

**Universidad de las Ciencias Informáticas**

**Facultad 7**



**Gestión de solicitud de productos a otras  
entidades y reportes para el control de productos del  
módulo Almacén del sistema alas HIS**

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

**Autor:** Abelardo Asensio Callol

**Tutores:** Ing. Juan Alfredo Guerra Góngora  
Ing. Adnier Guerrero Barea

La Habana, junio de 2011  
"Año 53 de la Revolución"

## ***Resumen:***

El Centro de Informática Médica de la Universidad de las Ciencias Informáticas desarrolla el Sistema de Información Hospitalaria alas HIS, con el propósito de lograr centralizar toda la información generada en las diferentes áreas de una institución hospitalaria. Este sistema se encuentra integrado por diversos módulos, entre ellos, Almacén.

El módulo Almacén gestiona gran cantidad de información de una de las áreas más significativas dentro de una institución hospitalaria, el almacén hospitalario. En dicho módulo no se gestionan las solicitudes que pueden ser realizadas a otras entidades, además no cuenta con funcionalidades que le permitan generar a través del sistema reportes asociados al control de producto.

El presente trabajo brinda la propuesta de un sistema informático, que permita la automatización del proceso de gestión de solicitud de productos a otras entidades y reportes para el control de productos de 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 empleó el estilo 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 del módulo Almacén y a su vez mejorar los servicios en los almacenes de las instituciones hospitalarias proporcionando además información actualizada de los productos, que permita la toma de decisiones adecuadas ante una situación en particular.

**Palabras claves:** *Entidad hospitalaria, Almacén hospitalario, Reportes.*

## Índice

<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación teórica</b> .....	<b>5</b>
1.1    Conceptos básicos relacionados con el campo de acción .....	5
1.2    Sistemas automatizados existentes vinculados al campo de acción .....	5
1.3    Tecnologías utilizadas en el proceso de desarrollo.....	10
1.3.1    Modelo .....	11
1.3.2    Vista .....	12
1.3.3    Controlador .....	14
1.4    Lenguaje de Programación.....	15
1.5    Metodologías de Desarrollo .....	15
<b>Capítulo 2: Descripción de la arquitectura</b> .....	<b>18</b>
2.1    Especificación de los requerimientos del software .....	18
3.8.1    Requisitos funcionales.....	19
3.8.2    Requisitos no funcionales.....	20
2.2    Descripción de la arquitectura .....	23
2.3    Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración. ....	24
2.4    Seguridad.....	25
2.5    Vista de despliegue.....	25
2.6    Estrategias de codificación. Estándares y estilos a utilizar .....	26
<b>Capítulo 3: Descripción y análisis de la solución propuesta</b> .....	<b>33</b>
3.1    Descripción de la solución propuesta .....	33
3.2    Valoración crítica del diseño propuesto por el analista .....	34
3.3    Modelo de Diseño .....	34

3.4	Diagramas de clases del diseño .....	35
3.5	Diagramas de interacción .....	43
3.6	Descripción de las nuevas clases u operaciones necesarias.....	45
3.7	Modelo de datos.....	47
3.8	Breve valoración de las Técnicas de validación (Integridad, Normalización de la Base de datos).....	48
3.8.1	Descripción de las tablas.....	49
3.9	Vista de Implementación. (Diagrama de Componentes) .....	51
<b>Capítulo 4: Modelo de Prueba.....</b>		<b>53</b>
4.1	Prueba de caja negra.....	53
4.2	Descripción de los casos de pruebas .....	54
4.2.1	Caso de uso: Crear solicitud de productos a otra entidad.....	55
4.2.2	Caso de uso: Ver detalles solicitud de productos a otra entidad .....	59
4.2.3	Caso de uso: Modificar solicitud de productos a otra entidad.....	61
4.2.4	Caso de uso: Eliminar solicitud de productos a otra entidad .....	63
<b>Conclusiones .....</b>		<b>64</b>
<b>Recomendaciones.....</b>		<b>65</b>
<b>Referencias Bibliográficas .....</b>		<b>66</b>
<b>Bibliografía.....</b>		<b>68</b>
<b>Anexos.....</b>		<b>71</b>

## **Índice de Tablas**

<b>Tabla 1-1 Características de los sistemas .....</b>	<b>9</b>
<b>Tabla 2-1 Requisitos funcionales .....</b>	<b>19</b>
<b>Tabla 2-2 Diagrama de despliegue .....</b>	<b>26</b>
<b>Tabla 2-3 Identación.....</b>	<b>28</b>
<b>Tabla 2-4 Variables y constantes.....</b>	<b>28</b>
<b>Tabla 2-5 Comentarios, separadores, líneas, espacios en blanco y márgenes .....</b>	<b>29</b>
<b>Tabla 2-6 Clases y objetos .....</b>	<b>30</b>
<b>Tabla 2-7 Bases de datos, tablas, esquemas y campos.....</b>	<b>31</b>
<b>Tabla 2-8 Tipo de datos .....</b>	<b>32</b>
<b>Tabla 3-1 Diagrama de paquetes .....</b>	<b>36</b>
<b>Tabla 3-2 Diagrama de Clases del Diseño _ Generar reporte de consumo.....</b>	<b>37</b>
<b>Tabla 3-3 Diagrama de Clases del Diseño _ Generar reporte de déficit .....</b>	<b>38</b>
<b>Tabla 3-4 Diagrama de Clases del Diseño _ Generar reporte de entrada de producto .....</b>	<b>39</b>
<b>Tabla 3-5 Diagrama de Clases del Diseño _ Generar reporte de pérdidas .....</b>	<b>40</b>
<b>Tabla 3-6 Diagrama de Clases del Diseño _ Generar reporte de productos licitados.....</b>	<b>41</b>
<b>Tabla 3-7 Diagrama de Clases del Diseño _ Generar reporte de existencia de productos.....</b>	<b>42</b>
<b>Tabla 3-8 Diagrama de Secuencia _ Generar reporte de déficit .....</b>	<b>43</b>
<b>Tabla 3-9 Diagrama de Secuencia _ Crear solicitud de productos a otra entidad.....</b>	<b>44</b>
<b>Tabla 3-10 Descripción textual de la clase ConsumoControladorReporte.....</b>	<b>45</b>

<b>Tabla 3-11 Descripción textual de la clase ConsumoWrapperReporte.....</b>	<b>47</b>
<b>Tabla 3-12 Modelo de datos .....</b>	<b>48</b>
<b>Tabla 3-13 Tabla del modelo de datos desincorporación_oferta .....</b>	<b>49</b>
<b>Tabla 3-14 Tabla del modelo de datos producto_en_solicitud.....</b>	<b>51</b>
<b>Tabla 3-15 Diagrama de Componentes .....</b>	<b>52</b>

## **Introducción**

Durante los cambios experimentados por la humanidad, el mundo se ha visto enmarcado en un devenir de adelantos científicos y tecnológicos. Desde el pasado siglo la ciencia ha influido de manera positiva en la sociedad trayendo consigo numerosos adelantos beneficiosos para su desarrollo. La Revolución Científico Técnica, como se le conoce, posee como elementos distintivos: la exploración y manipulación de la estructura de la materia, la biotecnología, las comunicaciones, la robótica y la informática.

La informática, como uno de los elementos rectores en el desarrollo actual de la humanidad, está enfocada al estudio del fenómeno de la información, procesamiento, transferencia y utilización. Mediante sistemas informáticos no solo se han logrado automatizar aquellos procesos empresariales, sino también la gestión del conocimiento dirigido a las diferentes ramas de la sociedad y la economía.

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), en la esfera de la salud han sido sistemáticos y progresivos los adelantos científicos con el fin de mejorar la calidad de los servicios que dentro del marco de la salud se brindan. Actualmente se cuenta con novedosos métodos para la gestión administrativa de instituciones hospitalarias, así como sistemas automatizados que faciliten el manejo de información inmersa en estas.

Durante la década del setenta se dieron los primeros pasos en la creación de sistemas automatizados de información médica. Dichos sistemas no cumplían con las exigencias actuales por lo que se comienzan a desarrollar otros conceptos de sistemas de gestión de salud. Los Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés), son ejemplos fehacientes de estos sistemas que facilitan el manejo de la información en instituciones hospitalarias.

Los HIS, están diseñados con el objetivo de mejorar la calidad y la eficiencia del trabajo en los centros de atención médica. Se encargan de almacenar, procesar, recopilar los datos obtenidos en la atención al paciente y en la labor administrativa de las instituciones hospitalarias. Por otra parte permiten la optimización de los recursos humanos y materiales además generan reportes e informes estadísticos que posibilitan la retroalimentación de información.

El Centro de Informática Médica de la Universidad de las Ciencias Informáticas (UCI), desarrolla actualmente un Sistema de Información Hospitalaria, el producto alas HIS. Este responde específicamente al funcionamiento de las diferentes áreas que conforman una institución hospitalaria y que mantienen una estrecha relación entre sí. Entre estas se encuentra el Almacén.

El almacén hospitalario no es más que el área encargada de recibir para su cuidado y protección todos los materiales y suministros necesarios, manteniendo a estos en buen estado para un correcto funcionamiento del hospital.

El módulo Almacén presente en el sistema alas HIS, controla los productos, materiales o útiles necesarios en un hospital, llevando registros diarios de su existencia. Se encarga, además, del despacho de los productos, mediante las solicitudes realizadas por los departamentos, 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 un adecuado control de máximos y mínimos.

En la actualidad dicho módulo no permite la generación de reportes asociados al control de productos. Esto trae consigo que el personal directivo encargado del almacén no cuente con funcionalidades que posibiliten obtener información referente a los productos o útiles existentes, con una visión más detallada y mejor organizada de la situación presente en el área de almacén.

Además, siendo el sistema alas HIS un sistema centralizado que en su composición es capaz de vincular diferentes entidades hospitalarias, no permite efectuar solicitudes entre estas. Esto reduce el radio de abastecimiento y la dependencia absoluta de proveedores o fabricantes de mercancías. Ambos casos repercuten en un deficiente manejo y funcionamiento del área encargada del almacén.

Luego de realizar el análisis correspondiente de la situación actual en el módulo Almacén del sistema alas HIS, se define como **problema a resolver**: ¿Cómo facilitar la gestión de solicitud de productos a otras entidades y reportes para el control de productos del módulo Almacén del sistema alas?



El **objeto de estudio** lo constituye el proceso de gestión de la información del módulo Almacén del sistema alas HIS, cuyo **campo de acción** se enmarca en el proceso de gestión de solicitud de productos a otras entidades y reportes para el control de productos del módulo Almacén del sistema alas HIS.

Para darle solución a los problemas anteriormente planteados, se determina como **objetivo general** del presente trabajo: Desarrollar el proceso de gestión de solicitud de productos a otras entidades y reportes para el control de productos del módulo Almacén del sistema alas HIS, que facilite la gestión de la información en el área de almacén de las instituciones hospitalarias.

Para darle cumplimiento al objetivo general se definen las siguientes **tareas a desarrollar**:

1. Analizar los procesos de negocio asociados a la gestión de solicitud de productos y reportes para el control de productos en el área de Almacén de las instituciones hospitalarias.
2. Analizar las tendencias actuales, ventajas y desventajas sobre el proceso de gestión de solicitud de productos a otras entidades y reportes para el control de productos en las instituciones hospitalarias.
3. Aplicar la metodología, plataforma, tecnologías, librerías, herramientas y pautas definidas por el Departamento de Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Obtener mediante el Proceso Unificado de Desarrollo, los artefactos correspondientes a los flujos de trabajo de "Implementación" y "Pruebas".
5. Desarrollar el proceso de gestión de solicitud de productos a otras entidades y reportes para el control de productos del módulo Almacén del sistema alas HIS.
6. Desarrollar el proceso de pruebas para obtener la correspondiente Acta de Liberación.

Con el presente trabajo se prevé facilitar la comprensión de la situación real del almacén a través de reportes, trayendo consigo que el personal encargado de esta área de la entidad hospitalaria tenga acceso a la información de los productos o útiles que en ella se controlan. Brinda la posibilidad de ampliar el radio de abastecimiento de la entidad hospitalaria y la no dependencia absoluta de los proveedores asociados al sistema. La posibilidad de gestionar las solicitudes de productos a otras entidades y la

creación de reportes para el control de productos, serían los mayores resultados que se obtendrían al término de esta investigación.

El presente documento se encuentra estructurado en cuatro capítulos.

**Capítulo 1: Fundamentación teórica:** Contiene un estudio del estado del arte de sistemas similares a la solución propuesta. Se describen las tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo.

**Capítulo 2: Descripción de la arquitectura:** Se refiere al patrón arquitectónico utilizado, además de los requerimientos del sistema, la seguridad, estrategias y estándares a utilizar para el cumplimiento del objetivo general.

**Capítulo 3: Descripción y análisis de la solución propuesta:** En este capítulo se realiza una breve descripción de los procesos identificados para darle solución a la problemática identificada. Además se muestran los diagramas, clases representativas al modelo y componentes a implementar.

**Capítulo 4: Modelo de pruebas:** Por último, se muestran los casos de pruebas realizados a la aplicación con el objetivo de validar el correcto funcionamiento verificando así el cumplimiento de los objetivos trazados.

## **Capítulo 1: Fundamentación teórica**

En este capítulo se abordan principales conceptos que sirven de ayuda para la interpretación y entendimiento del problema a resolver. Se realiza un estudio relacionado con la situación a nivel mundial de los sistemas de información hospitalaria o sistemas de almacenes que gestionen solicitudes de productos a entidades, así como los reportes generados en el control de productos en los almacenes. Además se definen tecnologías, metodologías y herramientas utilizadas para dar respuesta al objetivo general propuesto.

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

#### **Entidad hospitalaria**

Institución o centro de salud destinado a la prevención, diagnóstico, tratamiento y si es posible, cura de enfermedades físicas y/o psíquicas, incluyendo la realización de prácticas quirúrgicas. Integrada por diversos servicios que manejan datos médicos en las labores de atención al paciente.

#### **Almacén hospitalario**

Área perteneciente a una entidad hospitalaria designada para mantener el control de medicamentos, utensilios y productos sanitarios.

#### **Reporte**

Información estructurada generada por el sistema, relacionada con los productos o útiles existentes en el almacén hospitalario.

### **1.2 Sistemas automatizados existentes vinculados al campo de acción**

#### **MEDINOUS**

Es un sistema de gestión hospitalaria desarrollado para ofrecer beneficios a hospitales multi-especialidad y clínicas, el cual permite cubrir una amplia gama de administración y procesos. Proporciona información relevante en el hospital para apoyar la toma eficaz de decisiones para el cuidado del paciente, la administración del hospital y la contabilidad financiera crítica, en un flujo continuo.

Este sistema cuenta con varios módulos encargados de la gestión de la información. Uno de los módulos con que cuenta permite llevar el control del almacén general e Inventario. Este módulo en particular se encarga de la gestión de stocks, el número de productos, mantenimiento de stock, compra de artículos, solicitud en línea de stock de varios sub-almacenes, mantenimiento del material en los diferentes sub-almacenes, compra de acciones a través de la generación de Consulta, recepción y emisión de acciones externas del hospital y la posibilidad de incluir y hacer un seguimiento de regalo o donación de acciones.

Además realizar solicitudes de archivo de sub-almacenes, gestión correspondiente de citas, generación de pedidos de compra. La gestión de proveedores y la facilidad de pago a proveedores. [1]

### **SIGEHOS**

Es una suite de herramientas informáticas, que contribuye a la sistematización de tareas, a la mejora de los circuitos administrativos, e informatiza procedimientos hospitalarios normalizados, basados en políticas y estándares de salud. Cada herramienta se considera como un módulo del sistema global, dichas herramientas se interconectan e interactúan entre sí, básicamente; mediante una base de datos centralizada. La producción de los módulos presentes dentro de SUMINISTROS está los que gestionan los ítems de productos y su existencia (stock), solicitudes de provisión de productos a compras o programas sociales de distribución de insumos, pedidos y distribuciones a servicios, distribuciones de productos por servicios, y sus respectivos costos entre otros.

CONSUMO DEL SERVICIO, estos gestionan ingresos por proveedores, por productos, cantidades, precios promedio, distribuciones y servicios. [2]

### **GESTLOG**

Es un sistema para almacenes que automatiza la gestión de compras, distribución y control de almacén y stocks de farmacia. 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 el de no permitir generar reportes y ser una aplicación de escritorio. [3]

## **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 un módulo encargado de las actividades 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.

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.

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. [4]

### **KEWAN-COSMOSALUD**

Es un sistema de información hospitalario (HIS), de apoyo a la gestión hospitalaria, que posibilita:

- Procesos administrativos de admisión de pacientes, generación de documentos, imprimir etiquetas.
- Obtención de indicadores de gestión: actividad, demora.
- Gestión de áreas básicas: admisión, archivo, facturación.
- Gestión de sistemas departamentales: farmacia, rehabilitación.
- Además cubre la funcionalidad de atención primaria y permite la gestión de la información clínica: HCE.

Realizado por la empresa española Dimensión Informática S.L con mayor experiencia en el desarrollo, mantenimiento e implantación de soluciones en el ámbito de la salud. Este sistema dentro de sus características está: ser multicentro, y cuenta con un total de 18 módulos, dentro de los cuales está el módulo Central de solicitudes. [5]

A continuación se presenta una tabla relacional con las características presentes en los sistemas similares encontrados.

<b>Sistemas</b>	<b>Software Propietario</b>	<b>Multiplataforma</b>	<b>Aplicación de Escritorio</b>	<b>Aplicación Web</b>
<b>MEDINOUS</b>	X		X	
<b>SIGEHOS</b>	X			
<b>GESTLOG</b>	X		X	
<b>CNT PACIENTES</b>	X		X	
<b>KEWAN-COSMOSALUD</b>	X		X	X

**Tabla 1-1 Características de los sistemas**

Los sistemas anteriormente descritos en su mayoría concuerdan con el sistema propuesto. Presentan funcionalidades similares a desarrollar. Estos, son sistemas propietarios, aplicaciones de escritorio, características que hacen que sean totalmente dependientes de la plataforma especificada para cada sistema, además del elevado costo monetario por conceptos de licencia al adquirirlos. La generación de reportes estadísticos presentes en estos sistemas se hacen a través de tablas donde la información que aparece es rígida y en ocasiones escasa por contar con poco espacio para mostrar toda la información.

La utilización de alguno de estos sistemas no resolvería el problema científico identificado. El mismo, está centrado en cómo realizar solicitudes de productos entre entidades hospitalarias relacionadas entre sí, y la posibilidad de contar con una visión más detallada de la situación real del almacén a través de reportes, permitiendo acceder a la información sin tener que hacer un uso exclusivo de la aplicación. Por lo anteriormente explicado no es necesario integrar las funcionalidades que poseen estos sistemas, sino,

utilizar lo aprendido en el estudio de los mismos, para mejorar la calidad y aumentar la eficiencia de la solución.

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

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

Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java. Esta plataforma fue desarrollada por la empresa Sun Microsystems recientemente adquirida por Oracle Corporation. Java Enterprise Edition (o JEE por sus siglas en inglés) proporciona el estándar de desarrollo para lograr aplicaciones con un alto nivel de calificación.

El lanzamiento de esta edición, trajo consigo beneficios añadidos, por ejemplo, el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de las tareas de mantenimiento de bajo nivel. [6]

#### **Java virtual machine (JVM)**

Como su nombre lo indica, es una máquina virtual de java de proceso nativo, es decir, solo se ejecuta en una plataforma específica. Esta máquina es capaz de interpretar y ejecutar instrucciones en un lenguaje binario especial (el Bytecode) el cual no es más que un verdadero código de máquina. [7]



## **Modelo**

### **Hibernate**

Es una herramienta que facilita el mapeo entre los atributos de un modelo de objetos de una aplicación y los de una base de datos relacional. Archivos declarativos (XML) o anotaciones en los Beans de las entidades logran establecer dicho mapeo. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Hibernate, como software libre, *open source* (código abierto), es distribuido bajo licencia GNU LGPL. Permite expresar consultas en *Hibernate Query Language* o (HQL por sus siglas en inglés), así como en SQL nativo o utilizando criterios orientado a objetos. Brinda filtros para trabajar con datos históricos, regionales o condicionados por permisos. [8]

### **Enterprise Java Bean 3 (EJB 3)**

Los Enterprise Java Bean (EJB, por sus siglas en inglés) representan Interfaces de Programación de Aplicaciones (API, por sus siglas en inglés) que forman parte del estándar de construcción de aplicaciones empresariales (JEE). Los EJB proporcionan un modelo de componentes distribuidos en el estándar del lado del servidor. Permite al programador dotar de un modelo que permita abstraerse de los problemas generales en una aplicación empresarial (conurrencias, transacciones, persistencia, seguridad, etc.) para centrarse en la lógica del negocio en sí.

Los EJBs disponen de un contenedor en el servidor de aplicaciones que es el encargado de almacenar y recuperar los datos del objeto entidad mediante el mapeo entre las columnas de la tabla de la base de datos y los atributos del objeto. [9]

## **Vista**

### **XHTML**

Es el acrónimo en inglés de e**X**tensible **H**ipertext **M**arkup **L**anguage (lenguaje extensible de marcado de hipertexto). Su creación está formada por la unión de **XML** (Extensible Markup Language) y **HTML** (Lenguaje de marca de hipertexto) para la creación de páginas web estandarizadas. Este lenguaje es utilizado de forma nativa en Java Server Face. Básicamente vela por que las páginas webs creadas estén correctamente formadas y que todas las etiquetas utilizadas estén cerradas. [10]

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

Es un *framework* (marco de trabajo) orientado al patrón arquitectónico Modelo Vista Controlador (MVC) basado en componentes para el desarrollo web y dirigido por eventos. [11]

### **JavaScript**

Es un lenguaje de scripting o de procesamiento por lotes, liviano y basado en objetos no tipados. Sirve para incorporar interactividad y dinamismo en las aplicaciones web, permitiendo el desarrollo de interfaces de usuarios mejoradas. [12]

### **CSS**

Acrónimo en inglés de **C**ascade **S**tyle **S**heet (Hoja de estilo en cascada). Es un estándar para dar formato y estilo a las páginas web, que permite separar el contenido de la presentación. Su objetivo es de minimizar, en la medida de lo posible, las múltiples diferencias visuales que se dan a la hora de mostrar una misma página web. [13]

### **Facelets**

Representa un conjunto de plantillas que permiten crear vistas JSF usando estilos y templates HTML. Reduce la cantidad de código necesario para integrar los componentes en la vista, y no requiere un contenedor web. [14]

### **Ajax4JSF**

Es una librería de código abierto que se integra totalmente a la arquitectura de JSF. Mediante ésta, se puede recargar componentes individualmente sin la necesidad de actualizar la página completa a través de peticiones asíncronas al servidor. Estas peticiones son realizadas por el objeto XMLHttpRequest obtenido del contenedor web. [15]

### **Richfaces**

Es una biblioteca enriquecida de componentes con capacidad AJAX para aplicaciones JSF. Richfaces incluye entre otras características ciclo de vida, validaciones y gestión de recursos estáticos y dinámicos. Sus componentes permiten que los desarrolladores puedan ahorrar tiempo y aprovechar las características de éstos para crear aplicaciones web. [16]

### **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. [17]

## **Controlador**

### **Seam**

Poderoso y moderno *framework* que combina los dos framework EJB3 y JSF además de BPM (Business Process Management). Seam introduce un nuevo concepto, el de *Contexto*, donde las acciones que son ejecutadas por el usuario estarán asociadas a éstos. Se integra con librerías de controles de código abierto basadas en JSF como Richfaces e ICE Faces.

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. [18]

### **Java Business Process Manager (JBPM)**

Es una plataforma para manejar los procesos de negocio en java y lenguajes de procesos ejecutables, cubriendo desde gestión de procesos de negocio (BPM) bajo workflow hasta orquestación de servicio. Debido a su flexibilidad y extensibilidad con la que cuenta posee 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 flexible. Debido a que el manejo de la información varía en dependencia de la entidad, es necesario que la aplicación se adapte a cualquier entorno. Para lograr una buena integración se hace uso de JBoss Drool. [19]

### **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 ingles BRMS). Proporciona una plataforma unificada e integrada para las reglas de flujo de trabajo y el procesado de eventos. [20]

## **1.4 Lenguaje de Programación**

### **Java**

Es un lenguaje de programación, el cual hace uso del paradigma de la Programación Orientada a Objetos (POO). Es un lenguaje interpretado e independiente a la arquitectura. Programas desarrollados por este lenguaje presentan ventajas frente a los desarrollados en otros lenguajes como C/C++ ya que pueden ser ejecutados en cualquier plataforma. Java comprende un amplio espectro que va desde la programación tradicional, distribuida, GUI hasta desarrollar aplicaciones web dinámicas, con acceso a bases de datos, con cualquier tipo de conexión de red entre cualquier sistema. [6]

## **1.5 Metodologías de Desarrollo**

### **Proceso Racional Unificado (RUP)**

Es un proceso de desarrollo de software que proviene del trabajo con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) y el asociado Proceso Unificado de Desarrollo de software. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es un modelo que identifica cuatro fases diferentes: Inicio, Elaboración, Construcción y Transición. Además este proceso de software está definido por tres características fundamentales:

- 1. Guiado por los casos de uso:** Los casos de uso constituyen un elemento integrador y una guía de trabajo. Estos no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor durante todo el proceso de desarrollo.
- 2. Centrado en la arquitectura:** La arquitectura de un sistema es el factor fundamental en todo proceso de desarrollo. Permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.
- 3. Iterativo e incremental:** RUP divide el proceso de desarrollo en ciclos, teniendo pequeñas porciones del producto al culminar cada uno, estos se dividen en fases que finalizan con un hito

donde se deberá tomar una decisión importante. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. [21]

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

Lenguaje de modelado gráfico de sistemas que se utiliza para especificar, visualizar, construir y documentar un sistema de software orientado a objetos (OO). Está compuesto por elementos gráficos que se agrupan y conforman diagramas de acuerdo a reglas especificadas por el lenguaje. Un modelo UML describe el comportamiento de un sistema pero no dice cómo implementarlo. [22]

### **Herramientas de Desarrollo**

#### **Eclipse**

Eclipse es un entorno de desarrollo integrado (IDE por sus siglas en inglés) de código abierto multiplataforma para desarrollar *Aplicaciones de Cliente Enriquecido*, opuesto a las aplicaciones *Cliente-liviano* basadas en navegadores. Fue desarrollado por IBM y actualmente es desarrollado por la Eclipse Foundation, organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto. Este IDE cuenta con un editor de código, un compilador/intérprete y un depurador. Sirve como IDE para 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. [23]

#### **JBoss Tools**

JBoss Tools es un conjunto de plugins que se integra al Eclipse para JBoss y tecnologías relacionadas, que son el soporte para Hibernate, JBoss AS, Drools, JBPM, JSF, (X) HTML, Seam, Smooks, JBoss ESB, JBoss Portal y más. [24]

### **PgAdmin III**

Plataforma de desarrollo de código abierto para la administración de bases de datos PostgreSQL. Esta aplicación puede ser ejecutada en cualquier plataforma dígase (Linux, FreeBSD, Solaris, Mac OSX y Windows). Está diseñado para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta el desarrollo de bases de datos complejas. Incluye un editor de resaltado de sintaxis SQL, un editor de código del lado del servidor, soporte para el motor de replicación Slony-I y mucho más. Desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo, está disponible en más de una docena de idiomas. [25]

### **Visual Paradigm para UML**

Visual Paradigm para UML (VP-UML) es una herramienta de diseño UML y herramienta CASE UML diseñada para la ayuda al desarrollo de software. Bajo licencia gratuita VP-UML soporta los principales estándares de la industria tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI. Ofrece un conjunto de herramientas necesarias para el desarrollo del modelado de datos, la planificación de controles, el modelado de clases entre otras. Permite la integración con Eclipse además, permite la generación de documentación y de código base para diferentes lenguajes de programación como java y su exportación con HTML. [26]

Como resultado del estudio realizado en este capítulo, se puede concluir que los sistemas existentes vinculados al campo de acción no cumplen con las necesidades actuales del módulo almacén del Sistema de Información Hospitalaria alas HIS. Por lo que se evidenció la necesidad de desarrollar funcionalidades en la que los usuarios puedan crear solicitudes a otras entidades y obtener reportes de los productos con que cuenta el almacén. Que además, esté debidamente documentado para desarrollos futuros y esté desarrollado con tecnologías y herramientas no privativas. Se justificó la aplicación de los patrones de diseño y el empleo de las tecnologías y herramientas propuestas para la obtención de las funcionalidades

## **Capítulo 2: Descripción de la arquitectura**

En el presente capítulo se tratarán temas como la seguridad, estrategias de integración, despliegue, la especificación de los requerimientos funcionales y no funcionales que responden a la necesidad del sistema. Se describe la arquitectura empleada, así como la estrategia de codificación a emplear.

### **2.1 Especificación de los requerimientos del software**

Un requerimiento o requisito de software como también se le denomina, es un conjunto de necesidades y condiciones presentadas por él o los clientes, donde se especifican las funcionalidades y especificidades que debe poseer el sistema para su correcto funcionamiento. Estos requisitos son divididos en dos clasificaciones, Requisitos funcionales y Requisitos no funcionales.



### 2.1.1 Requisitos funcionales

Se entiende como requisito funcional a la capacidad o condiciones internas que un sistema informático debe cumplir acorde con las necesidades planteadas por el cliente.

**El sistema deberá permitir:**

REQUISITOS FUNCIONALES	
RF1 – Desincorporar productos a ofertar.	RF10 – Recepcionar productos de solicitud de productos.
RF2 – Crear solicitud de productos a otra entidad.	RF11 – Autorizar despacho de solicitud a otra entidad.
RF3 – Ver detalles de la solicitud de productos.	RF12 – Generar reporte de pérdida.
RF4 – Modificar solicitud de productos.	RF13 – Generar reporte de consumo.
RF5 – Eliminar solicitud de productos.	RF14 – Generar reporte de existencia de producto.
RF6 – Aprobar solicitud de productos.	RF15 – Generar reporte de productos licitados.
RF7 – Rechazar aprobación de solicitud de productos.	RF16 – Generar reporte de entrada de productos.
RF8 – Ver disponibilidad de productos solicitados.	RF17 – Generar reporte de productos en déficit.
RF9 – Aceptar solicitud de productos a otra entidad.	

**Tabla 2-1 Requisitos funcionales**

### **2.1.2 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 todas las funcionalidades requeridas, 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.

- **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. [26]

- **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. [26]

- **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. [26]

## **Hardware**

- **Estaciones de trabajo**

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 7 o superior o Firefox 2.x). Por lo que se escogieron estaciones de trabajo de 256Mb de memoria RAM y un microprocesador de 2.0Hz con Sistema operativo Linux. [26]

- **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. [26]

- 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. [26]

## **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. [26]

## **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. [26]

## **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. [26]

## **Interfaz**

- **Interfaces de usuario**

- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.
- La interfaz contará con teclas de función y menús desplegados que faciliten y aceleren su utilización.
- La entrada de datos incorrecta será detectada claramente e informada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español. [26]

## **2.2 Descripción de la arquitectura**

La arquitectura de software es la columna vertebral del software y el factor principal de cualquier aplicación. De ahí que una correcta arquitectura proveerá fortaleza y será una guía fundamental para el desarrollo exitoso del sistema.

Para el desarrollo de las funcionalidades correspondientes, se adopta el patrón arquitectónico Modelo Vista Controlador siguiendo la arquitectura definida en el desarrollo del módulo Almacén. Este patrón presenta ventajas como la separación de los datos de la aplicación en tres partes fundamentales: el modelo, la vista y el controlador.

El modelo de datos o también conocido como "*capa de acceso a datos*", representa las relaciones existentes entre todos los Beans persistentes del sistema que describen los elementos de la realidad que intervienen en el problema. Un Bean no es más que una clase simple de java que no depende de un

framework o un *Plain Old Java Object* (POJO) como también se le conoce, que en conjunto con *Hibernate*, visto en el capítulo anterior, facilitan el acceso a los datos.

La vista o “*capa de presentación*” alberga todos los archivos conocidos como “*interfaces*” con las que los usuarios interactúan con el sistema. A través de estas interfaces se muestran las informaciones correspondientes de los datos del modelo. Estos archivos principalmente con extensión XHTML están compuestos por componentes JSF, JavaScript, Ajax4jsf entre otros, los cuales enriquecen el diseño de las mismas.

El controlador o “*capa de negocio*” se compone de todas aquellas clases controladoras encargadas de llevar la lógica del negocio. En estas se gestionan eventos guiados a través de reglas definidas. Se realizan validaciones y se maneja el flujo de los datos desde la capa de presentación hasta la capa de acceso a datos y viceversa.

El uso de este patrón de diseño proporciona un mayor manejo y una mejor organización estructural del código y del sistema en general. Esto trae consigo que un cambio realizado en alguna de estas partes o capas no afecte en su totalidad a las demás.

### **2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados. Estrategias de integración.**

En la construcción de grandes soluciones informáticas, como es el caso del Sistema de Información Hospitalaria alas HIS, la reutilización de código es uno de los aspectos más beneficioso a tener en cuenta. Con esto se reduciría gran parte del tiempo y esfuerzo empleado para la culminación del sistema. Permite lograr uniformidad, limpieza y organización en el código, incrementando la calidad y eficiencia del producto terminado. El sistema alas HIS por su característica modular, es decir constituido por módulos que responden a un funcionamiento específico dentro de un área en la institución hospitalaria.

## **2.4 Seguridad**

La seguridad de la información es un aspecto fundamental a tener en cuenta y la cual debe estar presente en todo sistema informático. Un sistema seguro garantizaría la confidencialidad, la integridad y la disponibilidad de los datos. Es por ello que el sistema basa su seguridad en cuanto a diferentes aspectos aquí relacionados:

- El acceso al sistema se mantendrá a través del control de usuario, garantizando el acceso de los mismos solo a los niveles establecidos de acuerdo a la función que realizan.
- Existirá un administrador general del sistema, el cual tendrá permiso total de la aplicación y se encargará además de crear usuarios, asignarles los permisos adecuados según su cargo.
- 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.
- Todas las acciones realizadas en el sistema se registrarán en forma de (log), llevando así el control de las actividades de cada usuario en todo momento.
- Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude.
- 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.

## **2.5 Vista de despliegue**

Es un modelo de objeto que indica la situación física de los componentes lógicos 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.

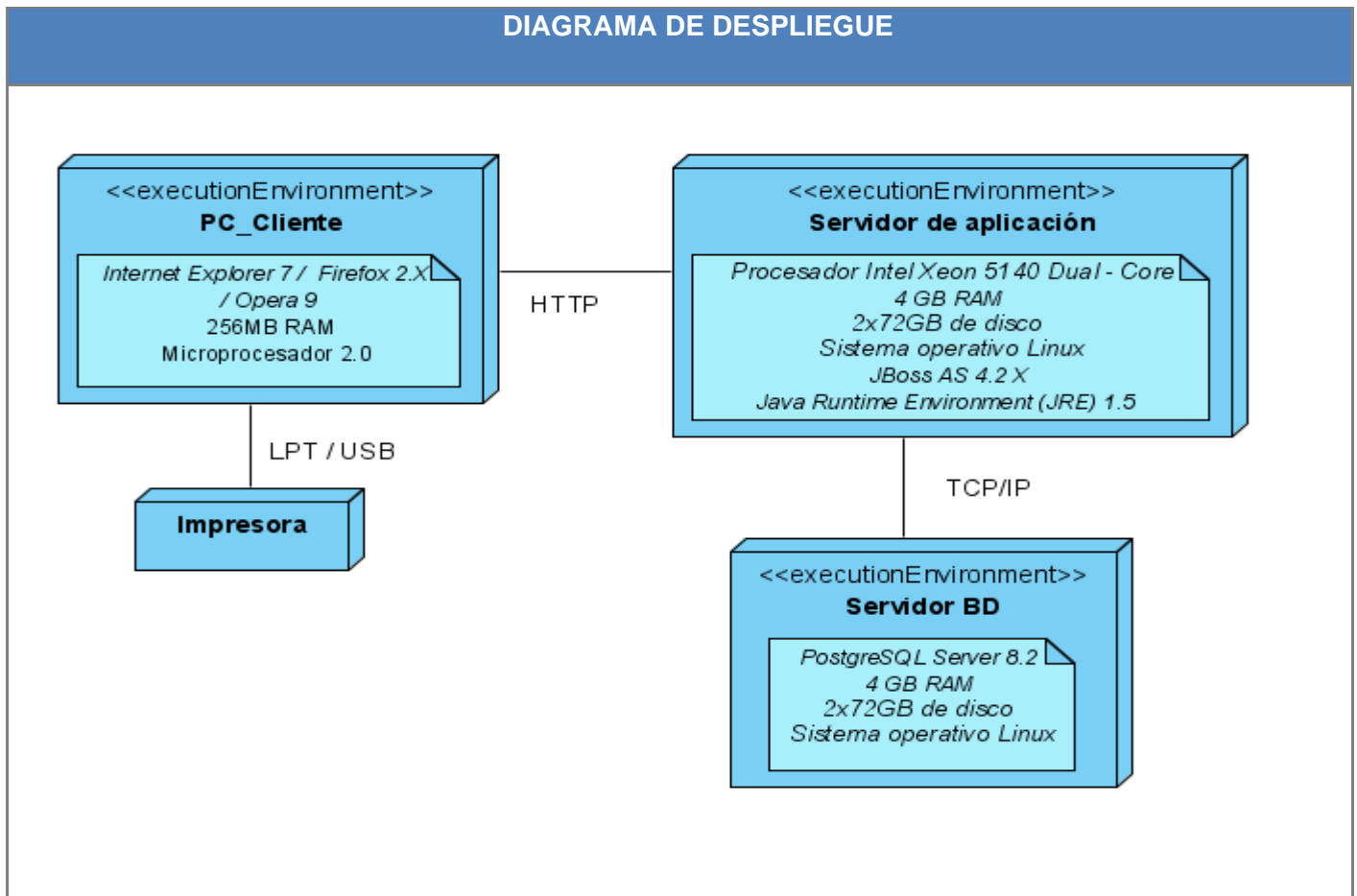


Tabla 2-2 Diagrama de despliegue

## 2.6 Estrategias de codificación. Estándares y estilos a utilizar

La estructura y organización dentro de un sistema informático juegan un papel fundamental para la obtención de un producto con calidad. La utilización de estándares y estilos de codificación ayudan tanto a los desarrolladores actuales como a los desarrolladores futuros encargados de actualizaciones del sistema, no solo en la gestión de la configuración sino también en la actualización de uno o varios procesos por los que está conformado el sistema.



Los estándares de codificación son precisamente los encargados de la correcta estructuración del código. Se pudieran definir como un conjunto de reglas aplicables para lograr la uniformidad y claridad en los códigos generados por el grupo de desarrolladores encargados de un sistema informático.

Para darle cumplimiento a las funcionalidades propuestas se utilizaron dos estándares de codificación:

- **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*. [27]
- **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*. [27]

A continuación se muestran las restricciones de nomenclatura y estándares de codificación que fueron utilizados basándose en los estándares antes mencionados: [27]

IDENTACIÓN	
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.	
<b>Inicio y fin de bloque</b>	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.
<b>Aspectos generales</b>	El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla. Los inicios ( { ) y cierre ( } ) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.

**Tabla 2-3 Identación**

<b>VARIABLES Y CONSTANTES</b>		
<b>Apariencia de variables</b>	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, en caso de que sea un nombre compuesto se empleará notación CamellCasing**. Ejemplo: sNombrePaciente.
<b>Apariencia de constantes</b>	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
<b>Aspectos generales</b>	Nombres de las variables y constantes	El nombre empleado, debe permitir que con solo leerlo se conozca el propósito de la misma.

**Tabla 2-4 Variables y constantes**

<b>COMENTARIOS, SEPARADORES, LÍNEAS, ESPACIOS EN BLANCO Y MÁRGENES.</b>		
<b>Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con solo leerlo una vez.</b>		
<b>Ubicación de comentarios</b>	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.
<b>Líneas en blanco</b>	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.
<b>Espacios en blanco</b>	Entre operadores lógicos y	Se recomienda usar espacios en blanco entre

	aritméticos.	estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto.
<b>Aspectos generales</b>	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.
<b>Sobre los espacios en blanco</b>		No se debe usar espacio en blanco: <ul style="list-style-type: none"> <li>• Después del corchete abierto y antes del cerrado de un arreglo.</li> <li>• Después del paréntesis abierto y antes del cerrado.</li> <li>• Antes de un punto y coma.</li> </ul>

**Tabla 2-5 Comentarios, separadores, líneas, espacios en blanco y márgenes**

## CLASES Y OBJETOS

**Objetivo: Nombrar las clases e instancias de forma estándar para todas las aplicaciones.**

<b>Apariencia de clases y objetos</b>	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.
---------------------------------------	----------------------------	---

<b>Apariencia de atributos</b>	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**.
<b>Declaración de parámetro en funciones</b>	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.
<b>Aspectos generales</b>	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 2-6 Clases y objetos**

<b>BASES DE DATOS, TABLAS, ESQUEMAS Y CAMPOS</b>		
<b>Apariencia de los campos</b>	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
<b>Nombre de los campos</b>	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
<b>Aspectos generales</b>	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 2-7 Bases de datos, tablas, esquemas y campos**

<b>TIPO DATOS</b>	<b>PREFIJO</b>	<b>EJEMPLO</b>
Int	i	iCantPacientes
flota	f	fPesoPaciente
double	d	dPesoCarro
bool	b	bPacienteActivo
string	s	sNombrePaciente
char	c	cLetra
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
Struct	st	stUnaStruct

**Tabla 2-8 Tipo de datos**

En el presente capítulo se llevó a cabo la descripción de unos de los aspectos fundamentales en la construcción de un software: La arquitectura. Se dieron a conocer los requerimientos necesarios obtenidos que benefician tanto al sistema como a los usuarios finales. Además se realizó el análisis de posibles implementaciones y componentes que se reutilizaron con vista a facilitar la implementación de las funcionalidades en un menor tiempo de desarrollo.

## **Capítulo 3: Descripción y análisis de la solución propuesta**

En el presente capítulo se realiza una detallada descripción de la aplicación propuesta. Se presenta una visión física y lógica del sistema, mostrando los diagramas de clases del diseño y los de interacción haciendo énfasis en el diagrama de secuencia, que permiten una efectiva comprensión en la etapa de implementación. Se expone el diagrama de paquetes y de componentes, además del modelo de datos usado para definir el mapeo entre las clases del diseño y la estructura de los datos. Se describen las clases u operaciones necesarias, así como las entidades asociadas a los diagramas presentes en este capítulo.

### **3.1 Descripción de la solución propuesta**

En el estudio realizado al módulo Almacén del Sistema de Información Hospitalaria alas HIS, se identificaron dos procesos fundamentales que contribuirían al cumplimiento del objetivo general. El proceso Solicitar productos a otra entidad y el proceso Generar reportes asociados al control de producto.

**Solicitar productos a otra entidad:** Este proceso comienza una vez que una de las entidades asociadas al sistema pone a disposición una serie de productos en oferta debido a su próximo vencimiento o al exceso de los mismos en el almacén (**ver Anexo 1**). El sistema brinda la posibilidad de crear solicitudes de los productos ofertados (**ver Anexo 2**) además realizar consultas y selección de los mismo para su consumo (**ver Anexo 3**). La entidad solicitante realiza una solicitud de productos ofertados y es enviada hacia la entidad que realiza la oferta (**ver Anexo 4**). Esta solicitud puede ser aceptada o rechazada por el Administrativo de la entidad que oferta los productos. En caso de ser aceptada, la solicitud pasa al almacenero el cual autoriza la cantidad parcial o total solicitada para luego ser despachada (**ver Anexo 5**). Una vez despachada, los productos pasan a ser recibidos por la entidad solicitante.

En caso de ser rechazada la solicitud, el Administrativo emite una notificación a los involucrados, dígase, Administrativo y Almacenero del almacén de la entidad solicitante explicando las causas por la cual no fue aceptada la solicitud (**ver Anexo 6**).

**Generar reportes asociados al control de producto:** Para llevar un correcto abastecimiento y a su vez funcionamiento de la entidad hospitalaria, se hace necesario controlar y gestionar toda la información

asociada a los productos existentes en la institución. Para obtener una visión detallada y mejor organizada de la información el Almacenero hará uso de las funcionalidades para generar reportes asociados al control de producto. Mediante criterios de búsqueda obtendrá información sobre la entrada, existencia, pérdida, consumo y déficit de los productos, logrando, si así lo desea, exportar a formatos como PDF, Excel o Word. (ver Anexo 7)

### **3.2 Valoración crítica del diseño propuesto por el analista**

Para realizar un mejor análisis de la aplicación se comienza por una serie de modelados que conducen a una refinación de los requisitos y a una representación completa del diseño. El modelo de análisis es la primera representación técnica de un sistema. Las actividades que en él se detallan se deben llevar a una revisión para evaluar su calidad, integridad y consistencia. De igual forma se continúa haciendo durante el diseño, el cual representa una vista interna del sistema descrita con el lenguaje de los desarrolladores.

En general los diagramas de clases del diseño, estarán compuestos por páginas servidoras que construyen una página cliente, esta a su vez, estará compuesta por uno o varios formularios los cuales podrán actualizar directamente las entidades, además puede realizar una petición hacia la página servidora o enviar los datos introducidos. La página servidora invocará métodos o responsabilidades presentes en las clases controladoras que según su responsabilidad pueden modificar las entidades.

### **3.3 Modelo de Diseño**

Es un modelo de objetos que describe la realización física de los casos de uso. Se centra 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. [28] El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. De hecho cuando es muy grande, la implementación puede realizarse por un generador de código.

Durante la confección del diseño se utilizan patrones de diseño. Estos describen una estructura que resuelve un problema en la elaboración, relaciones entre componentes o los mecanismos que permitan la comunicación de componente a componente dentro de un contexto específico, esto puede verse en la manera en que se utiliza y se aplica. Un patrón se emplea con la finalidad de describir la esencia de la



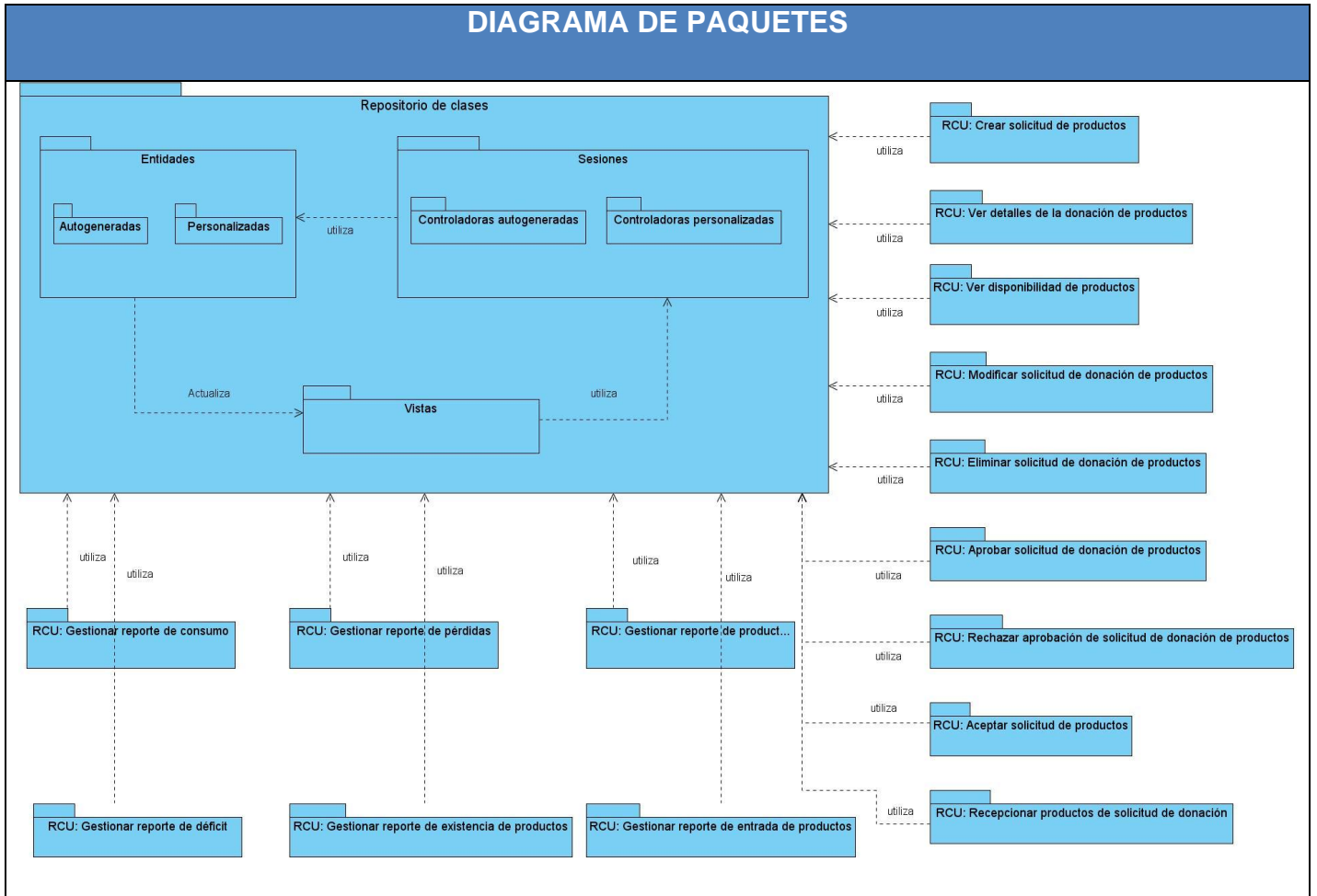
solución del problema y poderla usar en diferentes situaciones, sin la necesidad de hacerlo nunca dos veces de la misma forma.

Entre los patrones de diseño que se utilizaron se encuentran los Patrones de Software para la Asignación General de Responsabilidad (GRASP, por sus siglas en inglés). Estos patrones constituyen un apoyo para entender el diseño de objetos. Se utilizan con el propósito de asignar responsabilidades a las clases que son definidas en el diseño, como por ejemplo: iniciar acciones entre objetos, controlar actividades entre ellos. Los patrones GRASP se encuentran divididos en: experto, creador, controlador, alta cohesión, bajo acoplamiento.

### **3.4 Diagramas de clases del diseño**

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. En su contenido se expone generalmente las clases presentes en el diseño que son una guía para la implementación, asociaciones y atributos, interfaces, con sus operaciones y constantes, métodos, información sobre los tipos de los atributos, navegabilidad y dependencias.

A continuación se presenta el diagrama de paquetes del sistema propuesto.

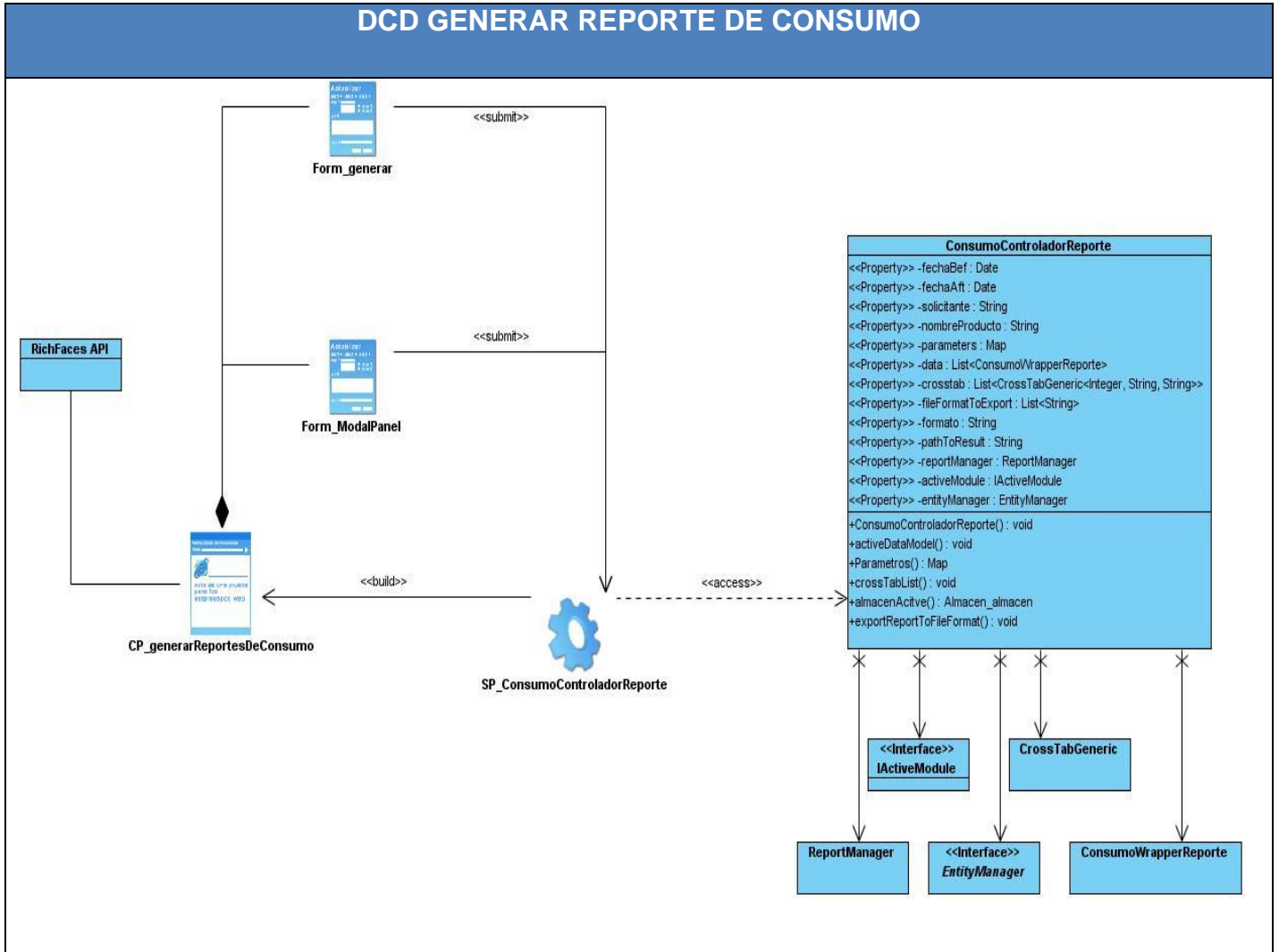


**Tabla 3-1 Diagrama de paquetes**

El paquete Repositorio de clases, está compuesto por los paquetes *Entidades*, *Sesiones* y *Vistas*. El paquete *Vistas* contiene los archivos de contenidos web referentes a las páginas clientes y los formularios que las componen. En el paquete *Entidades* se aprecia el conjunto de clases Autogeneradas, el cual contiene a todas las entidades generadas desde la base de datos mediante el proceso de mapeo, mientras que las clases Personalizadas, cuenta con las especializaciones que se le realizarán a las clases autogeneradas, ya sea haciendo uso de la herencia y el polimorfismo. Por último se encuentra el paquete *Sesiones* que contiene las clases Controladoras de procesos de igual forma autogeneradas y

personalizadas mediante el mapeo de la base de datos haciendo uso de relaciones de composición o herencia.

**Diagramas de clases del diseño**



**Tabla 3-2 Diagrama de Clases del Diseño \_ Generar reporte de consumo**

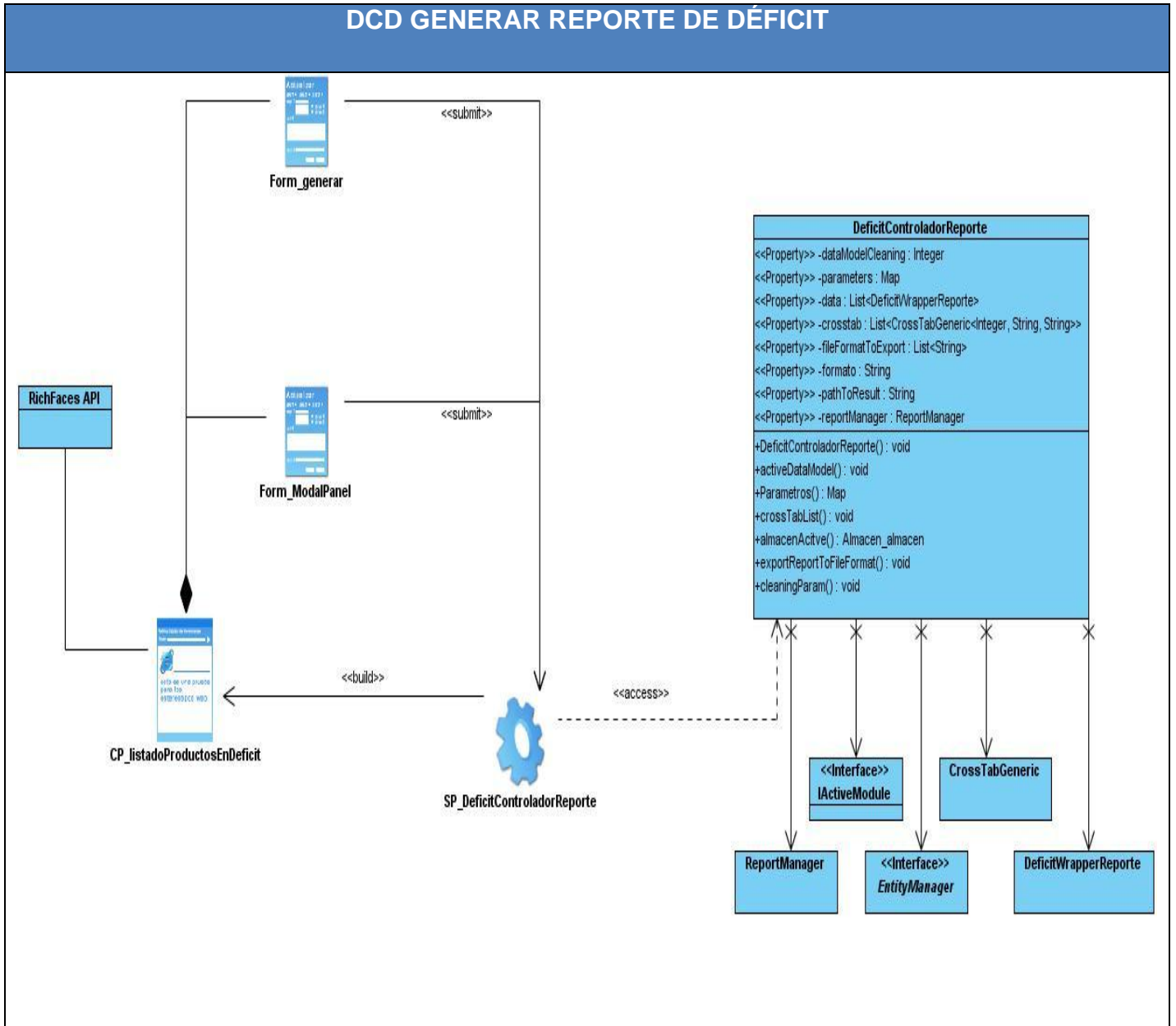
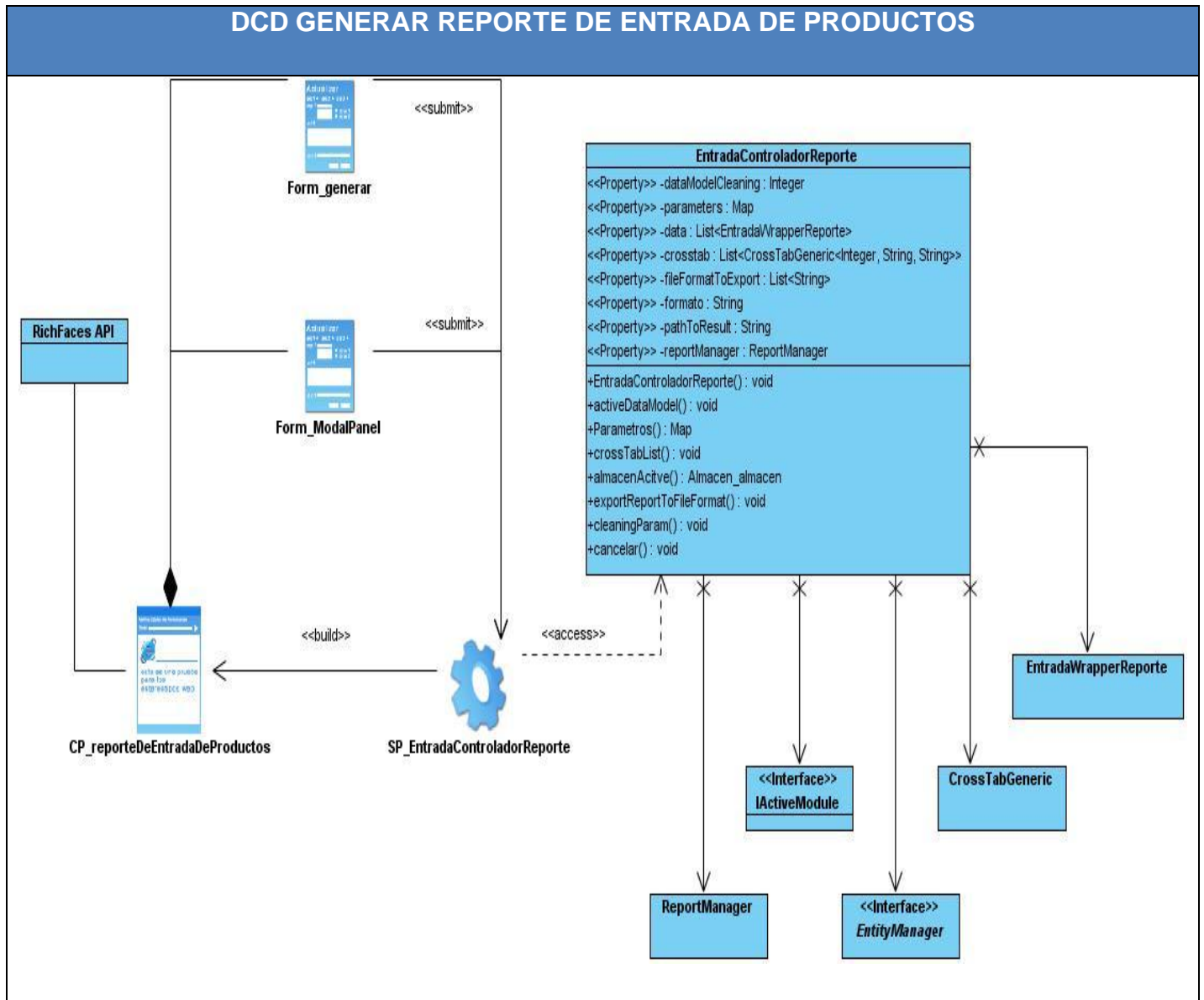
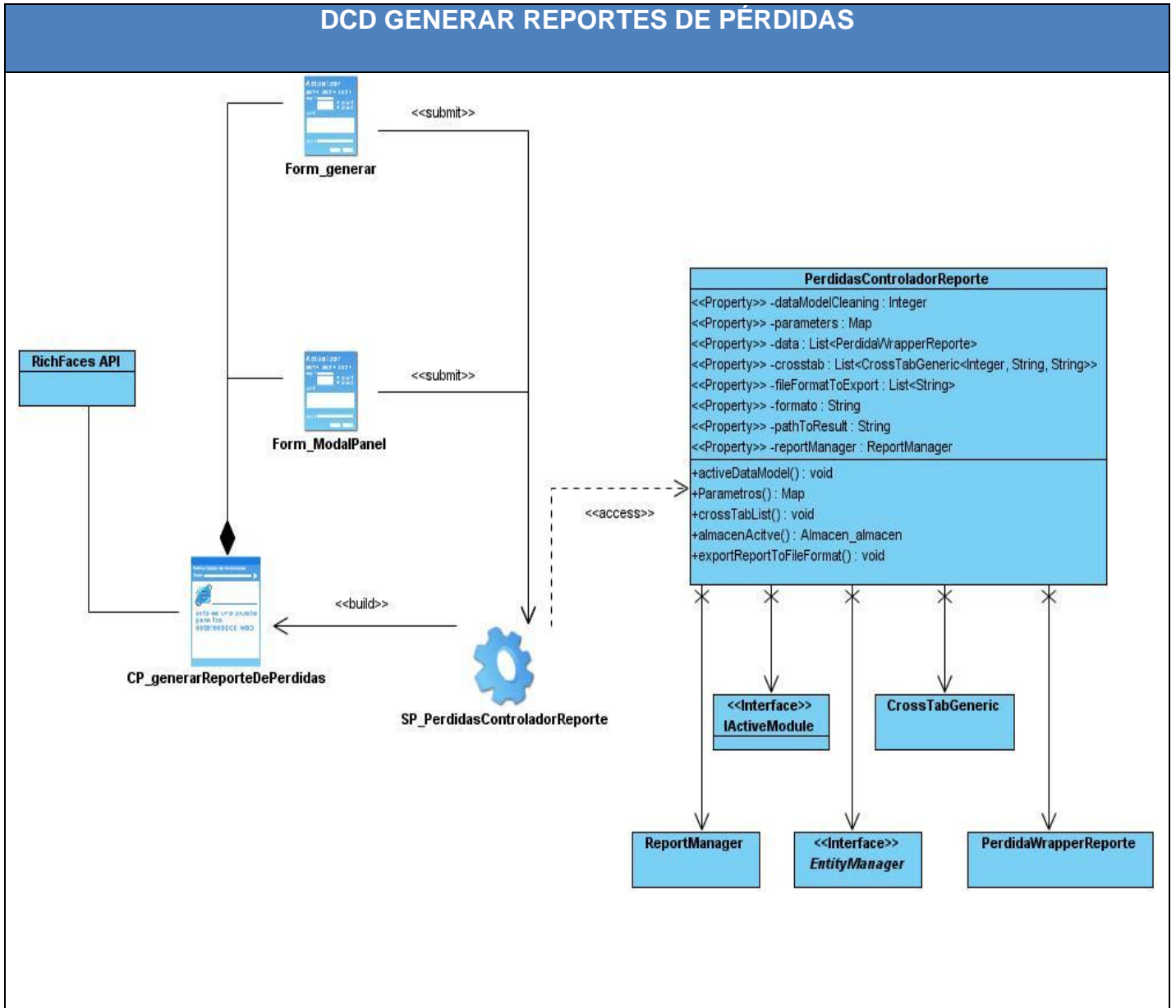


Tabla 3-3 Diagrama de Clases del Diseño \_ Generar reporte de déficit



**Tabla 3-4 Diagrama de Clases del Diseño \_ Generar reporte de entrada de producto**



**Tabla 3-5 Diagrama de Clases del Diseño \_ Generar reporte de pérdidas**

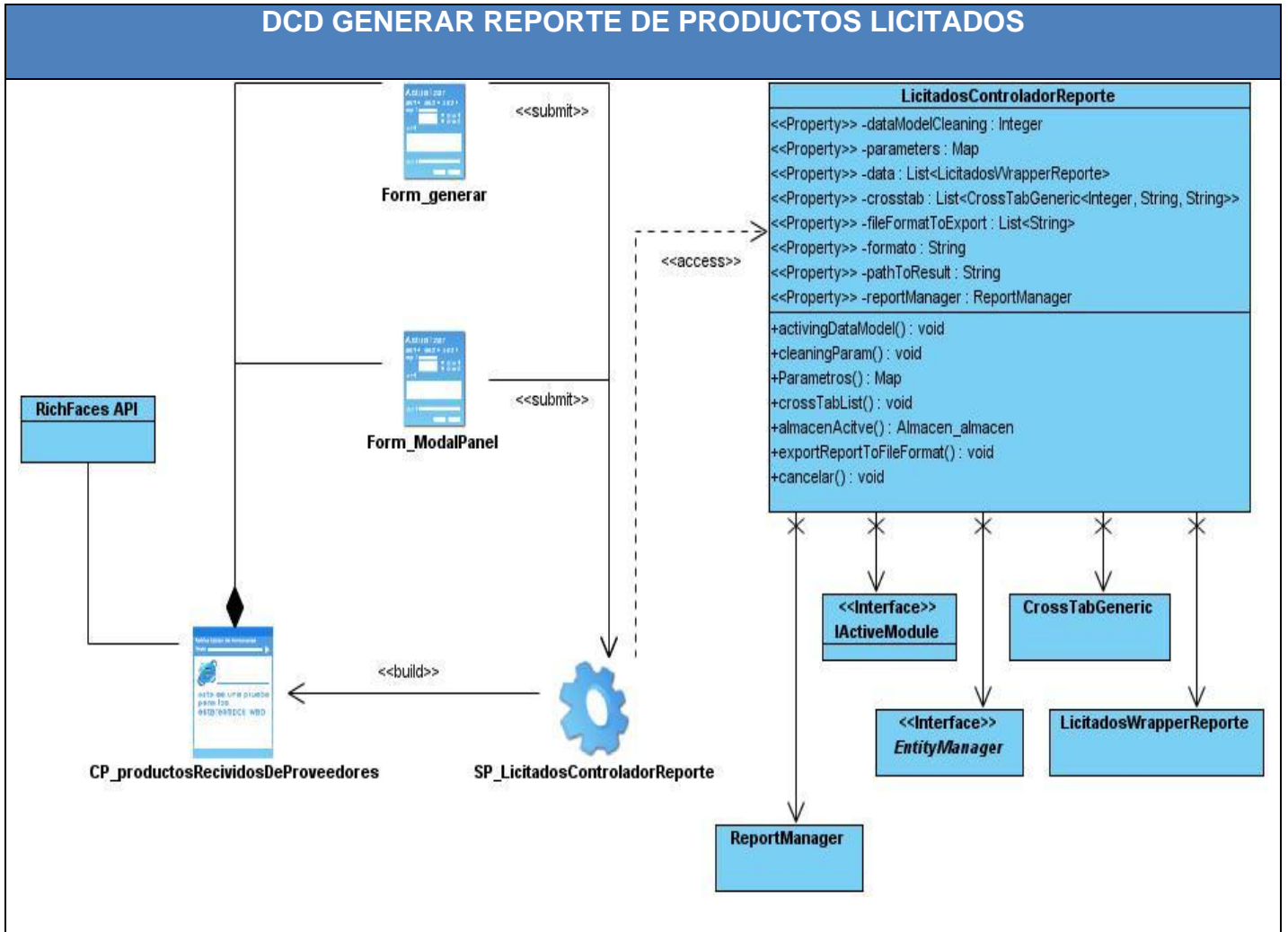


Tabla 3-6 Diagrama de Clases del Diseño \_ Generar reporte de productos licitados

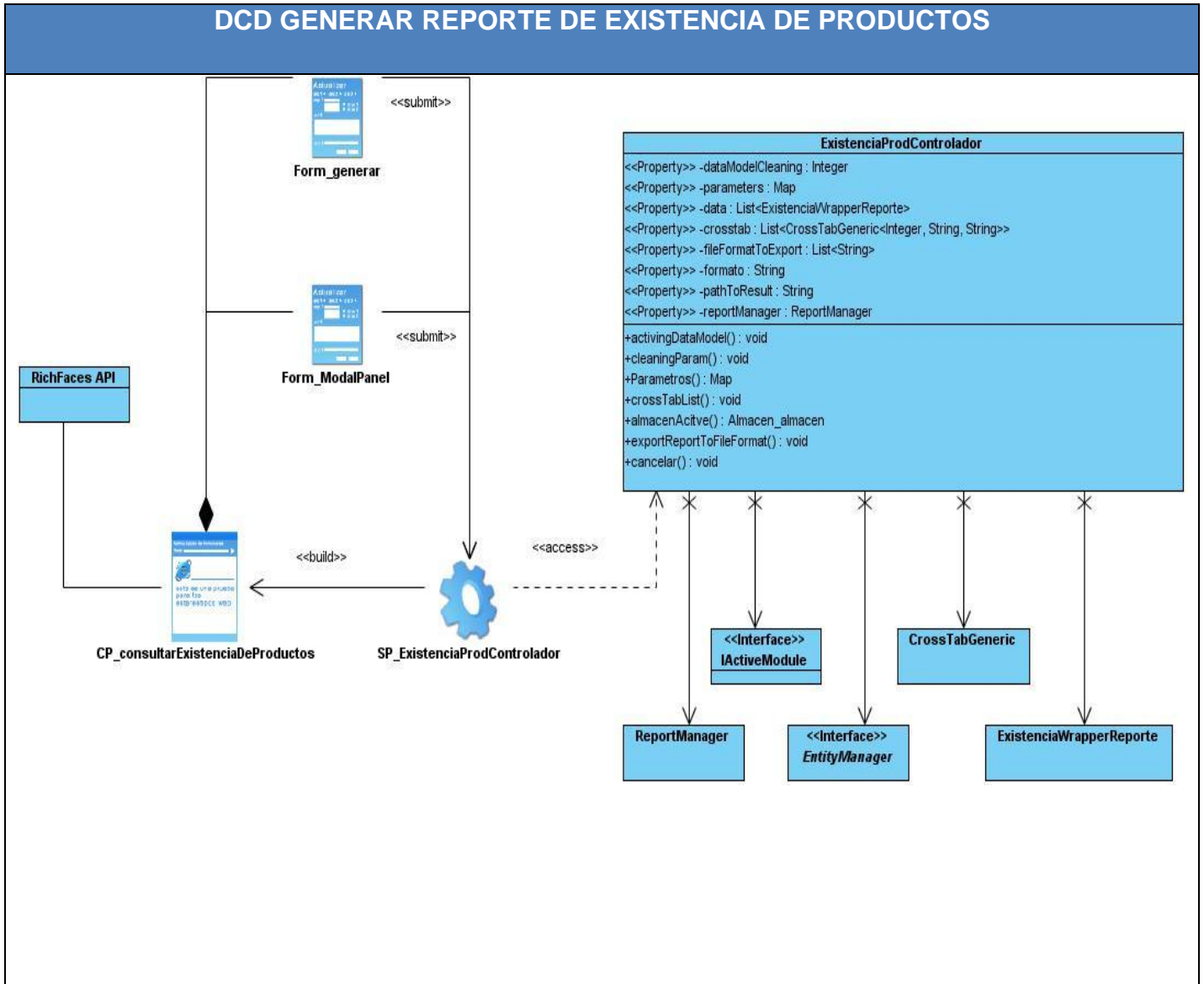


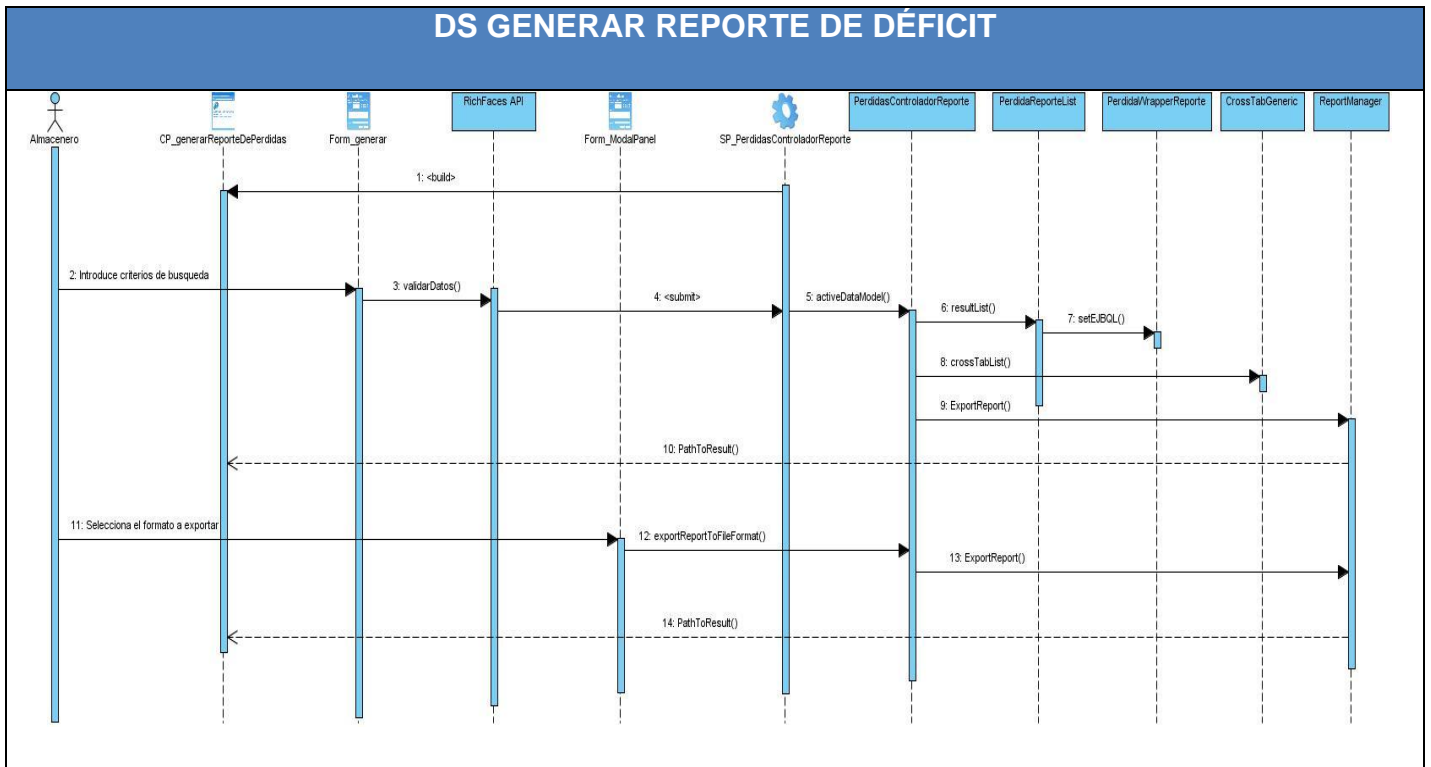
Tabla 3-7 Diagrama de Clases del Diseño \_ Generar reporte de existencia de productos



### 3.5 Diagramas de interacción

Se trata de un término genérico que se aplica a varios tipos de diagramas que resaltan las interacciones entre elementos. Utilizados para modelar los aspectos dinámicos de un sistema. Consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden enviarse entre ellos. Estos diagramas se dividen en: diagramas de colaboración los cuales muestran una interacción organizada en relación a los objetos que efectúan operaciones y los diagramas de secuencia que muestran los objetos que participan en la interacción mediante sus líneas de vida y los mensajes que intercambian, organizados en forma de una secuencia temporal.

#### Diagramas de secuencia



**Tabla 3-8 Diagrama de Secuencia \_ Generar reporte de déficit**

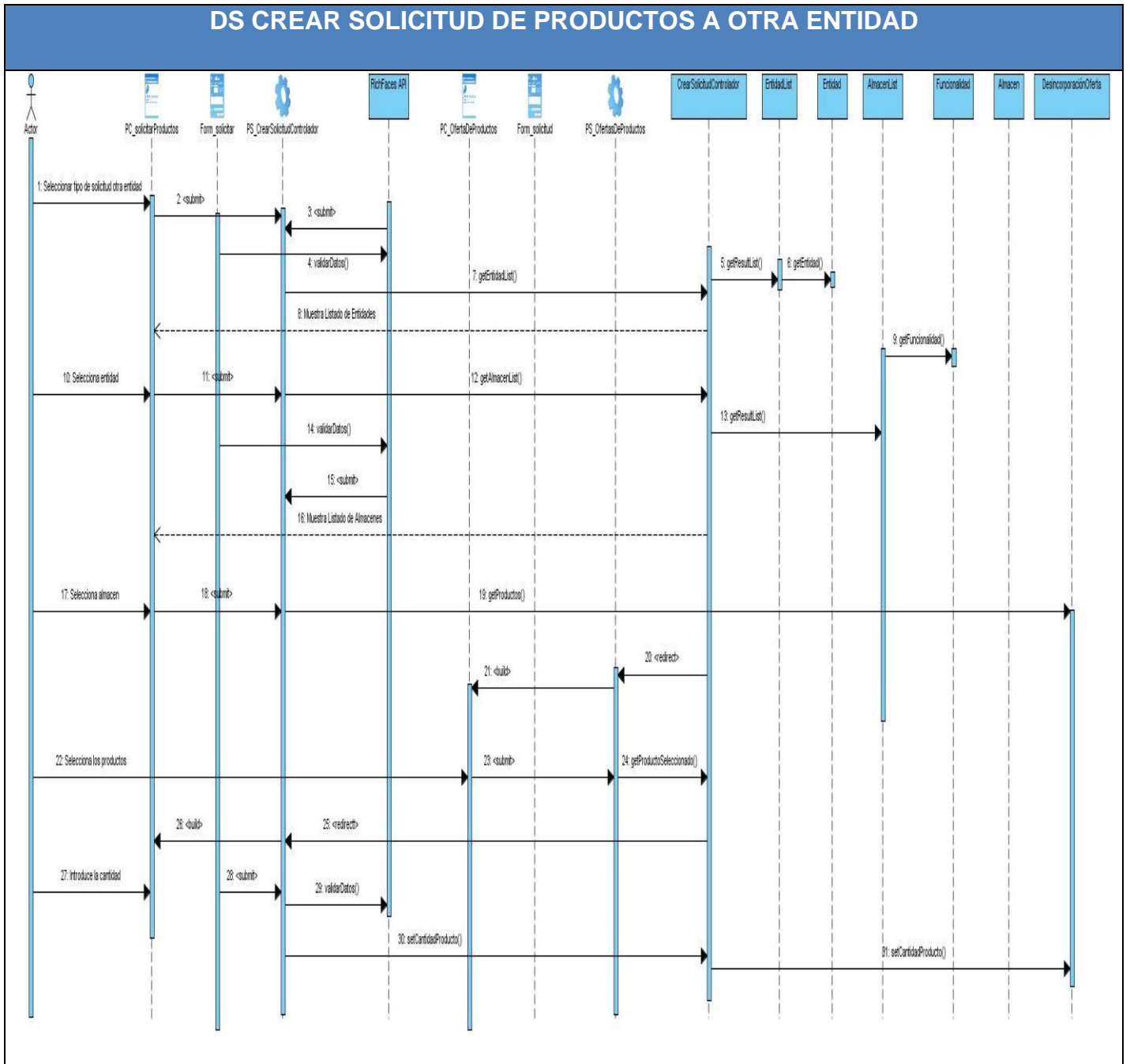


Tabla 3-9 Diagrama de Secuencia \_ Crear solicitud de productos a otra entidad

### 3.6 Descripción de las nuevas clases u operaciones necesarias

En este acápite se describen las clases que han sido identificadas en el diseño para su posterior implementación, con el propósito de lograr una efectiva comprensión de la aplicación.

<b>Nombre:</b>	<b>ConsumoControladorReporte</b>	
<b>Tipo de clase:</b>	<b>Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>	
fechaBef	Date	
fechaAft	Date	
solicitante	String	
parameters	Map	
data	List<ConsumoWrapperReporte>	
crosstab	List<CrossTabGeneric<Integer, String, String>>	
fileFormatToExport	List<String>	
formato	String	
pathToResult	String	
reportManager	ReportManager	
activeModule	IActiveModule	
entityManager	EntityManager	
<b>Para cada responsabilidad:</b>		
Nombre:	ConsumoControladorReporte()	
Descripción:	Constructor de la clase el cual se encarga de inicializar los atributos.	
Nombre:	activeDataModel()	
Descripción:	Inicializa los atributos correspondientes al reporte.	
Nombre:	Parametros()	
Descripción:	Inicializa los parámetros que conforman los datos estáticos en la plantilla que se genera devolviendo un objeto de tipo Map.	
Nombre:	crossTabList()	
Descripción:	Inicializa la lista de los datos que se mostrarán en la plantilla del reporte generado.	
Nombre:	almacenActive()	
Descripción:	Devuelve el almacén activo.	
Nombre:	exportReportToFileFormat()	
Descripción:	Asigna al atributo pathToResult la dirección de la plantilla compilada con el formato seleccionado por el usuario.	
Nombre:	cleaningParam()	
Descripción:	Se le asigna a los atributos valor null.	

**Tabla 3-10 Descripción textual de la clase ConsumoControladorReporte**

<b>Nombre:</b>	<b>ConsumoWrapperReporte</b>
----------------	------------------------------

**Capítulo 3: Descripción y análisis de la solución propuesta**

<b>Tipo de clase:</b>		<b>Entidad</b>
<b>Atributo</b>		<b>Tipo</b>
fechaDespacho		Date
codigo		String
nombre		String
valor		String
nombreAlm		String
cantidadAutorizada		Long
total		Long
<b>Para cada responsabilidad:</b>		
Nombre:	ConsumoWrapperReporte()	
Descripción:	Constructor de la clase el cual se encarga de inicializar los atributos.	
Nombre:	getFechaDespacho()	
Descripción:	Devuelve la fecha de despacho del producto.	
Nombre:	setFechaDespacho()	
Descripción:	Modifica la fecha de despacho del producto.	
Nombre:	getCodigo()	
Descripción:	Devuelve el código del producto.	
Nombre:	setCodigo()	
Descripción:	Modifica el valor del código del producto	
Nombre:	getNombre()	
Descripción:	Devuelve el nombre del producto.	
Nombre:	setNombre()	
Descripción:	Modifica el nombre del producto.	
Nombre:	getValor()	
Descripción:	Devuelve la presentación del producto (tableta, bulbo, etc...).	
Nombre:	setValor()	
Descripción:	Modifica la presentación del producto.	
Nombre:	getNombreAlm()	
Descripción:	Devuelve el nombre del almacén responsable del consumo.	
Nombre:	setNombreAlm()	
Descripción:	Modifica el nombre del almacén.	
Nombre:	getCantidadAutorizada()	
Descripción:	Devuelve la cantidad autorizada del producto.	
Nombre:	setCantidadAutorizada()	
Descripción:	Modifica la cantidad autorizada.	
Nombre:	getTotal()	
Descripción:	Obtiene el total del consumo del producto.	
Nombre:	setTotal()	
Descripción:	Modifica el total utilizado del producto.	

**Tabla 3-11 Descripción textual de la clase ConsumoWrapperReporte**

**3.7 Modelo de datos**

El modelo de datos describe la representación lógica y física de los datos persistentes. Este puede ser generado a partir del modelo de objetos y viceversa, brindando la posibilidad de ahorrar esfuerzo y tiempo de trabajo. El modelado de datos y el desarrollo de un prototipo arquitectónico que permita que el rendimiento de la base de datos se evalúe son esenciales para alcanzar una arquitectura estable. Como los casos de uso arquitectónicamente significativos se detallan y analizan en cada iteración, los elementos del modelo de datos se definen basándose en el desarrollo de los diseños de clases persistentes a partir de los casos de uso. Una ventaja que ofrece este modelo es que posibilita un conjunto de operaciones básicas para realizar consultas y actualizar los datos.

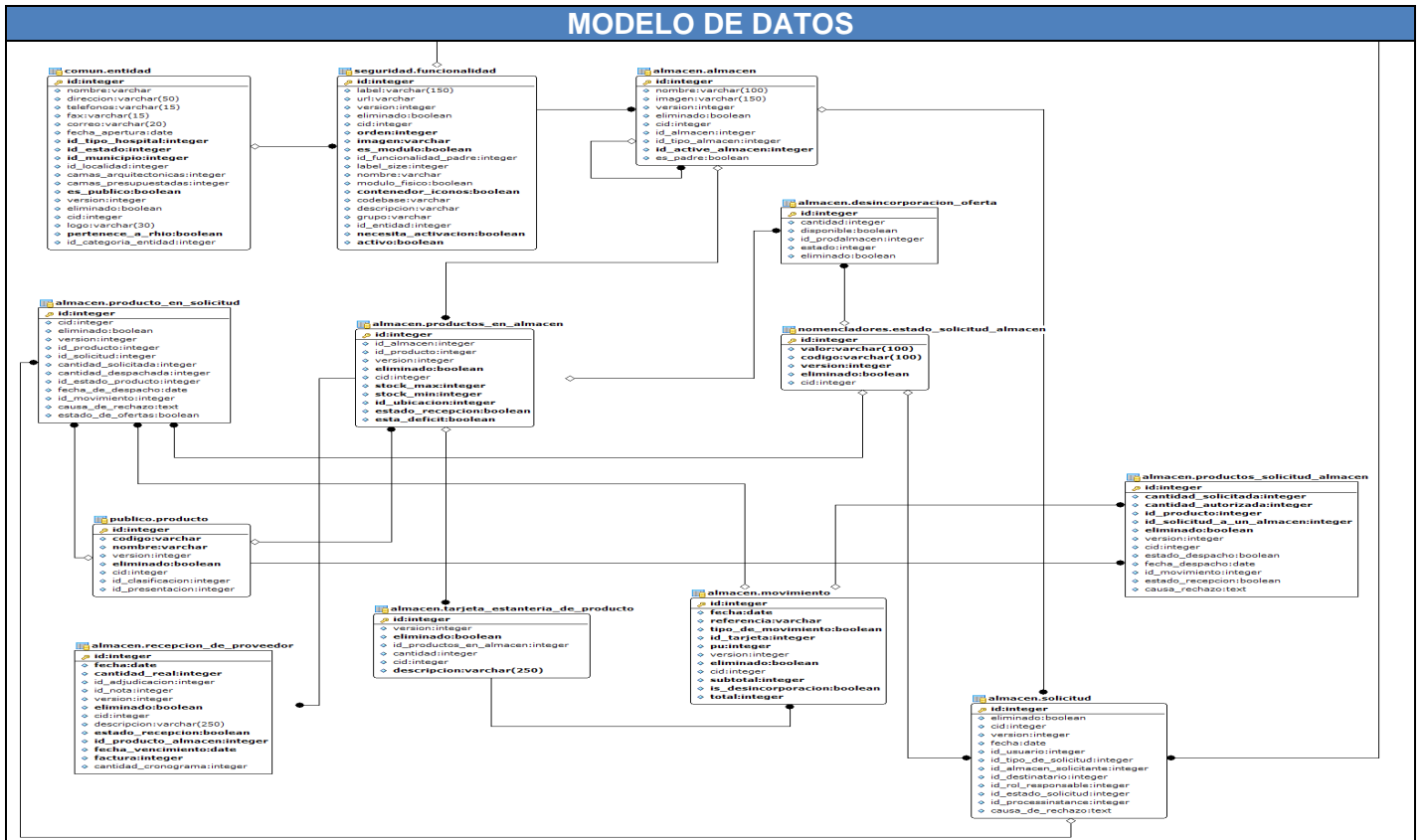


Tabla 3-12 Modelo de datos

### **3.8 Breve valoración de las Técnicas de validación (Integridad, Normalización de la Base de datos)**

En toda aplicación informática el tratamiento de errores constituye un aspecto importante a tener en cuenta. El manejo de acciones no esperadas permitiría el correcto funcionamiento y la debida respuesta del sistema informático. Con el fin de lograr el comportamiento adecuado del sistema se hacen uso de técnicas de validación encargadas del tratamiento de posibles errores ocasionados tanto de la aplicación como en la entrada indebida de datos por parte del cliente o usuario. Estas técnicas posibilitan que el sistema continúe con la ejecución de las tareas definidas ante la ocurrencia de un error o comportamiento erróneo.

En toda aplicación de gestión la entrada de datos por parte del usuario se realiza de forma constante. Es por ello que se realizan reiteradas validaciones sobre los datos introducidos. La utilización del framework JSF permite la manipulación y gestión de estas validaciones en las vistas, mostrando mensajes globales de errores generales como mensajes en errores particulares.

Mediante el uso de Hibernate como herramienta de mapeo objeto-relacional, se denotan los métodos de acceso a datos con anotaciones tales como `@Transactional` la cual permite prever que si existe algún error en las operaciones los cambios no surtan efecto en el modelo y se mantenga el estado de los datos.

Otra de las anotaciones existentes para la validación de los datos es `@Pattern` la cual permite la validación de los datos mediante expresiones regulares. Impidiendo, por ejemplo, no entrar datos de texto en campos de datos numéricos, entre otros casos de acuerdo a la expresión regular deseada.

**3.8.1 Descripción de las tablas**

<b>Nombre:</b>	<b>desincorporacion_oferta</b>	
<b>Descripción:</b>	<b>Almacena los productos que son puestos en oferta por las entidades para la solicitud de los mismos.</b>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id</b>	Integer (Primary key)	Identificador único del producto en la tabla.
cantidad	Integer	La cantidad del producto en oferta
disponible	Boolean	Variable de control, si es true el producto está disponible, si es false no lo está.
id_prodalmacen	Integer (Foreign Key)	Identificador único del producto perteneciente a la tabla <b>productos_en_almacen</b> .
estado	Boolean (Foreign Key)	Identificador único perteneciente a la tabla <b>estado_solicitud_almacen</b> , el cual indica el estado del producto en la oferta.
eliminado	Boolean	Variable de control, si es true el producto fue eliminado, si es false indica lo contrario.

**Tabla 3-13 Tabla del modelo de datos desincorporación\_oferta**

<b>Nombre:</b>	<b>producto_en_solicitud</b>	
<b>Descripción:</b>	<b>Almacena los productos que fueron solicitados por una entidad.</b>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id</b>	Integer (Primary key)	Identificador único del producto en la tabla.

### Capítulo 3: Descripción y análisis de la solución propuesta

cid	Integer (Primary key)	Registro de acción en la bitácora de sucesos.
eliminado	Boolean	Variable de control, si es true el producto fue eliminado, si es false indica lo contrario.
id_producto	Integer (Foreign Key)	Identificador único del producto perteneciente a la tabla <b>producto</b> .
id_solicitud	Integer (Foreign Key)	Identificador único de la solicitud perteneciente a la tabla <b>solicitud</b> .
cantidad_solicitada	Integer	La cantidad que fue solicitada del producto en oferta por la entidad.
cantidad_despachada	Integer	La cantidad que le fue despachada, puede ser total o parcial de la cantidad solicitada.
id_estado_producto	Integer (Foreign Key)	Identificador único perteneciente a la tabla <b>estado_solicitud_almacen</b> , el cual indica el estado del producto en la solicitud.
fecha_de_despacho	Date	Indica la fecha en que se realizó el despacho del producto solicitado.
id_movimiento	Integer (Foreign Key)	Identificador único del movimiento perteneciente a la tabla <b>movimiento</b> .
causa_de_rechazo	Text	Texto que explica las causas por las cuales la(s) solicitud(es) fueron rechazadas.

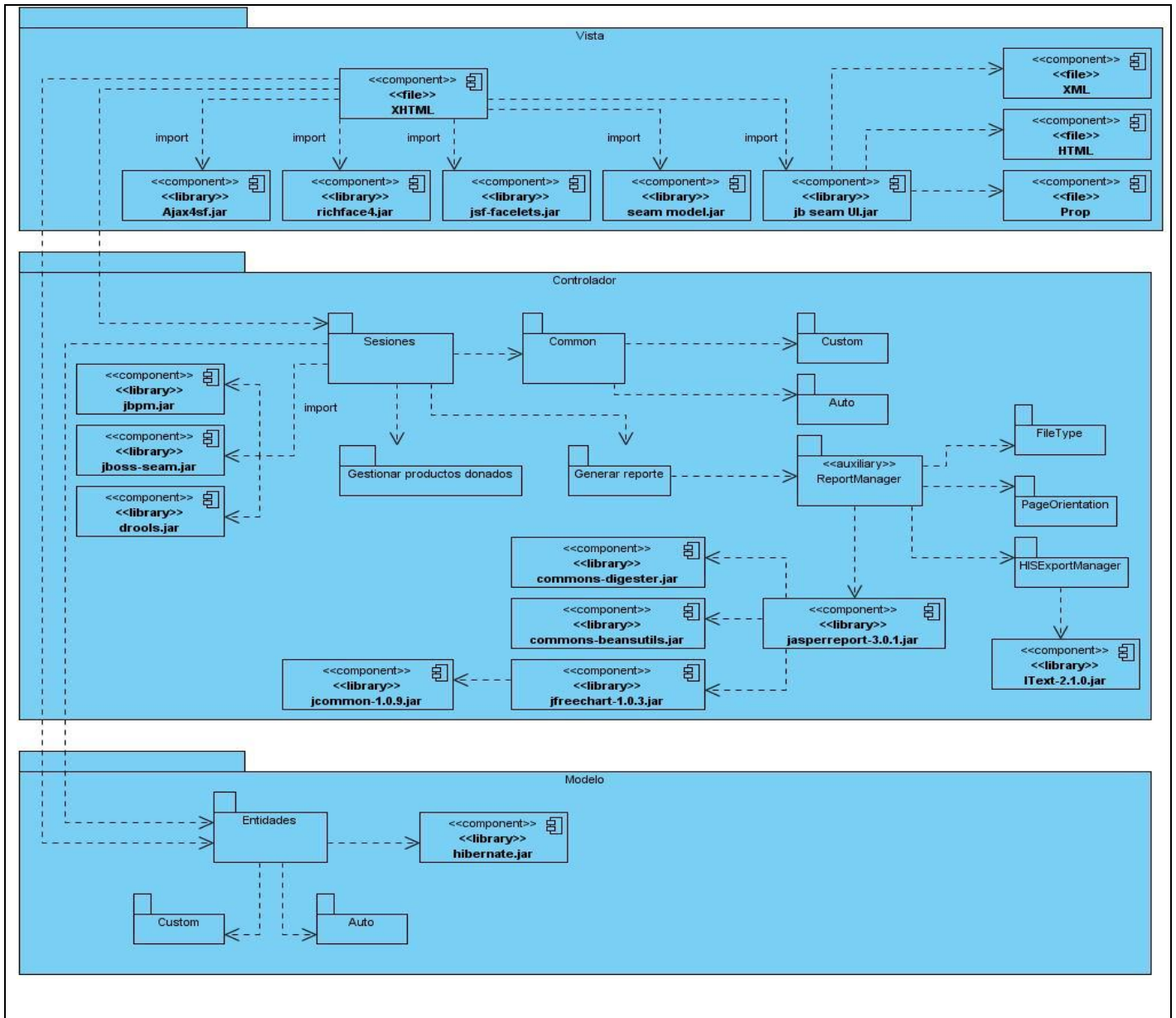


**Tabla 3-14** Tabla del modelo de datos producto\_en\_solicitud

### **3.9 Vista de Implementación. (Diagrama de Componentes)**

El diseño al nivel de componentes describe por completo el detalle interno de cada componente de la aplicación. A continuación se muestra el diagrama de componentes, integrado por todos los elementos asociados a tres paquetes fundamentales en él presente: Modelo Vista Controlador. Este diagrama representa la vista estática de un sistema. Su uso más importante es mostrar la estructura de nivel elevada del Modelo de Implementación, específicamente: Subsistemas de implementación y sus dependencias de importación y los subsistemas de implementación organizados en capas. [22]

## **DIAGRAMA DE COMPONENTES**



**Tabla 3-15 Diagrama de Componentes**

## Capítulo 4: Modelo de Prueba

El desarrollo de un producto lleva la realización de diversas actividades, en su etapa final, se debe pasar al producto por acciones que evalúen la calidad de producción, son pruebas de calidad que representan una revisión final de las especificaciones del diseño y de la codificación, a través de métodos para el desarrollo de ellas resulta más fácil la detección de errores.

El presente y último capítulo, Modelo de Prueba, se centra principalmente en la evaluación y determinación de la calidad del producto obtenido, se realiza una descripción del procedimiento utilizado para el diseño de los casos de pruebas con fin de evaluar a los casos de uso significativos en su correcto desempeño, además de mostrarse algunos como ejemplos.

Los casos de pruebas no pueden asegurar la ausencia de errores, solo pueden demostrar que existen defectos en el software. Las pruebas son actividades en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

### 4.1 Prueba de caja negra

Previamente a la aplicación de métodos para los diseños de los casos de pruebas es necesario analizar algunos de los principios básicos que guían las pruebas del producto.

- Una parte necesaria de un caso de prueba es la definición del resultado esperado.
- Los casos de pruebas deben ser escritos no solo para entradas válidas y esperadas sino también para entradas inválidas e inesperadas.
- Las pruebas deben empezar por lo pequeño e ir hacia lo grande.

Para la realización de los casos de pruebas existen dos métodos muy comunes, el procedimiento de **caja blanca** donde se verifica que las operaciones internas se ajustan a lo especificado y que los componentes internos hagan la función que deben hacer y el de **caja negra**, donde a través de pruebas se evalúa que el sistema funcione como se diseñó y los requerimientos hayan sido cumplidos.

En el siguiente capítulo es utilizado el método de **caja negra**, el cual se centra principalmente en validar los requisitos funcionales del sistema. Dentro del método se especifican diferentes técnicas, entre ellas se encuentran:

- **Técnica de la Partición de Equivalencia:** Esta técnica es la utilizada para diseñar los casos de pruebas, divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. En la realización de este capítulo se verá representada la utilización de esta técnica.
- **Técnica del Análisis de Valores Límites:** Esta técnica prueba la habilidad del sistema para manejar datos que se encuentran en los límites aceptables.
- **Técnica de Grafos de Causa-Efecto:** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

## **4.2 Descripción de los casos de pruebas**

**4.2.1 Caso de uso: Crear solicitud de productos a otra entidad**

Escenarios del crear solicitud de productos a otra entidad	Descripción de la funcionalidad	Flujo Central
<p>EC 1: Crear solicitud de productos a otra entidad</p>	<p>Crear una solicitud de productos a otra entidad satisfactoriamente.</p>	<p>Se selecciona la opción Crear solicitud de productos a otra entidad.                      Se seleccionan los datos correspondientes.                      Muestra un listado de entidades hospitalarias.                      Se seleccionan los datos correspondientes.                      Muestra un listado de almacenes.                      Se seleccionan los datos correspondientes.                      Muestra un listado de los productos asociados al almacén seleccionado.                      Se seleccionan los productos a solicitar.                      Se selecciona la opción Aceptar.                      Muestra información de la solicitud con el listado de los productos seleccionados.                      Se introduce la cantidad a solicitar por cada producto seleccionado.                      Se selecciona la opción Aceptar.                      Se crea la solicitud de productos a otra entidad.                      Muestra la interfaz Ver detalles de solicitud de productos a otra entidad. Ver DCP: Ver detalles de solicitud de productos a otra entidad.</p>
<p>EC 2: Eliminar los productos</p>	<p>Se eliminan los productos</p>	<p>Se selecciona la opción Crear</p>

<p>seleccionados</p>	<p>seleccionados.</p>	<p>solicitud de productos a otra entidad.                  Se seleccionan los datos correspondientes.                  Muestra un listado de entidades hospitalarias.                  Se seleccionan los datos correspondientes.                  Muestra un listado de almacenes.                  Se seleccionan los datos correspondientes.                  Muestra un listado de los productos asociados al almacén seleccionado.                  Se seleccionan los productos a solicitar.                  Se selecciona la opción Eliminar.                  Se eliminan los productos seleccionados.</p>
<p>EC 3: Cancelar operación</p>	<p>Cancelar la opción de Crear solicitud de productos a otra entidad.</p>	<p>Se selecciona la opción Crear solicitud de productos a otra entidad.                  Se seleccionan los datos correspondientes.                  Muestra un listado de entidades hospitalarias.                  Se seleccionan los datos correspondientes.                  Muestra un listado de almacenes.                  Se seleccionan los datos correspondientes.                  Muestra un listado de los productos asociados al almacén seleccionado.                  Se seleccionan los productos a solicitar.                  Se selecciona la opción Aceptar.                  Muestra información de la</p>

		<p>solicitud con el listado de los productos seleccionados.</p> <p>Se introduce la cantidad a solicitar por cada producto seleccionado.</p> <p>Se selecciona la opción Cancelar.</p> <p>Se cancela la solicitud de productos a otra entidad.</p> <p>Se regresa a la vista anterior.</p>
EC 4: Existen datos incorrectos	<p>Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los campos incorrectos.</p>	<p>Se selecciona la opción Crear solicitud de productos a otra entidad.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Muestra un listado de entidades hospitalarias.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Muestra un listado de almacenes.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Muestra un listado de los productos asociados al almacén seleccionado.</p> <p>Se seleccionan los productos a solicitar.</p> <p>Se selecciona la opción Aceptar.</p> <p>Muestra información de la solicitud con el listado de los productos seleccionados.</p> <p>Se introduce la cantidad a solicitar por cada producto seleccionado.</p> <p>Se selecciona la opción Aceptar.</p> <p>Muestra un indicador sobre los campos incorrectos.</p>

**SC 1: Crear solicitud de productos a otra entidad**

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5
Escenario	Crear solicitud de productos a otra entidad	Eliminar los productos seleccionados	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Combo 3 (Tipo)	V			I	I
Botón 1 (Eliminar)		NA			
Botón 2 (Aceptar)	NA			NA	NA
Botón 3 (Cancelar)			NA		
Respuesta del Sistema	Crea la solicitud de productos a otra entidad y se visualiza la interfaz ver detalles.	Se eliminan los productos seleccionados.	Regresa a la vista anterior.	Muestra un indicador sobre los campos incompletos.	Muestra un indicador sobre los campos incorrectos.
Resultado de la Prueba					



**4.2.2 Caso de uso: Ver detalles solicitud de productos a otra entidad**

Escenarios del Ver detalles solicitud de productos a otra entidad	Descripción de la funcionalidad	Flujo Central
EC 1: Ver detalles solicitud de productos a otra entidad.	Ver detalles solicitud de productos a otra entidad satisfactoriamente.	Se muestra la interfaz Ver detalles solicitud de productos a otra entidad. Se muestran los datos de la solicitud de productos a otra entidad. Selecciona la opción Salir. Regresa a la vista Solicitar productos.
EC 2: Modificar solicitud de productos a otra entidad.	Permite acceder al caso de uso Modificar solicitud de productos a otra entidad.	Se muestra la interfaz Ver detalles solicitud de productos a otra entidad. Se muestran los datos de la solicitud de productos a otra entidad. Se selecciona la opción Modificar. Ver DCP <b>Modificar solicitud de productos a otra entidad.</b>
EC 3: Eliminar solicitud de productos a otra entidad.	Permite acceder al caso de uso Eliminar solicitud de productos a otra entidad.	Se selecciona la opción Eliminar la solicitud de productos a otra entidad. Ver DCP <b>Eliminar solicitud de productos a otra entidad.</b>

**SC 1: Ver detalles solicitud de productos a otra entidad**

Id del escenario	EC 1	EC 2	EC 3
Escenario	Ver detalles solicitud de productos a otra entidad.	Modificar solicitud de productos a otra entidad.	Eliminar solicitud de productos
Botón 1	NA		

(Salir)			
Botón 2 (Modificar)		NA	
Botón 3 (Eliminar)			NA
Respuesta del Sistema	Regresa a la vista Solicitar productos.	Muestra la interfaz para modificar.	Regresa a la vista anterior.
Resultado de la Prueba			

**4.2.3 Caso de uso: Modificar solicitud de productos a otra entidad**

Escenarios del Modificar solicitud de productos a otra entidad	Descripción de la funcionalidad	Flujo Central
EC 1: Modificar solicitud de productos a otra entidad.	Modificar una solicitud de productos a otra entidad satisfactoriamente.	Se muestra la interfaz Modificar una solicitud de productos a otra entidad. Se modifican los datos correspondientes. Se selecciona la opción Aceptar. Se modifica la solicitud de productos a otra entidad. Muestra la interfaz Ver detalles solicitud de productos a otra entidad. Ver DCP <b>Ver detalles solicitud de productos a otra entidad.</b>
EC 2: Cancelar operación	Cancelar la opción de Modificar solicitud de productos a otra entidad.	Se selecciona la opción Modificar solicitud de productos a otra entidad. Se introducen los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incompletos muestra un indicador sobre los campos incompletos.	Se selecciona la opción Modificar solicitud de productos a otra entidad. Se introducen los datos incompletos. Muestra un indicador sobre los campos incompletos.
EC 4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los campos incorrectos.	Se selecciona la opción Modificar solicitud de productos a otra entidad. Se introducen los datos incompletos. Muestra un indicador sobre los

		campos incompletos.
--	--	---------------------

**SC 1: Modificar solicitud de productos a otra entidad**

Id del escenario	EC 1	EC 3	EC 4	EC 5
Escenario	Modificar solicitud de productos a otra entidad	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Variable 1 (Cantidad)	V		I	I
Botón 1 (Aceptar)	NA		NA	NA
Botón 2 (Cancelar)		NA		
Respuesta del Sistema	Modifica la solicitud de productos a otra entidad y se visualiza la interfaz ver detalles.	Regresa a la vista anterior.	Muestra un indicador sobre los campos incompletos.	Muestra un indicador sobre los campos incorrectos.
Resultado de la Prueba				

**4.2.4 Caso de uso: Eliminar solicitud de productos a otra entidad**

Escenarios del Eliminar solicitud de productos a otra entidad	Descripción de la funcionalidad	Flujo Central
EC 1: Eliminar solicitud de productos a otra entidad.	Eliminar una solicitud de productos a otra entidad satisfactoriamente.	Se muestra el mensaje: “Se eliminará la solicitud de productos a otra entidad seleccionada. Al seleccionar Sí se perderán todos los datos. ¿Desea continuar?”. Selecciona la opción Sí. Elimina la solicitud de productos a otra entidad.
EC 2: Cancelar operación	Cancelar la opción de Eliminar solicitud de productos a otra entidad.	Se muestra el mensaje: “Se eliminará la solicitud de productos a otra entidad seleccionada. Al seleccionar Sí se perderán todos los datos. ¿Desea continuar?”. Selecciona la opción No. Se regresa a la vista anterior.

**SC 1: Eliminar solicitud de productos a otra entidad**

Id del escenario	EC 1	EC 3
Escenario	Eliminar solicitud de productos a otra entidad	Cancelar operación
Botón 1 (Sí)		
Botón 2 (No)	NA	
Respuesta del Sistema	Elimina la solicitud de productos a otra entidad.	Regresa a la vista anterior.
Resultado de la Prueba		

Mediante el método de caja negra se definieron un conjunto de condiciones de entrada que ejercitaron completamente todos los requisitos funcionales de la aplicación, estas pruebas permitieron encontrar: funciones incorrectas o ausentes, errores de interfaz de usuario y en el acceso a datos.

## **Conclusiones**

Una vez desarrolladas las tareas de la investigación y haberle dado cumplimiento al objetivo general se puede concluir que:

- El análisis de los procesos de negocio permitió el entendimiento e identificación de los requerimientos funcionales del sistema para su correcta implementación.
- El estudio de los sistemas existentes relacionados con el campo de acción, evidenció que los mismos no comprenden todas las funcionalidades requeridas para realizar la solicitud de productos a otras entidades y la generación de reportes deseado para el módulo Almacén del Sistema de Información Hospitalaria alas HIS.
- Las tecnologías, herramientas y patrones utilizados, permitieron la construcción de un sistema robusto y flexible.
- La utilización de las pautas de diseño aplicadas permitieron lograr uniformidad y homogeneidad en las interfaces visuales obtenidas.
- Las funcionalidades implementadas permitieron el fortalecimiento del módulo Almacén del sistema alas HIS, lo cual facilitará la gestión de la información en el área Almacén de las instituciones hospitalarias.

## **Recomendaciones**

Durante el proceso de pruebas realizadas al sistema se identificaron posibles mejoras a desarrollar que ofrecerían una mejor solución y optimización de las funcionalidades implementadas. Para una posterior versión se recomienda:

- Unificar las funcionalidades para la generación de reportes.
- Implementar la funcionalidad para la cancelación de productos en oferta.

## **Referencias bibliográficas**

1. **Hospital Information System, MEDINOUS.** medinous.com. [En línea] <http://www.medinous.com/>.
2. **Proyecto Sistema de Gestión Hospitalario, SIGEHOS.** [En línea] <http://www.scribd.com/>.
3. **Gestión de almacenes, GESTLOG.** [En línea] [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).
4. **PACIENTE, CNT.** cnt.com.co. [En línea] <http://www.cnt.com.co>.
5. **Herrero Santoja, Joaguín.** La experiencia de KEWAN-KOSMOSALUD. España : s.n.
6. **Java Platform Enterprise Edition (JEE) documentation.** Oracle. [En línea] <http://www.oracle.com/technetwork/java/javasee/documentation/index.html>.
7. **Informáticos, Escuela Superior de Ingenieros.** *Aprenda java como si estuviera en primero.*
8. **Hibernate.** Jboss Community. [En línea] <http://www.hibernate.org/>.
9. **Lesson, Java Beans.** Oracle.com. [En línea] <http://www.download.oracle.com/javase/tutorial/javabeans/whatis/index.html>.
10. **Eugíluz Pérez, Javier.** Librosweb.es. [En línea] [http://www.librosweb.es/xhtml/capitulo1/html\\_y\\_xhtml.html](http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html).
11. **D. Mann, Kito.** JSF Central tm. [En línea] [http://www.jsfcentral.com/editorial/editorial\\_1.html](http://www.jsfcentral.com/editorial/editorial_1.html).
12. **Definition, javaScript.** techtarget.com. [En línea] <http://www.searchsoa.techtarget.com/definition/JavaScript>.
13. **Sheets, Cascade Style.** Ingside.net. [En línea] <http://www.ingside.net/man/css/index.php>.
14. **Hookom, Jacob.** JSF Central tm. [En línea] [http://www.jsfcentral.com/articles/facelets\\_1.html](http://www.jsfcentral.com/articles/facelets_1.html).
15. **JBoss Community.** JBoss Ajax4jsf Introduction. [En línea] <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/introduction.html>.



16. **Documentation, Community.** Jboss.org. [En línea] <http://www.jboss.org/file-access/default/members/jbossrichfaces/freezezone/docs/devguide/en/html/introductions.html>.
17. **UI, Seam.** refcardz.dzone.com. [En línea] <http://www.refcardz.dzone.com/refcardz/seam-ui>.
18. **Allen, Dan y Allen, Richards.** *Seam in action*. 2008.
19. **Cumberlidge, Matt.** *Bussiness Process Management with JBoss JBPM* . 2007.
20. **Library, Drool Documentation.** jboss.org. [En línea] <http://www.jboss.org/drools/documentation.html>.
21. **Sommerville, Ian.** *Ingeniería de software*. 2008.
22. **Introducción a RUP y UML, I.** Entorno Virtual de Aprendizaje. [En línea] [http://eva.uci.cu/file.php/102/Curso\\_2010-2011/Clases/Semana\\_02/Conferencia\\_3/Materiales\\_complementarios/Introducción\\_a\\_RUP\\_y\\_UML.pdf](http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_02/Conferencia_3/Materiales_complementarios/Introducción_a_RUP_y_UML.pdf).
23. **Eclipse.** wiki.eclipse.org. [En línea] [http://wiki.eclipse.org/FAQ\\_What\\_is\\_Eclipse%3F](http://wiki.eclipse.org/FAQ_What_is_Eclipse%3F).
24. JBoss Community,. *jboss.org*. [En línea] <http://www.jboss.org/tools>.
25. PostgreSQL Tools, pgAdmin. *pgadmin.org*. [En línea] <http://www.pgadmin.org/>.
26. **Visual Paradigm for UML.** visual-paradigm.com. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
27. **Carrandi Hernández, Yenisley y Guerrero Barea, Adnier.** *Módulo Almacén del Sistema de Información Hospitalaria alas HIS*. 2009.
28. **Clases de diseño.** Clases de diseño. [En línea] 2010. <http://www.taringa.net/posts/info/1492028/workflow-De-Dise%C3%B1o,-Clases-de-Dise%C3%B1o,-interfaces,-Diagrama.html>

## **Bibliografía**

1. **Hospital Information System, MEDINOUS.** medinous.com. [En línea] <http://www.medinous.com/>.
2. **Proyecto Sistema de Gestión Hospitalario, SIGEHOS.** [En línea] <http://www.scribd.com/>.
3. **Gestión de almacenes, GESTLOG.** [En línea] [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).
4. **PACIENTE, CNT.** cnt.com.co. [En línea] <http://www.cnt.com.co>.
5. **Herrero Santoja, Joaguín.** La experiencia de KEWAN-KOSMOSALUD. España : s.n.
6. **Java Platform Enterprise Edition (JEE) documentation.** Oracle. [En línea] <http://www.oracle.com/technetwork/java/javasee/documentation/index.html>.
7. **Informáticos, Escuela Superior de Ingenieros.** *Aprenda java como si estuviera en primero.*
8. **Hibernate.** Jboss Community. [En línea] <http://www.hibernate.org/>.
9. **Lesson, Java Beans.** Oracle.com. [En línea] <http://www.download.oracle.com/javase/tutorial/javabeans/whatis/index.html>.
10. **Eugíluz Pérez, Javier.** Librosweb.es. [En línea] [http://www.librosweb.es/xhtml/capitulo1/html\\_y\\_xhtml.html](http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html).
11. **D. Mann, Kito.** JSF Central tm. [En línea] [http://www.jsfcentral.com/editorial/editorial\\_1.html](http://www.jsfcentral.com/editorial/editorial_1.html).
12. **Definition, javaScript.** techtarget.com. [En línea] <http://www.searchsoa.techtarget.com/definition/JavaScript>.
13. **Sheets, Cascade Style.** Ingside.net. [En línea] <http://www.ingside.net/man/css/index.php>.
14. **Hookom, Jacob.** JSF Central tm. [En línea] [http://www.jsfcentral.com/articles/facelets\\_1.html](http://www.jsfcentral.com/articles/facelets_1.html).
15. **JBoss Community.** JBoss Ajax4jsf Introduction. [En línea] <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/introduction.html>.

16. **Documentation, Community.** Jboss.org. [En línea] <http://www.jboss.org/file-access/default/members/jbossrichfaces/freezezone/docs/devguide/en/html/introductions.html>.
17. **UI, Seam.** refcardz.dzone.com. [En línea] <http://www.refcardz.dzone.com/refcardz/seam-ui>.
18. **Allen, Dan y Allen, Richards.** *Seam in action*. 2008.
19. **Cumberlidge, Matt.** *Bussiness Process Management with JBoss JBPM* . 2007.
20. **Library, Drool Documentation.** jboss.org. [En línea] <http://www.jboss.org/drools/documentation.html>.
21. **Sommerville, Ian.** *Ingeniería de software*. 2008.
22. **Introducción a RUP y UML, I.** Entorno Virtual de Aprendizaje. [En línea] [http://eva.uci.cu/file.php/102/Curso\\_2010-2011/Clases/Semana\\_02/Conferencia\\_3/Materiales\\_complementarios/Introducción\\_a\\_RUP\\_y\\_UML.pdf](http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_02/Conferencia_3/Materiales_complementarios/Introducción_a_RUP_y_UML.pdf).
23. **Eclipse.** wiki.eclipse.org. [En línea] [http://wiki.eclipse.org/FAQ\\_What\\_is\\_Eclipse%3F](http://wiki.eclipse.org/FAQ_What_is_Eclipse%3F).
24. **JBoss Community, .** jboss.org. [En línea] <http://www.jboss.org/tools>.
25. **PostgreSQL Tools, pgAdmin.** pgadmin.org. [En línea] <http://www.pgadmin.org/>.
26. **Visual Paradigm for UML.** visual-paradigm.com. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
27. **Carrandi Hernández, Yenisley y Guerrero Barea, Adnier.** *Módulo Almacén del Sistema de Información Hospitalaria alas HIS*. 2009.
28. **Clases de diseño.** Clases de diseño. [En línea] 2010. <http://www.taringa.net/posts/info/1492028/workflow-De-Dise%C3%B1o,-Clases-de-Dise%C3%B1o,-interfaces,-Diagrama.html>.
29. Arenas Gutiérrez, René. La revolución Científico Técnica en la salud, desde la perspectiva de la informática médica. Facultad de Ciencia Medicas Julio Trigo López. [Online] <http://fcmjtrigo.sld.cu/materiales/informatica/rctdesdelaimedica.doc>

30. Departamento de lenguajes y sistemas informáticos. Universidad Rey Juan Carlos. Madrid, 2007. [Online] [http://kybele.escet.urjc.es/Documentos/ISI/Implem\\_pruebas.pdf](http://kybele.escet.urjc.es/Documentos/ISI/Implem_pruebas.pdf)
31. Dra. Assetta A, Dr. Fernández Romero D., Dr. Rosell S. Sistemas de Información Hospitalaria, 2006. [Online] <http://www.intramed.net/actualidad/contenidover.asp?contenidoID=44061>
32. Especificación de requisitos. Módulo Almacén del Sistema de Información Hospitalaria alas HIS. ALBET, Ingeniería y Sistemas, 2008.
33. Estándares de codificación de sistemas, 2006. [Online] <http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf>
34. Notas y transparencias de Sistemas Distribuidos. Universidad Politécnica de Madrid. Departamento de informática aplicada. [Online] [http://www.dia.eui.upm.es/Asignatu/Sis\\_dis/Paco/Contenido.pdf](http://www.dia.eui.upm.es/Asignatu/Sis_dis/Paco/Contenido.pdf)
35. Pressman, Roger. Ingeniería de software: Un enfoque práctico. 2005

## Anexos

**Desincorporar productos** Q Buscar...

**Criterios de búsqueda**

Código:  Producto:  Cantidad:

Presentación:

**Listado de productos en el almacén**

	Código	Producto	Presentación	Cantidad
<input type="checkbox"/>	11223344	ASPIRINA	Tableta	2
<input checked="" type="checkbox"/>	02154	DIAZEPAN	Tableta	728
<input type="checkbox"/>	1111	DIFFERINE	Tableta	871
<input type="checkbox"/>	3333	CLARITROMICINA	Tableta	5
<input type="checkbox"/>	12321	Backup	Tableta	865
<input type="checkbox"/>	345352	PENICILINA	Tableta	220
<input type="checkbox"/>	65375	ACETILCISTEINA	Bulbo	40
<input type="checkbox"/>	2222	CALCITONINA	Tableta	990
<input type="checkbox"/>	22222	Guante estéril	Tableta	80

⏪ ⏩ ⏴ ⏵

**Listado de productos seleccionados**

Código	Producto	Presentación	Cantidad	Motivo
02154	DIAZEPAN	Tableta	<input type="text" value="20"/>	<input type="text" value="Oferta"/> <input type="button" value="Aceptar"/>

Anexo 1 Desincorporación de productos por oferta

**Crear solicitud de productos** Q Buscar...

[+ Cargar perfil](#)

**Datos de la solicitud**

Fecha: 21/06/2011 Almacén solicitante: Almacén CC1 Tipo:

**Listado de entidades**

Logotipo	Nombre	
	CIREN	<input type="button" value="Agregar"/>
	Consultorio Edelberto	<input type="button" value="Agregar"/>
	Consultorio Jose Manuel	<input type="button" value="Agregar"/>
	Durango	<input type="button" value="Agregar"/>

⏪ ⏩ ⏴ ⏵

**Listado de almacenes**

Nombre	
Farmacia C2	<input type="button" value="Agregar"/>

Anexo 2 Crear solicitud de productos a otra entidad

**Seleccionar productos** Q Buscar...

---

**Criterios de búsqueda**

Código:  Producto:  Presentación:

---

**Listado de productos**

Código	Producto	Presentación	Cantidad en oferta
<input checked="" type="checkbox"/> 11223344	ASPIRINA	Tableta	70
<input type="checkbox"/> 2222	CALCITONINA	Tableta	30
<input checked="" type="checkbox"/> 333	DINISOR	Tableta	10
<input type="checkbox"/> 345352	PENICILINA	Tableta	10
<input type="checkbox"/> 65375	ACETILCISTEINA	Bulbo	70
<input type="checkbox"/> 22222	Guante estéril	Tableta	10
<input type="checkbox"/> 33333	Gasa simple	Tableta	50
<input type="checkbox"/> 12121	Aguja hipodérmica	Tableta	10

◀ ▶ ▶▶ ▶▶▶

---

**Listado de productos seleccionados**

Código	Producto	Presentación	Cantidad en oferta
11223344	ASPIRINA	Tableta	70
333	DINISOR	Tableta	10

Anexo 3 Seleccionar productos en oferta

**Aprobar solicitud de productos** Q Buscar...

---

**Criterios de búsqueda**

Fecha:  Solicitante:  Destinatario:

Responsable:

---

**Listado de solicitudes**

Fecha	Solicitante	Responsable	Entidad destinataria	Unidad destinataria
<input type="checkbox"/> 17/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 17/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 17/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 17/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 17/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 19/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input checked="" type="checkbox"/> 20/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2

◀ ▶ ▶▶ ▶▶▶

---

**Listado de solicitudes seleccionadas**

Almacén CC1

Fecha	Entidad destinataria	Unidad destinataria	Estado
20/06/2011	CIREN	Farmacia C2	Aceptada <input type="button" value="Aceptar"/>

Anexo 4 Crear solicitud de productos a otra entidad

**Autorizar despacho de productos** Q Buscar...

---

**Criterios de búsqueda**

Fecha:  Entidad:  Producto:

---

**Listado de productos solicitados**

Fecha	Entidad	Producto	Solicitados
<input checked="" type="checkbox"/> 20/06/2011	CIMEQ	ASPIRINA	20
<input checked="" type="checkbox"/> 20/06/2011	CIMEQ	DINISOR	5

◀ ▶ ⏪ ⏩

---

**Listado de productos solicitados**

Fecha	Entidad	Producto	Solicitados	Autorizados	Rechazar
20/06/2011	CIMEQ	DINISOR	5	<input type="text" value="5"/>	<input type="checkbox"/>
20/06/2011	CIMEQ	ASPIRINA	20	<input type="text" value="15"/>	<input type="checkbox"/>

Anexo 5 Autorizar despacho de productos a otra entidad

**Aprobar solicitud de productos** Q Buscar...

---

**Criterios de búsqueda**

Fecha:  Solicitante:  Destinatario:

Responsable:

---

**Listado de solicitudes**

Fecha	Solicitante	Responsable	Entidad destinataria	Unidad destinataria
<input type="checkbox"/> 16/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1			Farmacia C2
<input type="checkbox"/> 16/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2
<input checked="" type="checkbox"/> 16/06/2011	Almacén CC1	Alejandro Mario	CIREN	Farmacia C2

◀ ▶ ⏪ ⏩

---


**Listado de solicitudes seleccionadas**

**Almacén CC1**

Fecha	Entidad destinataria	Unidad destinataria	Estado
16/06/2011	CIREN	Farmacia C2	Rechazada

Anexo 6 Autorizar despacho de productos a otra entidad

**Reporte de entrada de productos**





Almacén CC1  
Alejandro Mario  
21/06/2011

Reporte de entrada de productos  
Total : 17

	Código	Producto	Presentación	Cantidad (u)	Fecha	Referencia
1	65375	ACETILCISTEINA	Bulbo	1	24/03/2010	Proveniente de Farmacia C1
2	345352	PENICILINA	Tableta	2	04/05/2010	Proveniente de Farmacia C1
3	11223344	ASPIRINA	Tableta	5	02/04/2010	Proveniente de Farmacia C1
4	11223344	ASPIRINA	Tableta	1	21/04/2010	Proveniente de Farmacia C1
5	02154	DIAZEPAN	Tableta	30	14/03/2011	Proveniente de Farmacia C1
6	02154	DIAZEPAN	Tableta	2	22/04/2010	Proveniente de Farmacia C1
7	22222	Guante estéril	Tableta	75	21/10/2010	Proveniente de LASBA
8	11223344	ASPIRINA	Tableta	10	20/04/2010	Proveniente de LASBA
9	22222	Guante estéril	Tableta	9	13/10/2010	Proveniente de LABIO
10	02154	DIAZEPAN	Tableta	100	21/04/2010	Proveniente de LASBA

Exportar
Salir

**Anexo 7 Generar reporte**