

**Universidad de las Ciencias Informáticas**

**Facultad 7**



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

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

**Autores:**

**Yenisley Carrandi Hernández**

**Adnier Guerrero Barea**

**Tutor:**

**Ing. Juan Alfredo Guerra Gongora**

**Ciudad de la Habana, Junio de 2009**

**“Año del 50 aniversario del triunfo de la Revolución”**

---

*“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica”.*

*Aristóteles*

---

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 26 días del mes de junio del año 2009.

\_\_\_\_\_  
Yenisley Carrandi  
Hernández

\_\_\_\_\_  
Adnier Guerrero Barea

\_\_\_\_\_  
Ing. Juan Alfredo Guerra Gongora

---

## DATOS DE CONTACTO

Tutor: Ing. Juan Alfredo Guerra Gongora ([jguerra@uci.cu](mailto:jguerra@uci.cu))

Instructor graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Profesor de la Facultad # 7. Ha impartido las asignaturas Matemática III y Matemática IV, actualmente imparte la asignatura de Teleinformática 1. Forma parte del proyecto Gestión Hospitalaria.

*Al compañero Fidel por darnos la oportunidad de estudiar en una escuela como esta.*

*A nuestros padres por brindarnos todo su apoyo y confianza.*

*Al tutor y oponente por su ayuda incondicional, muchas gracias.*

*A todos los profesores del proyecto.*

*A nuestros amigos.*

*A nuestros compañeros.*

*A todos lo que de una forma u otra nos han ayudado.*

*Gracias...*

**De Yenislely**

*A mi papito y a mi mamita que son mi razón de ser, sin ustedes esto no hubiera sido posible,  
los quiero mucho.*

*A Ale por ser la persona especial que es, por su paciencia, por ayudarme a salir adelante y por  
todas las cosas buenas que recibí de él.*

*A mi compañero de tesis y más que eso a mi amigo Adnier por ayudarme a hacer realidad mi  
sueño.*

*A Yulexy por ser siempre mi amiga y estar presente cuando más lo necesité.*

*A toda mi familia por su apoyo.*

**De Adnier**

*A mis padres, por todo el esfuerzo, cariño y comprensión que siempre me han dado.*

*A mi hermano, por ser la guía y espejo de mi andar en la vida.*

*A mi abuela Claudina por toda la fe que siempre me ha inspirado.*

*A Leticia, por los años que estuvo a mí lado, por su paciencia.*

*A Yisel por su apoyo incondicional.*

*A mi compañera de tesis y amiga Yeni, por su dedicación y esfuerzo por lograr realizar este  
trabajo.*

*A todas las personas que hicieron posible que pudiese estudiar en esta maravillosa escuela.*

*A toda mi familia.*

## **Resumen**

El almacén es una de las áreas de mayor importancia dentro del hospital, de su correcto funcionamiento depende el abastecimiento de los materiales, útiles y equipamientos necesarios para la atención al paciente. Sin embargo, en la actualidad, la mayoría de los hospitales no cuentan con un sistema automatizado y estándar, que lleve el control de las actividades que se realizan en esta área.

El presente trabajo brinda una propuesta de un sistema informático, que permita la automatización del proceso de recepción de los productos en los almacenes hospitalarios. Para el desarrollo del mismo se utilizó el Proceso Racional Unificado (RUP), así como la Notación para el Modelado de Procesos de Negocio (BPMN). Se utilizó el patrón arquitectónico Modelo-Vista-Controlador el cual permitió la implementación de un sistema robusto y flexible.

Se pretende que el sistema propuesto logre contribuir a la mejora de los servicios de abastecimiento en los hospitales, de igual manera que proporcione una información actualizada y accesible en tiempo real, con respecto a la disponibilidad de los productos almacenados.

## **PALABRAS CLAVES**

Sistema de información hospitalaria, almacén hospitalario, recepcionar productos.

**ÍNDICE**

<i>INTRODUCCIÓN</i> .....	<b>1</b>
<b><i>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</i></b> .....	<b>6</b>
1.1 <i>Conceptos básicos relacionados con el campo de acción</i> .....	6
1.2 <i>Sistemas automatizados existentes vinculados al campo de acción</i> .....	7
1.3 <i>Tecnologías utilizadas en el proceso de desarrollo</i> .....	11
1.4 <i>Metodologías de desarrollo</i> .....	15
1.5 <i>Lenguaje de programación</i> .....	16
1.6 <i>Herramientas</i> .....	17
1.7 <i>Sistemas distribuidos. Modelo cliente-servidor</i> .....	19
1.8 <i>Patrones de arquitectura y diseño. Patrón Modelo- Vista- Controlador (MVC)</i> .....	20
<b><i>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</i></b> .....	<b>21</b>
2.1 <i>Flujo actual de los procesos involucrados en el campo de acción</i> .....	21
2.2 <i>Objeto de automatización</i> .....	22
2.3 <i>Información que se maneja</i> .....	22
2.4 <i>Modelado del negocio</i> .....	23
2.4.1 <i>Roles del negocio</i> .....	24
2.4.2 <i>Diagramas de procesos del negocio</i> .....	24
2.5 <i>Propuesta del sistema</i> .....	28
2.5.1 <i>Especificación de los requisitos de software</i> .....	28
2.5.1.1 <i>Requisitos funcionales</i> .....	28
2.5.1.2 <i>Requisitos no funcionales</i> .....	29
2.6 <i>Modelo de casos de uso del sistema</i> .....	33
2.6.1 <i>Actores del sistema</i> .....	33
2.6.2 <i>Vista global de actores</i> .....	34
2.6.3 <i>Diagrama de casos de uso del sistema</i> .....	34
2.6.4 <i>Descripción textual de casos de uso</i> .....	36
<b><i>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</i></b> .....	<b>41</b>

3.1 Modelo de diseño.....	41
3.1.1 Descripción de la arquitectura .....	41
3.1.2 Realización de casos de uso del diseño.....	43
3.1.2.1 Diagrama de clases del diseño .....	46
3.1.2.2 Diagramas de Interacción.....	55
3.2 Descripción de las clases del diseño .....	63
<b>CAPÍTULO 4: IMPLEMENTACIÓN.....</b>	<b>67</b>
4.1 Modelo de datos .....	67
4.1.1 Descripción de las tablas .....	69
4.2 Modelo de implementación.....	81
4.2.1 Diagrama de despliegue .....	82
4.2.2 Diagrama de componentes .....	83
4.3 Tratamiento de errores .....	85
4.4 Seguridad.....	85
4.5 Estrategias de codificación. Estándares y estilos a utilizar .....	86
<b>CONCLUSIONES .....</b>	<b>93</b>
<b>RECOMENDACIONES.....</b>	<b>94</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>95</b>
<b>BIBLIOGRAFÍA.....</b>	<b>98</b>
<b>ANEXOS.....</b>	<b>101</b>
<i>Anexo 1.Descripcion textual de casos de uso .....</i>	<i>101</i>
<i>Anexo 2.Diagrama de clases del diseño.....</i>	<i>111</i>
<i>Anexos 3. Pantallas del sistema.....</i>	<i>115</i>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>135</b>

## ***Introducción***

El desarrollo científico y tecnológico, por el cual ha transitado el mundo desde el pasado siglo, ha tenido una significación especial para la actividad científica y una influencia extraordinaria en la sociedad. La humanidad ha sido protagonista de grandes avances que repercuten, en mayor o menor medida, en el surgimiento y desarrollo de la llamada Revolución Científico Técnica.

La informática es un elemento de gran impacto dentro de esta revolución, no solo en la automatización de los procesos empresariales, sino también en la gestión del conocimiento dirigido a las diferentes ramas socio-económicas. La aplicación de la informática en las más disímiles actividades de la vida humana ha traído como consecuencia importantes cambios en los viejos paradigmas sociales.

Actualmente, muchos países enfrentan el reto de integrarse plenamente a la infraestructura global de la información, que haciendo un uso más eficiente de las nuevas tecnologías, permitirá lograr incrementos sustanciales en la productividad, el mejoramiento de la calidad y la eficiencia en todas las actividades tanto industriales como de servicios. Para ello se lleva a cabo todo un proceso de informatización de los principales sectores públicos. La salud es una de las esferas que tributan grandes beneficios por concepto de la introducción de la informática en su quehacer cotidiano.

El número de transacciones y desorden existente en los centros hospitalarios dieron lugar a la necesidad de estructurar el desarrollo de los datos y de las actividades que se realizan en los mismos, de igual forma surge como prioridad inmediata el control de los procesos organizativos y procedimientos existentes. La imposibilidad de conocer y ordenar la gran cantidad de información que genera el hospital, se asentó sobre las ciencias informáticas, lo cual significó la mecanización de las funciones operativas para hacer más eficientes los procesos hospitalarios[1].

Desde la década de los setenta, en el campo de la medicina, se perfilaron los primeros sistemas de información médica. Pero con el poder actual de las aplicaciones informáticas, y la necesidad de que los hospitales del siglo XXI estén a la altura de las exigencias de los pacientes actuales, se comienza a manipular un nuevo concepto de los sistemas de gestión para la salud, denominado Sistema de Información Hospitalaria (HIS, por sus siglas en inglés).

A nivel mundial, un gran número de empresas están especializadas en el desarrollo e implantación de software para el sector de la salud. Algunas de estas aplicaciones son para hospitales o departamentos específicos de este. Cuba, se encuentra inmersa en la investigación y producción de sistemas informáticos para esta área. A este proceso se vinculan varias instituciones entre las que se destaca la Universidad de las Ciencias Informáticas (UCI), la cual tiene la tarea de desarrollar un HIS para las instituciones hospitalarias.

Un HIS es un sistema de información orientado a satisfacer las carencias existentes de generación de información para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución sanitaria. Esto permite hacer más eficiente la administración de los recursos humanos y materiales, además minimizar los inconvenientes burocráticos que enfrentan los pacientes[2].

Estos sistemas pretenden, de forma general, garantizar la calidad de los servicios médicos y gestionar de forma eficiente el cúmulo de información que se genera en los hospitales, así como permitir la comunicación en línea entre las instituciones hospitalarias. Además cuentan con varios módulos o subsistemas que responden a las necesidades de cada departamento o área especializada con características y funcionamientos particulares.

El almacén hospitalario es el área encargada de recibir para su cuidado y protección todos los materiales y suministros necesarios para un correcto funcionamiento del hospital, manteniendo estos en buen estado y llevando registros diarios de su existencia. Se encarga, además, del despacho de los productos, mediante solicitudes autorizadas, a los departamentos que lo requieran, evitando su retraso y llevando de forma minuciosa controles sobre los mismos. También se ocupa de los trámites pertinentes, de acuerdo a las necesidades de reposición del material existente, para lograr una adecuada existencia de mínimos y máximos.

Dentro del almacén hospitalario se llevan a cabo una serie de procesos, entre los que se pueden destacar:

- Recepción de los productos: Se encarga de recibir la mercancía que previamente ha sido solicitada según las necesidades del centro.
- Almacenamiento de los productos: Este ha de llevarse a cabo siguiendo criterios de clasificación que faciliten la localización cuando los productos sean solicitados. Pueden ser almacenados

ordenándolos según el tamaño, accesibilidad, caducidad o uso clínico. Normalmente a todos los productos se les asigna un código que servirá para facilitar esta labor[3].

- Distribución de pedidos: Para ello, los distintos servicios o unidades de la institución presentan sus pedidos en un impreso normalizado que recibe distintas denominaciones (vale de almacén, hoja de pedido), para que los almaceneros preparen los mismos. Una vez preparado, según la periodicidad previamente determinada (diaria, semanal), el personal auxiliar de cada servicio o unidad retirara el material[4].

Muchos países no cuentan con una aplicación que gestione de forma computarizada los procesos que se llevan a cabo en el almacén de un hospital. Los datos de entrada y salida de los productos se recogen de forma manual y en tarjetas difíciles de actualizar, lo que puede provocar errores en los niveles de existencia y contribuir a una gestión ineficaz de la distribución.

Además, al llevar toda esta información en el formato tradicional (papel), no existe manera de alertar al actor con la requerida antelación que la existencia o reserva de un producto se encuentra en déficit, lo que provoca que el mismo no esté disponible cuando sea solicitado.

El no tener un sistema que gestione de forma dinámica cuándo un producto está próximo a su fecha de caducidad, puede traer como consecuencia realizar una distribución de estos sin que cumplan con la calidad requerida, aumentar el riesgo de desincorporaciones, y al mismo tiempo causar pérdidas financieras por la necesidad de realizar nuevas licitaciones para mantener el *stock* requerido de los productos.

Por otra parte, el informe que se realiza en la recepción de los productos carece de uniformidad y no cuenta con toda la información necesaria, además de correr el riesgo de perderse o deteriorarse y no poder rescatar esta información tan necesaria para mantener un adecuado control en el almacén. Todo lo anteriormente expuesto provoca una carencia de información estadística a nivel de hospital generada en tiempo real, la cual es necesaria para planes de prevención y planificación.

Con este trabajo se pretende mejorar los servicios internos de los almacenes hospitalarios mediante un software que lleve el control de los productos, permita conocer el comportamiento de los ingresos y

egresos en el hospital, mejore la gestión de compras, la reducción de capital circulante, la disminución de errores y roturas de *stock*, además de lograr un abastecimiento continuo y adecuado del hospital.

Asimismo que permita controlar físicamente y mantener todos los artículos en inventario, contar con una constante actualización de los registros en el sistema, facilitar la localización inmediata de los artículos, evitar que se agoten los materiales y conservar los niveles permisibles de máximos y mínimos.

En base a lo antes expuesto, se plantea como **problema**: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de almacén de las instituciones hospitalarias?

En correspondencia con el problema, el **objeto de estudio** está enfocado en el proceso de gestión de información de las instituciones hospitalarias. El **campo de acción** se centra en el proceso de gestión de información en el área de almacén de las instituciones hospitalarias.

Para solucionar los problemas mencionados, se define como **objetivo general**: desarrollar el módulo almacén del Sistema de Información Hospitalaria alas HIS, para facilitar la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo se proponen las siguientes **tareas**:

- Analizar los procesos de negocio asociados al área de almacén de las instituciones hospitalarias.
- Evaluar las tendencias actuales en el mundo de los sistemas de información hospitalaria.
- Asimilar la arquitectura definida por el área temática Gestión hospitalaria para el desarrollo de sus aplicaciones.
- Obtener mediante el Proceso Unificado de Desarrollo, los artefactos generados a partir de los flujos de trabajo “Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
- Implementar el proceso de negocio relacionado con la recepción de productos del módulo almacén.

El presente documento está estructurado en cuatro capítulos:

**Capítulo 1:** Fundamentación Teórica: ubica al lector en el ambiente de desarrollo del módulo de almacén, justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo del mismo.

**Capítulo 2:** Características del sistema: contiene un marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

**Capítulo 3:** Diseño del sistema: se centra en la modelación detallada y la construcción de la estructura de la aplicación.

**Capítulo 4:** Implementación: se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área en cuestión.

## **Capítulo 1: Fundamentación Teórica**

Existen algunos conceptos que están relacionados con el campo de acción y que pueden resultar desconocidos; así como algunos sistemas de gestión hospitalaria que abarcan el área de almacén en los hospitales y de los que se hace necesario su estudio para poder comprender la necesidad concreta del diseño de un HIS. De igual forma existen muchas tecnologías, herramientas y metodologías que facilitan la forma de construir el producto. En el presente capítulo se hace un estudio de sus principales características y ventajas para la realización del sistema.

### **1.1 Conceptos básicos relacionados con el campo de acción**

#### **Almacén hospitalario**

El almacén hospitalario dirige sus esfuerzos a suministrar eficazmente los productos sanitarios y los fármacos necesarios para una correcta atención al paciente, así como los materiales en condiciones óptimas de uso para el funcionamiento del centro sanitario. Esto evita paralizaciones por falta de ellos o inmovilizaciones de capitales por sobre existencias.

#### **Gestión del almacén hospitalario**

Es la actividad encargada de la recepción, almacenaje, despacho, distribución y compra de los productos utilizados para los procesos internos del hospital. Se encarga del control de los movimientos de los productos considerando las entradas y salidas así como de la planeación, programación y puesta en marcha de inventarios periódicos.

Para que el hospital administre sus almacenes de manera eficiente deberá entregar los productos a sus clientes en los plazos pactados, sin que se produzcan demoras y reducir al máximo los costes de almacenamiento, consiguiendo que las existencias medias en el almacén sean las necesarias para evitar “roturas de *stock*”, es decir, dejar pedidos de clientes desatendidos o que el hospital deje de brindar sus servicios por falta de disponibilidad de *stocks*.

#### **Stock**

El *stock* o existencias de un hospital es el conjunto de materiales y artículos que se almacenan a la espera de una posterior utilización. Son recursos ociosos que tienen un valor económico y que están pendientes de ser vendidos o empleados en el proceso hospitalario.

## 1.2 Sistemas automatizados existentes vinculados al campo de acción

En la actualidad, existen muchos sistemas informáticos que automatizan los procesos que se desarrollan dentro de los almacenes hospitalarios, algunos han sido creados para gestionar la información generada en esta área en específico, mientras que otros están integrados como un HIS.

Algunas de las principales características que poseen los sistemas estudiados son:

	País	Software Propietario	Aplicación Web	Aplicación Escritorio	Multiplataforma
<b>CNT PACIENTES</b>	<b>Colombia</b>	x		x	
<b>ASSIST</b>	<b>México</b>	x		x	
<b>X-HIS</b>	<b>España</b>	x	x		x
<b>Kewan – Cosmosalud</b>	<b>España</b>	x		x	
<b>GESTLOG</b>	<b>España</b>	x		x	

Tabla 1- 1 Características de los sistemas

### CNT PACIENTES

Es una aplicación de gestión hospitalaria dirigida a grandes grupos de clínicas, hospitales o empresas prestadoras de servicios de salud, para satisfacer las necesidades de monitoreo y control de los servicios hospitalarios y de las diferentes actividades que diariamente se realizan en las instituciones hospitalarias de alta complejidad.

Este sistema integra varios módulos, entre ellos uno encargado de las actividades que se realizan en la farmacia y almacén que permite optimizar la gestión de inventarios. Provee funcionalidades para registrar órdenes de compra, entradas al almacén y demás movimientos. Permite 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 que ítems se encuentran por debajo del nivel mínimo o del punto de reposición[5].

Esta aplicación posee una edición especializada que fue diseñada para hospitales de primero y segundo nivel denominada CNT PACIENTES Basic Edition. Posee como eje funcional la historia clínica del paciente con un enfoque de prevenir enfermedades en la población[6].

Igualmente, tiene un módulo específico para la gestión en la farmacia y el almacén con las mismas funcionalidades además de permitir optimizar la gestión de inventarios controlando la administración de depósitos de medicamentos o elementos. También provee funcionalidades para el registro de cotizaciones.

CNT PACIENTES HIS Edition en su gran mayoría concuerda con el sistema propuesto pero con el inconveniente de ser un software implementado sobre una aplicación de escritorio. Además no posee algunas funcionalidades que son necesarias para una correcta gestión del almacén, entre ellas la de permitir registrar un producto en cuarentena, realizar devoluciones y aprobar las desincorporaciones.

### **ASSIST**

Sistema de información para hospitales y servicios clínicos que permite manejar la operación, administración y planeación estratégica de un hospital público o privado. Contempla la gestión y control de diferentes áreas, entre ellas se encuentra el área de abastecimientos. Esta es la encargada del surtido de medicamentos, materiales y de su administración.

A través de ASSIST es posible localizar cualquier artículo dentro del almacén, se mantienen varios atributos de identificación de artículos como departamento, familias, sub -familias, líneas, sub - líneas, marca y proveedor. Se pueden contar con múltiples reportes y consultas como el reporte de los movimientos del almacén en un período determinado, niveles de existencia del inventario, en donde el sistema sugiere que productos se deber resurtir y en qué cantidad; además de mostrar la información necesaria para la elaboración de las órdenes de compra[7].

Una importante ventaja del sistema es que desde el momento de realizar las compras se mantiene un control en la fecha de caducidad de todos los medicamentos, de manera que siempre estén vigentes tanto

en el almacén como en las farmacias. Genera solicitudes de compra en forma automática para reponer el inventario a partir de información como análisis de inventario, tasa de uso, ciclo de revisión, manejo de inventario de seguridad, costos de manejos y de órdenes[8].

El sistema también permite la consulta de información estadística de cada artículo, incluyendo costo de venta promedio y última entrada, costo de reposición, rotación de inventario y estadísticas de movimientos en almacén, así como el comportamiento de cada proveedor. Calcula el tiempo promedio de entrega, número de devoluciones e importe, número de pedidos recibidos e importe y porcentaje de pedidos completos[9]. Es un producto muy completo en cuanto a la labor del almacén, por lo que es una referencia obligada para el desarrollo del sistema propuesto.

### **X-HIS**

Sistema de información hospitalaria adaptable y extensible a cualquier centro o grupo hospitalario de carácter público o privado. Es una solución global en un entorno abierto que, además de gestionar electrónicamente la historia clínica del paciente, permite la integración con otros sistemas de información[10].

Este sistema posee funcionalidades para las diferentes áreas dentro del hospital, entre ellas la encargada de la logística facilitando el control, la gestión de almacenes y centrales de compras de cualquier organización sanitaria, cubriendo áreas funcionales como: gestión de pedidos a proveedores, recepción y distribución interna del material, codificación y explotación de datos, contratación administrativa, registro y comprobación de facturas. Es un sistema con una amplia gama de funcionalidades que aunque tiene la ventaja de ser una aplicación web y multiplataforma, no permite generar reportes ni llevar el control de las fechas de vencimiento de los productos.

### **Kewan – Cosmosalud**

Es un sistema de información hospitalaria que propone una solución para la gestión de centros sanitarios, hospitales públicos y privados, clínicas y policlínicos. Incluye la gestión completa de pacientes y sistemas departamentales. Dispone de información *on-line* como apoyo a la toma de decisiones; información en tiempo real: actividad, consumos, costos, demanda, calidad de la asistencia; mejora de procesos; mayor rentabilidad: ahorro de costos, aumento de la satisfacción, mejora en la gestión de recursos y retorno de la

inversión. Automatiza acciones tales como avisos y carga de actos a facturación. Posee herramientas de personalización: consultas, reportes, formularios[11].

Posee diferentes áreas, entre ellas la de logística que es la encargada, entre otras cosas, de la gestión de los almacenes y productos. Entre las funcionalidades de este sistema se encuentra la gestión de compras, de almacenes de farmacias, así como la facturación de compras y ventas.

### **GESTLOG**

Es un sistema para almacenes que automatiza la gestión de compras, distribución, así como el control de los productos en esta área y *stocks* de farmacia[12]. Mantiene un registro de salidas, control de consumos, gestión de *stocks*, cálculo de necesidades, caducidades, control de recepción y control de calidad.

Entre sus objetivos se encuentran:

- Gestión de pedidos a proveedores y control de almacén.
- Gestión de pedidos de clientes, distribución y control de almacén.
- Gestión y control de caducidades.

La aplicación desarrolla los módulos de:

- Clientes (áreas hospital)
- Proveedores
- Compras
- Almacenaje
- Distribución
- Devoluciones de material
- Contabilidad (general, presupuestaria y analítica)

Expone muchas de las funcionalidades necesarias para el correcto control de los productos en el almacén pero entre los inconveniente que presenta se encuentra no permitir generar reportes y ser una aplicación de escritorio.

Muchos de estos sistemas ofrecen soluciones deseables por cualquier institución hospitalaria pero en su mayoría poseen dos características fundamentales que los hacen poco aceptados, una es que son aplicaciones de escritorio lo que dificulta el proceso de despliegue al tener que instalar el sistema en cada una de las estaciones de trabajo. La otra característica que poseen la mayoría de estos sistemas es que no son portables o multiplataforma por lo que atan al usuario a la plataforma sobre la cual se ejecutan.

Por otro lado, estos sistemas no responden totalmente a las necesidades de los usuarios al no contar con todas las funcionalidades requeridas para llevar a cabo una correcta gestión del almacén hospitalario, como es la de generar reportes de entrada y desincorporación de productos, de consumos, de pérdidas, entre otras. Además en su mayoría son sistemas desarrollados para países capitalistas donde se dividen los sistemas hospitalarios en privados y públicos por lo que varios de sus módulos incluyen soportes que no cumplen con los requisitos fundamentales del sistema de salud cubano.

### **1.3 Tecnologías utilizadas en el proceso de desarrollo**

#### **Java Enterprise Edition 5 (JEE5)**

Es la plataforma que provee Sun Microsystems para dar soporte y desarrollar software para las empresas. El gran éxito de Java como plataforma para el desarrollo de aplicaciones se encuentra en esta especificación, que no es más que un conjunto de librerías que establecen un estándar para lograr un producto altamente calificado. Con el lanzamiento de esta edición, la especificación dio un gran paso hacia la excelencia, pues incorporó nuevas tecnologías de punta que no estaban contempladas anteriormente y que indiscutiblemente le incorporan a la plataforma gran robustez y simplicidad a la hora de trabajar, suficientes razones para no dudar en establecerla como base para el desarrollo.

#### **Java virtual machine**

No es más que la base sobre la que se ejecutan todas las aplicaciones Java, encargándose de interpretar todo el código Java y convertirlo al lenguaje nativo del sistema operativo en uso, esta es la razón por la que todas las aplicaciones Java son altamente portables.

#### **1.3.1 Capa de vista**

##### **XHTML**

Es un lenguaje de marcas establecido por la World Wide Web Center que combina el Lenguaje de Marcas Entensible (Extensible Markup Language, XML) y el Lenguaje de Marcas de Hipertexto (siglas de

HyperText Markup Language, HTML) en un mismo formato. Permite extender etiquetas propietarias, además de tener una codificación más rigurosa que HTML. Es el lenguaje de marcas por defecto que utiliza Java Server Face.

### **JavaScript**

Es un lenguaje script e interpretado que se utiliza para incorporar interactividad en las páginas del sistema, este permite capturar los eventos dentro de las páginas para ejecutar alguna acción así como ser responsable de controlar algunas validaciones a nivel de cliente. A diferencia de Java Script, CCS es un estándar para dar formato y estilo a las páginas web, que permite separar el contenido de la presentación.

### **Java Server Face (JSF)**

Es un *framework* (marco de trabajo) orientado a la construcción de interfaces gráficas de usuario (GUI), este marco separa el comportamiento de la presentación, teniendo su propio server como controlador, esto pone de manifiesto el uso del patrón de diseño modelo-vista-controlador, lo que hace más simple su utilización. Otra de las facilidades que brinda su uso, es proporcionar un modelo basado en componentes y dirigido por eventos, filosofía muy similar a las aplicaciones GUI standalone. Permite la creación de nuevos componentes por el desarrollador.

### **Facelets**

Es un marco de trabajo orientado a JSF para el trabajo con plantillas (*templates*), este permite contar con un tiempo cero para el desarrollo de los componentes de interfaces de usuario (UI) y permite separar dichos componentes en diferentes archivos, con soporte para el Lenguaje de Expresión (EL) para las cuestiones de validaciones sin la necesidad de configuraciones en archivos XML. Una de las ventajas que se explotan en el sistema de este *framework* es que no depende de un contenedor web.

### **Ajax4JSF**

Es una librería de código abierto que se integra totalmente a la arquitectura de JSF. Hereda de los componentes JSF y los dota de funcionalidad AJAX de forma limpia y sin necesidad de utilizar código java script. Mediante este *framework* se puede alargar el ciclo de vida de JSF, recargar componentes sin la necesidad de hacerlo en la página completa además de hacer peticiones asíncronas al servidor. Funciona a través de eventos que generan sus propios componentes, generando peticiones al contenedor AJAX sin

la necesidad de preocuparse por crear el XMLHttpRequest, que no es más que el objeto que se obtiene de los contenedores web para hacer peticiones asincrónicas al servidor.

### **Richfaces**

Es una biblioteca de componentes JSF y un avanzado *framework* para la integración de AJAX con facilidades en la capacidad de desarrollo de aplicaciones de negocio. Sus componentes vienen listos para su uso *out-of-the-box*, por lo que los desarrolladores pueden ahorrar tiempo y de inmediato aprovechar las características de los componentes para crear aplicaciones web, que proporcionan mejoras en gran medida en la experiencia del usuario, haciéndola más fiable y más rápida.

Incluye un fuerte apoyo para la *skinnability* (cambio de apariencia) de aplicaciones JSF y aprovecha al máximo los beneficios de este incluyendo la validación y conversión de instalaciones, junto con la gestión estática y dinámica los recursos.

### **Seam UI**

Serie de controles JSF altamente integrables con JBoss Seam. Adicionan varias mejoras a JSF, desde validación, expresiones Extended EL, así como la integración de la navegación en la interfaz de usuario basada en *pageflows* o procesos del negocio.

### **1.3.2 Capa de control**

#### **Seam**

Es un poderoso y moderno *framework* creado para unificar todas las tecnologías estándares JSF, EJB3, JPA, además de BPM (Business Process Management)[13]. Fue creado desde el inicio para eliminar la complejidad a nivel desde arquitectura hasta una Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés), permitiendo la creación de complejas aplicaciones web basadas en POJOs, componentes de UI y XML. Se integra con librerías de controles de código abierto basadas en JSF como Richfaces e ICEFaces[14].

Integra además el concepto de workspaces (entornos de trabajo) permitiendo que el usuario tenga en varios tabs o ventanas del navegador actividades del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía JBoss jBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones e interacciones con el usuario[15].

### Java Business Process Manager (JBPM)

Todo el negocio dentro del almacén hospitalario está caracterizado por las diferentes tareas llevadas a cabo por distintos actores, la necesidad de que cada actor responsable de alguna tarea la realice satisfactoriamente, sin demora en el proceso de realización de esta, lleva a la conclusión de considerar como parte de la arquitectura del sistema, la utilización de algunas de las herramientas existentes para Administrar Procesos de Negocio (BPM) como es JBPM.

Es una tecnología flexible y extensible que cuenta con un lenguaje para graficar y administrar flujos de procesos en términos de tareas, acciones automatizadas, estados de espera para comunicación asíncrona y una larga lista de facilidades que permiten desarrollar un sistema de gran flexibilidad.

Como el objetivo del sistema es lograr un almacén que pueda adaptarse a cualquier entorno en el que se desee utilizar, no se logra una exitosa administración de los flujos de sus procesos si no se tiene la medida de cuanto pueden cambiar las reglas del negocio en un entorno determinado. Para obtener una flexibilidad exitosa se debe contar con una buena administración de dichas reglas. Jboss Drool es un marco de trabajo que se integra con facilidad a Jboss JBPM, el cual pretende cumplir con todas las funcionalidades de un Sistema de Administración de Reglas de Negocio (siglas en inglés BRMS).

### 1.3.3 Capa de modelo

#### Hibernate

Es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y mejorada brinda la posibilidad de generar bases de datos en cualquiera de los entornos soportados: Oracle, DB2, MySql; además es *open source* (código abierto).

Permite expresar consultas en una extensión de SQL (HQL), así como en SQL nativo o utilizando criterios orientado a objetos. Brinda filtros para trabajar con datos históricos, regionales o condicionados por permisos. Puede ser usado para desarrollar y distribuir aplicaciones de forma gratuita. Posee una arquitectura de doble capa por lo que puede ser usado en un entorno agrupado. Soporta la generación automática de claves primarias. Realiza la persistencia de forma transparente sin procesamiento de código de bytes. Se integra con J2EE.

### Enterprise Java Bean 3 (EJB 3)

La necesidad de construir un sistema suficientemente robusto, obliga a identificar las tecnologías que mayor simplicidad ofrecen a la hora de implementar toda la lógica de negocio. EJB en su versión tres, se convierten en toda una plataforma que mediante POJOS permite obtener servicios, ejecutando pequeñas porciones de código en el contenedor EJB, en este caso Jboss Application Server. Se puede contar con servicios de seguridad, persistencia, transacciones y mensajería.

En el caso de la persistencia, debido al auge de los motores de Mapeo de Objetos Relacionales (ORM), surge una nueva especificación: Librería de Persistencia de Java (JPA, por sus siglas en inglés). Esta persiste automáticamente entidades basándose en la técnica de las ORM, define tres características principales: algunas *metadatas* (configuraciones) para establecer el mapeo objeto-relacional, un EntityManager para realizar las operaciones CRUD y un lenguaje de persistencia (JPQL).

### 1.4 Metodologías de desarrollo

#### Proceso Racional Unificado (RUP)

Es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El proceso de software propuesto por RUP tiene tres características esenciales:

- Guiado por los casos de uso: Los casos de uso no son solo una herramienta para especificar los requisitos del sistema sino que constituyen un elemento integrador y una guía del trabajo, de su diseño, implementación y prueba. Además estos no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.
- Centrado en la arquitectura: La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo[16].

- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

### **Notación para el Modelado de Procesos de Negocio (BPMN)**

Es un nuevo estándar para el modelo de flujos de procesos de negocio y servicios web. Proporciona una notación que es fácilmente comprensible por todos los usuarios del negocio. Asegura que los lenguajes basados en XML diseñados para la ejecución de los procesos de negocios, como por ejemplo BPEL4WS (Business Process Execution Language for Web Services) y el Lenguaje para la Administración de Procesos del Negocio (Business Process Modeling Language, BPML), puedan ser visualmente expresados con notación común.

Permite el modelado de los procesos de negocio para permitir que el espectador del diagrama pueda distinguir fácilmente entre las secciones del mismo. Junto con RUP (específicamente en el flujo modelado del negocio) se obtiene un resultado admirable, en el cual sobresalen la claridad, limpieza y calidad.

### **Lenguaje Unificado de Modelado (UML)**

Es un lenguaje que se utiliza para especificar, visualizar, construir y documentar un sistema de software orientado a objetos (OO). Es gratuito, accesible a todos, y conforma la colección de las mejores técnicas de ingeniería que han probado ser un éxito en el modelado de sistemas grandes y complejos.

Además tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue[17].

## **1.5 Lenguaje de programación**

### **Java**

Es un lenguaje de programación desarrollado por la compañía Sun Microsystems, que utiliza el paradigma de la Programación Orientada a Objetos (POO). Es un lenguaje interpretado e independiente a la arquitectura que posee una amplísima biblioteca estándar de clases predefinidas. Además las

aplicaciones Java pueden ser ejecutadas indistintamente en cualquier plataforma sin necesidad de recompilación, tienen un amplio espectro: programación tradicional, distribuida, GUI, web.

Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación[18].

### 1.6 Herramientas

#### Eclipse Europa

El término Eclipse identifica a la comunidad de software libre que lleva el desarrollo de la plataforma Eclipse. Esta comunidad tienen el objetivo de proporcionar una plataforma robusta, escalable y de calidad para el desarrollo de software con el Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) Eclipse. Este IDE cuenta con en un editor de código, un compilador/intérprete y un depurador. Sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación además de brindar la posibilidad de añadir a la plataforma base de Eclipse extensiones (*plugins*) para extender sus funcionalidades.

#### Jboss Tools

Es un *plugin* creado por la compañía Red Hat que se integra a la plataforma base de Eclipse para extender las funcionalidades de su desarrollo. Es toda una suite de herramientas que brinda un soporte desde eclipse a diferentes tecnologías que permiten un desarrollo óptimo para los programadores. Entre los módulos con los que cuenta se pueden apreciar:

- JBoss AS Tool: Provee cuatro servidores, con gran facilidad para arrancar, parar y depurar un proyecto de eclipse, además de presentar óptimas características de despliegue.
- Richfaces VE: Editor visual que propone Exadel, que le da soporte a HTML y JSF además de brindar soporte para los componentes específicos de JBoss Richfaces.
- JBPM Tool: Editor para JBPM (Java Business Process Management) que permite construir y desplegar este workflow.
- Struts Tool: Herramienta que da soporte al *framework* web Struts.

- JBossWS Tool: Herramienta para el trabajo con JBoss WebServices.
- JBoos ESB Tooling: Herramienta que permite el trabajo con los Enterprise Service Bus.
- Portlet Tool: Soporte para Portlet.
- Seam Tool: Incluye soporte para seam-gen, permitiendo todo lo relacionado a la refactorización del código.

### **PgAdmin III**

Es una popular plataforma de administración y desarrollo para el gestor de bases de datos PostgreSQL, permitiendo a los usuarios hacer desde simples consultas SQL hasta desarrollar bases de datos complejas. Esta herramienta fue diseñada para responder a las necesidades de todos los usuarios, cuenta con una interfaz gráfica que soporta todas las características de PostgreSQL y facilita enormemente su 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 y soporte para el motor de replicación Slony-I.

### **Visual Paradigm para UML**

Es una herramienta CASE profesional de licencia gratuita que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software[19].

Este software de modelado UML ayuda a una más rápida y mejor construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Soporta ingeniería inversa, generador de informes, editor de figuras, integración IDE con Eclipse, NetBeans y otros. Además entre sus ventajas se incluyen el modelado colaborativo con CVS y subversión, la generación de documentación y de código base para diferentes lenguajes de programación como Java y su exportación como HTML. Cabe destacar igualmente su robustez, usabilidad y portabilidad.

### **1.7 Sistemas distribuidos. Modelo cliente-servidor**

Un sistema distribuido se define como una colección de sistemas de cómputo autónomos conectados a través de una red y de software de comunicaciones, capaces de cooperar para la realización de una tarea común. En todo sistema distribuido se establecen una o varias comunicaciones siguiendo un protocolo prefijado mediante un esquema cliente-servidor.

Los sistemas cliente-servidor están contruidos de tal modo que la base de datos puede residir en un equipo central llamado servidor y ser compartida entre varios usuarios. Los usuarios tienen acceso al servidor a través de una aplicación de cliente o de servidor.

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

- Todos los elementos de datos están almacenados en una ubicación central en donde todos los usuarios pueden trabajar con ellos.
- No se almacenan copias separadas del elemento en cada cliente, lo que elimina los problemas de hacer que todos los usuarios trabajen con la misma información.
- Las reglas de la organización y las reglas de seguridad se pueden definir una sola vez en el servidor para todos los usuarios.
- Los servidores de base de datos relacionales optimizan el tráfico de la red al devolver solo los datos que la aplicación necesita.
- Los gastos en hardware se pueden minimizar. Como los datos no están almacenados en los clientes, estos no tienen que dedicar espacio de disco a almacenarlos. Los clientes tampoco necesitan la capacidad de proceso para administrar los datos localmente y el servidor no tiene que dedicar capacidad de proceso para presentar los datos.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, solo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### 1.8 Patrones de arquitectura y diseño. Patrón Modelo- Vista- Controlador (MVC)

Un patrón arquitectónico es un patrón de alto nivel que fija la arquitectura global de una aplicación. Definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades. También tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema.

El patrón Modelo-Vista-Controlador es un patrón arquitectónico que separa los datos, las interfaces de usuario y la lógica de control en tres componentes distintos. Cada componente agrupa en módulos toda la aplicación, donde la vista engloba lo referente a las interfaces de usuario, el modelo comprende las clases de acceso a datos, y el controlador toda la lógica de negocio.

- **Modelo:** Es el componente encargado del acceso a datos, representando las estructuras de datos del sistema. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.
- **Vista:** Define la interfaz de usuario, es la información presentada al usuario. Una vista puede ser una página web o una parte de una página.
- **Controlador:** Responde a eventos y actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para generar una página.

Ventajas de la utilización del patrón MVC:

- Facilidad para realizar cambios en la aplicación puesto que cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo se maneja uno de los componentes.
- Se puede modificar uno de los componentes sin conocer cómo funcionan los otros.

En este capítulo se analizaron algunos conceptos fundamentales asociados al dominio del problema y al campo de acción, así como las características principales de software existentes en el mundo que implementan la gestión en los almacenes hospitalarios. Se abordó acerca de las tecnologías, herramientas y metodología definidas para el desarrollo de la propuesta de solución, así como las principales características y ventajas de cada una de ellas.

## **Capítulo 2: Características del Sistema**

En el presente capítulo se presenta el flujo actual de los procesos involucrados en el campo de acción, así como la descripción de los procesos que serán objeto de automatización. Además se modela el negocio propuesto, se plantean los requisitos 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**

Uno de los procesos fundamentales que se llevan a cabo dentro del almacén hospitalario es la recepción de los productos que fueron previamente solicitados de acuerdo a las necesidades del centro. Este proceso tiene el objetivo de realizar la recepción de todos los productos que lleguen al almacén, así como registrarlos para darle entrada.

**Recepcionar productos:** Este proceso comienza cuando el almacenero recibe los productos solicitados, para ello consulta el cronograma de entrega del proveedor y su contrato, además verifica la cantidad que está en la orden de compra contra la que entrega el proveedor en la factura y la que entrega físicamente.

Si la cantidad física del producto entregado cumple con la solicitada y los datos de la factura están correctos, el almacenero elabora el informe de recepción y se lo envía al funcionario de control perceptivo para que lo apruebe. Si los productos cumplen con todas las especificaciones técnicas y administrativas y con la calidad solicitada, el funcionario de control perceptivo avala la recepción de los productos, aprueba el informe de recepción emitido por el almacenero y redacta el acta de control perceptivo. Luego el almacenero procede a darle entrada a los productos en el almacén registrando los mismos en sus respectivas tarjetas de estantería.

De presentarse problemas con los datos de la factura el almacenero le puede solicitar al proveedor una refacturación mediante una nota de remisión o puede devolver los productos al mismo y reflejar los motivos en dicha nota. Si el problema con la factura es porque existe una diferencia de precio, el almacenero emite una nota de débito o crédito al proveedor y crea el informe de recepción para que sea aprobado por el funcionario de control perceptivo. Luego se realizan las actividades descritas anteriormente para darle entrada a los productos. En caso de que en la recepción haya problemas con los productos, el almacenero se los devuelve al proveedor y refleja los motivos en una nota de remisión.

Este proceso se realiza de forma manual debido a que los documentos utilizados para llevarlo a cabo como son el cronograma de entrega, la orden de compra y el informe de recepción, por mencionar algunos, se encuentran en formato de papel. Esto provoca que el proceso de recepción sea lento y que la información recogida en ellos corra el riesgo de extraviarse o deteriorarse, quedando el almacén sin control alguno sobre sus productos. Además que no se tenga una información actualizada sobre la cantidad real de productos en el almacén lo que afectaría la distribución de estos a las demás áreas del hospital.

### **2.2 Objeto de automatización**

Se desea automatizar el proceso de recepción de los productos entregados por los proveedores para un óptimo funcionamiento en el almacén hospitalario. Dentro de este proceso intervienen un grupo de documentos que serán el principal objeto de automatización de la investigación en curso, como son el informe de recepción y el acta de control perceptivo de recepción y de desincorporación, además la nota de débito o crédito en caso de que sea necesario crearla.

También se automatizarán todas las actividades que se encuentran enmarcadas en las entradas de medicamentos y materiales, el control de las entradas de los productos en su respectiva tarjeta de estantería, así como todo el proceso de devolución y de desincorporación de cualquier producto en caso de ser necesario. De esta forma, se agilizará todo el proceso de recepción de los productos dentro del almacén hospitalario lo que ayudará a mantener un mejor control de los mismos contribuyendo así a un adecuado funcionamiento del hospital.

### **2.3 Información que se maneja**

Para la realización de este proceso en el almacén hospitalario se consultan, crean y archivan numerosos documentos con información de vital importancia para el correcto funcionamiento de esta área y a su vez del hospital en general, entre ellos se encuentran:

- a. **Documento de solicitud:** Recoge toda la información referente a la solicitud del producto, el servicio que la realiza, la fecha en que se realizó, el almacén al que se le hace la solicitud, la lista de productos que se requieren, la cantidad pedida y despachada, así como la persona que autoriza la entrega y quién la recibe.

- b. **Informe de recepción:** Guarda información relacionada con la recepción como es el listado de los productos recibidos con el código de cada uno de ellos, el nombre del proveedor, el número de la factura, el funcionario que elabora dicho informe, la fecha y la descripción de la recepción.
- c. **Acta de control perceptivo:** Registra toda la información referente a la recepción, la fecha en que se realizó, el nombre del proveedor, el monto total de la recepción, la descripción de los productos recibidos, la forma de pago del proveedor y el número de la factura y del contrato.
- d. **Tarjeta de estantería:** Recoge toda la información referente a la existencia del producto en el almacén, su código, nombre y descripción. Archiva toda la información referente a los movimientos del producto, de las diferentes entradas y salidas que realiza el almacén para este, en caso de ser una entrada se refleja el número de la factura y el proveedor. Además recoge la cantidad en existencia así como la fecha del movimiento.
- e. **Nota de débito o crédito:** En este documento se ponen los productos que faltan, que sobran o que su precio no coincide con el precio acordado, la cantidad de productos en débito o crédito, el nombre del proveedor, número de la factura y de la orden de compra, así como la fecha en que se realiza la nota y las causas de la misma.
- f. **Nota de remisión:** Guarda la información del centro que realiza la devolución, los productos que entregan, la cantidad y el motivo de la devolución.

### 2.4 Modelado del negocio

Una práctica útil previa al desarrollo de un sistema es el modelado del negocio. La metodología de desarrollo RUP la presenta como la primera dentro del ciclo de desarrollo de un software y tiene su mayor peso durante la fase de inicio debido a que permite obtener una visión de la organización, identificar roles y responsabilidades, así como definir los procesos de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Tiene entre sus objetivos comprender los problemas actuales de la organización e identificar las mejoras potenciales y derivar los requisitos del sistema que va a soportar la organización.

### 2.4.1 Roles del negocio

Roles	Justificación
Proveedor	Es el encargado de hacer las distribuciones de productos de las unidades proveedoras a los almacenes.
Almacenero	Es el encargado de recibir los productos y facturas, validar las mismas y en caso de presentar problemas solicitar una refacturación, crear una nota de débito o crédito o devolver el producto. Se encarga además de darle entrada a los productos.
Funcionario de control perceptivo.	Encargado del control de la calidad de los productos que se consumen en el hospital.

Tabla 2-1 Roles del negocio

### 2.4.2 Diagramas de procesos del negocio

El modelado en BPMN se realiza mediante diagramas con un conjunto de elementos gráficos divididos en cuatro categorías básicas: objetos de flujo, objetos de conexión, swimlanes (Carriles de piscina o calles) y artefactos.

Dentro de los objetos de flujo se encuentran:

- Evento: Son una acción que ocurre durante el curso del proceso de negocio. Estos eventos afectan al flujo del proceso y suelen tener una causa (trigger) o un impacto (resultado). Hay tres tipos de eventos, basados en cuando afectan al flujo: inicial, intermedio y final.
- Actividad: Son acciones que se realizan dentro de un proceso, puede ser atómica o compuesta y se clasifican en procesos, subprocesos y tareas.

- Bifurcaciones o alternativas (Gateways): Se usa para controlar la divergencia o convergencia de la secuencia de flujo, esto determina las tradicionales decisiones, así como la creación de nuevos caminos, la fusión de estos o la unión.

Los objetos de flujo se conectan entre ellos en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. Hay tres objetos conectores que hacen esta función. Estos conectores son:

- Flujo de secuencia: Se usa para mostrar el orden (la secuencia) en el que las diferentes actividades se ejecutarán en el proceso.
- Flujo de mensaje: Se usa para mostrar el flujo de mensajes entre dos participantes del proceso separados (entidades de negocio o roles de negocio).
- Asociación: Se usa para asociar datos, texto, y otros artefactos con los objetos de flujo. Las asociaciones se usan para mostrar entradas y salidas de las actividades.

BPMN soporta los swimlanes con dos constructores principales. Los dos tipos de objetos swimlanes son:

- Pool: Una pool representa un participante de un proceso. Además actúa como un contenedor gráfico para particionar un conjunto de actividades desde otras pools.
- Lane: Una lane es una sub-partición dentro de una pool y extiende la longitud del pool, verticalmente u horizontalmente. Las lanes se usan para organizar y categorizar actividades.

Se puede añadir cualquier número de artefactos a un diagrama como sea apropiado para un contexto de proceso de negocio específico. Entre los tipos de artefactos se encuentran los objetos de datos que son un mecanismo para mostrar como los datos son requeridos o producidos por las actividades. Están conectados a estas a través de asociaciones y sirven de entrada para la realización de las mismas. A continuación se representa el diagrama de procesos de negocio.

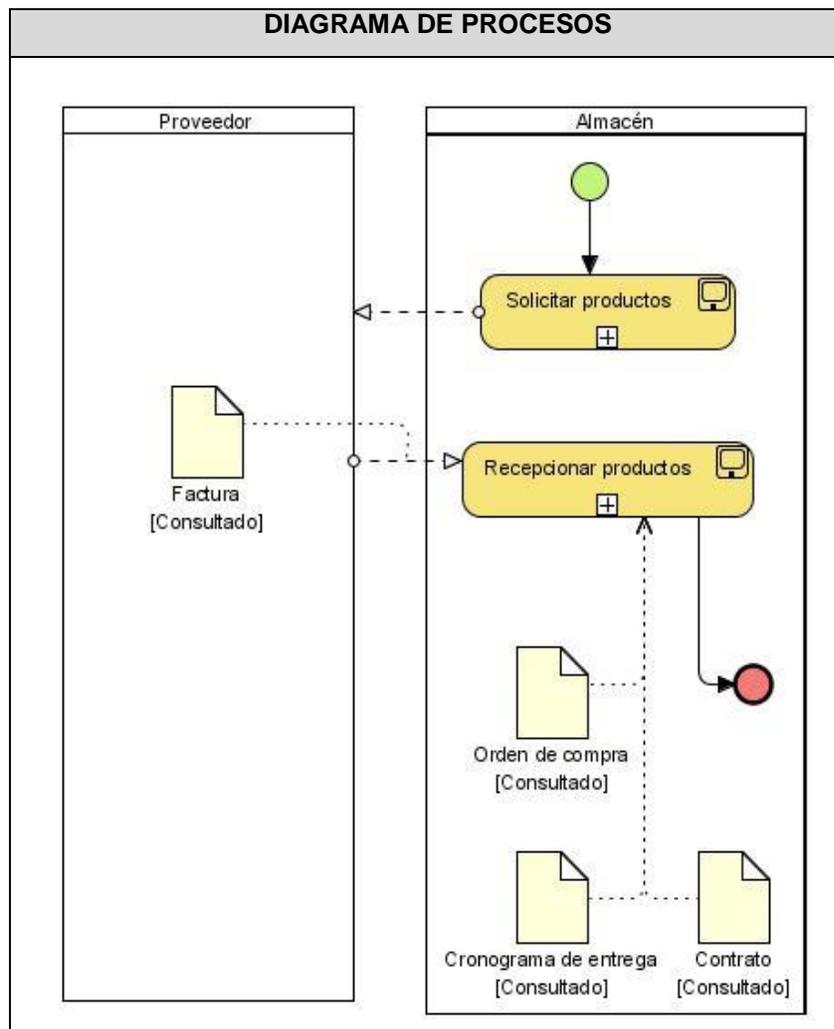


Diagrama 2-2 Diagrama de procesos de negocio

En el diagrama 2-2 se representa el proceso que constituye el objeto de estudio del trabajo de investigación. Además se muestra el proceso Solicitar productos para representar la dependencia que existe entre ambos.

A continuación se muestra el diagrama del proceso recepcionar productos donde se describen los flujos de actividades del mismo.



### 2.5 Propuesta del sistema

A partir de la información obtenida en la descripción del proceso recepcionar productos, el diagrama de dicho proceso y una vez identificado el objeto de automatización se pueden identificar y clasificar los requisitos que tendrá el software a construir, así como encontrar actores y casos de uso del sistema los cuales se forman agrupando requisitos funcionales.

#### 2.5.1 Especificación de los requisitos de software

Para la elaboración de un software con la calidad requerida la especificación de los requisitos constituye una etapa de vital importancia, para ello todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales.

##### 2.5.1.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:

**RF-1** Recepcionar productos de proveedores

**RF-2** Recepcionar productos

**RF-3 Gestionar informe de recepción**

3.1 Crear informe de recepción

3.2 Ver detalles de informe de recepción

3.3 Buscar informe de recepción

3.4 Ver informe de recepción

3.5 Modificar informe de recepción

3.6 Eliminar informe de recepción

- RF-4** Aprobar recepción
- RF-5** Crear acta de control perceptivo de recepción
- RF-6** Seleccionar desincorporación
- RF-7** Seleccionar solicitud de desincorporación de productos biológicos
- RF-8** Aprobar desincorporación
- RF-9** Desincorporar productos
- RF-10** Desincorporar productos biológicos
- RF-11** Crear acta de control perceptivo de desincorporación
- RF-12** Seleccionar devolución
- RF-13** Aceptar devolución
- RF-14** Registrar devolución a proveedores
- RF-15** Seleccionar adjudicación recibida
- RF-16** Crear nota de débito o crédito
- RF-17** Registrar cuarentena
- RF-18** Seleccionar cronograma de entrega
- RF-19** Consultar tarjeta de estantería
- RF-20** Buscar producto
- RF-21** Buscar solicitud de productos

### **2.5.1.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no

funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

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

### **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 y usuarios avanzados: 30 días.

### **2. Seguridad**

- Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos solo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando solo 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 base datos.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

### **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.

#### **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.

#### **5. Hardware**

##### **5.1 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.

##### **5.2 Servidores**

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

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

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

### 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 y Firefox 2 o superior.

### 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.

### 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.

### 9. Interfaz

#### 9.1 Interfaces de usuario

- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.
- La interfaz contará con teclas de función y menús desplegables que faciliten y aceleren su utilización.
- La entrada de datos incorrecta será detectada claramente e informada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.

## **2.6 Modelo de casos de uso del sistema**

El modelo de casos de uso del sistema representa la vista externa de este y es utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre qué debería y qué no debería hacer el sistema. Este modelo está formado por actores, casos de uso y las relaciones que se establecen entre estos.

### **2.6.1 Actores del sistema**

Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan de alguna manera con él. A continuación se definen los actores del sistema propuesto:

<b>Actores</b>	<b>Justificación</b>
Jefe de almacén	Participa activamente en el proceso de adquisición de productos supervisando, conjuntamente con el funcionario de control preceptivo, la recepción de los mismos procedentes de los diferentes proveedores de la institución o de otros almacenes. Se encarga además de registrar los productos en cuarentena cuando están próximos a vencer, aceptar las devoluciones que se le hagan al almacén, así como de realizar las desincorporaciones de los productos cuando sea necesario.
Funcionario de Control Preceptivo	Es el encargado de supervisar y aprobar la recepción de los productos que llegan al almacén dejando reflejado en un acta de control preceptivo de recepción la calidad de los mismos, además de algunos datos propios de la recepción como es el número de contrato y de la factura.
Administrativo	Es el encargado de aprobar las desincorporaciones solicitadas por un almacén.

**Tabla 2-4 Actores del sistema**

### 2.6.2 Vista global de actores

A continuación se muestra el diagrama de actores del sistema para una mayor comprensión de la relación entre ellos:

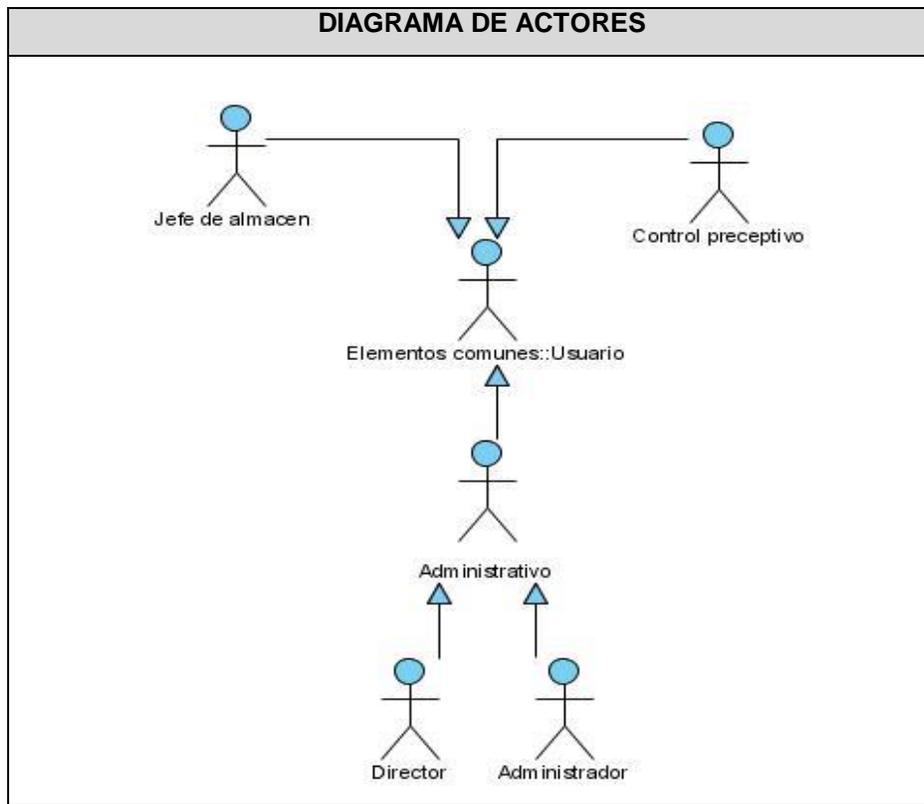


Diagrama 2-5 Diagrama de actores del sistema

### 2.6.3 Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente al proceso y su interacción con los actores. A continuación se muestra el diagrama de casos de uso del proceso recepcionar productos del módulo almacén.

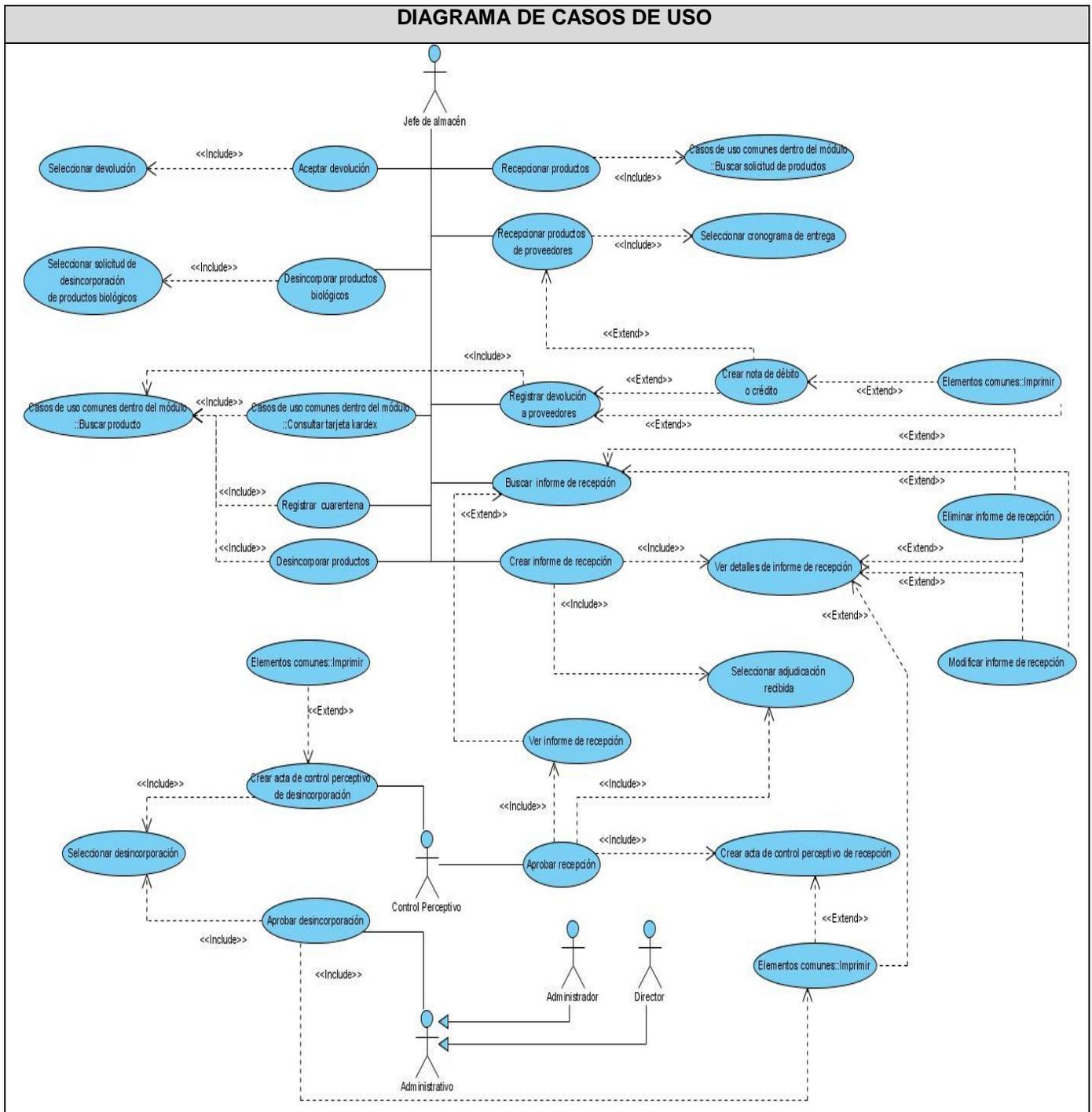


Diagrama 2-6 Diagrama de casos de uso del sistema

#### **2.6.4 Descripción textual de casos de uso**

Los casos de uso son un conjunto de secuencia de acciones que un sistema ejecuta y produce un resultado observable para un actor. A continuación se describen algunos de los casos de uso más significativos del sistema. Para consultar las restantes descripciones de casos de usos ver Anexo 1.

<b>CASO DE USO:</b>	<b>Recepcionar productos de proveedores</b>
<b>Propósito:</b>	Darle entrada a los productos que fueron solicitados a los proveedores.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide darle entrada a los productos de la solicitud de productos realizada a los proveedores según el cronograma de entrega, registrando la cantidad recibida de cada uno de los productos y el número de factura del proveedor, si el producto es nuevo en el almacén el sistema permite crearle una ubicación dentro del mismo, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se recepcionaron productos de proveedores.
<b>Referencia:</b>	RF - 1

**Tabla 2-7 CU Recepcionar productos de proveedores**

<b>CASO DE USO:</b>	<b>Recepcionar productos</b>
<b>Propósito:</b>	Darle entrada a los productos.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide darle entrada a los productos ya sea porque hayan sido solicitados a otro almacén o porque haya sido una donación, registrando para eso la cantidad recibida, si el producto recepcionado es nuevo en el almacén el sistema permite crearle una ubicación dentro del mismo, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se recepcionaron productos.
<b>Referencia:</b>	RF - 2

Tabla 2-8 CU Recepcionar productos

<b>CASO DE USO:</b>	<b>Crear informe de recepción</b>
<b>Propósito:</b>	Crear un informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción crear informe de recepción, el sistema brinda la posibilidad de introducir los datos para crear el informe de recepción, el actor introduce los datos de este, el sistema lo crea, el caso de uso termina.
<b>Precondiciones:</b>	Que se haya efectuado la recepción.
<b>Poscondiciones:</b>	Se creó un informe de recepción.
<b>Referencia:</b>	RF – 3.1

Tabla 2-9 CU Crear informe de recepción

<b>CASO DE USO:</b>	<b>Aprobar recepción</b>
<b>Propósito:</b>	Aprobar las recepciones de productos realizadas.
<b>Actores:</b>	Funcionario de control perceptivo
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide aprobar las recepciones de productos realizadas, el sistema le brinda la posibilidad de seleccionar la recepción que desea aprobar, le carga todos los datos de la misma, le brinda la posibilidad de crear el acta de control perceptivo de recepción y ver los datos del informe de recepción creado, el caso de uso termina.
<b>Precondiciones:</b>	Deben de estar registradas las recepciones y creado el informe de recepción de estas.
<b>Poscondiciones:</b>	Se aprobó una recepción.
<b>Referencia:</b>	RF - 4

**Tabla 2-10 CU Aprobar recepción**

<b>CASO DE USO:</b>	<b>Crear acta de control perceptivo de recepción</b>
<b>Propósito:</b>	Crear un acta de control perceptivo de recepción.
<b>Actores:</b>	Funcionario de control perceptivo
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción crear acta de Control perceptivo de recepción, el sistema brinda la posibilidad de introducir los datos para crear la misma, el actor introduce los datos de esta, el sistema crea el acta de control perceptivo, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se creó un acta de control perceptivo.
<b>Referencia:</b>	RF - 5

**Tabla 2-11 CU Crear acta de control perceptivo de recepción**

<b>CASO DE USO:</b>	<b>Crear acta de control perceptivo de desincorporación</b>
<b>Propósito:</b>	Crear acta de control perceptivo de desincorporación.
<b>Actores:</b>	Funcionario de control perceptivo
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide crear el acta de desincorporación, el sistema lista las tareas con las actas de control de desincorporación por crear, el usuario selecciona el acta que desea crear, el sistema le lista los productos que fueron desincorporados, el actor selecciona el producto deseado y el sistema le permite introducir los datos del acta de desincorporación, el actor introduce los datos, el sistema muestra los datos del acta de control de desincorporación, el sistema permite aceptar crear el acta o eliminar la selección del producto, el caso de uso termina.
<b>Precondiciones:</b>	Debe de estar registrada al menos una desincorporación.
<b>Poscondiciones:</b>	Se creó un acta de control perceptivo de desincorporación.
<b>Referencia:</b>	RF - 11

**Tabla 2-12 CU Crear acta de control perceptivo de desincorporación**

<b>CASO DE USO:</b>	<b>Registrar devolución a proveedores</b>
<b>Propósito:</b>	Registrar una devolución a proveedores.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor registra una devolución ya sea a otro almacén o a un proveedor, registrando la cantidad del producto que va a ser devuelta y las causas de la devolución, en caso de que la devolución sea a un proveedor se crea además una nota de débito o crédito, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se registró una devolución.
<b>Referencia:</b>	RF – 14

**Tabla 2-13 CU Registrar devolución a proveedores**

<b>CASO DE USO:</b>	<b>Crear nota de débito o crédito</b>
<b>Propósito:</b>	Crear una nota de débito o crédito.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción crear nota, el sistema brinda la posibilidad de escoger el tipo de nota que desea crear ya sea de débito o de crédito y le permite introducir las causas de la misma, el sistema crea la nota, el caso de uso termina.
<b>Precondiciones:</b>	Debe existir algún problema con la recepción de productos de proveedores o que haya una devolución a un proveedor.
<b>Poscondiciones:</b>	Se creó una nota de crédito o débito.
<b>Referencia:</b>	RF- 16

**Tabla 2-14 CU Crear nota de débito o crédito**

En el presente capítulo se detalló el funcionamiento del proceso de recepción de productos en el almacén hospitalario, sus principales consecuencias e inconvenientes. Con este análisis se definió el objeto de automatización que permite mejorar el funcionamiento del proceso descrito. Se especificaron los requisitos funcionales y no funcionales del sistema, los actores que intervienen, así como los casos de uso del mismo. Todo esto permitió alcanzar un mejor entendimiento del sistema que se desea construir y las restricciones que deben existir para satisfacer las necesidades de los clientes, permitiendo comenzar el desarrollo del diseño de la solución propuesta.

### **Capítulo 3: Análisis y Diseño del Sistema**

En el diseño se modela el sistema y encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos y las restricciones que se le suponen. Esto contribuye a una arquitectura estable y sólida. Tiene entre sus propósitos adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y de interfaz de usuario.

Permite crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases. Entre los artefactos que se generan en este flujo de trabajo se encuentra el modelo de diseño el cual sirve de abstracción a la implementación.

#### **3.1 Modelo de diseño**

Es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar[20].

Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas. Es utilizado como una entrada fundamental para las subsiguientes actividades de implementación. Entre los artefactos del modelo de diseño se encuentran: descripción de la arquitectura y realización de casos de uso.

##### **3.1.1 Descripción de la arquitectura**

La arquitectura de software propone los elementos necesarios que sirven como guía para lograr el desarrollo exitoso del software. En el Proceso Unificado Racional, específicamente la fase de elaboración, el principal artefacto que se obtienen es el documento de arquitectura, en el cual quedan plasmados los estilos y patrones arquitectónicos que se utilizaran durante el proceso, además de las tecnologías y herramientas, especificando de forma precisa como se integran para lograr disminuir el tiempo y costos de producción. La arquitectura de un sistema, ofrece de forma conceptual, la descripción del software en subsistemas y componentes, dejando bien claro la forma en que estos interactúan.

Teniendo en cuenta que el módulo de almacén de un HIS debe ser lo suficientemente configurable y flexible para que pueda ejecutarse en cualquier entorno donde se desee utilizar, se impone la necesidad de realizar un desarrollo modular, basado en componentes reutilizables, que haga del sistema un software donde los cambios en sus reglas de negocio, provoquen el menor costo en tiempo de producción.

Todos estos conceptos, hacen que la arquitectura sea un marco de suma importancia, donde quedan reflejados todos los argumentos anteriormente descritos. A continuación se presentan algunos elementos arquitectónicos del sistema.

### **Patrón Modelo-Vista-Controlador (MVC)**

El patrón MVC, generalmente, se pone de manifiesto en las aplicaciones web. Actualmente, existe una gran variedad de *framework*, que su utilización impone por defecto el uso de este patrón. Al proponerse como base de la arquitectura la plataforma de desarrollo Java, se tiene la posibilidad de contar con librerías y especificaciones que evidencian de forma clara su uso. El patrón MVC queda reflejado en el módulo de almacén hospitalario de la siguiente manera:

**El modelo:** Está representado por todas las entidades persistentes del sistema. El uso de un motor objeto relacional como es Hibernate, permite cumplir con el paradigma que propone este componente del patrón, lo que facilita una total independencia del tipo de gestor de base de datos. Todas las entidades del sistema hacen función de Bean, que no son más que clases de Objetos Simples de Java (POJOs), siendo estos el medio para conectar la vista con la lógica aplicativa.

**El controlador:** Engloba toda la lógica de negocio del módulo almacén. Contiene las reglas de gestión de eventos y se encarga de modificar tanto la vista como el modelo. Aquí residen las clases controladoras, que contienen alguna meta data que las convierten en componentes Seam, marco de trabajo escogido para el desarrollo.

**Las vistas:** En esta residen todas las clases interfaces con las que interactúa directamente el usuario, al ser llamadas la cara del sistema, estas clases están construidas mayormente con componentes JSF. Estos componentes hacen peticiones al controlador a través de eventos que lanzan y son controlados por un *servlet*, encargado del ciclo de vida de estos y que a su vez, transporta los datos provenientes de las controladoras, actualizando de forma eficiente la interfaz de usuario.

### **3.1.2 Realización de casos de uso del diseño**

Es una colaboración en el modelo de diseño que describe cómo se realiza un caso de uso específico, y cómo se ejecuta en términos de casos de uso del diseño. Una realización de caso de uso del diseño tiene diagramas de clases que muestran las clases de diseño participantes y diagramas de interacción que muestran la realización de un flujo o escenarios concretos de un caso de uso en términos de interacción entre objetos del diseño.

La realización de los casos de uso está estructurada en paquetes. Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí, lo que permite dividir el sistema en trozos más manejables para su futura implementación. Para la realización del diagrama de paquetes se utilizan dos criterios de empaquetamiento: por procesos y por clases. A continuación se explica la organización por paquetes del sistema y se muestra el diagrama de paquetes.

**Repositorio de clases:** Contiene todas las clases definidas en el diseño de acuerdo a la tecnología, que serán usadas en la implementación del módulo. Este paquete está compuesto por tres subpaquetes:

**Entidades (Entity):** Contiene las entidades autogeneradas y personalizadas de la base de datos.

- Autogeneradas: Entidades que se autogeneran desde la base de datos.
- Personalizadas: Entidades personalizadas, que son modificadas. Estas pueden heredar de las entidades autogeneradas.

**Sesiones:** Contiene las clases controladoras.

- Controladoras autogeneradas: Clases controladoras autogeneradas por el entorno de desarrollo.
- Controladoras personalizadas: Clases controladoras personalizadas que son modificadas.
- Controladoras del proceso: Clases controladoras propias del proceso.

**Vistas:** Contiene los contenidos web referentes a las páginas clientes y los formularios que las componen.

**Recepcionar productos:** Es el paquete referente al proceso el cual contiene subpaquetes que responden a las realizaciones de algunos de los casos de uso más significativos del diseño. El nombre de cada paquete se corresponde con el del caso de uso y están compuestos por los diagramas de clases del

diseño que evidencian las clases que se definen para el desarrollo del sistema, conjuntamente con cada uno de los diagramas de secuencia de las diferentes secciones o escenarios, que manejan objetos de las diferentes clases y muestran una secuencia lógica de llamadas entre los objetos, brindándole a los futuros desarrolladores una imagen clara de cómo deben implementar el software propuesto. Utiliza el paquete repositorio de clases para su funcionamiento.

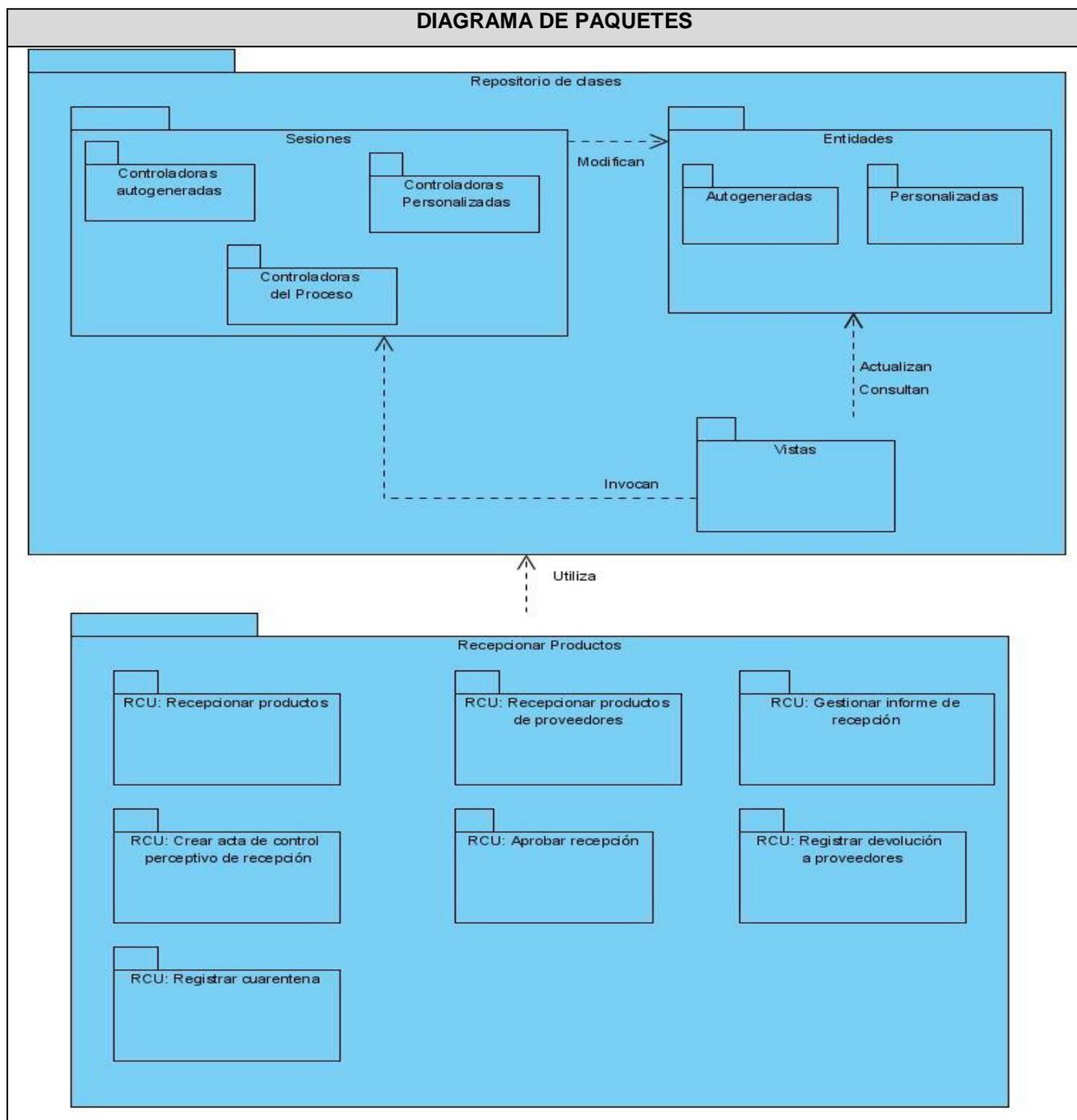


Diagrama 3-1 Diagrama de paquetes

### 3.1.2.1 Diagrama de clases del diseño

Una clase de diseño es una abstracción de una clase o construcción en la implementación del sistema. Los diagramas de clases del diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema y son importantes no solo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Los diagramas de clases del diseño fueron modelados incluyendo los estereotipos web. Un estereotipo es una sutileza que permite definir un nuevo significado de la semántica para el elemento a modelar. A continuación se describen los estereotipos web utilizados para modelar los diagramas de clases:

**Página servidora:** Representa la página web que tiene código que se ejecuta en el servidor y que interactúa con recursos en el mismo. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

**Página cliente:** Una instancia de página cliente es una página web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para páginas con cualquier función dentro de esta.

**Formulario:** Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (input boxes, text áreas, radio buttons, check boxes y hidden fields). No tienen operaciones.

**Build:** Representa una asociación especial que relaciona las páginas cliente con las páginas servidor, de forma general se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidor puede construir una o más páginas cliente.

**Submit:** Es la relación que se crea siempre entre una página servidor y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidor.

**Redirect:** La página de servidor además de construir una página cliente puede redireccionar el procesamiento a otra página. Esto se representa con una asociación con el estereotipo <<redirect>>.

Para la elaboración del diseño, generalmente, se utilizan un grupo de patrones o modelos para lograr objetivos específicos. Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, es una solución estándar para un problema común de programación y una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.

Los patrones de diseño facilitan la reutilización del conocimiento experto como componentes de diseño y mejoran así la documentación, comprensión, comunicación del diseño final y el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución. Identifican: clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Dentro de los patrones de diseño más conocidos están los Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés).

Los patrones GRASP se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Dentro de este grupo se identifican cinco patrones muy utilizados: experto, creador, alta cohesión, bajo acoplamiento y el controlador.

En los diagramas de clases que se elaboró no hay necesidad de crear nuevas clases, simplemente se aplican estos patrones a las que ya fueron definidas en el diseño. El uso de estos patrones permitió distribuir las responsabilidades entre las clases de forma tal que no existan muchas relaciones y que no se sobrecargara de métodos a una en específico pudiendo acomodarlos en otras.

Los diagramas de clases de diseño del sistema están separados en casos de usos, es decir, cada caso de uso cuenta con un diagrama de clases, para hacer más fácil su comprensión. A continuación se muestran los diagramas de clases del diseño de algunos de los casos de uso más significativos del sistema. Para consultar los restantes diagramas de clases del diseño ver Anexo 2.

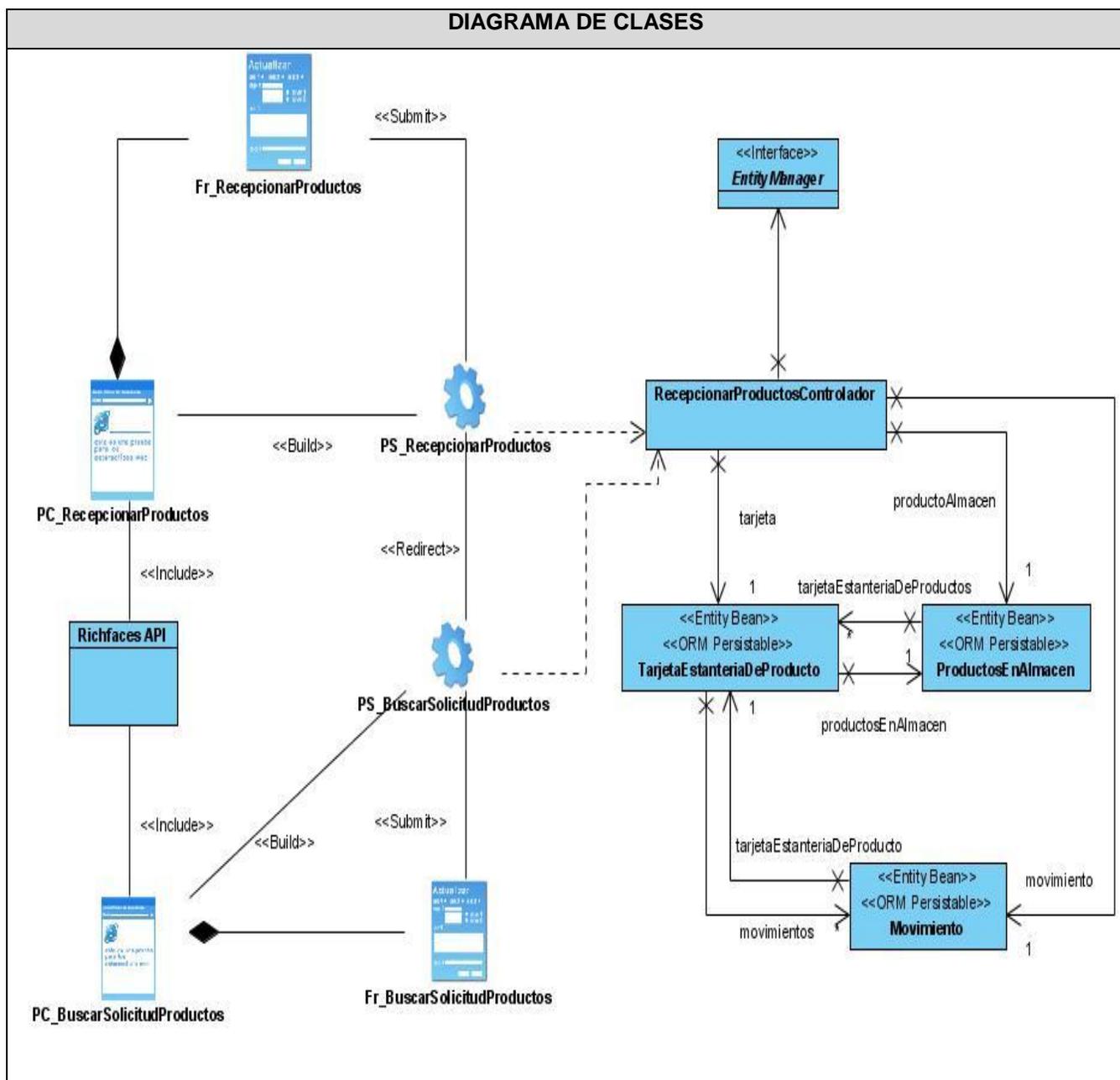


Diagrama 3-2 DCD Recepcionar productos

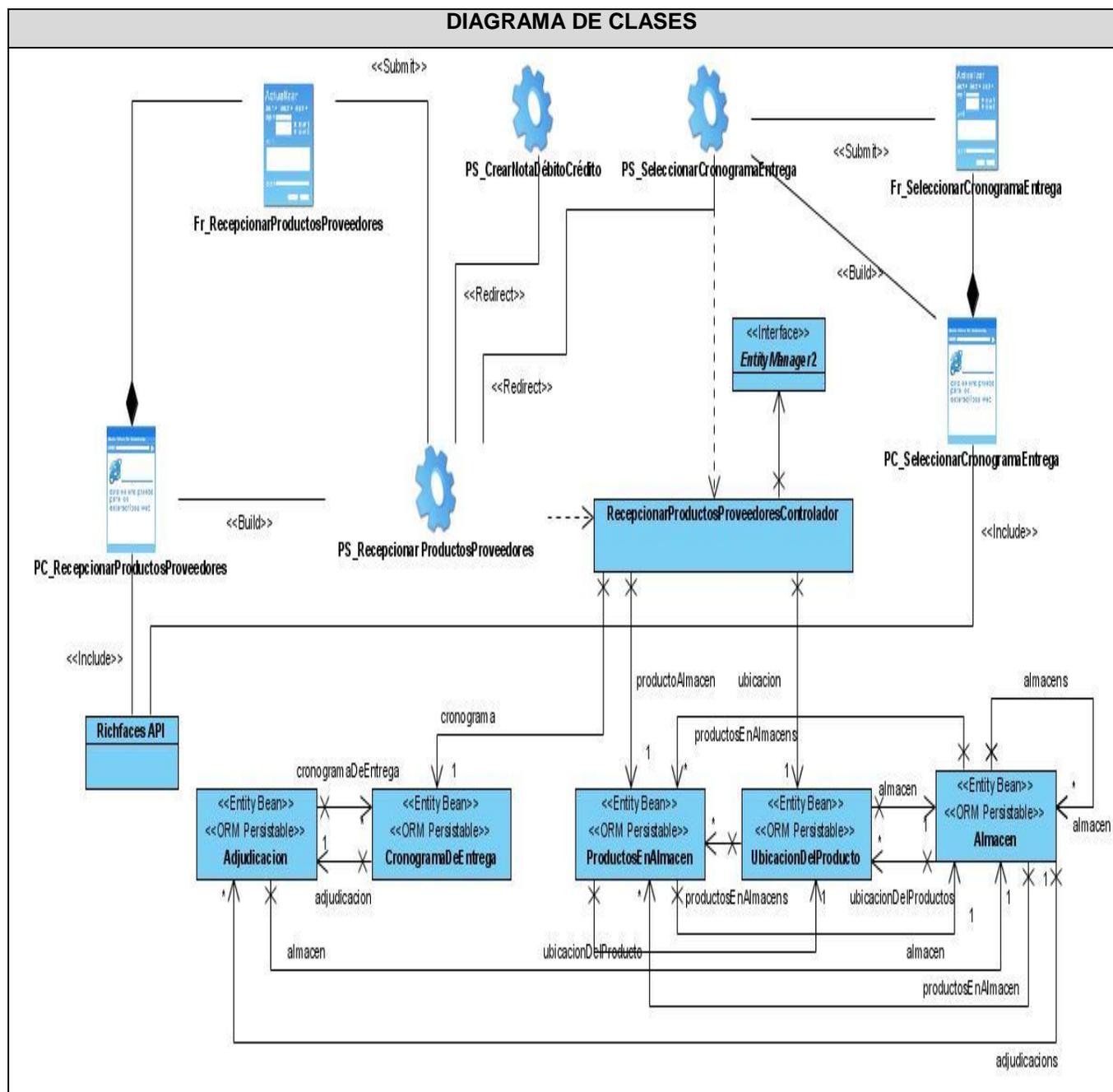


Diagrama 3-3 DCD Recepcionar productos de proveedores

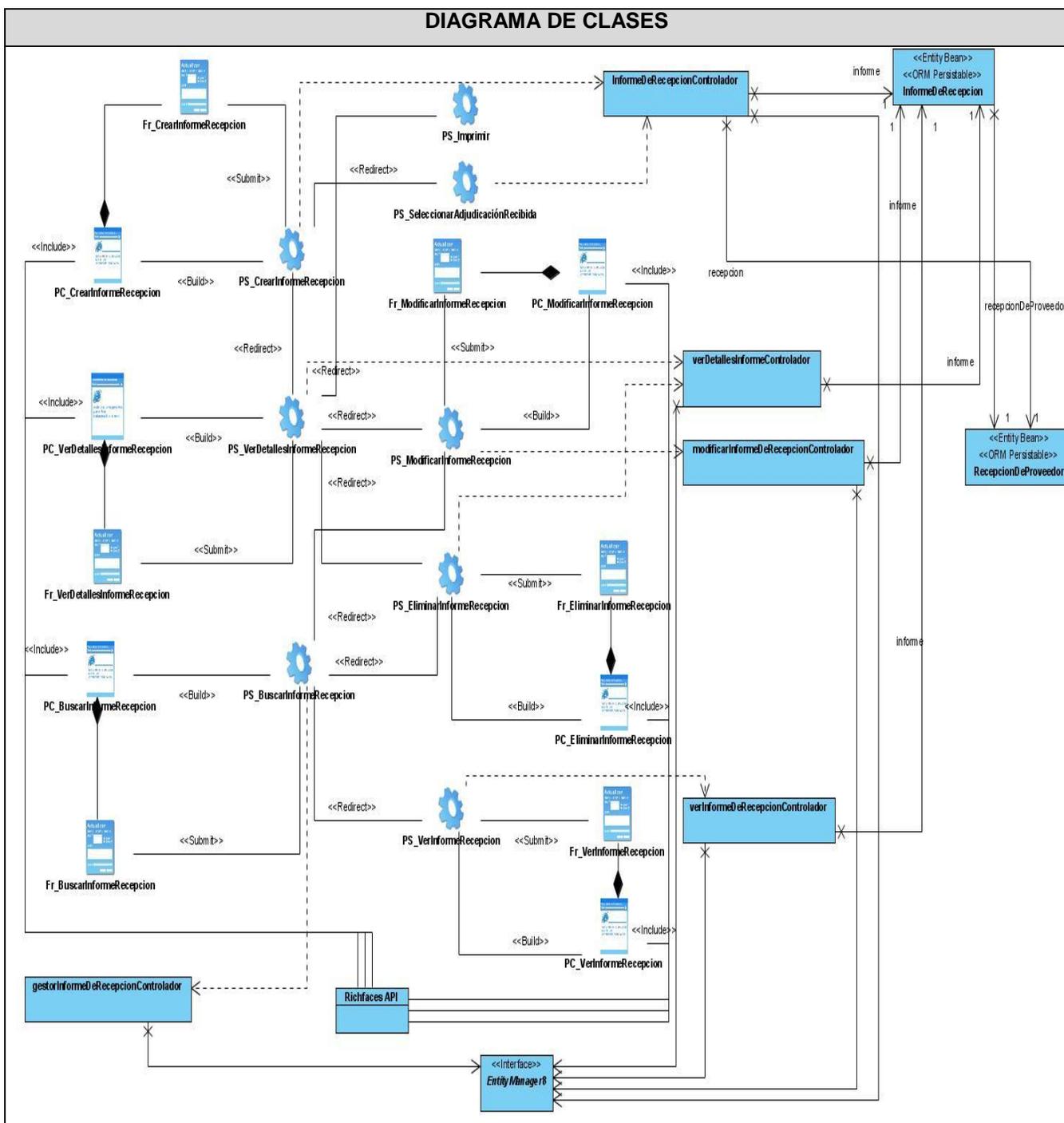


Diagrama 3-4 DCD Gestionar informe de recepción



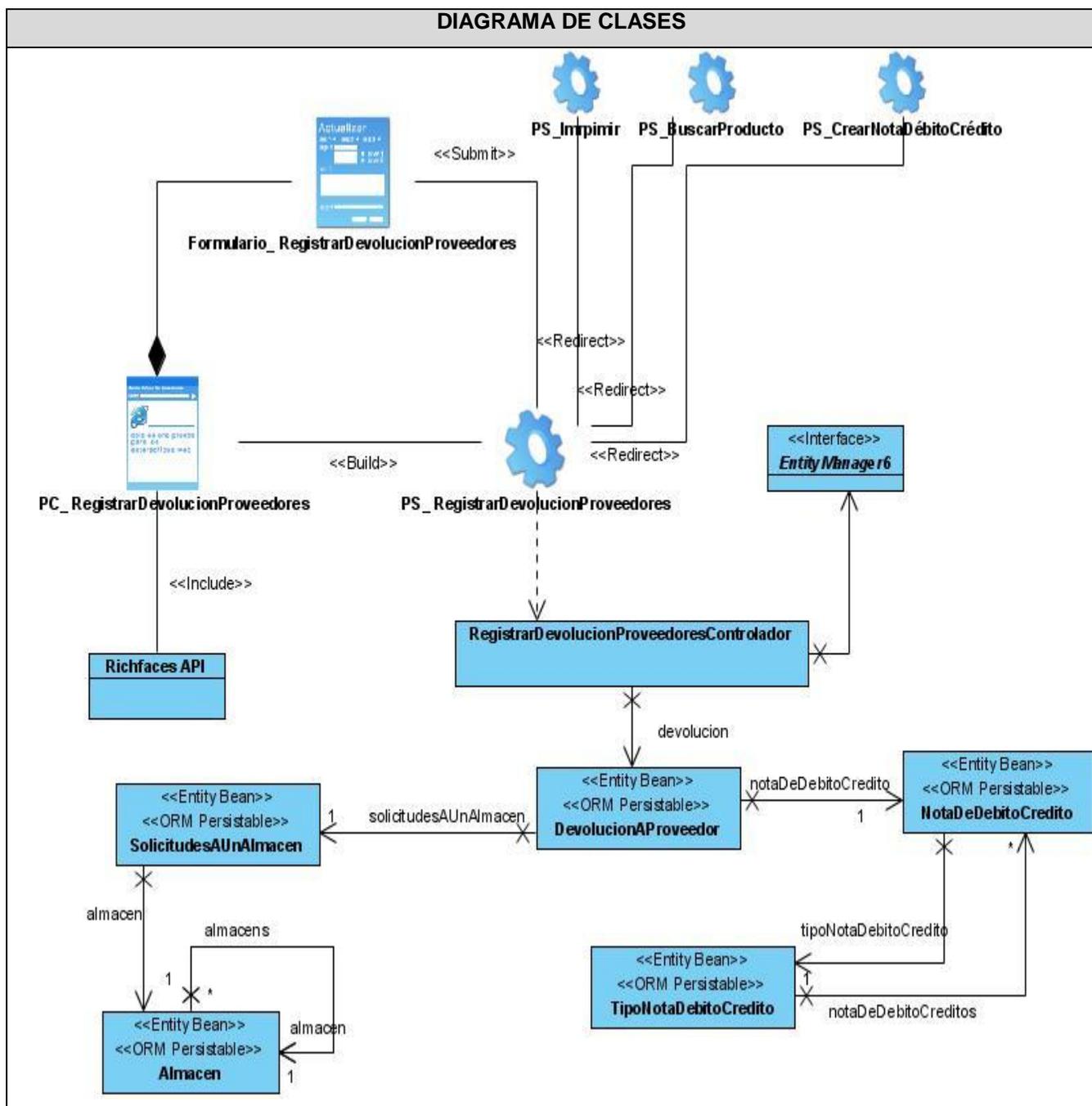


Diagrama 3-6 DCD Registrar devolución a proveedores

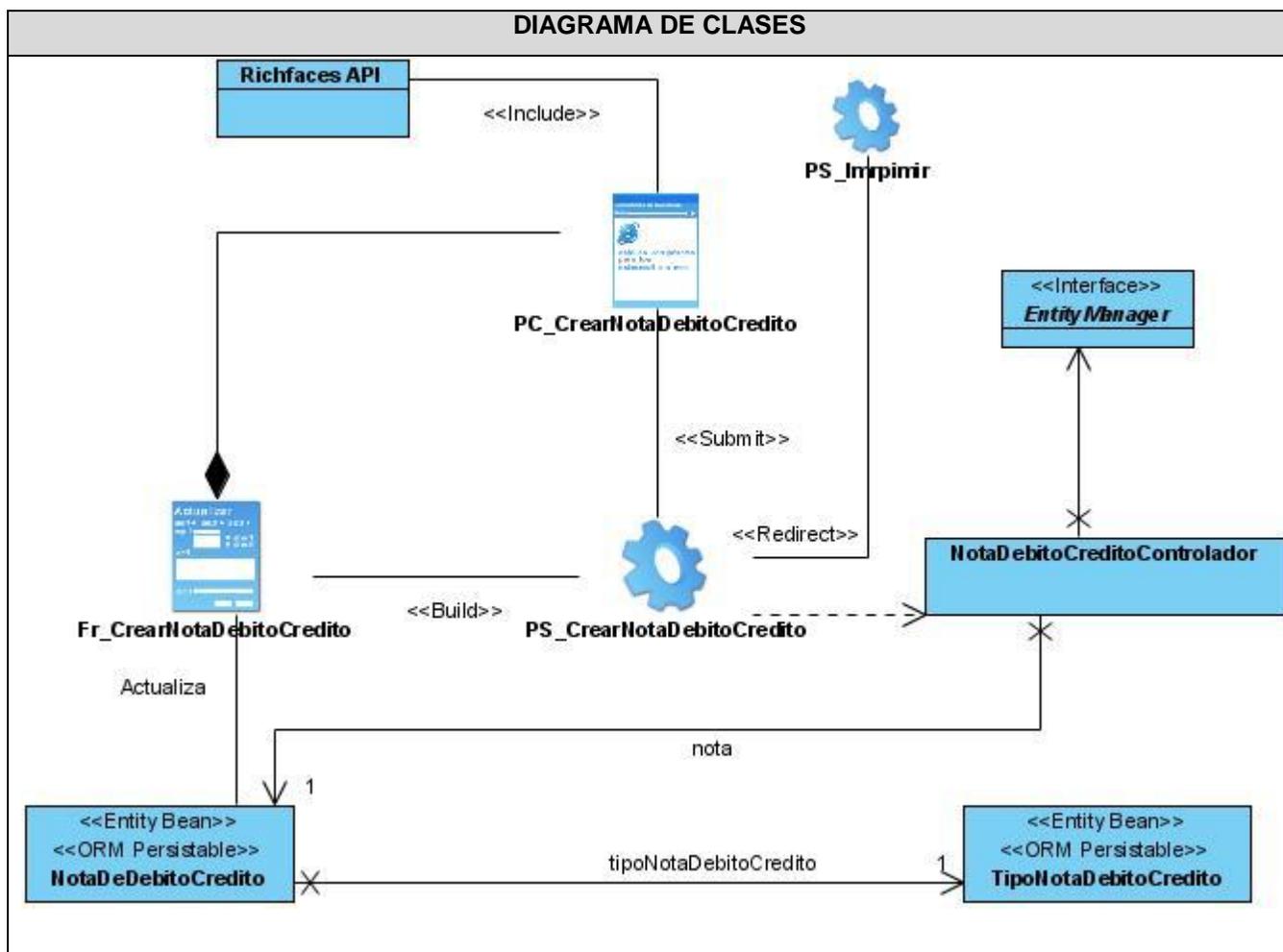


Diagrama 3-7 DCD Crear nota de débito o crédito

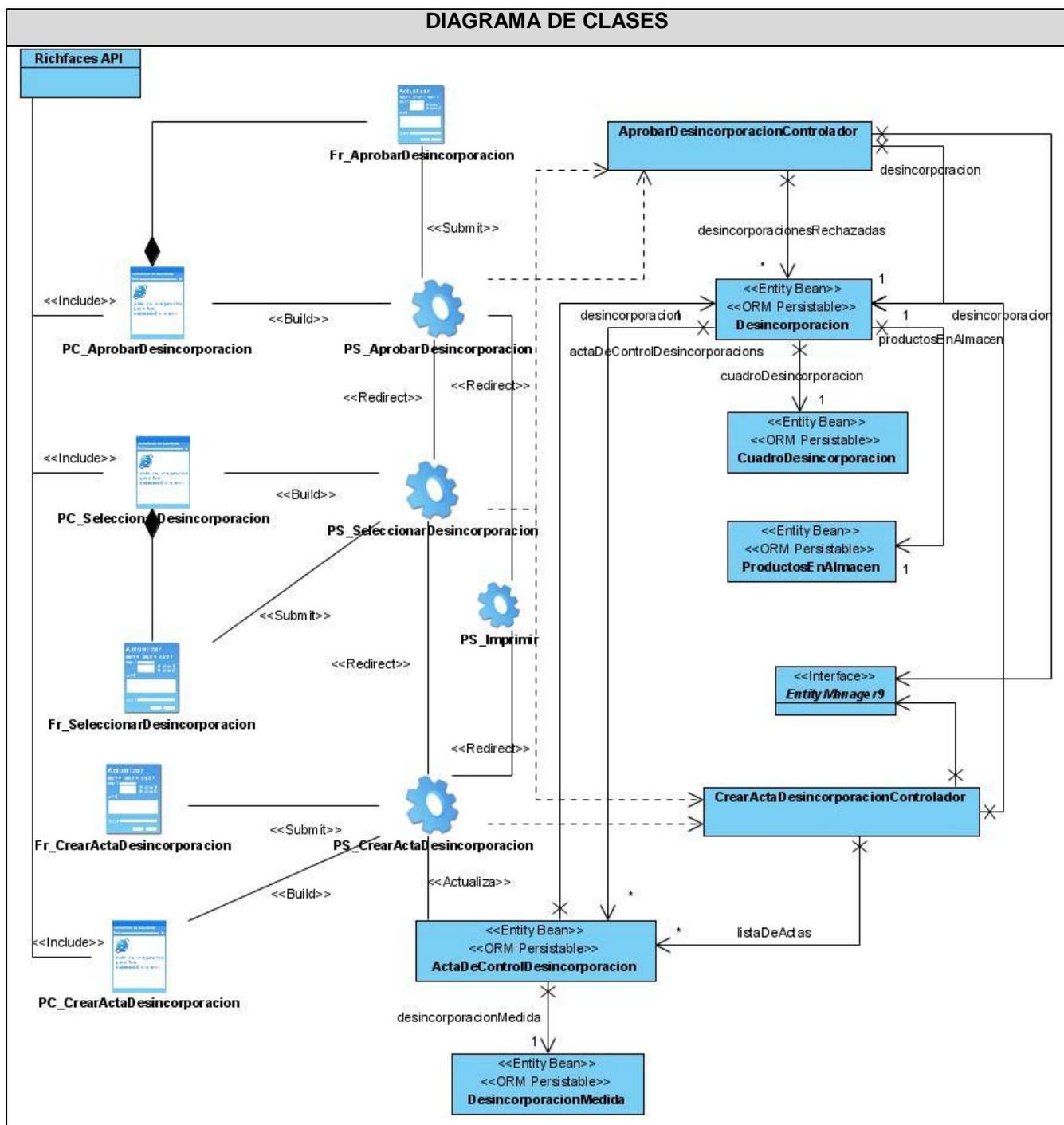


Diagrama 3-8 DCD Crear acta de control perceptivo de desincorporación

### **3.1.2.2 Diagramas de Interacción**

Muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos[21]. Se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos o para modelar un flujo de control particular de un caso de uso. Los diagramas de interacción no son solo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa.

- **Diagrama de secuencia**

Es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Se forma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, a lo largo del eje X. Normalmente, se coloca a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la derecha. A continuación, se colocan los mensajes que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo. Esto ofrece al lector una señal visual clara del flujo de control a lo largo del tiempo.

Los objetos poseen una línea de vida que es la línea discontinua vertical que representa la existencia de un objeto a lo largo de un período de tiempo y un foco de control que es un rectángulo delgado y estrecho que representa el período de tiempo durante el cual un objeto ejecuta una acción, bien sea directamente o a través de un procedimiento subordinado. La parte superior del rectángulo se alinea con el comienzo de la acción; la inferior se alinea con su terminación (y puede marcarse con un mensaje de retorno).

A continuación se muestran los diagramas de secuencia de algunos de los casos de uso más significativos del sistema.

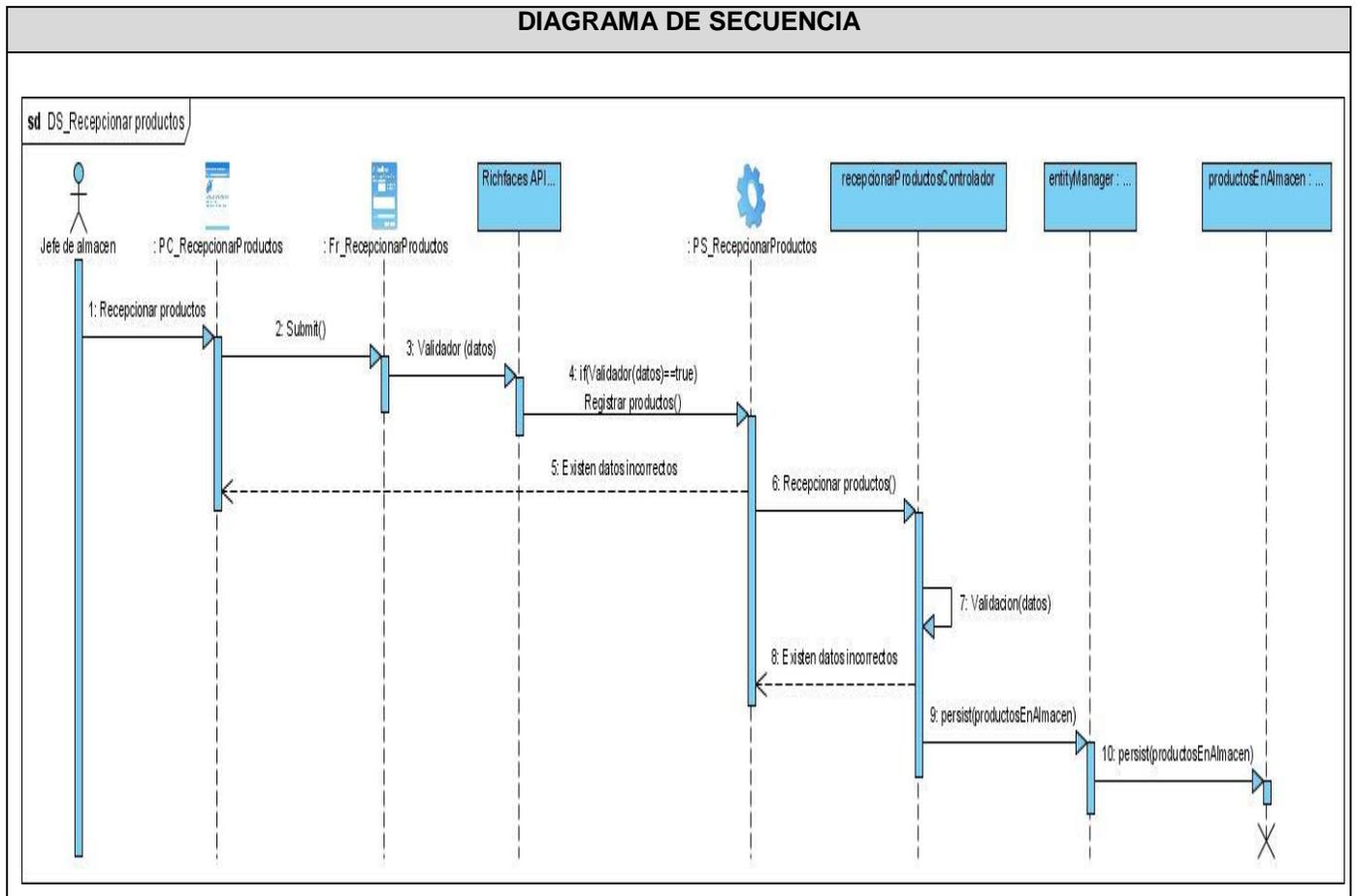


Diagrama 3-9 DS Recepcionar productos

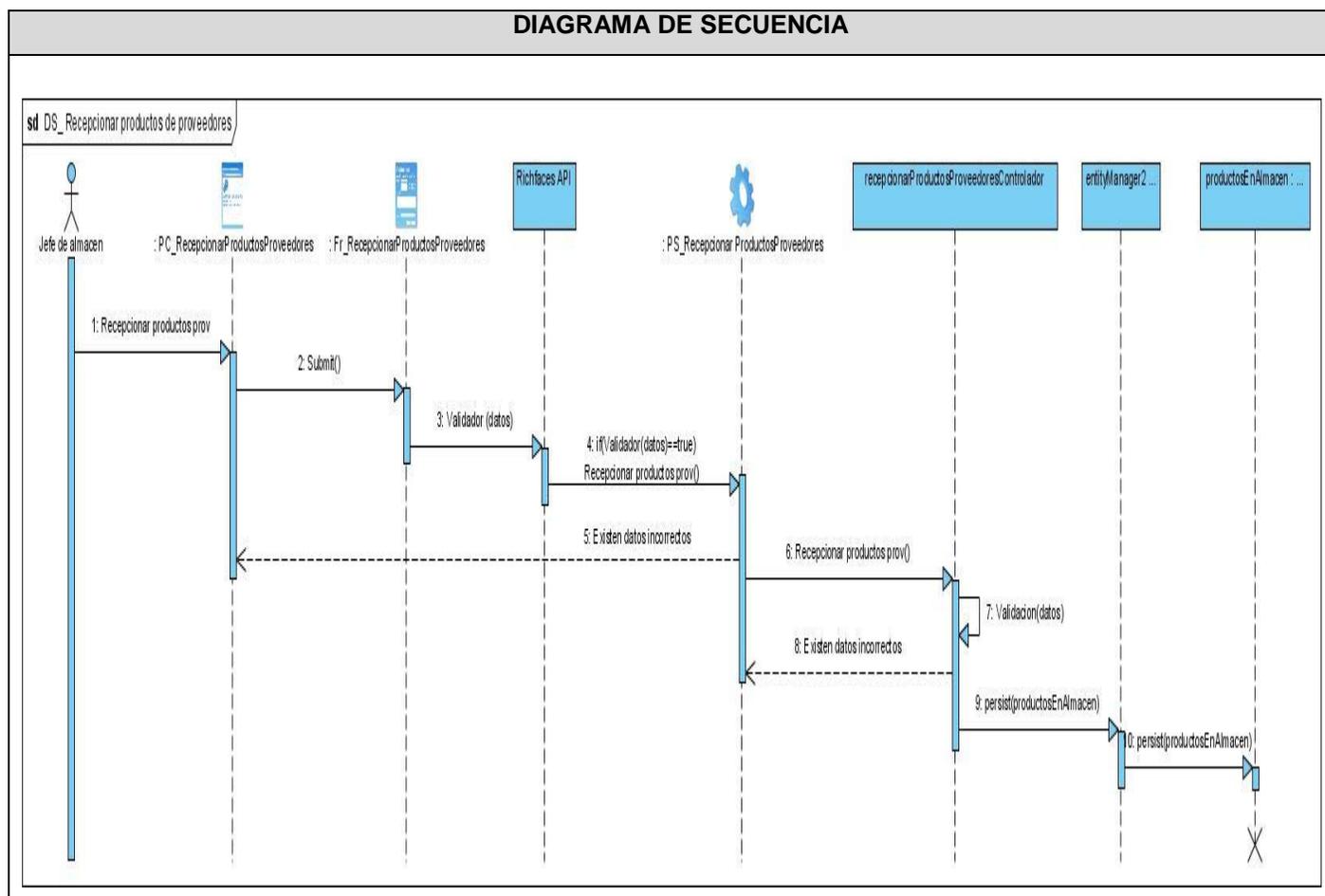


Diagrama 3-10 DS Recepcionar productos de proveedores

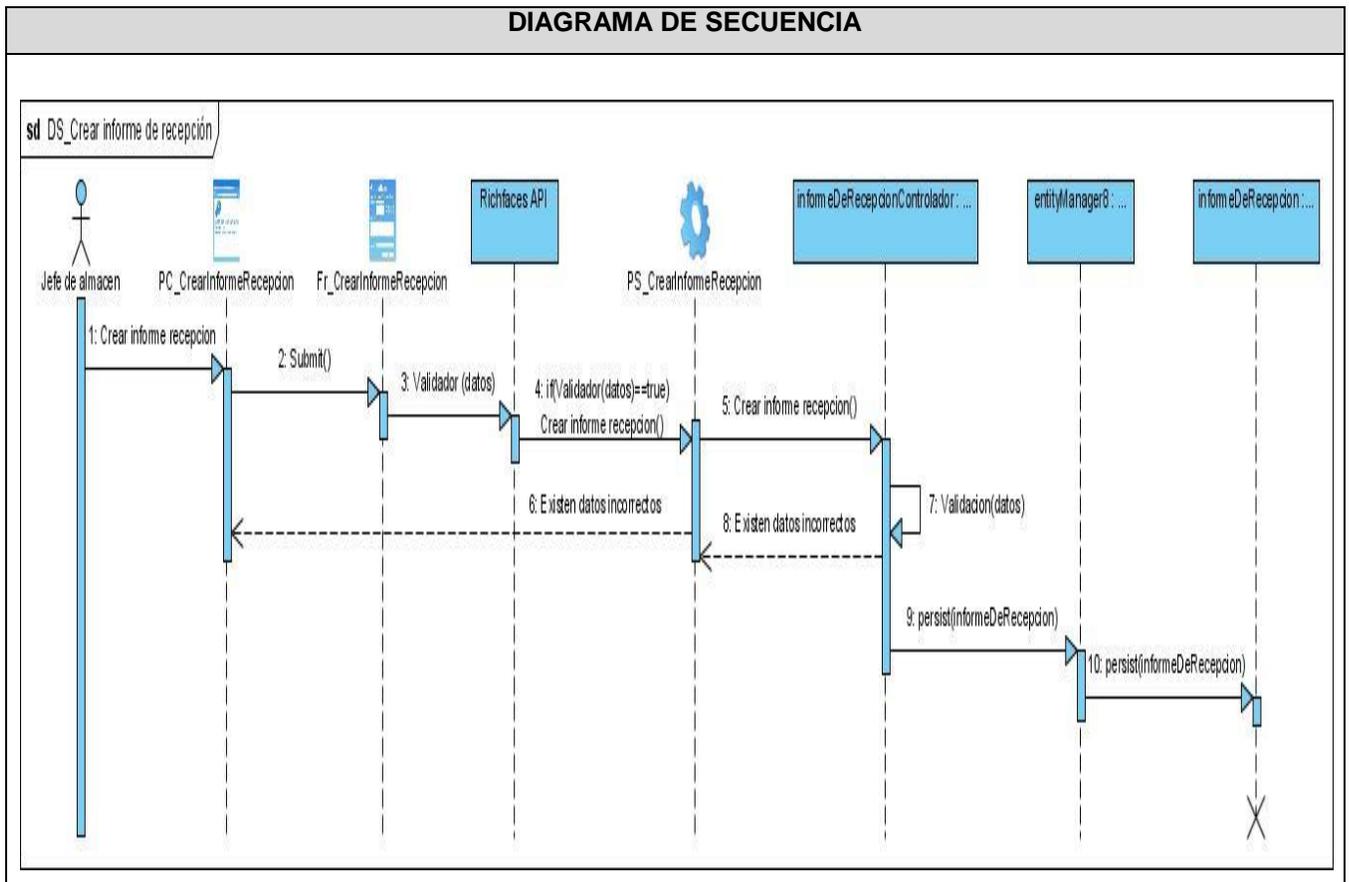


Diagrama 3-11 DS Crear informe de recepción

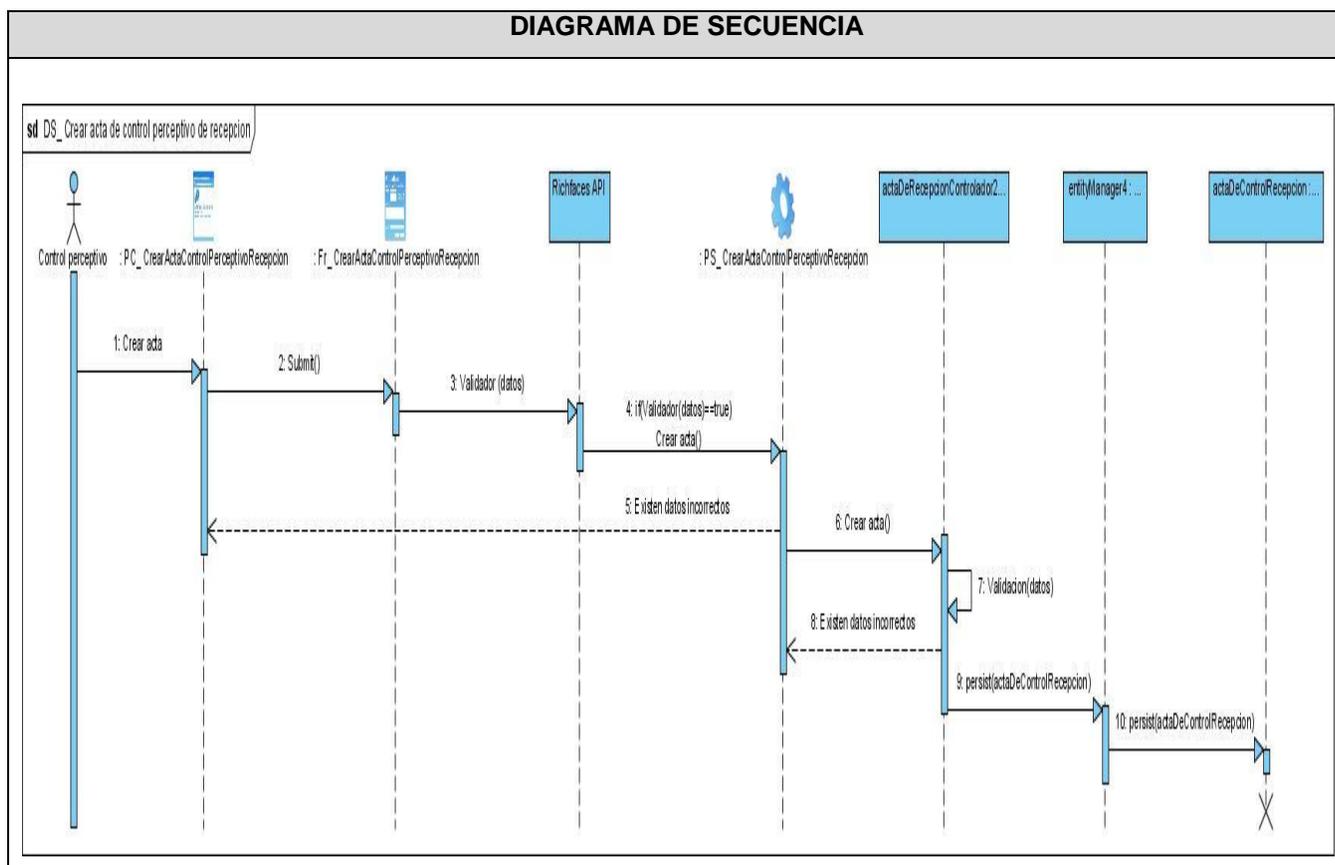


Diagrama 3-12 DS Crear acta de control perceptivo de recepción

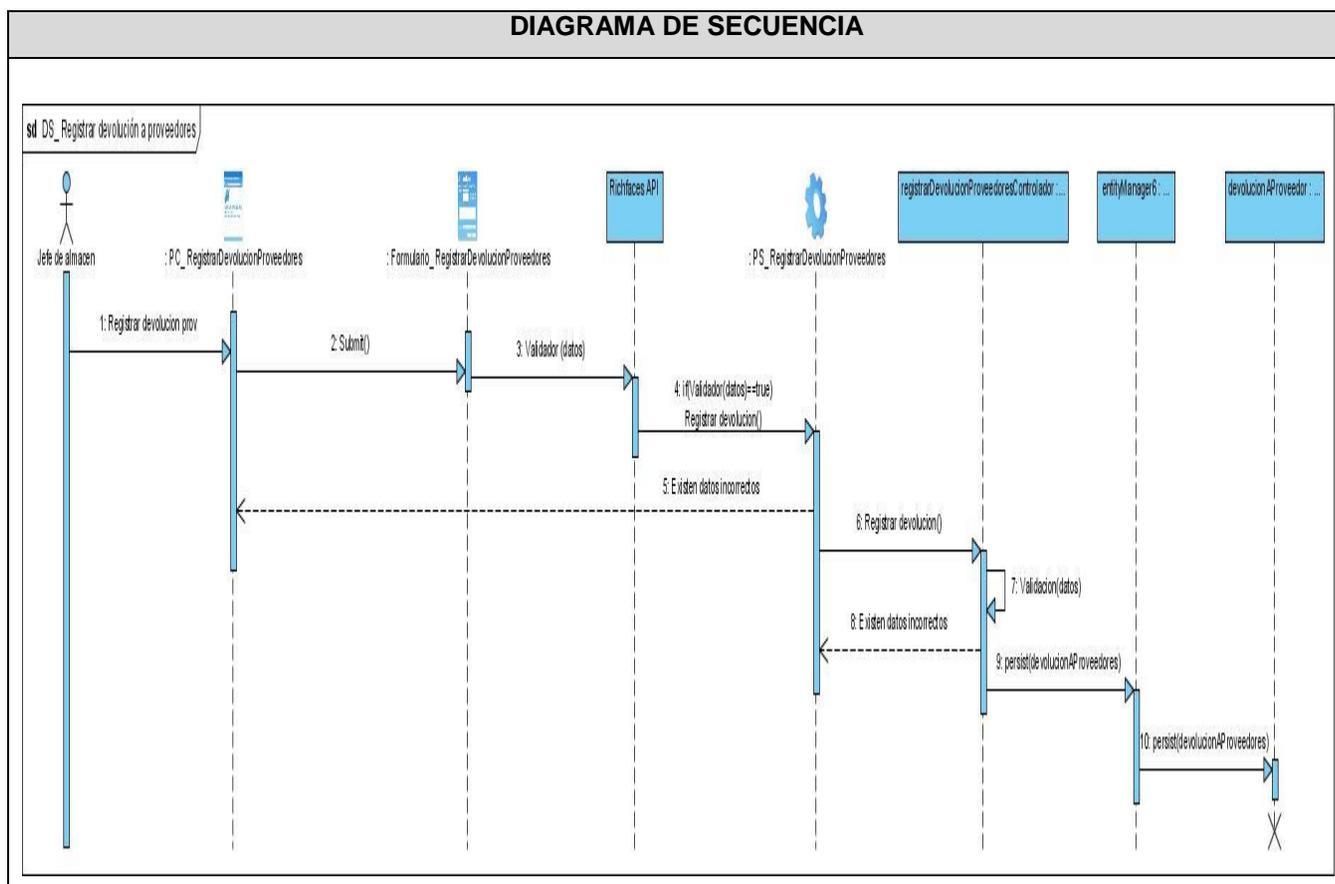


Diagrama 3-13 DS Registrar devolución a proveedores

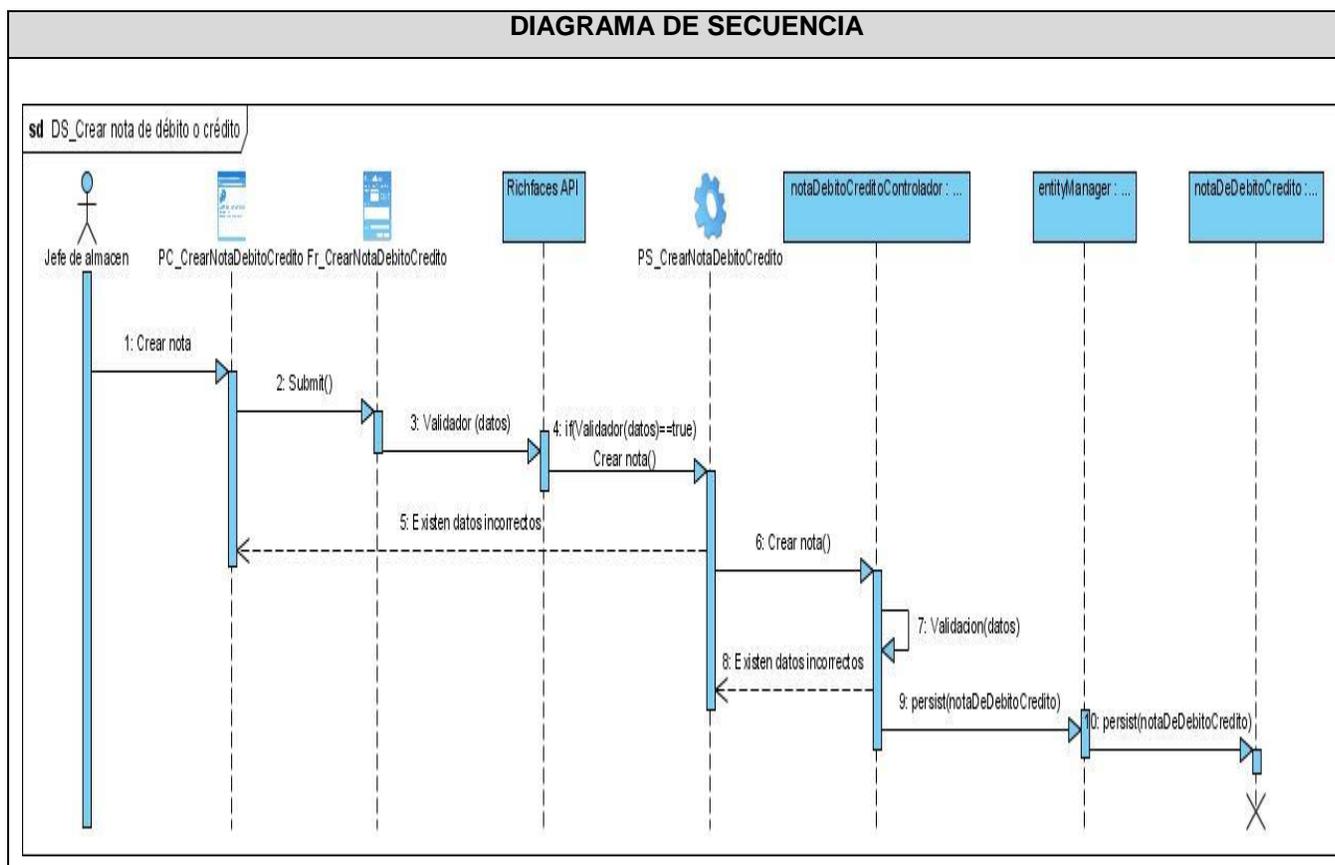


Diagrama 3-14 DS Crear nota de débito o crédito

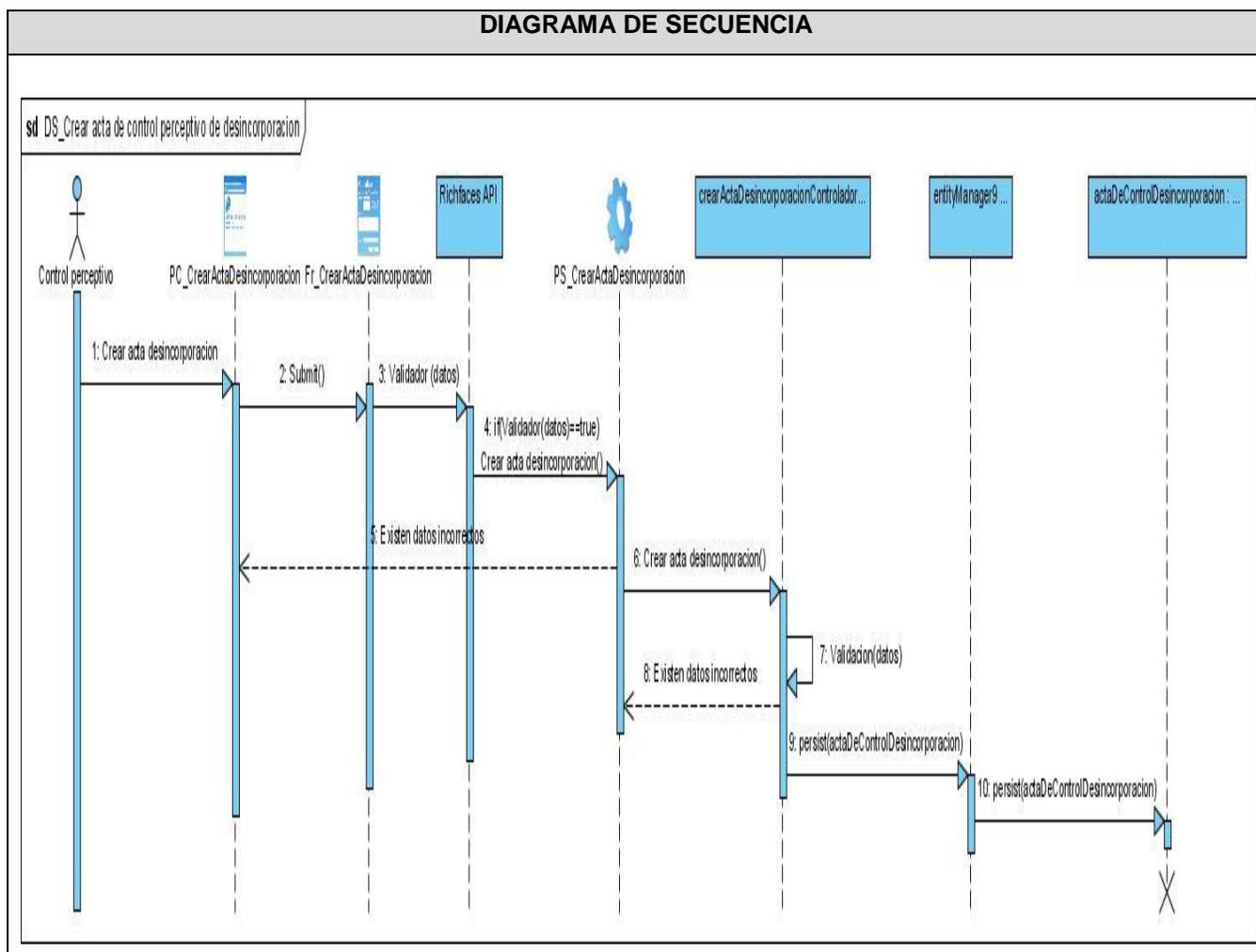


Diagrama 3-15 DS Crear acta de control perceptivo de desincorporación

### 3.2 Descripción de las clases del diseño

<b>Nombre:</b> RecepcionarProductosControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
cantidad	Integer
pu	Integer
stockMax	Integer
stockMin	Integer
descripcion	String
ubicacion	UbicacionDelProducto
producto	ProductosDeUnaSolicitud
activeAlmacenController	ActiveAlmacenController
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	recepcionarProducto
Descripción:	Se encarga de persistir el producto en la base de datos.
Nombre:	buscarMacheo
Descripción:	Busca si el producto ya ha sido recepcionado con anterioridad.
Nombre:	seleccSolicitud
Descripción:	Permite seleccionar una de las solicitudes para recepcionar sus productos.
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-16 CD RecepcionarProductosControlador

<b>Nombre:</b> RecepcionarProductosProveedoresControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
listadoDeProducto	List<ProductoRecepcionadoDeProveedor>
activeAlmacenController	ActiveAlmacenController
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	crearRecepcionDeProveedor
Descripción:	Se encarga de persistir el producto en la base de datos.
Nombre:	seleccionarAdjudicacion
Descripción:	Permite seleccionar una de las adjudicaciones para recepcionar sus productos.
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-17 CD RecepcionarProductosProveedoresControlador

<b>Nombre:</b> InformeDeRecepcionControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
listaInformes	List<InformeDeRecepcion>
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	crearInforme
Descripción:	Se encarga de persistir el informe en la base de datos.
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-18 CD InformeDeRecepcionControlador

<b>Nombre:</b> AprobarRecepcionControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
listaInformesRecepcion	List<InformeDeRecepcion>
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	aceptarRecepcion(InformeDeRecepcion informe)
Descripción:	Se encarga de aceptar una recepción, aceptando el informe de recepción de esta
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-19 CD AprobarRecepcionControlador

<b>Nombre:</b> ActaDeRecepcionControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
adjudicacionId	Integer
adjudicacion	Adjudicacion
actaDeControl	InformeDeRecepcion
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	crearInforme
Descripción:	Se encarga de crear el acta de control perceptivo
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-20 CD ActaDeRecepcionControlador

<b>Nombre:</b> RegistrarCuarentenaControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
listaCuarentena	List<Cuarentena>
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	crearCuarentenas
Descripción:	Se encarga de crear una nueva cuarentena
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-21 CD RegistrarCuarentenaControlador

<b>Nombre:</b> RegistrarDevolucionControlador	
<b>Tipo de clase:</b> Controlador	
<b>Atributo</b>	<b>Tipo</b>
listaDevoluciones	List<DevolucionAProveedor>
entityManager	EntityManager
<b>Para cada responsabilidad:</b>	
Nombre:	crearDevoluciones
Descripción:	Se encarga de crear una nueva devolución
Nombre:	Métodos de acceso(Propiedades)
Descripción:	Estos métodos son los que se encargan del encapsulamiento de cada atributo de la clase.

Tabla 3-22 CD RegistrarDevolucionControlador

<b>Nombre:</b> SP_RecepcionarProductos	
<b>Tipo de clase:</b> Página servidora	
<b>Descripción:</b>	Es una página que se ejecuta del lado del servidor la cual recibe y valida los datos que se envían desde la página cliente. Esta permite además, insertar los datos de los productos en un almacén determinado.

Tabla 3-23 SP\_RecepcionarProductos

<b>Nombre: Formulario_RecepcionarProductos</b>	
<b>Tipo de clase: Formulario</b>	
<b>Descripción:</b>	Está presente en la página cliente, permite el envío de los datos como criterios de búsqueda a la página servidora, la cual ejecuta la búsqueda y nuevamente el formulario muestra los resultados.

**Tabla 3-24 Formulario\_RecepcionarProductos**

<b>Nombre: PC_RecepcionarProductos</b>	
<b>Tipo de clase: Página cliente</b>	
<b>Descripción:</b>	Es una página que se ejecuta del lado del cliente. Permite al usuario introducir los criterios de búsqueda de los productos a recepcionar.

**Tabla 3-25 PC\_RecepcionarProductos**

En este capítulo se realizó una descripción de la arquitectura del sistema y la modelación del mismo en términos del diseño de los casos de uso del sistema, mostrando como resultado los diagramas de clases del diseño y los diagramas de secuencia. Además se describieron las clases del diseño, así como los atributos y métodos que deben tener las mismas para brindarle al desarrollador una idea clara de lo que debe implementar.

La realización de cada caso de uso es un artefacto muy importante que da entrada al flujo de trabajo de implementación, que tiene como principal objetivo crear componentes más específicos y asignarles a estos las clases del diseño correspondientes en dependencia de la trazabilidad definida.

## **Capítulo 4: Implementación**

El presente capítulo constituye la secuencia lógica del diseño, a través del mismo se implementan las clases y subsistemas en términos de componentes. Se expone el modelo de datos así como el diagrama de despliegue el cual muestra la distribución física de cada componente que se presentan más adelante con sus respectivas dependencias en el diagrama de componentes. También se explica cómo se lleva a cabo el tratamiento de errores, la seguridad del sistema, así como las estrategias de codificación utilizadas.

### **4.1 Modelo de datos**

Es usado para describir la estructura lógica y física de la información persistente manejada por el sistema. Controla que la información básica almacenada para cada atributo o propiedad de un concepto sea válida. Está formado por objetos (entidades que existen y que se manipulan), atributos (características básicas de estos objetos) y relaciones (forma en que enlazan los distintos objetos entre sí).

A continuación se muestra el modelo de datos del sistema:

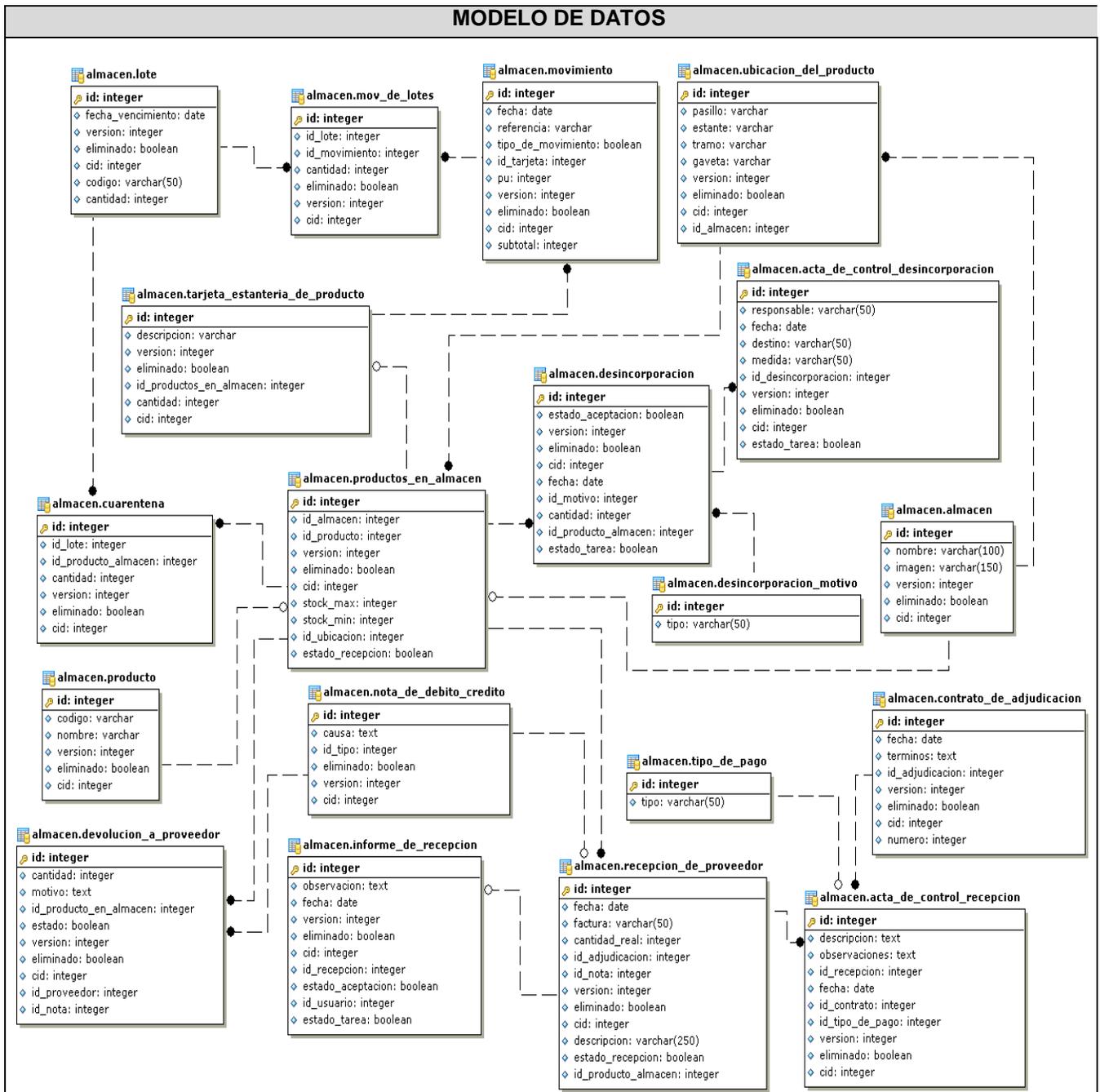


Diagrama 4-1 Modelo de datos

4.1.1 Descripción de las tablas

<b>Nombre: productos_en_almacen</b>		
<b>Descripción:</b> Almacenamiento de todos los productos que son recepcionados procedentes de cualquier vía en el almacén.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
id_almacen	integer	Identificador del almacén.
id_producto	integer	Identificador del producto.
stock_max	integer	Cantidad máxima del producto permitida.
stock_min	integer	Cantidad mínima del producto permitida.
id_ubicacion	integer	Identificador de la ubicación del producto dentro del almacén.
estado_recepcion	boolean	Posibilita la recepción física del producto, incluyendo la cantidad de este en la tarjeta de estantería.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-2 Productos en almacén

<b>Nombre: almacen</b>		
<b>Descripción:</b> se dispone de todos los almacenes con que cuenta el sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
almacen	integer	Nombre del almacén.
imagen	varchar	Nombre de la imagen que identifica el almacén en el sistema.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-3 Almacén

<b>Nombre: producto</b>		
<b>Descripción:</b> Nomenclador que dispone de todos los productos con que cuenta el sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
código	varchar	Código del producto.
nombre	varchar	Nombre del producto que lo identifica.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-4 Producto

<b>Nombre: ubicación_del_producto</b>		
<b>Descripción:</b> dispone de todas las ubicaciones que existen en un almacén.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
pasillo	varchar	Pasillo de la dirección física del producto.
estante	varchar	Estante de la dirección física del producto.
tramo	varchar	Tramo de la dirección física del producto.
gaveta	varchar	Gaveta de la dirección física del producto.
Id_almacen	ineteger	Identificador del almacén al que pertenece la ubicación.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-5 Ubicación del producto

<b>Nombre: tarjeta_estanteria_de_producto</b>		
<b>Descripción:</b> cada producto del que dispone un almacén presenta una tarjeta de estantería, en la cual se encuentra la cantidad física existente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
cantidad	integer	Cantidad física del producto en almacén.
descripción	varchar	Descripción del producto.
id_producto_en_almacen	integer	Identificador del producto en almacén.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-6 Tarjeta de estantería del producto

Nombre: tipo_de_pago		
<b>Descripción:</b> nomenclador que presenta los tipos de pagos que pueden ser efectuados en una compra.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
tipo	varchar	Tipo de pago.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-7 Tipo de pago

Nombre: movimiento		
<b>Descripción:</b> controla cada movimiento que se realiza a un producto, ya sea de entrada o salida, controlando la cantidad que se mueve en cualquiera de los casos.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación del movimiento.
referencia	varchar	Procedencia o destino del movimiento.
tipo_de_movimiento	boolean	Especifica si es de entrada o salida.
pu	integer	Precio unitario del producto.
subtotal	integer	Cantidad que es movida.
id_tarjeta	integer	Identificador de la tarjeta de estantería.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-8 Movimiento

<b>Nombre: lote</b>		
<b>Descripción:</b> agrupa los tipos de productos existentes de un almacén por fechas de vencimiento y tipos de productos.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
codigo	varchar	Código del lote.
cantidad	integer	Cantidad física de producto en el lote.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-9 Lote

<b>Nombre: mov_de_lotes</b>		
<b>Descripción:</b> controla cada movimiento realizado a un producto procedente de un lote, registrándose la cantidad que es movida.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
cantidad	integer	Cantidad movida del lote.
id_lote	integer	Identificador del lote.
id_movimiento	integer	Identificador del movimiento.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-10 Movimiento de lotes

<b>Nombre: desincorporación</b>		
<b>Descripción:</b> se registran todos los datos necesarios de los productos en almacén que son necesarios retirar por deterioro, pérdida o rompimiento.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
cantidad	integer	Cantidad del producto que se desincorpora.
estado_aceptacion	boolean	Aceptación o rechazo de la desincorporación, la realiza el administrador del hospital.
estado_tarea	boolean	Estado de la tarea que permite administrar el flujo del proceso.
id_motivo	integer	Identificador del motivo por el que se desincorpora.
id_producto_en_almacen	integer	Identificador del producto en almacén que se desincorpora.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-11 Desincorporación

<b>Nombre: desincorporacion_motivo</b>		
<b>Descripción:</b> nomenclador que presenta todos los tipos de motivos que presenta una desincorporación como deterioro, rompimiento o robo.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
motivo	varchar	Tipo de motivo.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-12 Desincorporación motivo

<b>Nombre: nota_de_debito_credito</b>		
<b>Descripción:</b> se crear opcionalmente para expresar la causa de una recepción o una devolución.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
causa	text	Causa de creación.
id_motivo	integer	Identificador del motivo por el que se crea la nota.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-13 Nota de débito o crédito

<b>Nombre: tipo_nota_debito_credito</b>		
<b>Descripción:</b> nomenclador que presenta los tipos de notas que pueden ser creadas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
tipo	varchar	Tipo de nota.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-14 Tipo de nota

<b>Nombre: devolucion_a_proveedor</b>		
<b>Descripción:</b> contiene los datos de las devoluciones que son necesarias realizar a un proveedor, ya sea por faltante o sobrante de la cantidad solicitada del producto.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
cantidad	integer	Cantidad que se entregará al proveedor.
motivo	varchar	Motivo por el que se realiza la devolución.
estado	boolean	Estado de aceptación o rechazo de la devolución.
id_nota	integer	Identificador de la nota de débito o crédito.
id_proveedor	integer	Identificador del proveedor al que se dirige la devolución.
id_producto_en_almacen	integer	Identificador del producto en almacén.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-15 Devolución a proveedor

<b>Nombre: recepcion_de_proveedor</b>		
<b>Descripción:</b> presenta los datos necesarios de una recepción de productos, que pertenecen a un proveedor específico.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
factura	varchar	Factura de la orden de compra realizada al proveedor.
cantidad_real	varchar	Cantidad recepcionada.
descripcion	varchar	Descripción del producto.
estado_recepcion	boolean	Estado de la recepción que ejecuta el departamento de control perceptivo.
id_nota	integer	Identificador de una nota de débito o crédito de ser necesaria su creación.
id_adjudicacion	integer	Identificador de la adjudicación realizada al proveedor.
id_producto_en_almacen	integer	Identificador del producto en almacén.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-16 Rrecepción de proveedor

<b>Nombre: informe_de_recepcion</b>		
<b>Descripción:</b> Se crea con el objetivo de especificar algunas observaciones identificadas por el jefe del almacén al realizar una recepción de productos procedentes de un proveedor.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
observaciones	varchar	Observaciones identificadas en la recepción del producto.
estado_aceptacion	boolean	Estado de aceptación o rechazo del informe.
estado_tarea	boolean	Estado de la tarea que permite administrar el flujo del proceso.
id_usuario	integer	Identificador del usuario del sistema que crea el informe.
id_recepcion	integer	Identificador de la recepción perteneciente al informe.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-17 Informe de recepción

<b>Nombre: acta_de_control_recepcion</b>		
<b>Descripción:</b> presenta los datos necesarios de un acta de control, que es creada por parte del departamento de control perceptivo donde se especifican las observaciones finales de la recepción aprobando el informe de recepción y dando como recepcionado el producto procedente del proveedor.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
descripcion	varchar	Descripción del resumen final de la recepción por parte del departamento de control perceptivo.
observaciones	varchar	Observaciones finales de la recepción por parte del departamento de control perceptivo.
id_contrato	integer	Identificador del contrato que se establece al proveedor.
id_recepcion	integer	Identificador de la recepción a la que pertenece el acta de control.
id_tipo_de_pago	integer	Identificador del tipo de pago que se establece en la orden de compra.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-18 Acta de control de recepción

<b>Nombre: contrato_de_adjudicacion</b>		
<b>Descripción:</b> se registran los datos de un contrato creado a los proveedores, por la necesidad de realizar compras.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
terminos	varchar	Términos del contrato.
numero	varchar	Número del contrato.
id_adjudicacion	integer	Identificador de la adjudicación que se realiza al proveedor.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-19 Contrato de adjudicación

<b>Nombre: cuarentena</b>		
<b>Descripción:</b> se registran todos los productos que son retirados del almacén por estar próximos a vencer.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	integer	Identificador de la tabla.
cantidad	integer	Cantidad que se pondrá en cuarentena.
id_lote	integer	Identificador del lote al que pertenece el producto.
id_producto_en_almacen	integer	Identificador del producto en almacén.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-20 Cuarentena

Nombre: <b>acta_de_control_desincorporacion</b>		
<b>Descripción:</b> se registran todos los datos necesarios por parte del departamento de control perceptivo, encargado de controlar cada desincorporación.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
fecha	date	Fecha de creación.
responsable	varchar	Responsable de llevar a cabo la desincorporación.
medida	varchar	Medida que se toma con los productos que se desincorporan.
destino	varchar	Destino final de los productos que se desincorporan.
estado_tarea	boolean	Estado de la tarea que permite administrar el flujo del proceso.
id_desincorporacion	integer	Identificador de la desincorporación perteneciente al acta de desincorporación.
version	integer	Versión del sistema.
eliminado	boolean	Estado de disponibilidad de la entidad en la base de datos.
cid	integer	Bitácora.

Tabla 4-21 Acta de control de desincorporación

## 4.2 Modelo de implementación

La implementación es el flujo de trabajo sobre el cual se realiza mayor esfuerzo durante las iteraciones de construcción en la metodología RUP, aunque también se lleva a cabo el trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura, y durante la fase de transición, para tratar defectos tardíos como los encontrados en los primeros lanzamientos del sistema.

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de

despliegue. Muestra las dependencias entre la estructura del sistema en ejecución (diagrama de despliegue) y las partes del código del sistema (diagrama de componentes).

#### 4.2.1 Diagrama de despliegue

Es un modelo de objeto que indica la situación física de los componentes lógicos desarrollados y muestra las relaciones entre los componentes hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación donde cada nodo representa una estación de trabajo, dispositivo o procesador.

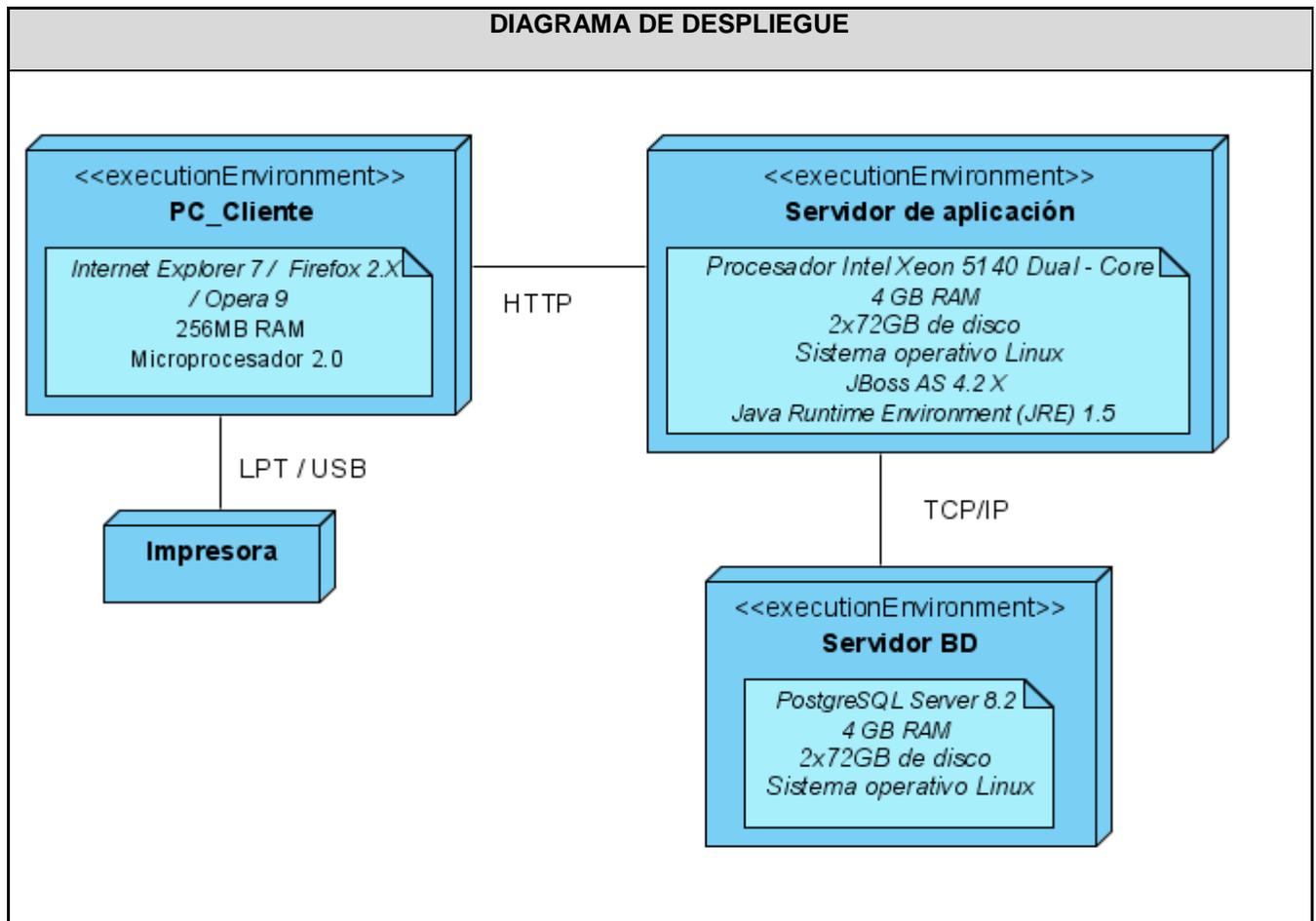


Diagrama 4-22 Diagrama de despliegue

#### **4.2.2 Diagrama de componentes**

Un componente es una parte física y reemplazable de un sistema que está conformado por un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces; poseen relaciones de traza con los elementos del modelo de diseño que implementan.

Un diagrama de componentes muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean estos de código fuente, librerías, binarios o ejecutables. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar el código y organizando los subsistemas de implementación en capas.

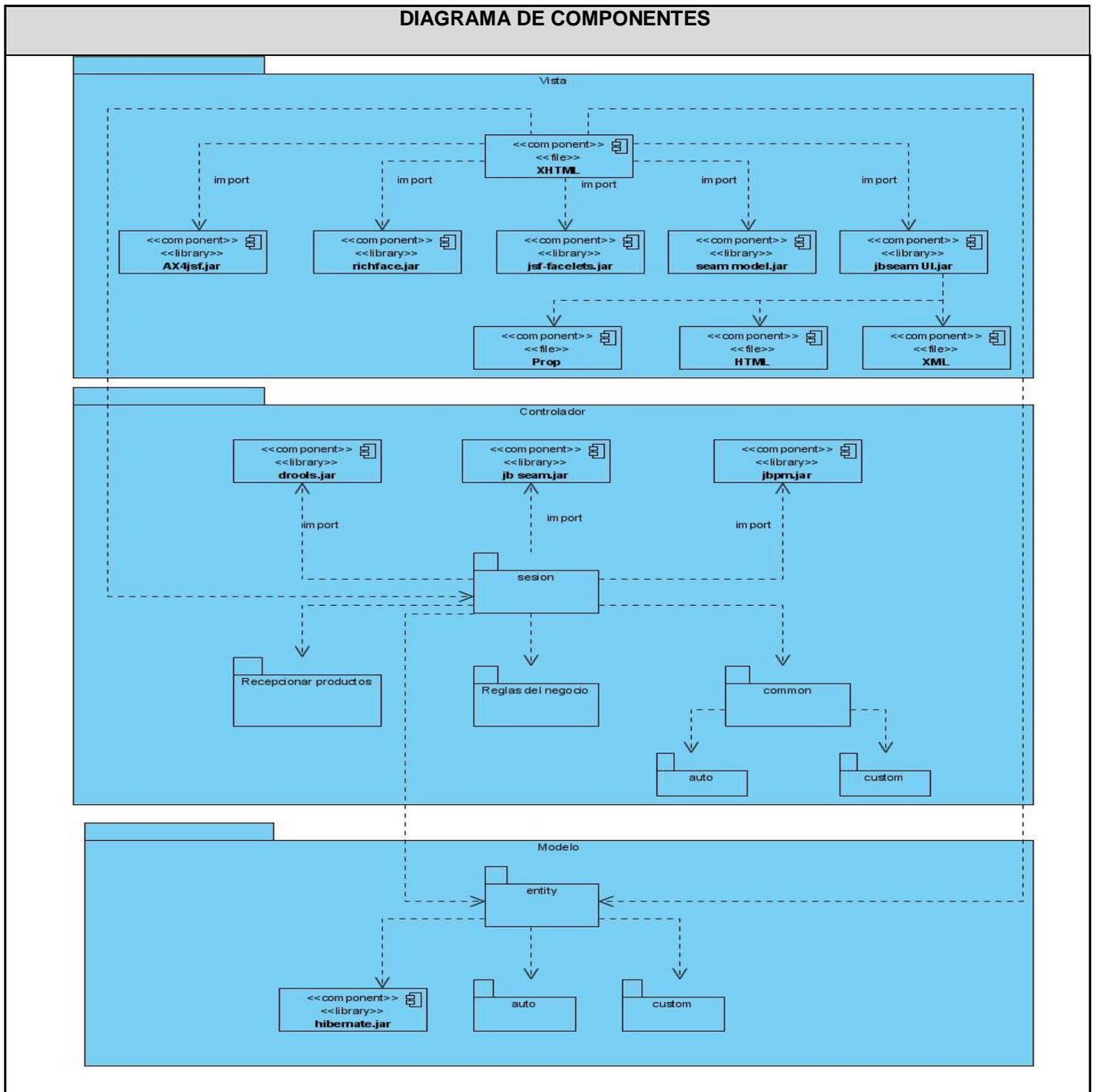


Diagrama 4-23 Diagrama de componentes

### 4.3 Tratamiento de errores

Las excepciones son un mecanismo de los lenguajes de programación que permiten definir acciones a realizar en caso de producirse situaciones anómalas (excepción) producida durante la ejecución del programa. Específicamente en el lenguaje de programación Java existen dos vías para lanzar excepciones, de forma implícita y explícita.

Se lanza una excepción implícita cuando el programa ejecuta una acción ilegal o no válida en el sistema como es la división por cero, estos tipos de excepciones son llamadas excepciones *runtime*. Las excepciones implícitas son aquellas que el desarrollador define para controlar algún error que espera que ocurra, para posteriormente controlar esta y que no interrumpa el funcionamiento normal del sistema.

En el sistema propuesto se utilizan todas las bondades que brinda la plataforma para el tratamiento de excepciones. Para cada porción de código donde se espere una situación anómala, se definen las excepciones a fines para luego ser tratadas evitando la interrupción del sistema. También se emplean un conjunto de tipos de excepciones predefinidas por los marcos de trabajos que se utilizan en el sistema.

Las diferentes tecnologías utilizadas y la integración armónica que existe entre ellas, permiten capturar y controlar posibles situaciones desde diferentes puntos del sistema. En las páginas clientes se cuenta con un conjunto de componentes denominados validadores, que permiten establecer tipos de datos y formatos controlando que el envío de los activos al servidor sean los esperados.

Además el marco de trabajo Seam brinda un potente conjunto de excepciones predefinidas, que conjuntamente con la clase *FacesMessages*, permite tratar estas situaciones desde los controladores correspondientes y mostrar mediante la clase antes mencionada, los resultados del tratamiento. Seam permite además mediante el fichero de configuración *page.xml*, todo un flujo de navegación basado en excepciones.

### 4.4 Seguridad

Debido a la importancia de los procesos que se llevan a cabo en el almacén hospitalario para el correcto funcionamiento del hospital, la necesidad de que estos sean realizados con la calidad requerida y que el acceso a la información de los productos allí almacenados solo sea posible para las personas que se encuentren acreditadas, provoca que se tengan en cuenta una serie de requisitos de seguridad de modo que no se pueda afectar el funcionamiento del sistema.

La seguridad se encuentra en todas las capas de la aplicación. En este caso toda la autorización a directorios, páginas, controles, opciones del menú, servicios del negocio, está basado en reglas, esto permite que ninguna de estas “reglas del negocio” esté ligada al código fuente en la aplicación y que el cambio de alguna de estas no requiera modificación alguna en el código, sólo en la definición del fichero de configuración.

Los requisitos de seguridad que se tuvieron en cuenta fueron:

- Se mantendrá la seguridad y control a nivel de usuarios y contraseñas, garantizando el acceso de los mismos solo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas solo podrán ser cambiadas por el propio usuario o por el administrador del sistema.
- El sistema permitirá registrar las acciones que se realizan por los usuarios en todo momento.
- El sistema posibilitará administrar la seguridad a través del otorgamiento de permisos para el acceso a las funcionalidades de los módulos a roles y usuarios definidos.
- El sistema posibilitará configurar las funcionalidades permitiendo una vez seleccionado un módulo adicionar o eliminar una categoría o funcionalidad.

Independientemente de las políticas de seguridad propias del sistema, estas no son efectivas si no existe organización en el manejo administrativo de la documentación y la autodeterminación en los hospitales, que garantice un uso correcto del sistema informático por las personas encargadas de operarlo.

La institución o instituciones correspondientes, en este caso el ministerio y los hospitales, deberán incorporar personal confiable para el manejo del sistema informático y garantizar el respeto por las políticas de acceso que propone el sistema.

### **4.5 Estrategias de codificación. Estándares y estilos a utilizar**

Los estándares de codificación son reglas específicas de cada lenguaje de programación cuyo cumplimiento reduce de forma significativa el riesgo de que los desarrolladores introduzcan errores.

El uso de los estándares tiene innumerables ventajas, entre ellas:

- Asegurar la legibilidad del código entre distintos programadores, facilitando el debugging del mismo.

- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y, posteriormente, se efectúan revisiones del código de rutina, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener. Para la implementación del sistema propuesto se utilizaron varios estándares de codificación como son:

**\*Notación PascalCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Ejemplo: NotacionPascalCasing.

**\*\*Notación CamellCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. Ejemplo: notacionCamelCasing.

A continuación se muestran las restricciones de nomenclatura y estándares de codificación que fueron utilizados basándose en los estándares antes mencionados:

<b>Indentación</b>	
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.	
Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.
Aspectos generales	<p>El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla.</p> <p>Los inicios ( { ) y cierre ( } ) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.</p> <p>Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.</p>

Tabla 4-24 Indentación

<b>Variables y constantes</b>		
Apariencia de variables	Las variables tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, la cual identificará el tipo de datos al que se refiere (ver tabla 4-29), en caso de que sea un nombre compuesto se empleará notación CamellCasing**. Ejemplo: sNombrePaciente.
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Aspectos generales	Nombres de las variables y constantes	El nombre empleado, debe permitir que con solo leerlo se conozca el propósito de la misma.

Tabla 4-25 Variables y constantes

<b>Comentarios, separadores, líneas, espacios en blanco y márgenes.</b> Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con solo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.
Líneas en blanco	Se emplean antes y después de métodos, clases y estructuras.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto
Aspectos generales	Sobre el comentario	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.
	Sobre los espacios en blanco	No se debe usar espacio en blanco: <ul style="list-style-type: none"> <li>• Después del corchete abierto y antes del cerrado arreglo.</li> <li>• Después del paréntesis abierto y antes del cerrado.</li> <li>• Antes de un punto y coma.</li> </ul>

Tabla 4-26 Comentarios, separadores, líneas, espacios en blanco y márgenes

<b>Clases y objetos</b>		
Objetivo: Nombrar las clases e instancias de forma estándar para todas las aplicaciones.		
Apariencia de clases y objetos	Primera letra en mayúscula	Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MiClase(). Para el caso de las instancias se comenzara con un prefijo que identificará el tipo de dato, este se escribirá en minúscula.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing**.
Declaración de parámetro en funciones	Agrupados por tipos. Poner los string 1, numéricos 2, además agrupar según valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos (tabla 4-29).
Aspectos generales	Sobre las clases, los objetos, los atributos y las funciones.	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

Tabla 4-27 Clases y objetos

<b>Bases de datos, tablas, esquemas y campos</b>		
Apariencia de los campos	Todas las letras en minúscula.	El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: 'id_producto';
Nombre de los campos	En caso de identificadores.	Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo Ejemplo: id_municipio.
Aspectos generales	Sobre las BD, vistas, tablas atributos y procedimientos.	El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos.

Tabla 4-28 Bases de datos, tablas, esquemas y campos

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes
flota	f	fPesoPaciente
double	d	dPesoCarro
bool	b	bPacienteActivo
string	s	sNombrePaciente
char	c	cLetra
De tipo enum	e	eSexo
byte	b	bCantDiasPaciente
sbyte	sb	sbEdadPaciente
short	sh	shVariableShort
ushort	us	usVariableUshort
uint	ui	uiVariableUint
long	l	lVariableLong
ulong	ul	ulVariableUlong
decimal	dc	dcVariableDecimal
Objetos	o	oPacienteHistorico
Objetos de tipo Struct	st	stUnaStruct

Tabla 4-29 Tipo de datos

En el presente capítulo se describieron los elementos fundamentales relacionados con la implementación. Se obtuvo el modelo de datos, diagrama de despliegue, diagrama de componentes, se describió como se llevó a cabo el tratamiento de errores en el sistema propuesto, así como los estándares de codificación que se utilizaron para la implementación del mismo. Una vez concluido este proceso se ha obtenido un producto listo para su despliegue inicial en el entorno del cliente, cumpliendo con los requisitos funcionales y no funcionales.

## **Conclusiones**

Con el desarrollo de las tareas trazadas se le dio cumplimiento al objetivo general y se arribaron a las siguientes conclusiones:

- Los procesos del negocio asociados al área de almacén de las instituciones hospitalarias demostraron la necesidad del diseño de un sistema informático para la gestión de la información. A través de estos procesos se identificaron las funcionalidades previas a tener en cuenta para el desarrollo del sistema propuesto.
- Los sistemas informáticos que automatizan los procesos que se desarrollan dentro de los almacenes hospitalarios, no responden totalmente a las funcionalidades identificadas.
- La arquitectura definida por el área temática provee un ambiente robusto, seguro y flexible para el desarrollo del sistema.
- El diseño del sistema propuesto está en correspondencia con la arquitectura definida garantizando una representación lógica y óptima de los componentes de la solución.
- Los artefactos obtenidos mediante la metodología de desarrollo RUP facilitaron la implementación del sistema propuesto.
- El sistema propuesto automatiza la recepción de los productos en los almacenes hospitalarios garantizando una mejor gestión de la información.

## ***Recomendaciones***

Con el objetivo de brindar a los hospitales un mejor sistema para la administración y control de sus almacenes, los autores recomiendan:

- Establecer una retroalimentación con el cliente sobre los posibles cambios en los flujos de los procesos que se llevan a cabo en los almacenes hospitalarios, con el fin de mantener el sistema lo más actualizado posible.
- Incorporar funcionalidades al sistema que permitan gestionar una gama de reportes relacionados con la entrada de productos, consumos, pérdidas, desincorporaciones y otros que sirvan al sistema para mejorar el control sobre los productos, esto permitirá contar con información estadística generada en tiempo real.
- Incluir un sistema de ayuda en línea para los usuarios que interactúan con la aplicación.
- Implementar las funcionalidades referentes al proceso de gestión de las solicitudes de compras y de licitaciones, así como el despacho de los productos en los almacenes hospitalarios; lo que posibilitará automatizar todo el ciclo de vida del producto dentro del almacén, desde el momento en que es solicitado hasta ser despachado.

**Referencias bibliográficas**

- [1]. José Ángel Sanguino, Dra. Tibisay Hernández, Dr. Osmán de Jesús Argüello. La Informática Médica en Venezuela, 2007. [Disponible en: <http://groups.google.com/group/asociacion-venezolana-de-informatica-en-salud/web/articulo-para-la-revista-espaola-de-informtica-en-salud>].
- [2]. Antonio Cerritos, Francisco J. Fernández Puerto, Florina Gatica Lara. Sistema de Información Hospitalaria, 2003. [Disponible en: <http://educacion.salud.gob.mx/cursos/informatica/HIS/his.pdf>].
- [3]. Confederación general del trabajo. Los suministros, 2006. [Disponible en: <http://www.cgtsanidadlpa.org/TEMARIO%20CELADOR%20PDF/TEMARIO/TEMA%206.pdf>].
- [4]. Idem [3].
- [5]. CNT Pacientes HIS Edition. [Disponible en: <http://www.cnt.com.co/pagina/index.asp?id=172>]
- [6]. CNT Pacientes Basic Edition. [Disponible en: <http://www.cnt.com.co/pagina/index.asp?id=200>]
- [7]. ASSIST una solución integral para el funcionamiento óptimo de un hospital. [Disponible en: [http://www.grupotca.com/es/filesCosas del escritorio/brochureassist\\_v1.pdf](http://www.grupotca.com/es/filesCosas%20del%20escritorio/brochureassist_v1.pdf)]
- [8]. Idem [7]
- [9]. Idem [7]
- [10]. X-HIS. [Disponible en: [http://www.xsiis.com.mx/pdfs/xHIS\\_080108.pdf](http://www.xsiis.com.mx/pdfs/xHIS_080108.pdf)]
- [11]. Joaquín Herrero Santoja. La experiencia de Kewan – Cosmosalud. [Disponible en: <http://www.informatica2007.sld.cu/Members/jherreroypf/la-experiencia-kewan-cosmosalud/2006-11-15.1968340577/download>]

- [12]. GESTLOG: Gestión de almacenes. [Disponible en:  
[http://www.settingconsultoria.cat/web/index.php?option=com\\_content&view=category&layout=blog&id=75&Itemid=101&lang=es](http://www.settingconsultoria.cat/web/index.php?option=com_content&view=category&layout=blog&id=75&Itemid=101&lang=es)]
- [13]. Documento de arquitectura del Sistema de Información Hospitalaria alas HIS. ALBET, Ingeniería y Sistemas, 2008.
- [14]. Idem [13]
- [15]. Idem [13]
- [16]. Rational Unified Process (RUP). Universidad Tecnológica de Matamoros. México. [Disponible en:  
<http://www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/RUP.pdf>]
- [17]. Enrique Hernández Orallo. El Lenguaje Unificado de Modelado (UML). Departamento de informática. Universidad Politécnica de Valencia. [Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>]
- [18]. García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros. Universidad de Navarra, España, 2000. [Disponible en:  
<http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf>]
- [19]. Herramientas CASE. Instituto Nacional de Estadísticas e Informática. [Disponible en:  
<http://informatica.gonzalonazareno.org/file.php/8/case.pdf>]
- [20]. Raquel Ochoa Ornelas. Facultad de ingeniería mecánica y eléctrica. Universidad de Colima, 2003. [Disponible en: [http://digeset.ucol.mx/tesis\\_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf](http://digeset.ucol.mx/tesis_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf)]
- [21]. Diagramas de comportamiento. Departamento de sistemas y computación del Instituto Tecnológico de La Paz. México. [Disponible en:<http://sistemas.itlp.edu.mx/tutoriales/fundamentosdeprog/t262.htm>]
- [22]. Adjudicación. [Disponible en: [http://www.funcionpublica.gob.mx/unaopspf/credito/bid\\_anex.htm](http://www.funcionpublica.gob.mx/unaopspf/credito/bid_anex.htm)]

[23]. Adquisición. [Disponible en: [http://www.funcionpublica.gob.mx/unaopspf/credito/bid\\_anex.htm](http://www.funcionpublica.gob.mx/unaopspf/credito/bid_anex.htm)]

[24]. AJAX. [Disponible en: <http://es.wikipedia.org/wiki/AJAX>]

[25]. Manual para la administración y control de almacenes e inventarios, 2004. [Disponible en: <http://www.sagarpa.gob.mx/Pronabive/PDF/ADMONALMAC.PDF>]

[26]. Idem [25]

[27]. API. [Disponible en: <http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=A&pagina=5>]

[28]. Idem [25]

[29]. Framework. [Disponible en: <http://es.wikipedia.org/wiki/Framework>]

[30]. Ing. Zamudio Mendoza Mario Said. Interfaz de usuario. [Disponible en: <http://univafuadoo.files.wordpress.com/2008/12/interfaz-de-usuario.ppt>]

[31]. Idem [25]

[32]. Asier Romero Andonegi. Nuevas herramientas para el estudio de la diplomática: Los lenguajes de marcado TEI y EAD, 2008. [Disponible en: [http://www.ucm.es/info/documen/articulos/art\\_sexta/006.pdf](http://www.ucm.es/info/documen/articulos/art_sexta/006.pdf)]

[33]. Lenguaje de programación interpretado. [Disponible en: [http://es.wikipedia.org/wiki/Lenguaje\\_interpretado](http://es.wikipedia.org/wiki/Lenguaje_interpretado)]

[34]. Licitaciones. [Disponible en: [http://www.funcionpublica.gob.mx/unaopspf/credito/bid\\_anex.htm](http://www.funcionpublica.gob.mx/unaopspf/credito/bid_anex.htm)]

[35]. Idem [25]

[36]. Modelado y Diseño Orientado a Objetos. Universidad Nacional de Colombia. [Disponible en: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100010/Lecciones/Cap7/OMT.htm>]

[37]. Software. [Disponible en: [http://es.wikipedia.org/wiki/Computer\\_software](http://es.wikipedia.org/wiki/Computer_software)]

## **Bibliografía**

Ajax4JSF Developer Guide, 2007. [Disponible en:

<http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/index.html>]

Andrea Delgado. Desarrollo de Software con enfoque en el Negocio. [Disponible en:

<http://www.sistedes.es/sistedes/pdf/2007/pnis-07-delgado-dsen.pdf>]

Borja Ozores Massó. Logística hospitalaria. Capítulo 2: Los hospitales y la logística. [Disponible en:

[http://tienda.marge.es/WebRoot/StoreES/Shops/ea2557/4869/0381/BB3C/629F/2023/D94C/8776/3D05/Log\\_0020\\_Hosp\\_0020\\_Cap\\_0020\\_2.pdf](http://tienda.marge.es/WebRoot/StoreES/Shops/ea2557/4869/0381/BB3C/629F/2023/D94C/8776/3D05/Log_0020_Hosp_0020_Cap_0020_2.pdf)]

Bruegge. B. Dutoit. A. Ingeniería de Software Orientado a Objetos, 2002.

David Ruiz Muñoz. Nuevas tendencias en la logística sanitaria. Universidad Pablo de Olavide.

Departamento de Economía y Empresa. Sevilla. [Disponible en:

<http://external.doyma.es/pdf/261/261v3n3a13082029pdf001.pdf>]

Departamento de Lenguajes y Sistemas Informáticos. Patrones Arquitectónicos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. [Disponible en:

<http://www.lsi.us.es/docencia/get.php?id=1891>]

Departamento de lenguajes y sistemas informáticos. Universidad Rey Juan Carlos. Madrid, 2007.

[Disponible en: [http://kybele.escet.urjc.es/Documentos/ISI/Implem\\_pruebas.pdf](http://kybele.escet.urjc.es/Documentos/ISI/Implem_pruebas.pdf)]

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

[http://www.sld.cu/galerias/doc/sitios/infodir//09\\_el\\_sistema\\_nacional\\_de\\_salud.doc](http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc)

Dra. Assetta A, Dr. Fernández Romero D., Dr. Rosell S. Sistemas de Información Hospitalaria, 2006.

[Disponible en: <http://www.intramed.net/actualidad/contenido.asp?contenidoID=44061>]

Drools Documentation Library. [Disponible en: <http://www.jboss.org/drools/documentation.html>]

Especificación de requisitos. Módulo Almacén del Sistema de Información Hospitalaria alas HIS. ALBET, Ingeniería y Sistemas, 2008.

Estándares de codificación de sistemas, 2006. [Disponible en:  
<http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf>]

Guía de Usuario de Enterprise Architect. [Disponible en:  
<http://www.sparxsystems.com.ar/download/Ayuda%20HTML/index.html?packagediagram.htm>]

Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Addison Wesley, 2000.

Jesús M. González Barahona, Joaquín Seoane Pascual, y Gregorio Robles. Introducción al software libre. [Disponible en: [http://www.atenas.cult.cu/ri/informatica/manuales/sl/introduccion\\_al\\_SL/eclipse.html](http://www.atenas.cult.cu/ri/informatica/manuales/sl/introduccion_al_SL/eclipse.html)]

Joseph Schmuller. Aprendiendo UML en 24 horas. Prentice Hall. México, 2000.

Manuel Sánchez Barrientos. Introducción a BPMN, 2008. [Disponible en:  
<http://www.aprendergratis.com/introduccion-a-bpmn.html>]

Modelo de casos de uso del sistema. Módulo Almacén del Sistema de Información Hospitalaria alas HIS. ALBET, Ingeniería y Sistemas, 2008.

Navarro Franco, Ángel José. UML en acción. Modelando Aplicaciones Web. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.

Notas y transparencias de Sistemas Distribuidos. Universidad Politécnica de Madrid. Departamento de informática aplicada. [Disponible en: [http://www.dia.eui.upm.es/Asignatu/Sis\\_dis/Paco/Contenido.pdf](http://www.dia.eui.upm.es/Asignatu/Sis_dis/Paco/Contenido.pdf)]

Procesos Elementales del Negocio\_Procesos actuales. Módulo Almacén del Sistema de Información Hospitalaria alas HIS. ALBET, Ingeniería y Sistemas, 2008.

René Arenas Gutiérrez. La revolución Científico Técnica en la salud, desde la perspectiva de la informática medica. Facultad de Ciencia Medicas Julio Trigo López. [Disponible en: <http://fcmjtrigo.sld.cu/materiales/informatica/rctdesdelaimedica.doc>]

Restricciones de nomenclatura y Estándares de codificación del Sistema de Información Hospitalaria alas HIS.

Richfaces Developer Guide. [Disponible en: [http://www.jboss.org/file-access/default/members/jbossrichfaces/freezezone/docs/devguide/en/pdf/richfaces\\_usersguide.pdf](http://www.jboss.org/file-access/default/members/jbossrichfaces/freezezone/docs/devguide/en/pdf/richfaces_usersguide.pdf)]

Roger S. Pressman. Ingeniería de Software, un enfoque práctico. Parte 1. La Habana, Cuba. Editorial Félix Varela, 2005

Santiago Marimón Suñol. La sanidad en la sociedad de la información, 1999. [Disponible en: [http://books.google.com.cu/books?id=TxbRL6Xj\\_bwC&printsec=frontcover&dq=la+sanidad+en+la+sociedad+de+la+informacion](http://books.google.com.cu/books?id=TxbRL6Xj_bwC&printsec=frontcover&dq=la+sanidad+en+la+sociedad+de+la+informacion)]

Sindicato de sanidad de Las Palmas, Canarias. Tema 6: Los suministros. Suministros internos y externos. Recepción y almacenamiento de mercancías. Organización del almacén. Distribución de pedidos. [Disponible en: <http://www.cgtsanidadlpa.org/TEMARIO%20CELADOR%20PDF/TEMARIO/TEMA%206.pdf>]

Teoría de sistemas. Universidad de Valparaíso, 2005. [Disponible en: [www.kyberne.cl/Documentos%20de%20Partners/13%20-%20TGS%20-%20ModProcesos-BPMN.ppt](http://www.kyberne.cl/Documentos%20de%20Partners/13%20-%20TGS%20-%20ModProcesos-BPMN.ppt)]

The Java EE5 Tutorial. Sun Java System Application Server 9.1. 2008. [Disponible en: <http://java.sun.com/javaee/5/docs/tutorial/doc/JavaEETutorial.pdf>]

Wilmer Jaramillo M. JBoos Application Server. 2006. [Disponible en: <http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf>]

## Anexos

### Anexo 1.Descripcion textual de casos de uso

<b>CASO DE USO:</b>	<b>Ver detalles de informe de recepción</b>
<b>Propósito:</b>	Ver informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor crea un informe de recepción, al dar aceptar se le muestra automáticamente los detalles del informe creado brindándole la posibilidad de modificar sus datos o de eliminar el mismo, el caso de uso termina.
<b>Precondiciones:</b>	Para ver los datos del informe de recepción este debe haber sido creado.
<b>Poscondiciones:</b>	Se vieron los datos del informe de recepción.
<b>Referencia:</b>	RF – 3.2

**Tabla 9-1 CU Buscar informe de recepción**

<b>CASO DE USO:</b>	<b>Buscar informe de recepción</b>
<b>Propósito:</b>	Buscar un informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción buscar informe de recepción, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el informe de recepción deseado, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los informes de recepción que cumplen con los criterios de búsqueda, el sistema permite seleccionar el informe de recepción deseado, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se buscó el informe de recepción deseado.
<b>Referencia:</b>	RF – 3.3

**Tabla 9-2 CU Buscar informe de recepción**

<b>CASO DE USO:</b>	<b>Ver informe de recepción</b>
<b>Propósito:</b>	Ver datos de un informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona un informe de recepción y accede a la opción de ver datos del informe de recepción, el sistema muestra los datos de este, el caso de uso termina.
<b>Precondiciones:</b>	Para ver los datos de un informe de recepción, este debe haber sido seleccionado.
<b>Poscondiciones:</b>	Se mostró los datos de un informe de recepción.
<b>Referencia:</b>	RF – 3.4

Tabla 9-3 CU Ver informe de recepción

<b>CASO DE USO:</b>	<b>Modificar informe de recepción</b>
<b>Propósito:</b>	Modificar un informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona un informe de recepción y accede a la opción modificar datos del informe de recepción, el sistema muestra los datos de este y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos del informe de recepción, el caso de uso termina.
<b>Precondiciones:</b>	Para modificar un informe de recepción, este debe haber sido seleccionado.
<b>Poscondiciones:</b>	Se modificó un informe de recepción.
<b>Referencia:</b>	RF – 3.5

Tabla 9-4 CU Modificar informe de recepción

<b>CASO DE USO:</b>	<b>Eliminar informe de recepción</b>
<b>Propósito:</b>	Eliminar un informe de recepción.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona un informe de recepción y accede a la opción eliminar informe de recepción, el sistema elimina este, el caso de uso termina.
<b>Precondiciones:</b>	Para eliminar un informe de recepción, este debe haber sido seleccionado.
<b>Poscondiciones:</b>	Se eliminó un informe de recepción.
<b>Referencia:</b>	RF – 3.6

Tabla 9-5 CU Eliminar informe de recepción

<b>CASO DE USO:</b>	<b>Seleccionar desincorporación</b>
<b>Propósito:</b>	Seleccionar una desincorporación.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción seleccionar desincorporación, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la desincorporación deseada, el actor introduce los datos que considera como criterios para buscar la misma, el sistema busca y muestra las que cumplen con los criterios de búsqueda, el actor selecciona la desincorporación deseada, el sistema carga en la vista anterior cada una de las seleccionadas, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se seleccionó una desincorporación.
<b>Referencia:</b>	RF - 6

Tabla 9-6 CU Seleccionar desincorporación

<b>CASO DE USO:</b>	<b>Seleccionar solicitud de desincorporación de productos biológicos</b>
<b>Propósito:</b>	Seleccionar la solicitud de desincorporación de productos biológicos.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción seleccionar solicitud de desincorporación de productos biológicos, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la solicitud de desincorporación de productos biológicos deseada, el actor introduce los datos que considera como criterios para buscar la misma, el sistema busca y muestra las que cumplen con los criterios de búsqueda, el actor selecciona la solicitud de desincorporación de productos biológicos deseada, el sistema carga en la vista anterior cada una de las seleccionadas, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se seleccionó la solicitud de desincorporación de productos biológicos deseada.
<b>Referencia:</b>	RF - 7

**Tabla 9-7 CU Seleccionar solicitud de desincorporación de productos biológicos**

<b>CASO DE USO:</b>	<b>Aprobar desincorporación</b>
<b>Propósito:</b>	Aprobar una desincorporación.
<b>Actores:</b>	Administrativo
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide aprobar las desincorporaciones realizadas por los almacenes. El sistema le muestra todas las desincorporaciones por aprobar, el actor selecciona la que desea aprobar, el sistema le muestra los datos del acta de desincorporación creada y los productos desincorporados, el sistema le permite al actor aprobar o no las desincorporación de los productos, en caso de que no la apruebe el sistema le permite introducir las causas del rechazo, el caso de uso termina.
<b>Precondiciones:</b>	Debe de estar registrada al menos una desincorporación.
<b>Poscondiciones:</b>	Se aprobó o rechazó una desincorporación.
<b>Referencia:</b>	RF - 8

Tabla 9-8 CU Aprobar desincorporación

<b>CASO DE USO:</b>	<b>Desincorporar productos</b>
<b>Propósito:</b>	Desincorporar productos.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción de desincorporar productos, el sistema le da la oportunidad de seleccionar los productos a desincorporar, permite establecer la salida por lotes especificando la cantidad a desincorporar e introduciendo los motivos de la desincorporación, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se desincorporó productos.
<b>Referencia:</b>	RF - 9

Tabla 9-9 CU Desincorporar productos

<b>CASO DE USO:</b>	<b>Desincorporar productos biológicos</b>
<b>Propósito:</b>	Desincorporar productos biológicos.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción desincorporar productos biológicos, el sistema brinda la posibilidad de seleccionar las solicitudes de desincorporación de epidemiología, y desincorporar estos productos.
<b>Precondiciones:</b>	Exista una solicitud de desincorporación de epidemiología.
<b>Poscondiciones:</b>	Se desincorporó los productos biológicos.
<b>Referencia:</b>	RF - 10

**Tabla 9-10 CU Desincorporar productos biológicos**

<b>CASO DE USO:</b>	<b>Seleccionar devolución</b>
<b>Propósito:</b>	Seleccionar la devolución deseada.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción seleccionar devolución, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la devolución deseada, el actor introduce los datos que considera como criterios para buscar la misma, el sistema busca y muestra las devoluciones que cumplen con los criterios de búsqueda, el actor selecciona la devolución deseada, el sistema carga en la vista anterior la devolución seleccionada, el caso de uso termina.
<b>Precondiciones:</b>	Que exista una solicitud de devolución.
<b>Poscondiciones:</b>	Se seleccionó la devolución deseada.
<b>Referencia:</b>	RF – 12

**Tabla 9-11 CU Seleccionar devolución**

<b>CASO DE USO:</b>	<b>Aceptar devolución</b>
<b>Propósito:</b>	Aceptar las devoluciones que se le realizan al almacén.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide aceptar devoluciones que le estén haciendo al almacén, el sistema permite listar todas las devoluciones que se le realizan, el actor selecciona las que proceden, consulta sus datos y la acepta o no, en caso de que la acepte el sistema le permite reasignar la cantidad que fue devuelta a un lote del producto, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se aceptaron las devoluciones.
<b>Referencia:</b>	RF - 13

Tabla 9-12 CU Aceptar devolución

<b>CASO DE USO:</b>	<b>Seleccionar adjudicación recibida</b>
<b>Propósito:</b>	Seleccionar una adjudicación recibida.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción seleccionar adjudicación recibida, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la adjudicación deseada, el actor introduce los datos que considera como criterios para buscar estas, el sistema busca y muestra las adjudicaciones que cumplen con los criterios de búsqueda, el actor selecciona las deseadas, se muestran sus datos, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se seleccionó la adjudicación deseada.
<b>Referencia:</b>	RF - 15

Tabla 9-13 CU Seleccionar adjudicación recibida

<b>CASO DE USO:</b>	<b>Registrar cuarentena</b>
<b>Propósito:</b>	Registrar un producto en cuarentena.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide registrar un lote de productos en cuarentena, el sistema le permite seleccionar el lote de productos que desea colocar en cuarentena, este registra el mismo como tal, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se registró al menos un producto en cuarentena.
<b>Referencia:</b>	RF- 16

Tabla 9-14 CU Registrar cuarentena

<b>CASO DE USO:</b>	<b>Seleccionar cronograma de entrega</b>
<b>Propósito:</b>	Seleccionar un cronograma de entrega.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción seleccionar cronograma de entrega, el sistema muestra el listado de entregas para ese día según el cronograma creado, el actor selecciona la entrega deseada, el caso de uso termina.
<b>Precondiciones:</b>	No existe
<b>Poscondiciones:</b>	Se seleccionó un cronograma de entrega.
<b>Referencia:</b>	RF- 18

Tabla 9-15 CU Seleccionar cronograma de entrega

<b>CASO DE USO:</b>	<b>Consultar tarjeta de estantería</b>
<b>Propósito:</b>	Consultar la tarjeta de estantería de un producto determinado.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor decide consultar los movimientos de un producto durante un período de tiempo determinado, el sistema lista los productos y permite seleccionar el producto del que desea ver la tarjeta de estantería, el sistema muestra los movimientos realizados a dicho producto, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se mostró los movimientos de un producto.
<b>Referencia:</b>	RF- 19

**Tabla 9-16 CU Consultar tarjeta estantería**

<b>CASO DE USO:</b>	<b>Buscar producto</b>
<b>Propósito:</b>	Buscar un producto.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción buscar productos, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar los productos deseados, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los productos que cumplen con los criterios de búsqueda, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se buscó un producto en el almacén.
<b>Referencia:</b>	RF- 20

**Tabla 9-17 CU Buscar producto**

<b>CASO DE USO:</b>	<b>Buscar solicitud de productos</b>
<b>Propósito:</b>	Buscar una solicitud de productos.
<b>Actores:</b>	Jefe de almacén
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción buscar solicitud de productos, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la solicitud de productos deseada, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra las solicitudes que cumplen con los criterios de búsqueda, el caso de uso termina.
<b>Precondiciones:</b>	No existen
<b>Poscondiciones:</b>	Se buscó una solicitud de productos.
<b>Referencia:</b>	RF- 21

**Tabla 9-18 CU Buscar solicitud de productos**

Anexo 2. Diagrama de clases del diseño

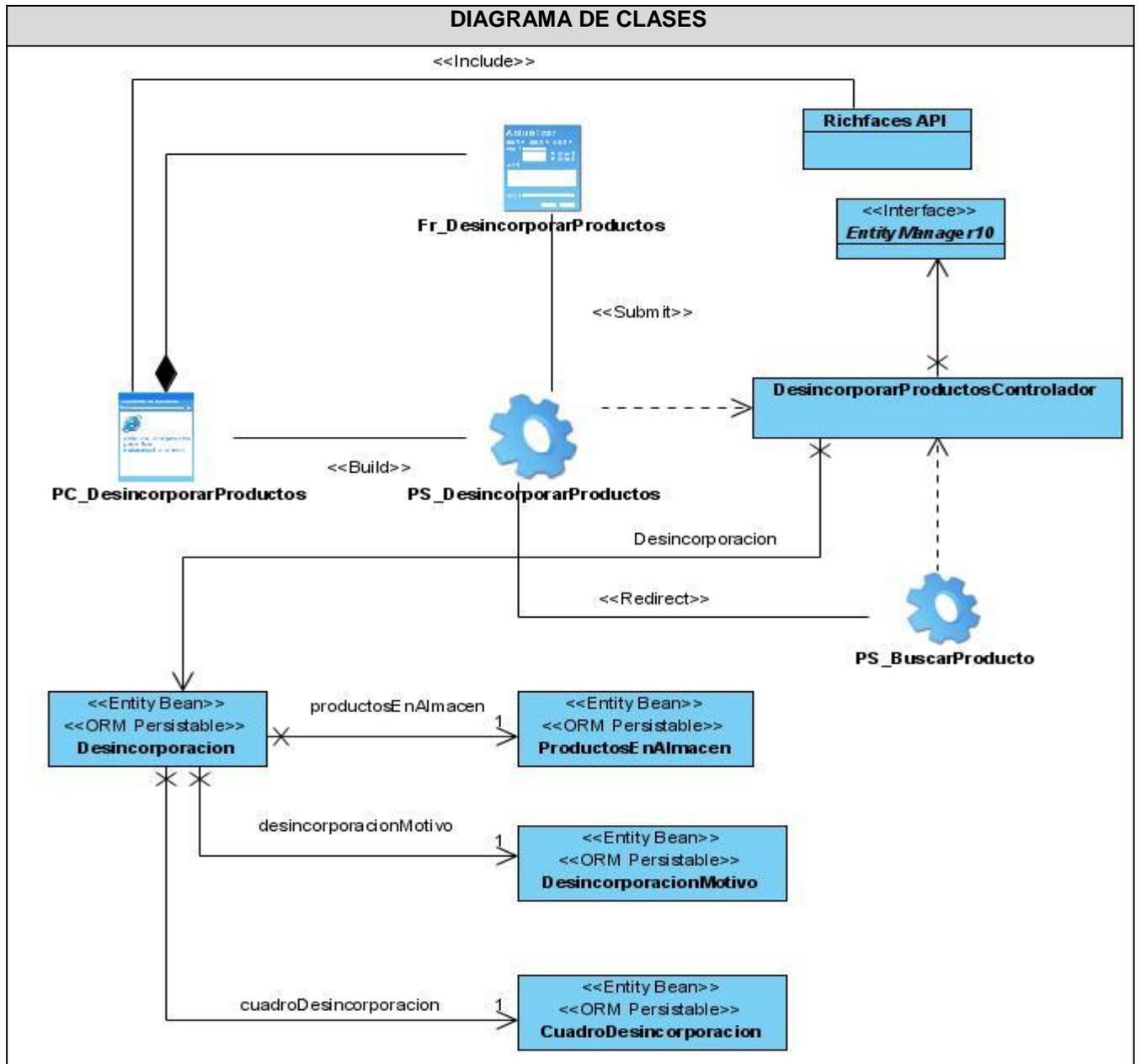


Figura 9-19 DCD Desincorporar productos

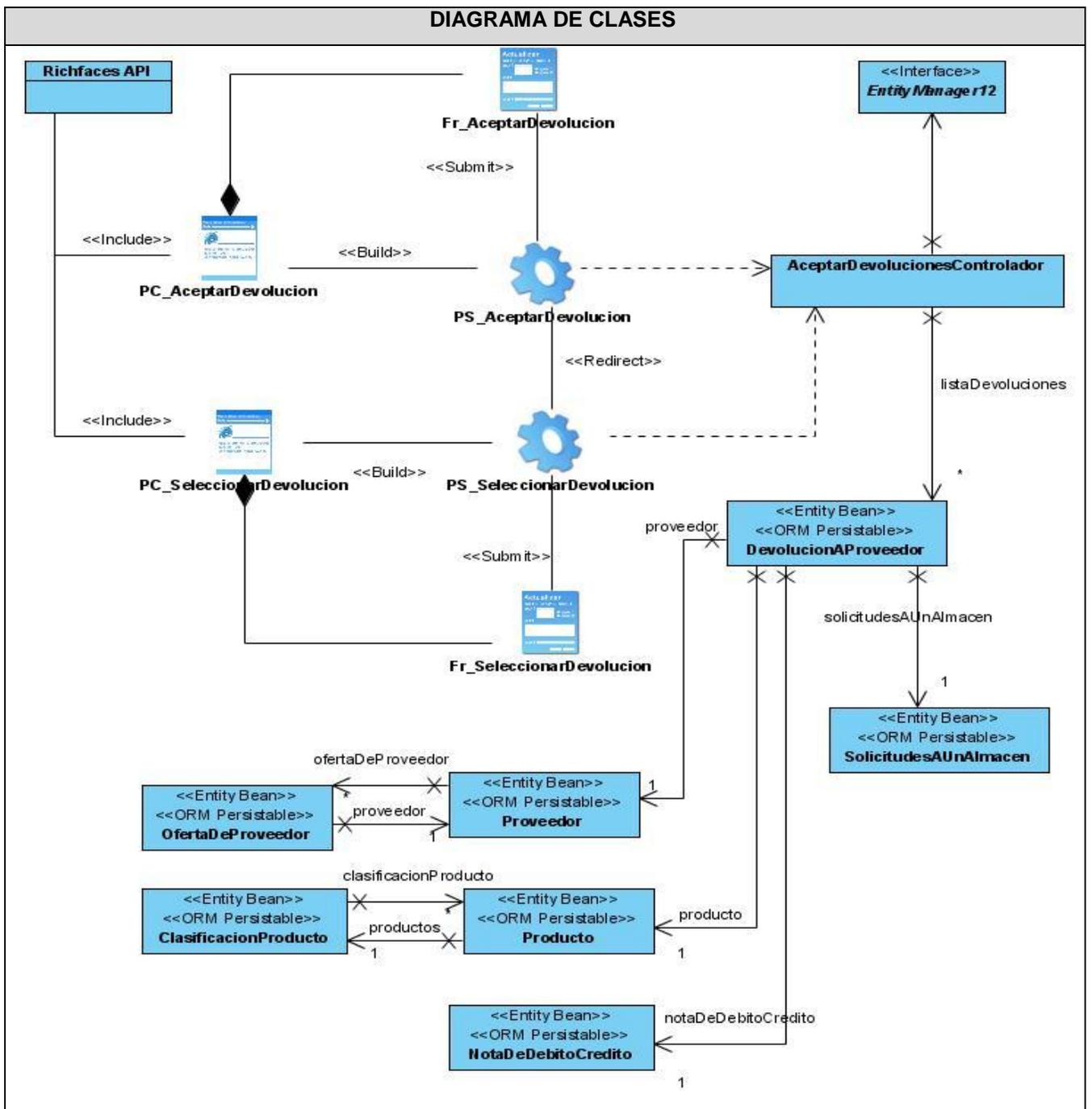


Figura 9-20 DCD Aceptar devolución

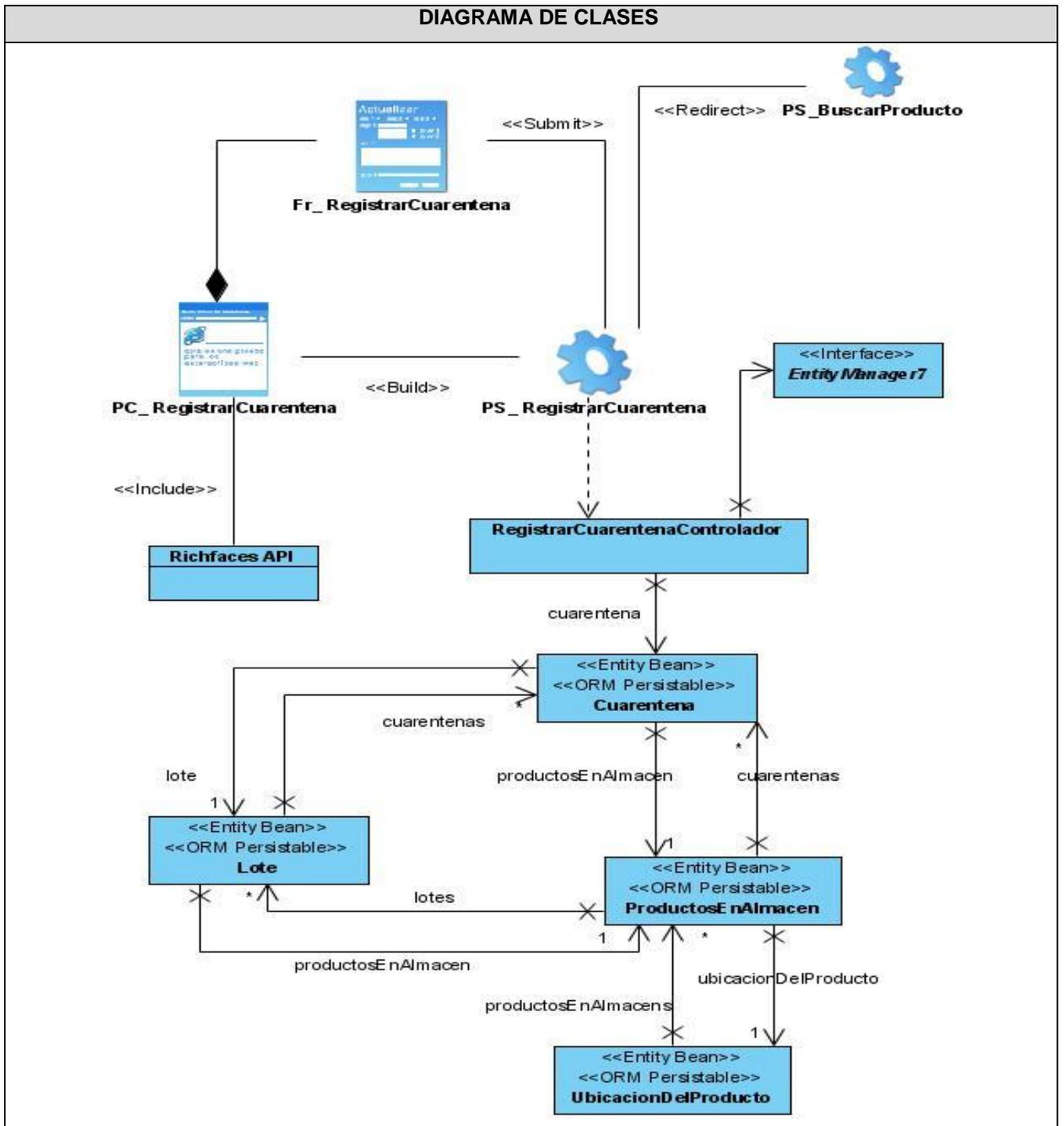


Figura 9-21 DCD Registrar cuarentena

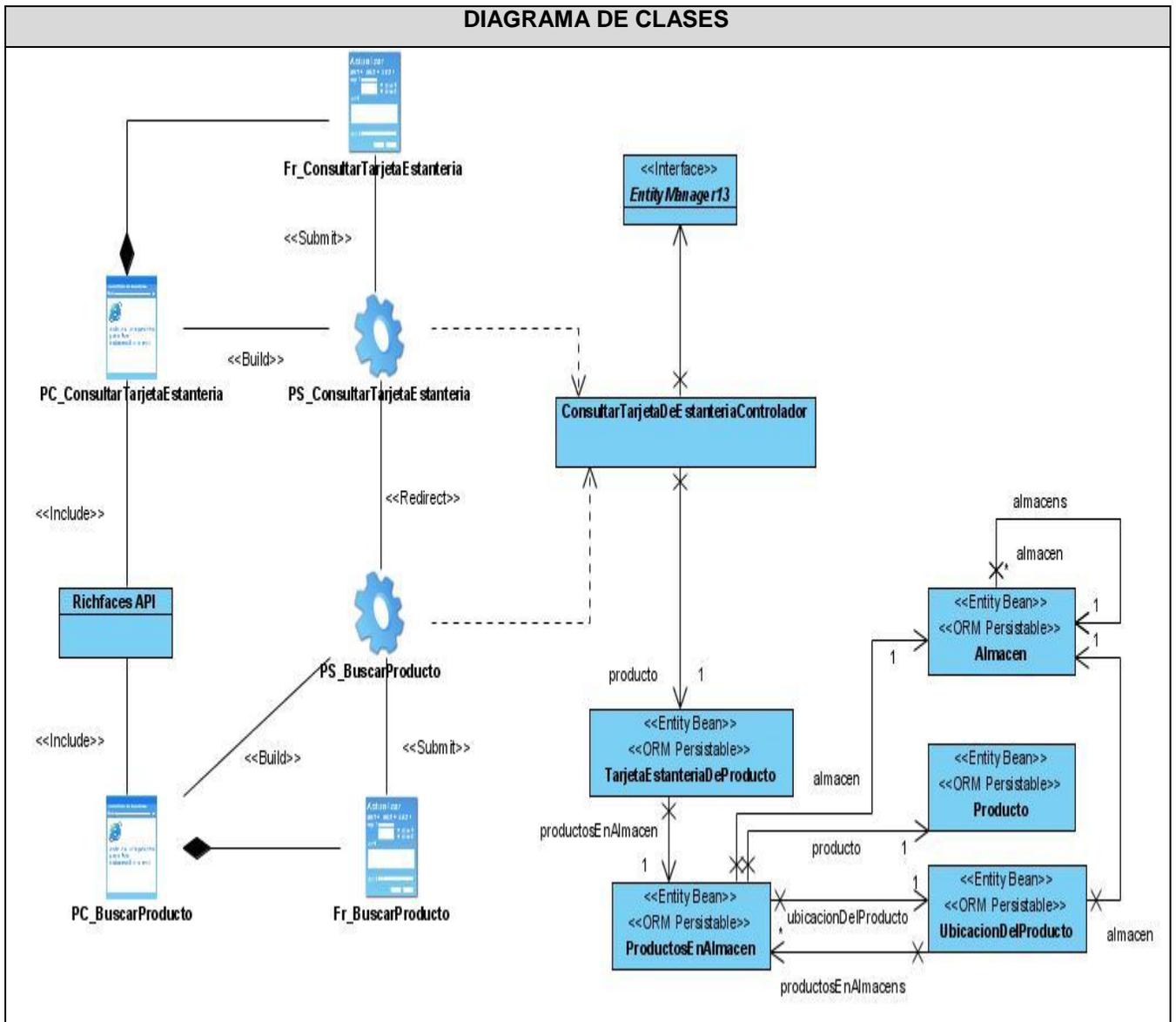


Figura 9-22 DCD Consultar tarjeta de estantería

Anexos 3. Pantallas del sistema

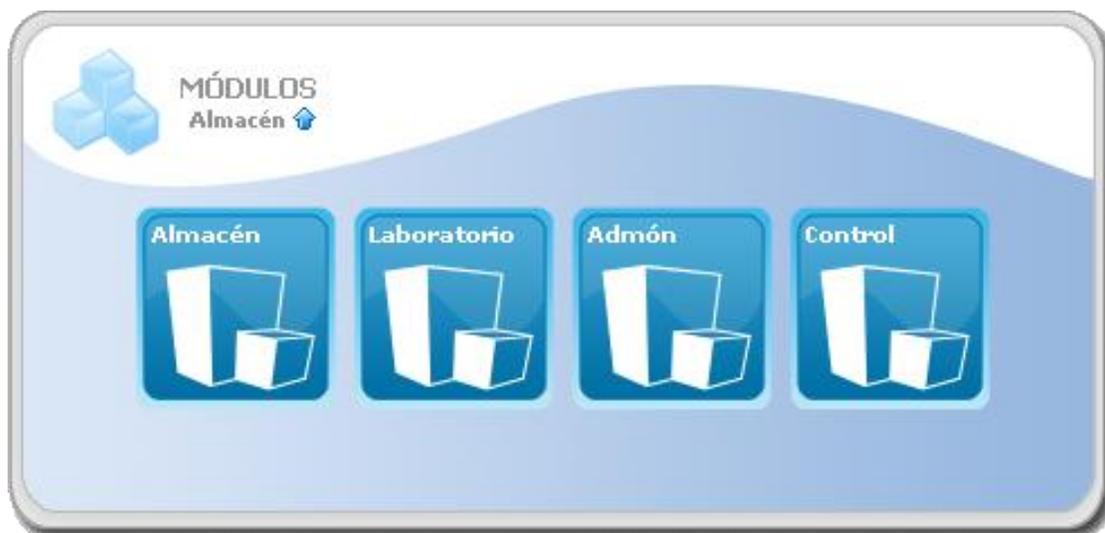


Figura 9-23 Selector de módulos

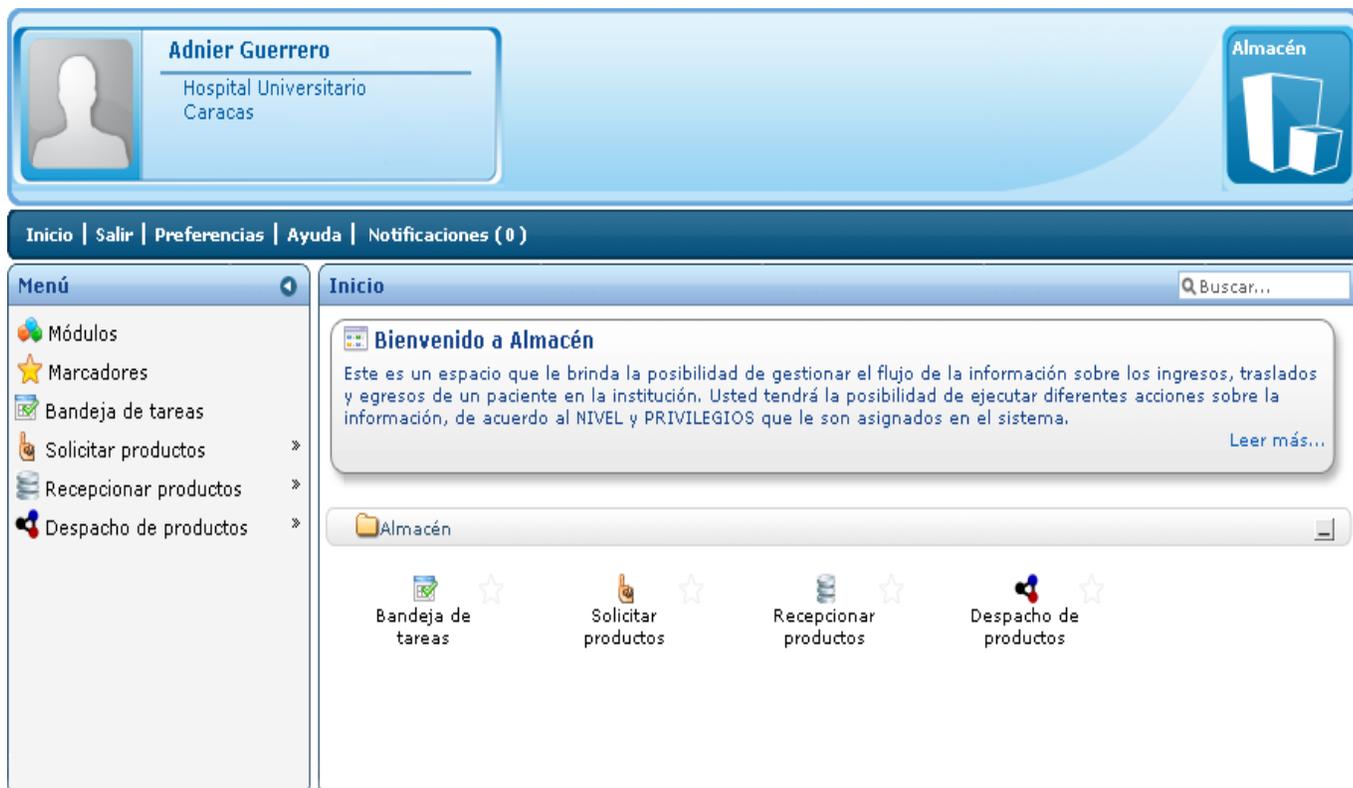


Figura 9-24 Selector de funcionalidades

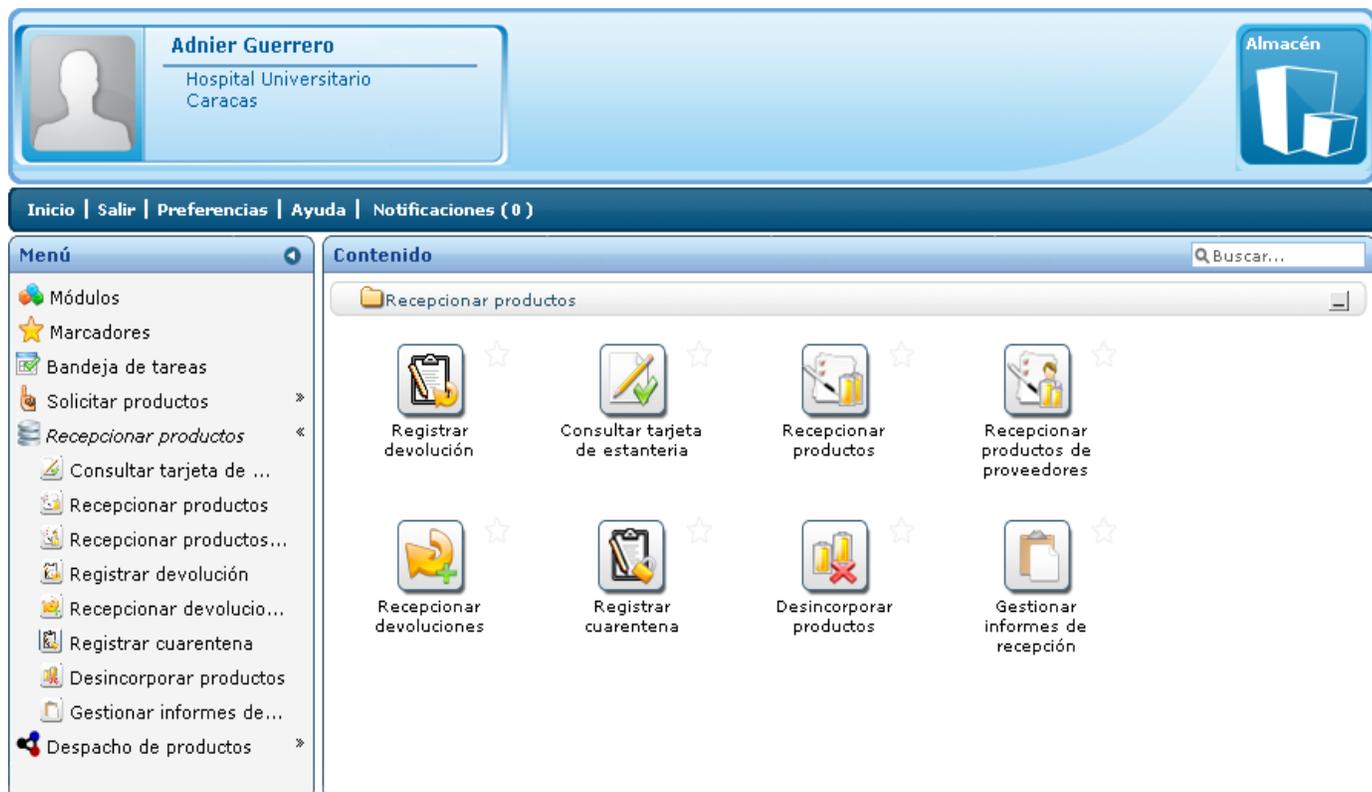


Figura 9-25 Selector de funcionalidades para: Recepcionar productos



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

Recepcionar productos Buscar...

**Criterios de búsqueda**

Fecha:  Destinatario:  Estado: Seleccione

**Listado de solicitudes de productos**

Fecha	Estado	Entidad	
16/6/2009	Despachada	Almacén Central	
18/6/2009	Despachada	Almacén Central	
19/6/2009	Despachada	Almacén Central	

**Productos de: Almacén Central**

Código	Producto	
	Cloraceposido	10

**Datos del producto: Cloraceposido**

Stock max:  Stock min:  Fecha:

Precio unitario:  Cantidad:  Descripción:

[+ Nueva ubicación](#)

**Ubicación**

Estante: Estante 1  
Pasillo: Pasillo 2  
Tramo: Tramo1  
Gaveta: Baveta 2

**Listado de ubicaciones**

Estante	Pasillo	Tramo	Gaveta	
Estante 1	Pasillo 2	Tramo1	Baveta 2	

Figura 9-26 Recepcionar productos



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

Buscar...

**Recepcionar productos de proveedores**

**Listado de entregas**

Código	Producto	Cantidad	Proveedor	p/u	
02563	Cloraceposido	300	LASBA	3	✓
01254	Penicilina	300	LASBA	2	📄

⏪ ⏩ ⏴ ⏵

**Adjudicación**



Proveedor: LASBA  
Código: 02563    Producto: Cloraceposido    Cantidad: 300    Orden de compra: 2416

Crear Nota: 📄

**Datos del producto**

Stock max:     Stock min:     Fecha vencimiento:

Factura:     Cantidad:     Descripción:

**Ubicación**

Estante: estante  
Pasillo: pasillo  
Tramo: tramo  
Gaveta: gaveta

+ Nueva ubicación

**Listado de ubicaciones**

Estante	Pasillo	Tramo	Gaveta	
estante	pasillo	tramo	gaveta	✓
estante 2	pasillo 2	tramo 2	baveta 2	📄

⏪ ⏩ ⏴ ⏵

Figura 9-27 Recepcionar productos de proveedores

119



Figura 9-28 Bandeja de tareas de los almacenes

**Adnier Guerrero**  
Hospital Universitario  
Caracas

Almacén

Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

**Crear informe de recepción**

**Datos de la recepción**

 Proveedor: LASBA  
Código: 01254    Producto: Penicilina    Solicitados: 300    Recibidos: 300

**Observaciones**

Aceptar    Cancelar

Figura 9-29 Crear informe de recepción



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones ( 0 )

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

**Buscar informe de recepción** Q Buscar...

**Criterios de búsqueda**

Proveedor:  Buscar Cancelar

**Listado de informes de recepción**

Fecha	Proveedor	Factura			
16/6/2009	LASBA		322		
16/6/2009	LASBA		5421		
19/6/2009	LASBA		4574		
19/6/2009	LASBA		32		
20/6/2009	LASBA		152		

« ‹ › »

Salir

Figura 9-30 Buscar informe de recepción



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones ( 0 )

**Menú**

- 🌐 Módulos
- ★ Marcadores
- 📧 Bandeja de tareas
- 👉 Solicitar productos >
- 📦 *Recepcionar productos* <
- 📄 Consultar tarjeta de ...
- 📄 Recepcionar productos
- 📄 Recepcionar productos...
- 📄 Registrar devolución
- 📄 Recepcionar devolucio...
- 📄 Registrar cuarentena
- 📄 Desincorporar productos
- 📄 Gestionar informes de...
- 🚚 Despacho de productos >

🔍 Buscar...

**Ver informe de recepción**

---

**Datos de la recepción**



Proveedor: LASBA  
Código: 01254    Producto: Penicilina

---

**Observaciones:**

informe

Figura 9-31 Ver informe de recepción

**Adnier Guerrero**  
Hospital Universitario  
Caracas

Almacén

Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos >
- Recepcionar productos <
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos >

**Modificar informe de recepción**

**Datos de la recepción**

	Fecha: 16/06/2009	Proveedor: LASBA	Elaborado por: Adnier Guerrero
	Código: 01254	Producto: Penicilina	Solicitados: 300
	Recibidos: 300		

**Observaciones:**

informe

Aceptar Cancelar

Figura 9-32 Modificar informe de recepción



Figura 9-33 Bandeja de tareas de control perceptivo



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones ( 0 )

**Menú**

-  Módulos
-  Marcadores
-  Bandeja de tareas

**Aprobar recepción** Q Buscar...

**Datos de la recepción**

	Fecha: 21/06/2009	Proveedor: LASBA	
	Código: 01254	Producto: Penicilina	Solicitdos: 300
	Contrato: 1425	Factura: 2124	Tipo de pago: Credito
			Recividos: 200
			Monto total: 400

**Descripción del material recibido:** «

**Observaciones de la recepción:** «

Figura 9-34 Aprobar recepción



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos >
- Recepcionar productos <
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos >

Buscar...

**Desincorporar productos**

**Criterios de búsqueda**

Código:  Producto:

[Búsqueda avanzada](#)

Listado de productos en el almacén			
Código	Producto	Cantidad	
01254	Penicilina	340	
02154	Diecepan	420	

⏪ ⏩ ⏴ ⏵

**Listado de productos desincorporados**

Código	Producto	Cantidad	Motivo		
02154	Diecepan	0	<Seleccione>		

Figura 9-35 Desincorporar productos



Figura 9-36 Bandeja de tareas de admón



**Adnier Guerrero**  
Hospital Universitario  
Caracas



[Inicio](#) | [Salir](#) | [Preferencias](#) | [Ayuda](#) | [Notificaciones \(0\)](#)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitudes

**Aprobar desincorporación**

**Criterios de búsqueda**

Fecha:  Motivo:

[Búsqueda avanzada](#)

**Datos del informe**

 Fecha: 19/6/2009  
Desincorporado por: Adnier Guerrero Almacén: Laboratorio

[Rechazar todo](#) [Cancelar todo](#)

**Listado de productos por desincorporados**

Código	Producto	Cantidad	Motivo	Destino
02154	Diecepan	10	Vencimiento	I

⏪ ⏩ ⏴ ⏵

Figura 9-37 Aprobar desincorporación



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas

Buscar...

**Acta de control de desincorporación**

**Criterios de búsqueda**

Fecha:  Motivo:  Buscar Cancelar

[Búsqueda avanzada](#)

Código	Producto	Cantidad	Motivo
01254	Penicilina	10	Vencimiento

« ‹ › »

**Listado de actas de control de desincorporación**

Fecha	Producto	Encargado	Medida	Lugar
26/6/2009	Penicilina	personal	medida	lugar

Aceptar Cancelar

Figura 9-38 Crear acta de control perceptivo de desincorporación



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos >
- Recepcionar productos <
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos >

Buscar...

**Registrar devoluciones a proveedores**

**Criterios de búsqueda**

Código:  Producto:  Buscar Cancelar

Búsqueda avanzada

Registrar devoluciones por cronogramas

**Listado de productos**

Código	Producto	Cantidad	Presentación	
02563	Cloraceposido	300	Tableta	🗑️
01254	Penicilina	300	Tableta	📄

⏪ ⏩ ⏴ ⏵

**Introducir datos de la devolución**

Cantidad:  Proveedor: LASBA    Producto: Cloraceposido    Crear Nota: 📄

Motivo de la devolución:

Aceptar

Figura 9-39 Registrar devolución a proveedores



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

Buscar...

**Criterios de búsqueda**

Entidad:  Producto:  Buscar Cancelar

**Listado de devoluciones**

Fecha	Código	Producto	Cantidad	Entidad	Encargado	Motivo
21/6/2009	01254	Penicilina	3	Laboratorio	Adnier Guerrero	✓

**Listado de devoluciones**

Fecha	Código	Producto	Cantidad	Entidad	Encargado	Estado
21/6/2009	01254	Penicilina	3	Laboratorio	Adnier Guerrero	✗

Aceptar Cancelar

Figura 9-40 Aceptar devolución



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos
- Recepcionar productos
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos

Buscar...

**Registrar cuarentena**

**Criterios de búsqueda**

Código:  Producto:  Buscar Cancelar

[Búsqueda avanzada](#)

**Listado de productos en el almacén**

Código	Producto	Cantidad	
01254	Penicilina	340	✓
02154	Diecepan	420	📦

**Listado de productos en cuarentena**

Código	Producto	Cantidad	Lote	
01254	Penicilina	340	<Seleccione>	🗑️

Aceptar Cancelar

Figura 9-41 Registrar cuarentena



**Adnier Guerrero**  
Hospital Universitario  
Caracas



Inicio | Salir | Preferencias | Ayuda | Notificaciones (0)

**Menú**

- Módulos
- Marcadores
- Bandeja de tareas
- Solicitar productos >
- Recepcionar productos <
- Consultar tarjeta de ...
- Recepcionar productos
- Recepcionar productos...
- Registrar devolución
- Recepcionar devolucio...
- Registrar cuarentena
- Desincorporar productos
- Gestionar informes de...
- Despacho de productos >

Buscar...

**Consultar tarjeta de estantería de producto**

**Criterios de búsqueda**

Código:  Producto:  [Buscar] [Cancelar]

Búsqueda avanzada

---

Desde:  Hasta:

**Listado de productos**

Producto

Diecepan

Penicilina

**Datos del producto**



Código: 02154 Descripción: des  
Nombre: Diecepan Total: 0

**Movimientos**

Tipo	Fecha	Referencia	Cantidad	Total
Entrada	16/6/2009	Proveniente de Almacén Central	20	20
Salida	18/6/2009	Despacho solicitado por Almacén Central	20	0
Entrada	18/6/2009	Devolución de Almacén Central	20	20
Salida	18/6/2009	Movimiento a cuarentena	20	0
Entrada	18/6/2009	Proveniente de Almacén Central	100	100
Salida	18/6/2009	Desincorporación por Laboratorio	50	50
Salida	19/6/2009	Despacho solicitado por Almacén Central	10	40
Salida	19/6/2009	Desincorporación por Laboratorio	10	30
Salida	19/6/2009	Desincorporación por Laboratorio	30	0

[Cancelar] [Salir]

Figura 9-42 Consultar tarjeta de estantería

## **Glosario de Términos**

**Adjudicación:** Es el acto mediante el cual el jefe de almacén tras un previo examen de las propuestas presentadas selecciona a la que, ajustándose sustancialmente a los documentos de licitación, resulte la más conveniente. La adjudicación equivale a la "aceptación" de la "oferta" del licitador [22].

**Adquisición:** Se refiere a toda compra de bienes o contratación de obras o servicios que lleve a cabo el jefe de almacén[23].

**AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma[24].

**Almacén:** Unidad de servicio en la estructura orgánica y funcional de una empresa, industria ó institución, con objetivos definidos de resguardo, custodia, control y abastecimiento de materiales y productos[25].

**Almacenar:** Consiste en la recepción, guarda, despacho y registro de los bienes destinados al uso y/o consumo de las diferentes áreas[26].

**API:** Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones[27].

**Cuarentena de productos:** Consiste en la observación y restricción de la movilización de los biológicos, químicos, farmacéuticos, las materias primas o los materiales de empaque y envase, durante un período determinado, con el objeto de comprobar que cumple con los requisitos establecidos[28].

**Framework:** En el desarrollo de software, un *framework* es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto[29].

**Interfaces de usuario:** Es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar[30].

**Inventario:** Operación que permite saber la cantidad y emplazamiento del material. Relación de bienes y sus valores existentes[31].

**Lenguaje de marcas:** Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. El lenguaje de marcas más extendido es el HTML, fundamento del World Wide Web[32].

**Lenguaje interpretado:** Un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. También se les conoce como lenguajes de script[33].

**Licitaciones:** Es un proceso formal y competitivo de adquisiciones mediante el cual se solicitan, reciben y evalúan oferta para la adquisición de bienes, obras y servicios y se adjudica el contrato al licitador que ofrezca la propuesta más ventajosa. Es la convocatoria que hace una empresa o entidad pública para cubrir alguna necesidad de bienes o servicios[34].

**Lote:** Cantidad de producto terminado identificado con el mismo número de fabricación y que fue elaborado en un solo proceso integral[35].

**Multiplataforma:** Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

**Orientado a Objetos:** Significa que el software se organiza como una colección de objetos discretos que contiene tanto estructura de datos como también un comportamiento[36].

**Plugin:** Es un módulo de hardware o software que añade nuevas funciones a un determinado sistema ya existente.

**Proveedores:** Persona o empresa que abastece de algunos artículos necesarios (materias primas, envases), que posteriormente se venderán, transformarán o elaborarán.

**Requisición:** Solicitud de pedidos, de compras.

**Software:** Se refiere al equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación[37].