

Universidad de las Ciencias Informáticas

Facultad 7



Tema: Análisis, diseño e implementación de los servicios oftalmológicos

Título: Servicios Oftalmológicos Oculoplastia y Baja Visión del Bloque Quirúrgico Oftalmológico

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores: Sarahí Palacio Rondón
Ernesto Sánchez Escobar

Tutores: Ing. Suleydis Suárez Serpa
Ing. Pedro Ernesto Salas Oliva

Ciudad de La Habana, Julio de 2008

“Año 50 de la Revolución”

*“Hay una fuerza motriz más poderosa que el vapor,
la electricidad y la energía atómica:
La voluntad.”*

Albert Einstein.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 2 días del mes de julio del año 2008.

Sarahí Palacio Rondón

Firma del Autor

Ernesto Sánchez Escobar

Firma del Autor

Ing. Suleydis Suárez Serpa

Firma del Tutor

Ing. Pedro E. Salas Oliva

Firma del Tutor

DATOS DE CONTACTO

Tutores:

Ing. Suleydis Suárez Serpa, recién graduada de la Universidad de las Ciencias Informáticas y actual profesora de la Facultad 7.

e-mail ssuarez@uci.cu

Ing. Pedro Ernesto Salas Oliva, recién graduado de la Universidad de las Ciencias Informáticas y actual profesor de la Facultad 7.

e-mail psalas@uci.cu

AGRADECIMIENTOS

A todos aquellos que de una forma u otra han marcado una huella en nuestras vidas y han permitido la realización de este trabajo: muchísimas gracias de todo corazón.

A nuestros tutores, Suleydís y Pedro por su ayuda, apoyo y por todo el tiempo dedicado.

A la Revolución Cubana y en especial al Comandante en Jefe Fidel Castro por hacer posible este sueño.

De Ernesto:

A mis queridos padres por su apoyo constante y confianza en mí. Por su ejemplo, por estar siempre ahí dándome consejos, por su infinito amor y comprensión. Gracias mami. Gracias papi.

A Lari por tantos momentos felices, tantas risas y amor, gracias mi hermanita y sobre todo, gracias por existir.

De todo corazón a mi tía Patricia por ser mi ejemplo a seguir, al tío Carlos y a mi primita María Carla por su apoyo incondicional.

A mis abuelos y demás familiares por sus consejos y por creer en mí.

A Mami-tía por todo lo que ha hecho por mí durante toda mi vida.

A mis amigos y compañeros de aula por estar siempre compartiendo y dándome ánimos para seguir adelante.

De Sarahí:

A mi papá por ser un gran padre y quererme tanto, además de mostrarme siempre el camino correcto a seguir y llenarme de su inmenso cariño.

A mi mamá por ser una gran madre y quererme tanto y siempre apoyarme en todo, además de enseñarme a luchar y nunca dejarme vencer por las dificultades.

A mis primas y en especial a Adriana por estar a mi lado en momentos difíciles y darme fuerzas con la magia de su ternura y cariño.

A mis segundas madres Liudmila y Dayami, por su gran cariño, apoyo y acogerme como una hija más.

A mi familia y en especial mi hermana Idania por estar ahí siempre cuando la necesite.

A Arellys, Ariel y familia por su ayuda, apoyo, por los momentos gratos y especiales que me dieron, quedando por siempre en mi vida.

A todas mis amistades y en especial a Maylin, Yadira, Dinaibys, Dailyn y Karina que han pasado junto a mí, momentos inolvidables, que han marcado mi vida para siempre.

DEDICATORIA

De Sarahi:

A mi madre Lilia Rondón y a mi padre Israel Palacio, por haber confiado en mí, por su amor, cariño y toda una vida de constante dedicación. Sin ellos siempre a mi lado, no sería esto posible.

A mi familia en general.

De Ernesto:

Dedico esta tesis a mis padres Mario Sánchez y Marta Escobar por su confianza, amor y su comprensión.

A mi hermanita del alma Laritza que me ha dado fuerzas para seguir adelante.

A mis tíos Patricia y Carlos por su apoyo incondicional.

A mis abuelos, a toda mi familia, si no fuera por ellos mi sueño no se habría cumplido.

A todos los que han contribuido de una forma u otra en mi formación y en la realización de este trabajo.

RESUMEN

Actualmente, la gestión de la información de los procesos que se realizan en los Servicios oftalmológicos Oculoplastia y Baja Visión del Bloque Quirúrgico Oftalmológico se desarrollan de forma manual. Lo que provoca que: se introduzcan errores, haya pérdida y deterioro de información, se alargue el período de actualización de la misma, exista redundancia y no se obtenga con rapidez. Por lo que se hace necesario el análisis, diseño e implementación de un sistema que permita el almacenamiento, la gestión y el control de la información.

Para el desarrollo del sistema se siguieron los pasos que propone el Rational Unified Process (RUP), para los flujos de trabajo: Modelación del Negocio, Requisitos, Análisis y Diseño. Se utilizó el Lenguaje Unificado de Modelado (UML) como soporte a la metodología y Rational Rose para el modelado visual. Para el modelado de la Base de datos se utilizó Case Studio y PostgreSQL 8.2 como gestor. Como herramienta para la implementación Visual Studio 2005 y como lenguaje de programación del lado del servidor: ASP.net

Con el desarrollo de esta propuesta, se espera lograr que los servicios oftalmológicos Oculoplastia y Baja Visión tengan a su alcance aplicaciones capaces de realizar los procesos que a diario se gestionan en estos. Logrando con la automatización de los procesos y la digitalización de la información; mayor capacidad organizativa, alto grado de confiabilidad y seguridad. Así como un aumento de la calidad de los servicios oftalmológicos a todas aquellas personas que hacen uso de los mismos.

PALABRAS CLAVE

Oculoplastia, Baja Visión, Intervención quirúrgica, paciente, examen.

ÍNDICE

AGRADECIMIENTOS.....	III
DEDICATORIA.....	IV
RESUMEN.....	V
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Conceptos básicos relacionados con el dominio del problema.....	5
1.2 Análisis de las posibles tecnologías y herramientas de desarrollo a utilizar.....	6
1.3 Sistemas automatizados existentes vinculados al campo de acción.....	30
1.4 Justificación de la propuesta de solución.....	34
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	36
2.1 Objetivos estratégicos de la organización.....	36
2.2 Flujo actual de los procesos.....	37
2.3 Objeto de automatización.....	39
2.4 Propuesta de sistema.....	39
2.5 Actores y trabajadores del Negocio.....	40
2.6 Requerimientos del sistema.....	47
2.7 Actores del Sistema.....	50
2.8 Definición de los casos de uso del sistema.....	50
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	56
3.1 Diagramas de clases de análisis.....	56
3.2 Modelo de Diseño.....	57
3.3 Patrones de Diseño.....	61
3.4 Subsistemas que interactúan.....	62
3.5 Diagramas de Secuencia.....	63
3.6 Descripción de las nuevas clases u operaciones necesarias.....	67
3.7 Diseño de la BD.....	86
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	90
4.1 Diagrama de despliegue.....	90
4.2 Diagrama de componentes.....	91
4.3 Modelo de pruebas.....	92
CONCLUSIONES.....	95
RECOMENDACIONES.....	96
REFERENCIAS BIBLIOGRAFICAS.....	97
BIBLIOGRAFÍA.....	100
ANEXOS.....	102
GLOSARIO:.....	109

Índice de Tablas

Tabla 1: Actores del Negocio.	41
Tabla 2: Trabajadores del Negocio.	41
Tabla 3: Descripción textual del Caso de Uso Realizar Consulta Oftalmología Especializada.	43
Tabla 4: Descripción textual del Caso de Uso Realizar Consulta Anestesia.	44
Tabla 5: Descripción textual del Caso de Uso Realizar Intervención Quirúrgica Oftalmológica.	45
Tabla 6: Descripción textual del Caso de Uso Realizar Planificación Quirúrgica.	46
Tabla 7: Descripción textual del Caso de Uso Aprobar Cirugía.	47
Tabla 8: Actores del Sistema.	50
Tabla 9: Casos de Uso por Ciclos.	53
Tabla 10: Caso de Uso del Sistema: Gestionar Consulta Especialista Oftalmológica en Oculoplastia.	54
Tabla 11: Caso de Uso del Sistema: Gestionar Consulta Especialista Oftalmológica en Baja Visión... ..	55
Tabla 12: Clase entidad PtoisPalpebral.	67
Tabla 13: Clase entidad Dacricistitis.	68
Tabla 14: Clase entidad MalposicionPalpebral.	69
Tabla 15: Clase entidad OjoSeco.	69
Tabla 16: Clase entidad Refraccion.	70
Tabla 17: Clase entidad ExamenRefraccionDinamica.	71
Tabla 18: Clase entidad ExamenRefraccionIndicada.	72
Tabla 19: Clase entidad ExamenLensometro.	72
Tabla 20: Clase entidad ExamenesOftalmologicos.	73
Tabla 21: Clase entidad AntecedentesPatologicosOftalmologicos.	73
Tabla 22: Clase entidad ExamenesAnexos.	73
Tabla 23: Clase entidad ExamenFondoOjo.	74
Tabla 24: Clase entidad ExamenMotilidadOcular.	74
Tabla 25: Clase entidad ExamenReflejoPupilar.	74
Tabla 26: Clase entidad ExamenSegmentoAnterior.	75
Tabla 27: Clase entidad ExamenVitreo.	75
Tabla 28: Clase entidad ParametroSegmentoAnterior.	75
Tabla 29: Clase entidad AntecedenteOftalmologico.	76
Tabla 30: Clase entidad HabitoToxico.	76
Tabla 31: Clase entidad EPDacricistitis.	76

Tabla 32: Clase entidad EPOjoSeco.	77
Tabla 33: Clase entidad EPptosisPalpebral.	77
Tabla 34: Clase entidad EPMalposicionPalpebral.	77
Tabla 35: Clase entidad EPRefraccion.	78
Tabla 36: Clase entidad EPLensometro.	78
Tabla 37: Clase entidad EPRefraccionIndicada.	78
Tabla 38: Clase entidad EPRefraccionDinamica.	79
Tabla 39: Clase controladora DacricistitisRepositorio.	79
Tabla 40: Clase controladora DacricistitisSelectionFactory.	79
Tabla 41: Clase controladora DacricistitisInsertFactory.	80
Tabla 42: Clase controladora DacricistitisUpdateFactory.	80
Tabla 43: Clase controladora DacricistitisDeleteFactory.	80
Tabla 44: Clase controladora EPDacricistitisNegocio.	81
Tabla 45: Clase interfaz bqo_ConsultaOftalmologiaOculoplastia.	82
Tabla 46: Clase interfaz ExamenesEspOculoplastia.	83
Tabla 47: Clase interfaz bqo_ConsultaOftalmologiaBajaVision.	85
Tabla 48: Clase Interfaz ExamenesEspBajaVision.	86
Tabla 49: Caso de prueba CU Gestionar Hoja del Especialista Oftalmológico en Baja Visión.	93
Tabla 50: Caso de prueba CU Gestionar Hoja del Especialista Oftalmológico en Oculoplastia.	94

Índice de Figuras

Figura 1: Arquitectura del Framework .NET. 18

Figura 2: Casos de Uso del Negocio..... 42

Figura 3: Diagrama de Paquetes del Sistema. 51

Figura 4: Diagrama de Paquetes ANUNCIAR_CIRUGÍA. 51

Figura 5: Diagrama del Caso de Uso Gestionar Hoja del Especialista Oftalmológico..... 52

Figura 6: Diagrama de clases de análisis del CU Gestionar Hoja del Especialista Oftalmológico en Oculoplastia. 57

Figura 7: Diagrama de clases de análisis del CU Gestionar Hoja del Especialista Oftalmológico en Baja Visión..... 57

Figura 8: Diagrama de Clases de Diseño Gestionar Hoja del Especialista Oftalmológico en Oculoplastia..... 59

Figura 9: Diagrama de Clases de Diseño Gestionar Hoja del Especialista Oftalmológico en Baja Visión. 60

Figura 10: Diagrama de subsistemas relacionados con el Modulo Bloque Quirúrgico Oftalmológico. .. 62

Figura 11: Diagrama de Secuencia del CU Crear Hoja del Especialista Oftalmológico en Oculoplastia. 63

Figura 12: Diagrama de Secuencia del CU Modificar Hoja del Especialista Oftalmológico en Oculoplastia. 64

Figura 13: Diagrama de secuencia del CU Crear Hoja del Especialista Oftalmológico en Baja Visión. 65

Figura 14: Diagrama de secuencia del CU Crear Hoja del Especialista Oftalmológico en Baja Visión. 66

Figura 15: Diagrama Entidad Relación primera parte..... 87

Figura 16: Diagrama Entidad Relación segunda parte. 88

Figura 17: Diagrama de Despliegue..... 91

Figura 18: Diagrama de Componentes. 92

Figura 19: Diagrama de actividades del CUN Realizar Consulta Oftalmología Especializada. 102

Figura 20: Diagrama de actividades del CUN Realizar Planificación Quirúrgica..... 103

Figura 21: Diagrama de actividades del CUN Realizar Intervención Quirúrgica. 104

Figura 22: Diagrama de actividades del CUN Realizar Aprobación Cirugía. 105

Figura 23: Diagrama de actividades del CUN Realizar Consulta Anestesia. 106

Figura 24: Interfaz de usuario Consulta Especializada Oftalmológica en Oculoplastia. 107

Figura 25: Interfaz de usuario Exámenes Específicos en Oculoplastia. 108

INTRODUCCIÓN

El futuro de la humanidad dependerá en gran medida del potencial humano, de la gestión de la producción y de los conocimientos que se alcancen. La informática en sus diferentes manifestaciones, tiene asegurado un papel protagónico en este futuro.

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.

Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva. La Informatización en Cuba se basa en los principios y valores fundamentales del sistema social, con una marcada tendencia a defender las conquistas que se han logrado.

Durante los últimos años un grupo de instituciones cubanas han desarrollado sistemas encaminados a lograr determinados niveles de informatización de la salud. Estas soluciones carecían de integración y de una definición generalizable, además de que no existían los recursos tecnológicos necesarios para su ejecución en el Sistema Nacional de Salud.

Actualmente el Ministerio de Salud Pública (MINSAP) ha definido a la informatización como una de sus prioridades y ha convocado para ello a un grupo de instituciones propias de sector, del Ministerio de Informática y Comunicaciones y de otros organismos de la administración central del estado, para definir de conjunto la estrategia a desarrollar. En algunos casos se ha tomado como punto de partida sistemas ya desarrollados en el país en el marco de aquella primera estrategia de desarrollo.

El eje fundamental y centro del proceso de informatización del sector lo constituye el paciente, el cual será el principal beneficiado, ya que el sector contará con aplicaciones de calidad, que brinden consistencia de la información, lo que incrementará la efectividad y eficiencia de los procesos relacionados con la salud, que en última instancia gravitarán en un incremento continuo y sostenido de la calidad en la atención médica.

Hoy día se trabaja en el desarrollo de un grupo de aplicaciones básicas para la informatización del sector de la salud. En su desarrollo e implementación participan diferentes empresas del Ministerio de la Informática y Comunicaciones como Desoft, Softel, PcMax, Sys, INFOMED, CEDISAP y la Universidad de Ciencias Informáticas (UCI).

Actualmente, el pueblo cubano está inmerso en la informatización de la sociedad, apoyados en instituciones como la UCI, que tiene entre sus tareas primordiales la informatización de la esfera de la salud, tarea que ha sido asignada a la facultad 7.

En esta facultad, la producción se encuentra dividida en áreas temáticas, una de las cuales es Gestión Hospitalaria, dicha área temática tiene como tarea desarrollar sistemas que permitan la gestión de la información en los diferentes departamentos de un hospital, entre los que se encuentra Bloque Quirúrgico Oftalmológico, con los servicios que lo conforman.

Hoy en día se hace necesario mejorar la calidad de los servicios quirúrgicos oftalmológicos que se brindan, así como agilizar el control de la información que se gestiona. Esta situación se agudiza debido al gran volumen de pacientes que son atendidos como parte del proyecto de colaboración médica internacional llevado a cabo por Cuba, conocido como Misión Milagro por lo que priorizar este servicio adquiere gran relevancia.

En el Hospital Oftalmológico Ramón Pando Ferrer, se desarrolló un sistema que permite gestionar y administrar parte de la información generada durante los procesos por los que atraviesa un paciente al ser atendido. Debido a los cambios realizados en los hospitales oftalmológicos, actualmente este sistema resulta primitivo pues no responde a los intereses de los usuarios, sólo ofrece una solución particular a este hospital y no responde a todos los servicios oftalmológicos que se prestan en el mismo.

Lo antes expuesto trae como consecuencia que el proceso de atención a los pacientes en los servicios de Oculoplastia y Baja Visión del Bloque Quirúrgico Oftalmológico se realiza manualmente en su totalidad, lo que conlleva a que haya demora en la recepción de los datos y el nivel de error en estos sea considerable además de que dificulta la entrega de reportes y provoca que el desarrollo de estos procesos sea más lento, provocando que la espera de los pacientes sea mayor. Debido a esto, los servicios en el hospital en ocasiones no son los esperados.

El proceso manual tiene muchas desventajas para los trabajadores de este hospital, pues impide la tenencia de registros actualizados a causa de la demora del llenado de datos y a los errores en general que se pueden cometer durante este procedimiento. Mencionar que la forma de gestionar actualmente la documentación generada por este departamento aumenta considerablemente el volumen de información recogida en formato duro creando en los hospitales enormes archivos con información repetida que pueden deteriorarse o en el peor de los casos, extraviarse.

Además el déficit de herramientas que permitan el almacenamiento de datos necesarios para generar estadísticas importantes sobre el comportamiento de la actividad quirúrgica oftalmológica en la base, afecta en la toma de decisiones para las futuras planificaciones tanto a nivel primario como a niveles superiores ya que esta situación provoca que los resultados finales en ocasiones no reflejen la realidad concreta.

Con los métodos empleados actualmente el trabajo desarrollado con los datos es lento y en ocasiones se necesitan con rapidez por lo que sería factible obtener con gran velocidad la información en muy poco tiempo.

Basado en lo antes expuesto se tiene como **problema científico** de la investigación: ¿Cómo facilitar la gestión de la información referente a los Servicios Quirúrgicos Oculoplastia y Baja Visión que tienen lugar en los hospitales oftalmológicos cubanos?

En correspondencia con el problema el **objeto de estudio** lo constituyen los procesos de gestión de la información que se desarrollan en los hospitales cubanos.

El **campo de acción está** centrado en la informatización de los procesos de gestión de la información del Bloque Quirúrgico Oftalmológico que tienen lugar en los hospitales cubanos.

Como **objetivo general** de este proyecto, se define: Implementar un sistema informático que facilite la gestión de la información referente a los servicios oftalmológicos Oculoplastia y Baja Visión del Bloque Quirúrgico Oftalmológico.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas**:

- Investigar el análisis realizado sobre el funcionamiento del bloque quirúrgico oftalmológico general en los hospitales.
- Seleccionar detalladamente las herramientas y metodologías a utilizar.
- Definir las herramientas a utilizar para el modelado y la implementación del sistema.
- Diseñar un sistema informático utilizando la metodología de ingeniería del software seleccionada.
- Implementar los servicios oftalmológicos propuestos.
- Obtener los artefactos necesarios de la metodología RUP.
- Analizar la estrategia de integración de la solución con otros módulos o sistemas.

- Realizar una valoración de los resultados y hacer propuestas de iteración en el diseño.

El documento está estructurado de la manera siguiente:

Capítulo 1: Fundamentación Teórica: Estudio preliminar de los Sistemas de Gestión Oftalmológica. Tecnologías y herramientas de desarrollo a utilizar.

Capítulo 2: Características del Sistema: Se muestra la modelación del negocio realizada, se recogen los requerimientos que debe cumplir la aplicación en forma de casos de uso, así como la modelación del sistema a desarrollar.

Capítulo 3: Análisis y Diseño: Se modelan un grupo de artefactos de los flujos de trabajo de Análisis y Diseño. Así como los patrones de diseño utilizados.

Capítulo 4: Implementación: Se modelan un grupo de artefactos del flujo de Implementación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se presentan los conceptos básicos relacionados con el campo de acción facilitando la comprensión del problema a resolver así como la realización de un estudio de sistemas oftalmológicos existentes a nivel internacional y nacional. Se hace un estudio del estado del arte de las tecnologías, metodologías y herramientas a utilizar en el desarrollo del trabajo.

1.1 Conceptos básicos relacionados con el dominio del problema.

El sentido de la vista, es el que permite percibir sensaciones luminosas y captar el tamaño, la forma y el color de los objetos, así como la distancia a la que se encuentran. Estas sensaciones llegan a través de los ojos, órganos encargados de la visión.

Existen patologías que aquejan al ser humano condenado a vivir en tinieblas durante varias décadas que sólo pueden ser curadas a través de una de las prácticas más radicales de la medicina, la cirugía oftalmológica. Desde tiempos remotos el hombre realizaba ciertas técnicas de curación que podrían definirse como quirúrgicas y quienes la practicaban recibieron el nombre de cirujanos.

Definiendo de forma sencilla la **cirugía**, también nombrada como **intervención quirúrgica**, no es más que la ejecución de diversos actos curativos sobre el cuerpo vivo, como extirpar, amputar, reseca, implantar, corregir y coser, órganos, miembros o tejidos, con ayuda de los instrumentos adecuados.[1] Esta es realizada en una misma oportunidad, por uno o más cirujanos en el pabellón o quirófano, generalmente bajo anestesia y para un tratamiento determinado.

Un **bloque quirúrgico** es definido como el área centralizada en la que se realiza toda la actividad quirúrgica del hospital, es decir, se realizan todas las cirugías programadas correspondientes a pacientes ya sean hospitalizados, ambulatorios o casos de urgencia.[2]

La **especialidad de oftalmología** es la que se ocupa de la prevención y el tratamiento, tanto médico como quirúrgico, de todo lo relacionado al ojo y sus anexos (párpados, vías lagrimales, órbita, etc.) como pueden ser Oculoplastia, Baja Visión, Miopía, Hipermetropía, Cataratas, Estrabismo, entre otros.
[3]

Se puede definir al **bloque quirúrgico oftalmológico** como el área centralizada donde se realizan todas las actividades quirúrgicas correspondientes a la especialidad de oftalmología.

La necesidad de tener un documento que actuara como recordatorio para el manejo clínico del paciente, y permitiera el análisis retrospectivo del quehacer de los profesionales sanitarios, provoca a partir de la segunda mitad del siglo XX el surgimiento de la *historia clínica*, único documento válido desde el punto de vista clínico y legal.

La **historia clínica** se define como el conjunto de documentos que contienen los datos, valoraciones e informaciones de cualquier índole sobre la situación y la evolución clínica de un paciente a lo largo del proceso asistencial.[4] Su principal función es asistencial, ya que permite reflejar el seguimiento de los pacientes durante toda su vida. Por eso conformando la historia clínica se encuentran la **hoja de anestesia** que contiene la hoja del pre-operatorio o de consulta pre-anestésica, y además las hojas del trans-operatorio y post-operatorio.

La **hoja pre-operatoria** es el documento donde se registra el estado general del paciente previo a la intervención. En la **hoja trans-operatoria** se recoge todo lo acontecido durante la intervención quirúrgica y los medicamentos que son suministrados de forma controlada. Estos documentos son certificados por el anestesiólogo o el cirujano según corresponda.

1.2 Análisis de las posibles tecnologías y herramientas de desarrollo a utilizar.

Las especializaciones del software están impulsando cada vez más el desarrollo de las tecnologías y metodologías que se utilizan en este propósito, a nivel mundial existen herramientas que permiten la realización de grandes sistemas en tiempos breves y con gran calidad debido a las facilidades que implementan.

Para desarrollar un software se deben tener en cuenta ciertos parámetros como metodologías a usar, lenguajes de programación, gestores de bases de datos, patrones arquitectónicos y de diseño entre otros. Como consecuencia a continuación se presenta el resumen realizado durante la investigación, y la propuesta tecnológica realizada por los arquitectos del proyecto.

1.2.1 Aplicación web

Una aplicación web es un sistema web que posibilita a los usuarios ejecutar la lógica de negocio o modificar el estado del negocio a través de un navegador y generar contenidos dinámicos mediante las tecnologías existentes. [5]

1.2.1.1 Navegadores Web

La difusión de los distintos sistemas a través de Internet se logra mediante los navegadores Web. Un navegador Web o *browser* es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet. La funcionalidad básica de un navegador Web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados.

Los documentos pueden estar ubicados en la computadora donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos. Existen diferentes navegadores Web tales como Mozilla, Windows Internet Explorer, Firefox, Netscape, Opera, Konqueror, entre otros.

1.2.1.2 Servidores web.

Un servidor web es un programa que sirve datos en forma de páginas web, hipertextos o páginas HTML (HyperText Markup Language), la comunicación de estos datos entre cliente y servidor se hace por medio de un protocolo, esencialmente del protocolo http. [6]

1.2.1.3 Internet Information Server

Internet Information Server (IIS) es un servidor Web propietario de funciones completas que posibilita la creación de aplicaciones Web y servicios Web XML. La arquitectura de IIS ha sido completamente reconstruida, con un nuevo modelo de proceso de tolerancia a errores que mejora significativamente la confiabilidad de las aplicaciones y los sitios Web.

Proporciona un entorno de servidor Web más inteligente y confiable para lograr la confiabilidad óptima. Este nuevo entorno incluye la supervisión del estado de las aplicaciones y el reciclaje automático de aplicaciones, además de una seguridad y capacidad de administración significativamente mejorada. Las mejoras de seguridad incluyen cambios tecnológicos y de procesamiento de solicitudes. Además, se ha mejorado la autenticación y la autorización.

1.2.1.4 Servidor Web Apache

Es un servidor Web de tecnología Open Source (código abierto) sólido, que ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad, siendo actualmente el más usado a nivel mundial.

El servidor Apache es un *software* que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo.

Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Este se caracteriza por correr en una multitud de sistemas operativos, lo que lo hace prácticamente universal. Es altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.

El uso de Apache como servidor, es precisamente a la necesidad de garantizar un servidor robusto, que pueda ser utilizado en varios sistemas operativos, que garantice la disponibilidad del código fuente además de las ventajas técnicas, como seguridad o rendimiento que ofrece y la excelente integración con el lenguaje de programación web Asp.net y sistema gestor de base de datos PostgreSQL.

Ventajas del Servidor Web Apache

- Fundamentalmente corre sobre una multitud de plataformas y Sistemas Operativos.
- Apache ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente se está instalando.
- Apache es un servidor Web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.
- Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor.
- Otra particularidad propia de Apache y que está muy ligada a su pensamiento y filosofía libre, es que al ser tan popular y utilizado, es posible encontrar gran cantidad de documentos, ejemplos y ayuda en internet en todos los idiomas.

1.2.2 Arquitecturas de software

El diseño de más alto nivel de la estructura de un sistema, programa o aplicación, es la arquitectura del software que tiene la responsabilidad de:

- Definir los módulos principales.
- Definir las responsabilidades que tendrá cada uno de estos módulos.

- Definir la interacción que existirá entre dichos módulos.
- Control y flujo de datos.
- Secuenciación de la información.
- Protocolos de interacción y comunicación.
- Ubicación en el hardware.

1.2.2.1 Arquitectura cliente-servidor. [7]

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

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

Este modelo permite descentralizar el procesamiento y recursos, sobre todo, de cada uno de los servicios y de la visualización de la interfaz gráfica de usuario lo que conlleva a que ciertos servidores estén dedicados solo a una aplicación determinada y por lo tanto ejecutarla en forma eficiente.

Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor.[8]

Ventajas que ofrece la arquitectura cliente-servidor.

- *Centralización del control:* los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- *Escalabilidad:* se puede aumentar la capacidad de clientes y servidores por separado. Mayor productividad en las estaciones de trabajo programables con interfaces gráfica de usuario, que permite acceder e integrar aplicaciones muy intuitivamente. Disponibilidad de potencia de cálculo a nivel personal, sin la responsabilidad del mantenimiento del sistema y del software de aplicaciones y disponibilidad de herramientas de desarrollo fáciles de usar, reduciendo la dependencia del departamento informático.
- Un mejor ajuste del Sistema de Información a la organización y a los procesos de negocio.
- Mayor protección de activos informáticos e integración de los sistemas y aplicaciones ya existentes.
- Acceso a la información cuándo y dónde la necesitan los usuarios.
- Libertad para migrar a plataformas de sistemas alternativos y usar servidores especializados.
- Un entorno de utilización más sencillo, que proporciona una mayor productividad. A largo plazo, las interfaces gráficas de usuario reducen los costos asociados a educación y formación de los usuarios.

1.2.2.2 Arquitectura orientada a servicios (SOA).

Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture o SOA), es una arquitectura de aplicación en la cual todas las funciones se definen como servicios independientes con interfaces invocables bien definidas, que pueden ser llamadas en secuencias definidas para formar procesos de negocios. SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa.[9]

Se basa en la independencia de plataformas de hardware, de sistemas operativos y de lenguajes de programación. Fortaleciendo la reutilización de los sistemas actuales que se construyeron y se utilizaron durante años; y crea un ambiente en el que los negocios y la tecnología de la información pueden interactuar entre sí.

Ventajas de SOA.

Es a la medida del cliente, absolutamente modular, más flexible, además facilita la extensión de la estructura que existe hoy. Por lo que el lema “usted para mejorar debe cambiarlo todo”, es algo que realmente ya no debe usarse. SOA permite expresar pedazos o partes de cosas que se tengan hoy, como servicio, y asociarlas de tal manera que permitan solucionar los problemas de negocio.[10]

1.2.3 Servicios Web y XML. SOAP (Protocolo de Acceso Simple a Objetos)

En el caso del manejo de datos se usa ampliamente lo que se conoce como Servicio Web (en inglés Web Service) que no es más que una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Ventajas de los servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.2.3.1 XML

XML, sigla en inglés de eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo.

Es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función

principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. [11]

Ventajas de XML

- **Comunicación de datos.** Si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- **Migración de datos.** Si se trabaja en formato XML sería muy sencillo mover datos de una base de datos a otra.
- **Aplicaciones Web.** Con XML hay una sola aplicación que maneja los datos y para cada navegador se puede tener una hoja de estilo o similar para aplicarle el estilo adecuado.

1.2.3.2 SOAP

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

Usa el código fuente en XML. Esto es una ventaja ya que facilita su lectura por parte de humanos, pero también es un inconveniente dado que los mensajes resultantes son más largos. El intercambio de mensajes se realiza mediante tecnología de componentes.

Es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí. [12]

1.2.3.3 XSL

XSL (siglas de Extensible Stylesheet Language, expresión inglesa traducible como "lenguaje extensible de hojas de estilo") es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico. [13]

Transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página Web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Actualmente, es muy usado en la edición Web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad. [14]

1.2.4 Lenguajes de Programación Web.

Un lenguaje de programación se compone de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que más tarde serán interpretadas. Estos se clasifican según el paradigma que usan en procedimentales, orientados a objetos, funcionales, lógicos, híbridos entre otras. Ejemplos de estos los constituyen los lenguajes: PHP, Prolog, ASP, PASCAL, Java, JavaScript.

Dentro de estos, se encuentran los lenguajes de programación orientados al Web, los cuales se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor.

Entre los lenguajes del lado del servidor, se destacan por sus características sobresalientes PERL, ASP, PHP, Java, JSP, etc. Estos desarrollan la lógica de negocio dentro del servidor, además de encargarse de los accesos a los distintos sistemas de gestión de bases de datos.

Dentro de los lenguajes que trabajan del lado del cliente se encuentran el JavaScript, XSLT y el Visual Basic Script, estos dos últimos al combinarse con el HTML forman lo que se conoce como DHTML, es decir, salida estándar dinámica o HTML dinámico.

1.2.4.1 ASP

Es un conjunto de tecnologías, distribuida por Microsoft, para el desarrollo de páginas web generadas dinámicamente. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP). Incorpora un modelo declarativo a la programación web, los controles de servidor funcionan en una página web simplemente declarándolos.

El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT. Incluso en sistemas Linux se puede utilizar las ASP si se instala un componente denominado Chilisoft, aunque parece claro que será mejor trabajar sobre el servidor web para el que está pensado: Internet Information Server. [15]

1.2.4.2 PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. [16]

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. Algunas de las más importantes capacidades de PHP son:

- Es multiplataforma, es decir, puede ser usado en cualquiera de los sistemas operativos del mercado actual.
- Para desarrollar en PHP no se requiere tener grandes capacidades de hardware. Luego en el caso de los servidores, una aplicación en PHP no requiere tanta memoria de máquina como podría requerir una aplicación en Java con sus servidores de aplicaciones que podrían requerir hasta varios procesadores y varios Gigas de memoria RAM.
- PHP tiene una de las comunidades más grandes en Internet, por lo que es fácil encontrar ayuda, documentación, artículos, noticias, y demás recursos.

Desventaja:

- La legibilidad de código puede verse afectada al mezclar sentencias HTML y PHP.

1.2.4.3 PERL

Acrónimo de Practical Extracting and Reporting Language, actualmente es utilizado en múltiples tareas dentro de las cuales se incluye la administración de sistemas , desarrollo web, programación en red , el desarrollo de GUI (interfaz gráfica de usuario), entre otras. [17]

Este es un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Una diferencia fundamental de Perl con respecto a los otros lenguajes es que no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible. [18]

1.2.4.4 XSLT

XSLT es el acrónimo de Extensible Stylesheet Language Transformation, es un lenguaje que su utiliza en el proceso de convertir documentos XML en otros e incluso a formatos que no son XML, puede convertir un documento XML que obedezca a un DTD a otro que obedezca otro diferente, un

documento XML bien formado a otro que siga un DTD, o, lo más habitual, convertirlo a "formatos finales", tales como WML. [19]

Los programas XSLT están escritos en XML, y generalmente, se necesita un procesador de hojas de estilo, o *stylesheet processor* para procesarlas, aplicándolas a un fichero XML. Definiéndose así un estilo de programación diferente al resto de los que ofrecen los restantes lenguajes de programación.

1.2.4.5 Javascript [20]

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, siendo uno de los más utilizados con este fin, brindando muchas posibilidades, ya que permite la programación de pequeños scripts, pero también de programas más grandes orientados a objetos con funciones, estructuras de datos, entre otras.

1.2.4.6 JSP

JSP (Java Server Page) es una tecnología orientada a crear páginas web con programación en Java.

Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP se pueden escribir con el editor HTML/XML habitual. [21]

1.2.5 Técnicas de programación.

1.2.5.1 Programación no Estructurada.

Este tipo de programación está basada en una secuencia de instrucciones modificando los datos globales en el transcurso de todo el programa. Donde los saltos y el fin de programa no seguían ninguna estructura. Los saltos podían apuntar a cualquier punto del código, lo que ocasionaba que el algoritmo terminara siendo un ovillo indescifrable. Tampoco se podía saber cuándo terminaba.

1.2.5.2 Programación Procedimental.

La programación procedimental es un tipo de programación estructurada en donde el código se divide en porciones llamadas "procedimientos" o "funciones".

1.2.5.3 Programación Modular.

La programación modular está basada en la técnica de diseño descendente, consiste en dividir el problema original en diversos sub-problemas que se pueden resolver por separado, para después recomponer los resultados y obtener la solución al problema.

1.2.5.4 Programación Orientada a Objetos.

La Programación Orientada a Objetos (*POO* u *OOP* según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (datos), *comportamiento* (procedimientos o *métodos*) e identidad (propiedad del objeto que lo diferencia del resto).

Expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulo más fáciles de escribir, mantener y reutilizar. De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos).

A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

1.2.5.5 Programación Estructurada.

Se refiere a un conjunto de técnicas que han ido evolucionando. Estas técnicas aumentan considerablemente la productividad del programa reduciendo el tiempo requerido para escribir, verificar, depurar y mantener los programas. La programación estructurada utiliza un número limitado de estructuras de control que minimizan la complejidad de los problemas y que reducen los errores. Ésta incorpora entre otros elementos: el diseño descendente, recursos abstractos y estructuras básicas.

Se basa en una metodología de desarrollo de programas llamada refinamientos sucesivos: se plantea una operación como un todo y se divide en segmentos más sencillos o de menor complejidad. Una vez terminado todos los segmentos del programa, se procede a unificar las aplicaciones realizadas por el pool de programadores. Si se ha utilizado adecuadamente la programación estructurada, esta

integración debe ser sencilla y no presentar problemas al integrar la misma, y de presentar algún problema, será rápidamente detectable para su corrección. [22]

Ventajas de la programación estructurada.

- Los programas son más fáciles de entender. Un programa estructurado puede ser leído en secuencia, de arriba hacia abajo, sin necesidad de estar saltando de un sitio a otro en la lógica, lo cual es típico de otros estilos de programación. La estructura del programa es más clara puesto que las instrucciones están más ligadas o relacionadas entre sí, por lo que es más fácil comprender lo que hace cada función.
- Reducción del esfuerzo en las pruebas. El programa se puede tener listo para producción normal en un tiempo menor del tradicional; por otro lado, el seguimiento de las fallas (debugging) se facilita debido a la lógica más visible, de tal forma que los errores se pueden detectar y corregir más fácilmente.
- Reducción de los costos de mantenimiento.
- Programas más sencillos y más rápidos.
- Aumento de la productividad del programador.
- Los programas quedan mejor documentados internamente.

1.2.6 Plataformas

1.2.6.1 .Net

Microsoft .NET Framework versión 2.0 Redistributable Package instala el entorno en tiempo de ejecución y los archivos asociados de .NET Framework necesarios para ejecutar aplicaciones desarrolladas para .NET Framework v2.0. [23]

.NET Framework versión 2.0 mejora la escalabilidad y el rendimiento de aplicaciones gracias a características mejoradas como el almacenamiento en caché, el desarrollo de aplicaciones y la actualización con ClickOnce; además, es compatible con la gama más amplia de exploradores y dispositivos con servicios y controles ASP.NET 2.0.

Esta nueva tecnología de Microsoft ofrece soluciones a los problemas de programación actuales, como son la administración de código o la programación para Internet. Para aprovechar al máximo las características de .Net es necesario entender la arquitectura básica en la que está implementada esta tecnología y así beneficiarse de todas las características que ofrece esta nueva plataforma.

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son: [24]

- Lenguajes de compilación
- Biblioteca de clases de .Net
- CLR (Common Language Runtime)

Arquitectura de .Net Framework

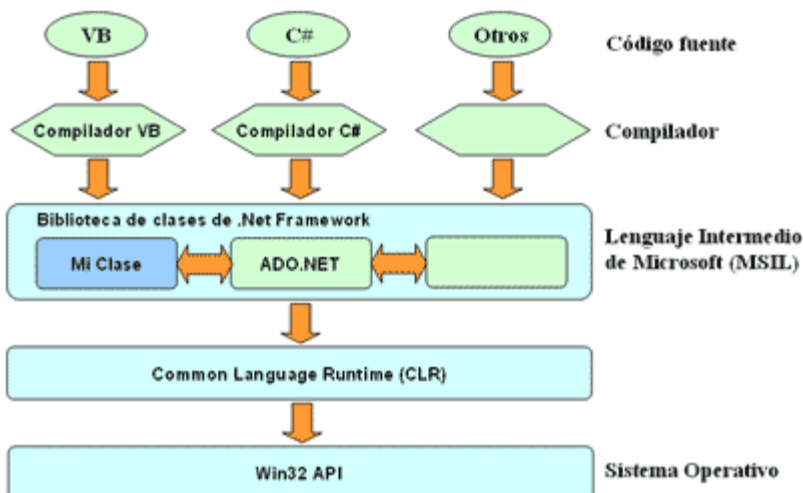


Figura 1: Arquitectura del Framework .NET.

El Common Language Runtime (CLR)

El CLR administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Es la máquina virtual de .NET. El código destinado al motor de tiempo de ejecución se denomina código administrado, a diferencia del resto de código, que se conoce como código no administrado.

La Biblioteca de Clases de .NET

La biblioteca de clases de .NET es una colección de tipos reutilizables que se integran estrechamente con el CLR. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. Esto ocasiona que los tipos de .NET sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET. Además, los componentes de terceros se pueden integrar sin dificultades con las clases de .NET.

Esta librería está escrita en MSIL (Microsoft Intermediate Language – Lenguaje Intermedio de Microsoft) que es un conjunto de instrucciones independiente del CPU que se pueden convertir de forma eficaz en código nativo. Por tanto esta librería puede ser utilizada desde cualquier lenguaje cuyo compilador genere MSIL.

1.2.6.2 Mono

Es una plataforma de desarrollo de código abierto basada en .NET. Es una plataforma para ejecutar y desarrollar aplicaciones modernas basadas en los estándares ECMA/ISO. Puede ejecutar aplicaciones hechas para los Framework .NET y Java.

Mono incluye las herramientas de desarrollo y la infraestructura necesarias para crear aplicaciones del cliente y el servidor .Posee importantes componentes útiles de desarrollo de *software*, entre los cuales está un compilador para el lenguaje C #; una máquina virtual de lenguaje común de infraestructura (CLI) que contiene un cargador de clases, un compilador en tiempo de ejecución (JIT), y unas rutinas de recolección de memoria.

Tiene un entorno de desarrollo integrado (IDE), MonoDevelop, gratuito, diseñado primordialmente para C#.

1.2.7 Entorno de Desarrollo Integrado

1.2.7.1 Visual Studio 2005

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo Entorno de Desarrollo Integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML.

Incorpora nuevas y mejoradas funciones de productividad: configuración del IDE, importar y exportar configuraciones, listas de tareas, lista de errores, teclas de método abreviado Brief y Emacs.

Presenta un nuevo diseñador de páginas Web que incluye muchas mejoras para la creación y edición de páginas Web de ASP.NET y páginas HTML. Proporciona una forma más fácil y rápida de crear páginas de formularios Web Forms que en Visual Studio .NET 2003.

La vista Diseño del diseñador HTML incluye muchas mejoras que admiten las nuevas funciones de ASP.NET o facilitan el diseño WYSIWYG de páginas Web. La edición basada en tareas mediante

etiquetas inteligentes le guía durante la ejecución de los procedimientos más comunes con controles, como el enlace de datos y la asignación de formato.

Puede editar visualmente las nuevas páginas principales de ASP.NET. La edición de plantillas se ha mejorado para facilitar el trabajo con controles de datos, así como con nuevos controles como el control Login. Editar tablas HTML para el diseño o mostrar la información en columnas ahora es más fácil e intuitivo.

1.2.7.2 SharpDevelop

Entorno Integrado de Desarrollo (IDE) libre. Es un proyecto que comenzó desde los comienzos del lenguaje C# en el año 2000, soporta C#, Visual Basic .NET, ASP.NET, XML. Es de diseño visual al igual que el Visual Studio, y bastante similar, es compatible con Net Framework 1.1, 2.0, Compact Framework y Mono, viene con depuradores, configuración de proyectos, además de ser escrito enteramente en C#.

SharpDevelop Incorpora:

- Diseñador de Formularios.
- Completado de Código.
- Auto-insertado de Código.
- Conversor de Código C# a VB.Net y viceversa.
- Importar/Exportar Soluciones VS.NET a Visual Studio .NET.
- Plegado de Código ("Folding").
- Visor gráfico para realizar pruebas con NUnit.
- Analizador del Código Ensamblador.
- Vista previa de Documentación en XML.
- También incluye sintaxis coloreada, paréntesis inteligentes, bookmarks, plantillas, herramientas para expresiones regulares, asistentes, exportación HTML, visor de clases, integración con NDoc, integración con Nprof, etc.

1.2.8 Sistemas gestores de bases de datos (SGBD)

Son conjuntos de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Existen varios gestores de bases de datos, entre ellos: SQLServer, PostgreSQL, MySql, entre otros.

1.2.8.1 SQLServer

SQL Server 2005 es un gestor global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.[25]

Principales ventajas:

- **Facilidad de uso:** SQL Server 2005 simplifica el empleo y la administración y optimización de las aplicaciones empresariales para análisis de datos.
- **Disponibilidad:** Las inversiones en tecnologías de alta disponibilidad, capacidades de restauración y copias de seguridad adicionales, y mejoras de réplica permitirán a las empresas crear e implementar aplicaciones de alta disponibilidad.
- **Gestión:** Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- **Soporta procedimientos almacenados.**
- **Permite trabajar en modo cliente-servidor** donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

Principales Desventajas:

- **No es multiplataforma,** ya que sólo está disponible en Sistemas Operativos de Microsoft.
- **Licencias con costos altos.**

1.2.8.2 MySql

El software MySQL es gestor de bases de datos relacional que proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU. [26]

Entre las principales ventajas que propicia este gestor de bases de datos se encuentran las que se relacionan a continuación:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos
- Velocidad superior para la realización de diferentes operaciones.
- Bajo consumo, que posibilita su ejecución en máquinas con escasos recursos.
- Gran facilidad de configuración e instalación.
- Desventajas.
- Carece de soporte para transacciones, rollback's y subconsultas.

El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.

1.2.8.3 PostgreSQL

Dentro de los gestores de bases de datos se tiene, como uno de los más representativos, a PostgreSQL, un motor de base de datos que es servidor de base de datos relacional libre, liberado bajo la licencia BSD. Ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: Clases, Herencia, Tipos, Funciones.

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

Las principales mejoras en PostgreSQL incluyen:

Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde `pg_dump` mientras la base de datos permanece disponible para consultas.

- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena literal, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

Este gestor de base de datos corre en sistemas operativos tales como: Linux, Unix y Windows. Tiene completo soporte para llaves extranjeras, restricciones (constraints), disparadores (triggers), vistas y procedimientos de almacenado en múltiples lenguajes. Soporta el almacenamiento de grandes números binarios además de imágenes, videos y sonidos.

1.2.9 Metodologías de software

Las metodologías software ocupan un lugar importante para producir software de calidad en cualquier contexto de desarrollo pues determinan la planeación y seguimiento del proyecto que se desarrolla.

Los procesos ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

1.2.9.1 Metodologías ágiles

Las *metodologías ágiles* están basadas en heurísticas provenientes de prácticas de producción de código, preparadas para cambios durante el proyecto y el cliente es parte del equipo de desarrollo. Los grupos de trabajo están diseñados con poco personal trabajando todos en el mismo sitio, generan pocos artefactos, poseen pocos roles, además de no hacer énfasis en la arquitectura del software.

1.2.9.1.1 eXtreme Programming (XP).

Es la más destacada de los procesos ágiles de desarrollo de software. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental, pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas, las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que calidad del código escrito de esta manera es mayor (el código es revisado y discutido mientras se escribe) esto es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario, un representante del cliente trabaja junto al equipo de desarrollo.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

1.2.9.1.2 Scrum

Esta es, después de XP, la metodología ágil mejor conocida y la que otros métodos ágiles recomiendan como complemento, aunque su porción del mercado (3% según el Cutter Consortium) es más modesta que el ruido que hace. [27]

Este define roles:

- Scrum Master: el cliente interactúa con el equipo y debe ser miembro del mismo y trabajar a la par. Propietario del Proyecto: responsable oficial del proyecto, gestión, control y visibilidad de la

lista de acumulación o lista de retraso del producto (product backlog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.

- *Equipo de Scrum*: tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos.
- *Cliente*: participa en las tareas relacionadas con los ítems del registro.
- *Management*: está a cargo de las decisiones fundamentales y participa en la definición de los objetivos y requerimientos y por último el *Usuario*.

Su ciclo de vida se compone de 4 fases, *Pre-Juego: Planeamiento. Pre-Juego: Montaje (Staging). Juego o Desarrollo y Pos-Juego: Liberación.*

Al fin de cada iteración de treinta días hay una demostración a cargo del Scrum Master. Las presentaciones en power point están prohibidas.

Es habitual que Scrum se complemente con XP; en estos casos, suministra un marco de administración basado en patrones organizacionales, mientras XP constituye la práctica de programación, usualmente orientada a objetos y con fuerte uso de patrones de diseño. Uno de los nombres que se utiliza para esta alianza es XP@Scrum. También son viables los híbridos con otras metodologías ágiles.

1.2.9.1.3 Feature Driven Development (FDD).

FDD es un método ágil, iterativo y adaptativo. A diferencia de otras MAS, no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción y se considera adecuado para proyectos grandes y de misión crítica. Aunque hay coincidencias entre la programación orientada por rasgos y el desarrollo guiado por rasgos, FDD no necesariamente implementa FOP.

No requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. Los principios de FDD son pocos y simples:

- Se requiere un sistema para construir sistemas si se pretende escalar a proyectos grandes.
- Los pasos de un proceso deben ser lógicos y su mérito inmediatamente obvio para cada miembro del equipo.
- Vanagloriarse del proceso puede impedir el trabajo real.

- Los buenos procesos van hasta el fondo del asunto, de modo que los miembros del equipo se puedan concentrar en los resultados.
- Los ciclos cortos, iterativos, orientados por rasgos son mejores.

1.2.9.2 Metodologías tradicionales.

Por el contrario las *metodologías tradicionales*, están basadas en normas provenientes de estándares, seguidos por el entorno de desarrollo, con cierta resistencia a los cambios. El proceso es mucho más controlado, con numerosas políticas o normas, existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones, los grupos de trabajos son grandes y posiblemente distribuidos. Generan gran cantidad de artefactos y roles. La arquitectura del software es esencial y se expresa mediante modelos. Una de las más usadas es RUP.

1.2.9.2.1 Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características: adaptable, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar. Escalable, puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más. Flexible, es utilizada en el ambiente de desarrollo de cualquier cliente. Tecnología agnóstica, porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.[28]

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.

1.2.9.2.2 RUP con notación UML

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

Sus principales características se centran en: implementar las mejores prácticas en Ingeniería de Software, disciplinar la forma de asignar tareas y responsabilidades, quién hace qué, cuándo y cómo,

administrar requisitos, usar arquitectura basada en componentes y controlar cambios y modelado visual del software.

Posee tres características fundamentales, la primera de ellas es que su desarrollo es iterativo e incremental por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo. La segunda es que está guiado por los casos de uso. Un caso de uso será aquello que describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante.

Los casos de uso guían el diseño construcción y prueba del sistema, esto significa que guían el proceso de desarrollo. Por último y no la menos importante RUP está centrada en la arquitectura, lo que le permite a los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas, dejando a un lado los detalles.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Utiliza como lenguaje de modelado UML (Unified Modeling Language) el mismo permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. UML se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE). [29]

El Proceso Unificado de Software define 6 flujos básicos de desarrollo y tres de apoyo con los cuales se cubre todo el ciclo de vida del software, FDD, sin embargo es un método ágil, iterativo y adaptativo pero no cumple con esta condición debido a que se centra sólo en las fases de diseño y construcción por lo que es considerado adecuado para proyectos de misión crítica.

1.2.10 Tecnologías AJAX.

Las siglas AJAX significan: Asynchronous JavaScript And XML. Es decir, la combinación de JavaScript y XML de forma asincrónica. Es una técnica desarrollada para aplicaciones Web interactivas, que

consiste en hacer que un conjunto de tres tecnologías ya existentes trabajen juntas, de manera efectiva. [30]

Estas tecnologías son:

- (X)HTML y CSS (Hojas de Estilo en Cascada), para dar una estructura y presentar la información en la página web.
- JavaScript, utilizado para la interacción dinámica con los datos.
- XML, utilizado para la interacción con el servidor web. Aunque no es siempre necesario utilizar XML con las aplicaciones AJAX, ya que por ejemplo, con ficheros de texto plano también se puede almacenar la información.

Al igual que el DHTML, AJAX no es de por sí una tecnología web independiente, sino un término que abarca las tres tecnologías antes mencionadas.

Sirve para efectuar cambios en una página web, a voluntad del usuario, sin tener que recargar toda la página de nuevo.

El proceso de mostrar los datos realizado en HTML íntegramente consumiría bastante ancho de banda, ya que todo el HTML debería volver a ser cargado para sólo mostrar los cambios efectuados. En cambio, con una aplicación AJAX es mucho más rápido y no consume ancho de banda.

El JavaScript utilizado en la aplicación AJAX es un lenguaje dinámico, capaz de efectuar cambios en una página Web sin necesidad de re-cargarla. AJAX se asegura de que sólo la información necesaria sea solicitada y procesada, empleando SOAP, o otro lenguaje de servicios web basado ligeramente en XML. [31]

El resultado de usar AJAX es muy factible pues resulta una aplicación con una interfaz con mayor respuesta, dado la cantidad de datos intercambiados entre el cliente y el servidor ya que es reducida grandemente por lo antes dicho. También se ahorra mucho tiempo de procesamiento en el servidor web ya que una gran parte de dicho procesamiento se realiza en el cliente.

De ahí que a nivel técnico, se obtengan 3 ventajas: un tiempo de carga mucho menor, ahorro del ancho de banda del usuario y carga mucho menor el servidor donde está alojada la web.

1.2.11 Herramientas CASE

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas

herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática y documentación o detección de errores.[32]

Rational Rose es una herramienta CASE que da soporte al modelado visual con UML ofreciendo distintas perspectivas del sistema, lo que significa que todo su equipo corporativo puede comunicarse con un lenguaje y una herramienta, dominando debido a sus beneficios el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientado a objetos.

Principales características. [33]

- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad en múltiples plataformas.
- Permite visualizar, entender, y refinar sus requerimientos y arquitectura antes de enfrentarse al código.

La Metodología RUP es más adaptable a nuestro proceso debido a que está definida para proyectos de largo plazo o en otras palabras de grandes dimensiones, no así XP, pues se recomienda para proyectos de corto plazo. RUP es una metodología que propone y exige el uso de artefactos en cada iteración característica que le permite al software alcanzar un grado de certificación en el desarrollo del mismo.

El desarrollo del sistema estará sustentado bajo la metodología RUP, la cual utiliza UML, lenguaje que permite la modelación de sistemas con tecnología orientada a objetos y la herramienta Rational Rose que da soporte a la misma, ya que posibilita:

- Modelado del negocio.
- Captura de requisitos (parcial).

- Análisis y diseño (completo).
- Implementación (como ayuda).
- Control de cambios y gestión de configuración (parte).

¿Por qué usar Rational Rose Enterprise Edition?

Rational Rose Enterprise Edition es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a Lenguaje Unificado del Software, pero no son compatibles con las mismas tecnologías de implementación.[34]

Es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, Java/J2EE, Visual C++ y Visual Basic. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software.

Incluye funciones que permiten controlar de forma independiente los componentes del modelo, lo que posibilita una gestión y uso de modelos más específicos. Complementa el modelado web, incluyendo funciones de visualización, modelado y herramientas para desarrollar aplicaciones web. Diseña bases de datos a través del modelado en UML, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos y por último permite integración con otras herramientas de desarrollo de IBM Rational.

1.3 Sistemas automatizados existentes vinculados al campo de acción.

Un gran número de empresas están especializadas en el desarrollo e implantación de software para el sector de la salud, desarrollando aplicaciones para los hospitales y especialmente sistemas para la gestión oftalmológica para dar una mejor atención a los pacientes que necesitan atención médica.

1.3.1 Sistemas a nivel internacional.

1.3.1.1 ActualOftalmo!

Se trata de un software desarrollado por la empresa argentina ActualSoft! creada en 1987, software propietario destinado a los consultorio oftalmológico, presenta un protocolo gráfico e interactivo de la consulta oftalmológica y una extensa base de datos de signos y síntomas oftalmológicos, desarrollado por un equipo formado por médicos especialistas y profesionales informáticos, sobre Visual Basic 6.0 y con compatibilidad de la bases de datos DAO (MS Access).

Está conformado por:

- Protocolo Gráfico e interactivo de la Consulta Oftalmológica.

- Extensa base de datos de Signos y Síntomas Oftalmológicos.
- Registro de Antecedentes Familiares y Personales.
- Registro Digital de Estudios c/aparatos especializados.
- Herramienta Gráfica para Registro de Estrabismo, Fondo de Ojo y Biomicroscopía.
- Base de datos de Tratamientos Oftalmológicos.
- Vademécum Oftalmológico.
- Panel de Alarmas Inteligentes y Manuales.
- Recetario de Anteojos.
- Material Instructivo para el paciente.

1.3.1.2 VisionDat

Es un software de tecnología avanzada que ofrece diversas opciones en el manejo de consultorios. Está destinado a los consultorios especializados en oftalmología y optometría. Software propietario que no brinda información técnica en la cual fue desarrollado.

Este recoge los datos de pacientes, de consulta, de las citas y actividades programadas así como gráficas y estadísticas.

Puede generar información de cada paciente sobre:

- Historia Clínica con los antecedentes personales y heredo-familiares tales como diabetes, hipertensión, glaucoma, alergias, etc.
- Padecimiento Actual: pérdida de visión, cefalea, hiperemia, fotofobia, enrojecimiento ocular, etc.
- Registro de información de vital interés para la elaboración de diagnósticos y tratamientos, tales como, presión sanguínea, nivel de glucosa, hallazgos en los segmentos oculares, agudeza visual con y sin corrección, tonometría, refracción completa, visión binocular, etc.
- Lentes de contacto. En este importante capítulo de su consulta, puede registrar queratometría, patrón de fluoresceína, además de los datos relacionados a la base curva, el diámetro, el valor Dk/L, etc. de los lentes terapéuticos, hidrofílicos y gas permeable.

1.3.1.3 Emetropía

Software Integral de Oftalmología, desarrollado y difundido en España, permite la gestión completa de una clínica de oftalmología, es un software adaptable y totalmente personalizado,

La aplicación está desarrollada en Visual Basic, sistema de datos cliente / servidor, con un entorno muy amigable Windows, compatible con versiones desde Windows Vista hasta Windows 98, Base de Datos estándar y robusta Microsoft SQL Server, información totalmente exportable y legible por otras bases de datos. Posibilidad de establecer seguridad por usuario. Este permite:

- Recoger datos personales del paciente.
- Facilidad de búsqueda del paciente.
- Recoger datos oftalmológicos del paciente.
- Extraer información de dispositivos externos.
- Capturar imágenes de aplicaciones externas.
- Recoger sus citas diarias.
- Agenda por médicos.
- Agenda por salas.
- Configuración de tramos de horarios por médico.
- Revisar con facilidad su historial de citas.
- Recoger llamadas del paciente.
- Controlar el estado del paciente en todo momento.

1.3.1.4 OFTALCLINIC

Es un intuitivo y flexible sistema integral de gestión de clínicas oftalmológicas que incorpora herramientas para la administración y dirección financiera así como para la gestión médica con historiales electrónicos, destinadas a optimizar los recursos humanos.

Emplea tecnología cliente-servidor sobre base de datos Microsoft SQL-Server de alto rendimiento, gracias a esto se puede interconectar varias clínicas centralizando así toda la información, tanto económica como clínica. Permite la transmisión de datos de equipos de medida computarizados, con lo

que se consigue seguridad y rapidez a la gestión de datos al incluirlos de manera automática en la ficha del paciente.

1.3.1.5 OFTALMOSALUS

OFTALMOSALUS es el Software especializado para la Gestión de Clínicas y Consultas de Oftalmología.

El Software, desarrollado por QSOFT desde 1995, Diseñado por oftalmólogos y por expertos en gestión de Clínicas Oftalmológicas, permite gestionar de manera completa e integrada las 3 grandes áreas de Gestión del Centro: Agendas, Historias Clínicas y Facturación. *OFTALMOSALUS* está actualmente implantado en Centros Oftalmológicos de referencia en España y otros países posibilitando: [35]

- Acceso sencillo y rápido al historial de los pacientes.
- Historia médica especializada para Oftalmología y configurable a medida.
- Generación automática de informes, recetas y demás documentos.
- Inserción de Croquis e Imágenes (Campimetrías, fotografías, etc.).
- Diagnósticos, Registro de Pruebas Complementarias, Protocolos, etc.
- Conexión a equipos de diagnóstico.

1.3.2 Sistemas a nivel nacional.

Actualmente solo existe un software oftalmológico funcionando en el país. **S.A.M.C** (Sistema Automatizado de Microcirugía).

Sistema que está funcionando actualmente en el hospital oftalmológico Ramón Pando Ferrer. Capaz de gestionar parte de la información generada durante el proceso quirúrgico por el que transitan los paciente. Desarrollado en "Clipper'85 Winter", lenguaje de programación creado en 1985 por Nantucket Corporation y herramienta líder de desarrollo, bajo sistema operativo MS-DOS y bases de datos ficheros mdf, presenta una interfaz gráfica poco amigable, de difícil configuración.

Bloque Quirúrgico Oftalmológico, unos de los intentos de desarrollo, para dar solución a las problemáticas existente en los hospitales oftalmológicos cubanos, aunque no ha habido la suficiente difusión e intercambio de experiencias como para realizar y estandarizar un producto que cubra las expectativas de los centros de salud oftalmológicos existentes, este se encuentra en desarrollo.

La meta es construir un software estandarizado donde el paciente sea el más beneficiado, y los profesionales de la salud encuentren en estos sistemas un recurso idóneo, amigable y flexible que responda a las necesidades de información de la institución hospitalaria, que además no solo posibilite su integración con sistemas existentes, sino que otros sistemas puedan ser diseñados e implementados contando con las funcionalidades que brinda.

1.4 Justificación de la propuesta de solución.

Después de analizar un grupo de lenguajes candidatos se decidió por el arquitecto del sistema el uso de la plataforma .NET y por ende el lenguaje de desarrollo del lado del servidor ASP.NET, como lenguaje de desarrollo del lado del cliente se usará JavaScript con AJAX. ASP.NET proporciona varias ventajas, lo primero que llama la atención es el enorme incremento de velocidad de respuesta del servidor.

El gestor de bases de datos que se pretende utilizar en nuestro sistema es PostgreSQL, teniendo en cuenta principalmente su condición de *software* multiplataforma y con licencia de uso libre, además de las potencialidades desde el punto de vista de funcionamiento que garantiza con su uso.

Se utiliza como metodología de desarrollo de *software* el RUP. Esta metodología permite mayor productividad en equipo y la realización de mejores prácticas de *software* a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo crítico del *software*.

La metodología RUP es un proceso tanto iterativo como incremental, por cual permite tener mini proyectos antes de terminar todo el *software*, con esto se puede presentar a nuestro cliente un avance de proyecto y el pueda evaluar cómo se está trabajando.

La herramienta CASE propuesta a utilizar es Rational Rose que permite completar una gran parte de las disciplinas (flujos fundamentales) del RUP, es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas del Rational Rose es que utiliza la notación estándar en la arquitectura de *software* (UML).

En este capítulo se profundizó en el análisis y conocimiento de conceptos necesarios para la comprensión de este trabajo, realizándose un estudio de lo relacionado al tema tratado en aras de enriquecer y profundizar los conocimientos al respecto. Además se realizó una investigación completa de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, especificándose la elecciones realizadas al respecto, fundamentándose la elección del lenguaje de programación, sistema gestor de bases de datos, metodología a utilizar , apoyado de las herramientas necesarias.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El Sistema Nacional de Salud (SNS) dentro de sus niveles de atención médica, incluye a los hospitales como parte de la atención secundaria. El área quirúrgica de un hospital es un eslabón fundamental en este nivel, donde se desarrollan procesos que a su vez generan un grupo de registros que soportan toda la información recuperada como parte de dichas acciones. Es por esto que se propone la elaboración de un sistema que informatice estos registros y se documenten los artefactos de desarrollo que se elaboren como resultado de esta tarea. En este capítulo se describen los procesos y la forma en que los actores interactúan con el sistema.

2.1 Objetivos estratégicos de la organización.

El Ministerio de Salud Pública (MINSAP) es el Organismo rector del Sistema Nacional de Salud y se encarga de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.

El Sistema Nacional de Salud (SNS), a su vez se compone por la forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado. Es el conjunto de Unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la Salud de una población.

La estructura del SNS cubano se compone por 3 niveles, en correspondencia con los niveles administrativos del país:

- El Nivel Nacional, representado por el Ministerio de Salud Pública, como órgano rector con funciones metodológicas, normativas, de coordinación y de control.
- El Nivel Provincial, representado por las Direcciones Provinciales de Salud, las cuales están directamente subordinadas administrativa y financieramente a la Asamblea Provincial del Poder Popular (órgano de gobierno a esa instancia).
- El nivel Municipal, representado por las Direcciones Municipales de Salud y dependientes, administrativa y financieramente de la Asamblea Municipal del Poder Popular.

Dentro de las funciones fundamentales del SNS se encuentra la atención médica hospitalaria que entra en la categoría de asistencia, para poder ejecutar esta función de forma correcta se hace necesario e imprescindible que en cada centro hospitalario exista un departamento de inscripción admisión que se

encargue de inscribir a los pacientes y gestionar todos los movimientos del mismo, dentro del hospital.

El nivel secundario da cobertura a cerca del 15 % de los problemas de salud, su función fundamental es tratar al hombre ya enfermo, ya sea desde el punto de vista individual como colectivo pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas (Especialidades). Comprende la atención médica brindada en los distintos Hospitales.

Los bloques quirúrgicos oftalmológicos son creados con el objetivo de dar atención a pacientes cuyo tratamiento necesite cirugía. Ellos brindan servicios entre los que se encuentran el servicio de Oculoplastia, Baja Visión, Neurología, Oftalmología Pediátrica, entre otros.

El “Hospital Ramón Pando Ferrer” especializado en el servicio de Oftalmología tiene como objetivo brindar una atención al paciente de calidad, para esto se desarrolla la presente aplicación para permitir que el procesamiento de la información generada por este servicio sea rápido y eficiente y que comprenda todos los subservicios que lo conforman. En estos momentos se prevé la informatización de este servicio debido a la afluencia de pacientes que transitan actualmente por este centro gracias a la Misión Milagro que realiza el país hace varios años y que ha devuelto la vista a miles de personas.

2.2 Flujo actual de los procesos.

Cuando el paciente llega al área oftalmológica del hospital lo primero que hace es dirigirse a la consulta oftalmológica, donde es atendido directamente por uno de los servicios oftalmológicos. El Departamento Quirúrgico Oftalmológico permite realizar el control del paciente, de forma que se incorporen a su Historia Clínica todas aquellas actividades, documentación o procesos que se desencadenan durante su estancia en el mismo.

Uno de los principales procesos que se llevan a cabo en el bloque quirúrgico Oftalmológico en estos momentos es la consulta especializada.

En esta consulta es valorado por el oftalmólogo, que realiza los exámenes y estudios pertinentes para determinar el padecimiento del paciente, si luego de esto, se decide la necesidad de cirugía se procede a llenar el anuncio operatorio como documento necesario para la realización de la misma.

Por lo general, el paciente llega a la consulta con una remisión a un servicio especializado de oftalmología, aunque, en caso de no ser así, el especialista puede decidir atenderlo fuera de servicio o remitirlo hacia el servicio especializado en la afección que presenta el paciente. Es precisamente aquí

donde la frontera entre los servicios se ve mucho más difusa, puesto que no existe una consulta general oftalmológica, sino que las consultas son directamente por cada uno de los servicios.

En todas estas consultas de manera general el médico recoge los datos generales del paciente, los antecedentes patológicos generales tanto personales como familiares y los antecedentes patológicos oftalmológicos igualmente personales y familiares. Posteriormente se realiza un examen físico al ojo del paciente, donde se revisan cada una de las estructuras del mismo, aunque se hace más énfasis en las estructuras que corresponden al servicio que se presenta. Seguidamente a ello se realizan los exámenes oftalmológicos generales, como son la refracción, la medición de la agudeza visual, entre otros por identificar.

A continuación se desarrollan los exámenes específicos de cada servicio, hasta que el médico tiene los elementos suficientes para emitir una impresión diagnóstica.

La impresión diagnóstica, la emiten también todos los servicios, posteriormente se le plantea al paciente la conducta a seguir que pueden ser en todos los casos las siguientes variantes:

Si el paciente necesita *cirugía* se procede a llenar el anuncio operatorio como documento necesario para la realización de la misma además se le indica al paciente el día y la fecha de la cirugía.

Si la conducta es *no quirúrgica* se especifica si la no cirugía es temporal o definitiva, se le explica la causa que puede ser debido a que el paciente presenta algún impedimento que impida realizar el acto quirúrgico o el padecimiento no requiera de la intervención invasiva.

Otra conducta a seguir es de *seguimiento*, donde el especialista decide volver al paciente en consultas posteriores debido a que los exámenes no son concluyentes y necesita repetirlos o continuar con un seguimiento para actuar sobre el futuro desarrollo de la enfermedad.

Si la conducta es la *interconsulta*, que es donde el oftalmólogo remite al paciente a otro especialista debido a que el diagnóstico se corresponde con un padecimiento relacionado con otro servicio oftalmológico.

Durante todo el proceso de la consulta oftalmológica el especialista anota todos los detalles en la Historia Clínica del paciente, sirviendo esto como material de apoyo para las posteriores consultas.

2.3 Objeto de automatización.

Como resultado de estos procesos se generan un grupo de documentos que serán el principal objeto de automatización de la investigación en curso. Los documentos que se presentan y explican a continuación han sido escogidos para automatizarse en la segunda iteración del software por lo importancia que reportan en el funcionamiento básico del bloque quirúrgico oftalmológico.

La hoja del especialista oftalmológico es un documento elaborado por cada especialista en su consulta, donde se recoge el motivo de la consulta y se realiza un estudio del padecimiento del paciente, dando al final un diagnóstico que no necesariamente debe ser el definitivo. Si el diagnóstico devela que el paciente necesita cirugía entonces se llena el anuncio operatorio.

Este es un documento de guía para el especialista en el momento de la cirugía además sirve para en el futuro obtener estadísticas importantes referentes a parámetros como cantidad de pacientes atendidos por un diagnóstico específico.

La hoja del pediatra son documentos elaborados por el pediatra y clínico respectivamente, donde se recogen datos del paciente relacionados con los antecedentes familiares y personales del mismo, además de las principales patologías y padecimientos que pueda presentar o haber presentado durante su vida. Según estos datos se da la valoración de apto quirúrgico o no, lo cual también se registra en estos documentos, así como las observaciones y la valoración realizada por los médicos.

En sentido general los documentos que se generan en este departamento auxilian a los médicos durante la estancia del paciente en el hospital y en las consultas posteriores. Además contienen los datos que permiten obtener estadísticas que facilitan la toma de decisiones futuras con mayor seguridad.

2.4 Propuesta de sistema.

El objetivo es diseñar un sistema capaz de gestionar la información de servicios quirúrgicos oftalmológicos que brinde un hospital. El mismo contará en su segunda iteración con un módulo que recoja todos los procesos de los servicios oftalmológicos, Oculoplastia y Baja Visión, servicios que no fueron desarrollados.

El módulo Bloque Quirúrgico Oftalmológico realizada en la primera iteración servirá de soporte mientras se diseñen los servicios restantes además de brindar la posibilidad de reutilizar los componentes ya realizados. Finalmente será integrado con el módulo desarrollado en la primera iteración que conformará un Sistema de Información Hospitalaria permitiendo la gestión de la información en los demás departamentos del hospital.

Sus características básicas estarán centradas en:

- Visualizar los datos registrados en la historia clínica siempre que se atienda a un paciente.
- Crear y modificar la consulta especializada oftalmológica.
- Realizar búsquedas de paciente, consultas, informes operatorios y anuncios oftalmológicos.
- Registrar los datos necesarios del paciente durante la intervención quirúrgica oftalmológica.
- Registrar las consultas oftalmológicas realizadas al paciente.
- Crear y modificar el anuncio operatorio oftalmológico.
- Crear y modificar el informe operatorio oftalmológico.

La disciplina de Modelado de Negocio es la primera que propone RUP dentro del ciclo de desarrollo de un SW, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Es en este flujo de trabajo donde se conocen a fondo cómo son iniciadas cada una de las actividades de un sector determinado, y a través del modelado de estos procesos se obtiene una visión más amplia del negocio existente.

RUP propone que para esta disciplina se modele el negocio siempre y cuando los procesos sean fácilmente identificables, en el caso de que no puedan percibirse se realizaría un modelo de dominio que englobaría los principales conceptos que sean encontrados para a partir de este modelo proseguir al Flujo de trabajo de Requerimientos.

2.5 Actores y trabajadores del Negocio.

Un actor del negocio: 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.

Actores del negocio	Justificación
Paciente	Este actor es el mayor beneficiado con los procesos que se desarrollan en el Bloque Quirúrgico. Inicia los casos de uso Realizar Intervención Quirúrgica, Realizar Consulta de Anestesia, Realizar Consulta Especializada y Realizar Consulta Aprobar Cirugía.
Vicedirector Quirúrgico	Este actor representa al directivo de la institución encargado interactuar directamente con la planificación quirúrgica, para obtener estadísticas como las distribuciones de salones por meses, cantidad de pacientes a operar entre otros. Es el encargado de realizar la distribución inicial de los salones del departamento de cirugía. Inicia el caso de uso Realizar Planificación Quirúrgica.

Tabla 1: Actores del Negocio.

Trabajadores del Negocio: es quién define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo.

Trabajadores del negocio	Justificación
Anestesiista	Es el encargado de llevar el control de la Hoja de Anestesia desde la Consulta de Anestesia hasta los procesos del la Intervención Quirúrgica, el preoperatorio, transoperatorio y postoperatorio.
Médico	Es la generalización de Clínico y Pediatra que son los encargados de dar la valoración de apto o no al paciente para ser intervenido. Modifican la historia clínica con el fin de plasmar en ella datos específicos de las consultas que se le hacen al paciente.
Auxiliar del Salón	Encargada de recoger las causas de suspensión de los anuncios operatorios. Llena y modifica el Informe Operatorio al finalizar cada Intervención Quirúrgica.
Especialista	Es el encargado de manejar los anuncios operatorios de los pacientes. Crea o modifica los mismos si es necesario en el tiempo establecido. Es el único autorizado a cambiar el estado de un anuncio.
Jefe de Servicio	Este trabajador puede en algunas ocasiones si es necesario ajustar la planificación quirúrgica para casos de urgencia o por problemas internos del servicio.

Tabla 2: Trabajadores del Negocio.

2.5.2 Modelo de Casos de Uso del Negocio

El modelo de casos de uso del negocio representado a continuación recoge los procesos que son desarrollados en el Bloque Quirúrgico de manera general. Fue diseñado en la primera iteración atendiendo a las semejanzas de los procesos que realizan los servicios del BQ, el flujo de los procesos es el mismo para cualquier servicio con la diferencia de que cada uno maneja sus datos específicos.

Diagrama de Casos de Uso del negocio

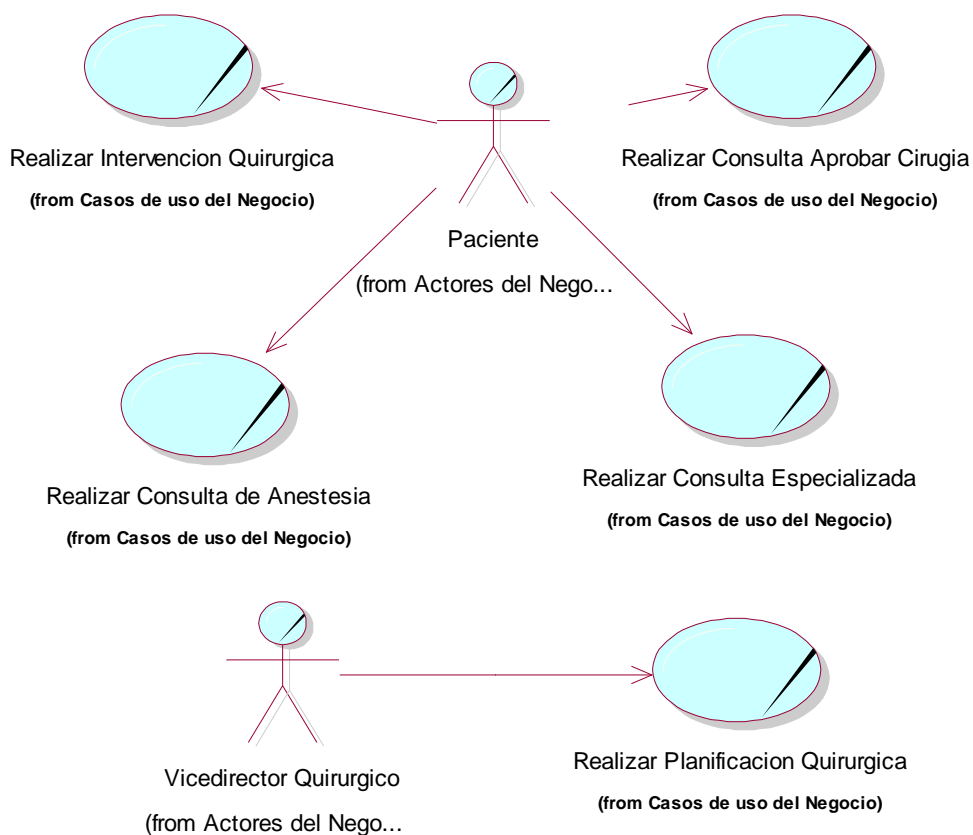


Figura 2: Casos de Uso del Negocio.

Caso de uso *Realizar Consulta Especializada* describe el proceso desarrollado en la consulta con el especialista.

Caso de uso *Realizar Consulta Aprobar Cirugía* describe el proceso desarrollado en las consultas Clínicas y Pediátricas.

El caso de uso *Realizar Consulta de Anestesia* es también una consulta de aprobación de cirugía pero por presentar especificidades en los documentos que maneja se decide separar y poner como un caso de uso aparte. Describe el proceso desarrollado durante la consulta de anestesia.

Caso de uso *Realizar Intervención Quirúrgica* describe el proceso pre-operatorio, trans-operatorio y post-operatorio por el que transita el paciente durante la intervención.

El caso de uso *Realizar Planificación Quirúrgica* describe como se realiza una planificación de manera general.

2.5.3 Descripción textual de los Casos de Uso de Negocio

CASO DE USO DEL NEGOCIO	Realizar Consulta Oftalmología Especializada
Actores	Paciente(Inicia)
Propósito	Registrar los datos del paciente y hacerle un anuncio operatorio
RESUMEN: El caso de uso se inicia cuando el paciente llega a la consulta. El médico lo reconoce, diagnostica, y realiza el anuncio operatorio.	
ACCIÓN DEL ACTOR	RESPUESTA DEL PROCESO DE NEGOCIO
1 El paciente llega a la consulta especializada.	2 El especialista analiza el motivo de la consulta.
	3 No es para egresar.
	4 El especialista llena la hoja de cargo.
	5 Analiza datos de la HC.
	6 Realiza interrogatorio.
7 El paciente le comunica los Síntomas.	8 El especialista emite un diagnóstico.
	9 Si necesita cirugía.
	10 Llena el Anuncio operatorio.
	11 Si necesita tratamiento.
	12 Receta fármacos y emite orden de tratamiento.
	13 Valora si necesita ingresarse.
	14 Si necesita ingreso emite orden de ingreso.
	15 Actualiza Historia Clínica.
Flujos Alternos	
	3.1 Si es para egresar emite orden de egreso.
3.2 El paciente se retira.	9.1 No necesita cirugía ir a la actividad numero 11.
	11.1 No necesita tratamiento, ir a la actividad numero 13.
	13.1 No necesita ingresarse, ir a la actividad numero 15.
Prioridad	
Mejoras	
Otras secciones	

Tabla 3: Descripción textual del Caso de Uso Realizar Consulta Oftalmología Especializada.

CASO DE USO DEL NEGOCIO	Realizar Consulta de Anestesia
Actores	Paciente(Inicia)
Propósito	Confirmar si el paciente puede ser operado o no.
RESUMEN: El caso de uso se inicia cuando el paciente llega a la consulta. El médico lo examina, revisa los análisis antes indicados por el especialista, los analiza y da confirmación de si puede operarse o no.	
ACCIÓN DEL ACTOR	RESPUESTA DEL PROCESO DE NEGOCIO
1 El paciente llega a la consulta.	2 El anestesista verifica si tiene anuncio operatorio.
	3 Solicita y verifica análisis orientados.
	4 Realiza interrogatorio.
5 Contesta preguntas.	6 Si está apto para ser operado.
	7 Si existen condiciones para realizar la intervención.
	8 Llena Hoja de anestesia.
Flujos Alternos	
	2.1 Si no tiene anuncio operatorio se remite con el especialista. (Termina el caso de uso.)
	6.1 Si no está apto para ser operado se remite nuevamente con el especialista.
	6.2 Suspende anuncio operatorio.
	6.3 Emite tratamiento. (Termina el caso de uso.)
	7.1 Si no existen condiciones.
	7.2 pospone el anuncio operatorio.
	7.3 Informa al paciente.
Prioridad	
Mejoras	
Otras secciones	

Tabla 4: Descripción textual del Caso de Uso Realizar Consulta Anestesia.

CASO DE USO DEL NEGOCIO	Realizar Intervención Quirúrgica Oftalmológica
Actores	Paciente(Inicia)
Propósito	Ejecución de la cirugía al paciente.
RESUMEN: El caso de uso se inicia cuando el paciente llega al salón preoperatorio. Se le aplican los medicamentos necesarios para sedarlo se llenan los documentos necesarios y se traslada al salón. Comienza el acto quirúrgico se van llevando los datos en el transcurso de la operación y al finalizar esta se llenan los informes finales.	
ACCIÓN DEL ACTOR	RESPUESTA DEL PROCESO DE NEGOCIO
1 El paciente llega a la sala del preoperatorio.	2 El anestesista consulta anuncio operatorio y hoja de anestesia de consulta.
	3 Aplica anestesia preoperatorio.
	4 Si existe complicación.
	5 El vicedirector decide si se cancela o no la intervención.
	6 Si no se cancela.
	7 Envía al paciente al salón de operaciones.
8 El paciente llega al quirófano.	9 El anestesista consulta hoja de anestesia y anuncio operatorio.
	10 Aplica anestesia operatoria al paciente.
	11 El cirujano comienza la intervención.
	12 El anestesista chequea el estado del paciente.
	13 El cirujano verifica si necesita biopsia.
	14 El cirujano separa la muestra y la auxiliar del salón (AS) llena la solicitud de biopsia.
	15 El técnico cuenta instrumentales y compresas.
	16 El cirujano finaliza la intervención quirúrgica.
	17 El anestesista actualiza la hoja de anestesia y la AS llena el informe operatorio.
18 El paciente pasa a recuperación	19 El anestesista se mantiene chequeando el estado del paciente (termina el CU).
Flujos Alternos	
	4.1 Si no existe complicación, pasar a la actividad numero 6.
	6.1 Si se cancela.
	6.2 El anestesista valora si se suspende o aplaza el anuncio operatorio.
	6.3 La auxiliar del salón registra la causa.
	13.1 Si no necesita biopsia, pasar a la actividad 15.
Prioridad	
Mejoras	
Otras secciones	

Tabla 5: Descripción textual del Caso de Uso Realizar Intervención Quirúrgica Oftalmológica.

Capítulo 2: Características del Sistema

Caso de Uso del Negocio	Realizar Planificación Quirúrgica
Actores	Jefe quirúrgico(Inicia)
Propósito	Realizar la planificación de las operaciones.
Resumen: El caso de uso se inicia cuando el Jefe quirúrgico entrega la distribución de los salones a los jefes de servicios. Estos escogen el método de planificación y planifican las cirugías.	
Acción del actor	Respuesta del proceso de negocio
1 Jefe quirúrgico entrega la distribución de los salones a los jefes de servicios	2 Los jefes de servicios escogen el método de planificación.
	3 Elabora planificación quirúrgica según método escogido.
Prioridad	
Mejoras	
Otras secciones	

Tabla 6: Descripción textual del Caso de Uso Realizar Planificación Quirúrgica.

Caso de Uso del Negocio	Aprobar Cirugía
Actores	Paciente(Inicia)
Propósito	Registrar los datos del paciente y hacerle un anuncio operatorio.
Resumen: El caso de uso se inicia cuando el paciente llega a la consulta. El médico lo reconoce y diagnostica, le orienta los análisis preoperatorios y le realiza el anuncio operatorio.	
Acción del actor	Respuesta del proceso de negocio
1 El caso de uso inicia cuando el paciente llega a la consulta para recibir atención médica.	2 Solicita la remisión que trae el paciente.
3 El paciente entrega la remisión.	4 Chequea la remisión que le es entregada por el paciente.
	5 Determina que el motivo del paciente en la consulta está dado, para determinar si está apto o no para ser operado.
	6 Chequea los datos que le brinda el anuncio operatorio.
	7 Solicita al paciente otros datos de interés para el proceso quirúrgico.
	8 El especialista emite un diagnóstico.
9 El paciente entrega los datos solicitados.	
	10 Hace una valoración a partir de los datos que le ha informado el paciente durante la consulta.
	11 Determina que el paciente no está apto para ser intervenido quirúrgicamente.
	12 Determina que el paciente no requiere ser remitido a una consulta fuera del hospital.

	13 Propone tratamiento al paciente.
	14 Actualiza la Historia Clínica del paciente.
	15 El caso de uso termina.
Flujos Alternos	
	5.1 Determina que el paciente llega a la consulta para ponerle un tratamiento.
	5.2 Solicita datos de su interés al paciente.
5.3 El paciente entrega los datos al pediatra.	5.4 Hace una valoración a partir de los datos que le informa el paciente.
	5.5 Notifica al paciente del tratamiento puesto.
5.6 El paciente recibe la notificación.	5.7 Actualiza la Historia Clínica del paciente.
	5.8 El caso de uso termina.
	11.1 Determina que el paciente está apto para ser intervenido quirúrgicamente.
	11.2 Actualiza la Historia Clínica del paciente.
	11.4 El caso de uso termina.
	12.1 Determina que el paciente requiere ser remitido a una consulta fuera del hospital.
	12.2 Elabora un documento: "Remisión de caso".
	12.3 El caso de uso termina.
Prioridad	
Mejoras	
Otras secciones	

Tabla 7: Descripción textual del Caso de Uso Aprobar Cirugía.

2.5.4 Diagrama de actividades por Casos de Uso de Negocio.

El diagrama de actividad es un grafo que contiene estados en que puede hallarse una actividad. Para profundizar en los diagramas de actividades ver Anexo 1.

2.6 Requerimientos del sistema.

El sistema que se propone forma parte del Sistema de gestión hospitalaria, siendo un subsistema del mismo y que al integrarse con otros conformarán el producto software esperado para brindar una solución informática a los hospitales del país para una mejor prestación de los servicios.

Requisitos Funcionales.

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

R1 Gestionar Hoja del Especialista Oftalmológico en Oculoplastia.

R1.1 Crear Hoja del Especialista Oftalmológico en Oculoplastia.

R1.2 Modificar Hoja del Especialista Oftalmológico en Oculoplastia.

R2 Gestionar Hoja del Especialista Oftalmológico en Baja Visión.

R2.1 Crear Hoja del Especialista Oftalmológico en Baja Visión.

R2.2 Modificar Hoja del Especialista Oftalmológico en Baja Visión.

Requerimientos no funcionales.

Requerimientos de apariencia o interfaz externa.

El sistema debe tener un ambiente amigable y entendible para los usuarios finales, de forma tal que no les sea muy complicado utilizar el software.

No debe utilizarse tecnología de frames.

Cada página no debe exceder los 500 Kb en imágenes.

Requerimientos de usabilidad.

La aplicación debe cumplir con los principales principios de usabilidad, debe brindar comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación mediante teclas de acceso rápido, la navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

Requerimientos de rendimiento.

El tiempo de respuesta de una petición al servidor debe ser el mínimo, garantizando actualidad en los datos para la toma de decisiones.

Requerimientos de soporte.

Se le debe dar mantenimiento periódico a los servidores de bases de datos controlando la integridad de la información.

La aplicación será capaz de actualizarse desde servidores de aplicaciones destinados para esa operación.

Requerimientos de portabilidad.

El producto podrá ser usado bajo cualquier sistema operativo ya sea Linux o Windows.

Requerimientos de seguridad y privacidad.

La información debe transmitirse de manera segura, se debe garantizar la seguridad a todos los niveles (interfaz, negocio y acceso a datos) restringiendo las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

Requerimientos de confiabilidad.

La información debe transmitirse a través de canales seguros. Se debe chequear la integridad de los datos.

Requerimientos de ayudas y documentación en línea.

Se debe brindar una interfaz amigable que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además los manuales de usuario y toda la documentación actualizada de cada módulo de la aplicación.

Requerimientos de hardware.

Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 10Gb HDD (disco duro).

Requerimientos para un servidor: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.

Requerimientos de software.

El sistema debe correr en sistemas operativos Windows, Unix y Linux utilizando la plataforma Mono 1.2.4 o superior. El cliente solo deberá disponer de un navegador web (IE6 o Firefox).

Restricciones en el diseño y la implementación.

El sistema será implementado utilizando como lenguaje de programación del lado del cliente JavaScript auxiliándose del grupo de tecnologías que contiene AJAX.

Se utilizará como plataforma de desarrollo Microsoft .NET versión 2.0 que brinda una gama de facilidades en su entorno y que da la posibilidad de utilizar el lenguaje C# del lado del servidor.

Se utilizará para la comunicación WSDL y Npgsql para la conexión al servidor de bases de datos.

2.7 Actores del Sistema.

Actores del Sistema	Justificación
Anestesiista	Es el encargado de crear y modificar la hoja de anestesia desde la consulta de anestesia hasta los procesos de la intervención quirúrgica, el pre-operatorio, trans-operatorio y post-operatorio.
Médico	Es la generalización de Clínico y Pediatra que son los encargados de dar la valoración de apto o no al paciente para ser intervenido. Modifican la historia Clínica con el fin de plasmar en ella datos específicos de las consultas que se le hacen al paciente.
Auxiliar del Salón	Crea y modifica el informe operatorio al finalizar cada Intervención Quirúrgica.
Especialista	Es el encargado de manejar los anuncios operatorios de los pacientes. Crea o modifica los mismos si es necesario en el tiempo establecido. Es el único autorizado a cambiar el estado de un anuncio.
Jefe de Servicio	Es el encargado de realizar la planificación quirúrgica del servicio al que pertenece.

Tabla 8: Actores del Sistema.

2.8 Definición de los casos de uso del sistema.

Un caso de uso no más que una secuencia de actividades que realiza un sistema y que da como resultado un valor para el actor. Estos han alcanzado un uso universal debido a dos razones básicas, la primera de ellas es que proporcionan un medio intuitivo y sistemático de capturar los requisitos anteriormente mencionados, centrándose en lo que quiere obtener el cliente, y la segunda es que dirigen todo el proceso apreciando que el análisis, diseño y prueba se realizan partiendo de los casos de usos.

El modelo de casos de uso del subsistema Bloque Quirúrgico fue diseñado en su primera iteración, por paquetes o subsistema, en ellos los casos de uso están agrupados por funcionalidades del sistema que responden al negocio estudiado, de manera que el diagrama se compone por cinco paquetes distribuidos de la siguiente forma:

- Paquete Anunciar Cirugía.
- Paquete Aprobar Cirugía.
- Paquete Acto Quirúrgico.

- Paquete Planificación.
- Paquete Búsquedas.

Las funcionalidades de Bloque Quirúrgico General fueron modeladas individualmente a las de Bloque Quirúrgico Oftalmológico con el objetivo de que fuera flexible a la hora de agregar al sistema los servicios restantes. Estos nuevos servicios serán añadidos en esta iteración, con el objetivo de una mejor comprensión del sistema se realizaron cambios en el Diagrama de CU. Paquete_Anunciar Cirugía.

Los criterios utilizados se basan en las funcionalidades que brindan los casos de uso, de manera que el sistema permita las funcionalidades para el desarrollo del software a desarrollar.

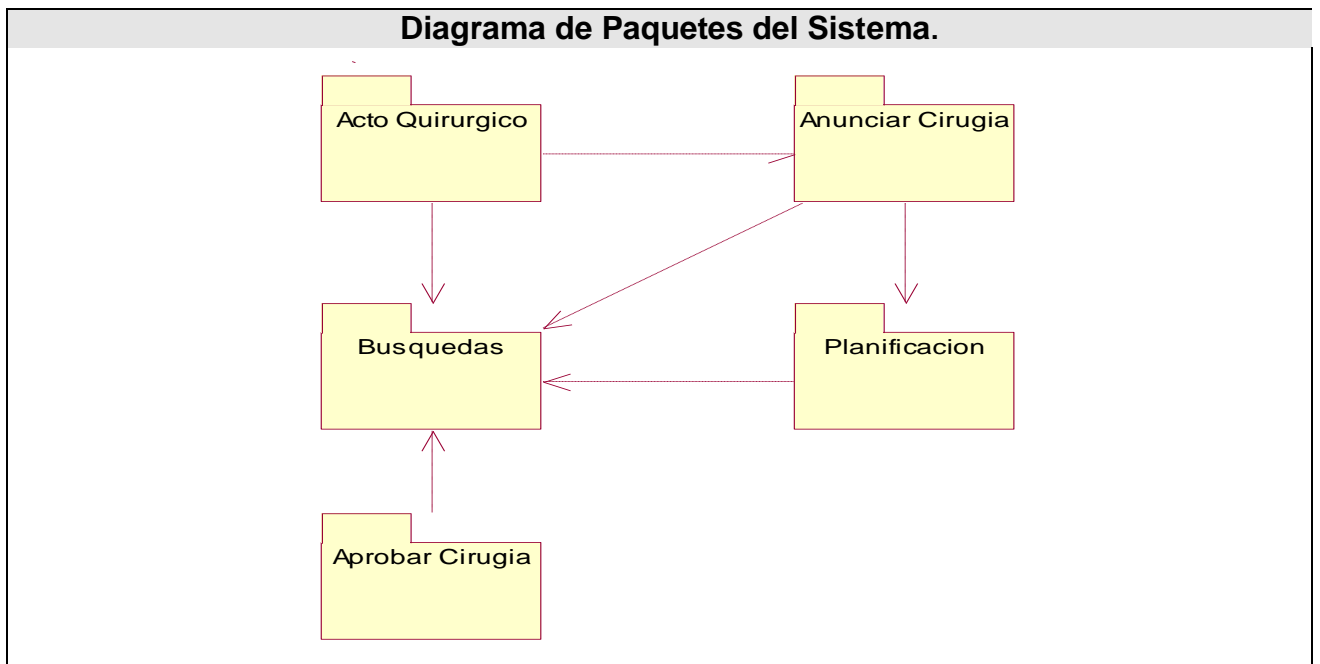


Figura 3: Diagrama de Paquetes del Sistema.

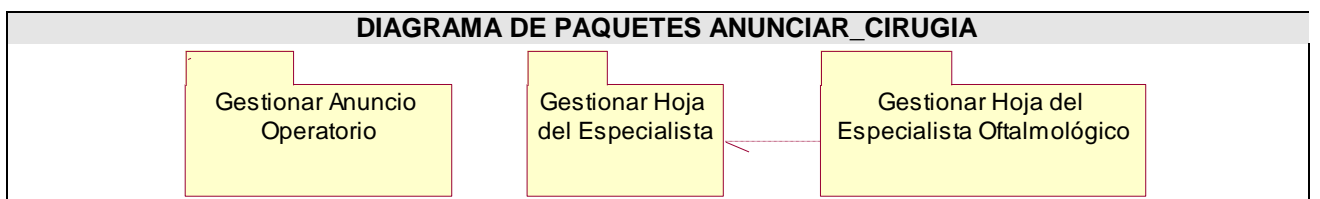


Figura 4: Diagrama de Paquetes ANUNCIAR_CIRUGÍA.

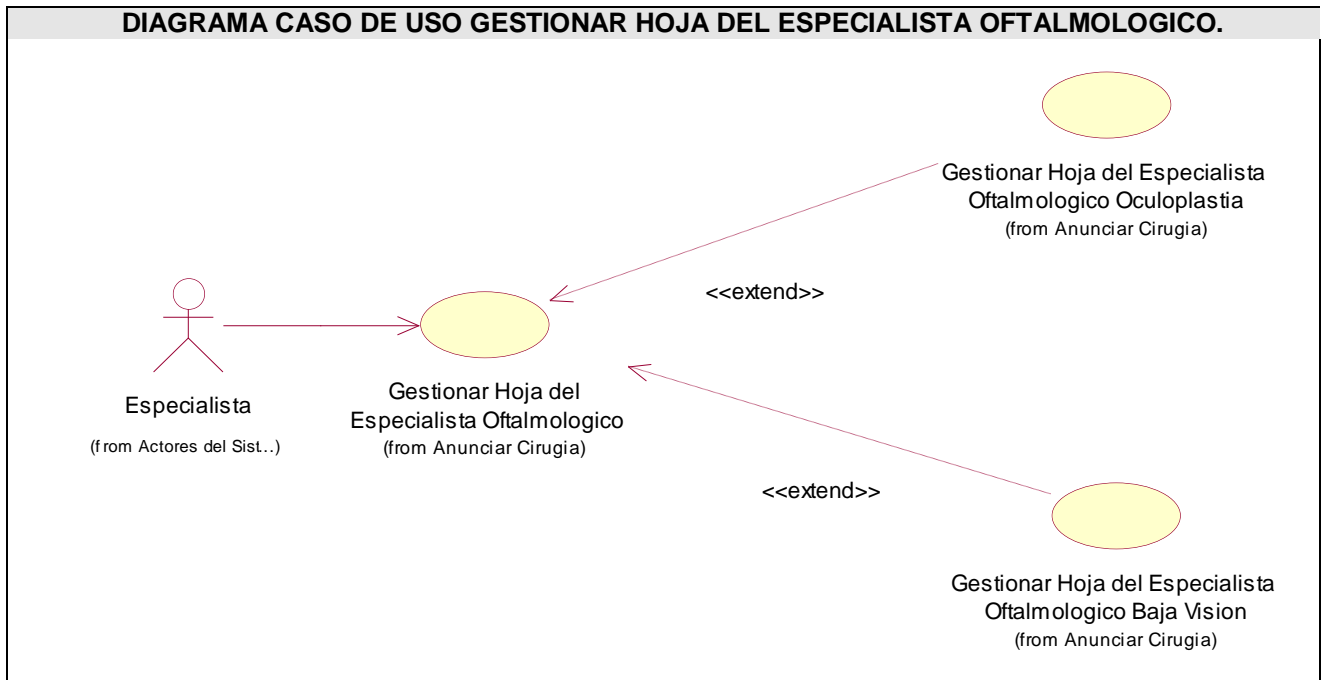


Figura 5: Diagrama del Caso de Uso Gestionar Hoja del Especialista Oftalmológico.

Casos de Uso por ciclos.

Código	Nombre de Caso de Uso	Paquete	Justificación de la selección.
CU-1	Gestionar Consulta Especialista Oftalmológica en Oculoplastia.	Anunciar Cirugía	Este caso de uso se decide desarrollar porque permite al Especialista guardar los datos de la consulta en la “hoja del especialista en Oculoplastia” de la Historia Clínica del paciente.
CU-2	Gestionar Consulta Especialista Oftalmológica en Baja Visión.	Anunciar Cirugía	Este caso de uso se decide desarrollar porque permite al Especialista guardar los datos de la consulta en la “hoja del especialista en Baja Visión” de la Historia Clínica del paciente.

Listado de Casos de Uso del sistema.

Tabla 9: Casos de Uso por Ciclos.

Caso de Uso	
CU_1	Gestionar Consulta Especialista Oftalmológica en Oculoplastia
Actores	Especialista (inicia)
Propósito	Permitir al Especialista guardar los datos de la consulta en la “hoja del especialista en Oculoplastia” de la HC del paciente.
Resumen	El Caso de Uso se inicia cuando el Especialista selecciona la opción de “Hoja del Especialista Oculoplastia” en el menú principal de trabajo y habiendo sido localizada esta, asienta los datos de la consulta (interrogatorio, examen físico, complementarios y valoración anestesia) dando el criterio de acto quirúrgico o no, finalizando el caso de uso.
Referencias	R1, CU_incluido: Buscar paciente (M_Inscripción_Admisión), Buscar Funcionarios (M_Configuración), Buscar Material Gastable (M_Farmacia).
Precondiciones	Anestesista autenticado satisfactoriamente por el sistema. Haberse ejecutado el previamente el caso de uso “Gestionar Consulta Especialista Oftalmológica”.
Poscondiciones	HC actualizada.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El especialista selecciona la opción “Hoja de Especialista en Oculoplastia” del menú principal de trabajo.	
	1.2 Muestra la interfaz de la hoja del especialista con las siguientes opciones: a) Crear Nueva Consulta. b) Modificar Consultas.
Sección: Crear Nueva Consulta	
Se invoca a la sección: “Crear Nueva Consulta” del caso de uso: “Gestionar Consulta Especialista”.	
4. El especialista completa los datos de la consulta.	
4. El especialista da la impresión diagnóstica.	
5. El especialista plantea conducta a seguir del paciente.	
6. El especialista selecciona la opción “Guardar”.	
	7.1 El sistema registra los datos de la consulta.
	7.2 Finaliza el caso de uso.
Sección “Modificar Consultas”	
Se invoca a la sección: “Modificar Consulta” del caso de uso: “Gestionar Consulta del especialista”.	
Puntos de Extensión	
7. El especialista no realiza la impresión	

Capítulo 2: Características del Sistema

diagnóstica. Ir al paso 6.	
Flujos Alternos	
Sección: “Crear Nueva Consulta”	
Línea 5.1: El sistema muestra el siguiente mensaje de error: “Error: El sistema no puede guardar el registro; existen datos incorrectos”	
Sección: “Modificar Consulta”	
Línea 6.3: El sistema no permite editar los datos de la consulta y muestra el siguiente mensaje: “Información: El tiempo requerido para editar los datos a expirado” .	
Línea 8.1 El sistema muestra el siguiente mensaje de error: “Error: El sistema no puede guardar el registro; existen datos incorrectos”	
Prioridad:	Critico

Tabla 10: Caso de Uso del Sistema: Gestionar Consulta Especialista Oftalmológica en Oculoplastia.

Caso de Uso	
CU_2	Gestionar Consulta Especialista Oftalmológica en Baja Visión
Actores	Especialista (inicia)
Propósito	Permitir al Especialista guardar los datos de la consulta en la “hoja del especialista en Baja Visión” de la HC del paciente.
Resumen	El Caso de Uso se inicia cuando el Especialista selecciona la opción de “Hoja del Especialista Baja Visión” en el menú principal de trabajo y habiendo sido localizada esta, asienta los datos de la consulta (interrogatorio, examen físico, complementarios y valoración anestesia) dando el criterio de acto quirúrgico o no, finalizando el caso de uso.
Referencias	R2, CU_incluido: Buscar paciente (M_Inscripción_Admisión), Buscar Funcionarios (M_Configuración), Buscar Material Gastable (M_Farmacia).
Precondiciones	Anestesiista autenticado satisfactoriamente por el sistema. Haberse ejecutado el previamente el caso de uso “Gestionar Consulta Especialista Oftalmológica”.
Poscondiciones	HC actualizada.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El especialista selecciona la opción “Hoja de Especialista en Baja Visión” del menú principal de trabajo.	
	1.2 Muestra la interfaz de la hoja del especialista con las siguientes opciones: a) Crear Nueva Consulta. b) Modificar Consultas.
Sección: Crear Nueva Consulta	
Se invoca a la sección: “Crear Nueva Consulta” del caso de uso: “Gestionar Consulta Especialista”.	
4. El especialista completa los datos de la consulta.	

4. El especialista da la impresión diagnóstica.	
5. El especialista plantea conducta a seguir del paciente.	
6. El especialista selecciona la opción "Guardar".	
	7.1 El sistema registra los datos de la consulta.
	7.2 Finaliza el caso de uso.
Sección "Modificar Consultas"	
Se invoca a la sección: "Modificar Consulta" del caso de uso: "Gestionar Consulta del especialista".	
Puntos de Extensión	
7. El especialista no realiza la impresión diagnóstica. Ir al paso 6.	
Flujos Alternos	
Sección: "Crear Nueva Consulta"	
Línea 5.1: El sistema muestra el siguiente mensaje de error: "Error: El sistema no puede guardar el registro; existen datos incorrectos"	
Sección: "Modificar Consulta"	
Línea 6.3: El sistema no permite editar los datos de la consulta y muestra el siguiente mensaje: "Información: El tiempo requerido para editar los datos a expirado" .	
Línea 8.1 El sistema muestra el siguiente mensaje de error: "Error: El sistema no puede guardar el registro; existen datos incorrectos"	
Prioridad:	Critico

Tabla 11: Caso de Uso del Sistema: Gestionar Consulta Especialista Oftalmológica en Baja Visión.

En este capítulo se definieron los conceptos fundamentales para una mejor comprensión del problema, definiéndose además los requerimientos funcionales y no funcionales, realizándose una descripción detallada de las acciones a realizar por cada uno de los actores del sistema siendo indentificados estos previamente.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

A partir del diagrama de casos de uso, obtenido como parte del flujo de trabajo de requerimientos, se puede establecer una traza con la disciplina que continúa el ciclo de vida de RUP, esta se encarga de describir en términos de clases cómo debe funcionar cada caso de uso y las colaboraciones que existen entre las clases definidas.

De esta forma se obtienen los diagramas de clases de análisis correspondientes a cada caso de uso, conformando las realizaciones de los mismos conjuntamente con los diagramas de interacción hasta llegar a refinar este análisis en el diseño esperado para la aplicación a desarrollar, con el objetivo de transformar los requerimientos, al diseño que el sistema debe tener y con la finalidad de adaptarlo al ambiente de implementación, trazando su funcionamiento. Para el desarrollo satisfactorio de esta disciplina la metodología propone el uso de patrones con el objetivo de lograr una arquitectura robusta para el sistema en cuestión.

3.1 Diagramas de clases de análisis.

Los diagramas de clases de análisis, expresan la definición y relación entre las clases, están compuestos por clases que pueden ser de tres tipos fundamentales: interfaz, controladora y entidad.

Las clases interfaz son las que se encargan de brindar un entorno gráfico amigable y entendible para que el usuario interactúe con el sistema de manera que modelan la interacción entre el sistema y sus actores.

Las clases controladoras se encargan de captar toda la información que es enviada a través de las clases interfaz y guardar esos datos a través de métodos en la Base de Datos (BD), coordinando la realización de uno o unos pocos casos de uso así como las actividades de los objetos que implementan la funcionalidad del caso de uso.

Las clases entidades vienen como resultado de las entidades definidas en el negocio estudiado y definido, aunque puede suceder que algunas de estas desaparezcan y surjan nuevas entidades, generalmente modelan información que posee larga vida y que es a menudo persistente, estas clases son las futuras tablas de la BD.

Los diagramas de clases de análisis además contienen relaciones que evidencian la navegabilidad y la dependencia entre clases, de esta forma se identifican relaciones de asociación, generalización, especialización, entre otras.

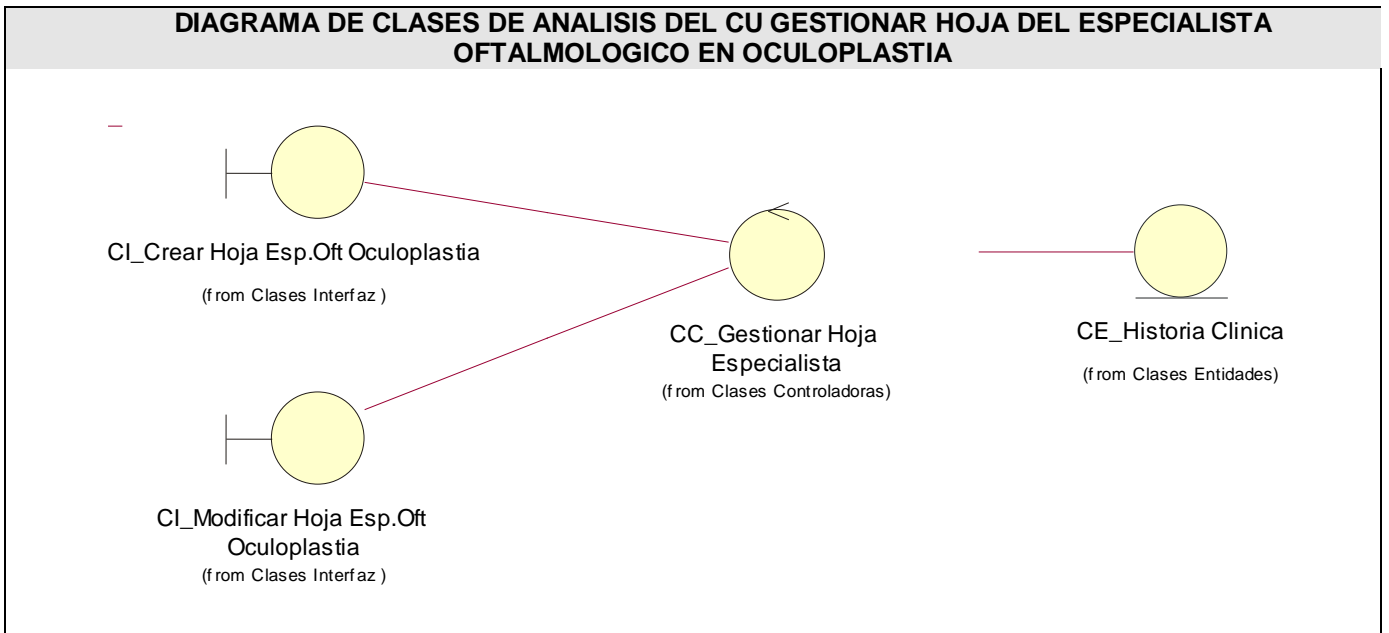


Figura 6: Diagrama de clases de análisis del CU Gestionar Hoja del Especialista Oftalmológico en Oculoplastia.

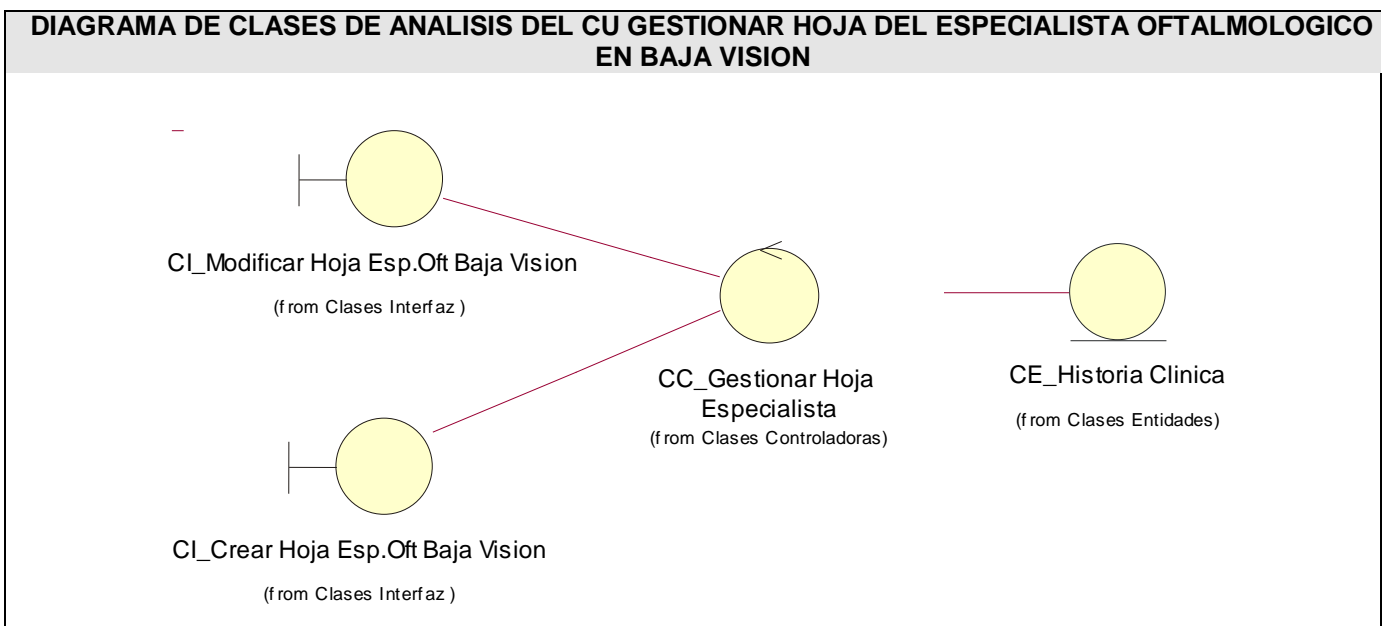


Figura 7: Diagrama de clases de análisis del CU Gestionar Hoja del Especialista Oftalmológico en Baja Visión.

3.2 Modelo de Diseño.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva como cumple el sistema sus objetivos. En este flujo se modela el sistema encontrando su forma

(incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos.

Además impone una estructura del sistema que debe esforzarse por conservar lo más fielmente posible cuando se da forma al sistema. El eje de esta fase es la elaboración de diagramas de interacción, que muestran gráficamente cómo los objetos se comunican entre ellos a fin de cumplir con los requerimientos. Estos diagramas permiten la realización de los diagramas de clases del diseño, los cuales resumen la definición de las clases que se pueden implementar en el software.

DIAGRAMA DE CLASES DE DISEÑO GESTIONAR HOJA DEL ESPECIALISTA OFTALMOLOGICO EN OCULOPLASTIA

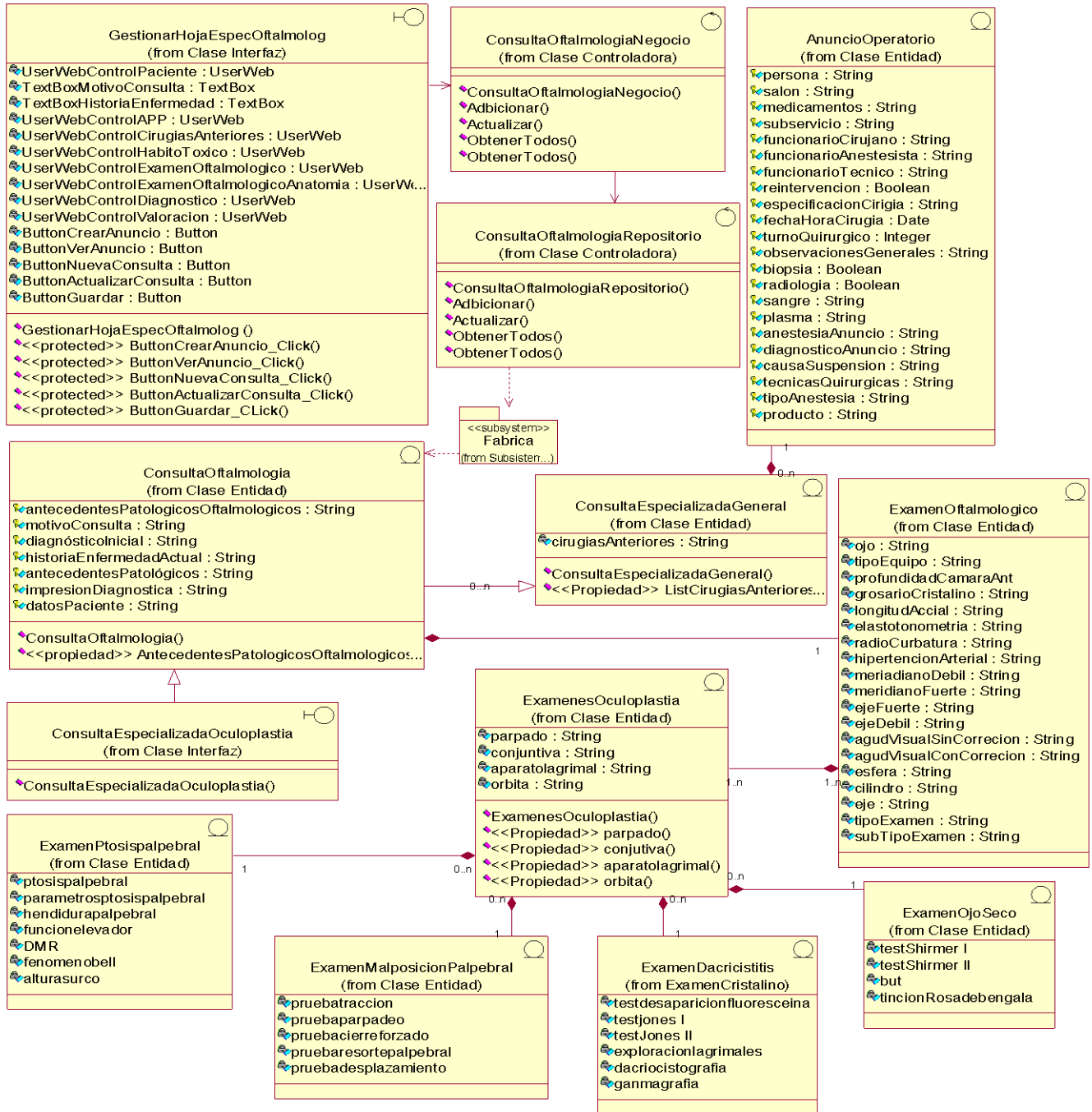


Figura 8: Diagrama de Clases de Diseño Gestionar Hoja del Especialista Oftalmológico en Oculoplastia.

DIAGRAMA DE CLASES DE DISEÑO GESTIONAR HOJA DEL ESPECIALISTA OFTALMOLOGICO EN BAJA VISIÓN

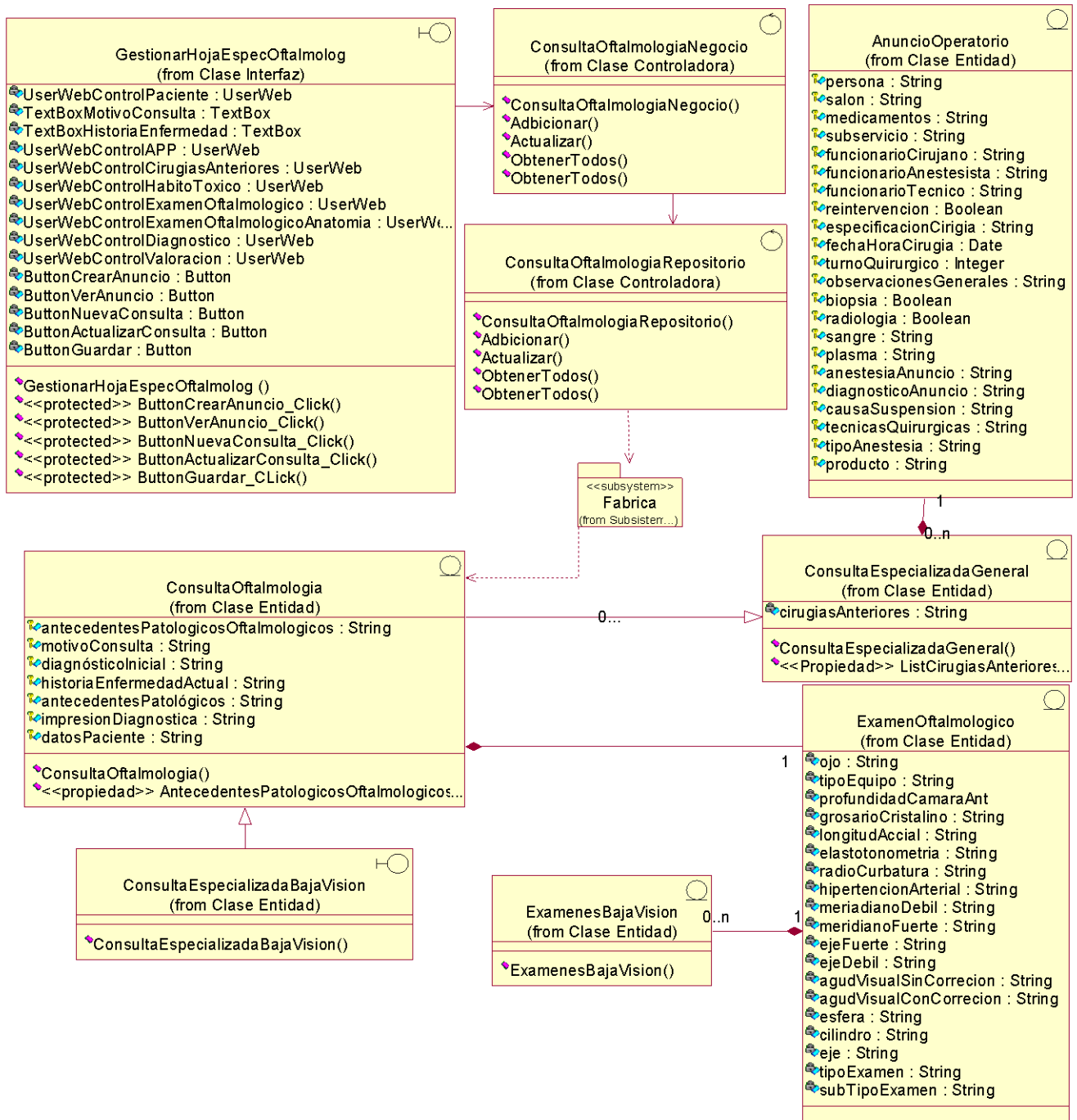


Figura 9: Diagrama de Clases de Diseño Gestionar Hoja del Especialista Oftalmológico en Baja Visión.

3.3 Patrones de Diseño.

Los patrones de diseño han adquirido gran popularidad entre investigadores y diseñadores de *software* orientado a objeto. La razón principal de su éxito radica en que estos forman parte de un vocabulario común de buenas soluciones perfectamente identificadas y aplicables a distintos problemas típicos de diseño que pueden encontrarse en diferentes contextos. Los patrones de diseño facilitan la reutilización del conocimiento experto como componentes de diseño y mejoran así la documentación, comprensión y comunicación del diseño final.

El objetivo de los patrones es agrupar una colección de soluciones de diseño que son validas en distintos contextos y que fueron aplicadas con éxito es anteriores ocasiones. Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

Los patrones son soluciones de sentido común que deberían formar parte del conocimiento de un diseñador experto. Además facilitan la comunicación entre diseñadores, pues establecen un marco de referencia.

Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución. Además, los patrones de diseño, también ayudarán a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces.

Patrones de creación: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades.

El patrón de diseño definido por los arquitectos para el desarrollo del sistema es el patrón de diseño *Abstract Factory*.

Abstract Factory: Es un patrón de creación que proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. Declara una interfaz de operaciones que crean productos abstractos.

Para el trabajo con este patrón se diseñó un subsistema llamado *Abstract Factory* al que acceden todas las clases controladoras del negocio a través de un repositorio que interactúa con las funciones fundamentales del subsistema *Abstract Factory* llamando los métodos esenciales de las interface de

cada una de las entidades del negocio. Estas interfaces contienen métodos abstractos y se encargan de realizar las operaciones básicas sobre los datos, insertar, modificar, eliminar y seleccionar.

3.4 Subsistemas que interactúan.

Para el diseño de este módulo también se tuvo en cuenta los subsistemas con los cuales debe interactuar por lo que el diagrama de subsistemas del diseño de la aplicación quedó representado como se muestra en la figura 10.

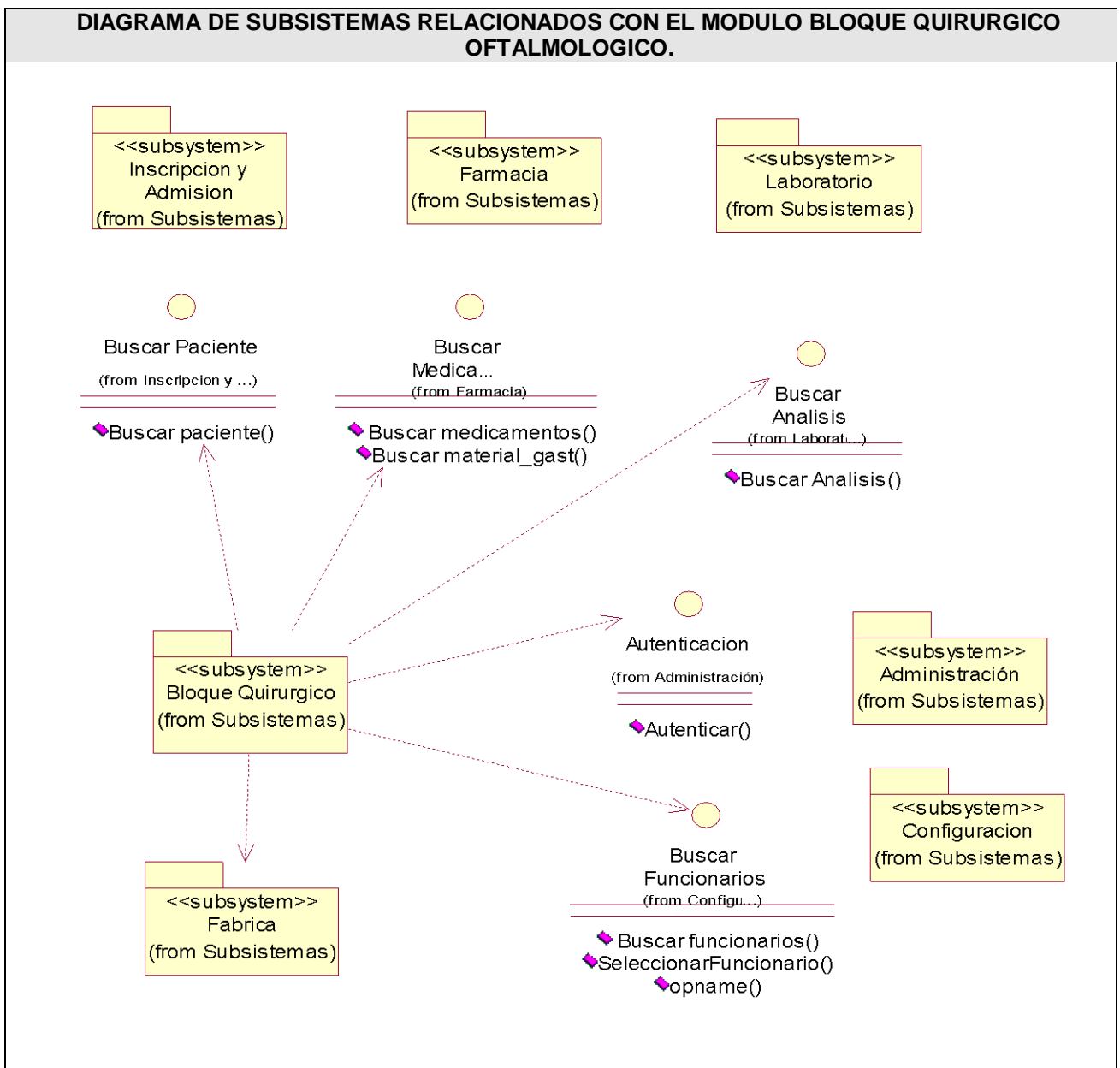


Figura 10: Diagrama de subsistemas relacionados con el Modulo Bloque Quirúrgico Oftalmológico.

3.5 Diagramas de Secuencia.

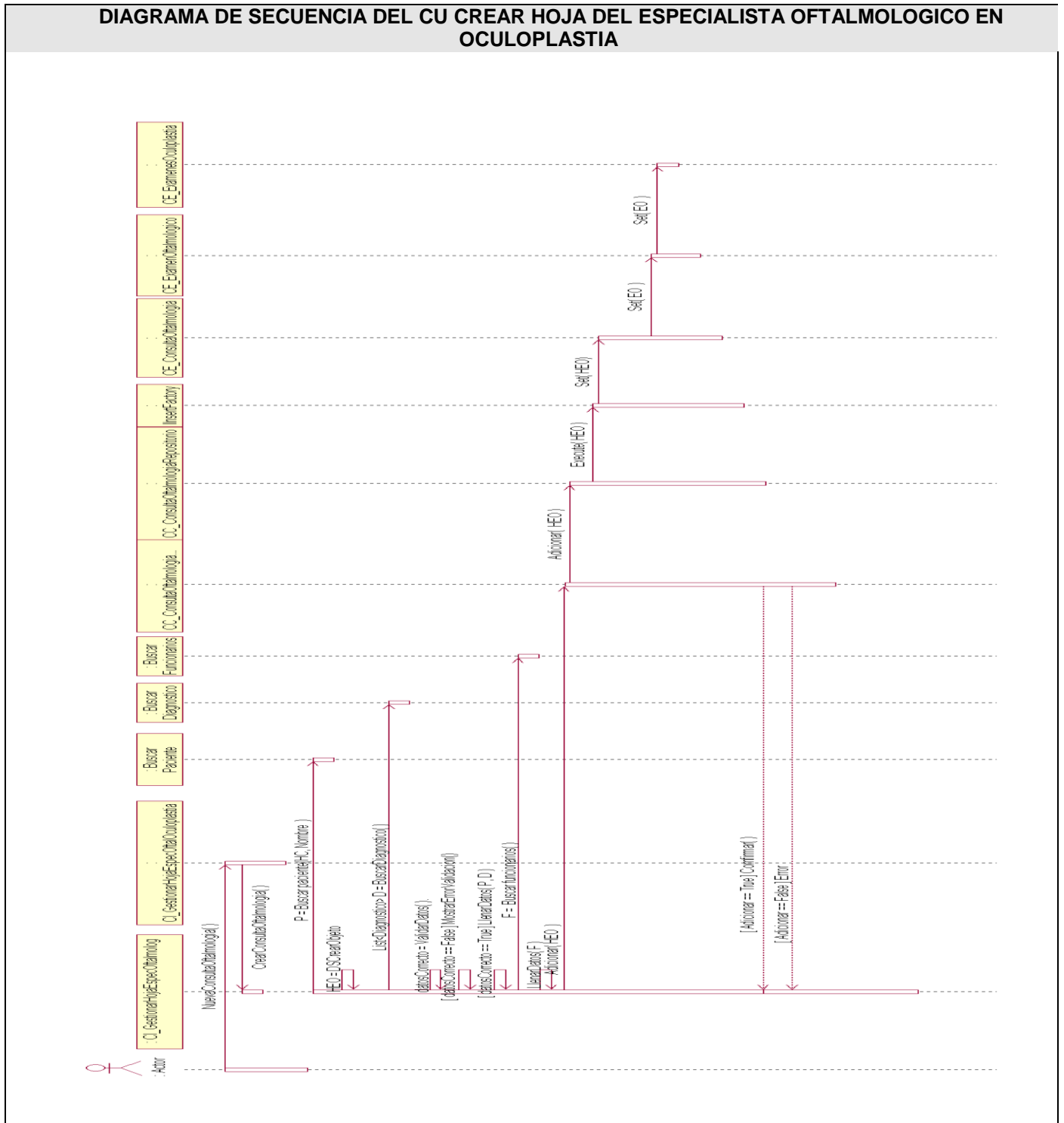


Figura 11: Diagrama de Secuencia del CU Crear Hoja del Especialista Oftalmológico en Oculoplastia.

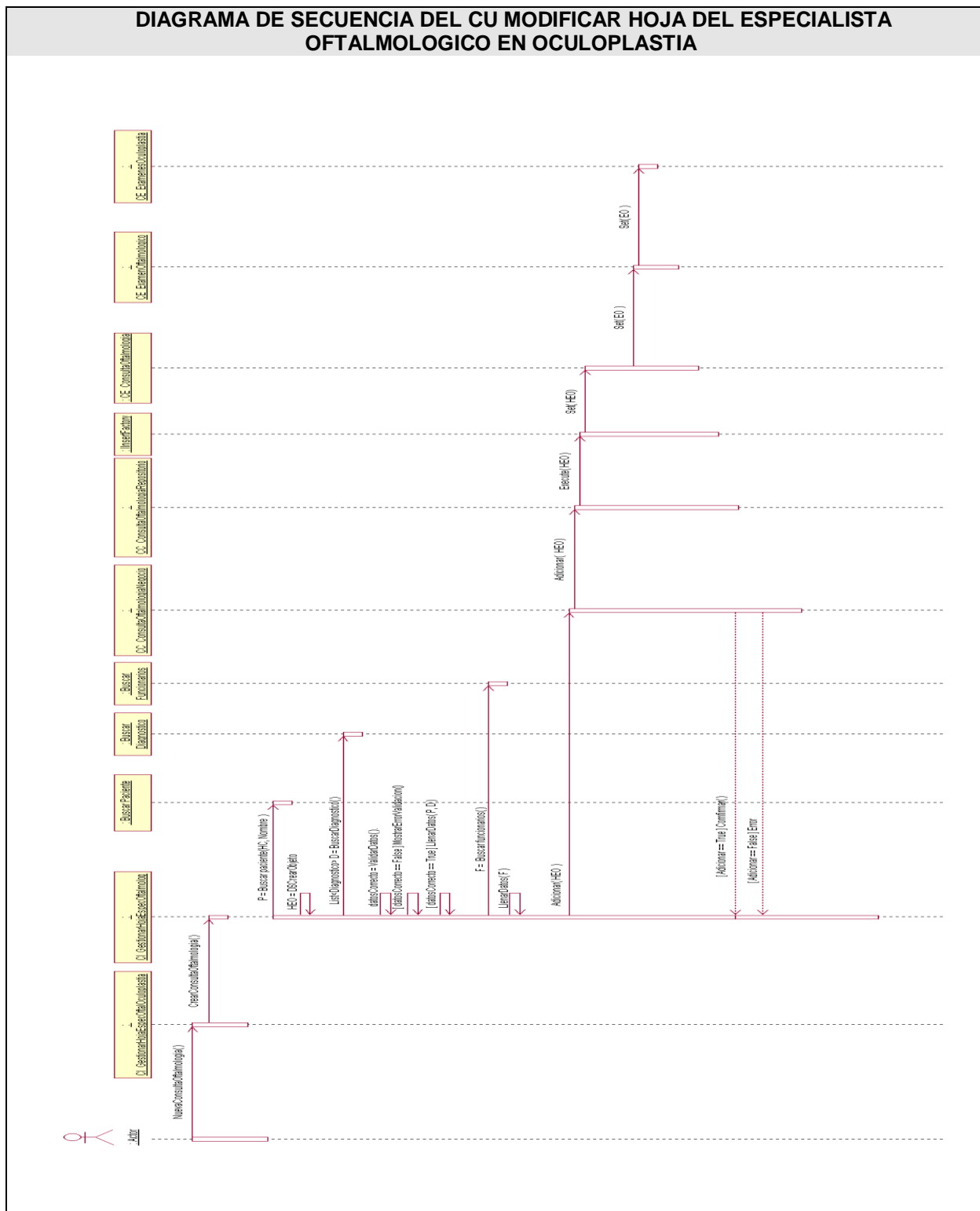


Figura 12: Diagrama de Secuencia del CU Modificar Hoja del Especialista Oftalmológico en Oculoplastia.

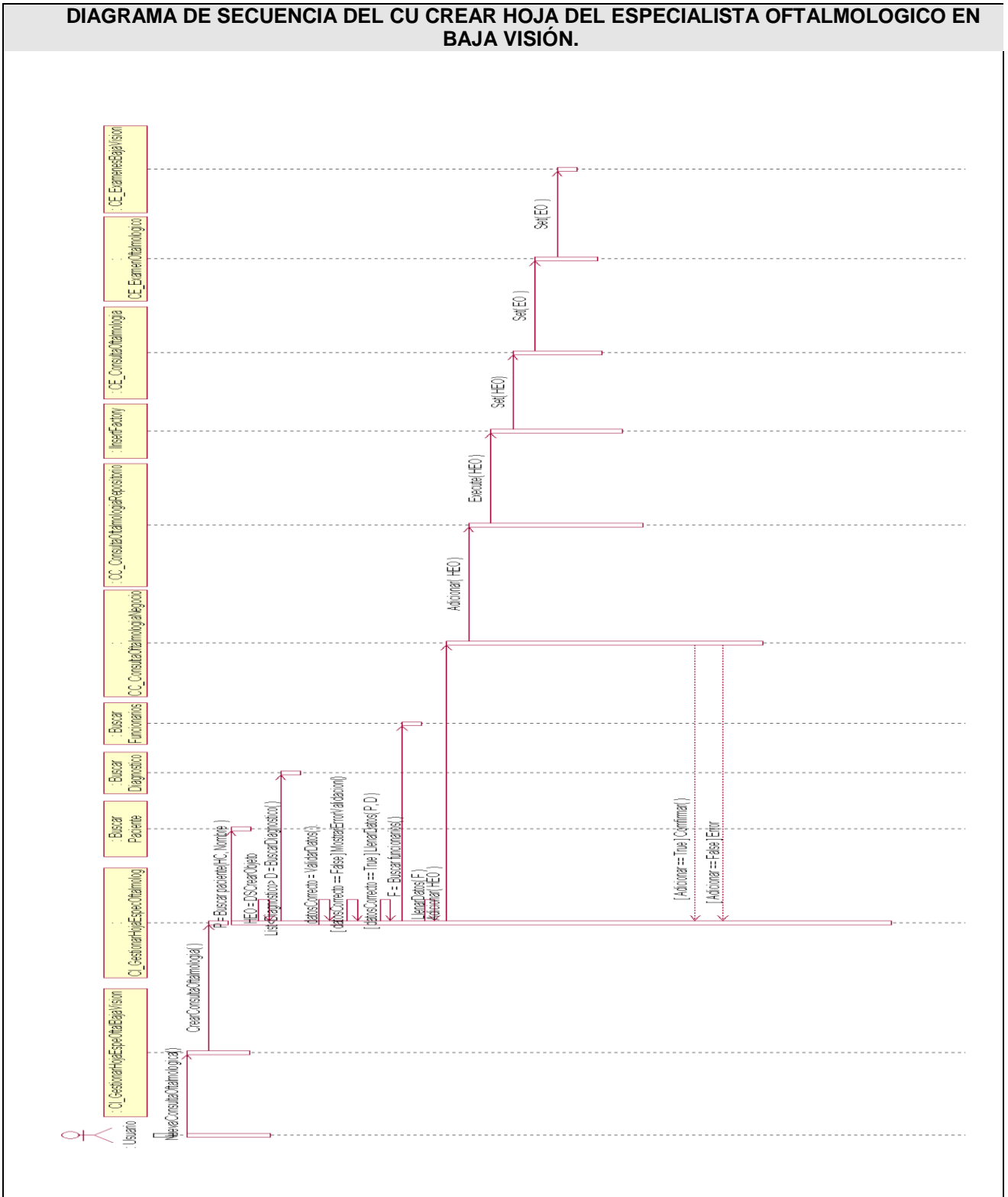


Figura 13: Diagrama de secuencia del CU Crear Hoja del Especialista Oftalmológico en Baja Visión.

DIAGRAMA DE SECUENCIA DEL CU MODIFICAR HOJA DEL ESPECIALISTA OFTALMOLOGICO EN BAJA VISION.

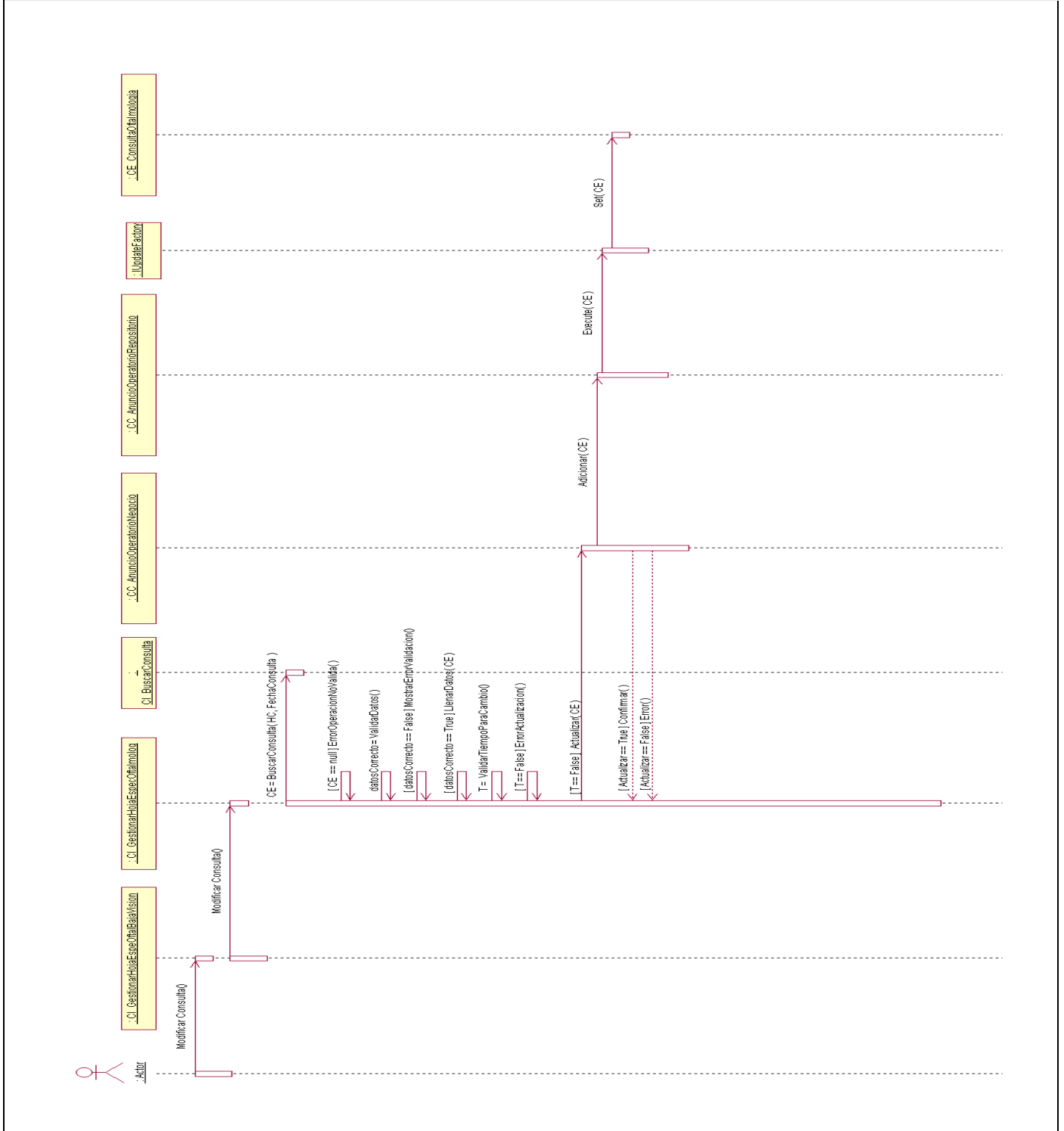


Figura 14: Diagrama de secuencia del CU Crear Hoja del Especialista Oftalmológico en Baja Visión.

3.6 Descripción de las nuevas clases u operaciones necesarias.

3.6.1 Entidades del negocio.

Estas entidades fueron creadas siguiendo el patrón de mapeo de datos, que propone crear un objeto por cada entidad persistente.

Nombre: PtoisPalpebral	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
hendiduraPalpebral	Int32
fenomenoBell	Boolean
dmr	Int32
alturaSurco	Int32
funcionElevador	Int32
Para cada responsabilidad:	
Nombre:	PtoisPalpebral
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	hendiduraPalpebral
Descripción:	Propiedad para mostrar y modificar la hendiduraPalpebral
Nombre:	fenomenoBell
Descripción:	Propiedad para mostrar y modificar el fenomenoBell
Nombre:	dmr
Descripción:	Propiedad para mostrar y modificar el dmr
Nombre:	alturaSurco
Descripción:	Propiedad para mostrar y modificar la alturaSurco
Nombre:	funcionElevador
Descripción:	Propiedad para mostrar y modificar la funcionElevador

Tabla 12: Clase entidad PtoisPalpebral.

Nombre: Dacricistitis	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
testDesaparicionFluoresceina	Boolean
testJonesi	Boolean
testJonesii	Boolean
exploracionViasLagrimales	Boolean

dacriocistografia	Boolean
ganmagrafia	Boolean
Para cada responsabilidad:	
Nombre:	Dacricistitis
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	testDesaparicionFluoresceina
Descripción:	Propiedad para mostrar y modificar el testDesaparicionFluoresceina
Nombre:	testJonesi
Descripción:	Propiedad para mostrar y modificar el testJonesi
Nombre:	testJonesii
Descripción:	Propiedad para mostrar y modificar el testJonesii
Nombre:	exploracionViasLagrimales
Descripción:	Propiedad para mostrar y modificar la exploracionViasLagrimales
Nombre:	dacriocistografia
Descripción:	Propiedad para mostrar y modificar la dacriocistografía
Nombre:	ganmagrafia
Descripción:	Propiedad para mostrar y modificar la ganmagrafía

Tabla 13: Clase entidad Dacricistitis.

Nombre: MalposicionPalpebral	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
pruebaTraccion	Boolean
pruebaParpadeo	Boolean
pruebaCierreForzado	Boolean
pruebaResortePalpebral	Boolean
pruebaDesplazamiento	Boolean
Para cada responsabilidad:	
Nombre:	MalposicionPalpebral
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	pruebaTraccion
Descripción:	Propiedad para mostrar y modificar la pruebaTraccion
Nombre:	pruebaParpadeo
Descripción:	Propiedad para mostrar y modificar la pruebaParpadeo
Nombre:	pruebaCierreForzado
Descripción:	Propiedad para mostrar y modificar la pruebaCierreForzado
Nombre:	pruebaResortePalpebral
Descripción:	Propiedad para mostrar y modificar la pruebaResortePalpebral
Nombre:	pruebaDesplazamiento

Descripción:	Propiedad para mostrar y modificar la pruebaDesplazamiento
---------------------	--

Tabla 14: Clase entidad MalposicionPalpebral.

Nombre: OjoSeco	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
testShirmeri	Int32
testShirmerii	Int32
but	Int32
tincionRosaBengala	Boolean
tincionFluoresceina	Boolean
Para cada responsabilidad:	
Nombre:	OjoSeco
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	testShirmeri
Descripción:	Propiedad para mostrar y modificar el testShirmeri
Nombre:	testShirmerii
Descripción:	Propiedad para mostrar y modificar el testShirmerii
Nombre:	but
Descripción:	Propiedad para mostrar y modificar el but
Nombre:	tincionRosaBengala
Descripción:	Propiedad para mostrar y modificar la tincionRosaBengala
Nombre:	tincionFluoresceina
Descripción:	Propiedad para mostrar y modificar la tincionFluoresceina

Tabla 15: Clase entidad OjoSeco.

Nombre: Refraccion	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
agudezaVisualSinCorreccion	Double
esfera	Double
cilindro	Double
eje	Double
agudezaVisualConCorreccion	Double
add	Double
dp	Double
Para cada responsabilidad:	
Nombre:	Refraccion
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos

Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	AgudezaVisualSinCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualSinCorreccion
Nombre:	Esfera
Descripción:	Propiedad para mostrar y modificar la esfera
Nombre:	Cilindro
Descripción:	Propiedad para mostrar y modificar el cilindro
Nombre:	Eje
Descripción:	Propiedad para mostrar y modificar el eje
Nombre:	AgudezaVisualConCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualConCorreccion
Nombre:	Add
Descripción:	Propiedad para mostrar y modificar la add
Nombre:	Dp
Descripción:	Propiedad para mostrar y modificar la dp

Tabla 16: Clase entidad Refraccion.

Nombre: ExamenRefraccionDinamica	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
agudezaVisualSinCorreccion	Double
esfera	Double
cilindro	Double
eje	Double
agudezaVisualConCorreccion	Double
add	Double
dp	Double
optotipologmar	Boolean
snellen	Boolean
pruebaFinal	Boolean
Para cada responsabilidad:	
Nombre:	Refraccion
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	AgudezaVisualSinCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualSinCorreccion
Nombre:	Esfera
Descripción:	Propiedad para mostrar y modificar la esfera
Nombre:	Cilindro
Descripción:	Propiedad para mostrar y modificar el cilindro
Nombre:	Eje

Descripción:	Propiedad para mostrar y modificar el eje
Nombre:	AgudezaVisualConCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualConCorreccion
Nombre:	Add
Descripción:	Propiedad para mostrar y modificar la add
Nombre:	Dp
Descripción:	Propiedad para mostrar y modificar la dp
Nombre:	Optotipologmar
Descripción:	Propiedad para mostrar y modificar el optotipologmar
Nombre:	Snellen
Descripción:	Propiedad para mostrar y modificar la snellen
Nombre:	PruebaFinal
Descripción:	Propiedad para mostrar y modificar la pruebaFinal

Tabla 17: Clase entidad ExamenRefraccionDinamica.

Nombre: ExamenRefraccionIndicada	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenesOftalmologicos	Int32
agudezaVisualConCristales	Double
esfera	Double
cilindro	Double
eje	Double
agudezaVisualConCorreccion	Double
add	Double
dp	Double
Para cada responsabilidad:	
Nombre:	Refraccion
Descripción:	Constructor de la clase.
Nombre:	IdExamenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExamenesOftalmologicos
Nombre:	AgudezaVisualConCristales
Descripción:	Propiedad para mostrar y modificar la AgudezaVisualConCristales
Nombre:	Esfera
Descripción:	Propiedad para mostrar y modificar la esfera
Nombre:	Cilindro
Descripción:	Propiedad para mostrar y modificar el cilindro
Nombre:	Eje
Descripción:	Propiedad para mostrar y modificar el eje
Nombre:	AgudezaVisualConCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualConCorreccion
Nombre:	Add
Descripción:	Propiedad para mostrar y modificar la add

Nombre:	Dp
Descripción:	Propiedad para mostrar y modificar la dp

Tabla 18: Clase entidad ExamenRefraccionIndicada.

Nombre: ExamenLensometro	
Tipo de clase: Entidad	
Atributo	Tipo
idExámenesOftalmologicos	Int32
esfera	Double
cilindro	Double
eje	Double
add	Double
dp	Double
Para cada responsabilidad:	
Nombre:	Refraccion
Descripción:	Constructor de la clase.
Nombre:	IdExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExámenesOftalmologicos
Nombre:	Esfera
Descripción:	Propiedad para mostrar y modificar la esfera
Nombre:	Cilindro
Descripción:	Propiedad para mostrar y modificar el cilindro
Nombre:	Eje
Descripción:	Propiedad para mostrar y modificar el eje
Nombre:	Add
Descripción:	Propiedad para mostrar y modificar la add
Nombre:	Dp
Descripción:	Propiedad para mostrar y modificar la dp

Tabla 19: Clase entidad ExamenLensometro.

Nombre: ExámenesOftalmologicos	
Tipo de clase: Entidad	
Atributo	Tipo
idExámenesOftalmologicos	Int32
idPersona	Int32
idOjos	Int32
ioIMaster	Boolean
fechaHoraExamen	DateTime
Para cada responsabilidad:	
Nombre:	ExámenesOftalmologicos
Descripción:	Constructor de la clase.
Nombre:	IdExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idExámenesOftalmologicos
Nombre:	IdPersona

Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	IdOjos
Descripción:	Propiedad para mostrar y modificar el idOjos
Nombre:	IdMaster
Descripción:	Propiedad para mostrar y modificar el idMaster
Nombre:	FechaHoraExamen
Descripción:	Propiedad para mostrar y modificar la fechaHoraExamen

Tabla 20: Clase entidad ExámenesOftalmologicos.

Nombre: AntecedentesPatologicosOftalmologicos	
Tipo de clase: Entidad	
Atributo	Tipo
idAntecedentesOftalmologicos	Int32
numeroConsulta	Int32
idPersona	Int32
Para cada responsabilidad:	
Nombre:	AntecedentesPatologicosOftalmologicos
Descripción:	Constructor de la clase.
Nombre:	IdAntecedentesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el idAntecedentesOftalmologicos
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona

Tabla 21: Clase entidad AntecedentesPatologicosOftalmologicos.

Nombre: ExámenesAnexos	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenOftalmologicoAnatomia	Int32
idParámetroAnatomia	Int32
Para cada responsabilidad:	
Nombre:	ExámenesAnexos
Descripción:	Constructor de la clase.
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia
Nombre:	IdParámetroAnatomia
Descripción:	Propiedad para mostrar y modificar el idParámetroAnatomia

Tabla 22: Clase entidad ExámenesAnexos.

Nombre: ExamenFondoOjo	
Tipo de clase: Entidad	
Atributo	Tipo

idExamenOftalmologicoAnatomia	Int32
idParámetroFondoOjo	Int32
Para cada responsabilidad:	
Nombre:	ExamenFondoOjo
Descripción:	Constructor de la clase.
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia
Nombre:	IdParámetroFondoOjo
Descripción:	Propiedad para mostrar y modificar el idParámetroFondoOjo

Tabla 23: Clase entidad ExamenFondoOjo.

Nombre: ExamenMotilidadOcular	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenMotilidad	Int32
idExamenOftalmologicoAnatomia	Int32
idOtopia	Int32
Para cada responsabilidad:	
Nombre:	ExamenMotilidadOcular
Descripción:	Constructor de la clase.
Nombre:	IdExamenMotilidad
Descripción:	Propiedad para mostrar y modificar el idExamenMotilidad
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia
Nombre:	idOtopia
Descripción:	Propiedad para mostrar y modificar el idOtopia

Tabla 24: Clase entidad ExamenMotilidadOcular.

Nombre: ExamenReflejoPupilar	
Tipo de clase: Entidad	
Atributo	Tipo
idExamenOftalmologicoAnatomia	Int32
idReflejoPupilar	Int32
Para cada responsabilidad:	
Nombre:	ExamenReflejoPupilar
Descripción:	Constructor de la clase.
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia
Nombre:	IdReflejoPupilar
Descripción:	Propiedad para mostrar y modificar el idReflejoPupilar

Tabla 25: Clase entidad ExamenReflejoPupilar.

Nombre: ExamenSegmentoAnterior

Tipo de clase: Entidad	
Atributo	Tipo
idExamenOftalmologicoAnatomia	Int32
idParámetroSegmentoAnterior	Int32
Para cada responsabilidad:	
Nombre:	ExamenSegmentoAnterior
Descripción:	Constructor de la clase.
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia
Nombre:	IdParámetroSegmentoAnterior
Descripción:	Propiedad para mostrar y modificar el idParámetroSegmentoAnterior

Tabla 26: Clase entidad ExamenSegmentoAnterior.

Nombre: ExamenVitreo	
Tipo de clase: Entidad	
Atributo	Tipo
idParámetroVitreo	Int32
idExamenOftalmologicoAnatomia	Int32
Para cada responsabilidad:	
Nombre:	ExamenVitreo
Descripción:	Constructor de la clase.
Nombre:	IdParámetroVitreo
Descripción:	Propiedad para mostrar y modificar el idParámetroVitreo
Nombre:	IdExamenOftalmologicoAnatomia
Descripción:	Propiedad para mostrar y modificar el idExamenOftalmologicoAnatomia

Tabla 27: Clase entidad ExamenVitreo.

Nombre: ParametroSegmentoAnterior	
Tipo de clase: Entidad	
Atributo	Tipo
idParámetroSegmentoAnterior	Int32
descripcion	String
Para cada responsabilidad:	
Nombre:	ParametroSegmentoAnterior
Descripción:	Constructor de la clase.
Nombre:	IdParámetroSegmentoAnterior
Descripción:	Propiedad para mostrar y modificar el idParámetroSegmentoAnterior
Nombre:	Descripcion
Descripción:	Propiedad para mostrar y modificar la descripcion

Tabla 28: Clase entidad ParametroSegmentoAnterior.

Nombre: AntecedenteOftalmologico	
Tipo de clase: Entidad	

Atributo	Tipo
idAntecedenteOftalmologico	Int32
descripcion	String
Para cada responsabilidad:	
Nombre:	AntecedenteOftalmologico
Descripción:	Constructor de la clase.
Nombre:	IdAntecedenteOftalmologico
Descripción:	Propiedad para mostrar y modificar el idAntecedenteOftalmologico
Nombre:	Descripción
Descripción:	Propiedad para mostrar y modificar la descripción

Tabla 29: Clase entidad AntecedenteOftalmologico.

Nombre: HabitoToxico	
Tipo de clase: Entidad	
Atributo	Tipo
idHabitoToxico	Int32
descripción	String
Para cada responsabilidad:	
Nombre:	HabitoToxico
Descripción:	Constructor de la clase.
Nombre:	IdHabitoToxico
Descripción:	Propiedad para mostrar y modificar el idHabitoToxico
Nombre:	Descripción
Descripción:	Propiedad para mostrar y modificar la descripción

Tabla 30: Clase entidad HabitoToxico.

Clases EP (Entidades Personalizadas).

Estas clases fueron creadas con el objetivo de agrupar las entidades mapeadas, creando las relaciones que existen entre ellas. Formando de esta manera las principales entidades del negocio con las que interactúa el sistema.

Nombre: EPDacricistitis	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
dacricistitis	Dacricistitis
Para cada responsabilidad:	
Nombre:	EPDacricistitis
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	dacricistitis
Descripción:	Propiedad para mostrar y modificar la dacricistitis

Tabla 31: Clase entidad EPDacricistitis.

Nombre: EPOjoSeco	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
ojoseco	OjoSeco
Para cada responsabilidad:	
Nombre:	EPOjoSeco
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	ojoseco
Descripción:	Propiedad para mostrar y modificar la ojoseco

Tabla 32: Clase entidad EPOjoSeco.

Nombre: EPPtosisPalpebral	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
ptosispalpebral	PtosisPalpebral
Para cada responsabilidad:	
Nombre:	EPPtosisPalpebral
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	PtosisPalpebral
Descripción:	Propiedad para mostrar y modificar la ptosispalpebral

Tabla 33: Clase entidad EPPtosisPalpebral.

Nombre: EPMalposicionPalpebral	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
malposicion	MalposicionPalpebral
Para cada responsabilidad:	
Nombre:	EPMalposicionPalpebral
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	Malposicion
Descripción:	Propiedad para mostrar y modificar la malposición

Tabla 34: Clase entidad EPMalposicionPalpebral.

Nombre: EPRefraccion

Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
refraccion	Refraccion
Para cada responsabilidad:	
Nombre:	ERefraccion
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	Refraccion
Descripción:	Propiedad para mostrar y modificar la refracción

Tabla 35: Clase entidad ERefraccion.

Nombre: EPLensometro	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
examenLensometro	ExamenLensometro
Para cada responsabilidad:	
Nombre:	EPLensometro
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	ExamenLensometro
Descripción:	Propiedad para mostrar y modificar la examenLensometro

Tabla 36: Clase entidad EPLensometro.

Nombre: ERefraccionIndicada	
Tipo de clase: Entidad	
Atributo	Tipo
exámenesOftalmologicos	ExamenOftalmologico
examenRefraccionIndicada	ExamenRefraccionIndicada
Para cada responsabilidad:	
Nombre:	ERefraccionIndicada
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar el exámenesOftalmologicos
Nombre:	ExamenRefraccionIndicada
Descripción:	Propiedad para mostrar y modificar la examenRefraccionIndicada

Tabla 37: Clase entidad ERefraccionIndicada.

Nombre: ERefraccionDinamica	
Tipo de clase: Entidad	
Atributo	Tipo

exámenesOftalmologicos	ExamenOftalmologico
examenRefraccionDinamica	ExamenRefraccionDinamica
Para cada responsabilidad:	
Nombre:	ERefraccionIndicada
Descripción:	Constructor de la clase.
Nombre:	ExámenesOftalmologicos
Descripción:	Propiedad para mostrar y modificar los exámenesOftalmologicos
Nombre:	ExamenRefraccionDinamica
Descripción:	Propiedad para mostrar y modificar el examenRefraccionDinamica

Tabla 38: Clase entidad ERefraccionDinamica.

3.6.2 Clases Controladoras.

Nombre: DacricistitisRepositorio	
Tipo de clase: Controladora	
Atributo	Tipo
factory	IDomainObjectFactory<Dacricistitis>
Para cada responsabilidad:	
Nombre:	DacricistitisRepositorio
Descripción:	Constructor de la clase.
Nombre:	ObtenerUno
Descripción:	Requiere de un parámetro de tipo Dacricistitis y devuelve la consulta especializada oftalmológica correspondiente a los datos entrados.
Nombre:	ObtenerTodos
Descripción:	Requiere de un parámetro de tipo Dacricistitis y devuelve una lista de consultas especializada oftalmológica correspondientes a los datos entrados.
Nombre:	Adicionar
Descripción:	Requiere de un parámetro de tipo Dacricistitis, el cual sera guardado en base de datos.
Nombre:	Actualizar
Descripción:	Requiere de un parámetro de tipo Dacricistitis, el cual sera actualizado en base de datos.
Nombre:	Eliminar
Descripción:	Requiere de un parámetro de tipo Dacricistitis, el cual sera suprimido de base de datos.

Tabla 39: Clase controladora DacricistitisRepositorio.

Nombre: DacricistitisSelectionFactory	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	DacricistitisSelectionFactory
Descripción:	Constructor de la clase.
Nombre:	Execute
Descripción:	Devuelve una lista de Dacricistitis, requiere de tres parámetros el helper, el domainObjectFactory y la entidad.

Tabla 40: Clase controladora DacricistitisSelectionFactory.

Nombre: DacricistitisInsertFactory	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	DacricistitisInsertFactory
Descripción:	Constructor de la clase.
Nombre:	Execute
Descripción:	Inserta en la BD un objeto de tipo Dacricistitis, requiere del parámetro helper

Tabla 41: Clase controladora DacricistitisInsertFactory.

Nombre: DacricistitisUpdateFactory	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	DacricistitisUpdateFactory
Descripción:	Constructor de la clase.
Nombre:	Execute
Descripción:	Actualiza en la BD el objeto de tipo Dacricistitis pasado por parámetro, requiere del helper.

Tabla 42: Clase controladora DacricistitisUpdateFactory.

Nombre: DacricistitisDeleteFactory	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	DacricistitisDeleteFactory
Descripción:	Constructor de la clase.
Nombre:	Execute
Descripción:	Suprime de la BD el objeto de tipo Dacricistitis pasado por parámetro, requiere del helper.

Tabla 43: Clase controladora DacricistitisDeleteFactory.

Nombre: EPDacricistitisNegocio	
Tipo de clase: Controladora	
Atributo	Tipo
dacricistitisRepositorio	EPDacricistitisRepositorio
Para cada responsabilidad:	
Nombre:	EPDacricistitisNegocio
Descripción:	Constructor de la clase.
Nombre:	ObtenerUno(<i>EPDacricistitis</i>)
Descripción:	Método que devuelve la consulta especializada oftalmológica correspondiente a los datos entrados.
Nombre:	ObtenerTodos(<i>EPDacricistitis</i>)
Descripción:	Método que devuelve una lista de consultas especializadas oftalmológicas correspondiente a los datos entrados.

Nombre:	Adicionar(<i>EPDacricistitis</i>)
Descripción:	Método que inserta en la BD la entidad pasada por parámetro.
Nombre:	Actualizar(<i>EPDacricistitis</i>)
Descripción:	Método que actualiza en la BD la entidad pasada por parámetro.
Nombre:	Eliminar(<i>EPDacricistitis</i>)
Descripción:	Método que suprime en la BD la entidad pasada por parámetro.

Tabla 44: Clase controladora EPDacricistitisNegocio.

2.6.3 Clases interfaces.

Nombre: bqo_ConsultaOftalmologiaOculoplastia	
Tipo de clase: Interfaz	
Atributo	Tipo
SSListConsulta	List<EPConsulta>
SSConsulta	EPConsultaEspecializadaOftalmologica
ControGuardar	Int32
ModificarConsulta	Boolean
PuedeCrearAnuncio	Boolean
SSDatosPersonas	DatosPersonas
SSEPAuncioOperatorioGeneral	EPAnuncioOperatorioGeneral
SSTokens	Tokens
autorizacion	SAAAutorizaWSDL
Para cada responsabilidad:	
Nombre:	Page_Load
Descripción:	Evento que se ejecuta cuando se carga la página. En el se inicializan las listas de los nomencladores a usar en esta página.
Nombre:	SeleccionarConsultaHistorial
Descripción:	Método para seleccionar una de las consultas que hay en el historial, se ejecuta en caso que se acceda a la página a través de la búsqueda de consulta o a partir de un anuncio operatorio.
Nombre:	RefrescarHistorialConsultas
Descripción:	Método para mostrar nuevamente la lista de consultas del historial.
Nombre:	LimpiarConsulta
Descripción:	Método para limpiar todos los registros referentes a la consulta. Se ejecuta cuando se selecciona otra consulta.
Nombre:	VerificarAnuncio
Descripción:	Método para verificar si el médico puede crear anuncios operatorios correspondientes a la consulta seleccionada.
Nombre:	GuardarDatos
Descripción:	Método para guardar toda la información recogida durante la consulta en una variable de sesión.
Nombre:	ControlesGuardados

Descripción:	Método para verificar si todos los controles fueron guardados satisfactoriamente.
Nombre:	RefrescarAnuncio
Descripción:	Método para mostrar nuevamente la lista de anuncios operatorios correspondientes a la consulta seleccionada.
Nombre:	InhabilitarControles
Descripción:	Método para inhabilitar los controles cuando haya pasado el tiempo de modificación de la consulta.
Nombre:	ButtonTab_Click
Descripción:	Evento que se ejecuta sobre los botones del tab. En este evento se activa el tab correspondiente al boton pulsado
Nombre:	Button_Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones de la cabecera. Según el botón que fue presionado se ejecutarán acciones como: Buscar un paciente para una nueva consulta. Adicionar la consulta actual a la BD Actualizar la consulta en la BD.
Nombre:	GridViewConsultas_RowCreated
Descripción:	Evento que se ejecuta cuando el gridview crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	GridViewConsultas_PageIndexChanging
Descripción:	Evento que se ejecuta al hacer click en el paginado del gridview. En el se le especifica como debe hacer el paginado.
Nombre:	GridViewConsultas_SelectedIndexChanging
Descripción:	Evento que se ejecuta al cambiar la selección del gridview. En el se le especifica como debe hacer la selección del nuevo elemento.
Nombre:	MultiViewConsulta_ActiveViewChanged
Descripción:	Evento que se ejecuta al cambiar de tab. En el se guardan los datos del tab que estaba activo.
Nombre:	GridViewAnuncio_RowCreated
Descripción:	Evento que se ejecuta cuando el gridview crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	Button2_Click
Descripción:	Evento que se ejecuta al hacer click en el button Exámenes especializados que redirecciona a nueva pagina donde se encuentran los exámenes especializados de la consulta en cuestión.

Tabla 45: Clase interfaz bqo_ConsultaOftalmologiaOculoplastia.

Nombre: ExámenesEspOculoplastia	
Tipo de clase: Interfaz	
Atributo	Tipo
SSDatosPersonas	DatosPersonas
ControlGuardar	Int32
examenRefraccion	EPRefraccion
examenOjoSeco	EPOjoSeco
examenMalposicionPalpebral	EPMalposicionPalpebral
examenDacricistitis	EPDacricistitis

examenRefraccionDinamica	EPRefraccionDinamica
examenLensometro	EPLensometro
examenPtosisPalpebral	EPPtosisPalpebral
examenRefraccionIndicada	EPRefraccionIndicada
VListaRefraccionDinamica	List<EPRefraccionDinamica>
VListaRefraccionIndicada	List<EPRefraccionIndicada>
SSExamenesMostrar	List<EPRefraccion>
VListaExamenes	List<EPRefraccion>
VListaExamenesPtosisPalpebral	List<EPPtosisPalpebral>
VListaExamenesOjoSeco	List<EPOjoSeco>
VListaExamenesMalposicionPalpebral	List<EPMalposicionPalpebral>
VListaExamenesDacricistitis	List<EPDacricistitis>
VListaExamenesLensometro	List<EPLensometro>
VSBandera	Int32
fechaExamen	DateTime
culture	System
autorizacion	SAAAAutorizaWSDL
SSTokens	Tokens
Para cada responsabilidad:	
Nombre:	Page_Load
Descripción:	Evento que se ejecuta cuando se carga la página. En el se inicializan las listas de los nomencladores a usar en esta página.
Nombre:	GridViewExamenes_RowCreated
Descripción:	Evento que se ejecuta cuando el GridView crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	GridViewExamenes_SelectedIndexChanging
Descripción:	Evento que se ejecuta al cambiar la selección del GridView. En el se le especifica como debe hacer la selección del nuevo elemento.
Nombre:	Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones del multiview. Según el botón que fue presionado se ejecutaran acciones como: Mostrar el control web que le corresponde de acuerdo con el Index del button.
Nombre:	GridViewExamenes_PageIndexChanging
Descripción:	Evento que se ejecuta al hacer click en el paginado del gridview. En el se le especifica como debe hacer el paginado.
Nombre:	ButtonCabecera1_Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones de la cabecera. Según el botón que fue presionado se ejecutaran acciones como: Nuevo examen. Guardar todos los controles.

Tabla 46: Clase interfaz ExamenesEspOculoplastia.

Nombre: bqo_ConsultaOftalmologiaBajaVision	
Tipo de clase: Interfaz	
Atributo	Tipo
SSListConsulta	List<EPConsulta>
SSConsulta	EPConsultaEspecializadaOftalmologica
ControGuardar	Int32
ModificarConsulta	Boolean
PuedeCrearAnuncio	Boolean
SSDatosPersonas	DatosPersonas
SSEPAuncioOperatorioGeneral	EPAnuncioOperatorioGeneral
SSTokens	Tokens
autorizacion	SAAAAutorizaWSDL
Para cada responsabilidad:	
Nombre:	Page_Load
Descripción:	Evento que se ejecuta cuando se carga la página. En el se inicializan las listas de los nomencladores a usar en esta página.
Nombre:	SeleccionarConsultaHistorial
Descripción:	Método para seleccionar una de las consultas que hay en el historial, se ejecuta en caso que se acceda a la página a través de la búsqueda de consulta o a partir de un anuncio operatorio.
Nombre:	RefrescarHistorialConsultas
Descripción:	Método para mostrar nuevamente la lista de consultas del historial.
Nombre:	LimpiarConsulta
Descripción:	Método para limpiar todos los registros referentes a la consulta. Se ejecuta cuando se selecciona otra consulta.
Nombre:	VerificarAnuncio
Descripción:	Método para verificar si el médico puede crear anuncios operatorios correspondientes a la consulta seleccionada.
Nombre:	GuardarDatos
Descripción:	Método para guardar toda la información recogida durante la consulta en una variable de sesión.
Nombre:	ControlesGuardados
Descripción:	Método para verificar si todos los controles fueron guardados satisfactoriamente.
Nombre:	RefrescarAnuncio
Descripción:	Método para mostrar nuevamente la lista de anuncios operatorios correspondientes a la consulta seleccionada.
Nombre:	InhabilitarControles
Descripción:	Método para inhabilitar los controles cuando haya pasado el tiempo de modificación de la consulta.
Nombre:	Click
Descripción:	Evento que se ejecuta sobre los botones del tab. En este evento se activa el tab correspondiente al boton pulsado
Nombre:	Button_Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones de la cabecera. Según el botón que fue presionado se ejecutarán acciones como: Buscar un paciente para una nueva consulta. Adicionar la consulta actual a la BD Actualizar la consulta en la BD.

Nombre:	GridViewConsultas_RowCreated
Descripción:	Evento que se ejecuta cuando el gridview crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	GridViewConsultas_PageIndexChanging
Descripción:	Evento que se ejecuta al hacer click en el paginado del gridview. En el se le especifica como debe hacer el paginado.
Nombre:	GridViewConsultas_SelectedIndexChanging
Descripción:	Evento que se ejecuta al cambiar la selección del gridview. En el se le especifica como debe hacer la selección del nuevo elemento.
Nombre:	MultiViewConsulta_ActiveViewChanged
Descripción:	Evento que se ejecuta al cambiar de tab. En el se guardan los datos del tab que estaba activo.
Nombre:	GridViewAnuncio_RowCreated
Descripción:	Evento que se ejecuta cuando el gridview crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	Button2_Click
Descripción:	Evento que se ejecuta al hacer click en el button Exámenes especializados que redirecciona a nueva pagina donde se encuentran los exámenes especializados de la consulta en cuestión.

Tabla 47: Clase interfaz bqo_ConsultaOftalmologiaBajaVision.

Nombre: ExámenesEspBajaVision	
Tipo de clase: Interfaz	
Atributo	Tipo
SSDatosPersonas	DatosPersonas
ControlGuardar	Int32
SSExámenesMostrar	List<EPExamenBajaVision>
VSListaExámenes	List<EPExamenBajaVision>
examenBajaVision	EPExamenBajaVision
VSBandera	Int32
fechaExamen	Date Time
culture	System
autorizacion	SAAAAutorizaWSDL
SSTokens	Tokens
Para cada responsabilidad:	
Nombre:	Page_Load
Descripción:	Evento que se ejecuta cuando se carga la página. En el se inicializan las listas de los nomencladores a usar en esta página.
Nombre:	GridViewExámenes_RowCreated
Descripción:	Evento que se ejecuta cuando el GridView crea cada una de las filas. En el se le especifica como debe crear esta fila.
Nombre:	GridViewExámenes_SelectedIndexChanging
Descripción:	Evento que se ejecuta al cambiar la selección del GridView. En el se le especifica como debe hacer la selección del nuevo elemento.
Nombre:	Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones del multiview. Según el botón que fue presionado se ejecutarán acciones como: Mostrar el control web que le corresponde de acuerdo con el Index del button.

Nombre:	GridViewExámenes_PageIndexChanging
Descripción:	Evento que se ejecuta al hacer click en el paginado del gridview. En el se le especifica como debe hacer el paginado.
Nombre:	ButtonCabecera2_Click
Descripción:	Evento que se ejecuta al hacer click sobre los botones de la cabecera. Según el botón que fue presionado se ejecutarán acciones como: Nuevo examen. Guardar todos los controles.

Tabla 48: Clase Interfaz ExámenesEspBajaVision.

3.7 Diseño de la BD.

A partir del diagrama de clases persistentes se realizaron una serie de transformaciones, derivadas de los procesos de normalización del diseño de la base de datos, y de la selección de una serie de tablas para la configuración para la conformación de un diagrama entidad relación homogéneo. También como parte de las relaciones que se presentan en una base de datos y que por lo general difieren de lo que se puede modelar a partir de clases, como el caso de una relación de mucho a mucho genera una tabla intermedia a nivel de base de datos y esto no se modela así en un diagrama de clases.

De esta forma se conforma el diseño físico del sistema de almacenamiento de la información correspondiente al bloque quirúrgico oftalmológico.

3.7.1 Diagrama Entidad Relación de la BD.

El modelo entidad relación se ha convertido en un estándar a la hora de modelar bases de datos relacionales por lo que se presenta este a continuación.

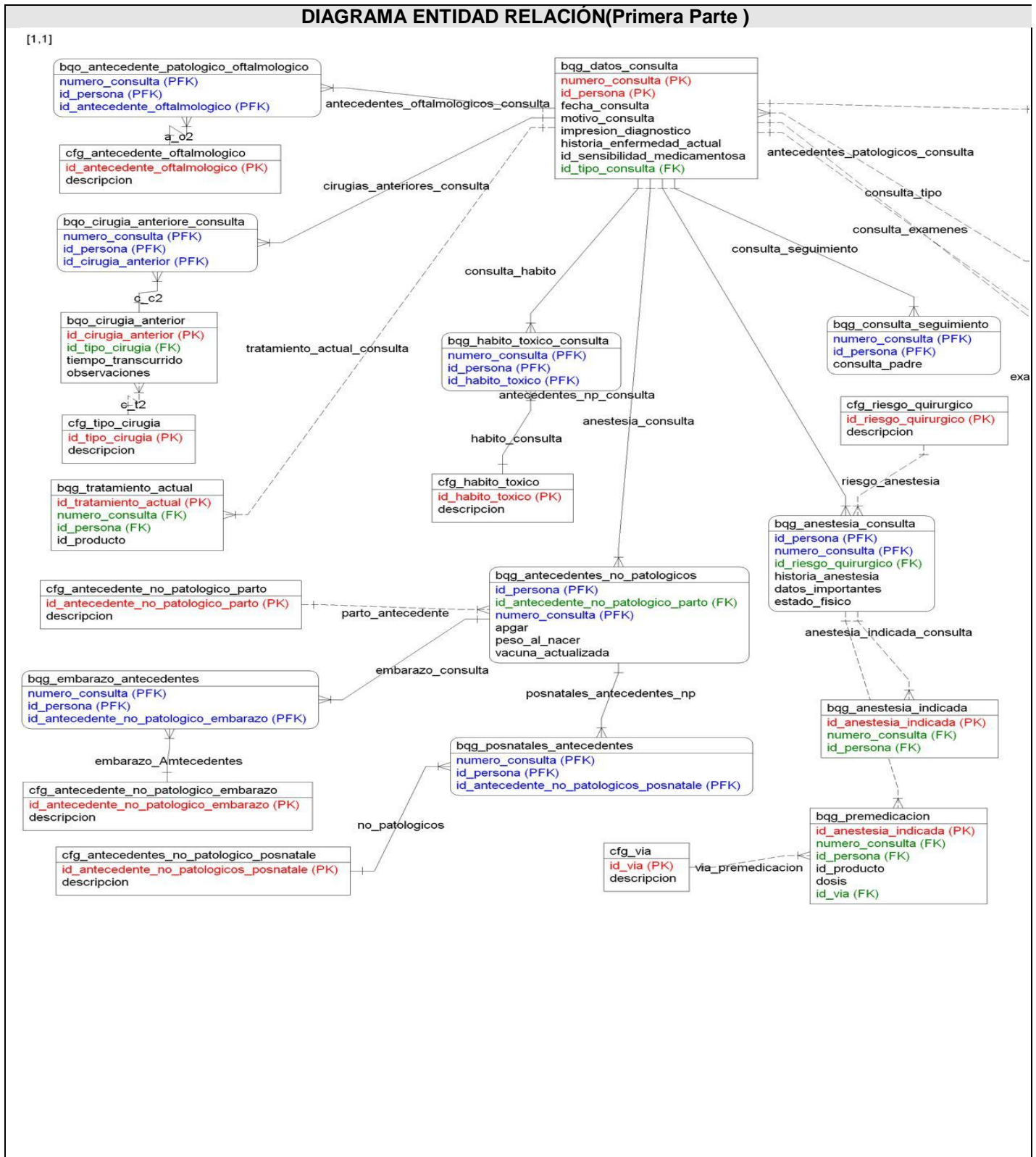


Figura 15: Diagrama Entidad Relación primera parte.

En este capítulo se realizó el análisis y diseño de la aplicación, para lograr esto se realizaron diagramas de clases de análisis, diagramas de clases del diseño así como la descripción textual de estas clases, y por último se presentó el diagrama entidad relación de la base de datos.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

La implementación y las pruebas son partes fundamentales en la elaboración de software, en este capítulo se llevan a hechos los elementos del diseño. Se elaboran además los diagramas de despliegue y componentes de la aplicación. Así como la realización de las pruebas para garantizar el resultado esperado.

La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación, se aplican diferentes técnicas de prueba a cada tipo de producto software.[36]

En ésta etapa intervienen solo los desarrolladores, los cuales actúan sobre un código fuente congelado, por lo que están restringidos a agregar nuevas funcionalidades y la imposibilidad de realizar grandes modificaciones a las interfaces gráficas de usuario. A pesar de que las pruebas no pueden asegurar la ausencia de defectos, ya que sólo se puede demostrar que existen defectos en el software, ya que es hecho por humanos, son parte fundamental antes de entregar el software final. [37]

La implementación es mucho más que conectar equipos electrónicos o software, fundamentalmente la implementación es la transformación del diseño en código ejecutable. De la calidad que tenga la implementación dependerá fundamentalmente la calidad de la aplicación en cuestión y aumentará o decaerá el prestigio de sus autores. La implementación también garantiza que el proceso de gestión de la información se realice de forma correcta y con los resultados esperados.

4.1 Diagrama de despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos (caso particular de un objeto).

En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).[38]

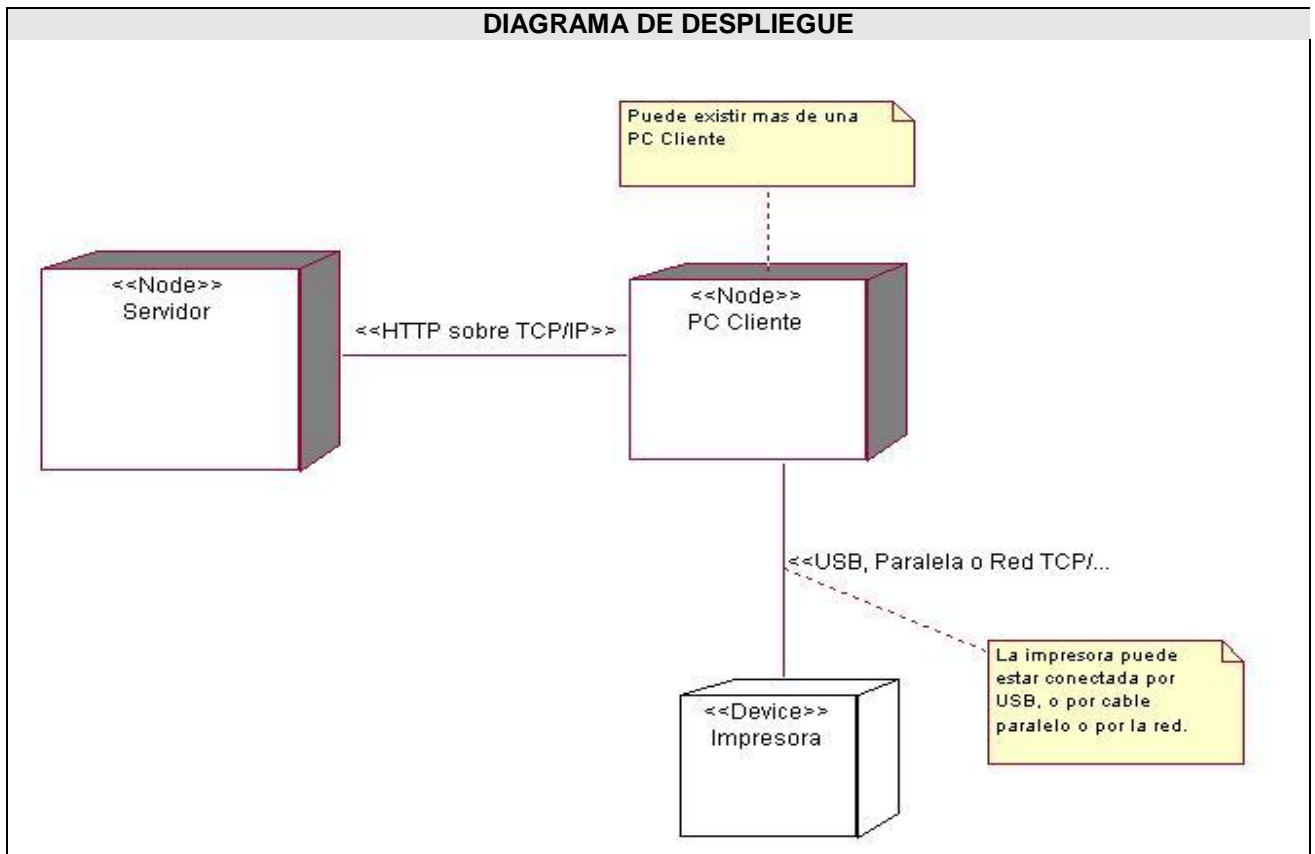


Figura 17: Diagrama de Despliegue.

4.2 Diagrama de componentes.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de

programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida.[39]

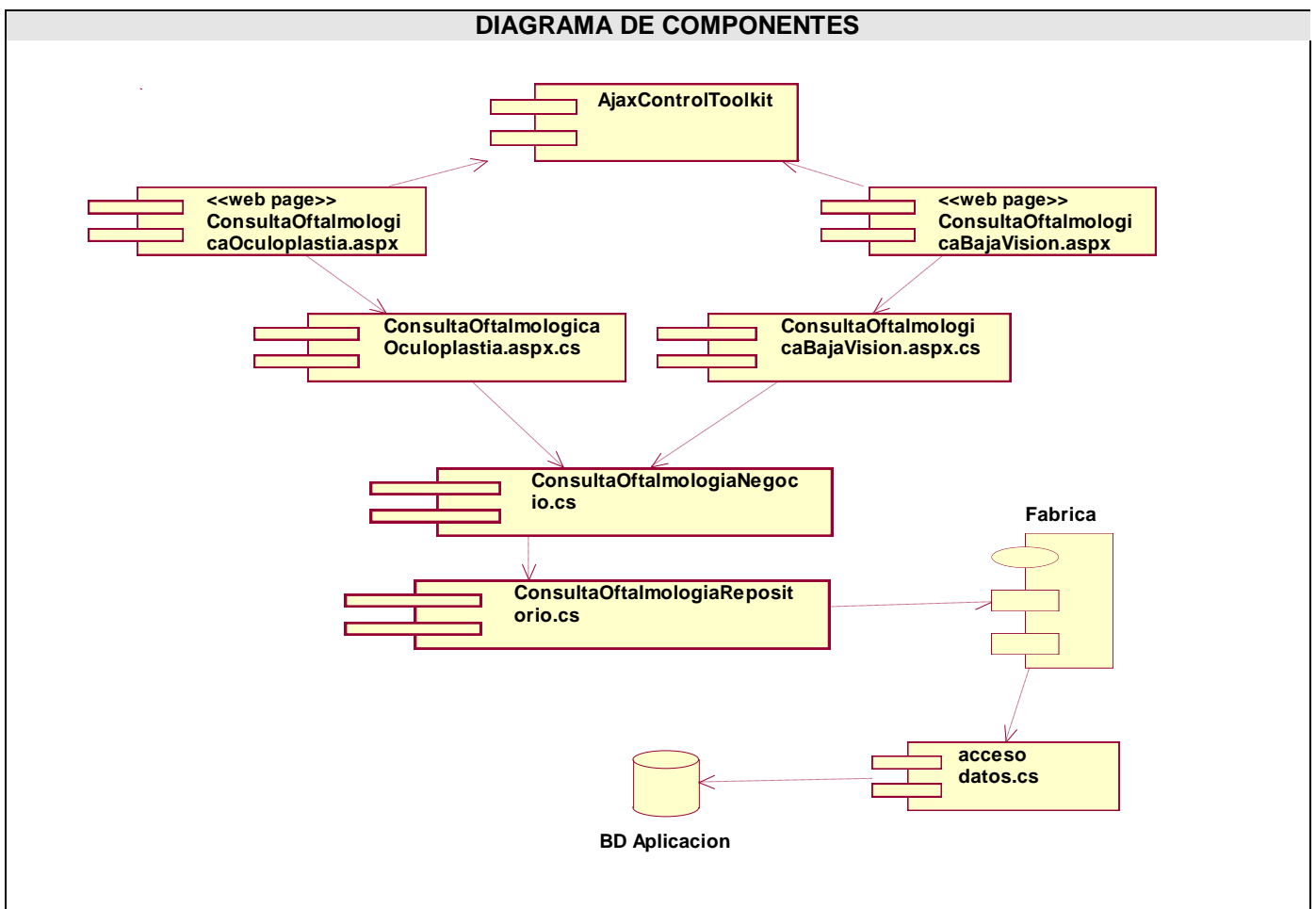


Figura 18: Diagrama de Componentes.

4.3 Modelo de pruebas.

Prueba de caja negra.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

4.3.1 Descripción de los casos de prueba de integración.

Caso de prueba	CU Gestionar Hoja del Especialista Oftalmológico en Baja Visión
Entrada	<p>El especialista autenticado y encargado de gestionar la Hoja del Especialista Oftalmológico en Baja Visión, presiona en el link Baja Visión, dando acceso solo a la opción: “Buscar paciente”. Luego de seleccionar o ingresar el paciente, aparece la interfaz correspondiente a la Hoja de la Consulta del Especialista Oftalmológico en Baja Visión.</p> <p>El especialista entra datos de la consulta incorrectos o algún otro carácter no válido en los exámenes. Se pulsa por último el botón Guardar.</p>
Resultados	El sistema emite un mensaje de error porque los datos de entrada son incorrectos y por dejar campos vacío que sean necesarios para dicha consulta.
Condiciones	Especialista autenticado satisfactoriamente por el sistema.

Tabla 49: Caso de prueba CU Gestionar Hoja del Especialista Oftalmológico en Baja Visión.

Caso de prueba	CU Gestionar Hoja del Especialista Oftalmológico en Oculoplastia
Entrada	<p>El especialista autenticado y encargado de gestionar la Hoja del Especialista Oftalmológico en Oculoplastia, presiona en el link Oculoplastia, dando acceso solo a la opción: “Buscar paciente”. Luego de seleccionar o ingresar el paciente, aparece la interfaz correspondiente a la Hoja de la Consulta del Especialista Oftalmológico en Oculoplastia.</p> <p>El especialista entra datos de la consulta incorrectos o algún otro carácter no válido en los exámenes. Se pulsa por último el botón Guardar.</p>

Resultados	El sistema emite un mensaje de error porque los datos de entrada son incorrectos y por dejar campos vacío que sean necesarios para dicha consulta.
Condiciones	Especialista autenticado satisfactoriamente por el sistema.

Tabla 50: Caso de prueba CU Gestionar Hoja del Especialista Oftalmológico en Oculoplastia.

En este capítulo se trata lo referente a la implementación del sistema así como las pruebas realizadas. Se realizó el diagrama de despliegue que muestra la distribución física del sistema así como el diagrama de componentes con sus dependencias. Además se describen los casos de pruebas de integración, realizados.

CONCLUSIONES

Al concluir el trabajo se han cumplido con el objetivo y las tareas planteadas, obteniendo los siguientes resultados:

- Se diseñó un sistema informático utilizando la metodología de ingeniería del software seleccionada.
- Se implementó un sistema informático que facilita la gestión de la información referente a los servicios oftalmológicos Oculoplastia y Baja Visión del Bloque Quirúrgico Oftalmológico.
- Se analizó la estrategia de integración de la solución con otros módulos o sistemas.

RECOMENDACIONES

Los autores del presente trabajo recomiendan:

- Hacer uso del trabajo de diploma como material de estudio para el futuro análisis y desarrollo de sistemas vinculados a la cirugía oftalmológica.
- Continuar con el desarrollo del sistema incorporándole otros servicios oftalmológicos.
- Desplegar el sistema en hospitales oftalmológicos del país.

REFERENCIAS BIBLIOGRAFICAS

1. *Miles de años*. [citado 2008 abril 5]; Disponible en: <http://www.latinsalud.com/articulos/00002.asp>.
2. Baranda., D.G.L.S.d. (2005) *Plan Funcional del Hospital de Villarrobledo*. **Volume**, 117
3. *Especialidades médicas*. [citado 2008 abril 8]; Disponible en: <http://www.paraqueestebien.com/especmedicas.htm>.
4. Azcárate, J.C.G.d. *DE LA HISTORIA CLÍNICA A LA HISTORIA DE SALUD ELECTRÓNICA (RESUMEN)* **Volume**, 46
5. *Historia y Funcionamiento de Internet*. MundoDigital [Monografía] 2005 [citado 2008 abril]; Disponible en: http://www.wikilearning.com/historia_de_internet-wkccp-3443-2.htm.
6. Peralta, A.d.P., *El servidor Web. Arquitectura y funcionamiento*. 2006.
7. INEI. *Arquitectura Cliente-Servidor*. 1997 [citado 2008 abril 2]; Disponible en: <http://www.inei.gov.pe/web/metodologias/attach/lib616/CAP0313.HTM>
8. CORPORATION, M. *.NET Framework*. 2006 [citado 2008 mayo 20]; Disponible en: <http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5>
9. Reynoso, B. *Arquitectura Orientada a Servicios (SOA)*. Universidad de Buenos Aires
10. *Beneficios de SOA*. 2006 [citado 2008 febrero 5]; Disponible en: <http://gluna.spaces.live.com/blog/cns!4519FFC35541B715!556.entry>.
11. XML. [citado 2007 diciembre 2]; Disponible en: <http://es.wikipedia.org/wiki/XML>
12. SOAP. [citado 2007 diciembre 2]; Disponible en: <http://es.wikipedia.org/wiki/SOAP>
13. XSL. [citado 2007 diciembre 3]; Disponible en: <http://es.wikipedia.org/wiki/XSL>.
14. XSLT. [citado 2007 diciembre]; Disponible en: <http://es.wikipedia.org/wiki/XSLT>.
15. *Que es ASP*. [citado 2007 diciembre]; Disponible en: <http://www.desarrolloweb.com/articulos/393.php>
16. *Qué es PHP*. [citado 2007 diciembre]; Disponible en: <http://www.desarrolloweb.com/articulos/392.php>.
17. *Perl*. . [citado 2008 marzo 29]; Disponible en: <http://es.wikipedia.org/wiki/Perl>
18. *Qué es Perl. Documento introductorio al lenguaje Perl*. . [citado 2008 marzo]; Disponible en: <http://www.desarrolloweb.com/articulos/541.php>

19. Merelo., J.J. *Generación de páginas Web usando XSLT y XML*. [citado; Disponible en: <http://geneura.ugr.es/~jmerelo/XSLT/>].
20. *Qué es Javascript*. . [citado 2008 mayo]; Disponible en: <http://www.desarrolloweb.com/articulos/25.php>
21. *Qué es JSP*. [citado 2007 diciembre 2]; Disponible en: <http://www.desarrolloweb.com/articulos/831.php>
22. [citado 2008 junio]; Disponible en: <http://www.lenguajes-de-programacion.com/programacion-estructurada.shtml>.
23. CORPORATION, M. *.NET Framework*. 2006 [citado 2008 Marzo 25]; Disponible en: <http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5>
24. *Arquitectura básica de la plataforma .Net. Descripción del Framework y sus principales componentes: Lenguajes, biblioteca de clases y CLR*. 2006 [citado 2008 mayo 21]; Disponible en: <http://www.desarrolloweb.com/articulos/1328.php>.
25. *¿Qué es SQL Server 2005?* [citado 2008 abril 23]; Disponible en: <http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.mspx>.
26. [citado 2008 abril1]; Disponible en: <http://tecnologia.glosario.net/terminos-tecnicos-internet/mysql-1192.html>.
27. *Microsoft. Scrum*. . [citado 2008 abril1]; Disponible en: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arg/heterodox.mspx
28. SANCHEZ, M.A.M. *Metodologías de desarrollo de software*.
29. *Clikear.com. UML*. [citado 2008 abril1]; Disponible en: <http://www.clikear.com/manuales/uml/introduccion.asp>.
30. Peñarroya, M. *AJAX, una técnica a usar con moderación*. [citado 2008 mayo 23]; Disponible en: http://www.geamarketing.com/articulos/que_es_ajax.php.
31. Traversa, E. *Qué Cosa Es Ajax Y Para Qué Es Bueno*. noviembre 1 2005 [citado 2008 mayo 24]; Disponible en: http://www.masternewmedia.org/es/tecnologías_diseño_de_interfaz/Ajax/Que_Cosa_Es_Ajax_Y_Para_Qué_Es_Bueno_20051101.htm.htm
32. *Herramientas CASE*. . [citado 2008 mayo]; Disponible en: http://es.wikipedia.org/wiki/Herramienta_CASE.
33. Caballero., A.V.I. *Una Herramienta CASE para ADOO: Rational Rose*.

34. IBM. . *Rational Rose Enterprise*. . [citado 2008 febrero 5]; Disponible en: http://www-306.ibm.com/software/info/ecatalog/es_ES/products/M221280M46834Z27.html
35. *OftalmoSalus, el software para oftalmología*. [citado 2008 marzo]; Disponible en: <http://www.ofthalmosalus.com/DefaultSalus.aspx>.
36. Usaola, D.M.P. *Mantenimiento Avanzado de Sistemas de Información. Pruebas del Software*[citado 2008 mayo 19; 37]. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/masi/doc/lec/parte5/polo-apuntesp5.pdf>
37. Jara, S.H.V. *13.8.2.6 Pruebas de Software terminado*. 2004 [citado febrero 15]; Disponible en: http://gbtcr.chileforge.cl/info_web/node139.html.
38. Vilas, A.F. 2001 [citado 2008 junio]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.
39. Vilas, A.F. . 2001 [citado 2008 junio]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.

BIBLIOGRAFÍA

- 1- *Arquitectura básica de la plataforma .Net. Descripción del Framework y sus principales componentes: Lenguajes, biblioteca de clases y CLR.* 2006
- 2- Azcárate, J.C.G.d. *DE LA HISTORIA CLÍNICA A LA HISTORIA DE SALUD ELECTRÓNICA (RESUMEN) Volume*, 46.
- 3- Baranda., D.G.L.S.d. (2005) *Plan Funcional del Hospital de Villarrobledo. Volume*, 117.
- 4- Beneficios de SOA. 2006
- 5- Caballero., A.V.I. *Una Herramienta CASE para ADOO: Rational Rose.*
- 6- [citado 2008 abril1]; Disponible en: <http://tecnologia.glosario.net/terminos-tecnicos-internet/mysql-1192.html>.
- 7- [citado 2008 junio]; Disponible en: <http://www.lenguajes-de-programacion.com/programacion-estructurada.shtml>.
- 8- *Clikear.com. UML.* [citado 2008 abril1]; Disponible en: <http://www.clikear.com/manuales/uml/introduccion.asp>.
- 9- CORPORATION, M. *.NET Framework.* 2006.
- 10- *Especialidades médicas.* [citado 2008 abril 8] Disponible en: <http://www.paraqueestebien.com/especmedicas.htm>.
- 11- Fernández, J. *Emetropía. Software General de Oftalmología.* 2007 [Citado 2008; Disponible en: <http://emetropia.com/descarga/Presentacion%20Emetropia.pps>].
- 12- *Historia y Funcionamiento de Internet.* Mundo Digital [Monografía] 2005
- 13- INEI. *Arquitectura Cliente-Servidor.* 1997
- 14- Jara, S.H.V. *13.8.2.6 Pruebas de Software terminado.* 2004
- 15- Merelo., J.J. *Generación de páginas Web usando XSLT y XML.*
- 16- Microsoft. *Scrum.*[citado; Disponible en: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.msp]
- 17- *OftalmoSalus, el software para oftalmología.*
- 18- Peñarroya, M. *AJAX, una técnica a usar con moderación.*
- 19- Peralta, A.d.P., *El servidor Web. Arquitectura y funcionamiento.* 2006.
- 20- *Perl.* . [citado 2008 marzo 29]; Disponible en: <http://es.wikipedia.org/wiki/Perl>
- 21- *Que es ASP.* [citado 2007 diciembre]; Disponible en: <http://www.desarrolloweb.com/articulos/393.php>
- 22- *Qué es Javascript.* [citado 2008 mayo]; Disponible en: <http://www.desarrolloweb.com/articulos/25.php>

- 23- *Qué es JSP*. [citado 2007 diciembre 2]; Disponible en: <http://www.desarrolloweb.com/articulos/831.php>
- 24- *Qué es Perl. Documento introductorio al lenguaje Perl*.
- 25- *Qué es PHP*. [citado 2007 diciembre]; Disponible en: <http://www.desarrolloweb.com/articulos/392.php>.
- 26- *¿Qué es SQL Server 2005?*
- 27- Reynoso, B. *Arquitectura Orientada a Servicios (SOA)*. Universidad de Buenos Aires
- 28- SANCHEZ, M.A.M. *Metodologías de desarrollo de software*.
- 29- SOAP. [citado 2007 diciembre 2]; Disponible en: <http://es.wikipedia.org/wiki/SOAP>
- 30- Traversa, E. *Qué Cosa Es Ajax Y Para Qué Es Bueno*. Publicado: noviembre 1 2005
- 31- Vilas, A.F. 2001 [citado 2008 junio]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.
- 32- 39. Vilas, A.F. 2001 [citado 2008 junio]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.
- 33- XML. [citado 2007 diciembre 2]; Disponible en: <http://es.wikipedia.org/wiki/XML>
- 34- XSL. [citado 2007 diciembre 3]; Disponible en: <http://es.wikipedia.org/wiki/XSL>.
- 35- XSLT. [citado 2007 diciembre]; Disponible en: <http://es.wikipedia.org/wiki/XSLT>.

ANEXOS

Anexo 1

Diagrama de actividades por Casos de Uso de Negocio.

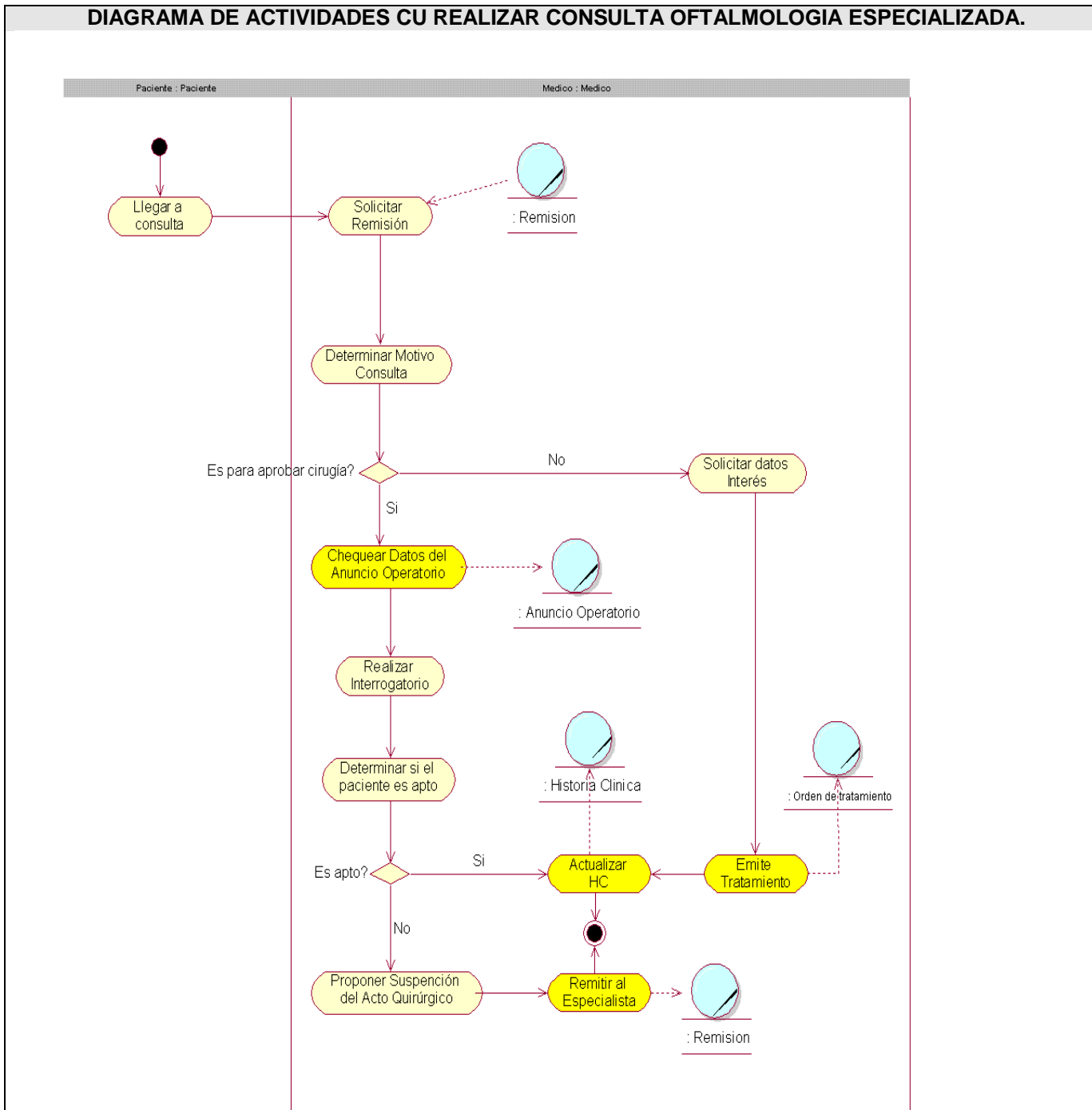


Figura 19: Diagrama de actividades del CUN Realizar Consulta Oftalmología Especializada.

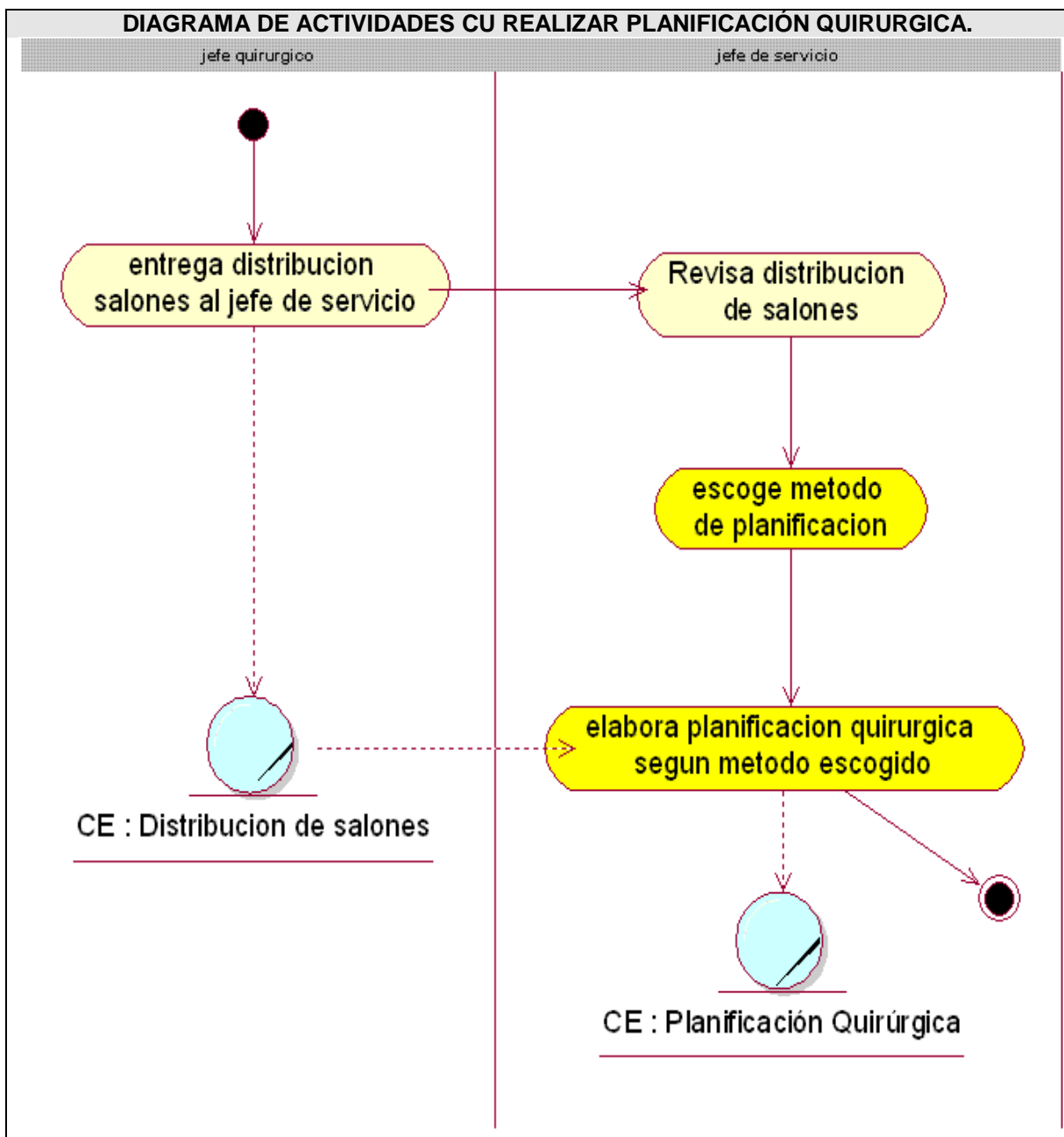


Figura 20: Diagrama de actividades del CUN Realizar Planificación Quirúrgica.

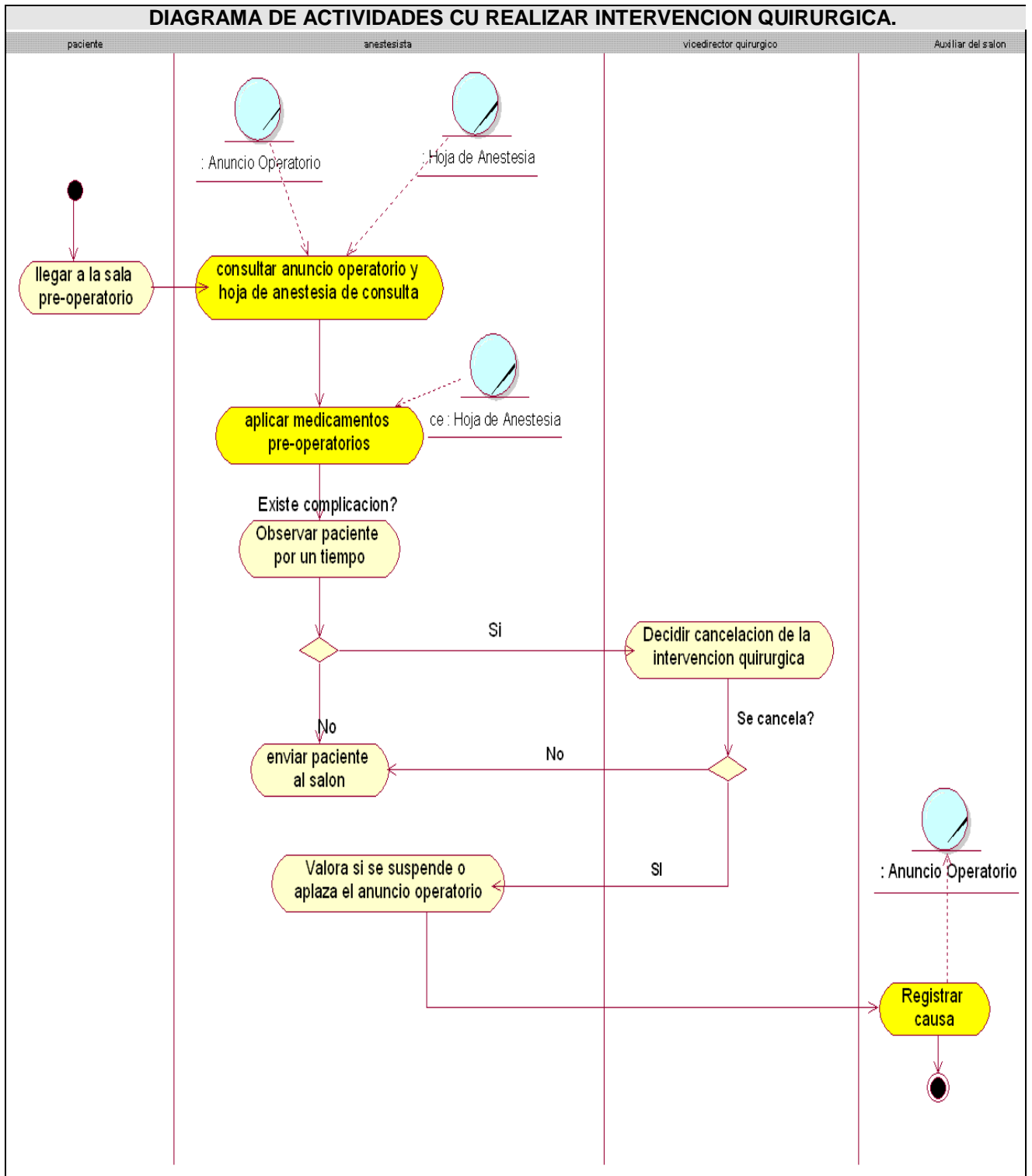


Figura 21: Diagrama de actividades del CUN Realizar Intervención Quirúrgica.

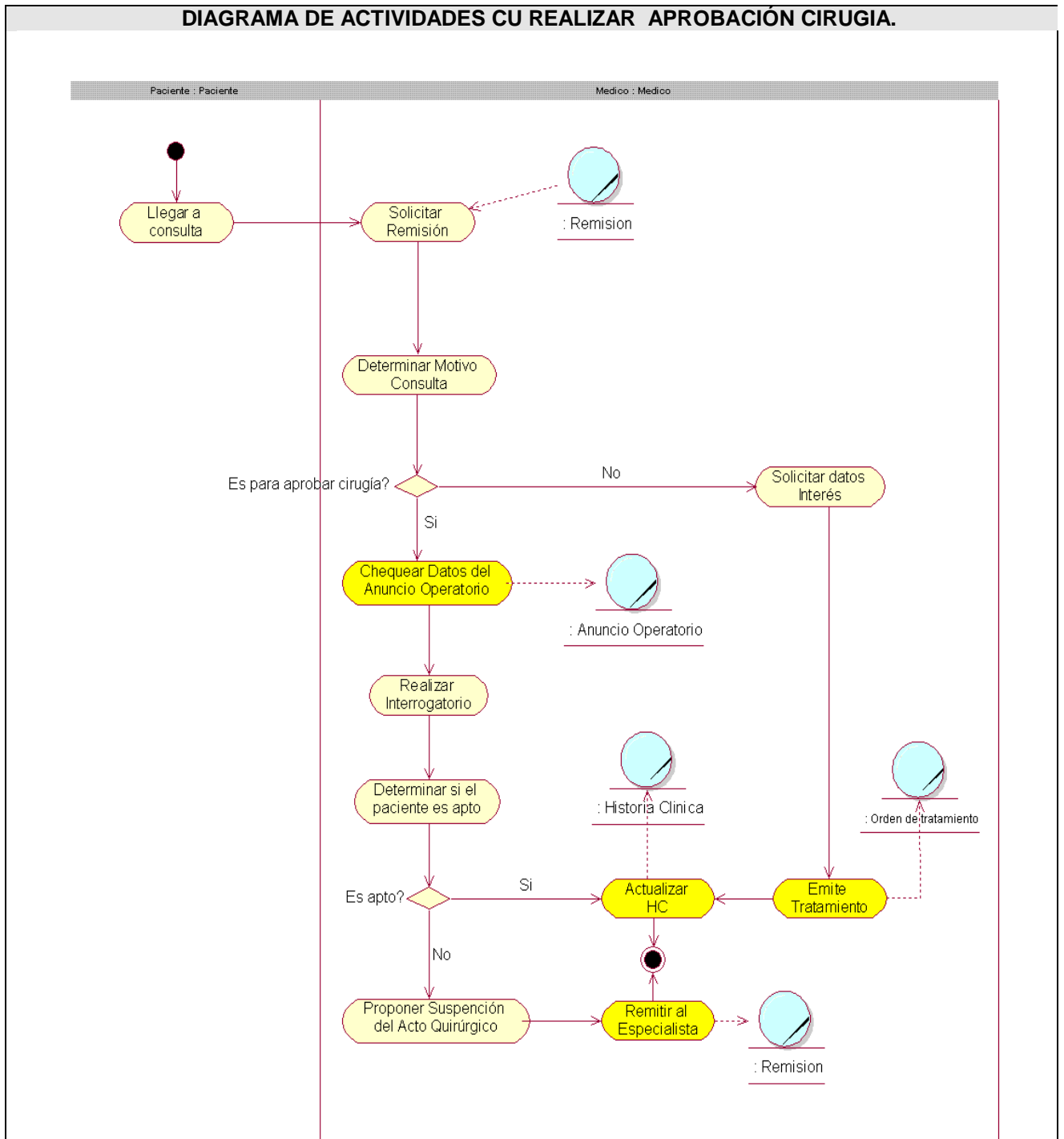


Figura 22: Diagrama de actividades del CUN Realizar Aprobación Cirugía.

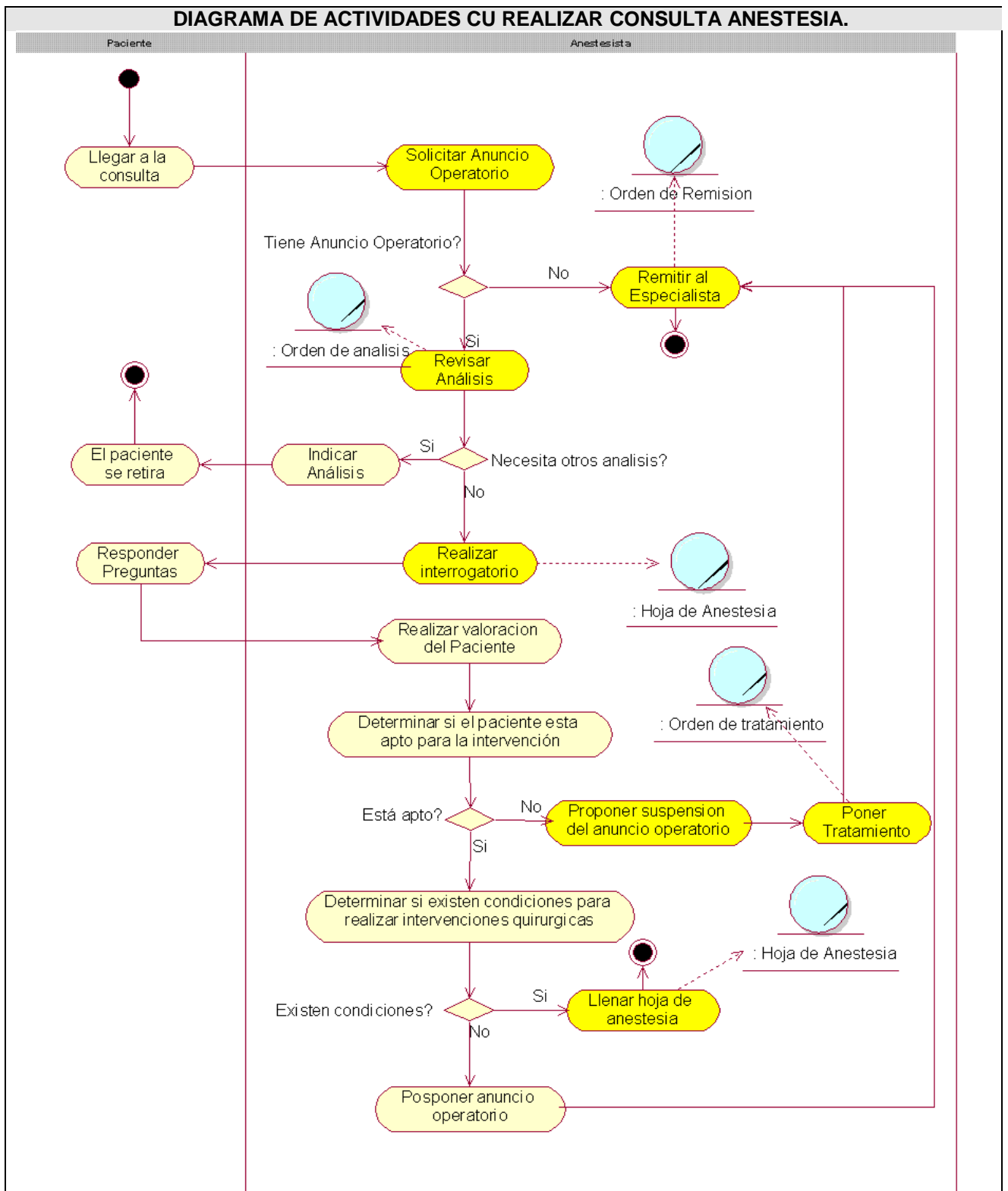


Figura 23: Diagrama de actividades del CUN Realizar Consulta Anestesia.

Anexo 2

Pantallas de la aplicación:

Bloque Quirúrgico Oftalmológico

Historia Clínica Oftalmológica UCT\spalacio Salir
26/06/2008

Consulta

23/06/2008
23/06/2008
09/06/2008
04/06/2008
04/06/2008

Buscar Paciente
 Remisión de caso
 Consultas Oftalmológicas
 Cirugía Refractiva
 Cristalino
 Cornea(Pterigium)
 Oculoplastia
 Neurooftalmología
 Consultas de Aprobación
 Consulta Pediatría
 Consulta Clínico
 Consulta Anestesia
 Lentes Intraoculares
 Planificación quirúrgica
 Administración
 Agregar Usuario
 Agregar Permisos
 Configuración de Sistema

Datos del Paciente

Historia Clínica:
Nombre y Apellidos:
Edad: Sexo:
Fecha:
Hora:

Datos de la consulta

Motivo de Consulta (MC):

Historia de la Enfermedad Actual (HEA):

Reconsulta:

Figura 24: Interfaz de usuario Consulta Especializada Oftalmológica en Oculoplastia.

Bloque Quirúrgico Oftalmológico

Historia Clínica Oftalmológica UCI\spalacio | Salir
26/06/2008

- Buscar Paciente
- Remisión de caso
- Consultas Oftalmológicas
- Cirugía Refractiva
- Cristalino
- Cornea(Pterigium)
- Oculoplastia
- Neurooftalmología
- Consultas de Aprobación
- Consulta Pediatría
- Consulta Clínico
- Consulta Anestesia
- Lentes Intraoculares
- Planificación quirúrgica
- Salon-Servicio-Cantidad
- Salon-Servicio-Tiempo
- Salon-Servicio-Medico
- Administración
- Agregar Usuario
- Agregar Permisos
- Configuración de Sistema

Exámenes para la Consulta de Oculoplastia

Nuevo Examen
Guardar

10 /06 /2008	02 /06 /2008	18 /06 /2008
Refracción	Lensómetro	Refracción Dinámica
Dacriocistitis	Ojo Seco	Ptosis Palpebral
Malposición Palpebral		

Examen de Dacriocistitis

	Fecha del Examen:	Hora del Examen:
	<input type="text"/>	<input type="text"/>
	<input type="text" value="OD"/>	<input type="text" value="OI"/>
Ganmagrafia:	<input type="text" value="..."/>	<input type="text" value="..."/>
Dacriocistografia:	<input type="text" value="..."/>	<input type="text" value="..."/>
Test de Jones I:	<input type="text" value="..."/>	<input type="text" value="..."/>
Test de Jones II:	<input type="text" value="..."/>	<input type="text" value="..."/>
Exploración de vías lagrimales:	<input type="text" value="..."/>	<input type="text" value="..."/>
Test de desaparición de fluoresceína:	<input type="text" value="..."/>	<input type="text" value="..."/>

Guardar Examen

Nuevo Examen
Consulta

Figura 25: Interfaz de usuario Exámenes Específicos en Oculoplastia.

GLOSARIO:

CU: Caso de uso del sistema.

CUN: Caso de uso del negocio.

GoF: Familia de patrones de diseño, que se dividen en patrones Creacionales, Estructurales y de Comportamiento, dichos patrones son esquemas de gran utilidad a utilizar en la construcción de software.

GRASP: Patrones de software para la asignación general de responsabilidades.

Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

MINSAP: Ministerio de Salud Pública.

RUP: Rational Unified Process (Proceso Unificado de Desarrollo).

SAAA: Servicio, Autenticación, Autorización y Auditoría

SOAP: Protocolo Simple de Acceso a Objetos.

SNS: Sistema Nacional de Salud.

UCI: Universidad de la Ciencias Informáticas.

UML: Unified Modeling Language.