajo. Mostrando como resultado final todos los diagramas de clases del diseño y diagramas de secuencia.

CAPÍTULO 4: IMPLEMENTACIÓN

Este capítulo contiene las principales características de la implementación del Sistema, además de los principales componentes y sus relaciones, mostrados a través del Diagrama de Componentes. También se exponen la estrategia de seguridad, los estándares de codificación así como los tratamientos de errores aplicados.

4.1 Modelo de datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. O sea un modelo de datos es la estructura o representación física de las tablas de la base de datos (figura 4.1).

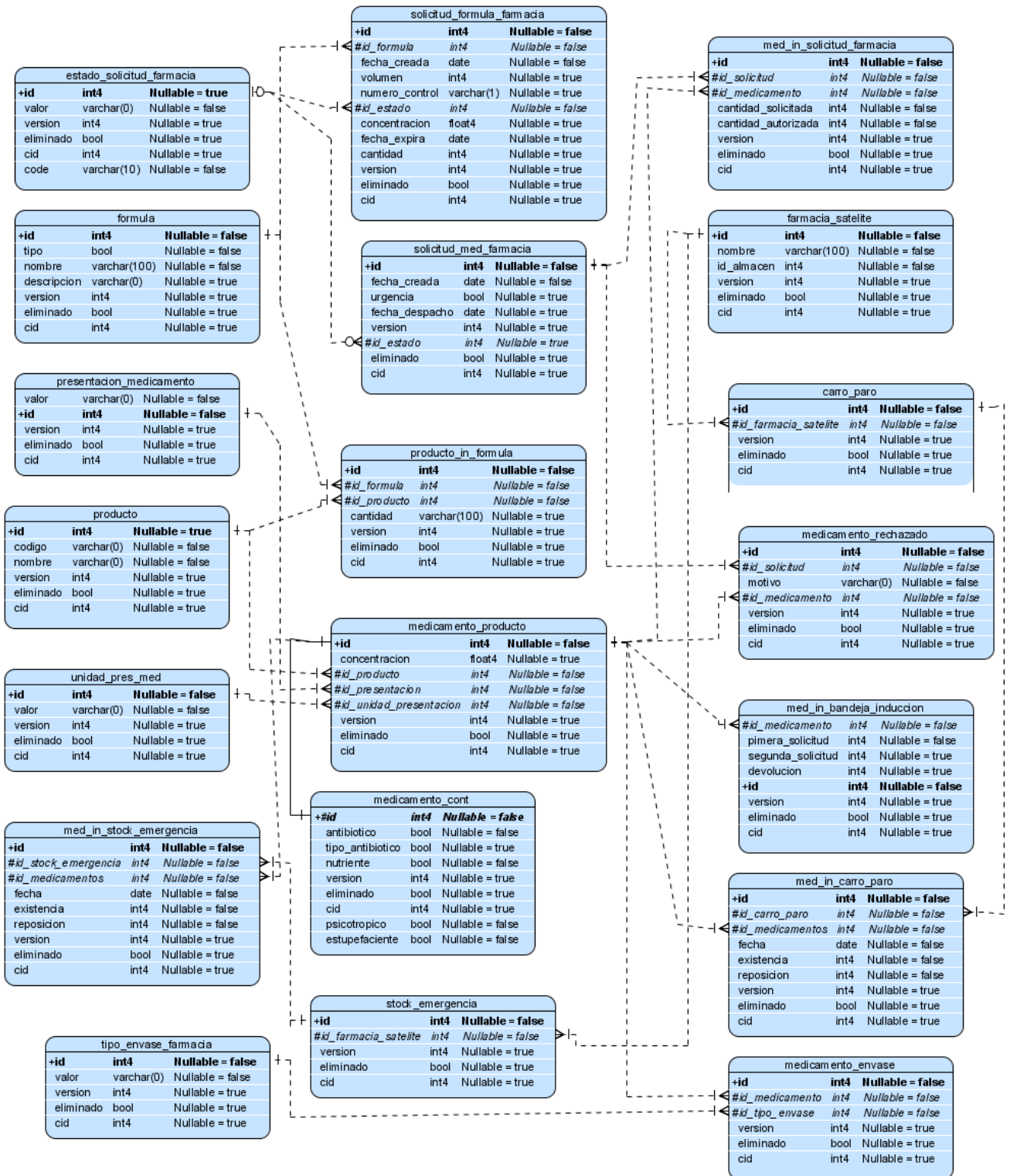


Figura 4.1. Modelo de datos.

4.1.1 Descripción de las tablas

Nombre	Documentación
 solicitud_formula_farmacia	Almacena las solicitudes de las fórmulas magistrales y oficinales, realizadas a la farmacia.
 med_in_solicitud_farmacia	Almacena los medicamentos incluidos en las solicitudes de medicamentos a la farmacia.
 estado_solicitud_farmacia	Almacena todos los estados que puede presentar una solicitud realizada a la farmacia.
 formula	Almacena todas las fórmulas, oficinales o magistrales, que pueden ser solicitadas a la farmacia.
 farmacia_satelite	Almacena las unidosis existentes en el hospital.
 solicitud_med_farmacia	Almacena todas las solicitudes de medicamentos, realizadas a la farmacia.
 presentacion_medicamento	Almacena todas las presentaciones en las cuales se pueden encontrar los fármacos.
 carro_paro	Almacena el estado de los carros de paro de cada unidosis.
 producto_in_formula	Almacena los productos contenidos dentro de las formulas.
 producto	Almacena todos los productos existentes en los almacenes
 medicamento_rechazado	Almacena los medicamentos que no van a ser dispensados por la farmacia por un motivo determinado.
 medicamento_producto	Almacena todos los medicamentos existentes en los almacenes de farmacia.
 unidad_pres_med	Almacena las unidades en las que se expresan cada uno de las presentaciones de los fármacos.
 med_in_bandeja_induccion	Almacena los medicamentos indicados en las bandejas de inducción.
 medicamento_cont	Almacena todos los medicamentos controlados por la farmacia.
 med_in_stock_emergencia	Almacena los medicamentos existentes en los stocks de emergencia de las unidosis existentes.
 med_in_carro_paro	Almacena los medicamentos existentes en los carros de paro de las unidosis existentes.
 stock_emergencia	Almacena el estado de los stocks de emergencia de cada unidosis.
 tipo_envase_farmacia	Almacena los tipos de envase de los medicamentos que requieren su devolución.
 medicamento_envase	Almacena los medicamentos, que requieren devolución de sus envases

Tabla 23 Tablas del modelo de datos.

Campos comunes

Nombre	Tipo de dato	PK/FK	Nulo	Documentación
id	int4	PK	No	Identificador
version	int4		Si	Campo para controlar la concurrencia optimista
eliminado	Bool		No	Campo para eliminación lógica
cid	int4		Si	Campo para tracking de los cambios en la bitácora.

Tabla 24 Campos comunes a todas las tablas



solicitud_formula_farmacia

Nombre	Valor
Clase mapeada	Solicitud_formula_farmacia
Esquema	publico
Nombre llave primaria	solicitud_formula_pkey

Tabla 25 Tabla solicitud_formula_farmacia.

Campos

Nombre	Tipo de dato	Nulo	Documentación
fecha_creada	date(0)	No	Fecha de creada
volumen	int4(0)	Si	Volumen total del producto
numero_control	varchar(100)	Si	Número de control del producto
concentracion	float4(0)	Si	Concentración del producto
fecha_expira	date(0)	Si	Fecha en la que expira la fórmula
cantidad	int4(0)	Si	Cantidad de la solicitud

Tabla 26 Campos de la tabla solicitud_formula_farmacia.



med_in_solicitud_farmacia

Nombre	Valor
Clase mapeada	Med_in_solicitud_farmacia
Esquema	publico
Nombre llave primaria	med_in_solicitud_farmacia_pkey

Tabla 27 Tabla med_in_solicitud_farmacia.

Campos

Nombre	Tipo de dato	Nulo	Documentación
cantidad_solicitada	int4(0)	No	Cantidad solicitada del medicamento.
cantidad_autorizada	int4(0)	No	Cantidad autorizada del medicamento.

Tabla 28 campos de la tabla med_in_solicitud_farmacia.



estado_solicitud_farmacia

Nombre	Valor
Clase mapeada	Estado_solicitud_farmacia
Esquema	Nomencladores
Nombre llave primaria	estado_soli_farmacia_pkey

Tabla 29 Tabla estado_solicitud_formula.

Campos

Nombre	Tipo de dato	Nulo	Documentación
valor	varchar(0)	No	Tipo del estado
code	varchar(10)	No	Internacionalización.

Tabla 30 Campos de la tabla estado_solicitud_formula.



formula

Nombre	Valor
Clase mapeada	Formula
Esquema	farmacia
Nombre llave primaria	formula_pkey

Tabla 31 Tabla formula.

Campos

Nombre	Tipo de dato	Nulo	Documentación
tipo	bool(0)	No	Verdadero, para Magistral y falso para Oficial
nombre	varchar(100)	No	Nombre de la fórmula
descripcion	varchar(0)	Si	Descripción de la fórmula

Tabla 32 Campos de la tabla formula.

 **farmacia_satelite**


Nombre	Valor
Clase mapeada	 Farmacia_satelite
Esquema	farmacia
Nombre llave primaria	farmacia_satelite_pkey

Tabla 33 Tabla farmacia_satelite.

Campos

Nombre	Tipo de dato	Nulo	Documentación
nombre	varchar(100)	No	Nombre de la Unidosis

Tabla 34 Campos de la tabla farmacia_satelite.

 **solicitud_med_farmacia**


Nombre	Valor
Clase mapeada	 Solicitud_med_farmacia
Esquema	publico
Nombre llave primaria	solicitud_med_farmacia_pkey

Tabla 35 Tabla solicitud_med_farmacia.

Campos

Nombre	Tipo de dato	Nulo	Documentación
fecha_creada	date(0)	No	Fecha de creada la solicitud.
urgencia	bool(0)	Si	Clasificación (Urgencia o no)
fecha_despacho	date(0)	Si	Fecha del despacho de la solicitud

Tabla 36 Campos de la tabla solicitud_med_farmacia.

 **presentacion_medicamento**

Nombre	Valor
Esquema	nomencladores
Nombre llave primaria	unidadMedic_pkey

Tabla 37 Tabla presentacion_medicamento.

Campos

Nombre	Tipo de dato	Nulo	Documentación
valor	varchar(0)	No	Tipo de presentación del medicamento.

Tabla 38 Campos de la tabla presentacion_medicamento.



producto_in_formula

Nombre	Valor
Clase mapeada	Producto_in_formula
Esquema	farmacia
Nombre llave primaria	producto_in_formula_pkey

Tabla 39 Tabla producto_in_formula.

Campos

Nombre	Tipo de dato	Nulo	Documentación
cantidad	varchar(100)	Si	Cantidad necesaria del producto para la elaboración de la fórmula.

Tabla 40 Campos de la tabla producto_in_formula.



producto

Nombre	Valor
Clase mapeada	Producto
Esquema	publico
Nombre llave primaria	producto_pkey

Tabla 41 Tabla producto.

Campos

Nombre	Tipo de dato	Nulo	Documentación
codigo	varchar(100)	No	Código del producto.
nombre	varchar(100)	No	Nombre del producto.

Tabla 42 Campos de la tabla producto.



medicamento_rechazado

Nombre	Valor
Clase mapeada	Medicamento_rechazado
Esquema	farmacia
Nombre llave primaria	medicamento_rechazado_pkey

Tabla 43 Tabla medicamento_rechazado.

Campos

Nombre	Tipo de dato	Nulo	Documentación
motivo	varchar(100)	No	Almacena el motivo por el cual se rechaza el medicamento.

Tabla 44 Campo de la tabla medicamento_rechazado.



medicamento_producto

Nombre	Valor
Clase mapeada	Medicamento_producto
Esquema	publico
Nombre llave primaria	medicamento_producto_pkey

Tabla 45 Tabla medicamento_producto.

Campos

Nombre	Tipo de dato	Nulo	Documentación
concentracion	float4(0)	Si	Concentración del medicamento.

Tabla 46 Campo de la tabla medicamento_producto.



unidad_pres_med

Nombre	Valor
Clase mapeada	Unidad_pres_med
Esquema	nomencldores
Nombre llave primaria	unidad_pres_med_pkey

Tabla 47 Tabla unidad_pres_med.

Campos

Nombre	Tipo de dato	Nulo	Documentación
valor	varchar(0)	No	Unidad de medida.

Tabla 48 Campo de la tabla unidad_pres_med.



med_in_bandeja_induccion

Nombre	Valor
Clase mapeada	Med_in_bandeja_induccion
Esquema	publico
Nombre llave primaria	med_in_bandeja_induccion_pkey

Tabla 49 Tabla med_in_bandeja_induccion.

Campos

Nombre	Tipo de dato	Nulo	Documentación
primera_solicitud	int4(0)	No	Cantidad solicitada en la primera solicitud.
segunda_solicitud	int4(0)	Si	Cantidad solicitada en la primera solicitud.
devolucion	int4(0)	Si	Cantidad devuelta.

Tabla 50 Campos de la tabla med_in_bandeja_induccion.



medicamento_cont

Nombre	Valor
Clase mapeada	Medicamento_cont
Esquema	publico
Nombre llave primaria	medicamento_cont_pkey

Tabla 51 Tabla medicamento_cont.

Campos

Nombre	Tipo de dato	Nulo	Documentación
antibiotico	bool(0)	No	Almacena si el medicamento es un antibiótico o no.
tipo_antibiotico	bool(0)	Si	Almacena si el antibiótico es semicontrolado o controlado.
nutriente	bool(0)	No	Almacena si el medicamento es un nutriente.
psicotropico	bool(0)	No	Almacena si el medicamento es un psicotrópico.

estupefaciente	bool(0)	No	Almacena si el medicamento es un estupefaciente.
----------------	---------	----	--

Tabla 52 Campos de la tabla medicamento_cont.



med_in_stock_emergencia

Nombre	Valor
Clase mapeada	Med_in_stock_emergencia
Esquema	farmacia
Nombre llave primaria	med_in_stock_emergencia_pkey01

Tabla 53 Tabla med_in_stock_emergencia.

Campos

Nombre	Tipo de dato	Nulo	Documentación
fecha	date(0)	No	Fecha de la reposición de los medicamentos.
existencia	int4(0)	No	Almacena la existencia antes de la reposición.
reposicion	int4(0)	No	Almacena la cantidad que se repone.

Tabla 54 Campos de la tabla med_in_stock_emergencia.



med_in_carro_paro

Nombre	Valor
Clase mapeada	Med_in_carro_paro
Esquema	farmacia
Nombre llave primaria	med_in_carro_paro_medicamentos_in_carro_paro_pkey01

Tabla 55 Tabla med_in_carro_paro.

Campos

Nombre	Tipo de dato	Nulo	Documentación
fecha	date(0)	No	Fecha de la reposición de los medicamentos.
existencia	int4(0)	No	Almacena la existencia antes de la reposición.
reposicion	int4(0)	No	Almacena la cantidad que se repone.

Tabla 56 Campos de la tabla med_in_carro_paro.



tipo_envase_farmacia

Nombre	Valor
Clase mapeada	Tipo_envase_farmacia
Esquema	nomencladores
Nombre llave primaria	tipo_envase_farmacia_pkey

Tabla 57 Tabla tipo_envase_farmacia.

Campos

Nombre	Tipo de dato	Nulo	Documentación
valor	varchar(0)	No	Almacena el nombre del tipo de envase (blister, ampolla).

Tabla 58 Campo de la tabla tipo_envase_farmacia.

4.2 Modelo de clases persistentes

El Modelo de clases persistentes no es más que el mapeo de las tablas de la base de datos lo cual permite manejar la lógica del negocio a partir de la lógica del Modelo de diseño. (Figura 4.2)

4.3 Modelo de despliegue

El Modelo de despliegue es utilizado para definir la estructuración, descripción y relaciones físicas del hardware concebido para el despliegue del sistema. (Figura 4.3)

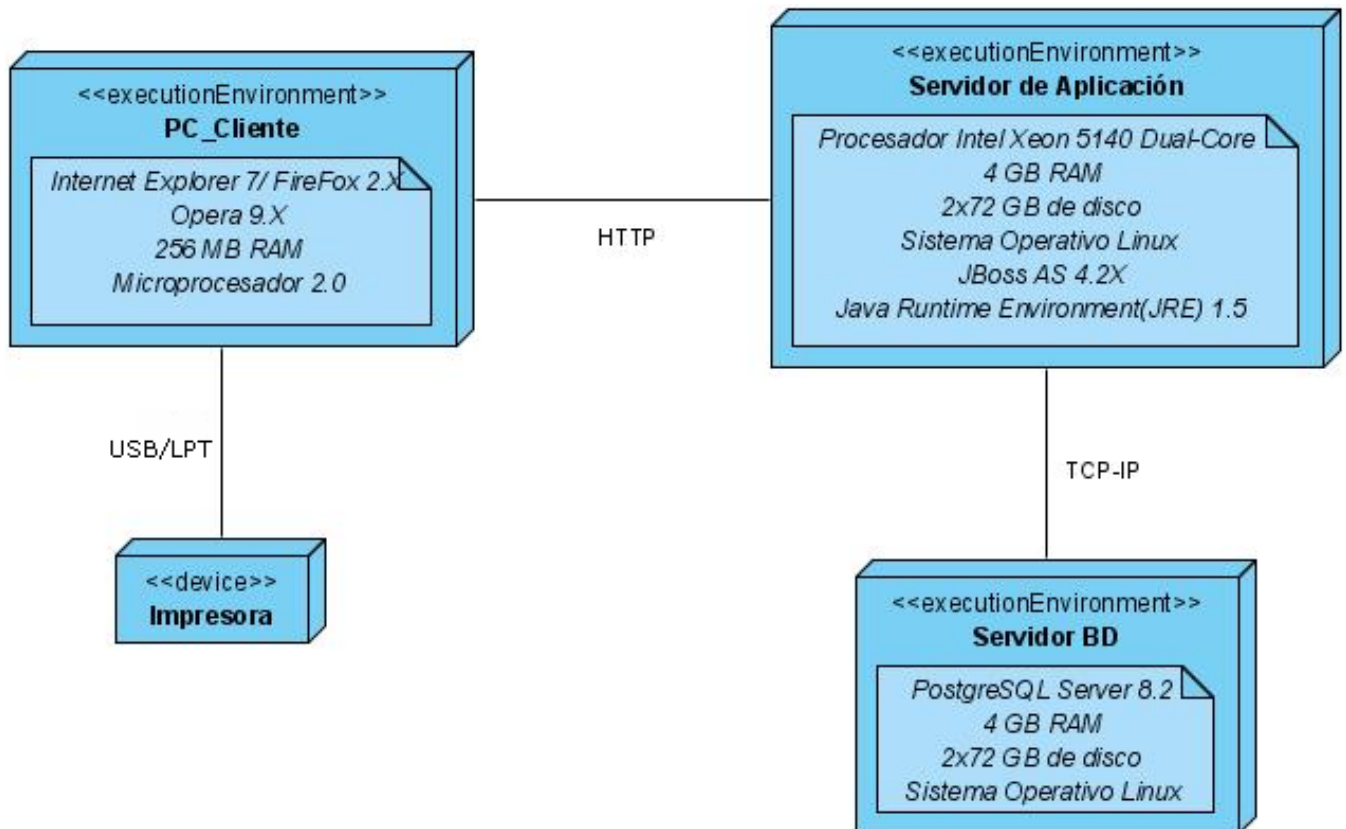


Figura 4.3 Modelo de despliegue.

4.4 Diagrama de componentes

Este diagrama describe la estructura de los componentes del sistema agrupados por paquetes lógicos.

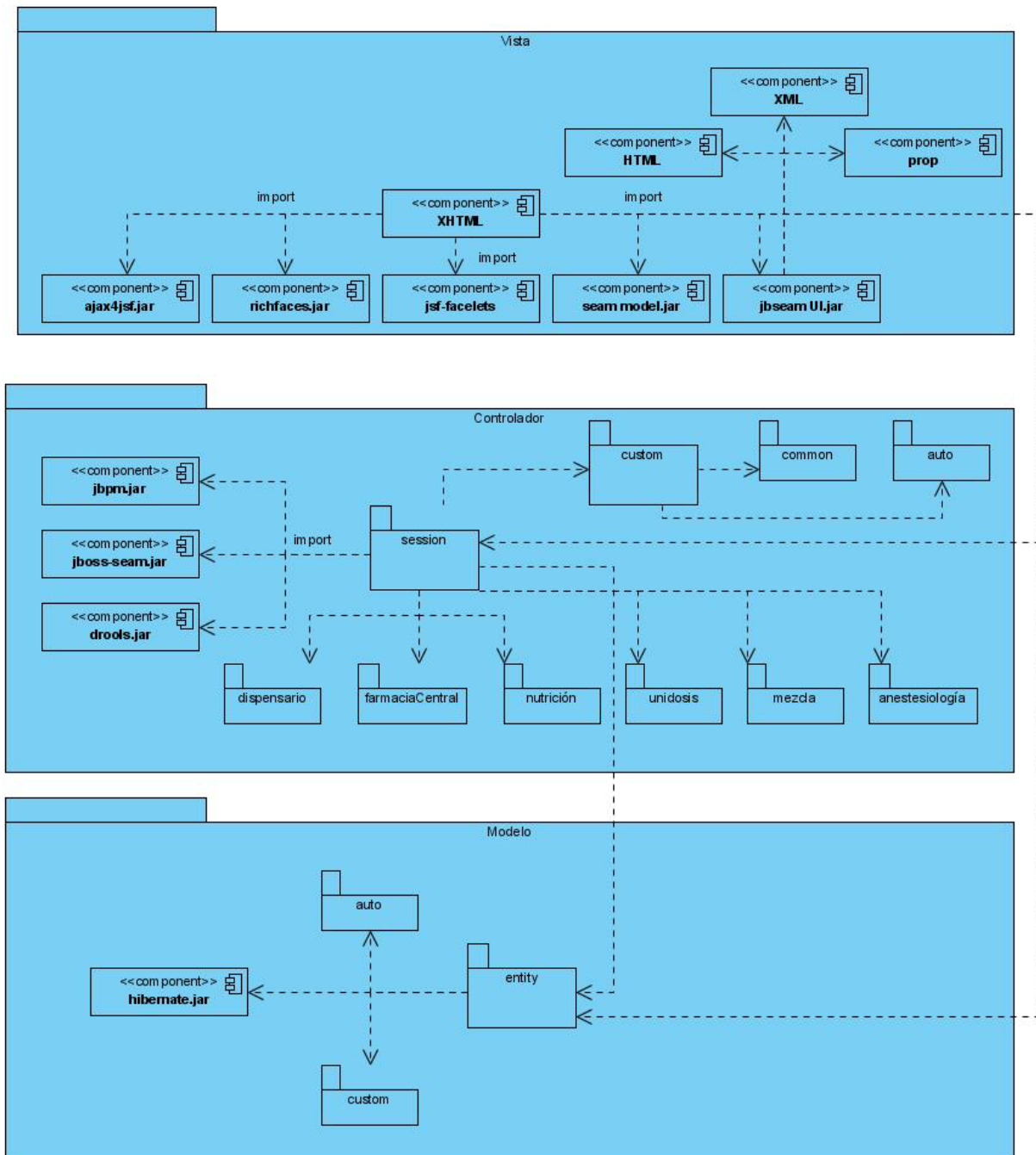


Figura 4.4 Diagrama de componentes.

4.5 Tratamiento de errores

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Las excepciones son el mecanismo recomendado para la propagación de errores que se produzcan durante la ejecución de las aplicaciones.

Cuando dicho un error ocurre dentro de un método Java, automáticamente se crea un objeto 'Excepcion' el cual es tratado en el sistema de ejecución. Este objeto contiene información sobre la excepción, incluyendo su tipo y el estado del programa cuando ocurrió el error.

En el HIS se propone el tratamiento de excepciones principalmente en las regiones críticas de código, o sea en los fragmentos donde se manipulen los datos son insertados o modificados en la base de datos y además en las validaciones de los datos que son insertados por los usuarios en la interfaz de usuario.

El control de la navegación, en caso de ocurrir una excepción que implique una redirección, se maneja mediante los '.pages.xml', los mismos encargan de capturar globalmente las excepciones y ejecutar las instrucciones determinadas. Para el control de las demás excepciones es utilizado el componente FacesMessages del framework Seam el cual se encarga de mostrar los mensajes que se manejan a través del objeto facesMessages inyectado en las clases controladoras. El mismo trata los mensajes por tipo(error, alerta y notificación).

4.6 Seguridad

La seguridad es un tema de gran importancia para cualquier Sistema de Información y toma mayor relevancia cuando se gestiona información relacionada con la salud de las personas. Para garantizar la seguridad del Módulo Farmacia toda la autorización está basada en reglas que restringen el acceso a directorios, páginas, controles, opciones del menú y servicios del negocio. Ninguna de estas reglas del negocio están contenidas en el código de la aplicación, sino en ficheros (no compilados) dotando a la aplicación de mayor dinamismo al no requerir una recompilación en caso que cambie alguna. Esto puede llevarse a cabo por la posibilidad de integración con el motor de reglas JBoss Rules, que brinda el Framework de Seguridad de JBoss Seam.

Se realiza además el control a nivel de usuarios y contraseñas, garantizando el acceso sólo a los niveles establecidos de acuerdo a la función que realiza cada usuario. Las contraseñas sólo pueden ser cambiadas por el propio usuario o por el administrador del sistema. También se validan todos los ids o llaves enviadas por URL a fin de asegurar que sean correctos.

Todas las actividades realizadas por los usuarios quedan registradas a cada momento en una especie de bitácora, almacenándose la fecha, la hora, el usuario y la actividad que realizó el mismo.

Conclusiones

En el capítulo se describió la estructura de tablas en la que se sustenta el modelo de datos del sistema, así como sus atributos y relaciones. Se estructuraron las clases del diseño en paquetes y subsistemas de implementación. En términos generales fue implementado el sistema en términos de componentes con el objetivo de dar solución a los requisitos especificados.

CONCLUSIONES GENERALES

Con el desarrollo del Módulo Farmacia del Sistema de Información Hospitalaria alas HIS se arribaron a las siguientes conclusiones:

- Los sistemas analizados a nivel mundial no satisfacen las necesidades que en materia de gestión de la información presentan las instituciones hospitalarias.
- Se evidenció la necesidad de automatizar los procesos presentes en el área de Farmacia de las instituciones hospitalarias, con el objetivo de mejorar la calidad de los servicios que en esta se brindan y lograr facilitar la gestión de la información.
- Se obtuvo un Sistema seguro, robusto y flexible, que requiere un mínimo de cambios para su adaptación a las necesidades de cualquier institución hospitalaria en particular.
- El diseño del sistema propuesto facilitó un mejor entendimiento a los diseñadores de base de datos e implementadores.
- Se implementaron los procesos: Unidad de nutrición parenteral, Unidad de anestesiología, Despachar medicamentos, Elaborar preparados y Farmacia satélite del módulo Farmacia del Sistema de Información Hospitalaria alas HIS, agilizando los mismos en función de elevar calidad de la atención al paciente.

De esta forma se cumplió con el objetivo y las tareas trazadas para la elaboración del trabajo de diploma, por lo que se obtuvo un sistema informático, que permite mejorar los procesos de gestión de la información en el área de Farmacia en las instituciones hospitalarias.

RECOMENDACIONES

Por la experiencia adquirida en el desarrollo del presente trabajo de diploma, se recomienda:

- Someter la aplicación a un proceso de pruebas de calidad para su estabilización y posterior instalación en las Instituciones hospitalarias que lo requieran
- Extender las funcionalidades para el control de los medicamentos mediante el manejo de códigos de barras.
- Incorporar funcionalidades que permitan brindar el servicio farmacéutico a los pacientes de consulta externa.
- Incluir sistema de ayuda en línea para los usuarios que interactúan con el sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. **González Núñez, José.** *La farmacia en la historia: la historia de la farmacia. Una aproximación desde la ciencia, el arte y la literatura.* Barcelona : s.n., 2006. 9788497511797.
2. **Alderete, Ethel.** *Salud y pueblos indígenas .* Quito - Ecuador : ABYA YALA, 2004. 9978223479.
3. **Sanchez Santos, Leonardo, et al.** *Introducción a la Medicina General Integral.* La Habana, Cuba : Ciencias Médicas, 2001.
4. **CNT.** Sistemas de Información S.A. [Online] CNT, 2009.
<http://www.cnt.com.co/pagina/index.asp?id=178>.
5. **Telvent.** Farmacia Hospitalaria. [Online] Televent, 2008.
<http://www.telvent.com/sites/telvent/es/soluciones/administracionespublicas/salud/soluciones/sistemasinformaciondepartamental/farmacia.html>.
6. **CNT.** Hospital Information System Edition. [Online] CNT PACIENTES, 2009.
<http://www.cnt.com.co/pagina/index.asp?id=178>.
7. **Paúl Céspedes, Jesús Armando.** Manuales SIGHO. [Online] 2007.
<http://sigho.saludtlax.gob.mx/wp-content/uploads/2007/08/farmacia-210.pdf>.
8. **Booch, Grady.** *Object-Oriented Analysis and Design.* s.l. : Second Edition, 1994.
9. **Allen, Dan.** *Seam in Action.* s.l. : Manning Publication, 2008.
10. **Javid Jamae, Peter Johnson.** *JBoss in Action.* s.l. : Manning Publications, enero 2009. 1933988029.
11. *UI Development with JaveServer Faces.*
12. **Red Hat.** *RichFaces developer Guide.* 2007.
13. —. *Ajax4jsf Developer Guide.* 2007.
14. **The postgresSQL Global Development Group.** *PostgreSQL 8.3.6 Documentation.* 2008.
15. **Cristian Bauer, Gavin King.** *Java Persistence with Hibernate.* 2005. 1-932394-88-5.
16. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño.* 2004.
17. **Pressman, Roger S.** *Ingeniería de software, un enfoque práctico .* 2005.
18. **Pérez Jiménez, Juan Diego, Durán Toro, Amador and Ruiz Cortes, Antonio.** *¿Por qué OMG ha elegido BPMN para modelar de Procesos de Negocios si ya existe UML?* s.l. : Dept.de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, ETS Ingeniería Informática, 2007.
19. **Stephen A. White, PhD y Miers, Derek.** *BPMN Modeling an References Guide. Understanding and Using BPMN.* 2008. 13:978-0977752720.

20. **Velázquez Carralero, Alejandro Mario.** *IH-SW-DR-091 Alas-His_Documento de Arquitectura del Sistema.* 2008.

BIBLIOGRAFÍA

- Alderete, Ethel.** *Salud y pueblos indígenas*. Quito - Ecuador : ABYA YALA, 2004. 9978223479.
- Allen, Dan.** *Seam in Action*. s.l. : Manning Publications, 2008.
- Booch, Grady.** *Object-Oriented Analysis and Design*. s.l. : Second Edition, 1994.
- C-DAC.** [En línea] C-DAC. <http://www.cdac.in/html/his/sushrut.asp>.
- CARE2X Integrated Healthcare Environment. [En línea] <http://www.care2x.org/>.
- CNT.** Sistemas de Información S.A. [Online] CNT, 2009.
<http://www.cnt.com.co/pagina/index.asp?id=178>.
- CNT.** Hospital Information System Edition. [Online] CNT PACIENTES, 2009.
<http://www.cnt.com.co/pagina/index.asp?id=178..>
- Cristian Bauer, Gavin King.** *Java Persistence with Hibernate*. 2005. ISBN 1-932394-88-5.
- Farley, Jim.** *Practical JBoss® Seam Projects*. 2007.
- González Núñez, José.** *La farmacia en la historia: la historia de la farmacia. Una aproximación desde la ciencia, el arte y la literatura*. Barcelona : s.n., 2006. 9788497511797.
- Hibernate. [En línea] Red Hat. <https://www.hibernate.org/>.
- Hibernate Reference Documentation*.
- Hospitales_InscripcionAdmicion_Conceptos*. 2008.
- Ivan Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software*. 2004.
- JavaHispano.** [En línea] <http://www.javahispano.org/>.
- Javid Jamae, Peter Johnson.** *JBoss in Action*. s.l. : Manning Publications, enero, 2009. ISBN: 1933988029.
- JBoss.** [En línea] Red Hat, 2009. <http://www.jboss.com/>.
- JBoss Community.** [En línea] <http://www.jboss.org/>.
- JBoss Community.** Drools Documentation Library. [En línea] JBoss Community.
<http://www.jboss.org/drools/documentation.html>.
- JBOSS.** *Seam - Contextual Components A Framework for Enterprise Java*.
- JBPM Documentation Library. [En línea] JBoss. http://www.jboss.org/jbossjbpm/jbpm_documentation/.

Larman, Craig. *UML y patrones. Introducción al análisis y diseño orientado a objetos.* 2004.

UI development with JavaServer Faces.

Michael Juntao Yuan, Thomas Heute. *JBoss Seam Simplicity and Power Beyond Java EE.* 2007. ISBN 0-13-134796-9.

OMG Object Management Group. [En línea] Object Management Group, Inc. <http://www.bpmn.org/>.

OMS Organización Mundial de la Salud. [En línea] OMS. <http://www.who.int/es/>.

Paúl Céspedes, Jesús Armando. Manuales SIGHO. [Online] 2007. <http://sigho.saludtlax.gob.mx/wp-content/uploads/2007/08/farmacia-210.pdf>.

Pérez Jiménez, Juan Diego, Durán Toro, Amador and Ruiz Cortes, Antonio. *¿Por qué OMG ha elegido BPMN para modelar de Procesos de Negocios si ya existe UML?* s.l. : Dept.de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, ETS Ingeniería Informática, 2007.

Pérez Zurita, Paúl. *IH-SW-DR-080 ALAS-HIS_Visión.* 2008.

PostgreSQL. [En línea] PostgreSQL Global Development Group. <http://www.postgresql.org/>.

Pressman, Roger S. *Ingeniería de software, un enfoque práctico.* 2005.

Red Hat. *Ajax4jsf Developer Guide.* 2007.

Red Hat. *RichFaces Developer Guide.* 2007.

RichFaces developer guide. [En línea] <http://www.jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html/index.html>.

RichFaces live demo. [En línea] <http://livedemo.exadel.com/richfaces-demo/index.jsp>.

Sanchez Santos, Leonardo, et al. *Introducción a la Medicina General Integral.* La Habana, Cuba : Ciencias Médicas, 2001.

Seamframework.org. [En línea] Red Hat. <http://seamframework.org/>.

SIGHO Sistema de Información para la Gerencia Hospitalaria. [En línea] <http://www.sigho.gob.mx/quees.htm>.

Stephen A. White, PhD y Miers, Derek. *BPMN Modeling and Reference Guide. Understanding and Using BPMN.* 2008. ISBN-13: 978-0977752720.

Sun microsystems. Sun España. [En línea] <http://es.sun.com/>.

Telvent. Farmacia Hospitalaria. [Online] Televent, 2008.

<http://www.telvent.com/sites/telvent/es/soluciones/administracionespublicas/salud/soluciones/sistemasinformaciondepartamental/farmacia.html>.

The PostgreSQL Global Development Group. *PostgreSQL 8.3.6 Documentation.* 2008.

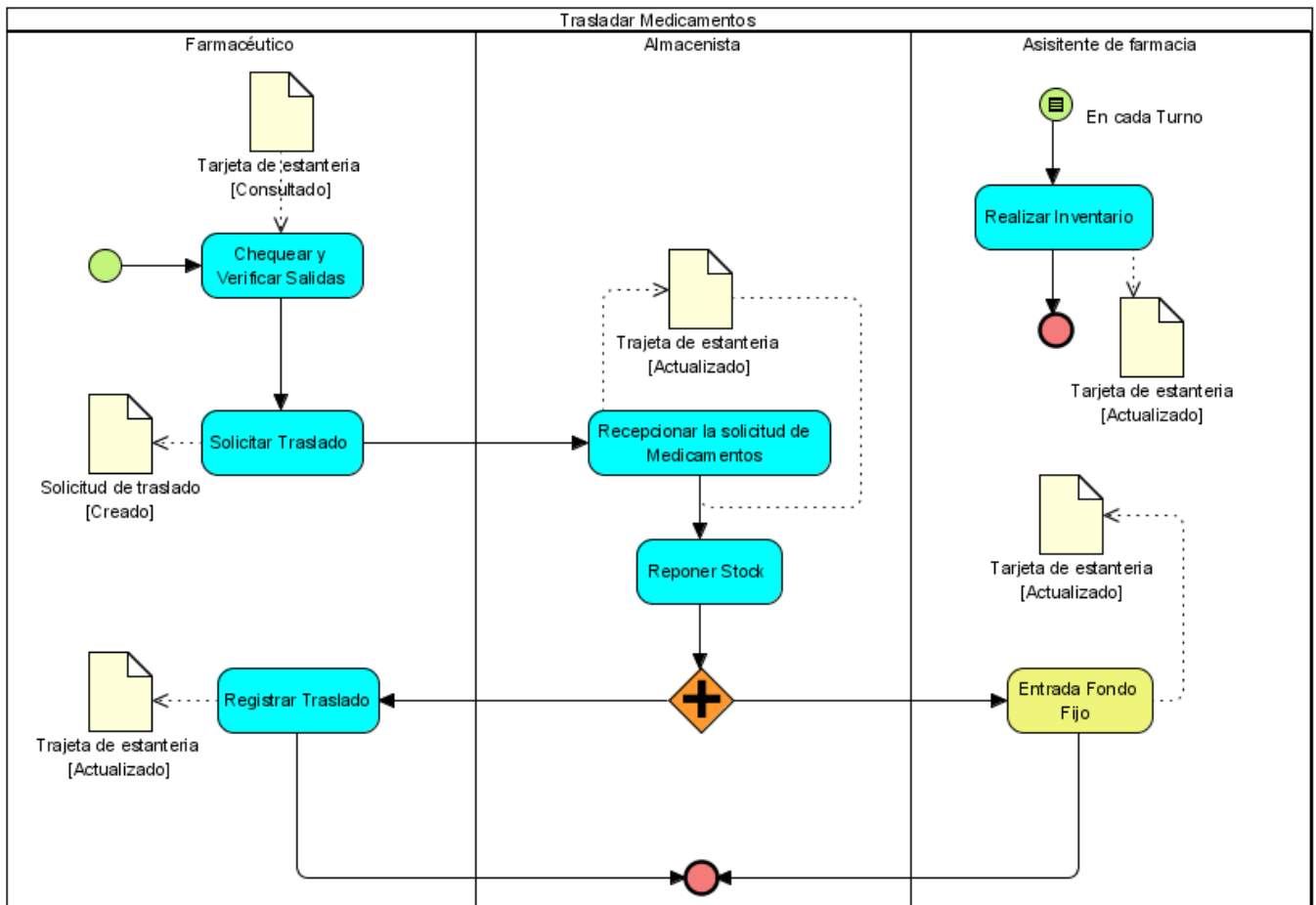
Velázquez Carralero, Alejandro Mario. *IH-SW-DR-091 ALAS-HIS_Documento de Arquitectura del Sistema.* 2008.

Wikipedia. La enciclopedia libre. [En línea] http://es.wikipedia.org/wiki/Aplicación_web.

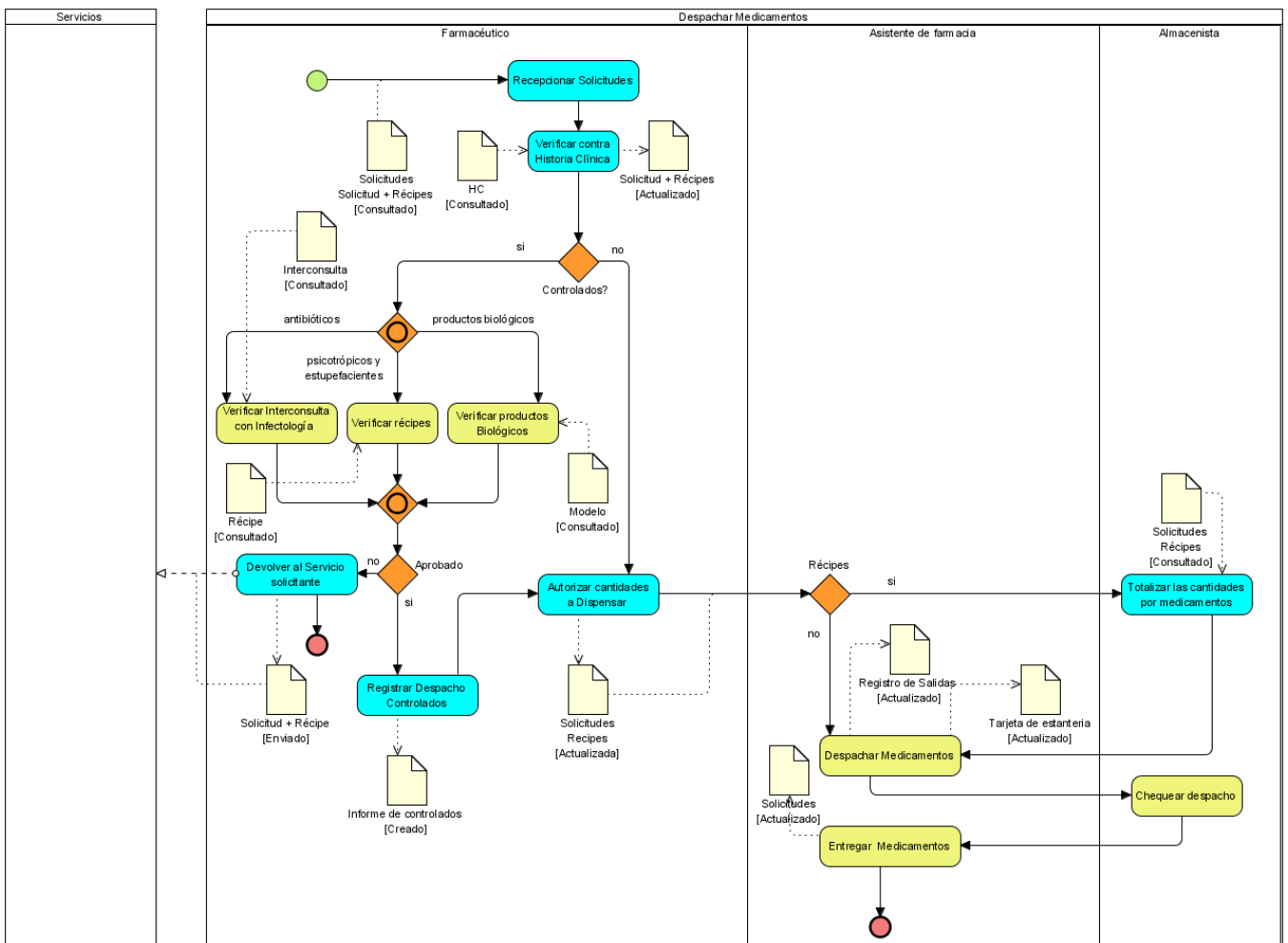
ANEXOS

Anexo #1 Diagrama de procesos del negocio.

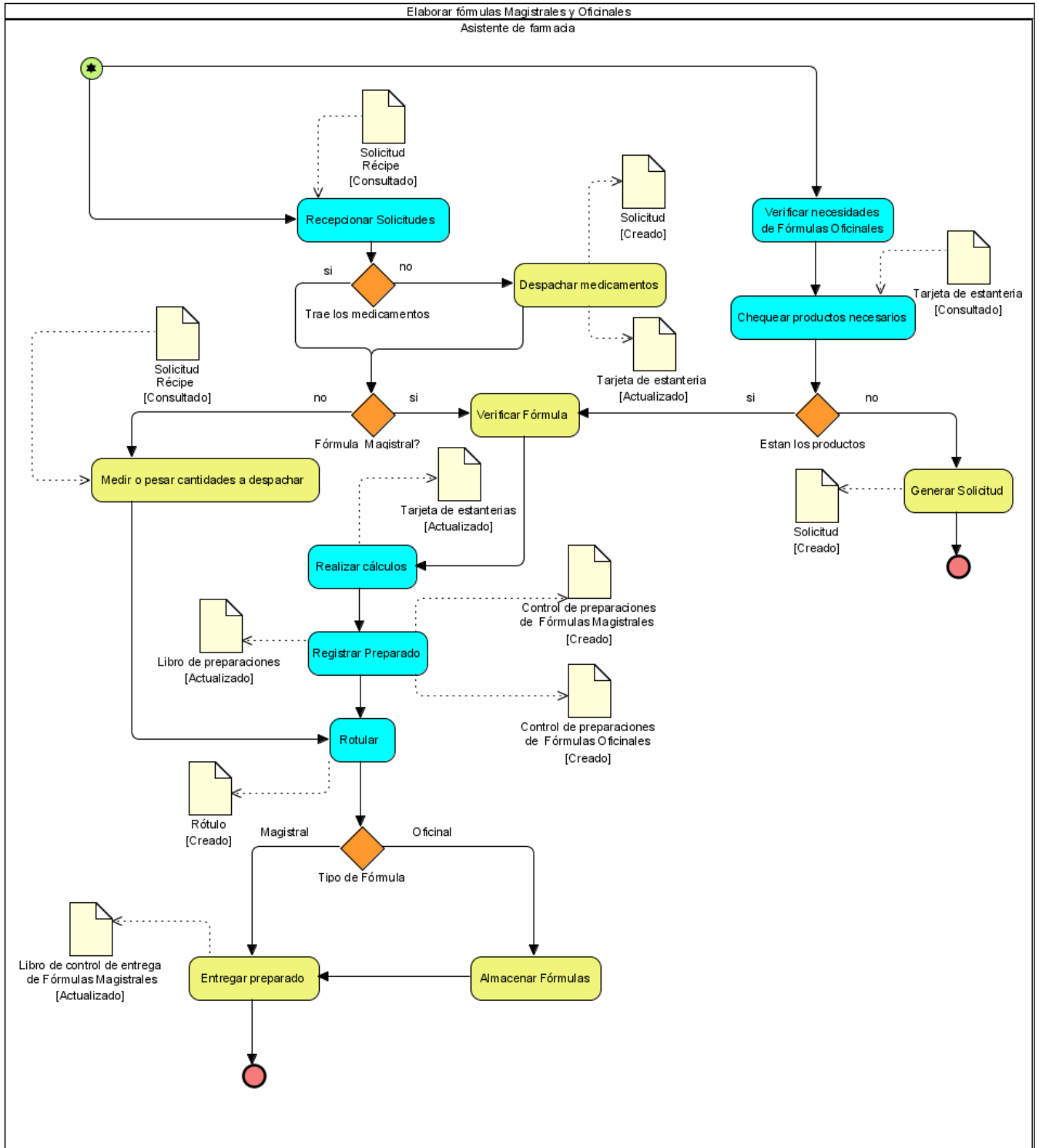
Trasladar medicamentos



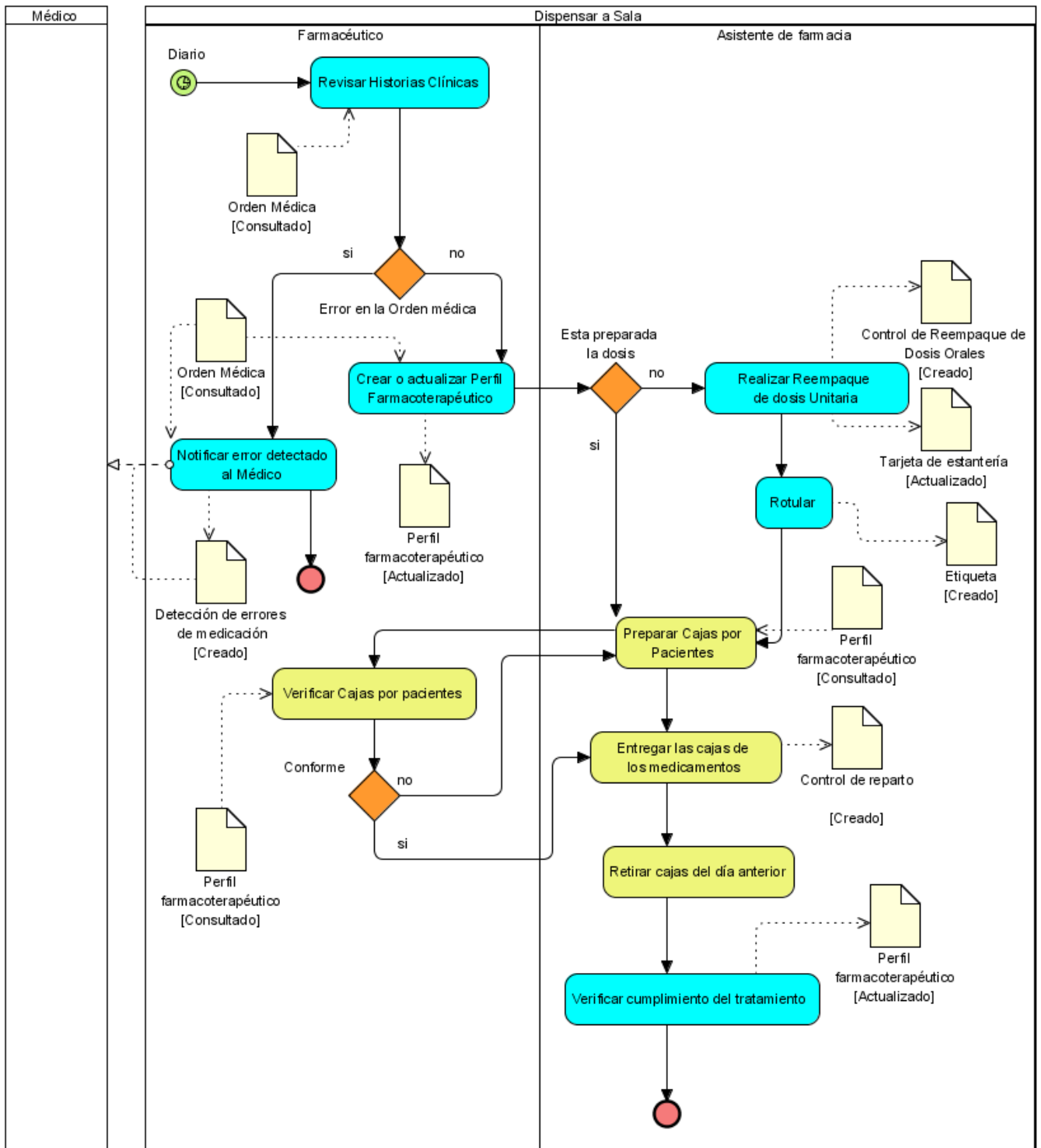
Dispensar medicamentos



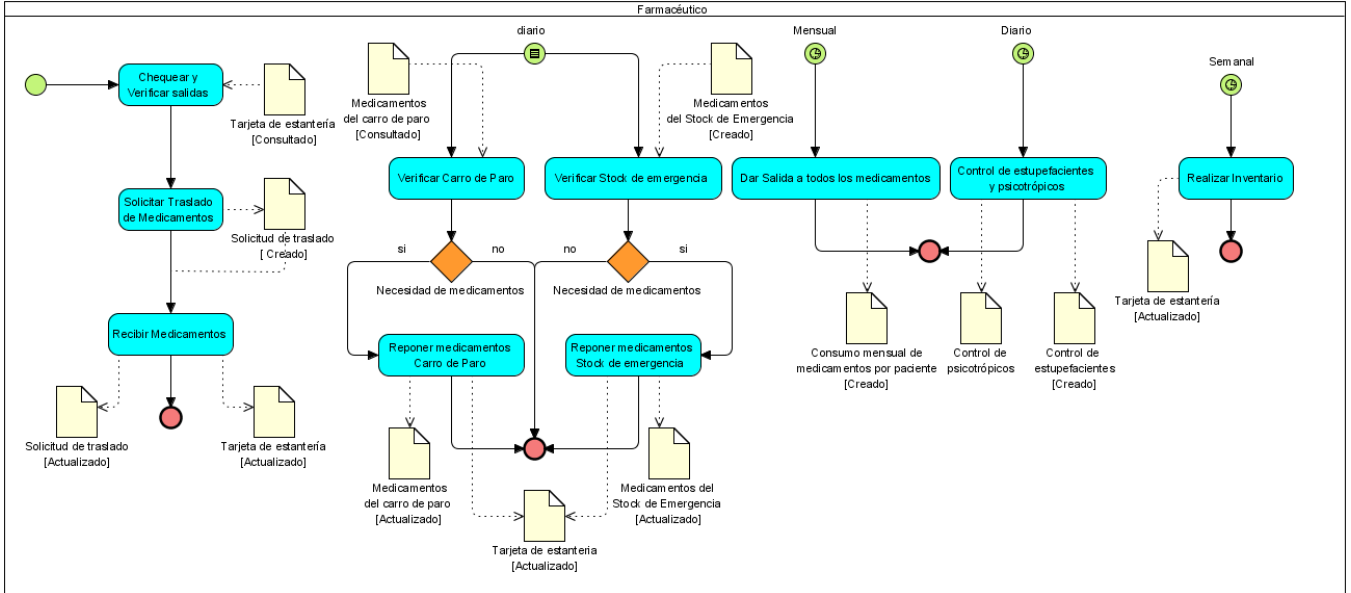
Elaborar formulas magistrales y oficinales



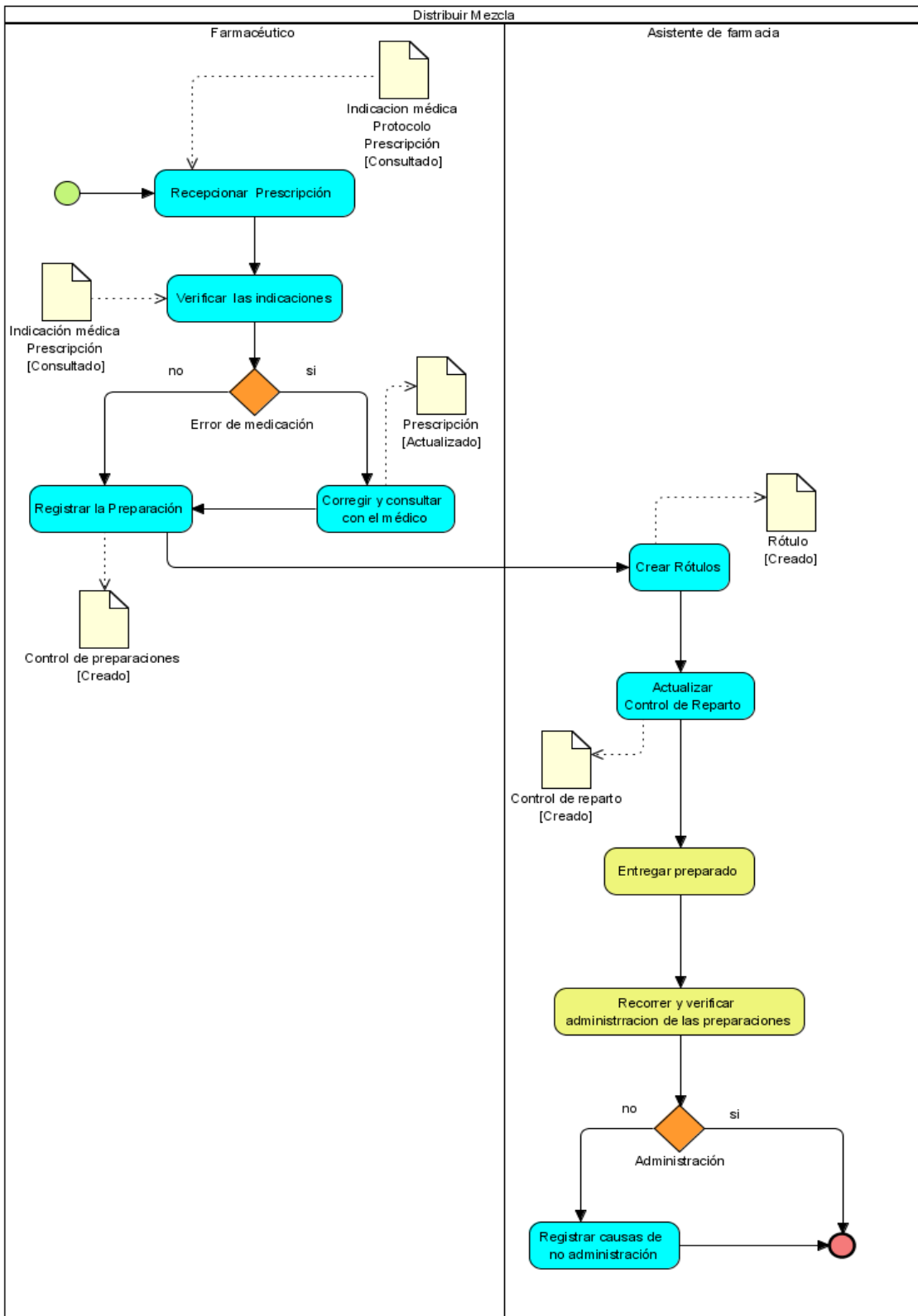
Dispensar a sala



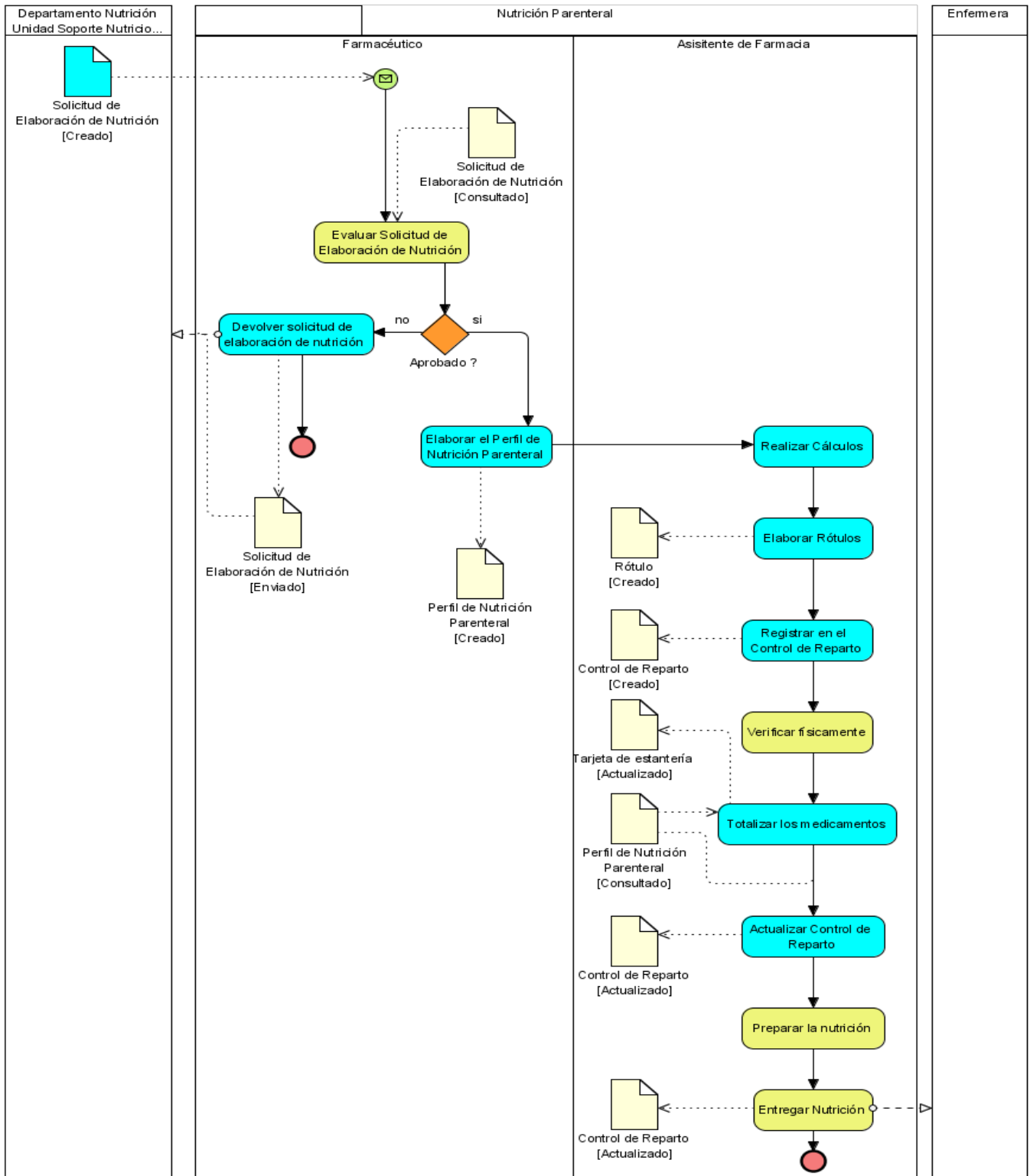
Reponer stock



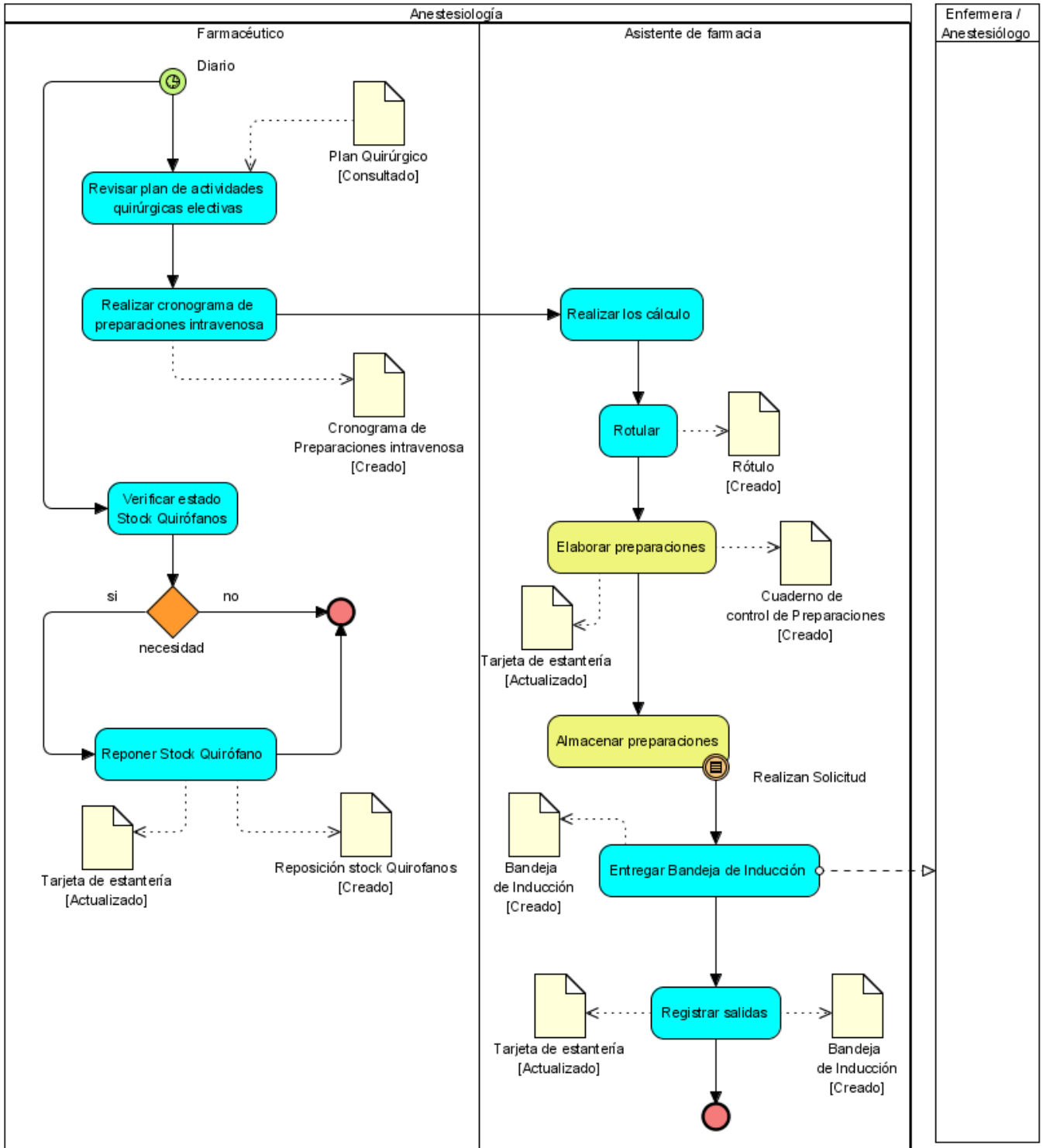
Distribuir mezclas



Nutrición parenteral



Anestesiología



GLOSARIO

API

(*Application Programming Interface*) o 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 librería para ser utilizado por otro software como una capa de abstracción.

Bean

Es un componente *software* que tiene la particularidad de ser reutilizable.

Estupefaciente y psicotrópicos.

Sustancias que su consumo pueda producir un estado de dependencia, estimulación o depresión del sistema nervioso central, o que tenga como resultado alucinaciones, trastornos de la función motora, del juicio, del comportamiento, de la percepción o del estado de ánimo.

Fórmula magistral

Todo medicamento destinado a un paciente determinado, preparado por el farmacéutico o bajo su dirección, según las normas técnicas del arte farmacéutico, a fin de cumplir expresamente una prescripción facultativa individualizada de las sustancias medicamentosas que incluye; éste será dispensado en la farmacia, con la debida información al paciente, sin que se requiera Registro Sanitario para su expendio.

Fórmula oficial

Todo medicamento elaborado y garantizado por un farmacéutico o bajo su dirección, el cual será dispensado en la farmacia, con la debida información al paciente. Para su elaboración se seguirá la normativa establecida en los textos oficiales, sin que se requiera Registro Sanitario para su expendio.

Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

HTML

(*HyperText Markup Language*), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP

(*HyperText Transfer Protocol*), protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la Red Global Mundial.

Java Server Pages (JSP)

Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

LPT

Puerto paralelo para conectar periféricos a una computadora.

Perfil farmacoterapéutico.

Formato por paciente, en donde se registra los datos del paciente, el diagnóstico, los medicamentos que el médico le indicó y un control diario de la entrega y devoluciones de dichos medicamentos.

Plain Old Java Object (pojo)

Enfatiza el uso de clases simples y que no dependen de un *framework* en especial.

Récipe

Documento donde se registra el medicamento que debe consumir el paciente según el tratamiento.

Sistema de dispensación por dosis unitaria.

El Sistema de Distribución de Medicamentos en Dosis Unitarias (SDMDU) es el sistema de distribución que mejor representa al sistema de dispensación con participación previa del farmacéutico. En este sistema, el Servicio Farmacéutico proporciona el medicamento en un envase unificado que contiene la dosis oral o parenteral prescrita al paciente, lista para su administración inmediata sin la necesidad de manipulación o preparación adicional por parte del personal de enfermería.

Sistema de dispensación por prescripción individualizada

Este sistema de dispensación por prescripción individualizada es más profesional que el anterior. Consiste en una petición individualizada de los medicamentos al Servicio Farmacéutico, por parte del personal de enfermería, diariamente o por períodos de 48h – 72 h para cada uno de los pacientes, contra la orden médica original, su copia directa o r cipe m dico del paciente. Estos medicamentos son retirados por el personal de enfermer a en el Servicio Farmac utico y almacenados en el stock en el servicio de hospitalizaci n, para luego ser administrados al paciente.

Sistema de dispensaci n por stock de piso

El sistema de dispensaci n por stock de piso o sistema "tradicional", consiste en establecer en el servicio cl nico correspondiente dep sitos de medicamentos controlados por el personal de enfermer a, con cantidades previamente fijadas de los medicamentos que se requieren para cumplir la farmacoterapia de los pacientes a los que se atienden.

Stock

Existencias o reservas de productos.

TCP/IP

Es un conjunto de protocolos de red que permiten la transmisi n de datos entre redes de computadoras.

Unidad de mezcla intravenosa

Es un área aséptica ubicada dentro de un Servicio de Farmacia de una Institución Hospitalaria, bajo la coordinación de un Farmacéutico, quien mediante el uso de técnicas especializadas se encarga de la preparación, reconstitución, mezcla y dilución de los productos medicamentosos de uso parenteral, así como de la dispensación y control.

Unidosis

Extensión de la Farmacia Central que provee de medicamento a servicios determinados.

USB

El *Universal Serial Bus* (USB), es un puerto que sirve para conectar periféricos a una computadora.

Web

Web, (*World Wide Web*) o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

XML

Extensible Markup Language o lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium*.