

Universidad de las Ciencias Informáticas
Facultad 7



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

Desarrollo del quirófano especializado de Oftalmología del módulo
Bloque Quirúrgico del Sistema de Información Hospitalaria alas HIS

Autores: Lenia Palau Yú
William Frances García

Tutores: Ing. Alexander Rodríguez Rabelo
Ing. Yosmel Martínez Díaz

Ciudad de la Habana, Julio 2010
“Año 52 de la Revolución”

DATOS DE CONTACTO

Ing. Alexander Rodríguez Rabelo: Graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Especialista del Departamento de Sistema de Gestión Hospitalaria y profesor Instructor vinculado a la Facultad 7. Ha impartido las asignaturas Matemática II y Matemática IV.

Ha tutorado hasta el momento tres trabajos de diploma.

Correo electrónico: arodriguezra@uci.cu

Ing. Yosmel Martínez Díaz: Instructor recién graduado en el año 2009 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor vinculado a la Facultad 7 y miembro del Departamento de Sistema de Gestión Hospitalaria. Ha impartido la asignatura de Práctica Profesional y diferentes cursos optativos relacionados con la Facultad.

Correo electrónico: ymdiaz@uci.cu

RESUMEN

El presente trabajo tiene como objetivo desarrollar los procesos de gestión de la información referentes al área de Quirófano especializado en Oftalmología: Realizar solicitud de intervención quirúrgica oftalmológica, Realizar consulta pre-anestésica oftalmológica y Realizar intervención quirúrgica oftalmológica.

Su desarrollo se basa en tecnologías libres, multiplataformas y utiliza el patrón arquitectónico Modelo Vista Controlador. Emplea Java como lenguaje de programación, Eclipse como Entorno de Desarrollo, PostgreSQL como Sistema de Gestión de Bases de Datos y como servidor de aplicaciones el JBoss Server. El proceso de desarrollo estuvo guiado por el Proceso Unificado de Desarrollo en conjunto con el Modelo de Capacidad y Madurez.

La aplicación trae como beneficios lograr que el proceso de gestión de información sea confiable, seguro y centralizado. Permite la generación de reportes médicos de importancia vital para el área de Quirófano especializado en Oftalmología, elevando la calidad y excelencia de los servicios prestados al paciente por dicha área.

Palabras claves: Solicitud de intervención quirúrgica oftalmológica, consulta pre-anestésica oftalmológica, intervención quirúrgica oftalmológica, Quirófano especializado en Oftalmología.

TABLA DE CONTENIDOS

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 5 |
| 1.1 Conceptos básicos relacionados con el dominio del problema | 5 |
| 1.2 Sistemas de Información Hospitalaria (HIS) | 6 |
| 1.3 Antecedentes..... | 7 |
| 1.4 Tendencias y tecnologías actuales a utilizar | 10 |
| 1.5 Tecnologías y estándares utilizados en el proceso de desarrollo | 13 |
| 1.6 Herramientas..... | 21 |
| CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA..... | 23 |
| 2.1 Flujo actual de los procesos involucrados en el campo de acción | 23 |
| 2.2 Objeto de automatización | 24 |
| 2.3 Modelado del Negocio | 24 |
| 2.4 Diagramas de Procesos del Negocio..... | 26 |
| 2.5 Propuesta del Sistema..... | 28 |
| 2.6 Modelo de casos de uso del sistema..... | 35 |
| CAPÍTULO 3: DISEÑO DEL SISTEMA | 44 |
| 3.1 Descripción de la arquitectura | 44 |
| 3.2 Modelo de diseño | 44 |
| CAPÍTULO 4: IMPLEMENTACIÓN | 55 |
| 4.1 Modelo de datos..... | 55 |
| 4.2 Modelo de Implementación..... | 62 |
| 4.3 Tratamiento de excepciones | 65 |
| 4.4 Seguridad..... | 66 |
| 4.5 Estrategias de codificación. Estándares y estilos a utilizar | 67 |
| CONCLUSIONES | 70 |
| RECOMENDACIONES | 71 |
| REFERENCIAS BIBLIOGRÁFICAS | 72 |
| BIBLIOGRAFÍA..... | 74 |
| GLOSARIO DE TÉRMINOS..... | 76 |

INTRODUCCIÓN

El ser humano para su desarrollo necesita constantemente de la información, por ello ha ideado diversos medios a través del tiempo para comunicarla. El ritmo acelerado de cambios, evolución e innovación de las Tecnologías de la Información y las Comunicaciones han proporcionado herramientas para el óptimo manejo de los recursos y el acceso a la misma. Estos adelantos tecnológicos provocan cambios que inciden en la vida cotidiana del hombre y elevan considerablemente su nivel de desarrollo.

Durante la segunda mitad del siglo XX, con la llegada de la Revolución Científico-Técnica, la generalización de la computación y la gestión de la información se han apoderado de todas las esferas de la sociedad. La salud como una de las más importantes, ha tomado la vanguardia en este sentido.

El uso de la informática en la medicina se ha convertido en una herramienta esencial de trabajo, en un entorno caracterizado por el aumento del número y la complejidad de las especialidades; estando en presencia de una sociedad donde se manejan amplios volúmenes de datos que afectan a la práctica clínica, debido a la gran cantidad de operaciones que se realizan durante la interacción médico-paciente.

La informática médica ha introducido una nueva dimensión en el pensamiento humano, haciendo realidad los sueños de varios años de trabajo y constante renovación y perfeccionamiento del sistema de salud, obteniendo grandes beneficios como: la posibilidad de consultar grandes volúmenes de bibliografía médica, mediante la aparición de bases de datos bibliográficas para ayudar a la toma de decisiones, el desarrollo de redes de información para reducir costes y elevar la calidad de atención del paciente. (1)

El uso de las tecnologías en función de la actividad médica ha generado una serie de cambios progresivos que dan lugar a la aparición de estrategias administrativas y docentes que promueven la integración de todas las áreas de salud al proceso de informatización. (2) Es así que han surgido un conjunto de sistemas, capaces de ayudar a los profesionales de la salud en sus actividades diarias, desde la gestión de la información en los grandes hospitales, hasta los consultorios, clínicas, la docencia médica, la generación de estadísticas, etc.

En los años 70 surgen los primeros Sistemas de Información Hospitalaria HIS (Hospital Information System), estos persiguen elevar constantemente la calidad de atención al paciente mediante la mejoría de

los servicios que se le brindan. Además, permiten obtener mejores resultados en la investigación, la docencia y la administración hospitalaria.

Actualmente es interés de la administración central del estado cubano trabajar en el desarrollo de un grupo de aplicaciones básicas para la informatización del sector de la salud. Por lo que se le da la tarea a la Universidad de Las Ciencias informáticas de realizar un conjunto de estas, hoy en día en el departamento de Sistemas de Gestión Hospitalaria de la Facultad 7 se desarrolla el Sistema de Información Hospitalaria alas HIS, el cual está compuesto por varios módulos que representan las diversas áreas de un hospital.

Uno de estos módulos es el de Bloque Quirúrgico, donde se administran los recursos humanos y materiales que están disponibles para la realización de las intervenciones quirúrgicas a los pacientes; permitiendo planificar las cirugías de los servicios: Obstetricia, Ginecología, Nefrología, Traumatología, Neurocirugía, Cirugía Cardiovascular, Urología, Otorrinolaringología, Gastroenterología, Pediatría quirúrgica, Hemodinamia, Cirugía de la mano, Oncología, Unidad de Cirugía Bariátrica y Cirugía General; brindando como resultado final el plan quirúrgico. Se gestiona además el acto quirúrgico y la consulta pre-anestésica.

Sin embargo, dicho módulo, no abarca con total exactitud lo concerniente a la especialidad de Oftalmología, ya que no es capaz de gestionar el quirófano especializado de oftalmología, la consulta pre-anestésica oftalmológica y la planificación quirúrgica de este servicio. Esto trae como resultado que la información oftalmológica no se maneje correctamente, ya que la misma puede extraviarse, dañarse, y afectar la generación de estadísticas y reportes médicos, llegando incluso a cometer omisiones de datos importantes en la Historia Clínica del paciente.

En consecuencia con la situación analizada anteriormente se plantea como **problema a resolver**: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Quirófano especializado en Oftalmología de las instituciones hospitalarias?

El **Objeto de Estudio** de la investigación es: Proceso de gestión de información en las instituciones hospitalarias.

El **campo de acción** está comprendido en: Proceso de gestión de la información en el área de Bloque Quirúrgico especializado en Oftalmología de las instituciones hospitalarias.

Para resolver el problema identificado se propone el siguiente **objetivo general**: Desarrollar el quirófano especializado en Oftalmología del módulo Bloque Quirúrgico del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la información en esta área.

Tareas a desarrollar para dar cumplimiento al objetivo trazado:

- Identificar los procesos del negocio del área oftalmológica.
- Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria.
- Asimilar la arquitectura definida por el Departamento de Sistema de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
- Obtener los artefactos correspondientes a los flujos de trabajo “Modelado de Negocio”, “Gestión de Requerimientos”, “Análisis y Diseño” e “Implementación”.
- Implementar, aplicando las pautas de diseño los procesos de negocio: Realizar Solicitud de intervención quirúrgica, Realizar consulta Pre-anestésica y Realizar intervención quirúrgica, relacionados con la elaboración del plan quirúrgico y administración del quirófano especializado en Oftalmología del módulo de Bloque Quirúrgico.

Con el desarrollo de los procesos del área de Quirófano especializado en Oftalmología del Sistema de Información Hospitalaria alas HIS se obtendrán beneficios como:

- Centrar el proceso de atención al paciente en la historia clínica electrónica única, lo que permitirá a los Especialistas la obtención de información sobre la salud del paciente y su seguimiento.
- Lograr que el proceso de gestión de información en el área de Quirófano especializado en Oftalmología sea confiable, seguro y centralizado.
- Elevar la calidad y excelencia de los servicios quirúrgicos de Oftalmología prestados al paciente mediante una adecuada planificación quirúrgica centrada en los recursos humanos y materiales.
- Generar los reportes médicos de importancia vital para el funcionamiento del área de Quirófano especializado en Oftalmología, lo que permitirá obtener estadísticas reales y actualizadas sobre la situación de salud de los pacientes.

El documento está estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica: Estudio preliminar de los HIS. Tecnologías, metodologías y herramientas de desarrollo a utilizar.

Capítulo 2: Características del sistema: Descripción de la solución propuesta a la problemática planteada.

Capítulo 3: Diseño del sistema: Concepción de la solución propuesta del sistema.

Capítulo 4: Implementación: Desarrollo de la solución propuesta del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace referencia a los conceptos básicos relacionados con el Proceso de gestión de la información en el área de Bloque Quirúrgico especializado en Oftalmología. El estudio de los sistemas de información hospitalaria en el ámbito nacional e internacional. Además se describen las tecnologías, herramientas y metodologías a utilizar en el desarrollo del software a partir de la arquitectura definida por el departamento de Sistemas de Gestión Hospitalaria.

1.1 Conceptos básicos relacionados con el dominio del problema

Para brindar un mejor entendimiento acerca del negocio del problema a resolver, se presentan a continuación conceptos asociados con el proceso de gestión de la información en el área de Bloque Quirúrgico especializado en Oftalmología:

Oftalmología: Especialidad dedicada al estudio de las enfermedades de los ojos y se encarga de su tratamiento médico-quirúrgico.

Especialista: Médico especializado en Oftalmología.

Anestesiólogo: Especialista en Anestesiología que se encarga de suministrar la anestesia al paciente para que en el transcurso de la operación no sienta dolor y monitorea su estado de salud antes, durante y después de la intervención quirúrgica.

Solicitud de intervención quirúrgica oftalmológica: Refleja los datos referentes a la petición de la realización de la operación al paciente.

Consulta pre-anestésica: En esta consulta el anestesiólogo determina si el paciente es apto quirúrgico y le elabora la hoja de anestesia.

Hoja de anestesia oftalmológica: Documento en el cual se recogen los datos de la consulta pre-anestésica oftalmológica y del acto quirúrgico de Oftalmología que contempla todo lo relacionado al paciente y la anestesia que se utilizó durante la operación.

Quirófano: Local acondicionado para realizar operaciones quirúrgicas de manera que no exista riesgo de contaminación para el paciente o el médico.

Intervención quirúrgica: Procedimiento realizado por un cirujano con la ayuda del anestesiólogo, enfermeros y otros asistentes en un quirófano con el objetivo de sanar alguna patología al paciente mediante la operación. Se utiliza para su realización, la anestesia, instrumentación, medicamentos, insumos y técnicas requeridas en dependencia del tipo de cirugía. La intervención se divide en tres períodos fundamentales: Pre-Operatorio, Trans-Operatorio y Post-Operatorio.

Nota operatoria oftalmológica: Recoge los datos obtenidos durante el proceso de intervención quirúrgica oftalmológica del paciente, registrando además cualquier complicación ocurrida durante la operación.

1.2 Sistemas de Información Hospitalaria (HIS)

Los sistemas de información hospitalaria se encargan de procesar, recolectar, recuperar, almacenar y comunicar un gran volumen de información en las diversas áreas de un hospital mediante el uso de las nuevas tecnologías. Estos le brindan varias ventajas a las instituciones hospitalarias como: adquirir la información en menor tiempo, respaldar la toma de decisiones de los médicos, generar múltiples reportes estadísticos e integrar todas las áreas hospitalarias. Presentan el objetivo fundamental de satisfacer las necesidades de la docencia, la investigación, las actividades clínicas y la administración hospitalaria. Y así elevar la calidad de atención al paciente.

1.3 Antecedentes

Los HIS contienen e integran módulos que representan las diversas áreas de un hospital, una de estas es “Bloque Quirúrgico” que entre otros servicios brinda los relacionados con la especialidad de Oftalmología en ocasiones. En algunos casos se proponen soluciones que sólo brindan las funcionalidades del Bloque Quirúrgico especializado en Oftalmología como un sistema independiente.

La siguiente tabla representa algunos sistemas informáticos a nivel nacional e internacional que gestionan la información referente a los procesos del área quirúrgica especializada en Oftalmología o del bloque quirúrgico de las instituciones hospitalarias.

| Nombre | País | Multiplataforma | Tipo de Licencia | | Tipo de Aplicación | |
|-------------------|-----------|-----------------|------------------|-------------|--------------------|------------|
| | | | Libre | Propietario | Web | Escritorio |
| BQO | Cuba | -- | -- | X | X | -- |
| OftalClinic | España | -- | -- | X | -- | X |
| SIGHO | México | -- | -- | X | -- | X |
| Care2X | Alemania | X | X | -- | X | -- |
| Sushrut | India | X | -- | X | -- | X |
| Optim OTM | Francia | -- | -- | X | X | -- |
| Soarian™ MedSuite | Argentina | -- | -- | X | X | -- |

Tabla 1.1 Sistemas de Información Hospitalaria

A continuación se presentan las características de algunos de los HIS mencionados en la tabla anterior, considerados significativos para el objeto de estudio de la investigación.

BQO (Bloque Quirúrgico Oftalmológico): Es una aplicación realizada por el Área Temática Gestión Hospitalaria. El sistema permitirá a los usuarios controlar toda la información referente a un paciente que será intervenido quirúrgicamente por causas oftalmológicas. Presenta una arquitectura cliente servidor. (3)

Entre sus principales funciones se encuentran:

- Visualizar los datos registrados en la historia clínica siempre que se atienda a un paciente.

- Gestionar Hoja de consulta de Anestesia.
- Realizar búsquedas de paciente, consultas, informes operatorios y anuncios.
- Registrar los datos necesarios del paciente durante la intervención quirúrgica.
- Crear y modificar el Anuncio operatorio.
- Crear y modificar el Informe operatorio.
- Realizar la Planificación quirúrgica por los diferentes criterios.
- Mostrar a los especialistas su planificación personal. (4)

OftalClinic: Es un sistema de gestión integral de clínicas oftalmológicas desarrollado por la Empresa Española DISA S.L. que abarca todas las áreas de planificación, administración, gestión médica y control de la clínica oftalmológica destinadas a optimizar los recursos humanos, permitiendo la conexión de equipos de diagnóstico computarizados. Emplea tecnología cliente servidor sobre base de datos Microsoft SQL-Server de alto rendimiento, gracias a esto se pueden interconectar varias clínicas centralizando así toda la información. El sistema está diseñado como una aplicación de escritorio.

Contiene módulos de cirugía oftalmológica general donde se gestionan los siguientes procesos

- Cirugía de catarata con facoemulsificación.
- Cirugía de pterigium con injertos.
- Cirugía láser para retinopatía diabética.
- Trabeculectomía para glaucoma.
- Cirugía refractiva.

SIGHO (Sistema de Información para la gerencia Hospitalaria): El sistema es desarrollado en México e inicia a operar el 18 de noviembre de 2004. Permite realizar registros individuales alrededor de la historia clínica electrónica, en cada uno de los módulos relacionados con la atención al paciente. Se compone por 18 módulos que representan áreas de un hospital, algunas de ellas son: Admisión Hospitalaria, Medicina Interna, Terapia Intensiva, Consulta Externa y Cirugía. Prescinde del servicio de Oftalmología en el área de Cirugía. En el área de Cirugía gestiona procesos como:

- Crear Solicitud de Quirófanos.

- Gestionar Programación de procedimientos quirúrgicos.
- Generar Reporte de procedimientos quirúrgicos realizados.
- Gestionar Nota Pre-Operatoria.
- Gestionar Nota Post-Operatoria.
- Gestionar Nota Pre-Anestésica.
- Gestionar Nota Post-Anestésica.

Es un software basado en la Norma Oficial Mexicana *NOM-168-SSA1-1998* referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital y se apoya de estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC. (5)

Sushrut: C-CAD (Centre for development of advanced computing) Sushrut, es un sistema de información hospitalaria (HIS) y se ha desarrollado con el objetivo de racionalizar el flujo de tratamiento de un paciente en el hospital. Presenta una arquitectura cliente servidor de base de datos. El módulo de Operación contiene información sobre la disponibilidad de todos los quirófanos, equipos y herramientas. La planificación de operaciones es la función principal de este módulo a través de la cual se puede aprobar, cancelar o reprogramar la operación. Las actividades programadas han de ser validadas por los cirujanos principales. Permite la entrada y validación de un registro detallado de la operación, la anestesia y el mantenimiento del post-operatorio. (6)

Soarian™ MedSuite: Organiza el trabajo en 4 direcciones, ellas son la gestión administrativa del paciente, la gestión clínica del paciente, la gestión de medicamentos y materiales, y el laboratorio. Presenta una arquitectura cliente servidor.

El módulo Quirófano es usado para procesar y administrar la información y servicios de los quirófanos de un hospital. A través de este módulo, el sistema permite consultar y procesar todos los servicios de cirugía requeridos. Los servicios de cirugía pueden estar vinculados a agendas que se configuran independientemente y se asocian a uno o varios quirófanos. Este módulo permite solicitar los instrumentos al sector de esterilización así como a llevar un control de stock de instrumentos del quirófano. Por otra

parte es posible gestionar el stock de medicamentos y materiales del quirófano interactuando con la Farmacia Central, el depósito y los demás substocks del hospital. El módulo permite registrar en forma automática o manual los elementos utilizados en una práctica quirúrgica sobre un paciente, resolviendo el consumo de stock a ser facturado. Permite también capturar detalles pre y post operatorios. (7)

Luego de analizar las principales características de los HIS que forman parte de la investigación, se concluye que estos en su mayoría cubren las funcionalidades de varias áreas del hospital, sin contar con la totalidad de las especificidades del quirófano especializado en Oftalmología.

Sólo BQO y OftalClinic realizan ciertos procesos relacionados a esta área, en el caso específico de BQO gestiona la planificación quirúrgica y el quirófano en las especialidades de (Refractiva, Cristalino, Oculoplastia y Pterigium) y en el caso de OftalClinic gestiona la información referente a (Pterigium, Refractiva, Cristalino y Glaucoma).

Ambos sistemas cuentan con las desventajas de que no gestionan el resto de las áreas intrahospitalarias, no son multiplataformas y están desarrollados sobre software propietario, lo que limita al país de su adquisición y de futuras mejoras a las funcionalidades que brindan, debido a las restricciones y el alto precio de sus licencias.

Por lo tanto se propone elaborar un sistema basado en open source, multiplataforma y web; que sea capaz de gestionar los procesos referentes al área de quirófano especializado en Oftalmología. A continuación se mencionan las tecnologías y herramientas seleccionadas para su desarrollo.

1.4 Tendencias y tecnologías actuales a utilizar

En este epígrafe se exponen los elementos asociados al ambiente de desarrollo del sistema y para ello en primer lugar, es preciso definir qué es la Arquitectura de Software, que según Roger Pressman representa, "... la descripción de los subsistemas o componentes de un sistema informático y las relaciones entre ellos..." (8)

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (9)

1.4.1 Sistema Modelo Cliente Servidor

El esquema cliente servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí." (10) Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el proceso cliente sólo se ocupa de la interacción con el usuario. Éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones. Comenzó a ser aceptado a finales de los años 80.

Brinda como ventajas:

- La posibilidad de utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Facilita la integración entre sistemas diferentes, permitiéndoles compartir información.
- El mantenimiento y el desarrollo de aplicaciones es rápido ya que se pueden emplear las herramientas existentes.
- No es siempre necesario transmitir información gráfica por la red, pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

1.4.2 Patrones de Diseño

Los patrones de Diseño son una forma estandarizada de resolver problemas específicos de programación para mejorar la calidad y diseño del sistema. Ayudan a reutilizar las piezas de software, a un sistema en particular o a varios sistemas. Están estrechamente ligados a la Programación Orientada a Objetos (POO), ya que se valen de varios conceptos de la misma para definir las estrategias para su

implementación. Normalmente con la POO se piensa que se tiene todo, pero como no se conoce la aplicación de los patrones de Diseño entonces en ocasiones se reinventa la rueda. Estos patrones se categorizan en tres tipos: Patrones de Creación, Patrones de Estructura y Patrones de Comportamiento.

1.4.3 Patrones de Arquitectura

En el desarrollo de aplicaciones informáticas se utilizan los llamados patrones de arquitectura o estilos arquitectónicos que definen las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura fundamental de los sistemas. En una forma más específica, un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas. (11)

1.4.4 Modelo Vista Controlador (MVC)

Sus características principales son que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Usado frecuentemente en aplicaciones web donde se utilicen diferentes interfaces de usuario. Las tres partes de este patrón se definen:

Modelo: Representa los datos del programa, manejándolos y controlando todas sus transformaciones. Lleva un registro de las vistas y controladores del sistema. Es independiente de cualquier representación de salida y/o comportamiento de entrada.

Vista: Maneja la presentación visual de los datos representados por el modelo. Muestra la información al usuario a través de una Interfaz de Usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador: Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Une la vista y el modelo. Interactúa con el modelo a través de una referencia al propio modelo.

La arquitectura Modelo Vista Controlador permite implementar cada uno de sus componentes por separado, donde tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. La conexión entre el modelo y sus vistas es dinámica, se produce en tiempo de ejecución, no en

tiempo de compilación. Simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

1.5 Tecnologías y estándares utilizados en el proceso de desarrollo.

A continuación se describen las tecnologías que fueron utilizadas para el proceso de desarrollo del software, según la capa que le corresponde.

1.5.1 Vista

JSF

JSF (JavaServer Faces) es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador). Permite crear interfaces de usuario (UI) para aplicaciones web, mediante componentes reutilizables.

“JSF trata la vista UI de una forma algo diferente a lo que se acostumbra en aplicaciones web. Sería más similar al estilo de Swing, Visual Basic o Delphi, donde la programación de la interfaz se hace a través de componentes y basada en eventos. Es muy flexible ya que posibilita crear componentes o “render” para pintar los mismos según convenga.” (12)

RichFaces

JBoss 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. Provee facilidades de validación y conversión de los datos proporcionados por el usuario, administración avanzada de recursos como imágenes, código JavaScript y Hojas de Estilo en Cascada (CSS). Se integra completamente dentro del ciclo de vida JSF. Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente

configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora grandemente la experiencia de usuario. (13) (14)

Ajax4jsf

Es una librería open source 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 automáticas al servidor y controlar los eventos de usuario. (15)

Seam UI

Conjunto de controles JSF modificados que se integran con JBoss Seam. Están dirigidos a complementar los controles JSF incorporados, y los controles de otras bibliotecas externas.

Facelets

Facelets es un framework ligero que permite el uso de plantillas en aplicaciones JSF. Sus principales ventajas son:

- Construcción de interfaces basadas en plantillas.
- Rápida creación de componentes por composición.
- Fácil creación de funciones y librerías de componentes. (16)

ExtendedEL

Lenguaje de expresión que provee JSF el cual es utilizado en las páginas de las aplicaciones web con el fin de acceder a las entidades de java. (17)

XHTML

Lenguaje de Marcado de Hipertexto Extensible (XHTML). Familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basadas en

XML. Los documentos XHTML son conformes a XML. Como tales, son fácilmente visualizados, editados y validados con herramientas XML estándar. Estos documentos XHTML pueden escribirse para que funcionen igual o mejor que lo hacían antes tanto en las aplicaciones de usuario conformes a HTML 4.0 como en las nuevas aplicaciones conformes a XHTML 1.0. Además pueden usar aplicaciones (scripts y applets) que se basen ya sea en el Modelo de Objeto del Documento de HTML o XML [DOM]. (18)

1.5.2 Controlador

JBoss Seam

JBoss Seam 2.0 es un framework para el desarrollo de aplicaciones web en Java. Revitaliza el estándar Java EE poniéndole fin a sus divergencias y unificando sus componentes, haciéndolos más accesibles y funcionales. Integra fácilmente tecnologías estándares como Java Server Faces (JSF), modelo de componentes para la capa de presentación; Enterprise JavaBeans (EJB3), modelo de componentes para la lógica de negocio y persistencia del lado del servidor; Java Persistence API (JPA), y de Business Process Management (BPM). Integra además librerías de controles de código abierto basadas en JSF como RichFaces e ICEFaces. (19) (20)

JBoss Server

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada Java EE, soporta todas las funcionalidades de Java EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, y esto hace que el desarrollo de las aplicaciones del empresario sean mucho más simples. (21)

Drools

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. (22)

1.5.3 Modelo

EJB3

Enterprise Java Beans (EJB) es la tecnología del lado del servidor para la arquitectura de componentes de la Plataforma Java, (Java EE). La tecnología EJB permite el desarrollo rápido y simplificado de aplicaciones distribuidas, transaccionales, seguras y portátiles basadas en tecnología Java. EJB 3.0 define la nueva API EJB simplificado dirigido a la facilidad de desarrollo e incluye el nuevo Java Persistence API para la gestión de la persistencia. (23)

JPA

Java Persistence API (JPA) es el estándar para la gestión de la persistencia y provee facilidades para el mapeo objeto / relacional a desarrolladores de aplicaciones, haciendo uso del modelo de dominio de Java para administrar bases de datos relacionales. JPA es parte de la plataforma Java EE. (24)

Hibernate

Framework que provee herramientas de mapeo objeto/relacional y permite reducir significativamente el tiempo de desarrollo. Con su API nativa es el servicio base para la persistencia de datos. Posee un lenguaje de consultas llamado HQL (bastante parecido al lenguaje de consultas SQL). Sus herramientas soportan distintos tipos de base de datos lo que confiere cierto nivel de portabilidad a las aplicaciones que lo utilizan. A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos. (25)

Entre algunas características técnicas aporta:

- Modelo de programación natural.
- Gran escalabilidad.
- Software libre.
- Persistencia transparente.

- Mapeado flexible.
- Facilidades en consultas.
- Facilidades en metadatos.

1.5.4 Tecnologías horizontales

Java

Lenguaje de programación desarrollado por la compañía Sun Microsystems, que utiliza el paradigma de la Programación Orientada a Objetos (POO). 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. (26)

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. (27)

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. (28)

1.5.5 Modelo de Capacidad y Madurez (CMMI)

Es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios. Nacido en el Software Engineering Institute perteneciente a la Carnegie Mellon University, CMMI es la nueva generación de una línea de modelos de madurez que se inició a principios de los 90 con el famoso CMM-SW (Capability Maturity Model for Software Engineering). (29)

CMMI cuenta con 22 áreas de procesos que se distribuyen dentro de 5 distintos modelos de madurez, tales como: Inicial, Administrado, Definido, Cuantitativamente administrado y Optimizado. El enfoque en el que se definen estos niveles de madurez para las organizaciones se denomina “Representación Escalonada”.

CMMI también propone otro enfoque que depende la capacidad del proceso, el cual establece 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.

- Metas específicas: describen las características únicas que deben estar presentes para satisfacer las necesidades de dicha área. A su vez, cada meta específica consta de un conjunto de prácticas específicas y sub prácticas. 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.

- 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 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).

El área REQM se encarga de administrar los requisitos del proyecto, identificar inconsistencias entre dichos requisitos, administrar los planes del proyecto y administrar los productos de trabajo para lo cual se guía por el proceso IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos definido en la Universidad de las Ciencias Informáticas. Esta área se apoya en el ciclo de vida que siguen los proyectos que están involucrados en el proceso de mejora es definido por el libro “0505_Ciclo de vida de proyectos pilotos del programa de mejora”, que consta de 9 fases: Estudio preliminar, Modelación del negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación, Despliegue y Soporte, estableciéndose por cada fase la relación con los subprocesos descritos en el libro de procesos específico del área Administración de Requisitos.

La definición del IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos incluye, la relación de roles que realizan las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades. La metodología de Proceso Unificado de Desarrollo (RUP) propone cómo deben ser ejecutadas las actividades para la obtención de los productos típicos de trabajo y la forma de documentar todas las acciones que se llevan a cabo en las mismas.

RUP organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Es un proceso dirigido por los casos de usos o funcionalidades que el sistema debe cumplir para satisfacer las necesidades del cliente, centrado en la arquitectura que se propone para el diseño de la aplicación e iterativo e incremental a lo largo del ciclo de vida del software. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes releases ejecutables. Además describe como diseñar una arquitectura flexible, que se acomode a los cambios, comprensible intuitivamente y promueve una efectiva reutilización de software. Soporta el desarrollo de software basado en componentes: módulos no triviales que completan una función clara.

Esta metodología utiliza UML para la descripción de los productos de trabajo. Estos son obtenidos durante el desarrollo de las fases propuestas en el ciclo de vida del IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos, sin embargo en el caso específico de la fase Modelación del negocio se hace uso del estándar para la descripción de procesos de negocio BPMN.

1.5.6 Lenguaje Unificado de Modelado (UML)

El (UML¹, por sus siglas en inglés), lenguaje utilizado para especificar, visualizar y documentar los artefactos de un sistema de desarrollo de software Orientado a Objetos. Unifica principalmente los métodos de Booch, Rumbaugh (OMT) y Jacobson. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía se encuentra en proceso de estandarización por el OMG (Object Management Group). Ya que permite modelar estructuras complejas, se adecúa a las necesidades de conectividad actual y futura, es concurrente, distribuido y genera código a partir de los modelos y a la inversa. Además cuenta con varios tipos de diagramas los cuales muestran diferentes aspectos de las entidades representadas.

1.5.7 Notación para el Modelado de Procesos de Negocio (BPMN)

Provee una notación estándar que permite el modelado de procesos de negocio de modo más entendible y fácil de leer por parte de todos los involucrados e interesados del negocio. Entre estos interesados están

¹ *Unified Modeling Language*

los analistas del negocio, los desarrolladores técnicos y los administradores del negocio. BPMN crea un puente estándar entre el diseño del proceso de negocio y su implementación.

BPMN es un facilitador de fondo para una nueva iniciativa en el mundo de la arquitectura empresarial, la Administración de Procesos de Negocio (BPM, por sus siglas en inglés). Esta permite gestionar el cambio para mejorar los procesos de negocio y unifica las disciplinas de Modelación de Procesos, Simulación, Flujo de trabajo, Integración de Aplicaciones Empresariales (EAI, por sus siglas en inglés) y la integración Business to Business (B2B) en un único estándar. (30) (31)

1.5.8 Clasificación Internacional de Enfermedades (CIE)

La CIE fue publicada por la Organización Mundial de la Salud, se utiliza a nivel internacional para fines estadísticos sobre la información referente a la morbilidad y mortalidad de los pacientes en diferentes países o áreas, siendo aplicada en los sistemas automáticos y de gestión de información médica para lograr un análisis sistemático de los datos. La misma le asigna códigos alfanuméricos compuestos por cuatro caracteres, a las diversas enfermedades y diagnósticos médicos, permitiendo un fácil almacenamiento y una posterior recuperación de la información. Su estructura se basa en tres volúmenes que contienen capítulos, los cuales se dividen en grupos y dichos grupos cuentan con categorías y subcategorías.

La CIE cuenta con una novena edición **CIE-9**, que se encarga de clasificar enfermedades, traumatismos, envenenamientos, síntomas y signos, hallazgos anormales clínicos y de laboratorio, procedimientos y técnicas quirúrgicas. En la actualidad **CIE-9** fue mejorada por la décima edición denominada **CIE-10**.

1.6 Herramientas

Con los elementos expuestos anteriormente, se pueden definir las herramientas que conforman el ambiente de desarrollo del módulo Bloque Quirúrgico especializado en Oftalmología del Sistema de Información Hospitalaria alas HIS.

Se utilizará Visual Paradigm como herramienta CASE (Computer - Aided Software Engineering) para la modelación de los diagramas UML que soportan el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Utilizando la notación BPMN para generar la documentación del negocio y el lenguaje UML 2.1, el cual ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Se tomará como Entorno Integrado de Desarrollo (IDE², por sus siglas en inglés) a Eclipse ya que, posibilita añadir nuevas funcionalidades al editor, a través de nuevos módulos. Trabaja con librerías RichFaces y los framework Seam e Hibernate, además de realizar la administración y configuración mediante JBoss Server. Permite generar documentación en varios formatos como HTML, Word, PDF entre otros. Y genera código partiendo de los diagramas creados en el mismo. Tiene la ventaja de facilitar la programación en distintos lenguajes y ser multiplataforma. Utilizando finalmente como lenguaje de programación Java y para la administración de la base de datos pgAdmin III.

En este capítulo, luego de realizar un estudio del estado del arte de los sistemas de gestión de información hospitalaria en Cuba y en otros países, se determinó que estos no brindan una solución al problema científico que se plantea. Además, se describieron las tecnologías escogidas para la modelación, diseño e implementación del sistema que se necesita.

² *Integrated Development Environment*

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En los niveles de atención médica secundaria se incluyen las instituciones hospitalarias. Las mismas cuentan en ocasiones con un quirófano especializado en Oftalmología, que se encarga de gestión de recursos humanos y materiales, además de la realización de cirugías oftalmológicas a los pacientes; esto requiere del manejo de grandes volúmenes de información. Por lo cual en el presente capítulo, se propone la elaboración de un sistema que informatice los registros utilizados y se documentan los artefactos de desarrollo que son elaborados como resultado de esta tarea.

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

Producto del estudio realizado se determinó que el área de quirófano especializado de Oftalmología del módulo Bloque Quirúrgico, tiene como objetivo el tratamiento del paciente mediante procedimientos quirúrgicos oftalmológicos; para ello se apoya en los procesos que se describen a continuación, que garantizan en gran medida el correcto funcionamiento de los servicios quirúrgicos oftalmológicos.

2.1.1 Realizar Solicitud de intervención quirúrgica oftalmológica

En caso de que la conducta a seguir de la consulta oftalmológica sea quirúrgica, el Especialista Oftalmológico crea la Solicitud de intervención quirúrgica oftalmológica. Los datos de la Solicitud de intervención quirúrgica oftalmológica varían en dependencia de la especialidad oftalmológica en la que el paciente es atendido. Luego, se consulta la planificación quirúrgica oftalmológica y se asigna una fecha de operación, actualizando la Solicitud de intervención quirúrgica oftalmológica. Además se crean las indicaciones quirúrgicas y si el paciente necesita algún lente, se le asigna un lente intraocular.

2.1.2 Realizar intervención quirúrgica oftalmológica

El proceso Realizar intervención quirúrgica oftalmológica permite la realización de la operación al paciente, el cual consta de tres períodos: Pre-Operatorio, Trans-Operatorio y Post-Operatorio. Durante el acto Pre-Operatorio se le suministra al paciente la anestesia pre-operatoria, se chequea continuamente y en caso de no presentar complicaciones se somete a cirugía. En el acto Trans-Operatorio se le aplica al paciente la anestesia operatoria, es intervenido quirúrgicamente, se monitorean sus signos vitales y si el Especialista Oftalmológico necesitara una muestra de Biopsia del paciente, el Auxiliar de salón llena una

Solicitud de biopsia. Luego en el Post-Operatorio se procede a recuperar al paciente siguiendo su evolución mientras abandona los efectos de la anestesia. Como resultado se obtiene la Nota operatoria oftalmológica, el Listado de funcionarios quirúrgicos y la Hoja de anestesia actualizada durante la operación.

Si durante el acto Pre-Operatorio surge algún inconveniente o complicación del paciente, el Anestesiólogo anuncia las causas en la Solicitud de intervención quirúrgica oftalmológica y se cancela la operación.

2.2 Objeto de automatización

Se desea automatizar la gestión de los procesos mencionados anteriormente para los servicios de: Cirugía Refractiva, Cirugía del Cristalino, Cirugía Oculoplastia, Cirugía Pterigium, Cirugía de Glaucoma y Cirugía de Neurooftalmología. Por lo que se propone la informatización de los documentos que se derivan de cada proceso, entre los que se encuentran: Solicitud de intervención quirúrgica oftalmológica, Planificación, Indicaciones quirúrgicas, Hoja de anestesia, Solicitud de análisis de laboratorio, Resultado de análisis de laboratorio, Solicitud de interconsulta, Turno, Solicitud de biopsia, Nota operatoria oftalmológica y Listado de funcionarios quirúrgicos.

2.3 Modelado del Negocio

BPMN define diagramas de procesos de negocios basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización. Se compone de un conjunto de elementos gráficos que facilitan un diagrama entendible tanto por audiencias de negocios como técnicas. Presenta cuatro categorías básicas de elementos:

1. Objetos de flujo: Eventos, Actividades, Rombos de control de flujo (Gateways).
2. Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación.
3. Swimlanes (Carriles de piscina): Pool, Lane.
4. Artefactos: Objetos de Datos, Grupo, Anotación.

CMMI en el área de Administración de Requisitos propone el artefacto, "Modelo de procesos de negocio con BPM", que recoge los datos reflejados durante la fase de modelación del negocio.

2.3.1 Actores del negocio

Un actor del negocio es el principal beneficiado de los procesos existentes, es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (32)

| Actor | Descripción |
|-----------|---|
| Paciente. | Es el principal beneficiado con los procesos de negocio del área de quirófano especializado de Oftalmología del módulo Bloque Quirúrgico. |

Tabla 2.1 Descripción de los actores del negocio.

2.3.2 Trabajadores del negocio

Un trabajador del negocio es la abstracción de una persona o sistema, que representa un rol desempeñado en las realizaciones de las actividades identificadas en un proceso de negocio. Un trabajador del negocio colabora con otros trabajadores, es notificado de los eventos y manipula las entidades para realizar sus responsabilidades. (33)

| Trabajador | Descripción |
|-----------------------------|--|
| Especialista Oftalmológico. | Médico especializado en Oftalmología. |
| Anestesiólogo. | Médico encargado de suministrar la anestesia al paciente durante la intervención quirúrgica. |
| Técnico. | Realiza al paciente los análisis de laboratorio. |
| Auxiliar de Salón. | Encargado de llenar la Solicitud de biopsia. |

Tabla 2.2 Descripción de los trabajadores del negocio

2.4 Diagramas de Procesos del Negocio

2.4.1 Realizar Solicitud de intervención quirúrgica oftalmológica

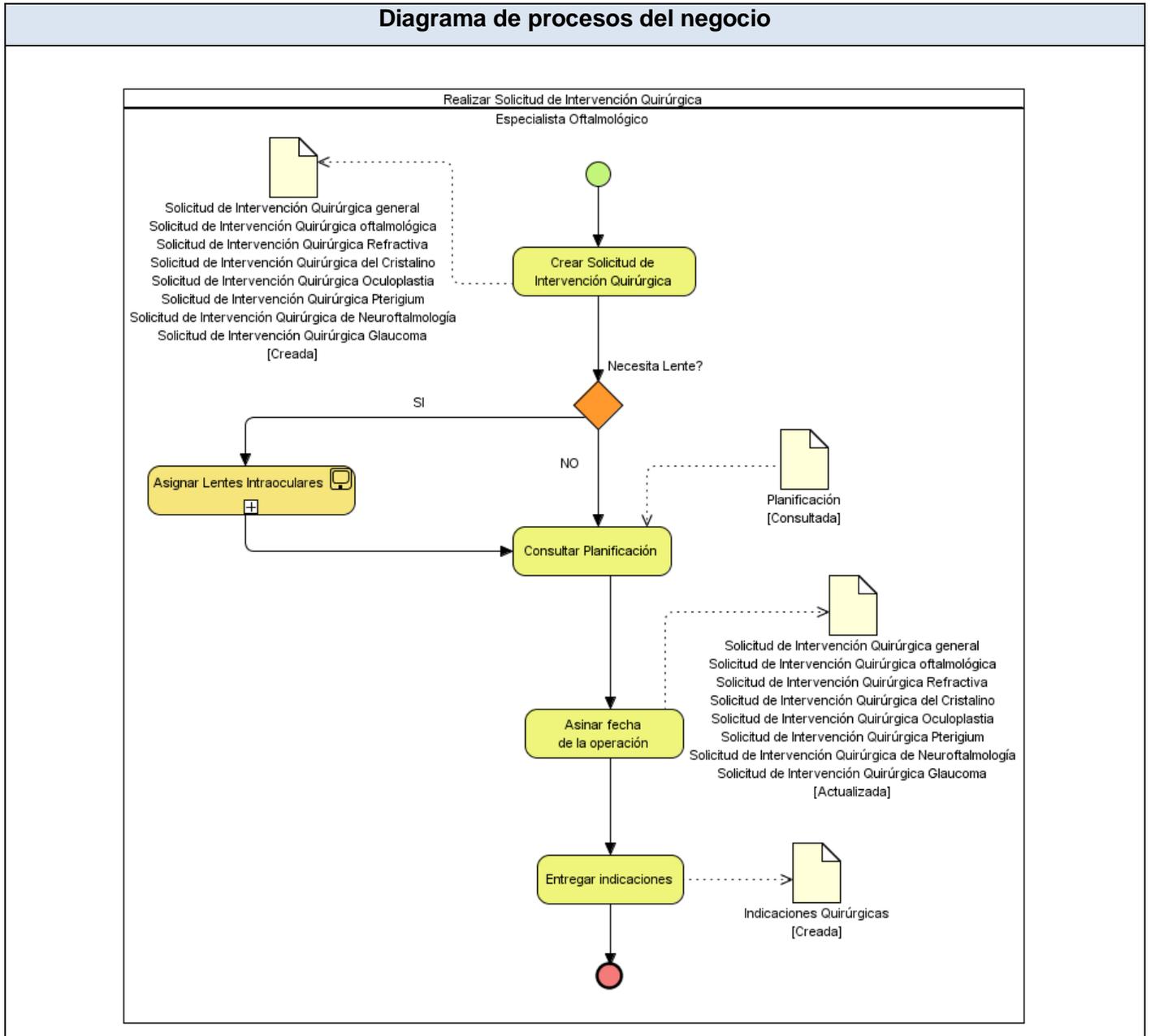


Figura 2.1 Diagrama de proceso del negocio *Realizar Solicitud de intervención quirúrgica*

2.4.2 Realizar intervención quirúrgica oftalmológica

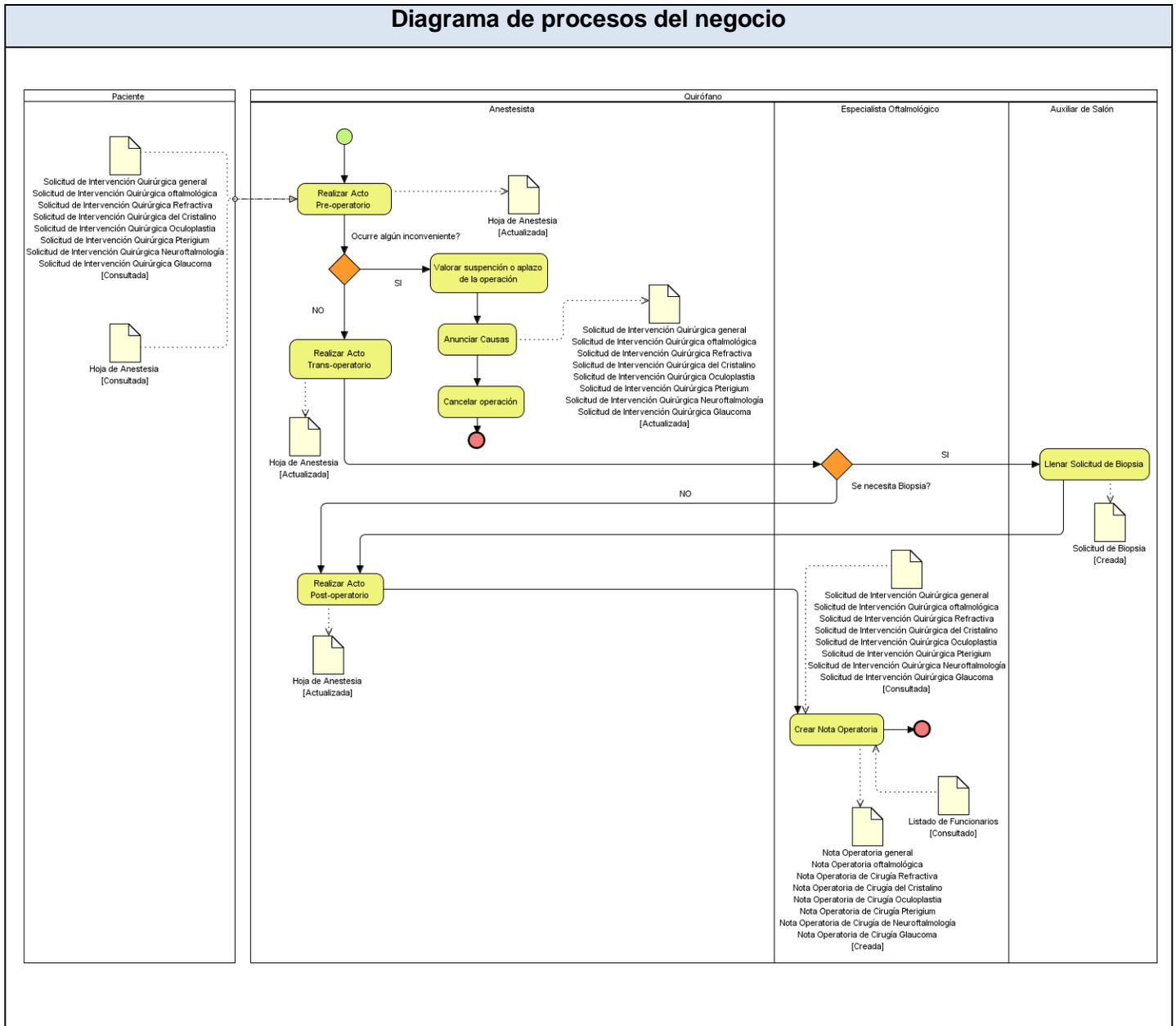


Figura 2.2 Diagrama de proceso del negocio *Realizar intervención quirúrgica*

2.5 Propuesta del Sistema

Surge una propuesta a partir del Modelado de Negocio, donde se obtuvieron las principales funcionalidades que el sistema debe realizar. A continuación se gestionan los requerimientos, los cuales representan una guía para elaborar el Diagrama de casos de uso del sistema, donde se identifican los casos de uso arquitectónicamente significativos.

2.5.1 Especificación de Requisitos de Software

Durante la primera etapa de desarrollo de Software se extraen los requisitos de software. Estos son de gran importancia ya que permiten elaborar un producto de software de mayor calidad. Los requisitos de software son definidos como condiciones o capacidades que deben estar presentes en un sistema o componentes de éste, para satisfacer un contrato, estándar, especificación u otro documento formal, según el Glosario de la IEEE³. Los requisitos pueden dividirse en requisitos funcionales y requisitos no funcionales.

2.5.2 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen y muestran como los casos de uso serán llevados a la práctica. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas, se definieron los siguientes requisitos funcionales:

| Nº | Funcionalidad | Descripción | Complejidad |
|------|---|--|-------------|
| RF1. | Crear Solicitud de intervención quirúrgica oftalmológica | Crear la Solicitud de intervención quirúrgica oftalmológica | Alta |
| RF2. | Ver datos de Solicitud de intervención quirúrgica oftalmológica | Ver la Solicitud de intervención quirúrgica oftalmológica seleccionada con | Media |

³ *Learning Technology Standards Committee*

| | | | |
|-------|---|---|-------|
| | | anterioridad | |
| RF3. | Crear Solicitud de intervención quirúrgica de Refractiva | Crear la Solicitud de intervención quirúrgica para el servicio de cirugía Refractiva | Alta |
| RF4. | Ver datos de Solicitud de intervención quirúrgica de Refractiva | Ver la Solicitud de intervención quirúrgica de Refractiva seleccionada con anterioridad | Media |
| RF5. | Crear Solicitud de intervención quirúrgica de Oculoplastia | Crear la Solicitud de intervención quirúrgica para el servicio de Oculoplastia | Alta |
| RF6. | Ver datos de Solicitud de intervención quirúrgica de Oculoplastia | Ver Solicitud de intervención quirúrgica de Oculoplastia seleccionada con anterioridad | Media |
| RF7. | Crear Solicitud de intervención quirúrgica de Pterigium | Crear la Solicitud de intervención quirúrgica para el servicio de Pterigium | Alta |
| RF8. | Ver datos de Solicitud de intervención quirúrgica de Pterigium | Ver la Solicitud de intervención quirúrgica de Pterigium seleccionada con anterioridad | Media |
| RF9. | Crear Solicitud de intervención quirúrgica de Glaucoma | Crear la Solicitud de intervención quirúrgica para el servicio de Glaucoma | Alta |
| RF10. | Ver datos de Solicitud de intervención quirúrgica de Glaucoma | Ver Solicitud de intervención quirúrgica de Glaucoma seleccionada con anterioridad | Media |
| RF11. | Crear Solicitud de intervención quirúrgica de Neuroftalmología | Crear la Solicitud de intervención quirúrgica para el servicio de Neuroftalmología | Alta |

| | | | |
|-------|---|--|-------|
| RF12. | Ver datos de Solicitud de intervención quirúrgica de Neuroftalmología | Ver Solicitud de intervención quirúrgica de Neuroftalmología seleccionada con anterioridad | Media |
| RF13. | Crear Solicitud de intervención quirúrgica del Cristalino | Crear la Solicitud de intervención quirúrgica para el servicio de Cirugía del Cristalino | Alta |
| RF14. | Ver datos de Solicitud de intervención quirúrgica del Cristalino | Ver la Solicitud de intervención quirúrgica del Cristalino seleccionada con anterioridad | Media |
| RF15. | Crear Nota operatoria oftalmológica | Crear la Nota operatoria oftalmológica | Alta |
| RF16. | Ver detalles de Nota operatoria oftalmológica | Ver la Nota operatoria oftalmológica seleccionada con anterioridad | Media |
| RF17. | Crear Nota operatoria de Refractiva | Crear la Nota operatoria del servicio de cirugía Refractiva | Alta |
| RF18. | Ver detalles de Nota operatoria de Refractiva | Ver la Nota operatoria de Refractiva seleccionada con anterioridad | Media |
| RF19. | Crear Nota operatoria de Oculoplastia | Crear la Nota operatoria del servicio de Oculoplastia | Alta |
| RF20. | Ver detalles de Nota operatoria de Oculoplastia | Ver la Nota operatoria de Oculoplastia seleccionada con anterioridad | Media |
| RF21. | Crear Nota operatoria de Pterigium | Crear la Nota operatoria del servicio de Pterigium | Alta |
| RF22. | Ver detalles de Nota operatoria de Pterigium | Ver la Nota operatoria servicio de Pterigium seleccionada con anterioridad | Media |

| | | | |
|-------|---|--|-------|
| RF23. | Crear Nota operatoria de Glaucoma | Crear la Nota operatoria de Glaucoma | Alta |
| RF24. | Ver detalles de Nota operatoria de Glaucoma | Ver la Nota operatoria de Glaucoma seleccionada con anterioridad | Media |
| RF25. | Crear Nota operatoria de Neuroftalmología | Crear la Nota operatoria del servicio de cirugía Neuroftalmología | Alta |
| RF26. | Ver detalles de Nota operatoria de Neuroftalmología | Ver la Nota operatoria de Neuroftalmología seleccionada con anterioridad | Media |
| RF27. | Crear Nota operatoria del Cristalino | Crear la Nota operatoria del servicio de cirugía del Cristalino | Alta |
| RF28. | Ver detalles de Nota operatoria del Cristalino | Ver la Nota operatoria del Cristalino seleccionada con anterioridad | Media |

Tabla 2.3 Requisitos funcionales del sistema.

2.5.3 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades se deben concebir para obtener un sistema atractivo, usable, rápido y confiable. Por lo que ayudarán a definir si un producto es bien aceptado o poco aceptado. A continuación se especifican los requisitos no funcionales que se tendrán en cuenta para elaborar el sistema:

- **RNF Usabilidad**

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

Usuarios normales: 20 días

Usuarios avanzados: 30 días

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

Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.

Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.

El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.

Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, 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.

- **RNF 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.

- **RNF Soporte**

- ✓ **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.

- ✓ **Monitoreo de funcionamiento**

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

- ✓ **Respaldo y recuperación de base de datos**

Se permitirán 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.

- ✓ **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.

✓ Configuración de parámetros

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

✓ 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.

• RNF Restricciones de diseño

El sistema estará dividido en las siguientes capas:

✓ 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. Estas mismas condiciones se aplican para los servidores de aplicación del Centro de Datos.

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.

✓ 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.

- **RNF Interfaz**

- ✓ **Interfaces de usuario**

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

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

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

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

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.

Se diseñarán salidas de información en forma de gráficos, estadísticas, pronósticos y análisis comparativos que puedan ser usados como soporte para la toma de decisiones.

El sistema incluirá reportes estándares y parametrizables que permitirán al usuario configurar la información de salida y el orden en que aparecen los datos. Las salidas se podrán generar en el formato de fichero PDF.

2.6 Modelo de casos de uso del sistema

El Modelo de casos de uso del sistema facilita la comunicación entre desarrolladores y clientes para llegar a un entendimiento sobre las condiciones que debe cumplir el sistema. Representan gráficamente los requisitos funcionales, ya que describen qué es lo que debe hacer el sistema, pero no cómo. Está formado

por actores, casos de usos y las relaciones que se establecen entre ellos. Es de gran valor para las siguientes fases de construcción del software.

2.6.1 Definición de actores

| Actores | Descripción |
|---|---|
| Gestor de la Hoja oftalmológica. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica oftalmológica. |
| Gestor de la Hoja especializada de Refractiva. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica de Refractiva. |
| Gestor de la Hoja especializada del Cristalino. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica del Cristalino. |
| Gestor de la Hoja especializada de Oculoplastia. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica de Oculoplastia. |
| Gestor de la Hoja especializada de Pterigium. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica de Pterigium. |
| Gestor de la Hoja especializada de Glaucoma. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica de Glaucoma. |
| Gestor de la Hoja especializada de Neurooftalmología. | Es el especialista oftalmológico encargado de crear la Solicitud de intervención quirúrgica de Neurooftalmología. |
| Cirujano oftalmológico. | Es el especialista oftalmológico encargado de crear la Nota operatoria oftalmológica. |

| | |
|--|--|
| Cirujano oftalmológico de Refractiva. | Es el especialista oftalmológico encargado de crear la Nota operatoria de Refractiva. |
| Cirujano oftalmológico del Cristalino. | Es el especialista oftalmológico encargado de crear la Nota operatoria del Cristalino. |
| Cirujano oftalmológico de Oculoplastia. | Es el especialista oftalmológico encargado de crear la Nota operatoria de Oculoplastia. |
| Cirujano oftalmológico de Pterigium. | Es el especialista oftalmológico encargado de crear la Nota operatoria de Pterigium. |
| Cirujano oftalmológico de Glaucoma. | Es el especialista oftalmológico encargado de crear la Nota operatoria de Glaucoma. |
| Cirujano oftalmológico de Neurooftalmología. | Es el especialista oftalmológico encargado de crear la Nota operatoria de Neurooftalmología. |
| Anestesiólogo. | Encargado de crear y actualizar la Hoja de anestesia. |
| Técnico. | Encargado de crear los Resultados de los análisis de laboratorio. |
| Auxiliar de salón. | Encargado de llenar la Solicitud de biopsia en caso de que se necesite durante el acto quirúrgico. |

2.4 Tabla de descripción de actores del sistema

2.6.2 Vista global de actores

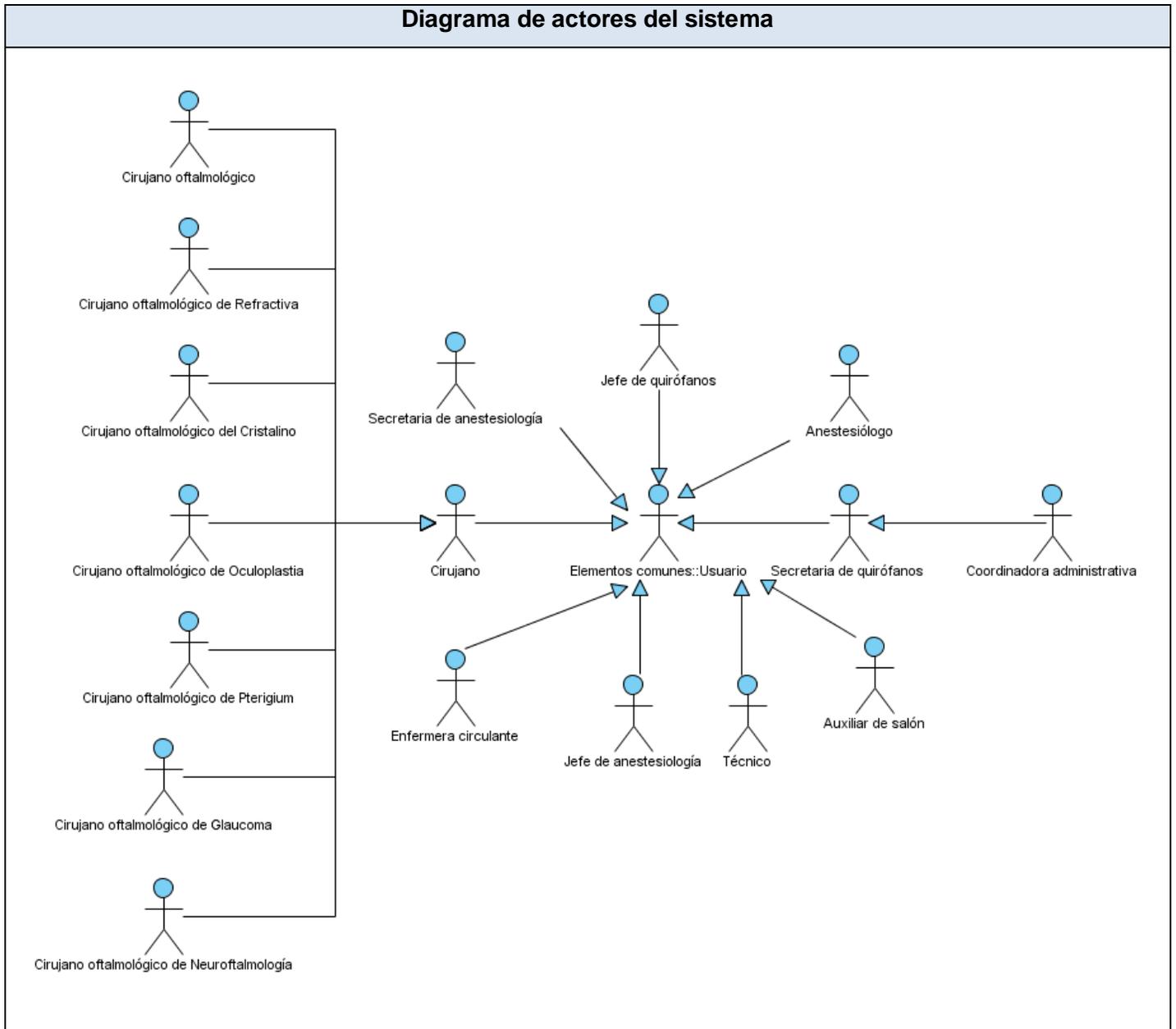


Figura 2.4 Representación de los actores del sistema

2.6.3 Diagrama de casos de uso públicos

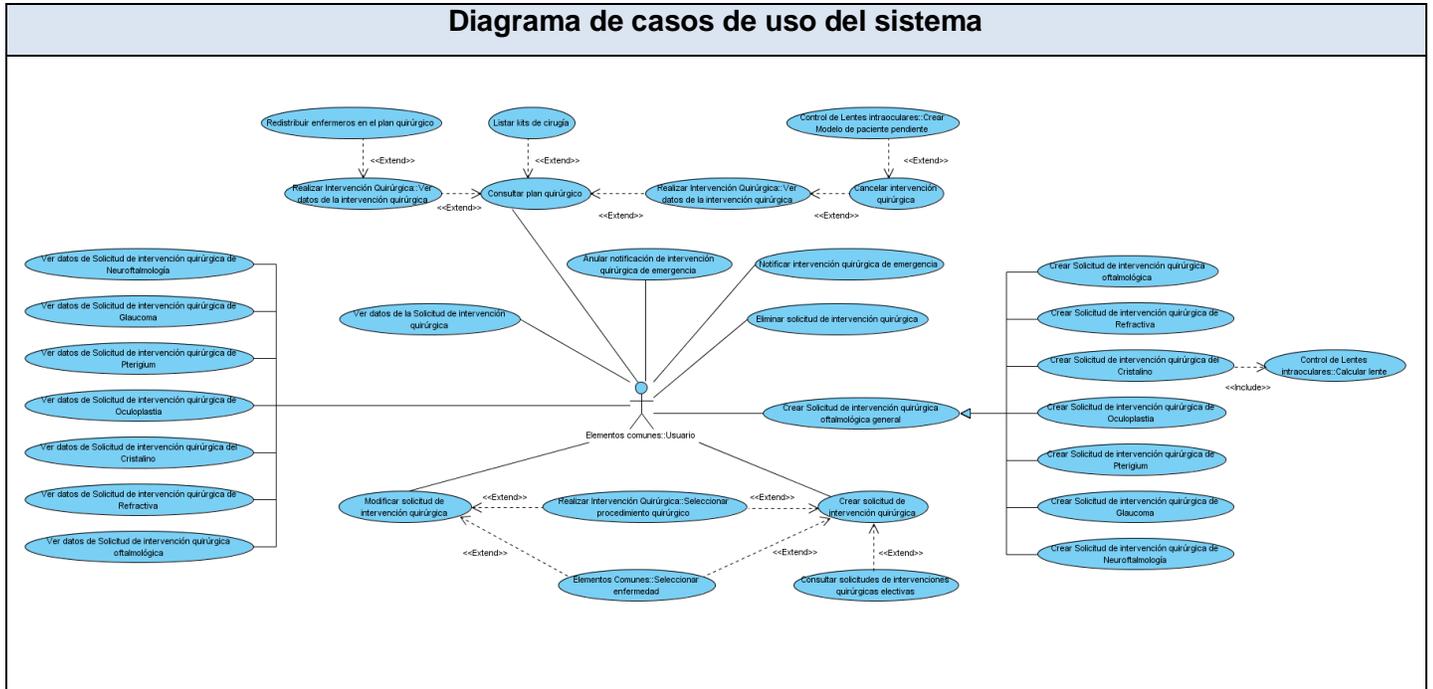


Figura 2.5 Representación de los casos de uso públicos del sistema

2.6.4 Diagrama de casos de uso “Realizar intervención quirúrgica”

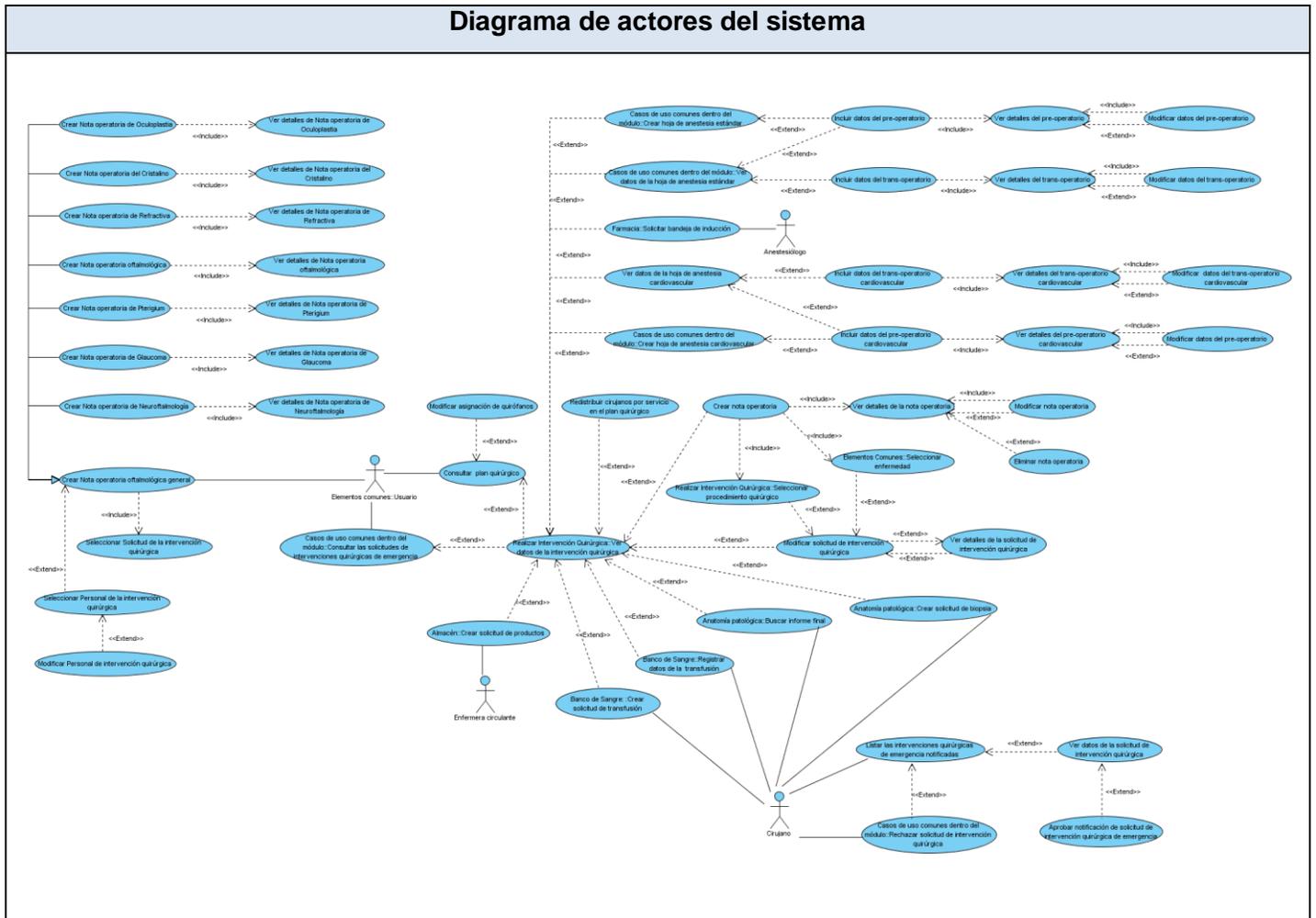


Figura 2.6 Representación de los casos de uso del sistema *Proceso Realizar intervención quirúrgica*

2.6.4 Descripción Textual de los Casos de Uso

CU1. Crear Solicitud de intervención quirúrgica oftalmológica

| | |
|------------------------|---|
| Objetivo | Crear Solicitud de intervención quirúrgica oftalmológica |
| Actores | Gestor de la Hoja oftalmológica |
| Resumen | El caso de uso inicia cuando el actor accede a la opción Crear Solicitud de intervención quirúrgica oftalmológica, el sistema brinda la posibilidad de introducir los datos para crear la Solicitud de intervención quirúrgica oftalmológica, el actor introduce los datos de la Solicitud de intervención quirúrgica oftalmológica, el sistema crea la Solicitud de intervención quirúrgica oftalmológica, el caso de uso termina. |
| Complejidad | Alta |
| Prioridad | Crítico |
| Precondiciones | |
| Postcondiciones | Se creó la Solicitud de intervención quirúrgica oftalmológica |

Tabla 2.5 Descripción textual CU1 *Crear Solicitud de intervención quirúrgica oftalmológica*

CU2. Crear Nota operatoria oftalmológica

| | |
|------------------------|---|
| Objetivo | Crear Nota operatoria oftalmológica |
| Actores | Cirujano oftalmológico |
| Resumen | El caso de uso inicia cuando el actor accede a la opción Crear Nota operatoria oftalmológica, el sistema brinda la posibilidad de introducir los datos para crear la Nota operatoria oftalmológica, el actor introduce los datos de la Nota operatoria oftalmológica, el sistema crea la Nota operatoria oftalmológica, el caso de uso termina. |
| Complejidad | Alta |
| Prioridad | Crítico |
| Precondiciones | La Hoja de anestesia ha sido actualizada |
| Postcondiciones | Se creó la Nota operatoria oftalmológica |

Tabla 2.6 Descripción textual CU2 *Crear Nota operatoria oftalmológica*

CU3. Ver datos de Solicitud de intervención quirúrgica oftalmológica

| | |
|------------------------|---|
| Objetivo | Ver datos de Solicitud de intervención quirúrgica oftalmológica |
| Actores | Gestor de Hoja oftalmológica |
| Resumen | El caso de uso inicia cuando el actor selecciona una Solicitud de intervención quirúrgica oftalmológica y accede a la opción de Ver datos de Solicitud de intervención quirúrgica oftalmológica, el sistema muestra los datos de la Solicitud de intervención quirúrgica oftalmológica, el caso de uso termina. |
| Complejidad | Media |
| Prioridad | Crítico |
| Precondiciones | La Solicitud de intervención quirúrgica oftalmológica ha sido creada. |
| Postcondiciones | Se vieron los datos de la Solicitud de intervención quirúrgica oftalmológica. |

Tabla 2.7 Descripción textual CU3 *Ver datos de Solicitud de intervención quirúrgica oftalmológica*

CU4. Ver detalles de Nota operatoria oftalmológica

| | |
|------------------------|--|
| Objetivo | Ver detalles de Nota operatoria oftalmológica |
| Actores | Cirujano oftalmológico |
| Resumen | El caso de uso inicia cuando es creada o modificada una Nota operatoria oftalmológica, el sistema muestra los datos de la Nota operatoria oftalmológica, el caso de uso termina. |
| Complejidad | Media |
| Prioridad | Crítico |
| Precondiciones | La Nota operatoria oftalmológica ha sido creada. |
| Postcondiciones | Se vieron los datos de la Nota operatoria oftalmológica. |

Tabla 2.8 Descripción textual CU4. *Ver detalles de Nota operatoria oftalmológica.*

Este capítulo ha permitido obtener un mejor entendimiento del sistema, teniendo en cuenta las condiciones que el mismo debe cumplir para satisfacer las necesidades de los clientes. Se especificaron los requisitos funcionales y no funcionales del sistema, se identificaron los actores que intervienen y casos de uso. Además, se propuso la automatización de las actividades más importantes de cada proceso y se obtuvieron los artefactos que permiten comenzar con el flujo de trabajo de Análisis y Diseño para la elaboración del software.

CAPÍTULO 3: DISEÑO DEL SISTEMA

El diseño se traza el objetivo de traducir los requisitos funcionales a una especificación que indica como implementar el sistema, además se tienen en cuenta los requisitos no funcionales para que la implementación del sistema sea rápida y sin ambigüedades. El presente capítulo está encaminado a comprender los aspectos relacionados con los requisitos, lenguajes de programación, componentes reutilizables, tecnologías de distribución y concurrencia. Se obtendrán artefactos como: Diagrama de paquetes del diseño, Diagramas de clases de diseño y Diagramas de interacción.

3.1 Descripción de la arquitectura

La arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución. Se utilizará como patrón arquitectónico el Modelo Vista Controlador para el desarrollo del sistema, ya que permite aislar el modelo de datos, la lógica del negocio y la vista o interfaz de usuario, en tres componentes distintos. La modularidad y el diseño independiente de cada componente, permiten realizar cambios en cualquiera de estos, sin que afecten a los demás.

El modelo de datos contiene entidades persistentes y utiliza Hibernate para el mapeo objeto relacional, mediante el cual se realizará el mapeo de la base de datos a las clases Java y provee facilidades de consultas de datos y de recuperación. La lógica del negocio está representada por clases controladoras, las cuales son las encargadas de recibir y gestionar los eventos enviados por las interfaces y acorde al carácter de la petición ejecuta una acción. La interfaz de usuario está formada por formularios contenidos en las páginas XHTML, que mediante controles JSF y componentes de la librería RichFaces gestionan los datos con los cuales el usuario realiza determinadas operaciones.

3.2 Modelo de diseño

El Modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones

relacionadas con el entorno de implementación, tienen impacto en el sistema. Además describe las clases más importantes, su organización en paquetes y subsistemas, sirviendo como abstracción de la implementación y utilizándose como entrada fundamental de las actividades de implementación. Es representado por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es una forma de organización de este artefacto, en porciones más manejables. Los casos de uso son realizados por las clases del diseño y sus objetos mediante colaboraciones en el Modelo de diseño, denotando la realización de casos de uso del diseño. (34)

3.2.1 Estructura del diseño

El Modelo de diseño se organiza en subsistemas o paquetes, que contienen sus interfaces y las dependencias entre cada una de ellas, conformando de esta manera la estructura principal del sistema. A continuación se muestra el diagrama con los paquetes del diseño, estructurados según el paradigma de la Arquitectura en tres capas.

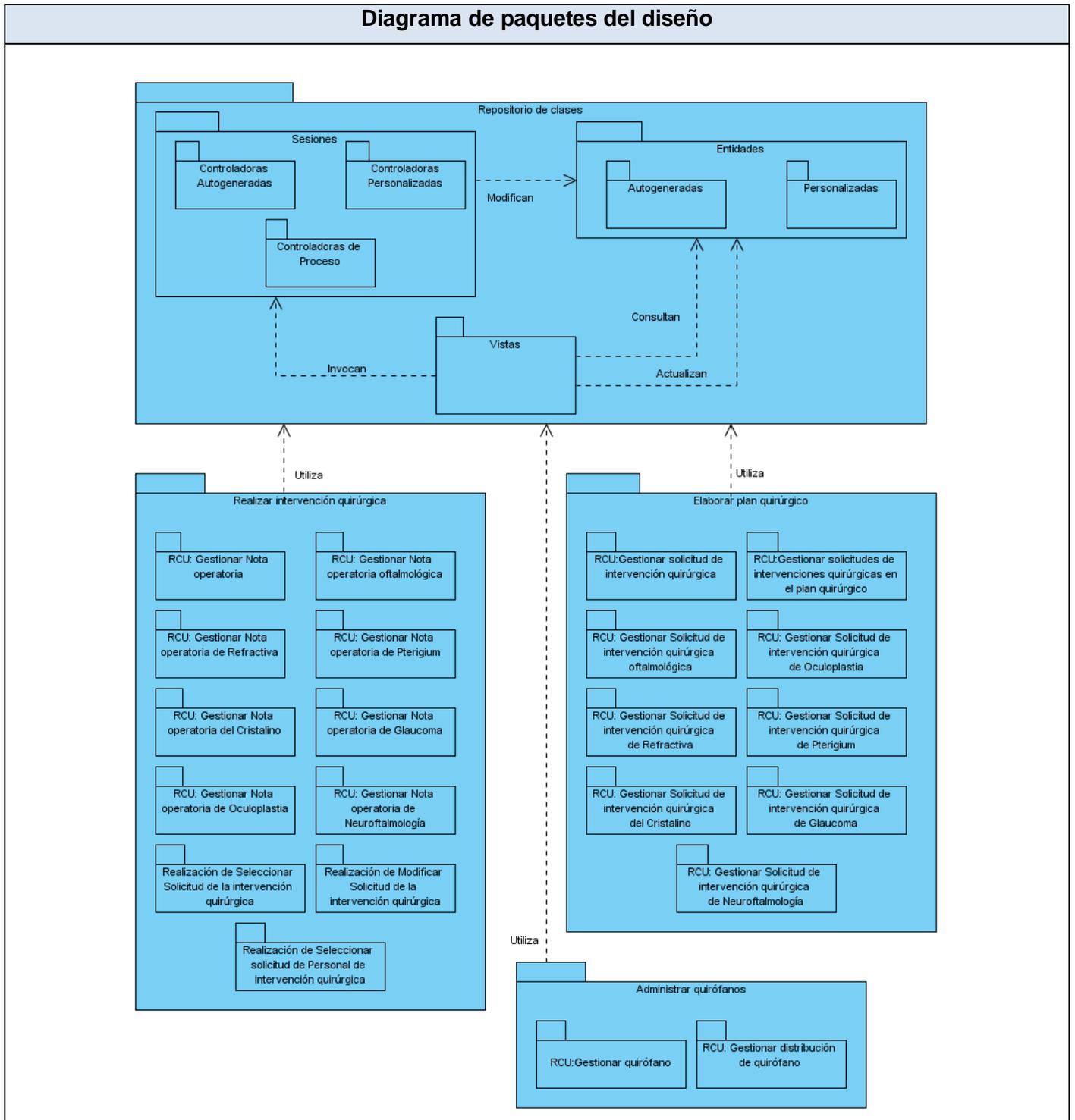


Figura 3.1 Diagrama de paquetes del diseño

A continuación se describen cada uno de los paquetes que conforman el diagrama de paquetes del diseño:

Repositorio de clases: Almacena las clases definidas en el diseño, de acuerdo a las tecnologías que serán utilizadas en la implementación del módulo Bloque Quirúrgico especializado en Oftalmología.

- **Entidades:** Contiene las entidades autogeneradas (obtenidas mediante el proceso de ingeniería inversa de la base de datos) y las personalizadas (modificadas por los desarrolladores y en algunos casos, pueden presentar relaciones de herencia o composición con las autogeneradas).

- **Sesiones:**

Controladoras autogeneradas: Clases controladoras autogeneradas por el entorno de desarrollo, las cuales se obtienen a través del mapeo de objeto relacional.

Controladoras personalizadas: Clases controladoras personalizadas que heredan de las clases controladoras autogeneradas y añaden nuevas funcionalidades.

Controladoras del proceso: Clases controladoras propias del proceso.

- **Vistas:** Contienen todas las vistas del sistema.

Las vistas se ocupan de actualizar y consultar las entidades y a la vez invocan los métodos de las clases controladoras, las cuales modifican los datos de las entidades.

3.2.2 Diagramas de clases del diseño.

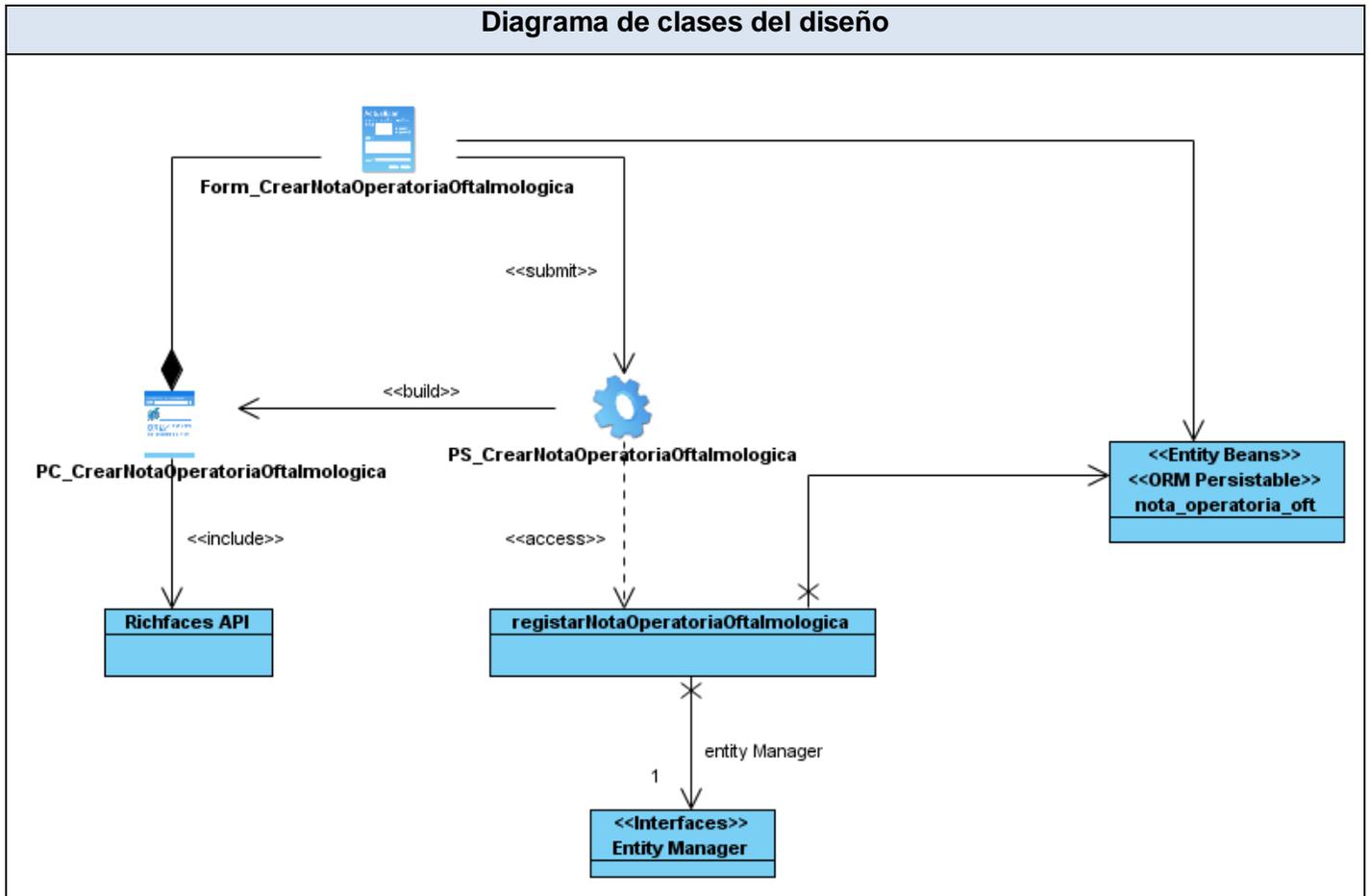


Figura 3.2 Diagrama de clases del diseño *Crear Nota operatoria oftalmológica*

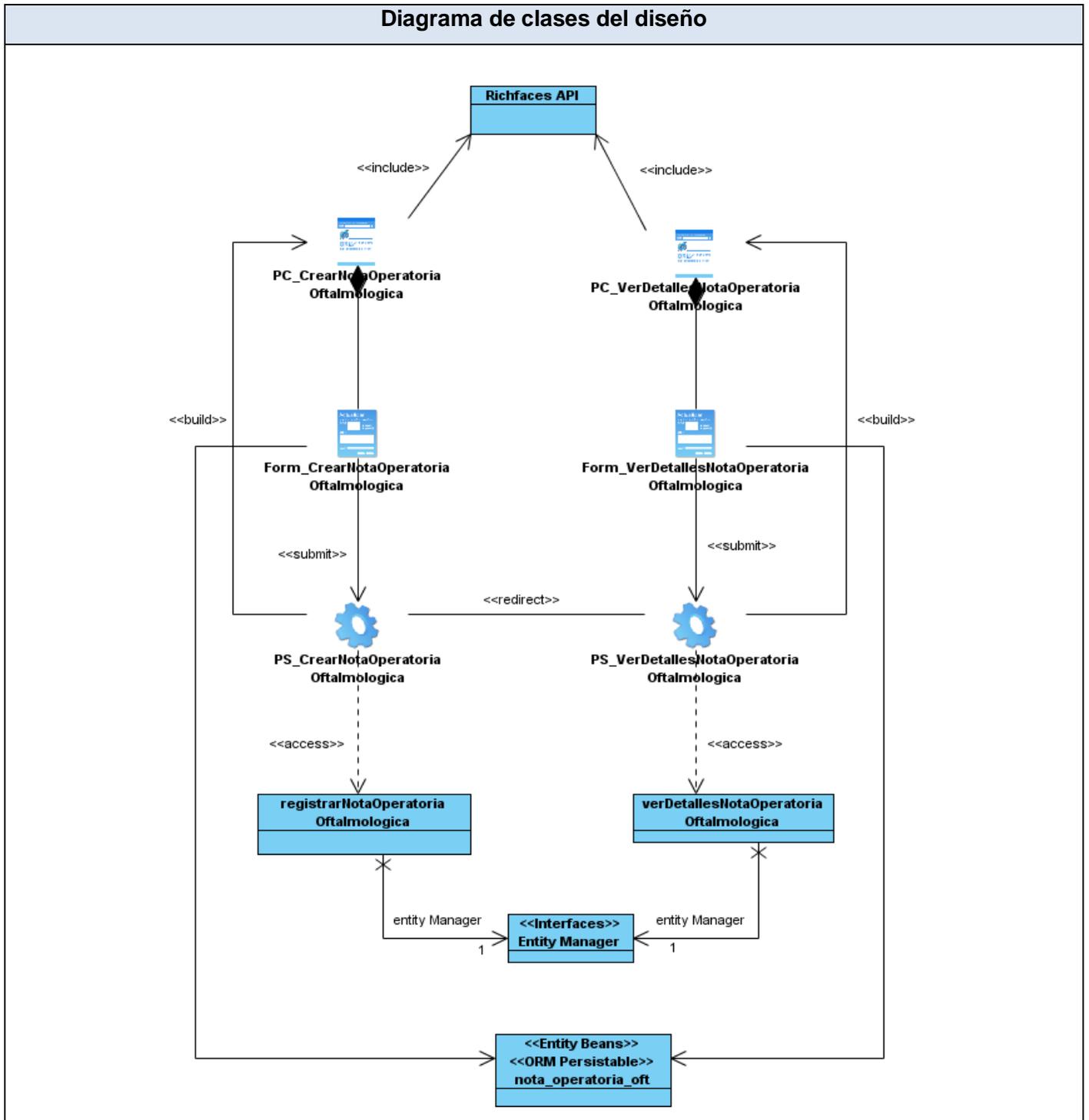


Figura 3.3 Diagrama de clases del diseño Ver detalles de la Nota operatoria oftalmológica

3.2.3 Diagramas de secuencia.

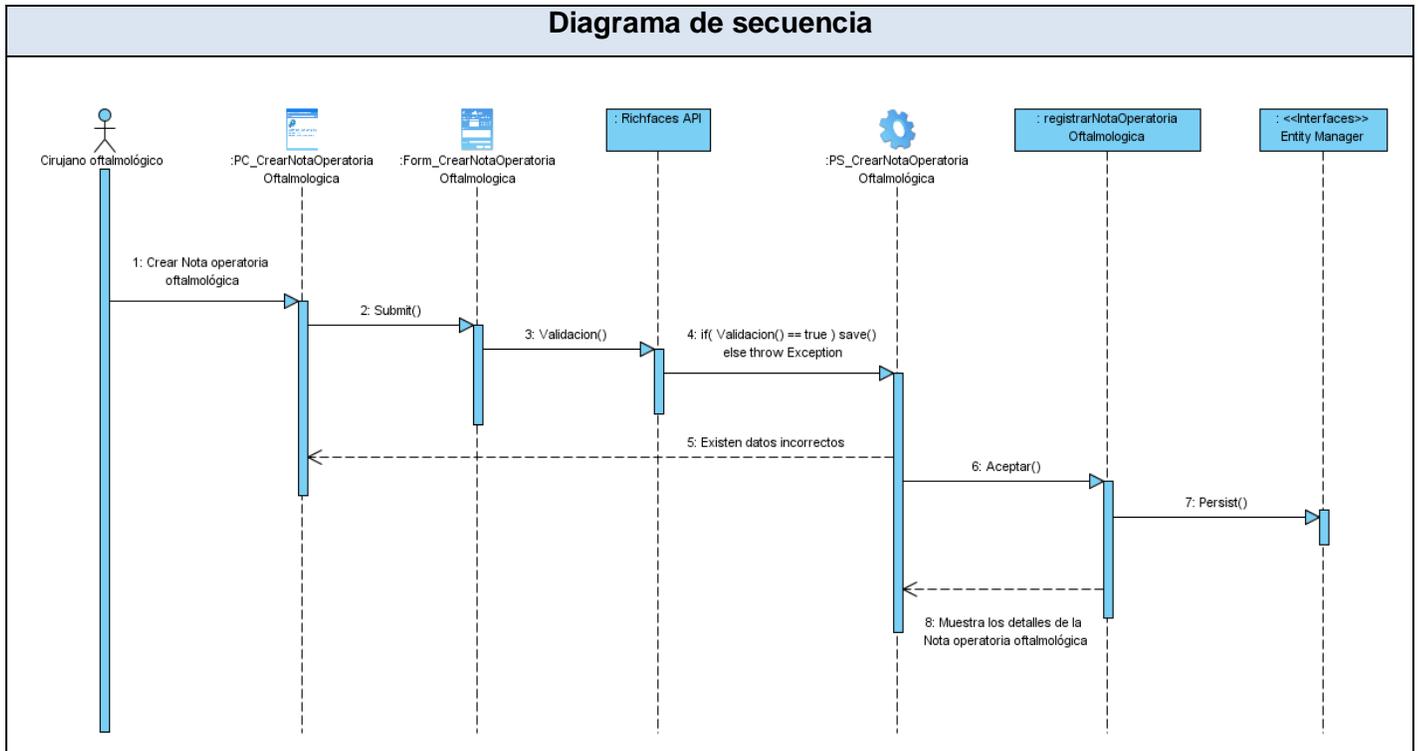


Figura 3.4 Diagrama de secuencia *Crear Nota operatoria oftalmológica*

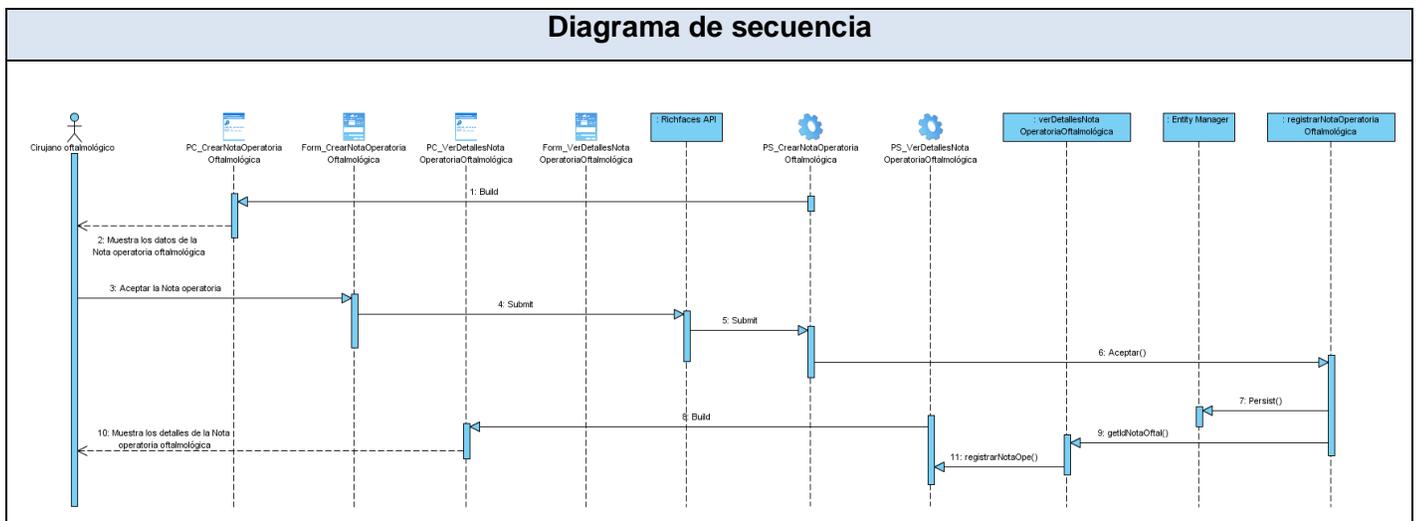


Figura 3.5 Diagrama de secuencia *Ver detalles de la Nota operatoria oftalmológica*

3.2.4 Descripción de las clases del diseño.

| Nombre: registrarNotaOperatoriaOftalmologica | |
|---|------------------------------|
| Tipo de Clase: Controladora | |
| Atributo | Tipo |
| idSolicitud | Integer |
| solicitudIntervencion | SolicitudIntervencion |
| notaOperatoriaOftl_quirofano | NotaOperatoriaOftl_quirofano |
| medicoQuirofano | MedicoQuirofano |
| idProcedimiento | Integer |
| procedimiento | CipProcedimiento |
| diagnostico | DiagnosticoMedico |
| idDiagnostico | Integer |
| complicacionesTrans | DiagnosticoMedico |
| idComplicacionesTrans | Integer |
| idMedicoComplicacion | Integer |
| medicoComplicacion | MedicoQuirofano |
| especOftalmologia | EspecOftalmologia_quirofano |
| especOftalmologiaSeleccionado | String |
| ojoOperar_quirofano | OjoOperar_quirofano |
| ojo_operar | String |
| tipoLente_quirofano | TipoLente_quirofano |
| tipoLente | String |

| | |
|---------------------------|--|
| metodoRefraccion | MetodoRefraccion_quirofano |
| metodo_refraccion | String |
| tipoTratamiento | TipoTratamiento_quirofano |
| tipo_tratamiento | String |
| clasificacionCaracter | ClasificacionOperCaracter_quirofano |
| clasificacion_caracter | String |
| clasificacionComp | ClasificacionOperComplejidad_quirofano |
| clasificacion_complejidad | String |
| clasificacionHerida | ClasificacionOperHerida_quirofano |
| clasificacion_herida | String |
| tipoAnestesia | TipoAnestesiaOft_quirofano |
| tipoIncision | TipoIncision_quirofano |
| esfera | Esfera_quirofano |
| cilindro | Cilindro_quirofano |
| eje | Eje_quirofano |
| cirujanos | List<SolicitudMedicos> |
| anestesiologos | List<SolicitudAnestesiologos> |
| ayudantes | List<SolicitudAyudante> |
| medicosComp | List<MedicoQuirofano> |
| medicamentos | List<MedicamentosInNotaOftalmologicaWrapper> |
| diagnosticoDefString | List<DiagnosticoMedicoEnfermedad_quirofano> |
| complicacionTransString | List<DiagnosticoMedicoEnfermedad_quirofano> |

| | |
|-----------------------------------|---|
| nota | NotaOperatoriaOftl_quirofano |
| Para cada responsabilidad: | |
| Nombre | Descripción |
| registrarNotaOpe. | Permite crear la Nota operatoria oftalmológica. |

Tabla 3.1 Descripción de la clase controladora *registrarNotaOperatoriaOftalmologica*

| |
|--|
| Nombre: Form_CrearNotaOperatoriaOftalmologica |
| Tipo de Clase: Formulario |
| Descripción General: Formulario para introducir y mostrar los datos de la Nota operatoria oftalmológica que contiene las especialidades de cirugía Refractiva, Cristalino, Oculoplastia, Pterigium, Glaucoma y Neurooftalmología. |

Tabla 3.1 Descripción del formulario *Form_CrearNotaOperatoriaOftalmologica*

| |
|--|
| Nombre: PS_CrearNotaOperatoriaOftalmologica |
| Tipo de Clase: Página Servidora |
| Descripción General: Página servidora que se encarga de construir la página cliente de la Nota operatoria oftalmológica que contiene las especialidades de cirugía Refractiva, Cristalino, Oculoplastia, Pterigium, Glaucoma y Neurooftalmología. |

Tabla 3.2 Descripción de la página servidora *PS_CrearNotaOperatoriaOftalmologica*

| |
|---|
| Nombre: PC_CrearNotaOperatoriaRefractiva |
| Tipo de Clase: Página Cliente |
| Descripción General: Página cliente que contiene el formulario de la Nota operatoria oftalmológica que contiene las especialidades de cirugía Refractiva, Cristalino, Oculoplastia, Pterigium, Glaucoma y Neurooftalmología. |

Tabla 3.3 Descripción de la página cliente *PC_CrearNotaOperatoriaRefractiv*

En este capítulo se obtuvieron las realizaciones de casos de uso por procesos a partir de los requisitos funcionales. Se identificaron las clases principales que deben ser implementadas para que la aplicación funcione correctamente. Además se elaboraron los diagramas de secuencia y diagramas de clases del diseño que conforman el modelo de diseño, los cuales constituyen la entrada fundamental en las actividades que se realizan en el flujo de trabajo de Implementación.

CAPÍTULO 4: IMPLEMENTACIÓN

La implementación comienza con el resultado del diseño, se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables, entre otros. La implementación está fuertemente determinada por el lenguaje de programación.

Este capítulo describe como será concebida la base de datos, a partir del modelo de diseño en términos de componentes y la distribución física a través del diagrama de despliegue. Además se aborda sobre el tratamiento de excepciones, la seguridad del sistema, así como las estrategias de codificación, estándares y estilos que se utilizan.

4.1 Modelo de datos

Es el modelo que describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. El Modelo de Datos tiene gran importancia en el ciclo de desarrollo de software, y de manera particular para la fase de implementación, pues define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; y atributos: características básicas de dichos objetos y relaciones: forma en que se enlazan los objetos entre sí. (35)

4.1.1 Descripción de las tablas de la base de datos

| Nombre: hoja_frontal | | | |
|--|--------|------|--|
| Descripción: Entidad que almacena los elementos que identifican a un paciente en un hospital. | | | |
| Atributo | Tipo | Nulo | Descripción |
| numero_hc | string | No | Número de la historia clínica. |
| foto | string | Si | Dirección donde se encuentra la foto. |
| nombres | string | Si | Nombre del paciente. |
| apellido1 | string | Si | Primer apellido del paciente. |
| apellido2 | string | Si | Segundo apellido del paciente. |
| fecha_nacimiento | date | Si | Fecha de nacimiento del paciente. |
| cedula | string | Si | Cadena de números que identifican al paciente. |

Tabla 4.1 Descripción de la tabla *hoja_frontal*

| Nombre: diagnostico_medico | | | |
|---|------|------|---------------------------------------|
| Descripción: Entidad que almacena los tipos de diagnósticos médicos. | | | |
| Atributo | Tipo | Nulo | Descripción |
| fecha_emision | date | No | Fecha en que se emite el diagnóstico. |
| Hora_emision | time | No | Hora en que se emite el diagnóstico. |

Tabla 4.2 Descripción de la tabla *diagnostico_medico*

| Nombre: cip_procedimiento | | | |
|--|--------|------|--|
| Descripción: Entidad que almacena características del procedimiento quirúrgico. | | | |
| Atributos comunes | Tipo | Nulo | Descripción |
| valor | string | No | Nombre del procedimiento. |
| codigo | string | No | Cadena de caracteres que identifican al procedimiento. |

Tabla 4.3 Descripción de la tabla *cip_procedimiento*

| Nombre: quirófano | | | |
|---|--------|------|-----------------------|
| Descripción: Entidad que almacena las características del quirófano. | | | |
| Atributo | Tipo | Nulo | Descripción |
| valor | string | No | Nombre del quirófano. |

Tabla 4.4 Descripción de la tabla *quirófano*

| Nombre: medicamento_producto | | | |
|---|-------|------|--------------------------------|
| Descripción: Entidad que almacena las características de los medicamentos. | | | |
| Atributo | Tipo | Nulo | Descripción |
| concentracion | float | No | Concentración del medicamento. |

Tabla 4.5 Descripción de la tabla *medicamento_producto*

| Nombre: solicitud_intervencion | | | |
|--|--------|------|--|
| Descripción: Entidad que almacena las características de la solicitud de intervención quirúrgica. | | | |
| Atributo | Tipo | Nulo | Descripción |
| fecha_solicitud | date | No | Fecha en la cual se realiza la solicitud de intervención quirúrgica. |
| hora | time | Si | Hora en la cual se realiza la solicitud de intervención quirúrgica. |
| electiva | bool | Si | Indica si la intervención solicitada es electiva. |
| duracion_aproximada | time | Si | Duración aproximada de la intervención solicitada. |
| aprobada | bool | Si | Indica si la solicitud de intervención quirúrgica es aprobada. |
| causa_modificacion_qx | string | Si | Causa por la cual se modifica el quirófano asignado a la solicitud de intervención quirúrgica. |
| causa_rechazo_solicitud | string | Si | Causa por la cual se rechaza la solicitud de intervención quirúrgica. |
| fecha_fijada | date | No | Fecha para la cual se solicita la intervención quirúrgica |
| id_hoja_frontal | int | Si | Identificador único para la tabla hoja _frontal (FK). |

| | | | |
|------------------------|--------|----|---|
| id_quirofano | int | Si | Identificador único para la tabla quirofano (FK). |
| id_procedimiento | int | Si | Identificador único para la tabla cip_procedimiento (FK). |
| id_diagnostico | int | Si | Identificador único para la tabla diagnostico_medico (FK). |
| id_funcionario_aprueba | int | Si | Identificador único para la tabla medico (FK). |
| otros_anestesiologos | string | Si | Médicos especializados en Anestesia que forman parte del personal que realiza la intervención quirúrgica |
| otros_enfermeras | string | Si | Médicos especializados en Enfermería que forman parte del personal que realiza la intervención quirúrgica |

Tabla 4.6 Descripción de la tabla *solicitud_intervencion*

| Nombre: nota_operatoria | | | |
|---|---------|------|---|
| Descripción: Entidad que almacena las características de la nota operatoria. | | | |
| Atributo | Tipo | Nulo | Descripción |
| id_solicitud_intervencion | integer | Si | Identificador único de la tabla solicitud_intervencion (FK) |
| id_cirujano | integer | Si | Identificador único de la tabla cirujano (FK) |
| fecha | integer | Si | Fecha en la cual se realizará la intervención quirúrgica. |
| procedimiento | string | Si | Procedimiento quirúrgico. |
| id_procedimiento | integer | Si | Identificador único de la tabla cip_procedimiento(FK) |
| Id_diagnostico_posoperatorio | integer | Si | Identificador único de la tabla. |

Tabla 4.7 Descripción de la tabla *nota_operatoria*

| |
|--|
| Nombre: nota_operatoria_ofi |
| Descripción: Entidad que almacena las características de la solicitud de intervención quirúrgica. |

| Atributo | Tipo | Nulo | Descripción |
|------------------------|---------|------|--|
| quirofano | string | No | Nombre de salón donde se realizará el acto quirúrgico. |
| turno_quirurgico | integer | No | Turno de la intervención quirúrgica. |
| cirugia_ambulatoria | bool | Si | Indica si la cirugía será ambulatoria. |
| refraccion_esperada | float | No | Indica la refracción que presenta el paciente. |
| actoQx_laborioso | bool | Si | Indica si fue una operación difícil. |
| observaciones | string | Si | Observaciones especificadas por el especialista. |
| evaluacion | string | No | Evaluación del Especialista. |
| min_tiempo_trat | float | Si | Mínimo tiempo de tratamiento indicado en minutos. |
| no_mensajes_eye | integer | Si | Número de Mensajes de eye tracking. |
| no_pulsaciones | float | Si | Cantidad de pulsaciones. |
| no_errores | integer | Si | Cantidad de errores. |
| tiempo_entre_pulsos | integer | Si | Tiempo entre el primer y último pulso. |
| estado_eyetracking | string | Si | Estado del eye tracking. |
| tratamiento_realizado | string | Si | Se describe el tratamiento. |
| tratamiento_finalizado | string | Si | Se describe como finaliza el tratamiento. |
| ablacion | float | Si | Cantidad de ablaciones por capas. |
| laserHV_iniciar | integer | Si | Número de Laser HV al iniciar el tratamiento. |
| laserHV_finalizar | integer | Si | Número de Laser HV al finalizar el tratamiento. |
| scanner | float | Si | Scanner offset. |
| energia_laser | integer | Si | Energía de láser. |
| tratamiento_Tfluence | string | Si | Tratamiento desde el último test fluence. |

| | | | |
|--------------------------------|---------|----|--|
| temperatura | float | Si | Temperatura. |
| humedad | int | Si | Humedad. |
| comentario | string | Si | Comentario. |
| lente_otro_ojo | bool | Si | Indica si presenta lente en el otro ojo o no. |
| id_espec_of | integer | Si | Identificador único de la tabla espec_ofthalmologia (FK) |
| id_tipo_incisiones | integer | Si | Identificador único de la tabla tipo_incisiones (FK) |
| id_ojo_operar | integer | Si | Identificador único de la tabla ojo_operar (FK) |
| id_clas_oper_complejidad | integer | Si | Identificador único de la tabla clasificacion_oper_complejidad (FK) |
| id_clas_oper_herida | integer | Si | Identificador único de la tabla clasificacion_oper_herida (FK) |
| id_clas_oper_caracter | integer | Si | Identificador único de la tabla clasificacion_oper_caracter (FK) |
| id_tipo_anest_of | integer | Si | Identificador único de la tabla tipo_anest_of (FK) |
| id_metodo_refraccion | integer | Si | Identificador único de la tabla metodo_refraccion (FK) |
| id_tipo_tratamiento | integer | Si | Identificador único de la tabla tipo_tratamiento (FK) |
| id_tipo_lente | integer | Si | Identificador único de la tabla tipo_lente (FK) |
| id_diam_zona_ablacion_cilindro | integer | Si | Identificador único de la tabla diametro_zona_ablacion_cilindro (FK) |
| id_diam_zona_ablacion_esfera | integer | Si | Identificador único de la tabla diametro_zona_ablacion_esfera (FK) |
| id_diam_zona_optica_cilindro | integer | Si | Identificador único de la tabla diametro_zona_optica_cilindro (FK) |
| id_diam_zona_optica_esfera | integer | Si | Identificador único de la tabla |

| | | | |
|--------------------------|---------|----|--|
| | | | diametro_zona_optica_esfera (FK) |
| id_valor_cornea_esfera | integer | Si | Identificador único de la tabla valor_cornea_cilindro (FK) |
| id_valor_cornea_cilindro | integer | Si | Identificador único de la tabla valor_cornea_esfera (FK) |
| id_valor_cornea_eje | integer | Si | Identificador único de la tabla valor_cornea_eje (FK) |

Tabla 4.8 Descripción de la tabla *nota_operatoria_oft*

4.2 Modelo de Implementación

Está conformado por los Diagramas de Componentes y de Despliegue, describiendo cómo los elementos del Modelo de Diseño se implementan en términos de componentes, ficheros de código fuente y ejecutables. El Modelo de Implementación describe además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. (36)

4.2.1 Diagrama de Componentes

Es una representación gráfica que muestra un conjunto de elementos del modelo tales como componentes, subsistemas o paquetes de implementación y sus relaciones. Permite modelar la vista estática del sistema mostrando las dependencias lógicas entre un conjunto de componentes de software. Este diagrama se estructura en paquetes, que son divisiones físicas del sistema. Los paquetes están organizados en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Los componentes pueden ser de código fuente, librerías, binarios o ejecutables y tienen relaciones de traza con los elementos del modelo que implementan. En el Diagrama de Componentes elaborado se evidencian las relaciones existentes entre los paquetes y los componentes. Dichas relaciones son de inclusión y pueden ser identificadas por el estereotipado <<import>>. (37)

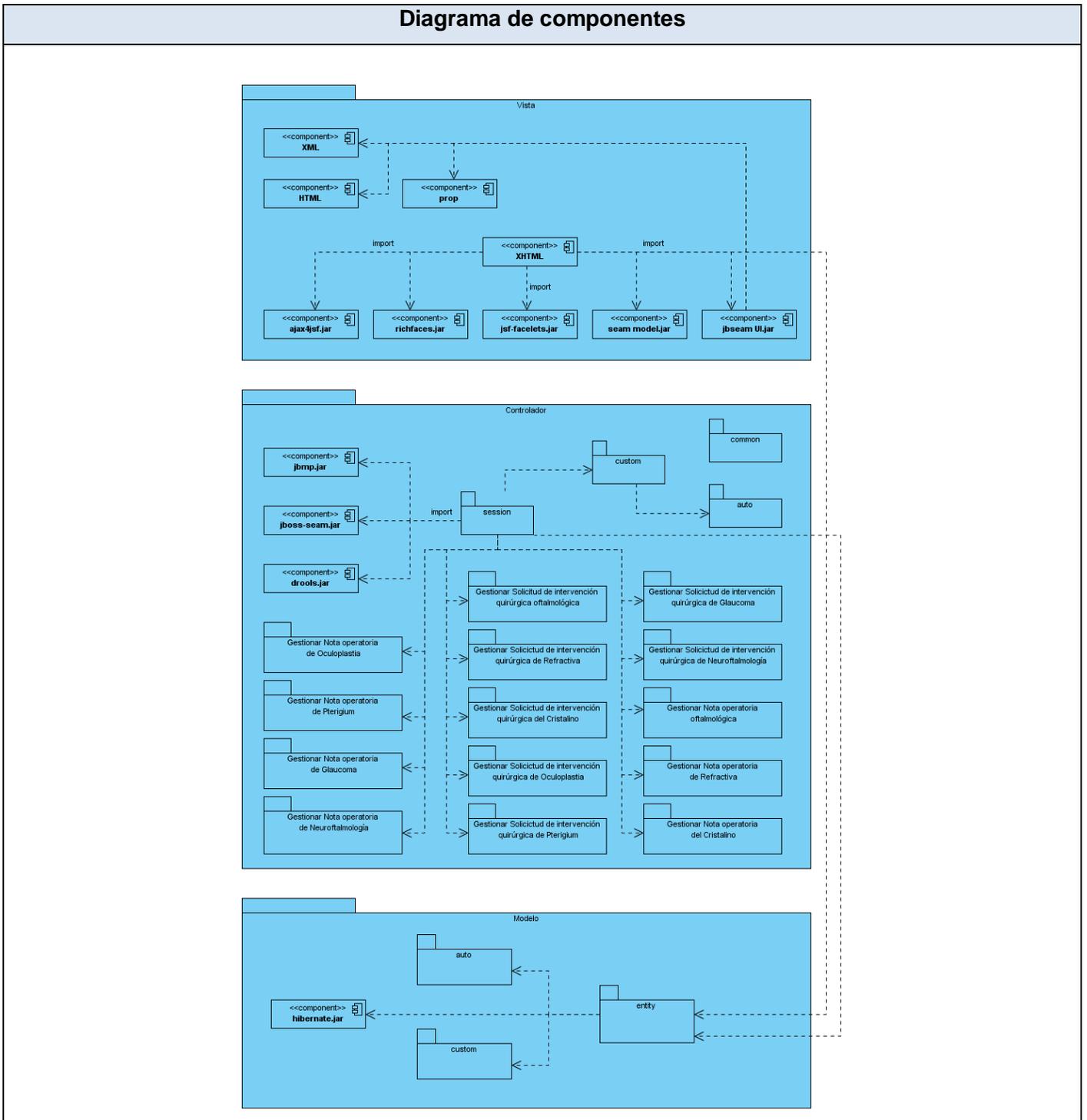


Figura 4.2 Diagrama de componentes

4.2.2 Diagrama de Despliegue

Es un diagrama que muestra la configuración del hardware que participa en la ejecución de los componentes en los que se encuentra distribuido el sistema. Gráficamente, un diagrama de despliegue es una colección de nodos físicos en tiempo de ejecución que pueden representar dispositivos o computadores con capacidad de procesamiento. Estos se relacionan a través de arcos que definen los protocolos de comunicación para el intercambio de información. (38)

El módulo de Bloque Quirúrgico especializado en Oftalmología cuenta con 2 servidores, uno de aplicaciones web y otro de base de datos, conectados entre sí mediante el Protocolo de Control de Transmisión/ Protocolo de Internet (TCP/IP⁴, por sus siglas en inglés). Además contará con las estaciones clientes que se comunicarán con el servidor web mediante el Protocolo de Transferencia de Híper Texto (HTTP⁵, por sus siglas en inglés). A continuación se muestra el Diagrama de despliegue que evidencia las características de software y de hardware de cada uno de los nodos físicos.

⁴ *Hypertext Transfer Protocol*

⁵ *Transmission Control Protocol/Internet Protocol*

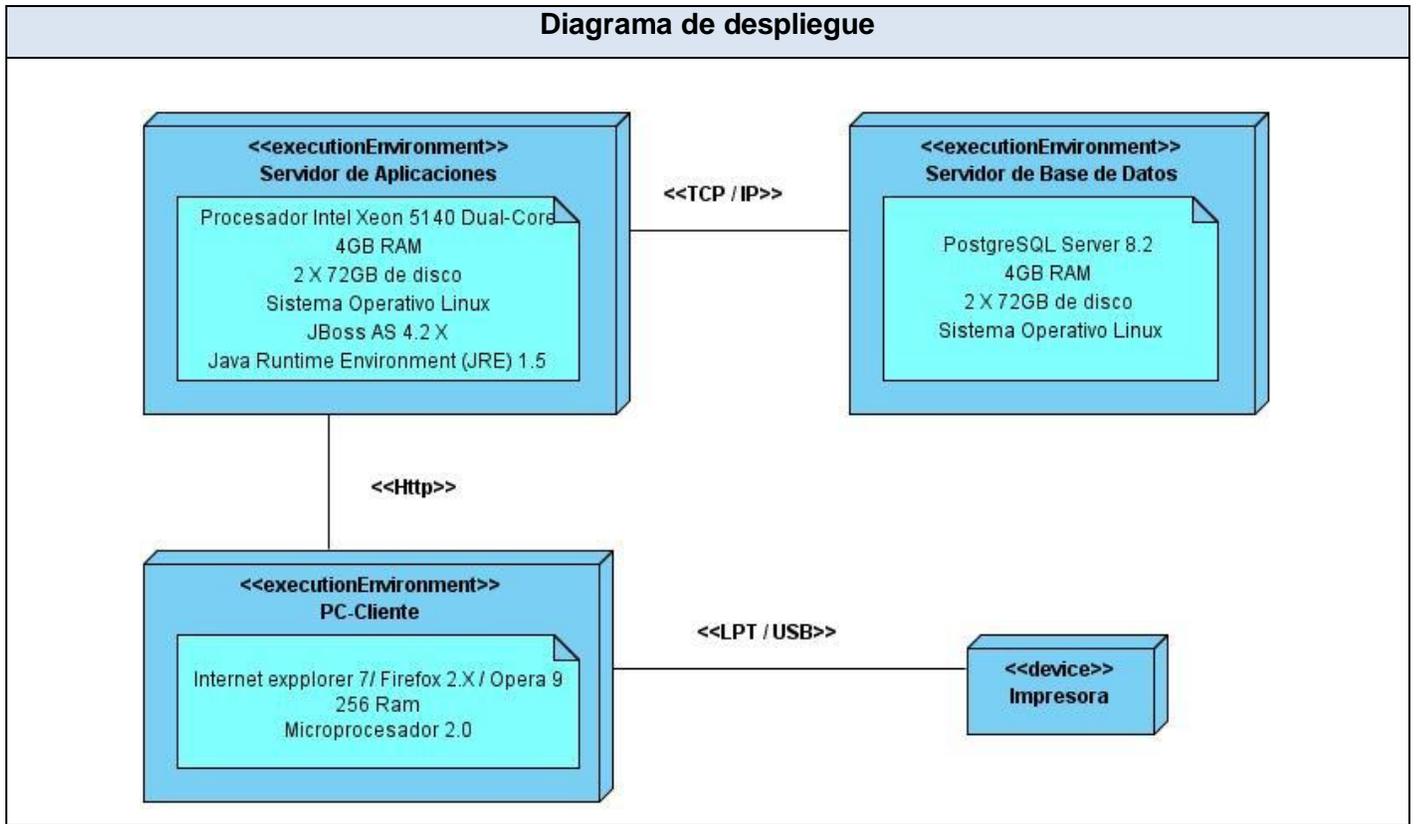


Figura 4.3 Diagrama de despliegue

4.3 Tratamiento de excepciones

Una excepción es un evento que ocurre durante la ejecución de un programa interrumpiendo el flujo normal de las sentencias. Dicho evento puede ser desde serios problemas de hardware, como la avería de un disco duro, hasta los simples errores de programación y pueden ser tratados mediante una estructura de control que poseen los lenguajes de programación de alto nivel, diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla. A esta estructura de control se le conoce como tratamiento de excepciones. (39) Un buen tratamiento de excepciones permite el aumento de la calidad del software, además de obtener un sistema más robusto y fiable.

En el sistema a desarrollar el control de las excepciones se llevará a cabo en todo el código, donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos. También se controlan los errores que pueden surgir en validación de datos provenientes de la interfaz de usuario.

En las clases controladoras de procesos, se utilizará el bloque try para detectar cuando ocurra algún fallo y mediante el catch se manejarán dichas excepciones, mediante mensajes que se muestran en la interfaz de usuario, por las facilidades que brinda el FacesMessages, componente del framework Seam. Existe además un archivo XML, denominado page.xml, que engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, así como la página a la que el sistema redirecciona en caso de la aparición de un error sorpresivo.

4.4 Seguridad

La seguridad en los Sistemas de Información Hospitalaria es de gran importancia y se garantiza cuando se cumplen los aspectos de integridad, confidencialidad y disponibilidad. La unión de estas características trae consigo que la información sea modificada y vista por el personal autorizado; de forma controlable y que esté disponible cuando se necesite.

A continuación se describen detalladamente las funcionalidades que ofrece el módulo de Configuración del Sistema de Información Hospitalaria alas HIS, el cual se encargará de garantizar la seguridad en el sistema a desarrollar.

Iniciar/Cerrar sesión de trabajo

Cuando el usuario necesita acceder al sistema, este solicita: nombre de usuario y contraseña. El usuario introduce los datos solicitados, el sistema verifica que los datos introducidos sean válidos, si es así, el usuario accede al módulo. El sistema muestra como opciones del menú las funcionalidades a las que tiene permiso de acceder el usuario en el módulo, lo que garantiza el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. El sistema permite: cerrar sesión y salir del módulo.

Registrar trazas

Cuando el usuario realiza una acción sobre el sistema, que puede ser: inicio o cierre de sesión, acceso al módulo, modificación a un atributo de una entidad o cualquier otra operación sobre el sistema, este registra una traza en la base de datos.

Administrar seguridad

El sistema brinda la posibilidad de asignar o denegar permiso a roles y usuarios en las funcionalidades del módulo.

Configurar funcionalidades

El sistema brinda la posibilidad de configurar las funcionalidades del módulo.

4.5 Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de codificación definen un conjunto de reglas para escribir el código, los cuales permiten obtener un código de alta calidad y cumplir con las buenas prácticas establecidas en la Ingeniería de Software. Además posibilita que el software que se obtiene sea fácil de comprender y de mantener en el tiempo. A continuación se describen los estándares de codificación que propone el lenguaje de programación Java (Java Code Conventions), que se utilizarán para el desarrollo del sistema.

4.5.1 Idioma

Se debe utilizar como idioma el español, las palabras no se acentuarán.

4.5.2 Variables, parámetros y métodos

En esta notación, el identificador para las variables, los parámetros y los métodos se definen escribiendo las palabras de la siguiente forma, la primera con minúsculas y en caso de que sea un nombre compuesto se empleará notación CamellCasing. Ejemplo: `int cantidadReal`. El nombre empleado, debe permitir que con sólo leerlo se conozca su propósito.

4.5.3 Identación

Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento. Se deben dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque ({ }). Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach. 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.

4.5.4 Comentarios, separadores, líneas, espacios en blanco y márgenes

Establecer un modo común para comentar el código, utilizar separadores, línea, espacios en blanco y márgenes, de forma tal que sea comprensible el código, por lo tanto 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).
- 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.
- Se deben usar espacios en blanco entre los operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nombreProducto.
- 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. No se debe usar espacio en blanco:
 - ✓ Después del corchete abierto y antes del cerrado de un arreglo.
 - ✓ Después del paréntesis abierto y antes del cerrado.
 - ✓ Antes de un punto y coma.

4.5.4. Constantes

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

4.5.5 Clases y Objetos

El objetivo fundamental es nombrar las clases e instancias de forma estándar para todas las aplicaciones. Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing. Ejemplo: MiClase(). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, este se escribirá en minúscula. Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing*. Ejemplo: function BuscarUnidad(). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.

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

En este capítulo se ha concluido el proceso de implementación, donde se obtuvieron los artefactos Diagrama de despliegue y Diagrama de componentes que conforman el Modelo de implementación. Se implementaron los procesos Realizar Solicitud de intervención quirúrgica, consulta pre-anestésica e intervención quirúrgica; de acuerdo a los requerimientos de seguridad necesarios para garantizar la confidencialidad y privacidad de la información que se manipula en todo el sistema; además de cumplir con los estándares de codificación. Permitiendo perfeccionar el funcionamiento del área de Quirófano especializado en Oftalmología.

CONCLUSIONES

Una vez finalizado el trabajo se llega a las siguientes conclusiones:

- Los sistemas analizados vinculados con el campo de acción no constituyen una solución a la problemática identificada, ya que no satisfacen las necesidades requeridas por el área de Quirófano especializado en Oftalmología.
- Se identificaron procesos del negocio que permitieron la comprensión del problema existente y fueron el punto de partida para la obtención de las funcionalidades que la aplicación debe brindar.
- Con la realización del diseño se demostró que la arquitectura definida permite el desarrollo de un sistema robusto y flexible, a través del uso de tecnologías y herramientas no privativas.
- Se implementaron los procesos de negocio: Realizar Solicitud de intervención quirúrgica oftalmológica y Realizar intervención quirúrgica oftalmológica, lo que facilitará la gestión de la información en el área de Quirófano especializado en Oftalmología.

RECOMENDACIONES

Se recomienda:

- Desarrollar otras especialidades oftalmológicas en el sistema.
- Agregar las funcionalidades eliminar y modificar para la Nota operatoria oftalmológica.
- Integrar el sistema al equipo excímer láser, para obtener automáticamente los datos de la cirugía Refractiva.

REFERENCIAS BIBLIOGRÁFICAS

1. **Serón Arbeloa, Carlos y Puzo Foncillas, José.** Diseño y validación de un programa informático de nutrición artificial. Su utilización en una unidad de cuidados intensivos. [En línea] Universidad de Zaragoza (1999), Informática y Medicina. <http://www.terra.es/personal/cseron/tesis/indice1.htm>.
2. Idem 1. [En línea]
3. *Documento de Arquitectura de Software de Bloque Quirúrgico Oftalmológico.*
4. *Especificación de Requisitos de software de Bloque Quirúrgico Oftalmológico.*
5. **SIGHO.** Sistema de Información para la gerencia Hospitalaria. [En línea] 2006. [Citado el: 10 de Febrero de 2010.] http://www.ssn.gob.mx/html/sigho/intro_sigho.html.
6. SUSHRT. [En línea] 2009. [Citado el: 10 de Febrero de 2010.] <http://www.cdac.in/html/his/sushrut.asp>.
7. **Bolotchi, Ing. Hernan.** Siemens. [En línea] 2003. [Citado el: 10 de Febrero de 2010.] http://www.siemens.com.ar/sites/internet/legacy/sie-med/soluciones_informaticas_para_la_gestion_hospitalaria.htm.
8. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.* La Habana, Cuba. : Félix Varela, 2005. Parte 1.
9. **Reinoso, Billy.** Introducción a la Arquitectura de Software. [En línea] [Citado el: 11 de Febrero de 2010.]
10. El Modelo Cliente/Servidor. [En línea] 1994. [Citado el: 11 de Febrero de 2010.] <http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>.
11. **Rojo, J. Oscar.** Asociación de usuarios de GNU/Linux de Castilla y León. [En línea] 2003. [Citado el: 11 de Febrero de 2010.] <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>.
12. **Pérez García, Alejandro.** JSF-Java Server Faces. *DesarrolloWeb.com.* [En línea] 21 de Febrero de 2006. [Citado el: 12 de Febrero de 2010.] <http://www.desarrolloweb.com/articulos/2380.php>.
13. Richfaces Developer Guide. Red Hat. *RichFaces.* [En línea] 2008. <http://www.jboss.org/jbossrichfaces/docs/>. p 17.
14. **Katz, Max.** *Practical Richfaces.* Apress. 2008. p 12.
15. **Ramos, Juan Alonso.** Adictos al trabajo. [En línea] 9 de Abril de 2007. [Citado el: 12 de Febrero de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
16. **Sánchez Suárez, José Manuel.** Adictos al trabajo. [En línea] 22 de Octubre de 2008. [Citado el: 12 de Febrero de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=migrateJSF2Facelets>.
17. Sun Microsystems. [En línea] 2006. [Citado el: 12 de Febrero de 2006-2010.] http://developers.sun.com/docs/jscreator/help/jsp-jsfel/jsf_expression_language_intro.html.
18. W3C. [En línea] 1999. [Citado el: 12 de Febrero de 2010.] <http://www.sidar.org/recur/desdi/traduc/es/xhtml/xhtml11.htm>.
19. **Allen, Dan.** *Seam in action. Manning Early Access Program. Manning Publications.* 2008. p 4, p 5, p 6.
20. *Documento de arquitectura de Software alas HIS.* s.l. : ALBET, 2008. p 69.
21. Osalt.com. [En línea] 2009. [Citado el: 12 de Febrero de 2010.] <http://www.osalt.com/es/jboss>.
22. **Devesa Navarro, Arianna y Muñoz Pérez, Reinel.** *Módulo Hospitalización del Sistema de Información Hospitalaria alas HIS.* 2008-2009.
23. Sun Developer Network. [En línea] 2010. [Citado el: 12 de Febrero de 2010.] <http://java.sun.com/products/ejb/>.

24. Idem 22. [En línea]
25. **Bauer, Christian y King, Gavin.** *Java Persistence with Hibernate.* 2007. p 67.
26. **García de Jalón, Javier, y otros.** *Aprenda Java como si estuviera en primero.* s.l. : Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España, Enero 2000.
27. **PostgreSQL Global Development Group.** About. *PostgreSQL.* [En línea] 1996 – 2010. [Citado el: 13 de Febrero de 2010.] <http://www.postgresql.org/about>.
28. —. Awards. *PostgreSQL.* [En línea] 1996-2010. [Citado el: 13 de Febrero de 2010.] <http://www.postgresql.org/about/awards>.
29. **Axentia.** Una Introducción a CMMI. [En línea] 15 de Febrero de 2010. [Citado el: 13 de Febrero de 2010.] <http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf>.
30. **OMG.** Business Process Modeling Notation Specification. [En línea] 2009. [Citado el: 13 de Febrero de 2010.] <http://www.omg.org/technology/agreement.htm>.
31. **Owen, Martin y Raj, Jog.** Introduction to the New Business Process Modeling Standard. Popkin Software. . *BPMN and Business Process Management.* [En línea] 2003. http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf. p 3.
32. **Presman, Roger S.** *Ingeniería de Software, un enfoque práctico.* McGraw-Hill. 2001.
33. *Ídem* 31. [En línea]
34. *Ídem* 32.
35. *Ídem* 32.
36. *Ídem* 32.
37. *Ídem* 32.
38. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Educación, 2000. 458.
39. *Ídem* 22. [En línea]

BIBLIOGRAFÍA

- **Allen, Dan.** *Seam in action. Manning Early Access Program. Manning Publications.* 2008. p 4, p 5, p 6.
- **Axentia.** Una Introducción a CMMI. [En línea] 15 de Febrero de 2010. [Citado el: 13 de Febrero de 2010.]
- Awards. *PostgreSQL.* [En línea] 1996-2010. [Citado el: 13 de Febrero de 2010.]
- **Bauer, Christian y King, Gavin.** *Java Persistence with Hibernate.* 2007. p 67. <http://www.postgresql.org/about/awards>.
- **Bolotchi, Ing. Hernan.** Siemens. [En línea] 2003. [Citado el: 10 de Febrero de 2010.] http://www.siemens.com.ar/sites/internet/legacy/siemed/soluciones_informaticas_para_la_gestion_hospitalaria.htm.
- Care2X. [En línea] 2002. [Citado el: 10 de Febrero de 2010.] <http://www.care2x.org/>.
- **Cueva Lovelle, Juan Manuel.** Introducción UML Lenguaje para modelar objetos. [En línea] Octubre de 1999. [Citado el: 11 de Febrero de 2010.]
- **Devesa Navarro, Arianna y Muñoz Pérez, Reinel.** *Módulo Hospitalización del Sistema de Información Hospitalaria alas HIS.* 2008-2009.
- *Documento de Arquitectura de Software alas HIS. S.l.:* ALBET. La Habana, Cuba : s.n., 2008.
- *Documento de Arquitectura de Software de Bloque Quirúrgico Oftalmológico.*
- El Modelo Cliente/Servidor. [En línea] 1994. [Citado el: 11 de Febrero de 2010.] <http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>.
- *Especificación de Requisitos de software de Bloque Quirúrgico Oftalmológico.*
- **García de Jalón, Javier, y otros.** *Aprenda Java como si estuviera en primero.* s.l. : Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España, Enero 2000.
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=migrateJSF2Facelets>.
- http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf .
- <http://gidis.ing.unlpam.edu.ar/downloads/pdfs/IntroduccionUML.PDF>.
- **Jiménez, Surelys G. Pérez y Jaime Puldón, Joan.** Mejora al proceso de obtención de requerimientos para el desarrollo de sistemas de información con el uso de técnicas BPA .[En línea] 2009. [Citado el: 22 e febrero de 2010]. CUJAE, la Habana, Cuba. [http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=7&ved=0CBcQFjAG&url=http%3A%2F%2Fwww.gestec.disaic.cu%2FFPONENCIAS2009%2Fgestec%2FCuba%2FFP46.doc&rct=j&q=bpmn%2BObjetos+de+Carriles+\(Swimlane\)+&ei=eESES7bQPMq2IAfd5YTjAQ&usg=AFQjCNHXwKLn2l89kv1zmbRSRV-LLmjXoA](http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=7&ved=0CBcQFjAG&url=http%3A%2F%2Fwww.gestec.disaic.cu%2FFPONENCIAS2009%2Fgestec%2FCuba%2FFP46.doc&rct=j&q=bpmn%2BObjetos+de+Carriles+(Swimlane)+&ei=eESES7bQPMq2IAfd5YTjAQ&usg=AFQjCNHXwKLn2l89kv1zmbRSRV-LLmjXoA)
- **Katz, Max.** *Practical Richfaces.* Apress. 2008. p 12.
- **OftalClinic.** OftalClinic. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.oftalclinic.com/>.
- **OMG.** Business Process Modeling Notation Specification. [En línea] 2009. [Citado el: 13 de Febrero de 2010.] <http://www.omg.org/technology/agreement.htm>.

- **Orozco Africano, Julio Mario.** Evaluación de la aplicación de políticas de salud en la Ciudad de Cartagena. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.eumed.net/libros/2006c/199/1d.htm>.
- Osalt.com. [En línea] 2009. [Citado el: 12 de Febrero de 2010.] <http://www.osalt.com/es/jboss>.
- **Owen, Martin y Raj, Jog.** Introduction to the New Business Process Modeling Standard. Popkin Software. . *BPMN and Business Process Management*. [En línea] 2003. http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf. p 3.
- **Pérez García, Alejandro.** JSF-Java Server Faces. *DesarrolloWeb.com*. [En línea] 21 de Febrero de 2006. [Citado el: 12 de Febrero de 2010.] <http://www.desarrolloweb.com/articulos/2380.php>.
- **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico*. La Habana, Cuba. : Félix Varela, 2005. Parte 1.
- **Ramírez Márquez, Dr. Abelardo, Castell-Florit Serrate, Dr. Pastor y Mesa, Dr. Guillermo.** *El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP)*. La Habana, Cuba : s.n., 2003-2004.
- **Ramos, Juan Alonso.** Adictos al trabajo. [En línea] 9 de Abril de 2007. [Citado el: 12 de Febrero de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
- **Reinoso, Billy.** Introducción a la Arquitectura de Software. [En línea] [Citado el: 11 de Febrero de 2010.]
- Richfaces Developer Guide. Red Hat. *RichFaces*. [En línea] 2008. <http://www.jboss.org/jbossrichfaces/docs/>. p 17.
- **Rojo, J. Oscar.** Asociación de usuarios de GNU/Linux de Castilla y León. [En línea] 2003. [Citado el: 11 de Febrero de 2010.] <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>.
- **SIGHO.** Sistema de Información para la gerencia Hospitalaria. [En línea] 2006. [Citado el: 10 de Febrero de 2010.] http://www.ssn.gob.mx/html/sigho/intro_sigho.html.
- **Sánchez Suárez, José Manuel.** Adictos al trabajo. [En línea] 22 de Octubre de 2008. [Citado el: 12 de Febrero de 2010.]

GLOSARIO DE TÉRMINOS

AJAX: Técnica para el desarrollo Web que posibilita la creación de aplicaciones interactivas.

API (Application Programming Interface): Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

Bloque Quirúrgico: Se define desde el punto de vista estructural y organizativo, como el espacio en el que se agrupan todos los quirófanos, con los locales de apoyo, instalaciones y equipamiento necesarios para realizar los procedimientos quirúrgicos previstos, por parte del equipo multiprofesional que ofrece asistencia multidisciplinar, que garantiza las condiciones adecuadas de seguridad, calidad y eficiencia, para realizar la actividad quirúrgica.

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

ICEFaces: Biblioteca de componentes JSF Ajax.

Post-Operatorio: Período de cuidados que comienza cuando el paciente termina la cirugía. Tiene el propósito de complementar las necesidades psicológicas y físicas directamente después de la cirugía.

Pre-Operatorio: Período que abarca el espacio de tiempo comprendido desde que el paciente es informado de que su problema de salud ha de ser tratado quirúrgicamente, acepta este tratamiento y se fija una fecha para la intervención quirúrgica hasta que el enfermo es trasladado al área quirúrgica.

Sistema de Información Hospitalaria (HIS): Son sistemas software que trabajan acoplados y permiten la informatización de los distintos servicios de las instituciones de salud.

Trans-Operatorio: El tiempo que transcurre desde que el paciente está en la sala de operaciones hasta que son retirados los campos quirúrgicos.

XML: (Extensible Markup Language): Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium.