

Universidad de las Ciencias Informáticas
Facultad 7



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

Desarrollo de las consultas de Cristalino y Aprobación de cirugía del
Sistema de Información Hospitalaria alas HIS

Autores: Greter Menéndez Noda

Claudia Fuentes Escobar

Tutores: Ing. Joselín Miló Pérez

Lic. Ángel Fabra Torres

Ciudad de La Habana, Julio 2010

“Año 52 de la Revolución”

Declaración de autoría

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

Para que así conste firmamos la presente a los __ días del mes de ____ del año ____.

Greter Menéndez Noda

Firma del autor

Claudia Fuentes Escobar

Firma del autor

Joselín Miló Pérez

Firma del tutor

Ángel Fabra Torres

Firma del tutor

Datos de contacto

Ing. Joselín Miló Pérez: Instructor recién graduado en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor Facultad # 7, ha impartido las asignaturas Física I y Física II. Forma parte del proyecto Bloque Quirúrgico Oftalmológico del Departamento Gestión Hospitalaria del Centro de Informática Médica (CESIM).

Correo electrónico: jmilo@uci.cu

Lic. Angel Fabra Torres: Licenciado en Ciencias de la Computación, graduado en la Universidad de Oriente en el año 2004. Posee la Categoría Docente de Profesor Asistente. Ha impartido las asignaturas de Práctica Profesional 1, Sistemas de Bases de Datos, Introducción a la Programación, Introducción a la Informática y Principios de Algoritmización, así como cursos optativos o de postgrado. Ha participado en varios proyectos de desarrollo vinculados fundamentalmente con los perfiles de salud y bioinformática. Actualmente labora en el Departamento de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como desarrollador del Módulo de Consulta Externa del Sistema de Información Hospitalaria alas HIS.

Correo electrónico: afabra@uci.cu

RESUMEN

El presente trabajo tiene como principal objetivo desarrollar la *consulta de Microcirugía del Cristalino, el control de lentes intraoculares y las consultas de Aprobación de cirugía* del módulo Consulta externa en el servicio de oftalmología del Sistema de Información Hospitalaria alas HIS, para facilitar la gestión de información en esta área de las instituciones hospitalarias.

Su desarrollo está basado en tecnologías libres, multiplataforma y usando como patrón arquitectónico el Modelo Vista Controlador. Se utilizó Java como lenguaje de programación orientado a objetos, Eclipse como Entorno de Desarrollo, PostgreSQL como gestor de base de datos, Hibernate como herramienta de mapeo objeto relacional para la persistencia de los datos y el framework Seam, para la unión entre el modelo, la vista y el controlador.

Todo el proceso de desarrollo estuvo guiado por los libros: Libro de Proceso para la Administración de Requisitos y Ciclo de vida de proyectos pilotos del programa de mejora, ambos definidos por la Universidad de las Ciencias Informáticas.

La aplicación permitirá eliminar el procesamiento manual de la información, garantizándose que la misma no se encuentre repetida o redundante. Además el cálculo del lente intraocular que se le implantará al paciente se realizará automáticamente, evitando errores de cálculo que puedan afectar la salud del paciente. Todo esto trae consigo una mejora en la prestación de los servicios ofrecidos al mismo.

Palabras claves

Especialidad del Microcirugía del Cristalino del servicio de oftalmología, Consulta externa, Lentes intraoculares, consultas de Aprobación de cirugía.

Índice de contenidos

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	1
1.1 Conceptos básicos asociados al dominio del problema	1
1.2 Sistemas automatizados analizados vinculados al campo de acción	2
1.3 Tecnologías actuales a utilizar	6
1.4 Tecnologías utilizadas en el proceso de desarrollo	8
1.5 Tecnologías Horizontales	11
1.6 Capability Maturity Model Integration (CMMI)	13
1.7 Herramientas de desarrollo	16
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	1
2.1- Flujo actual de los procesos involucrados en el campo de acción.....	1
2.2- Objeto de automatización.	4
2.3- Modelo de negocio	6
2.4- Especificación de los requerimientos de software.....	9
CAPÍTULO 3: DISEÑO DEL SISTEMA	1
3.1- Descripción de la arquitectura.....	1
3.2- Modelo de diseño.....	2
CAPÍTULO 4: IMPLEMENTACION	10

4.1- Modelo de datos	10
4.2- Modelo de Implementación	15
4.3- Tratamiento de excepciones	18
4.4- Seguridad	18
4.5- Estrategias de codificación. Estándares y estilos a utilizar	19
CONCLUSIONES	25
RECOMENDACIONES	26
Referencias Bibliográficas.....	27
Bibliografía.....	30

INTRODUCCIÓN

El hombre desde su surgimiento y a medida que se ha ido desarrollando, ha necesitado transmitir y procesar la información, para lo que ha creado métodos y máquinas que le faciliten esta labor; surgiendo así la informática. En los últimos años esta ha alcanzado un notable desarrollo en las distintas ramas de la sociedad, brindando tecnologías que agilizan la obtención de resultados y el almacenamiento de grandes volúmenes de información.

Unos de los sectores donde se ha incorporado progresivamente es la salud, debido al gran número de información que constantemente se maneja en los centros hospitalarios y la necesidad de obtención de reportes estadísticos cada vez más específicos; además de mejorar los servicios de planificación sanitaria y de los recursos hospitalarios.

Los primeros sistemas de información médica que surgieron se dedicaban a tareas administrativas, admisión y alta de pacientes y control de inventarios, entre otras. Estos sistemas han evolucionado llegando a especializarse en los diferentes niveles de la salud, alcanzando un notable desarrollo en los hospitales, dando lugar al surgimiento de los Sistemas de Información Hospitalaria, conocido por sus siglas en inglés como HIS (Hospital Information System).

Estos sistemas han tenido gran impacto, ya que aumentan la calidad de la atención al paciente y la productividad de los profesionales y técnicos de la salud; además, apoyan la administración de los recursos humanos y materiales de la institución, incrementando la eficacia en la gestión de los mismos.

En la actualidad, un sistema informático que gestione la información generada en un hospital debe contener: un registro histórico de la información médica del paciente, control de laboratorios y quirófanos desde una sala de mando, generación de recetas, órdenes y generación de reportes estadísticos, etcétera.

En el año 2007 a raíz del proceso de informatización de la sociedad cubana, la Facultad 7 de la Universidad de las Ciencias Informáticas, se propuso el desarrollo de un HIS que automatice los procesos de gestión de la información médico-administrativa; así como de los recursos humanos y materiales de las

distintas áreas de las instituciones hospitalarias, proporcionando una mejora en la calidad de los servicios ofrecidos al paciente.

Este sistema se divide en módulos, correspondientes a cada una de las áreas hospitalarias. Una de estas es la de Consulta externa, la cual se divide en varios servicios, dentro de ellos se encuentra el de oftalmología. Sin embargo dicho servicio no incluye información específica de la especialidad de Microcirugía del Cristalino, por lo que no se gestiona el cálculo del lente intraocular; siendo éste el lente permanente que es implantado de forma quirúrgica en lugar del cristalino del ojo del paciente.

En estos momentos el cálculo y control de estos lentes se realiza a partir de programas específicos, no integrados en una única solución, lo que dificulta al especialista el trabajo en la consulta. Además el control de la disponibilidad de dichos lentes se vuelve engorroso, ya que el movimiento de estos es constante y es necesario llevar el registro de lentes, modelos y poder dióptrico que existen antes de planificar la cirugía. Así mismo se hace necesaria la gestión de los pacientes que se encuentran esperando por la llegada al almacén del lente intraocular que necesitan.

Otras de las especialidades no gestionadas por el módulo Consulta externa en el servicio de oftalmología, son las consultas de Aprobación de cirugía, en las cuales se le da el apto quirúrgico a todo paciente proveniente del servicio de oftalmología que necesite ser operado; lo que provoca una demora en la atención ofrecida al mismo. Además, la información del paciente puede encontrarse repetida o redundante, ocasionando el almacenamiento de elevados volúmenes de información.

Por lo anteriormente planteado se define el siguiente **problema científico** a resolver: ¿Cómo facilitar la gestión de información relacionada con el cálculo y control de lentes intraoculares, la consulta de Microcirugía del Cristalino y las consultas de Aprobación de cirugía en el área de Consulta externa en el servicio de oftalmología de las instituciones hospitalarias?

Este problema está enmarcado en el **objeto de estudio** concerniente a los procesos de gestión de la información en las instituciones hospitalarias y **campo de acción** que comprende los procesos de gestión

de información en el área de Consulta externa en el servicio de oftalmología de las instituciones hospitalarias.

Según lo planteado anteriormente se define como **objetivo general**: Desarrollar *la consulta de Microcirugía del Cristalino, Control de lentes intraoculares y las consultas de Aprobación de cirugía* del módulo Consulta externa en el servicio de oftalmología del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la información en estas especialidades de las instituciones hospitalarias.

Para dar cumplimiento al objetivo planteado se definen las siguientes **tareas de la investigación**:

1. Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria.
2. Identificar los principales procesos asociados al área de Consulta externa en el servicio de oftalmología de las instituciones hospitalarias.
3. Definir las funcionalidades del módulo de Consulta externa en el servicio de oftalmología del Sistema de Información Hospitalaria.
4. Asimilar la arquitectura definida por el Dpto. de Sistema de Gestión Hospitalaria para el desarrollo de sus aplicaciones
5. Obtener los artefactos correspondientes a los flujos de trabajo de “Modelado de Negocio”, “Gestión de Requerimientos”, “Análisis y Diseño” e “Implementación”.
6. Implementar los subprocesos:
 - Cálculo y control de Lentes Intraoculares.
 - Servicio oftalmológico Microcirugía del Cristalino.
 - Consulta de Aprobación de cirugía Clínico.
 - Consulta de aprobación de cirugía Pediatra.

De esta forma se puede destacar que el desarrollo de *la consulta de Microcirugía del Cristalino, Control de lentes intraoculares y las consultas de Aprobación de cirugía* del módulo Consulta externa en el servicio de oftalmología del Sistema de Información Hospitalaria alas HIS, brindará los siguientes beneficios a partir de su puesta en práctica:

- Rapidez y eficacia en la prestación de los servicios al paciente.

- Mejora en la gestión de la información relacionada con el cálculo y control de lentes intraoculares, la consulta de Microcirugía del Cristalino y las consultas de Aprobación de cirugía del área de Consulta externa en el servicio de oftalmología.
- Seguridad, confidencialidad e integridad de la información concerniente al paciente y al resultado de los procesos por los que este transita durante la consulta de Microcirugía del Cristalino y las consultas de Aprobación de cirugía del área de Consulta externa en el servicio de oftalmología.
- Disponer y acceder a información única e integrada del paciente para facilitar los procesos de diagnóstico, tratamiento y seguimiento.
- Automatización del cálculo del lente intraocular que se le implantará al paciente en la consulta de Microcirugía del Cristalino del área de Consulta externa en el servicio de oftalmología.
- Integración con el resto de los módulos del Sistema de Información Hospitalaria alasHIS.

El presente documento se encuentra estructurado en cuatro capítulos:

Capítulo 1: Fundamentación Teórica: investigación sobre el estado del arte de los Sistemas de Información Hospitalaria y otras aplicaciones independientes relacionados con el campo de acción. Descripción de las tecnologías y herramientas a utilizar.

Capítulo 2: Características del sistema: descripción de las funcionalidades del sistema, tomando como punto de partida los procesos del negocio.

Capítulo 3: Diseño del sistema: modelación y construcción de la estructura de la aplicación.

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

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se exponen los conceptos básicos referentes al campo de acción que servirán de apoyo para un mejor entendimiento del mismo. Además son analizados los sistemas de información hospitalaria existentes en la actualidad y se fundamenta la selección de tecnologías y un conjunto de herramientas que son utilizadas para el desarrollo de la investigación.

1.1 Conceptos básicos asociados al dominio del problema

En las instituciones hospitalarias, el área de Consulta externa, y dentro de esta el servicio de oftalmología, tiene como objetivo prestarle una atención especializada a todo paciente que lo necesite, para lo cual se le valora, diagnostica y se le indica una conducta a seguir.

Dentro de las consultas que se tiene este servicio se encuentra la de Microcirugía del Cristalino, en la cual se atiende a todos los pacientes que presenten dificultades con el correcto funcionamiento de este órgano, encargado de enfocar los rayos de luz que entran al globo ocular sobre la retina. En dicha consulta se le crea a cada paciente una *Hoja de Consulta del Cristalino*, en la cual quedan reflejados todos los datos del mismo, así como los *exámenes físicos oftalmológicos*, realizados a través de la observación y las *pruebas diagnósticas*, hechas con equipos especiales, entre otros aspectos que el especialista haya tenido en cuenta para llegar a un diagnóstico. En este documento también se expone la conducta a seguir por el paciente, que puede ser quirúrgica o no. En el caso de que la conducta a seguir sea del tipo quirúrgica, se emite una *Solicitud de Intervención Quirúrgica del Cristalino*, en la cual queda plasmada la información necesaria para llevar a cabo el acto quirúrgico.

En toda cirugía del cristalino se le implanta al paciente como máximo dos *lentes intraoculares*, que tienen como función corregir el enfoque visual después de efectuada la intervención quirúrgica; el tipo de lente y el poder dióptrico de este(os) son determinados por el especialista a partir de un conjunto de reglas y fórmulas preestablecidas con este fin. En caso de que la institución no cuente con el(los) lente(s) intraocular(es) que requiere el paciente, se le crea un *Modelo de paciente pendiente*, documento que expone los datos del paciente y del(los) lente(s) intraocular(es) en cuestión.

El paciente previamente a la cirugía debe de asistir a la *Consulta de Aprobación de cirugía*, que se divide en dos consultas: Consulta del Pediatra, para los pacientes menores de 16 años, y la Consulta del Clínico para los pacientes de 17 años en adelante. En ambas consultas se le hace un interrogatorio al paciente sobre su estado de salud de forma general y en caso de ser necesario se le indican exámenes de laboratorio; al culminar la consulta el médico emitirá una valoración que dirá si el paciente está apto quirúrgicamente o no. Todo este proceso queda plasmado en la *Hoja de consulta del clínico* o del *pediatra* según corresponda.

Por otra parte cuando llega a la institución un lote de lentes intraoculares, se crea un *Modelo de alta*, en el cual se refleja la cantidad de lentes y las características de los mismos, adicionándolos, además, al *Registro de lentes*, el que contiene, entre otros aspectos, la cantidad real de lentes existentes en el almacén. En caso de vencimiento, defecto o ruptura de algún(os) lente(s), se crea el *Modelo de Baja*, que refleja los datos de el(los) mismo(s).

1.2 Sistemas automatizados analizados vinculados al campo de acción

En la actualidad existen numerosos sistemas que brindan soluciones informáticas para el sector de la salud y en especial para los centros hospitalarios. Estos tienen como principal objetivo mejorar la calidad de la atención ofrecida a los pacientes y la administración de los recursos de la institución. Muchos de estos sistemas gestionan e integran los procesos de los departamentos de un hospital, dentro de ellos el servicio de oftalmología del área Consulta externa. Otros, sin embargo, solo brindan las funcionalidades de este servicio como un sistema independiente. Algunos de estos sistemas se muestran a continuación.

Sistemas de Información Hospitalaria			Tipo de Licencia		Tipo de Aplicación	
Nombre	País	Multiplataforma	Libre	Propietario	Web	Escritorio
HIS CNT Pacientes	Colombia	---	---	x	---	x
Medisys Oftal	España	---	---	x	---	x
Oftalmosalus	España	---	---	x	---	x
Galen Hospital	Cuba	---	---	x	---	x

Bloque Quirúrgico Oftalmológico	Cuba	x	---	x	x	---
Sistema Automatizado de Microcirugía	Cuba	---	---	x	---	x

Tabla 1.1 Sistemas de Información Hospitalaria.

HIS CNT Pacientes: sistema de información integrada diseñado para Clínicas y Hospitales de complejidad alta. Su función principal es brindar una mejora en la prestación de los servicios ofrecidos al paciente tanto terapéuticos como de diagnóstico. El eje principal del producto lo constituye la Historia Clínica del Paciente, completamente digital, única y adaptable a todas las áreas de consulta externa. Es una aplicación de escritorio que además brinda la funcionalidad de definir el horario de los médicos para la asignación de citas, logrando la optimización del tiempo, y ofreciendo a sus pacientes una mejor oportunidad del servicio. [1]

Medisys Oftal: es un completo programa para consultas y clínicas oftalmológicas que facilita las consultas y mejora la atención de los pacientes. Permite la impresión de informes médicos directamente. Mediante una cámara de video, se pueden ver, incorporar, capturar, aumentar, mostrar e imprimir imágenes de una lámpara de hendidura. Se instaló por primera vez en España en el año 1996. Sus características principales son:

- Administración de datos del paciente.
- Consultorio electrónico.
- Sistema de captura de imágenes.
- Visor de imágenes.
- Procesamiento de textos integrado.
- Recetas.

El programa es una aplicación para, Windows 98, Windows 2000 y Windows XP. [2]

Oftalmosalus: software de gestión integral para clínicas oftalmológicas. Es intuitivo y fácil de usar. Entre sus funcionalidades más importantes están: [3]

- Envío de recordatorios a clientes vía correo electrónico y SMS.
- Resúmenes de actividades mensuales.
- Sistema para envío de mensajes entre los profesionales del centro.
- Generación automática de recetas, informes y otros documentos.
- Diagnósticos, registro de pruebas complementarias, etc.
- Conexión a equipos de diagnóstico.

Galen Hospital: es un sistema cubano orientado hacia la informatización de la gestión de pacientes, la optimización del uso del personal, el incremento de la calidad de los servicios hospitalarios y la disminución de sus costos, para lo cual incluye módulos vinculados con el registro, admisión y alta de los pacientes. Uno de estos módulos es el Galen Lab, el cual va dirigido a la gestión de los medios de diagnóstico y las consultas, facilitando la solicitud de exámenes y el registro y evaluación de los resultados obtenidos, así como la generación de reportes estadísticos.

El Galen Hospital versión 2.0, fue desarrollado en plataforma Windows 32 bits (Windows NT y Windows 98) con una configuración Cliente/Servidor y el uso del gestor de bases de datos relacional SQL Server 7.0 como reservorio de la información. [4]

Bloque Quirúrgico Oftalmológico: sistema para la gestión clínico-quirúrgica de las instituciones especializadas en oftalmología. Entre sus características básicas se destacan: [5]

- Visualizar los datos registrados en la historia clínica siempre que se atienda a un paciente.
- Registrar las consultas realizadas al paciente.
- Registrar los medicamentos y tratamientos puestos al paciente a medida que pasa por las diferentes consultas.
- Visualizar antecedentes del paciente.
- Visualizar exámenes oftalmológicos.
- Permitir conocer la disponibilidad de lentes intraoculares.
- Permitir el cálculo de los lentes intraoculares por las diferentes fórmulas posibles.
- Introducir resultados de los exámenes complementarios.
- Crear y modificar un anuncio operatorio.

Usa como lenguaje de programación ASP.NET y sus datos son organizados con el gestor de base de datos PostgreSQL.

Sistema Automatizado de Microcirugía (SAMC): solución diseñada para el Hospital Oftalmológico “Ramón Pando Ferrer” de la Ciudad de la Habana. Este sistema tiene entre sus principales funcionalidades la gestión de la información relacionada con el pre y post operatorio solo del servicio de Microcirugía del Cristalino, así como el control y cálculo de los lentes intraoculares que se lleva a cabo en el mismo.

Presenta problemas con la gestión de la información a nivel de base de datos, específicamente a la hora de realizar reportes y consultas. Está implementado en "Clipper'85 Winter", lenguaje de programación creado en 1985 por Nantucket Corporation y herramienta líder de desarrollo bajo MS-DOS de aplicaciones relacionadas con bases de datos. [6]

Luego de analizar algunos de los productos existentes tanto en el mercado nacional como internacional, que le dan solución total o parcialmente a los problemas relacionados con la gestión de la información generada en las instituciones hospitalarias en el área de Consulta externa y en especial en las consultas de oftalmología, se arribó a las siguientes conclusiones.

- Muchos de estos productos no cumplen con la gestión de todos los procesos que se efectúan en las consultas externas y no incluyen lo referente al servicio de oftalmología. Por otra parte existen sistemas especializados en oftalmología que no contemplan los procesos del resto de las consultas externas ni de las diferentes áreas de las instituciones hospitalarias.
- La mayoría de los productos están desarrollados sobre software propietario, ocasionando considerables gastos tanto en la compra del producto, como en las licencias que son necesarias adquirir para los gestores de base de datos, servidores y sistemas operativos. Unido a esto se tiene el hecho de que muchos de estos sistemas son aplicaciones de escritorio, lo que trae consigo la necesidad de equipos con mayores prestaciones en relación al uso de memoria y espacio en disco.

Teniendo en cuenta todos estos aspectos, se plantea la necesidad de desarrollar un sistema basado en tecnologías libres, que englobe todos o la mayoría de los procesos que se llevan a cabo en el servicio de oftalmología del área de Consulta externa de las instituciones hospitalarias. A continuación se hace mención de las tecnologías a utilizar y sus características.

1.3 Tecnologías actuales a utilizar

1.3.1- Sistemas distribuidos. Modelo Cliente - Servidor

Los sistemas distribuidos no son más que una colección de ordenadores autónomos, enlazados por una red de ordenadores y soportados por un software que hace que el usuario perciba un único e integrado sistema de computación aún cuando las máquinas estén dispersas geográficamente.

Una de las arquitecturas más conocidas y más ampliamente adoptadas en la actualidad por los sistemas distribuidos es la arquitectura Cliente – Servidor. Esta consiste básicamente en un programa cliente que realiza peticiones a otro programa, el servidor, que le da respuesta sin necesidad de que ambos se estén ejecutando en el mismo ordenador. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. [7]

El uso de esta arquitectura tiene entre sus principales ventajas:

- Centralización de los recursos, ya que mediante el servidor se puede administrar los recursos comunes para todos los usuarios.
- Centralización del control, ya que los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Fácil mantenimiento debido a que al estar distribuidas las funciones y responsabilidades en diferentes equipos independientes, se puede reemplazar, reparar, actualizar o trasladar un servidor, mientras sus clientes no son afectados por el cambio.
- Como la información no se almacena en los clientes, no hay necesidad de dedicar espacio de disco a almacenar los datos, por lo que los gastos en hardware se pueden minimizar.

1.3.2- Patrones de arquitectura y diseño

Un patrón es un modelo o guía que se puede seguir para realizar algo, estos capturan la experiencia existente y probada para promover buenas prácticas. Para el desarrollo de un software existen varios patrones a tener en cuenta como son los de arquitectura, encaminados a describir la estructura u organización de un software, teniendo en cuenta determinados componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción. También existen los patrones de diseño que proporcionan un esquema para refinar los subsistemas o componentes de un sistema software y las relaciones entre ellos, por lo que están más próximos a lo que sería el código fuente final.

El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software, que permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Esto proporciona múltiples vistas sobre un mismo modelo de datos. Este patrón se usa frecuentemente en aplicaciones web donde se utilicen diferentes interfaces de usuario. Los tres elementos esenciales de este patrón son los siguientes: [8]

Modelo: Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio para modificar el estado de estos datos, habitualmente desde el controlador.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Entre las ventajas que tiene este modelo están:

- Soporta múltiples vistas: Como la vista se encuentra separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación y como el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

1.4 Tecnologías utilizadas en el proceso de desarrollo

Debido a la necesidad de que el sistema esté libre de costos relacionados con patentes de software para el servidor de aplicaciones, servidor de base de datos y sistema operativo huésped, se presentan a continuación una serie de tecnologías y herramientas para el proceso de desarrollo que no necesitan licencia para su utilización. Las mismas se describirán agrupadas según su uso en el patrón Modelo – Vista - Controlador.

1.4.1- Vista.

Java Server Faces (JSF) es un framework Java que permite crear interfaces de usuario (UI) para aplicaciones web, mediante componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web, la especificación de la navegación del usuario.

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

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

y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización del —look-and-feel- que puede ser fácilmente incorporado dentro de las aplicaciones JSF. [10]

Ajax es el conjunto de tecnologías que permite que al ser necesario que desde una página web se ejecute una acción en el servidor, esta se realice de forma asíncrona y se busquen los datos que son usados para actualizar la página mostrándose u ocultándose porciones de la misma sin necesidad de que sea recargada toda la página nuevamente. [11]

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, controlar cualquier evento de usuario, entre otras. [12]

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

XHTML, acrónimo inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas. [13]

Cascading Style Sheets (CSS), las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Lo que se persigue con el desarrollo de CSS es separar la estructura de un documento de su presentación. [14]

1.4.2- Controlador

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB), funcionando, según versa su significado en español, como una “costura” entre estos componentes. Seam también se integra perfectamente con otros frameworks como: RichFaces, ICEFaces, MyFaces, Hibernate y Spring. [15]

Drools es un sistema de administración de reglas de negocio que se vale de la programación declarativa para decir “qué hacer” y no “cómo hacerlo”. Reduce la brecha entre el negocio y las Tecnologías de la Información y las Comunicaciones (TIC), aumentando considerablemente el mantenimiento y la calidad del código escrito por los desarrolladores. Su principal beneficio es que permite separar la lógica de negocio del código de la aplicación, de esta manera un cambio en el flujo del negocio no implica la recompilación ni la aplicación de modificaciones al sistema. [16]

1.4.3- Modelo

Hibernate es una herramienta de mapeo objeto relacional. Utiliza para esto archivos declarativos (XML) que permiten establecer estas relaciones. Es una tecnología de software libre distribuida bajo los términos de la licencia GNU LGPL.

Como todas las herramientas de su tipo, busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Le permite a la aplicación manipular los datos de la base de datos operando sobre objetos, con todas las características de la programación orientada a objetos, Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL.

Esta herramienta genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias. Logra mantener la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Hibernate ofrece también un lenguaje

de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API¹ para construir las consultas programáticamente conocida como "Criteria". [17]

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

JPA es la Interfaz de Programación de Aplicaciones (API) estándar para la persistencia y el mapeo objeto/relacional para la plataforma Java EE, y permite utilizar un modelo de dominio Java para administrar bases de datos relacionales. [18]

Existe un conjunto de tecnologías que sirven de soporte a las antes mencionadas. Dichas tecnologías se describen a continuación.

1.5 Tecnologías Horizontales

Java es un lenguaje de programación desarrollado por la compañía Sun Microsystems, que utiliza el paradigma de la Programación Orientada a Objetos (POO). Es un lenguaje robusto pues no permite el manejo directo del hardware ni de la memoria. Dentro de sus principales ventajas se encuentra la de ser multiplataforma. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. [19]

Java Platform Enterprise Edition (JavaEE 5.) o Java versión 5 es una plataforma de programación (parte de la Plataforma Java) para desarrollar y ejecutar software de aplicaciones en lenguaje de

¹ Interfaz de Programación de Aplicaciones (API) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca que permite la comunicación entre componentes de un software.

programación Java con arquitectura de N niveles distribuida. Se basa ampliamente en componentes de software modulares y se ejecuta sobre un servidor de aplicaciones. [20]

JRE 6.

JRE es el acrónimo de Java Runtime Environment (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas. JVM (máquina virtual Java) es una instancia de JRE en tiempo de ejecución. Este interpreta el código Java y está compuesto además por las librerías de clases estándar que implementan el API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto. [21]

JBoss Server es el servidor de aplicaciones Java más utilizado actualmente en el mercado. Cientos de profesionales y desarrolladores de código abierto han contribuido a su creación y desarrollo. Es una plataforma certificada que cumple con las especificaciones de J2EE para el despliegue y desarrollo de aplicaciones empresariales Java, aplicaciones web y portales. Provee servicios extendidos de almacenamiento de datos en memoria y de manera persistente. Permite la integración de todas las tecnologías y herramientas utilizadas por JBoss Seam. Es actualizado e integrado constantemente con lo último del estado del arte de las aplicaciones web. [22]

PostgreSQL es un Sistema Gestor de Base de Datos (SGBD, por sus siglas en inglés) relacional de código abierto, muy poderoso, con una arquitectura probada. Puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día. Posee protección de transacciones u operaciones de Atomicidad, Consistencia, Aislamiento, Durabilidad (ACID, por sus siglas en inglés). Es un gestor de base de datos empresarial, que posee características sofisticadas como Control de Concurrencia Multi - Versión (MVCC, por sus siglas en inglés), replicación asíncrona, transacciones anidadas, realización de respaldo de datos en línea, optimizador o planificador de consultas, soporta internacionalización. Es altamente escalable en cuanto a la cantidad de información que puede manejar y al número de usuarios concurrentes que puede alojar. [23]

El tamaño máximo para las bases de datos PostgreSQL es ilimitado. Es totalmente compatible con el estándar ANSI-SQL 92/99. Permite realizar relaciones de herencia entre tablas, definir sistemas de reglas y eventos. PostgreSQL ejecuta procedimientos almacenados en más de una docena de lenguajes de programación tales como Java, Perl, Python, Ruby, Tcl, C/C++, y su propio lenguaje PL/pgSQL. De esto se deriva que existan muchas librerías para realizar la conexión a PostgreSQL desde varios lenguajes, compilados o interpretados, entre ellos ODBC, Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme, para nombrar algunos. Desde el año 1999 hasta la fecha ha ganado un gran número de premios que ratifican a este SGBD como el más completo dentro del mundo del código abierto. [24]

1.6 Capability Maturity Model Integration (CMMI)

Capability Maturity Model Integration (CMMI) o Modelo de Capacidad y Madurez es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios.

CMMI es la nueva generación de una línea de modelos de madurez que proponen un conjunto de prácticas que las organizaciones pueden adoptar para implantar procesos productivos más efectivos. Son llamados *modelos de madurez* porque proponen adoptar dichas prácticas en forma gradual: primero deben ponerse en práctica áreas de proceso² pertenecientes a un nivel determinado, para luego, sobre esta base, introducir las correspondientes al nivel siguiente. [25]

CMMI consta de 22 áreas de proceso distribuidas dentro de 5 niveles de madurez:

- Nivel 1: Inicial.
- Nivel 2: Administrado.
- Nivel 3: Definido.
- Nivel 4: Cuantitativamente administrado.
- Nivel 4: Optimizado.

² **Área de proceso:** conjunto de prácticas, que al ser ejecutadas colectivamente, permiten cumplir con algún objetivo de importancia para la organización.

Este enfoque en el cual se definen niveles de madurez para la organización se denomina “Representación Escalonada”.

CMMI propone otro enfoque en dependencia de la capacidad del proceso, estableciendo 6 niveles de capacidad posibles para cada una de las 22 áreas de proceso. La organización puede decir qué áreas de proceso quiere mejorar determinando así su perfil de capacidad. Esto se conoce como “Representación Continua”.

En la representación Escalonada cada área de proceso tiene metas específicas y genéricas. Las metas específicas describen las características únicas que deben estar presentes para satisfacer las necesidades de dicho área. A su vez, cada meta específica consta de un conjunto de prácticas específicas y subprácticas. Las prácticas específicas describen las actividades que se deben efectuar para producir logros en dicha meta de un área de proceso y las sub prácticas constituyen descripciones detalladas que proporcionan una guía para interpretar e implementar una práctica específica o genérica. Las metas genéricas describen las características que deben estar presentes para institucionalizar (llevar a cabo en otras áreas) los procesos que implementa un área. Estas metas tienen un grupo de prácticas genéricas, las cuales se asocian a las distintas áreas de un mismo nivel.

Actualmente la Universidad de las Ciencias Informáticas se encuentra inmersa en un proceso de mejora para certificarse en el Nivel 2 de CMMI, en el cual se encuentran las siguientes áreas de proceso.

- Planeación del Proyecto (PP).
- Monitoreo y Control del Proyecto (PMC).
- Administración de Acuerdos con Proveedores (SAM).
- Medición y Análisis (MA).
- Aseguramiento de la Calidad de Procesos y productos (PPQA).
- Administración de la Configuración (CM).
- Administración de Requisitos (REQM).

En el caso particular del área REQM tiene como propósito administrar los requisitos del proyecto, identificar inconsistencias entre dichos requisitos, los planes del proyecto y los productos de trabajo para

lo cual se guía por el proceso IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos. Este libro incluye la relación de roles que ejecutan las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades. Además, se establece la relación de los subprocesos con las fases que conforman el ciclo de vida, las cuales son descritas en el libro 0505_Ciclo de vida de proyectos pilotos del programa de mejora.

Los libros antes mencionados fueron confeccionados por la Universidad de las Ciencias Informáticas siguiendo lo establecido por CMMI en su nivel 2 para el área REQM y el criterio de expertos, los cuales, en su mayoría, habían utilizado como metodología de desarrollo el Proceso Unificado de Desarrollo (RUP).

Para el desarrollo de los productos de trabajo definidos en el Libro de Proceso para la Administración de Requisitos se utilizará el lenguaje de modelado UML, sin embargo en el caso específico de la fase Modelación de negocio se hace uso del estándar para la descripción de procesos de negocio BPMN.

RUP es el resultado de varios años de desarrollo y uso práctico, en el que se han unificado técnicas de desarrollo y trabajo de muchas metodologías utilizadas por los clientes. En la actualidad RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. En RUP las actividades están agrupadas en grupos lógicos definiéndose 9 flujos de trabajos principales, 6 de ingeniería y 3 de apoyo, además cuenta con tres características fundamentales: centrado en la arquitectura, guiado por casos de uso e iterativo e incremental.

Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con tecnología orientada a objetos. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

Notación para el Modelado de Procesos de Negocio (BPMN): es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo.

El principal objetivo de BPMN es proveer una notación estándar que sea fácil de entender por parte de todos los involucrados e interesados del negocio, díganse los analistas de negocio que son los encargados de definir y redefinir los procesos, los desarrolladores técnicos quienes implementan los procesos y los gerentes y administradores del negocio que son lo que monitorean y gestionan los procesos. [26]

1.7 Herramientas de desarrollo

Eclipse es un entorno de desarrollo integrado de código abierto, portable y multiplataforma. Este fue diseñado originalmente por la empresa IBM y actualmente, es desarrollado por la Fundación Eclipse, una organización independiente, sin ánimo de lucro que fomenta una comunidad de código abierto. [27]

Se basa en el uso de módulos (*plugins*), lo cual hace posible el trabajo en múltiples lenguajes de programación como son Java, C++, PHP, Perl y que se le puedan añadir otras funcionalidades. Mediante el SDE Enterprise edition permite la integración con la herramienta Visual Paradigm, propiciando un mejor entendimiento de todas las partes involucradas en el desarrollo del sistema. Cuenta además con un sistema de control de versiones, el cual usando una combinación de vistas y editores que muestran los diversos aspectos de los recursos del proyecto organizados por el rol o la tarea del desarrollador, hace más fácil y eficiente el trabajo en equipo.

PgAdmin es una aplicación gráfica que se utiliza para interactuar con el gestor de bases de datos PostgreSQL. Esta aplicación está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux. Presenta una interfaz gráfica amigable con todas las características de PostgreSQL, facilitando enormemente la administración de la base de datos. La aplicación incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, entre otras funcionalidades que hacen de este una de las herramientas más usadas con licencia Open Source en la actualidad.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objeto, construcción, pruebas y despliegue. Ofrece una ayuda para la construcción más rápida de aplicaciones de calidad, ya que permite dibujar todos los tipos de diagramas de clases, provee soporte para la generación de código, tiene integración con diversos IDE's

como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate.

En este capítulo se hizo mención de un conjunto de tecnologías que garantizan que el sistema a desarrollar sea independiente de costos por concepto de compra de licencias de software, que sus partes estén totalmente integradas con el fin de hacer pocas transformaciones ante la necesidad de un cambio y sea amigable, fácil de usar y multiplataforma.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se hace una descripción de los principales procesos asociados al área de Consulta externa en el servicio de oftalmología, específicamente en la consulta de Microcirugía del Cristalino y las consultas de Aprobación de cirugía, ofreciendo como apoyo para una mejor comprensión, los diagramas de procesos del negocio. Además se identifican los actores y casos de uso del sistema, así como los requisitos funcionales y no funcionales que soportará el mismo.

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

Atender paciente en la consulta especializada de oftalmología.

Un paciente llega a la consulta de Microcirugía del Cristalino del área Consulta externa en el servicio de oftalmología por una remisión de otro centro hospitalario, una cita de interconsulta o por decisión personal, en cualquiera de los casos el médico especialista puede enviarlo a otro servicio especializado en dependencia del padecimiento que presente.

El especialista en oftalmología crea una Hoja de Consulta del Cristalino, reflejando los datos generales del paciente, los antecedentes patológicos generales y los antecedentes patológicos oftalmológicos, ambos personales y/o familiares, y los resultados de los exámenes físicos oftalmológicos realizados a través de la inspección y observación a uno o ambos ojos. En caso de ser necesario realiza pruebas diagnósticas con equipos especiales, como *fondo de ojo*, *medición de la agudeza visual*, entre otras, las cuales se anexan a la hoja de consulta. Basándose en toda esta información el facultativo emite un diagnóstico y le entrega al paciente las indicaciones médicas que debe cumplir; posteriormente decide cuál es la conducta que este debe seguir, que puede ser quirúrgica o no. En el mejor de los casos en el que la conducta a seguir no es quirúrgica el médico puede darle alta al paciente, realizar una solicitud de interconsulta u orientarlo a una consulta de seguimiento; en otro caso se crea la Solicitud de Intervención Quirúrgica del Cristalino.

Asignar lente intraocular a un paciente

En la Solicitud de Intervención Quirúrgica del Cristalino se recogen una serie de datos necesarios para efectuar la intervención quirúrgica, dentro de los cuales se encuentran los referentes a el(los) lente(s) que se le implantará(n) al paciente en lugar del cristalino. Dicho(s) lente(s) presenta(n) un tipo y poder dióptrico, el cual es calculado por el especialista usando reglas y fórmulas preestablecidas, y apoyándose en las pruebas diagnósticas. Una vez determinado el o los lentes, el facultativo consulta el Registro de lentes para ver si este(os) está(n) en existencia en el almacén.

En caso negativo se crea el Modelo de paciente pendiente, en el cual quedan reflejados los datos del paciente y del lente o los lentes que este necesita, posteriormente este modelo se inserta en el Registro de pacientes pendientes, en el que se encuentran todos los pacientes que están en espera de lentes.

En el caso de que exista(n) en el almacén el o los lentes que el paciente necesita, este(os) se le asigna(n) al mismo y se actualiza la Solicitud de Intervención Quirúrgica del Cristalino, agregándole los datos del lente o los lentes en cuestión.

Control del lente

Para llevar a cabo un correcto control del lente intraocular se realizan un conjunto de procesos con este fin.

- **Dar alta a lentes intraoculares**

Una vez que llega a la institución hospitalaria un lote de lentes, el administrador es el encargado de revisar que estos estén en buen estado. Posteriormente consulta el Registro de pacientes pendientes para verificar que no existan pacientes esperando por los lentes del lote. En este caso, consulta el Registro de lentes comprobando que el modelo de los lentes del lote esté en existencia. Si no está en existencia, se crea el Modelo de lentes con los datos del mismo y se actualiza dicho registro agregando esta información e incrementando la cantidad de lentes. Existiendo o no, ese modelo en el Registro de lentes, el administrador crea el Modelo de alta y agrega el mismo al Registro de movimientos.

En caso de que existieran pacientes aguardando por los lentes del lote se procede a realizar la *asignación de lentes intraoculares a los pacientes pendientes*. Después de culminado este proceso, se consulta el

lote de lentes con el objetivo de ver si aún quedan lentes y de esta forma darle alta oficialmente a los mismos.

- **Asignar lente intraocular a pacientes pendientes**

El especialista oftalmológico es el encargado de consultar el Registro de pacientes pendientes, seleccionar el paciente que más tiempo lleve en espera y asignarle el(los) lente(s) que necesite. Seguido a esto disminuye la cantidad disponible de lentes del lote y envía un telegrama al paciente para que confirme si está en disposición de operarse en la fecha prevista.

En caso de que el paciente se encuentre indispuerto se le retira la asignación del lente o los lentes y se aumenta la cantidad de lentes disponibles en el lote. Si el paciente no se realizará la intervención quirúrgica en otro momento, se elimina del Registro de pacientes pendientes y su Solicitud de Intervención Quirúrgica del Cristalino. Dado el caso de que el paciente se encuentre en plena disposición de operarse, se le elimina del Registro de pacientes pendientes y se actualiza su Solicitud de Intervención Quirúrgica del Cristalino con la nueva fecha de cirugía.; así mismo elimina oficialmente del lote los lentes asignados. Este proceso se realiza hasta que se hayan agotado los lentes del lote o ya no existan pacientes esperando por estos.

- **Dar baja a lentes intraoculares**

Puede darse el caso de que algún(s) lente(s) presente(n) defecto, se haya(n) roto o vencido, en tal caso es responsabilidad del administrador darle baja al lente o los lentes. Para esto crea el Modelo de baja, lo adiciona al Registro de movimientos y actualiza el Registro de lentes disminuyendo la cantidad de lentes en cuestión.

- **Eliminar paciente pendiente**

Cada cierto período de tiempo es responsabilidad del Administrador eliminar los pacientes pendientes que hayan estado esperando por un(os) lente(s) durante mucho tiempo. Para ello consulta el Registro de lentes y elimina los pacientes pendientes a partir de una fecha dada; eliminando, además, su Solicitud de Intervención Quirúrgica del Cristalino.

Atender paciente en la Consulta de Aprobación

Un paciente asiste a la Consulta de aprobación cuando el especialista, de cualquiera de los servicios ofrecidos por la consulta externa de oftalmología, le indicó como conducta a seguir la intervención quirúrgica. Esta Consulta de Aprobación se divide a su vez en dos consultas, la del Pediatra, en caso de que el paciente tenga menos de 16 años, y la del Clínico, si el paciente tiene de 17 años en adelante. Dichas consultas tienen como objetivo determinar si el paciente está apto quirúrgicamente.

Una vez que el paciente se presenta en la consulta, el Pediatra o Clínico le pide la Solicitud de Intervención Quirúrgica del Cristalino y recoge en la Hoja de consulta del Clínico o Pediatra, según corresponda, datos tanto personales como familiares que son de interés para la consulta. Luego si el facultativo lo considera necesario, le indica análisis de laboratorio para lo cual crea la Solicitud de análisis de laboratorio. Dichos análisis son realizados por el Técnico, que una vez terminado le envía los resultados al médico. Este teniendo en cuenta la información obtenida del paciente y el resultado de los análisis del mismo, realiza una valoración del estado de salud del paciente y determina si este se encuentra apto quirúrgicamente o no, lo cual queda plasmado en la Hoja de consulta del Clínico o Pediatra.

En caso de que no se le pueda efectuar la operación al paciente, el Pediatra o Clínico puede proceder de varias formas: dándole alta al paciente, enviándolo a una interconsulta para que sea valorado por otro especialista o a una consulta de seguimiento con el objetivo de ver cómo ha evolucionado su enfermedad y en un futuro, si es posible, darle el apto quirúrgico, para esto le entrega un turno o cita. El proceder es registrado en la Hoja de consulta del Clínico o Pediatra.

2.2- Objeto de automatización.

A raíz de la descripción de los procesos anteriores surge la necesidad de la creación de un sistema para automatizar las actividades y documentos que intervienen en los mismos. Esto traerá consigo que toda la información que se genera de forma manual se elimine y así disminuya el almacenamiento de volúmenes elevados de información.

Para efectuar la atención del paciente en la consulta oftalmológica en el servicio del Microcirugía Cristalino, el especialista crea la Hoja de Cirugía del Cristalino en la cual se contempla la información referente al paciente, se anexan las pruebas diagnósticas y el resto de las acciones realizadas al mismo durante la consulta.

La asignación de lente intraocular a un paciente es un proceso que se lleva a cabo cuando a este se le indica una cirugía del Cristalino y el resultado del mismo queda reflejado en la Solicitud de Intervención Quirúrgica del Cristalino. Con la automatización de este proceso el especialista oftalmológico no tendrá que realizar manualmente actividades como el cálculo del lente que se le implantará al paciente, actualizar la Solicitud de Intervención Quirúrgica del Cristalino con los datos del lente obtenido, crear el Modelo de paciente pendiente ni agregar al paciente en el Registro de pacientes pendientes.

Además, es necesario automatizar el control del lente intraocular, en el cual se encuentran procesos como el Alta del lente, donde se crean el Modelo de lentes, cuando el entrada en el lote de lentes no está en existencia en el almacén y el Modelo de Alta, cuando se ingresan oficialmente los lentes al almacén. Unido a esto se visualizan el Lote de lentes, el Registro de pacientes pendientes y el Registro de lentes, este último, al igual que el Registro de Movimientos, se actualizan. Otro proceso que se realiza es la Baja del lente, en el que se crea el Modelo de baja y se actualizan el Registro de Movimientos y el Registro de lentes.

También se efectúa la Asignación de lentes intraoculares a pacientes pendientes, para ello se visualiza el Registro de pacientes pendientes, se elimina al paciente de dicho registro, se actualiza la Solicitud de Intervención Quirúrgica del Cristalino con la fecha oficial de la operación y el lote de lentes, disminuyendo y/o aumentando la cantidad disponible de lentes. Además se eliminan el paciente del Registro de pacientes pendientes y la Solicitud de Intervención Quirúrgica del Cristalino, dado el caso que el paciente no se pueda realizar la intervención quirúrgica en la fecha asignada ni en otro momento. Por último se lleva a cabo el proceso de Eliminar pacientes pendientes, para ello se consulta el Registro de pacientes pendientes y se eliminan los pacientes pendientes del mismo a partir de una fecha dada; eliminando, además, la Solicitud de Intervención Quirúrgica del Cristalino.

También resulta importante automatizar el proceso que se lleva a cabo en las Consultas de Aprobación, destacándose como actividad fundamental la creación de la Hoja de consulta del Clínico o Pediatra, en la que se reflejan todas las acciones y valoraciones hechas al paciente.

2.3- Modelo de negocio

El principal objetivo del Modelo de negocio es comprender problemas actuales de organización e identificar mejoras que se pueden hacer para darle solución a los mismos, además de que clientes, desarrolladores y el resto del personal involucrado en la creación del sistema tengan un entendimiento común de la organización.

Para lograr este objetivo se usará la notación BPMN que está formada por un conjunto de elementos gráficos con los que se confecciona el diagrama de procesos del negocio, basado en la técnica de diagrama de flujo. Estos elementos se dividen en cuatro categorías básicas: Objetos de flujo, Objetos de conectores, Artefactos y Swimlanes.

Dentro de los *Objetos de flujo* se encuentran las actividades que representan las acciones que se llevan a cabo en un proceso. Estas pueden ser atómicas (actividades que no pueden ser descompuestas en otras actividades) y compuestas (actividad compuesta por agrupaciones de flujos de actividades). Otro de los elementos que este encierra es el Gateway o las llamadas decisiones, usándose para controlar la divergencia o convergencia de la secuencia de flujos. Por último están los eventos, que representan algo que “pasa” durante el curso del proceso de negocio y que afecta el flujo de actividades del mismo, suelen tener una causa o un impacto.

Los *Objetos conectores* son los encargados de conectar los objetos de flujo, entre ellos se encuentran:

- Flujo de secuencia: se utiliza para representar el orden en que se ejecutan las diferentes actividades.
- Flujo de mensajes: representa el flujo de mensajes entre dos participantes del proceso.
- Asociaciones: se emplea para asociar datos, textos y otros artefactos con los objetos del flujo. Representan entradas y salidas de las actividades.

Swimlanes es un mecanismo para organizar actividades en categorías separadas visualmente con el objetivo de ilustrar diferentes capacidades funcionales o responsabilidades. Los objetos swimlanes son:

- Piscina (pool): representa un participante de un proceso
- Pista (lane): representa una sub-partición dentro de una piscina, usada para organizar y categorizar las actividades de las mismas.

Según lo establecido por la universidad, rigiéndose por el área de procesos RQM de CMMI, toda la información que se obtiene a partir de la fase Modelación del negocio debe ser reflejada en el documento “Modelo de Procesos de Negocio con BPM”, el cual tiene como objetivo asegurar el entendimiento del funcionamiento interno del negocio por clientes, usuarios, desarrolladores y otras partes; identificar las restricciones que impone el negocio y que el sistema debe cumplir y asegurar la aprobación y conformidad del cliente ante la interpretación de los procesos del negocio y cómo el software puede mejorarlos.

A continuación se presentan los participantes dentro de los procesos de negocio y algunos de los diagramas de estos procesos.

2.3.1- Participantes

Participante	Descripción
Administrador	Persona que se ocupa de atender el control del lente intraocular
Especialista oftalmológico	Médico especialista en oftalmología encargado de brindarle la atención al paciente
Paciente	Principal beneficiado de los procesos del negocio
Clínico/Pediatra	Médico especialista en la materia encargado de brindarle la atención al paciente

Tabla 2.1- Actores del negocio

2.3.2- Diagramas de Procesos del Negocio

2.3.2.2- Dar alta a lentes intraoculares

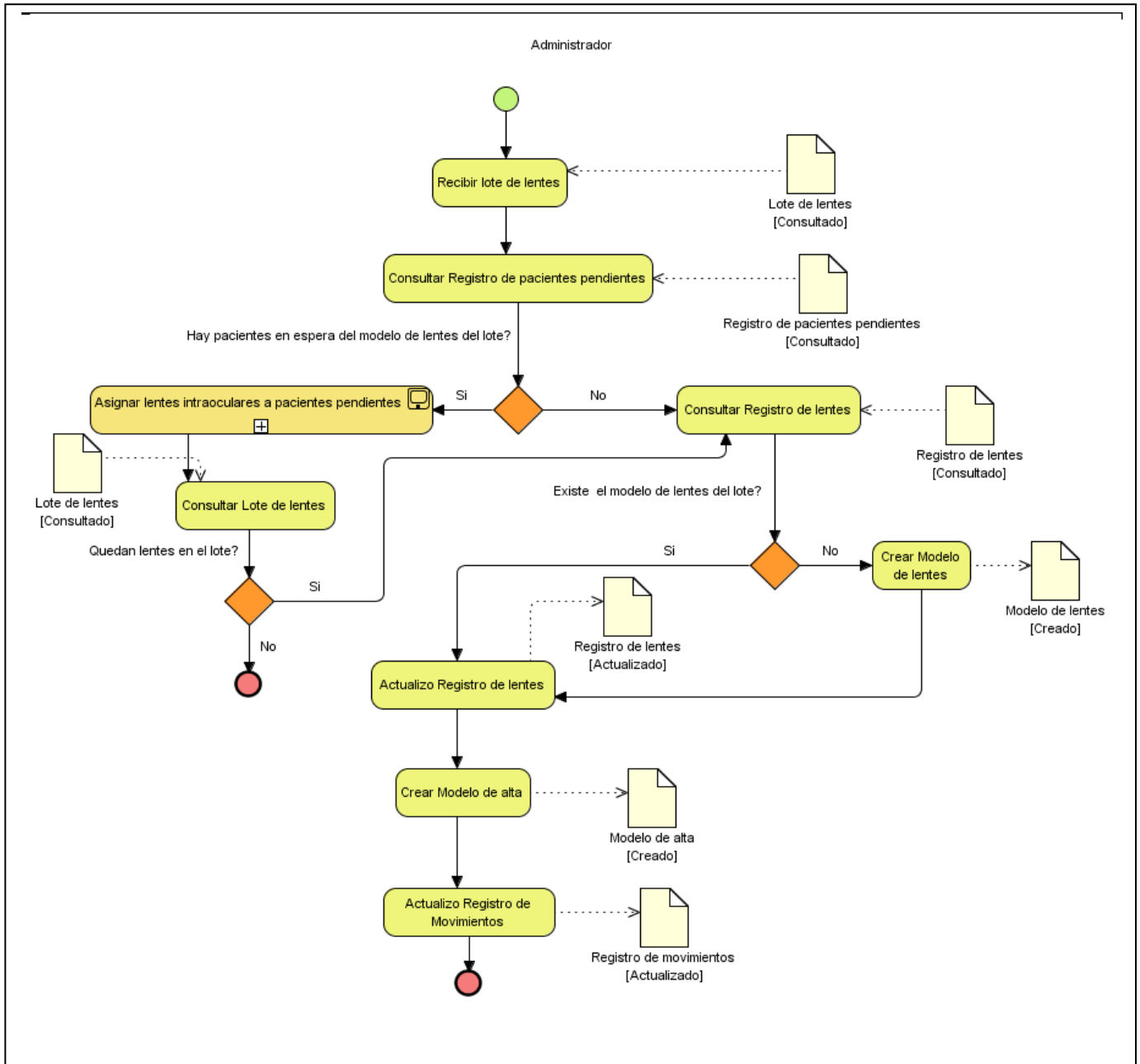


Figura 2.1- Diagrama de procesos del negocio: Dar alta a lentes intraoculares

2.3.2.4- Dar baja a lentes intraoculares

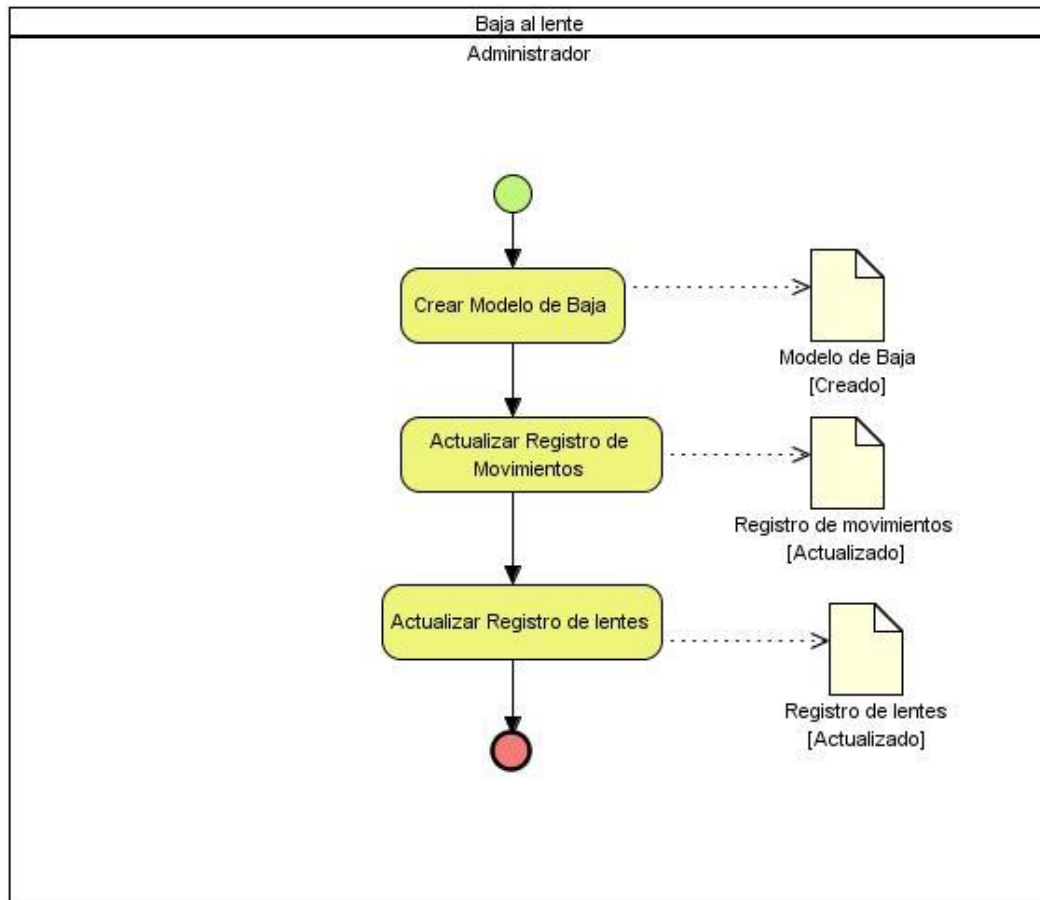


Figura 2.3- Diagrama de procesos del negocio: Dar baja a lentes intraoculares

2.4- Especificación de los requerimientos de software

2.4.1- Requisitos funcionales del sistema.

Los Requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y que definen el comportamiento interno del software. Estos describen lo que el sistema debe ser capaz de realizar para satisfacer las necesidades del usuario.

1. Crear Lote de lentes intraoculares
2. Ver detalles del Lote de lentes intraoculares
3. Modificar Lote de lentes intraoculares
4. Eliminar Lote de lentes intraoculares
5. Seleccionar Lote de lentes intraoculares
6. Crear Modelo de paciente pendiente
7. Ver datos del Modelo de paciente pendiente
8. Modificar Modelo de paciente pendiente
9. Eliminar Modelo de paciente pendiente
10. Seleccionar Modelo de paciente pendiente
11. Buscar Modelo de paciente pendiente
12. Asignar lente intraocular
13. Crear Modelo de lentes intraoculares
14. Ver detalles del Modelo de lentes intraoculares
15. Modificar Modelo de lentes intraoculares
16. Eliminar Modelo de lentes intraoculares
17. Ver datos del Modelo de lentes intraoculares
18. Buscar Modelo de lentes intraoculares
19. Seleccionar Modelo de lentes intraoculares
20. Crear Lente intraocular

21. Ver detalles del Lente intraocular
22. Modificar Lente intraocular
23. Eliminar Lente intraocular
24. Seleccionar Lente intraocular
25. Buscar Lente intraocular
26. Ver datos del Lente intraocular
27. Crear Modelo de alta
28. Ver detalles del Modelo de alta
29. Modificar Modelo de alta
30. Eliminar Modelo de alta
31. Crear Modelo de baja
32. Ver detalles del Modelo de baja
33. Modificar Modelo de baja
34. Eliminar Modelo de baja
35. Calcular lente intraocular
36. Crear Hoja especializada del Cristalino
37. Ver datos de Hoja especializada del Cristalino
38. Crear Hoja de consulta de Aprobación
39. Ver datos de Hoja de consulta de Aprobación

2.4.2- Requisitos no funcionales

2.4.2.1- Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

Usuarios de pocos conocimientos informáticos: 30 días

Usuarios de avanzados conocimientos informáticos: 20 días

2.4.2.2- Fiabilidad

En los servidores de los hospitales se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se garantizarán además, políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema. Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros hospitales por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.

Se mantendrá la seguridad y el control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema. Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

El sistema implementará un control de cambios a determinados campos de información (seleccionados por su importancia), de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado. Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista. El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

2.4.2.3- Eficiencia

El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

2.4.2.4- Soporte

2.4.2.4.1- Seguridad de acceso y administración de usuarios

Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP.

2.4.2.4.2- Monitoreo de funcionamiento

Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.

2.4.2.4.3- Respaldo y recuperación de base de datos

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.

2.4.2.4.4- Auditoría

Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema, para esto debe existir un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso como mínimo: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información contenida en el registro modificado.

2.4.2.4.5- Configuración de parámetros

Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

2.4.2.4.6- Réplica

Se permitirá realizar réplica de la base de datos de los hospitales. Esta réplica se podrá hacer de forma manual y automatizada a través de la red.

2.4.2.5- Restricciones de diseño

El sistema estará dividido en las siguientes capas:

2.4.2.5.1- Capas físicas

Cliente: Computadora con cualquier tecnología o sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 6 o superior o Firefox 2.x).

Servidor de Aplicaciones: Servidor con cualquier tecnología o sistema operativo que soporte el Java Runtime Environment (JRE) 1.5 o superior y al JBoss AS 4.2 o superior.

Servidor de Base de Datos: Servidor con cualquier tecnología o sistema operativo que soporte a PostgreSQL Server 8.2 o superior en los servidores de base de datos de cada hospital, y Oracle 11g o superior para los servidores de base de datos del Centro de Datos.

2.4.2.5.2- Capas lógicas

Presentación: Contiene todas las vistas y la lógica de la presentación. El flujo web se maneja de forma declarativa y basándose en definiciones de procesos del negocio.

Negocio: Mantiene el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. En los casos de que algún objeto del negocio tenga una interfaz externa, siendo accesible la misma desde sistemas legados o directamente del cliente, se garantiza la seguridad a nivel de objeto y métodos.

Acceso a Datos: Contiene las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

2.4.2.6- Interfaz

2.4.2.6.1- Interfaces de usuario

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

Se incorporarán asistentes que faciliten el uso del sistema por los usuarios, en procesos con determinado nivel de complejidad, que lo guíen paso a paso para minimizar la posibilidad de errores. El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.

2.4.2.6.2- Interfaces de comunicación

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven). La utilización de este estándar aporta las siguientes ventajas:

- Es un estándar independiente de la plataforma y la tecnología.
- Está ampliamente implementado, por lo que es posible conseguir otros productos compatibles con HL7.
- Permite el desarrollo y mantenimiento de una interface única, común a todas las implementaciones, por tanto no es necesaria la definición de mensajería específica para cada sistema que se desee interconectar.

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

2.4.3- Modelo de casos de uso del sistema.

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Un actor es una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema.

Los actores del sistema identificados son:

Actores del sistema	Funciones
Administrador	Persona que se ocupa de atender el control del lente intraocular
Gestor de la Hoja especializada del Cristalino	Médico especialista en oftalmología encargado de brindarle la atención al paciente.
Gestor de la Hoja de Aprobación	Médico especialista en medicina general encargado de brindarle la atención al paciente.

Tabla 2.2 - Actores del sistema

Se identificaron un conjunto de casos de uso que son propios de otros módulos del sistema alas HIS, pero que por la funcionalidad que estos describen son de importancia en la consulta de Microcirugía del Cristalino en el área Consulta externa del servicio de oftalmología. Estos casos de uso son utilizados en el Diagrama de Casos de Uso con el siguiente formato: <Nombre del módulo>:: <Nombre del caso de uso>.

2.4.3.1- Diagrama de Casos de Uso del Sistema.

2.4.3.1.1- Diagrama de casos de uso públicos para otros módulos

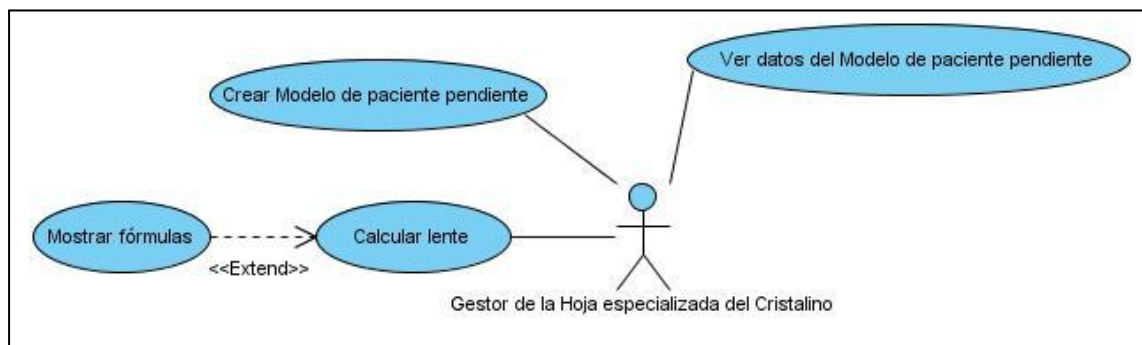


Figura 2.4- DCU públicos para otros módulos

2.4.3.1.2- Diagrama de caso de uso común dentro del módulo

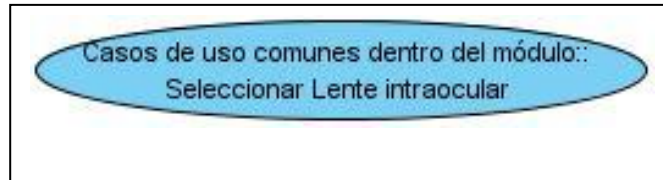


Figura 2.5- DCU comunes dentro del módulo

2.4.3.1.3- Diagrama de caso de uso Dar alta a lentes intraoculares

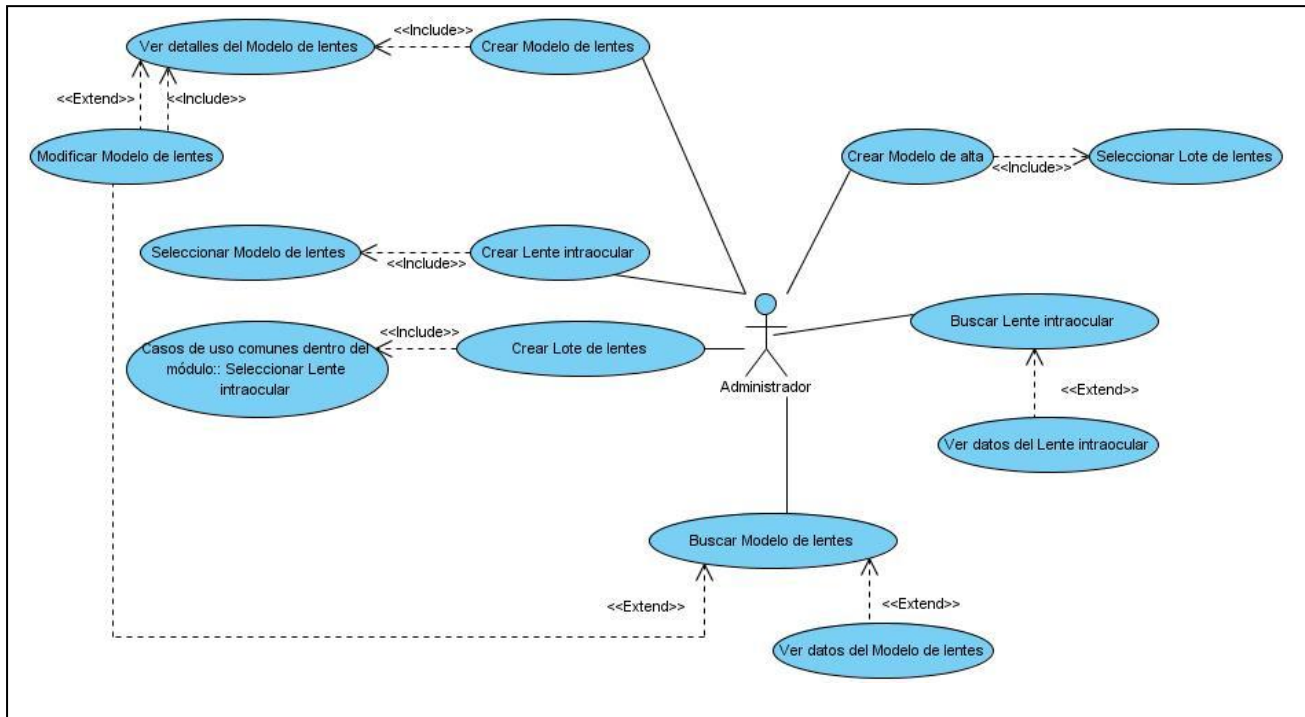


Figura 2.6- DCU Dar alta a lentes intraoculares

2.4.3.1.4- Diagrama de caso de uso Dar baja a lentes intraoculares

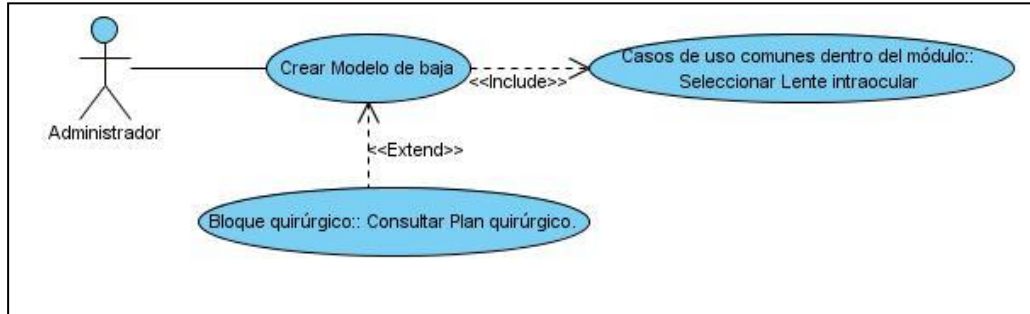


Figura 2.7- DCU Dar alta a lentes intraoculares Dar baja a lentes intraoculares

2.4.3.2- Descripción textual de los casos de uso

Objetivo	Seleccionar Lente intraocular	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el actor accede a la opción Seleccionar Lente intraocular, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el Lente intraocular deseado, el actor introduce los datos que considera como criterios para buscar el Lente intraocular, el sistema busca y muestra los Lentes intraoculares que cumplen con los criterios de búsqueda, el actor selecciona el Lente intraocular deseado, el sistema devuelve el Lente intraocular seleccionado para que sus datos sean manejados en la vista anterior, el caso de uso termina.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones		
Relaciones	CU Incluidos	
	CU	

	Extendidos	
--	-------------------	--

Tabla 2.3- Descripción textual del Caso de Uso: Seleccionar Lente intraocular

Objetivo	Ver detalles del Modelo de lentes intraoculares	
Actores	Administrador	
Resumen	El caso de uso inicia cuando es creado o modificado un Modelo de lentes intraoculares, el sistema muestra los datos del Modelo de lentes intraoculares, el caso de uso termina.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Para ver los datos de un Modelo de lentes intraoculares, este se debe haber creado o modificado.	
Relaciones	CU Incluidos	
	CU Extendidos	<i>Modificar Modelo de lentes intraoculares</i>

Tabla 2.4- Descripción textual del Caso de Uso: Ver detalles del Modelo de lentes intraoculares

Objetivo	Modificar Modelo de lentes intraoculares	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el actor ve los detalles del Modelo de lentes intraoculares y accede a la opción Modificar Modelo de lentes intraoculares, el sistema muestra los datos del Modelo de lentes intraoculares y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos del Modelo de lentes intraoculares, el caso de uso termina.	
Complejidad	Alta	
Prioridad	Secundario	
Precondiciones	Para modificar los datos de un Modelo de lentes intraoculares se deben haber visto sus datos.	

Relaciones	CU Incluidos	<i>Ver detalles del Modelo de lentes intraoculares</i>
	CU Extendidos	

Tabla 2.5- Descripción textual del Caso de Uso: Modificar Modelo de lentes intraoculares

Objetivo	Buscar Modelo de lentes intraoculares	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar Modelo de lentes intraoculares, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el Modelo de lentes intraoculares, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los Modelos de lentes intraoculares que cumplen con los criterios de búsqueda, el caso de uso termina.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones		
Relaciones	CU Incluidos	
	CU Extendidos	<i>Ver datos del Modelo de lentes intraoculares</i> <i>Modificar Modelo de lentes intraoculares</i>

Tabla 2.6- Descripción textual del Caso de Uso: Buscar Modelo de lentes intraoculares

En este capítulo se analizaron los procesos de negocio que se efectúan en el área de Consulta Externa en el servicio de oftalmología, específicamente en la consulta de Microcirugía del Cristalino y las consultas de Aprobación de cirugía, identificándose los requisitos funcionales y los no funcionales que debe cumplir el sistema. Además se definieron los actores y los casos de uso del sistema.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En el presente capítulo se hace una descripción detallada del sistema propuesto, describiendo la arquitectura definida por el Dpto. de Sistema de Gestión Hospitalaria para el desarrollo de sus aplicaciones, además se expone el diseño de la solución, haciendo énfasis en los artefactos necesarios para la implementación de la misma.

3.1- Descripción de la arquitectura

La Arquitectura de Software no es más que la estructura de un sistema informático, por lo que define los principales componentes del mismo y sus relaciones; además, contiene un grupo de patrones y abstracciones coherentes que sirven de guía para la construcción del software.

El sistema propuesto presenta como parte de la línea base de su arquitectura el patrón Modelo Vista Controlador, el cual separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos: el modelo para la administración de los datos; la vista que muestra las interfaces de usuario y el controlador para la gestión de las acciones del usuario, invocando cambios tanto en la vista como en el modelo. Con el uso de este patrón se garantiza la no dependencia entre la vista y el modelo, por lo que un cambio en la vista trae consigo poco o ningún impacto en los datos o en la lógica de negocio.

La vista está compuesta fundamentalmente por páginas XHTML, controles JSF, Seam UI y Richfaces, encargados de capturar y validar los datos introducidos por el usuario en su interacción con el sistema; además con el uso de los componentes ajax4jsf se logra optimizar el proceso de envío y carga de los datos. Todo esto provee a la interfaz de usuario de usabilidad y un agradable diseño.

El controlador está compuesto por clases controladoras que engloban la lógica del negocio. A estas clases se les especifica el contexto en que se encuentran (conversacional, evento, página, etcétera) usando las anotaciones del framework Seam, definiendo así el estado de los datos y las entidades que manejan.

El modelo tiene como principal función el acceso, modificación, eliminación y persistencia de datos en la base de datos, para ello utiliza Hibernate, que realiza todas estas acciones usando el paradigma orientado a objetos y abstrayendo al desarrollador del gestor de base de datos usado.

Una vez descrita la arquitectura se procede a diseñar los casos de uso del sistema.

3.2- Modelo de diseño

En el Modelo de diseño se estructura el sistema para que soporte tanto los requisitos funcionales como los no funcionales y las restricciones que se imponen, produciéndose un refinamiento del análisis y constituyendo una base para la implementación. Con el objetivo de lograr una mejor confección del diseño, se utilizan un conjunto de patrones o modelos que constituyen una solución a problemas que surgen con mucha frecuencia durante este.

Dentro de los patrones de diseño, los más usados son los Patrones Generales de Software para Asignar Responsabilidades (GRASP), que se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Dentro de este grupo se identifican cinco patrones muy utilizados: experto, creador, alta cohesión, bajo acoplamiento y el controlador [28]. Con el uso del patrón experto y creador, se logró que a cada clase del diseño se le asignaran tareas acordes a la información que estas manejaban y crear instancias de acuerdo a las responsabilidades de las mismas.

Para la construcción del Modelo de diseño se organizó el sistema en paquetes, usando el criterio de empaquetamiento por proceso, según los procesos definidos en el sistema; logrando de esta forma la estructura principal del sistema. A continuación se presenta el Diagrama de paquetes del diseño.

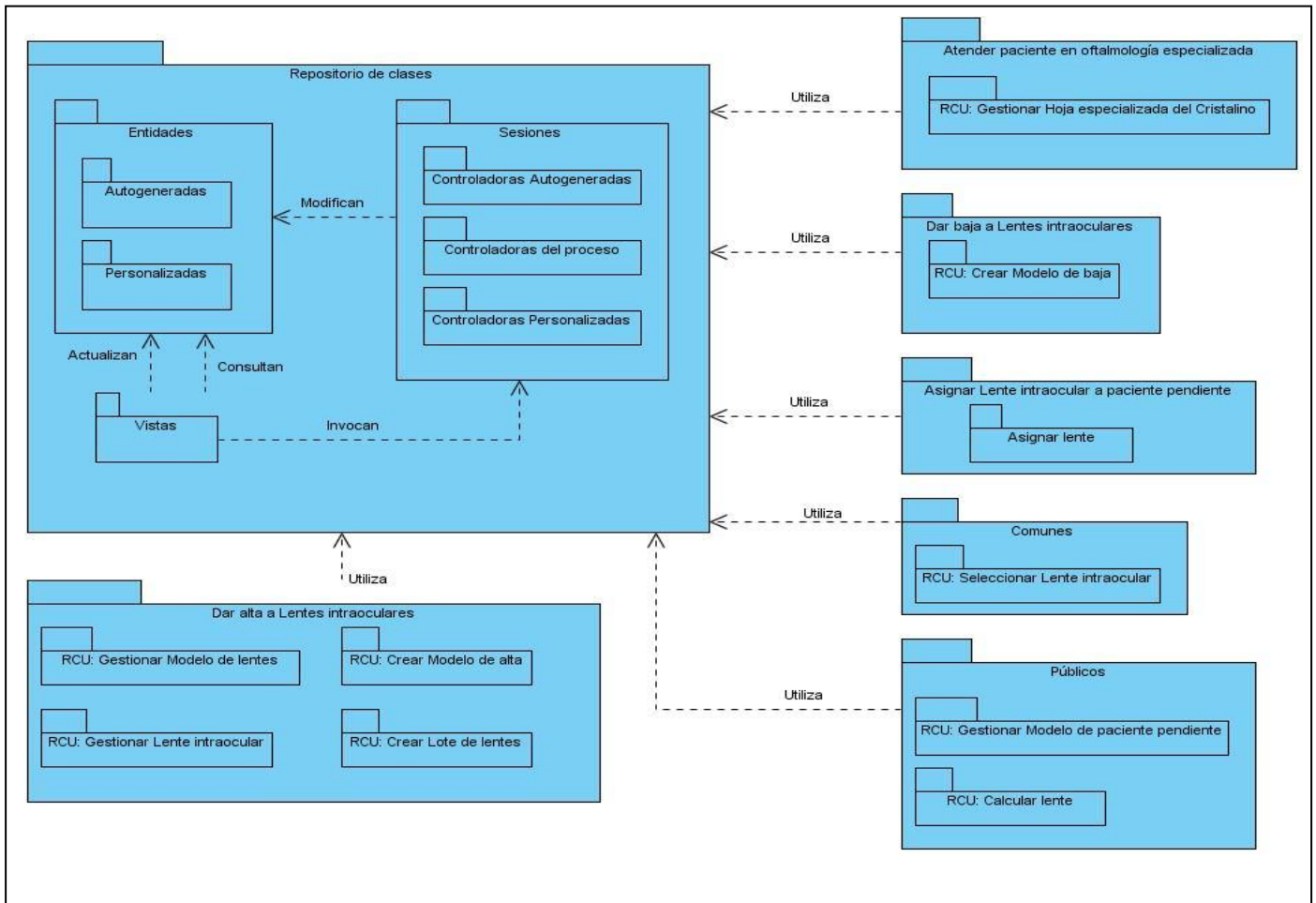


Figura 3.1- Diagrama de paquetes del diseño

El Diagrama de paquetes del diseño está formado por los siguientes paquetes:

Repositorio de clases, contiene las clases definidas en el diseño de acuerdo a las tecnologías utilizadas en la implementación del módulo, dividido en tres paquetes:

Entidades: contiene las entidades autogeneradas de la base de datos, así como las personalizadas, que no son más que las que han tenido cambios por la lógica de negocio.

Sesiones: contiene las clases controladoras autogeneradas por el entorno de desarrollo, las controladoras del proceso y las controladoras personalizadas que son modificadas.

Vistas: contiene las clases que representan la interfaz de usuario.

En los paquetes correspondientes a los procesos de Atender paciente, Dar baja a Lentes intraoculares, Asignar Lente intraocular a paciente pendiente y Dar alta a Lentes intraoculares, se encuentran las realizaciones de los casos de usos identificados en cada uno de estos procesos. Estas realizaciones de los casos de uso, son representadas por las clases del diseño y sus relaciones en el Diagrama de clases del diseño y por los Diagramas de interacción. A continuación se definen cada uno de estos.

Una *clase del diseño* es una abstracción de una clase o construcción en la implementación del sistema.

Los **Diagramas de clases del diseño** permiten visualizar, especificar y documentar modelos estructurales, ya que exponen un conjunto de interfaces, colaboraciones y sus relaciones.

Los **Diagramas de interacción** modelan los aspectos dinámicos de un sistema, debido a que muestran gráficamente cómo los objetos se comunican entre sí a fin de cumplir con los requerimientos. Dentro del Diagramas de interacción se encuentran los diagramas de secuencia y de colaboración. Por lo general los diagramas de colaboración se realizan en el análisis y los de secuencia en el diseño.

Los **Diagramas de secuencia** muestran la ordenación temporal de los mensajes.

Los Diagramas de clases del diseño se modelarán utilizando la extensión de UML para la utilización de estereotipos web, la cual presenta como principales elementos las clases “Server Page”, “Client Page” y “Form”, utilizadas para código servidor, código fuente y formularios respectivamente.

Las páginas servidoras construyen el resultado XHTML que conforma las páginas cliente, a través de la relación <<build>>, estas contienen los formularios que envían sus datos a la página servidora para ser procesados los pedidos usando la relación <<submit>>. Entre las páginas clientes pueden existir vínculos (<<link>>) o redireccionamientos (<<redirect>>). Las páginas servidoras se relacionan con las clases controladoras mediante el <<include>>, permitiéndole a estas últimas efectuar operaciones sobre las entidades, las que permiten a los formularios que se actualicen en caso de ocurrir algún cambio.

La nomenclatura a utilizar por estas clases en los Diagramas de diseño, es la siguiente:

CP_<NombreClaseCliente>, Form_<NombreFormulario>, SP_<NombreClaseServidora> y <NombreClaseControladora>, para las páginas clientes, formularios, clases servidoras ubicadas en la Vista y clases de la Lógica de Negocio respectivamente. Las entidades pueden reconocerse mediante los estereotipos <<Entity Bean>> y <<ORM Persistable>>.

Para cada escenario de dicho diagrama se modelará un Diagrama de interacción, en este caso el Diagrama de secuencia. A continuación se muestran algunos de estos diagramas.

3.2.1- Diagramas de clases del diseño

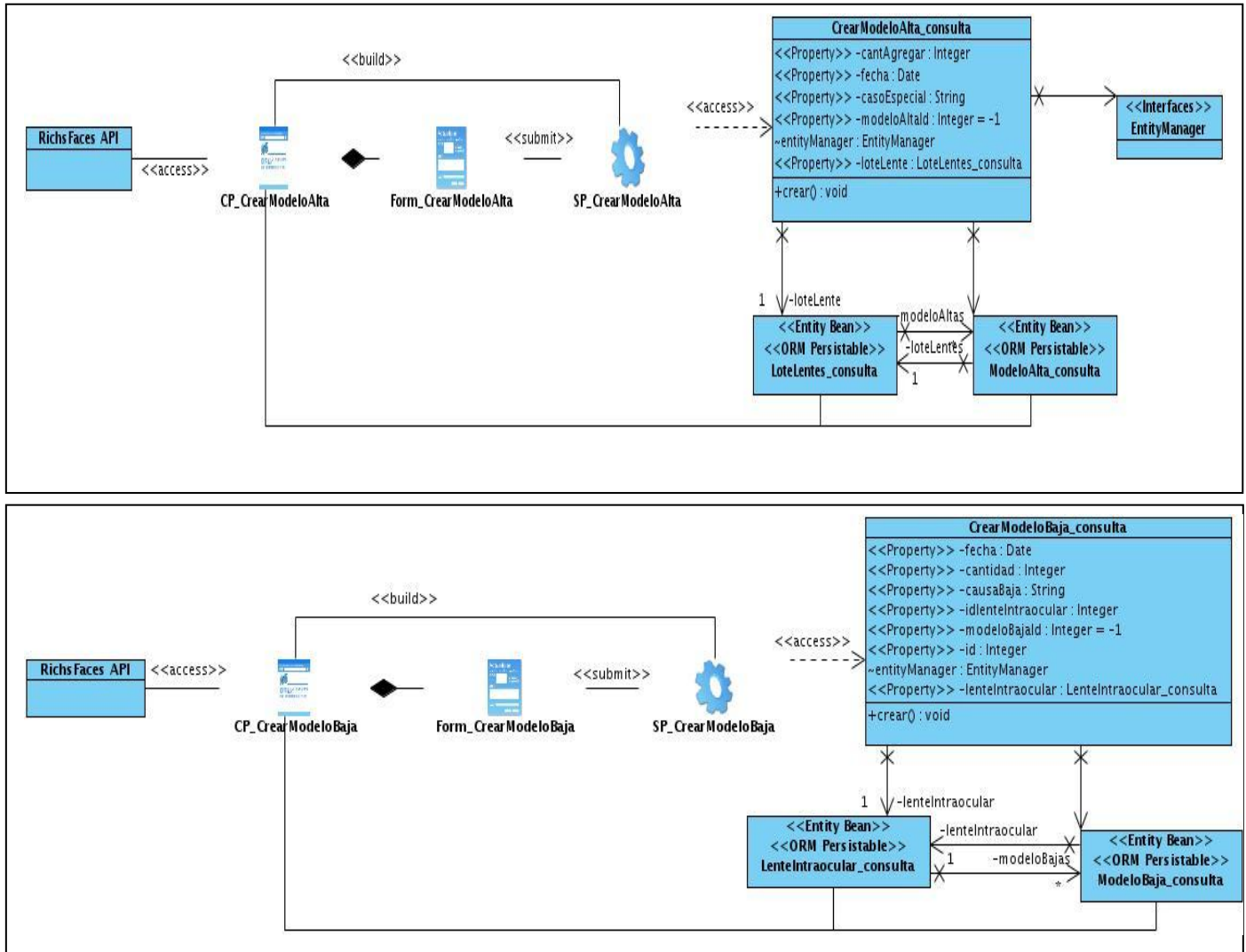


Figura 3.2.2 DCD Dar baja a lentes intraoculares.

3.2.2- Diagramas de secuencia

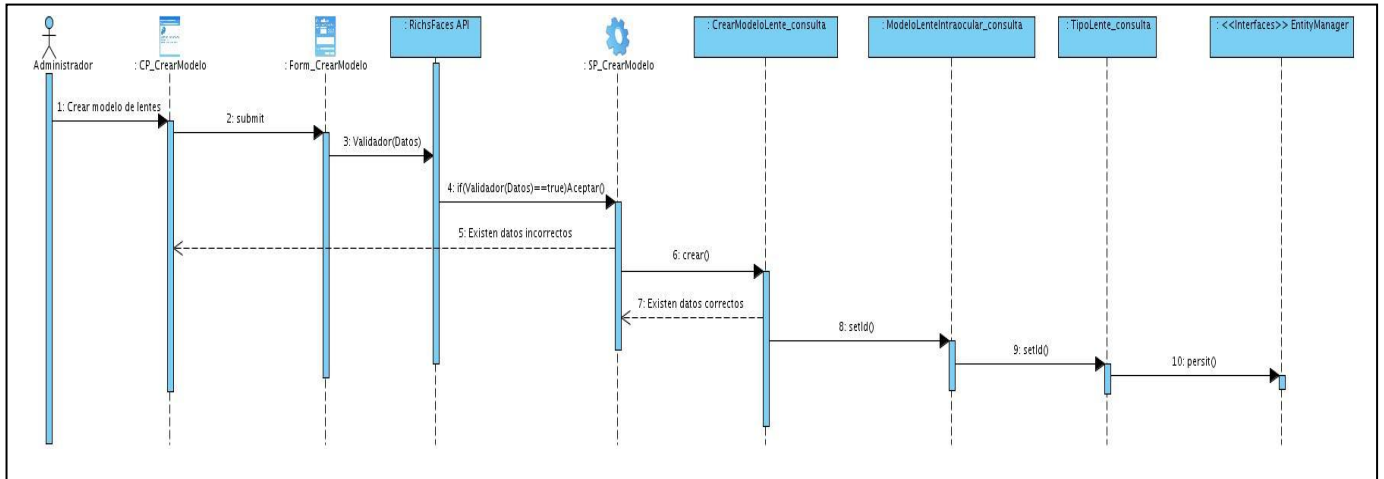


Figura 3.3 DS Dar baja a lentes intraoculares

A continuación se describen algunas de las clases identificadas para la futura implementación del sistema. Así mismo se explican algunas de las responsabilidades que realizarán las páginas servidoras que responden a la Lógica de Negocio, con el objetivo de obtener un mejor entendimiento del funcionamiento del sistema en desarrollo.

Nombre: CrearModeloAlta_consulta	
Tipo de clase: Controladora	
Atributo	Tipo
cantAgrega	Integer
fecha	Date

casoEspecial	String
modeloAltaId	Integer
loteLente	LoteLentes_consulta
entityManager	entityManager
Para cada responsabilidad:	
Nombre:	CrearModeloAlta_consulta
Descripción:	Permite crear el modelo de alta del lente intraocular

Nombre: CrearModeloBaja_consulta	
Tipo de clase: Controladora	
Atributo	Tipo
fecha	Date
cantidad	Integer
causaBaja	String
idlenteIntraocular	Integer
modeloBajaId	Integer
Id	Integer
entityManager	EntityManager

lenteIntraocular	LenteIntraocular_consulta
Para cada responsabilidad:	
Nombre:	CrearModeloBaja_consulta
Descripción:	Permite crear el modelo de baja del lente intraocular

Con la culminación de este capítulo se obtuvo el Modelo de diseño, identificándose y describiéndose las clases necesarias para el correcto funcionamiento del sistema. Además se confeccionaron los diagramas de clases del diseño y los de interacción, proporcionándole al desarrollador una idea más clara de lo que debe implementar.

CAPÍTULO 4: IMPLEMENTACION

En el presente capítulo, basándose en el diseño obtenido, se obtiene el Modelo de datos. Además se define el Modelo de implementación, formado por los diagramas de componentes y el de despliegue. Sobre el tratamiento de excepciones, la seguridad del sistema, las estrategias de codificación, estándares y estilos que se utilizan, también se abordará en este capítulo.

4.1- Modelo de datos

El Modelo de datos describe de forma abstracta las entidades y sus características, además de las relaciones entre estas dentro de la base de datos.

Las entidades son objetos que guardan información necesaria para el sistema. Su símbolo es un rectángulo. Los atributos son características de una entidad, se representan colocando su nombre dentro del rectángulo de la entidad. Los atributos se clasifican en: obligatorios, opcionales, claves foráneas y claves primarias (estas se dividen en simples y compuestas). Gráficamente la clave primaria se representa con un signo de suma y la foránea con un símbolo de número. Las relaciones muestran la asociación entre dos entidades, representadas por una línea que une a las entidades involucradas.

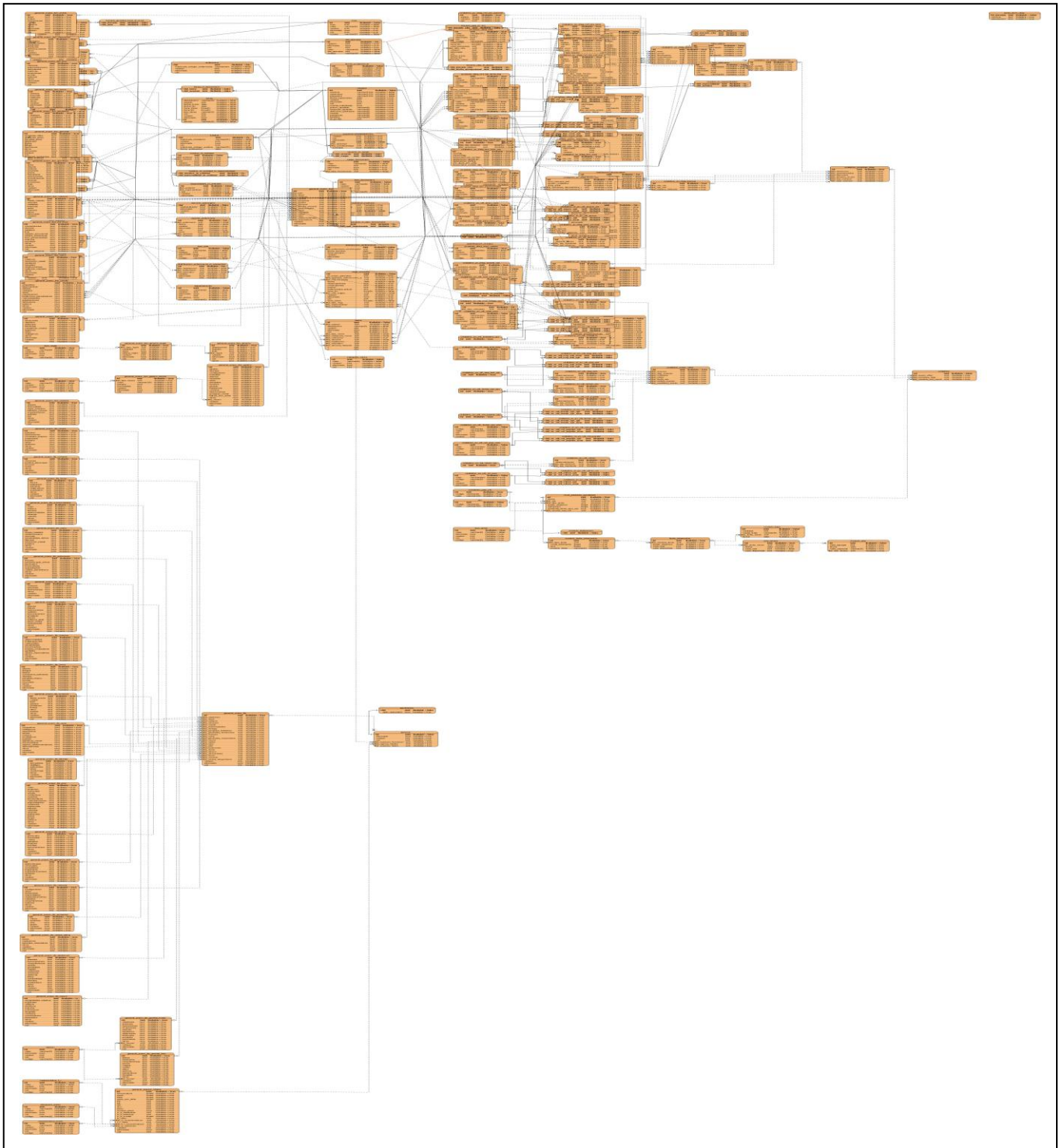


Figura 4.1 Modelo de datos

4.1.2- Descripción de las tablas de la Base de Datos

Los siguientes atributos son comunes para todas las entidades, ya que resultan de importancia para la implementación de algunas funcionalidades.

Atributo	Tipo	Descripción
Id.	int4.	Id necesario en cada entidad para las referencias en las relaciones entre tablas.
versión.	int4.	Indica con qué versión de la entidad se está trabajando. Es usado para garantizar que se está trabajando con la versión de la entidad más actualizada que existe en la base de datos.
eliminado.	bool.	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid.	int4.	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 4.1- Descripción de atributos comunes entre todas las entidades.

tb_modelo_lente_intraocular		
Entidad que recoge la información relacionada con el Modelo de lente intraocular.		
Atributo	Tipo	Descripción
const_fabricante	float4	Constante del fabricante.
acd	float4	ACD.
modelo_lente	varchar(255)	Modelo del lente intraocular.

Tabla 4.2 -Descripción de la tabla tb_modelo_lente_intraocular.

tb_lente_intraocular		
Entidad que recoge la información relacionada con el Lente intraocular.		
Atributo	Tipo	Descripción
poder_dioptrico	float4	Poder dióptrico del lente intraocular.
cant_real	int4	Cantidad real de lentes existentes en el almacén.
cant_disponible	int4	Cantidad disponible de lentes existentes en el almacén.

Tabla 4.3- Descripción de la tabla tb_lente_intraocular

tb_lote_lentes		
Entidad que recoge la información relacionada con el Lote lentes.		
Atributo	Tipo	Descripción
fecha	date	Fecha de entrada del Lote de lentes al almacén.
cant_real	int4	Cantidad real de lentes existentes en el Lote de lentes.
cant_disponible	int4	Cantidad disponible de lentes existentes en el Lote de lentes.
código	varchar	Código del Lote de lentes

Tabla 4.4- Descripción de la tabla tb_lote_lentes.

tb_modelo_alta		
Entidad que recoge la información relacionada con el Modelo de alta.		
Atributo	Tipo	Descripción
fecha	date	Fecha del alta de los lentes al almacén.

cant_agregar	int4	Cantidad de lentes a dar alta al almacén.
caso_especial	varchar	Caso especial por el cual se crea el Modelo de alta.

Tabla 4.5- Descripción de la tabla tb_modelo_alta.

tb_modelo_baja		
Entidad que recoge la información relacionada con el Modelo de baja.		
Atributo	Tipo	Descripción
fecha	date	Fecha de la baja de los lentes en el almacén.
cantidad	int4	Cantidad de lentes a dar baja del almacén.
causa_baja	varchar	Causa por la cual se le da baja a los lentes del almacén

Tabla 4.6- Descripción de la tabla tb_modelo_baja.

tb_aprobacion		
Entidad que recoge la información relacionada con la Consulta de aprobación de cirugía.		
Atributo	Tipo	Descripción
apto_quirurgico	bool	Indica si el paciente está apto quirúrgicamente o no.

Tabla 4.7- Descripción de la tabla tb_aprobacion.

tb_cristalino_pr_diag_ten_ocul_ojo		
Entidad que recoge la información relacionada con las pruebas diagnósticas tensión ocular.		
Atributo	Tipo	Descripción
normal	bool	Indica si la tensión ocular es normal o no.
hipertenso	bool	Indica si el paciente está hipertenso oftalmológicamente o no.

tonometro_no_contac	int4	Indica el resultado del tonómetro no contacto
aplanacion_goldman	int4	Indica el resultado de la aplanación goldman
aplanacion_perkis	int4	Indica el resultado de la aplanación perkis
pio_ajust_por_paq	int4	Indica la presión ocular ajustada por paquimetría
tonometria_contac_din	int4	Indica el resultado de la tonometría por contacto dinámico

Tabla 4.8- Descripción de la tabla tb_cristalino_pr_diag_ten_ocul_ojo.

tb_cristalino_pr_diag_querat_ojo		
Entidad que recoge la información relacionada con las pruebas diagnósticas de queratometría.		
Atributo	Tipo	Descripción
meridiano_fuerte	float4	Indica el resultado del meridiano fuerte
meridiano_debil	float4	Indica el resultado del meridiano débil
eje_fuerte	int4	Indica el resultado del eje fuerte
eje_debil	int4	Indica el resultado del eje débil

Tabla 4.9- Descripción de la tabla tb_cristalino_pr_diag_querat_ojo.

tb_cristalino_pr_diag_paq_corn_ojo		
Entidad que recoge la información relacionada con las pruebas diagnósticas de paquimetría corneal.		
Atributo	Tipo	Descripción
grosor_corneal	int4	Indica el resultado del grosor corneal

Tabla 4.10- Descripción de la tabla tb_cristalino_pr_diag_paq_corn_ojo

4.2- Modelo de Implementación

El Modelo de Implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación, representando cómo estos se organizan en el Modelo de despliegue.

4.2.1- Diagrama de despliegue

Los Diagramas de despliegue muestran la disposición física de los distintos nodos que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos nodos. [29]

Un nodo representa un recurso físico computacional, por lo general con memoria y capacidad de procesamiento.

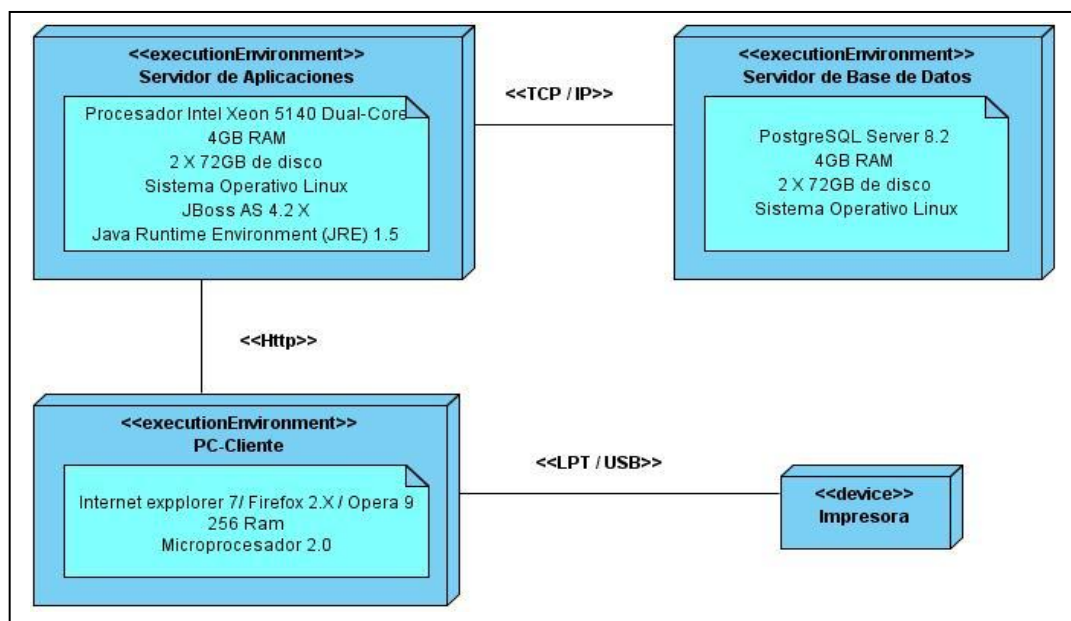


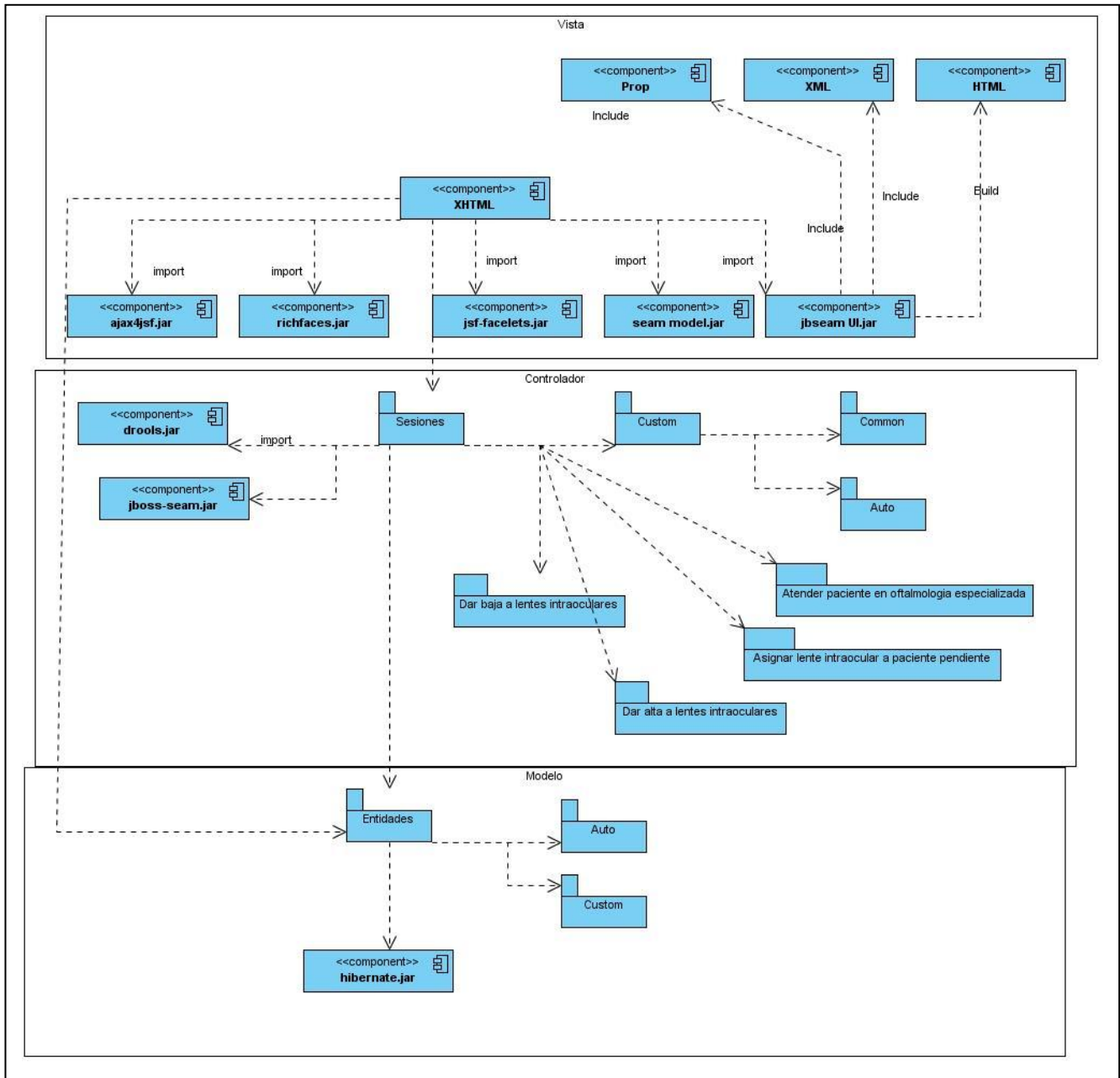
Figura 4.2- Diagrama de despliegue

4.2.2- Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas; pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. [30]

Cada componente puede agruparse en paquetes para obtener una mejor organización y simplificar la implementación.

A continuación se muestra la distribución de los componentes del sistema.



4.3- Tratamiento de excepciones

Una excepción es un error que puede ocurrir debido a una mala entrada por parte del usuario, un mal funcionamiento en el hardware, un argumento inválido para un cálculo matemático, etc., que interrumpe la ejecución normal de un programa. El hecho de realizar el tratamiento de estas excepciones le brinda robustez y flexibilidad al sistema, ya que permite que el programa siga ejecutándose aún teniendo un error, transfiriendo el control de este, al manipulador adecuado para su tratamiento.

En el sistema propuesto el tratamiento de excepciones se realiza en las porciones de código propensas a situaciones inesperadas, especialmente donde se realicen inserciones o modificaciones a la base de datos y en las validaciones de los datos insertados por los usuarios en la interfaz de usuario. El tratamiento de estos errores se realiza usando la siguiente sentencia:

```
try
{
    //declaración de código que causa la excepción
}
catch(NombredeExcepcion obj)
{
    //código para tratar el error
}
```

Así mismo se hace uso de un archivo XML (page.xml) que contine la configuración de los mensajes a mostrar según el tipo de excepción ocurrida, además de la página a la cual se redirecciona en caso de ocurrir un error inesperado. Junto a esto se utiliza el componente Seam FacesMessages, propio de framework Seam, que captura los mensajes de excepción provenientes de las clases controladoras, permitiendo mostrarlos en las interfaces de usuario.

4.4- Seguridad

La seguridad de un sistema informático está dada por la capacidad que este tenga para que solo los usuarios autorizados puedan ver y modificar la información, además que esta esté disponible cuando se

necesite. Esto garantiza la integridad, confidencialidad y disponibilidad de la información. Lo cual alcanza mayor relevancia en sistemas como el propuesto donde se manejan un grupo de datos personales de los pacientes.

Para conseguir esto, el sistema cuenta con un control de acceso a nivel de usuarios y contraseñas, logrando que dichos usuarios accedan solo a las áreas que están establecidas según la función que realizan; pudiendo, solamente el administrador, cambiar las contraseñas y permisos de cada usuario. Además, realiza una traza de todas las acciones efectuadas por cada usuario autenticado en el sistema, mediante el uso de una bitácora de sucesos.

4.5- Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de codificación establecen un conjunto de reglas que los desarrolladores deben seguir para escribir el código fuente de un software. Esto tiene como objetivo que dicho código sea entendible por cualquier desarrollador del grupo de trabajo, no solo por el que lo creó; garantizando un mantenimiento del sistema más rápido y eficiente, ya sea creando nuevas funcionalidades o modificando las ya existentes.

Para el desarrollo del sistema propuesto se utilizan varios estándares de codificación, tales como:

Notación Camell: se utiliza para denotar variables, parámetros y métodos. Si el identificador es una palabra simple se escribe todo con minúscula, en caso de que sea compuesta, se escribe con minúscula la primera letra del identificador y las que vienen a continuación con mayúscula.

A continuación se muestran algunas excepciones para la nomenclatura basadas en el estándar a utilizar:

- **Identación**

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

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

cierres (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.

- **Comentarios, separadores, líneas, espacios en blanco y márgenes.**

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

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

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

Aspectos generales: 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. Con respecto a los espacios en blanco, no se deben usar: después del corchete abierto y antes del cerrado de un arreglo, después del paréntesis abierto y antes del cerrado ni antes de un punto y coma.

- **Variables y Constantes.**

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

Aspectos generales: El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

- **Clases y Objetos**

Apariencia de clases y objetos: 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 Pascal Ejemplo: MiClase(). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, este se escribirá en minúscula.

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

Declaración de parámetro en funciones: 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. (**Ver Tabla 4.1**)

Aspectos generales: El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

- **Bases de datos, tablas, esquemas y campos**

Apariencia de la base de datos: Los nombres de las Bases de Datos deben comenzar con el prefijo bd a continuación underscore y luego el nombre completamente en minúscula, en caso de que sea un nombre compuesto se empleará notación. Los nombres serán cortos y descriptivos.

Apariencia de las vistas: El nombre a emplear para las vistas deben comenzar con el prefijo vt seguido de underscore y el nombre debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

Apariencia de las tablas: El nombre a emplear para las tablas debe comenzar con el prefijo tb seguido de underscore y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscore para separarlo.

Tablas que representen Relaciones: El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tr seguido de underscore y el nombre de la tabla será la concatenación del nombre de las dos tablas que la generaron separados por underscore todo en minúscula.

Tablas que representen nomencladores: El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tn seguido de underscore. El nombre será corto y descriptivo, todo en minúscula.

Apariencia de los procedimientos almacenados: El nombre a emplear para los procedimientos debe comenzar con el prefijo pa seguido de underscore y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscore para separarlo.

Apariencia de los campos: El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

Nombre de los campos: Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo.

Sentencias SQL: Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

Aspectos generales: 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.

- **Controles**

Apariencia de los controles: El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de dato al que se refiere (**ver Tabla 4.2**), en caso de que sea un nombre compuesto se empleará notación CamellCasing**.

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes

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

Tabla 4.1- Tipo de datos

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre
Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

Tabla 4.2- Controles

En este capítulo se obtuvo el Modelo de datos y el Modelo de implementación formado por los diagramas de componentes y el Diagrama de despliegue. Además se estableció el tratamiento de excepciones, la seguridad del sistema y las estrategias de codificación que se emplean para el desarrollo del sistema.

CONCLUSIONES

Con el desarrollo del presente trabajo se arriban a las siguientes conclusiones:

- Mucho de los sistemas informáticos nacionales e internacionales analizados no cumplen con la gestión de todos los procesos que se efectúan en las consultas externas y no incluyen lo referente al servicio de oftalmología. Otros son especializados en oftalmología, pero no contemplan los procesos del resto de las consultas externas ni de las diferentes áreas de las instituciones hospitalarias.
- El estudio de los procesos de negocio permitió definir las funcionalidades del sistema, enfocadas en la atención al paciente y el cálculo y control de lentes intraoculares en la consulta de Microcirugía del Cristalino y las consultas Aprobación de cirugía, enmarcadas en el área Consulta externa del servicio de oftalmología.
- El diseño del sistema se realizó siguiendo la arquitectura propuesta, obteniendo un sistema dotado de flexibilidad y robustez.
- La implementación del sistema permitió la obtención de una solución informática dotada de la seguridad necesaria para garantizar la confidencialidad, integridad y disponibilidad de la información de los pacientes.

RECOMENDACIONES

Se recomienda:

- Incorporar funcionalidades que permitan personalizar la constante del lente intraocular.
- Incorporar funcionalidades que permitan gestionar las afecciones concernientes a cada uno de los exámenes oftalmológicos.
- Incorporar funcionalidades que permitan cancelar la asignación de un lente intraocular a un paciente pendiente.
- Integrar los datos de las pruebas diagnósticas de Queratometría y Biometría desde el equipo IOLMáster.

Referencias Bibliográficas

1. *CNT Sistemas de Información S.A.* (s.f.). Recuperado el Febrero de 2010, de <http://cnt.com.co/PAGINA/index.asp?id=178>
2. *Oftal.com.* (s.f.). Recuperado el Febrero de 2010, de <http://www.oftal.com/servicio.htm>
3. *Oftalmosalus.com.* (s.f.). Recuperado el Febrero de 2010, de <http://www.oftalmosalus.com/folletooftalmosalus.pdf>
4. C. R., César (s.f.). *Crean software para beneficio de salud pública.* Recuperado el Febrero de 2010, de http://www.idict.cu/UserFiles/File/Boletines/Novidades/boletin02_0105/especial020105.html
5. Rodríguez Rabelo, Alexander Ing. I. J. (2009). Sistema para la gestión de la información en las áreas clínico-quirúrgicas de oftalmología. C. Habana, C.Habana, Cuba.
6. Rodríguez Rabelo, Alexander (Julio de 2007). Sistema de Almacenamiento de la Información del Bloque Quirúrgico Oftalmológico del Sistema de Información Hospitalaria. Ciudad de la Habana, Cuba.
7. *Desarrolloweb.com.* (s.f.). Recuperado el Febrero de 2010, de <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
8. *Reynoso, C. (2004). Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* Recuperado el Febrero de 2010, de http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10
9. Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture.* Gran Bretaña.
10. *Guía de desarrollo del RichFaces.* (s.f.). Recuperado el Febrero de 2010, de http://jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html_single/index.html
11. *Asynchronous Javascript And XML (AJAX).* (s.f.). Recuperado el Febrero de 2010, de <http://tutoriales.maborak.com/ajax>
12. *JBoss Community.* (s.f.). Recuperado el Febrero de 2010, de <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/Introduction.html>
13. Pérez, J. E. (s.f.). *Librosweb.es.* Recuperado el Febrero de 2010, de http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html
14. *ignside.net.* (s.f.). Recuperado el Febrero de 2010, de <http://www.ignside.net/man/css/index.php>

15. *Web Application - Plataforma J2EE*. (s.f.). Recuperado el Febrero de 2010, de JBoss Seam Framework: <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>
16. Devesa Navarro, Arianna., & Muñoz Pérez, R. (Junio de 2009). Módulo Hospitalización del Sistema de Información Hospitalaria alas HIS. . Ciudad de la Habana, Cuba.
17. *Hibernate Tools for Eclipse and Ant*. (s.f.). Recuperado el Febrero de 2010, de Hibernate.org: <http://www.hibernate.org/255.html>
18. *Java Persistence API FAQ. Sun Microsystems*. (s.f.). Recuperado el Febrero de 2010, de <http://java.sun.com/javaee/overview/faq/persistence.jsp>
19. García de Jalón, J., Rodríguez Iñigo, M. J., Alfonso Brazález, A. I., Larzabal, A., Calleja, J., & García, J. (Enero de 2000). *Aprenda Java como si estuviera en primero*. Escuela Superior de Ingenieros Industriales. Universidad de Navarra, España.
20. Franky, C. (s.f.). *Java EE 5 (sucesor de J2EE)*. Recuperado el Febrero de 2010, de http://www.acis.org.co/fileadmin/Conferencias/ConfConsueloFranky_Abr19.pdf
21. Lucifer, P. (s.f.). *Java Runtime Environment - JRE*. Recuperado el Febrero de 2010, de <http://www.elleonplateadodeojosrojos.es/blog/java-runtime-environment-jre/>
22. *JBoss Server Manager Reference Guide*. (s.f.). Recuperado el Febrero de 2010, de <http://www.jboss.org/jbossas/docs/>
23. *PostgreSQL About*. (s.f.). Recuperado el Febrero de 2010, de <http://www.postgresql.org/about>
24. *PostgreSQL Awards*. (s.f.). Recuperado el Febrero de 2010, de <http://www.postgresql.org/about/awards>
25. Axentia. (s.f.). *Introducción al CMMI*. Recuperado el Febrero de 2010, de [sergiovillagra.com: http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf](http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf)
26. *Teoría de Sistemas*. (s.f.). Recuperado el Febrero de 2010, de <http://www.kyberne.com/Documentos%20de%20Partners/13%20-%20TGS%20-%20ModProcesos-BPMN.ppt>.
27. *Eclipse (Software)*. (s.f.). Recuperado el Febrero de 2010, de <http://plataformaclipse.com/>
28. Saavedra, Jorge. PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidades). Parte 2. [En línea] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/> .

29. Visconti, Marcello y Astudillo, Hernán. Fundamentos de la Ingeniería de Software. *Departamento de Informática de la Universidad Técnica Federico Santa María*. [En línea]
30. (s.f.). Recuperado el Abril de 2010, de Dpto de Sistemas Informáticos, Universidad de Castilla-La Mancha: <http://www.info-ab.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>

Bibliografía

- *Asociación de Usuarios de GNU/Linux de Castilla y León*. (s.f.). Recuperado el Febrero de 2010, de <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>
- *Care2x*. (s.f.). Recuperado el Febrero de 2010, de <http://www.care2x.org/>
- Chacón García, Y., & García Bonelly, O. (Junio de 2009). Módulo Epidemiología del Sistema de Información Hospitalaria alas HIS. Ciudad de la Habana, Cuba.
- CINTAO, A. M. (s.f.). *www.trabajadores.co.cu*. Obtenido de <http://edicionesanteriores.trabajadores.cu/Cumbre%20Mundial%20de%20la%20Sociedad%20de%20la%20Informacion/informatizacion.htm>
- Francisco J. Fernández Puerto, F. G. (2003). *Portal de la Facultad de Medicina de México*. Recuperado el 5 de Noviembre de 2009
- Junta de Andalucía. (s.f.). *Madeja*. Recuperado el 22 de 10 de 2009, de <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Hibernate#ventajas>
- Alemán Antelo, Lorena, Gacia Orduñez Juan Manuel. (Junio de 2009). Módulo Emergencia del Sistema de Información Hospitalaria alas HIS. C. Habana, C. Habana, Cuba.
- *Microsoft patterns and practices. Model-View-Controller. MSDN*. (s.f.). Recuperado el Febrero de 2010, de <http://msdn.microsoft.com/en-us/library/ms978748.aspx>
- Oktaba, H. (s.f.). *Intranet de la facultad de Ciencia e Ingeniería de la Computación de la Universidad Nacional de México*. Recuperado el Febrero de 2010, de <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>
- Pérez, J. E. (s.f.). *Librosweb.es*. Recuperado el Febrero de 2010, de http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html
- *RealClinic*. (s.f.). Recuperado el Febrero de 2010, de http://www.tdsa.com.br/produtos/realclinic/content.php?vai=introducao&ref=produto_introducao
- *Revista cubana de Informática Médica*. (s.f.). Recuperado el Febrero de 2010, de http://www.cecarn.sld.cu/pages/rcim/revista_6/articulo_hm/sistema.htm
- *softel.cu*. (s.f.). Recuperado el Febrero de 2010, de <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>

- *Teoría de Sistemas.* (s.f.). Recuperado el Febrero de 2010, de <http://www.kyberne.com/Documentos%20de%20Partners/13%20-%20TGS%20-%20ModProcesos-BPMN.ppt>.