
Universidad de las Ciencias Informáticas
Facultad 7



**Título: Desarrollo de los servicios oftalmológicos de
Neuroftalmología, Glaucoma, Uveítis y Córnea**

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

Autores:

Alexey Otero Barrera

Yadia Ruiz González

Tutores:

Ing. Alexander Rodríguez Rabelo

Ing. Joselín Miló Pérez

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

Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.

“La excelencia moral es resultado del hábito. Nos volvemos justos realizando actos de justicia; templados, realizando actos de templanza; valientes, realizando actos de valentía”.

Aristóteles

Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.

DATOS DE CONTACTO

Tutor: Ing. Alexander Rodríguez Rabelo:

Graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Especialista del Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM) y profesor Instructor vinculado a la Facultad # 7. Ha impartido las asignaturas Matemática II y Matemática IV. Ha tutorado hasta el momento cinco trabajos de diploma.

Correo electrónico: arodriguezra@uci.cu

Tutor: Ing. Josélin Miló Pérez

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

Correo electrónico: jmilo@uci.cu

*Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma,
Uveítis y Córnea.*

De Alexey:

Al compañero Fidel por darme la oportunidad de estudiar en una universidad como esta.

A mi abuelita por haberme guiado siempre por el camino correcto y por todo su amor y cariño.

A mi madre, que ha sido mi padre también, por haber estado siempre a mi lado apoyándome incondicionalmente y dándome aliento en todo momento, por todo su sacrificio, abnegación y confianza, por ser la persona más especial del mundo.

A mi compañera de tesis, mis tutores y oponente por haberme ayudado a culminar esta labor.

A todos mis amigos y compañeros, a ellos muchas gracias por todo su apoyo.

A los profesores del proyecto por haber formado parte de mi equipo de trabajo durante todos estos años y por toda la ayuda que muchos me brindaron cuando más la necesité.

Gracias...

De Yadia:

Agradecer a mi familia por ser la fuerza que me impulsa a superar las adversidades que se me presenta día a día, por guiarme, aconsejarme, cuidarme, por estar siempre pendiente de mí, pero sobre todo por apoyarme incondicionalmente.

A mi mamá por nunca perder la fe y creer siempre en mí, por la educación brindada, por ser eternamente la reina de su corazón, gracias a ella he tenido fuerzas para seguir adelante.

A mi papá por estar para mí siempre que lo necesito, por su preocupación y dedicación.

A mi familia de San Cristóbal por acogerme como una hija para ellos, por cuidarme, quererme y sobre todo por estar conmigo en los momentos más difíciles que pase en la Universidad, sin su ayuda no hubiera podido estar graduándome hoy.

A mi abuelita bella, por sufrir, llorar y reír conmigo durante toda mi vida.

A mi madrastra por quererme como su hija y estar pendiente de mí.

A mis cuatros hermanos varones que son mi motor impulsor para superarme cada día, por ser un ejemplo para mí.

A mi primo Rafael que es más que un primo por estar conmigo en todos momentos aun estando lejos, por ser bueno preocupado y ser de todos mis primos el que más me quiere.

A mis amigos de Guantánamo Pupi, Pao, Yunieski, Carlos Manuel, Tania y Evelyn por escucharme y aconsejarme siempre.

A Reinier y Dennys que desde el momento en que entre a la Universidad estuvieron siempre para mí en todo momento.

A los amigos que encontré aquí especialmente a Boris, José Adrian, Nuria, Lisannys, Yaquelin y Bexy por nunca dejarme sola en los momentos difíciles.

A mis tutores y compañero de tesis por su paciencia y ayuda brindada en todos los momentos que la necesite y que no fueron pocas.

A los profes del proyecto en especial a Isnel, Rodney, Annier y Yosmel, Nadieska por estar siempre dispuestos en el momento que los necesite, por nunca recibir un no como respuestas de ellos.

A todos mis compañeros de los grupos por lo que he pasado porque cada uno de ellos ha puesto su granito de arena para ayudarme.

De Alexey:

A mi abuelita María que fue como una madre para mí y a pesar de la vida no permitirle estar aquí hoy a mi lado físicamente, se que comparte este momento junto a mí, pues este también era su sueño.

A mi madre querida que es mi razón de ser, mi guía y ejemplo en mí andar por la vida, por ser esa persona tan especial que es, por darme su apoyo incondicional en todo momento y por todo su esfuerzo, sin ella mi sueño no se hubiera hecho realidad, mami te adoro.

A mi primo Arnaldo por ser como un padre para mí y estar presente en todo momento.

A Yeni por haber sido amiga, hermana y mi apoyo emocional en todos esos magníficos años que estuvo a mi lado, por su paciencia y abnegación, por darme aliento cuando más lo necesité y por todas las cosas buenas que recibí de ella.

A Yailyn por ser esa persona tan maravillosa que se ha mantenido junto a mí durante esta recta final dándome toda su dedicación y cariño, te quiero.

A mi compañera de tesis por haberme ayudado a hacer realidad mi sueño.

Al Dani, Ramses, Raidel, el Migue, Danilo, en fin, a todos mis amigos y compañeros que compartieron junto a mí el día a día durante todo este tiempo, a ellos muchas gracias por todo su apoyo.

A toda mi familia.

De Yadia:

A mi mama y mi papa por que la bendición más grande para ellos es verme ingeniera

A mis hermanos y mi primo Rafael para los que espero ser un ejemplo y que espero que sientan orgullosos de mí

A toda mi familia en especial a mi abuela y mi familia de San Cristóbal que ha estado para mí siempre contribuyendo a mi formación como una excelente persona y una profesional mejor.

Resumen.

En la actualidad los hospitales oftalmológicos no cuentan con un sistema especializado para gestionar la información que se obtiene como parte de los procesos que se llevan a cabo en los servicios de Glaucoma, Córnea, Uveítis, y Neurooftalmología. Los procesos 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 en la información, no se obtenga con rapidez la información que se necesite consultar.

El presente trabajo brinda una propuesta de un sistema informático, que permita la automatización de los procesos que se llevan a cabo en los servicios de Glaucoma, Córnea, Uveítis, y Neurooftalmología dentro de los hospitales oftalmológicos. Para el desarrollo del mismo se utilizó el Proceso Racional Unificado (RUP por sus siglas en inglés), permitiendo obtener los artefactos generados a partir de los flujos de trabajo: "Implementación" y "Prueba". El patrón arquitectónico seleccionado fue Modelo-Vista-Controlador el cual permitió la implementación de un sistema robusto y flexible.

Se pretende que el sistema propuesto logre contribuir a la mejora de la gestión de la información que se genera en las consultas oftalmológicas Glaucoma, Córnea, Uveítis, y Neurooftalmología, logrando así mayor capacidad organizativa, alto grado de confiabilidad y seguridad, así como un aumento de la calidad de los servicios oftalmológicos, de igual manera que proporcione una información actualizada con respecto a los procedimientos de los pacientes.

Palabras claves

Bloque Quirúrgico Oftalmológico, servicios oftalmológicos

ÍNDICE

<i>Introducción</i>	1
<i>Capítulo 1: Fundamentación Teórica</i>	6
1.1 Conceptos básicos relacionados con el dominio del problema.	6
1.2 Sistemas automatizados existentes vinculados al campo de acción.....	7
1.3 Análisis de las posibles tendencias, tecnología y herramientas de desarrollo a utilizar.....	14
<i>Capítulo 2: Características del sistema</i>	24
2.1 Requerimientos no funcionales.....	24
2.2 Descripción de la arquitectura.....	25
2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.....	27
2.4 Seguridad.....	29
2.5 Vista de Despliegue.....	29
2.6 Estrategias de codificación. Estándares y estilos a utilizar.	30
<i>Capítulo 3. Descripción y análisis de la solución propuesta</i>	36
3.1 Valoración crítica del diseño propuesto por el analista.	36
3.2 Modelo de datos y Breve valoración de las Técnicas de validación (Integridad, Normalización de la Base de datos).....	50
3.3 Vista de Implantación.....	56
<i>Capítulo 4. Validación de la solución propuesta</i>	57
4.1 Pruebas de Caja negra.....	57
4.2 Descripción de los Casos de Pruebas.....	58
<i>Conclusiones</i>	67
<i>Recomendaciones</i>	68
Referencias bibliográficas.....	¡Error! Marcador no definido.

Introducción.

Durante varias décadas se ha evidenciado como el desarrollo de la informática se hace cada día más importantes en las diversas esferas socio-económicas de un país, debido a su rapidez para analizar, procesar y comunicar grandes cantidades de información. Una de estas esferas de la sociedad que se ha visto beneficiada es la salud.

La informática médica surge con la aplicación de la informática y las comunicaciones en las diferentes disciplinas de la medicina mediante el uso del software médico. (1) Esta nueva ciencia resulta imprescindible para la adquisición no sólo de conocimientos, sino de herramientas que le posibilitan al profesional de salud acceder a la información de forma rápida y eficiente.

En Cuba, la informatización del Sistema Nacional de Salud Pública (SNS) está definida por el conjunto de métodos, técnicas, procedimientos y actividades gerenciales encaminadas al manejo de la información en el sector de salud. La cual comprende información sobre el estado de salud de la población, información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones, clínico-epidemiológicas, operativas y estratégicas.

El eje fundamental del sector lo constituye el paciente quien será el principal beneficiado, ya que, al garantizarse aplicaciones con calidad y consistencia de la información, traería consigo un incremento de 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.

En el 2003, el Ministerio de Salud Pública (MINSAP) traza una estrategia de colaboración con el propósito de alcanzar un alto grado de informatización en el sector de la salud pública, (2) para lo cual convocó a un grupo de instituciones propias del sector, el Ministerio de la Informática y las Comunicaciones (MIC) y otros organismos de la administración central del estado.

Una de las instituciones que participa en dicha estrategia es la Universidad de la Ciencias Informáticas (UCI), específicamente el Centro Especializado en Soluciones de Informática Médica (CESIM), desarrolla el sistema Bloque Quirúrgico Oftalmológico (BQO).

El BQO se encuentra actualmente instalado el hospital Ramón Pando Ferrer, permite gestionar y administrar parte de la información generada durante los procesos por los que atraviesa un paciente al ser atendido en las especialidades como Cirugía Refractiva y Cristalino, pero no responde a todos los servicios que se prestan en dicha institución, siendo el caso de Neuroftalmología, Glaucoma, Córnea y Uveítis.

Otro sistema con el que se trabaja en dicha institución es Galen Hospital desarrollado por la empresa de Softel para el Ministerio de Salud Pública. Está formado por diferentes módulos como Inscripción /Admisión, Archivo, Turnos de Consulta ambulatoria entre otros, pero no gestiona la mayoría de los procesos clínico-quirúrgicos oftalmológicos como la nota pre-operatoria o Anuncio Operatorio.

Dicho centro también cuenta con el Sistema Automatizado de Microcirugía (SAMC), implementado en "Clipper'85 Winter" y desarrollo bajo MS-DOS, encargado de la gestión de la información referente al área de Microcirugía del Cristalino.

Sin embargo los sistemas antes mencionados no gestionan la información que se obtiene como parte del proceso de atención al paciente en servicios tales como Neuroftalmología, Glaucoma, Córnea y Uveítis.

Como principal deficiencia se detecta que la información que se recoge como parte del proceso de atención al paciente, en los servicios mencionados, se realiza de forma manual. Además, una vez terminada la consulta el paciente se lleva la historia clínica, por lo que puede extraviarla y traería como consecuencia que una vez que el paciente regrese a la consulta tenga que realizarse nuevamente los exámenes, impidiendo la existencia de un historial del mismo.

La historia clínica ambulatoria trae como consecuencia que no sea posible gestionar la información que se encuentran en la misma en un momento determinado. Lo que afecta la calidad de atención al paciente, debido a la carencia de datos específicos que

pueden ser necesarios a la hora de emitir diagnósticos, implicando una posible toma de decisiones equivocada.

Por lo anterior, se evidencia que no se realiza la gestión de la información que se genera como parte de los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.

Apoyándose en lo antes expuesto, se definió como **problema a resolver**: ¿Cómo facilitar la gestión de información de los servicios de Neuroftalmología, Glaucoma, Córnea y Uveítis de la especialidad de oftalmología?

Tomándose como **objeto de estudio**: El proceso de gestión de la información en las instituciones hospitalarias.

El **campo de acción** se enmarca en el proceso de gestión de la información de los servicios Neuroftalmología, Glaucoma, Uveítis y Córnea del sistema Bloque Quirúrgico Oftalmológico de las instituciones hospitalarias.

Para solucionar el problema planteado se establece como **objetivo general**: Desarrollar un sistema informático que gestione la información referente a los servicios Neuroftalmología, Glaucoma, Uveítis y Córnea del Bloque Quirúrgico Oftalmológico para facilitar la gestión de la información en las áreas de las instituciones hospitalarias.

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

- ❖ Evaluar las tendencias actuales a nivel mundial referente a los sistemas de información hospitalaria.
- ❖ Asimilar la arquitectura definida por el proyecto Bloque Quirúrgico Oftalmológicos para el desarrollo de sus aplicaciones.
- ❖ Implementar todas las funcionalidades identificadas para los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.
- ❖ Lograr la certificación por el grupo de calidad de las funcionalidades implementadas y documentación obtenida de los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.

Con el desarrollo del sistema se beneficiarán:

1. Pacientes de las instituciones oftalmológicas.
 2. Personal médico que labora en los hospitales.
 3. Institución hospitalaria
 4. MINSAP.
1. Desde el punto de vista del paciente.
 - a) Disponer de un sistema que tenga como centro de referencia al paciente, dando respuesta a sus necesidades de asistencia médica especializada en oftalmología, facilitando su acceso a los recursos de salud.
 - b) Atender al paciente como un cliente único de la institución hospitalaria.
 - c) Historia Clínica Electrónica única por paciente y su seguimiento.
 - d) Calidad en los informes médicos que se le entregan a los pacientes.
 - e) Seguridad de toda la información médica del paciente.
 2. Desde el punto de vista del profesional médico.
 - a) Poner a disposición del profesional médico herramientas de gestión clínica y gestión administrativa que den respuesta a sus necesidades reales.
 - b) Disponer y acceder a información única e integrada del paciente para facilitar los procesos de diagnóstico y tratamiento oftalmológico.
 - c) Dar acceso a herramientas que faciliten acceder y compartir conocimiento científico.
 - d) Revisión de diagnóstico de pacientes atendidos con anterioridad.
 3. Desde el punto de vista de la institución hospitalaria.
 - a) Mejor organización de los diferentes procesos que se llevan a cabo en el hospital.
 - b) Servicios informáticos orientados a la parte administrativa del hospital y al control de los recursos.
 4. Desde el punto de vista del MINSAP.
 - a) Control de gestión y utilización óptima de recursos de salud.
 - b) Obtención de datos estadísticos en tiempo real, posibilitando realizar estudios científicos de diagnósticos más comunes, etc.

- c) El presente trabajo de tesis consta de 4 capítulos y a continuación una breve reseña de los mismos.
- d) **Capítulo 1:** Fundamentación teórica: Ubica al lector en el ambiente de desarrollo de los servicios Neuroftalmología, Glaucoma, Uveítis y Córnea del sistema Bloque Quirúrgico Oftalmológico, justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo del mismo.
- e) **Capítulo 2:** Descripción del sistema: Se plantea una valoración crítica del diseño propuesto por el analista, un análisis de posibles implementaciones de componentes o módulos ya existentes que puedan ser rehusados y las estrategias de integración.
- f) **Capítulo 3:** Descripción y análisis de la solución propuesta: Se realiza una valoración crítica por los analistas donde se presentan los diagramas de clases del diseño y los diagramas de interacción, se describen las clases u operaciones nuevas que son necesarias a la hora de realizar el sistema. Además, muestra el modelo de datos con una breve valoración de las técnicas de validación y la vista de Implementación donde se propone el diagrama de componentes.
- g) **Capítulo 4:** Validación de la solución propuesta: Se realiza una valoración de la fase de prueba de un software donde se describe sus características, objetivo y los métodos que posee.

Capítulo 1: Fundamentación Teórica.

En el presente capítulo se exponen y se describen los conceptos básicos relacionados con el dominio del problema y algunos de los sistemas automatizados tanto nacionales como internacionales que se encuentran vinculados al campo de acción. Además se realiza un estudio de las tecnologías y herramientas que pueden ser empleadas en el desarrollo de la solución propuesta.

1.1 Conceptos básicos relacionados con el dominio del problema.

Para la mejor comprensión del trabajo es necesario conocer los conceptos básicos relacionados con el dominio del problema que se necesitan para comprender mejor el desarrollo del trabajo.

Glaucoma: Enfermedad ocular cuyo cuadro clínico completo se caracteriza por incremento de la presión intraocular, excavación y degeneración del disco óptico y típico daño de las fibras retinales provocando característicos defectos en el campo visual. (3)

Uveítis: Se define como la inflamación de la úvea, lámina intermedia del ojo que se encuentra entre la esclerótica y la retina, la cual aporta la mayor parte del suministro sanguíneo a la retina. (4)

Córnea: Tejido delgado, duro y transparente, situado en la parte anterior del globo del ojo, por delante del iris y la pupila, que sirve para enfocar las imágenes: delante de la córnea está únicamente la conjuntiva. (5)

Oftalmología: Es la especialidad médico-quirúrgica que se relaciona con el diagnóstico y tratamiento de los defectos y de las enfermedades del aparato de la visión. (6)

Bloque Quirúrgico Oftalmológico: Es el área centralizada en la que se genera toda la actividad quirúrgica del hospital oftalmológico.

Cirugía: Es la rama de la medicina que se dedica a curar las enfermedades por medio de operaciones. (7)

Neuroftalmología: Es una rama de la oftalmología que se dedica al estudio anatomofisiológico y patológico de la estrecha relación que existe entre el ojo y el sistema nervioso central y en particular el nervio óptico y la vía visual. (8)

1.2 Sistemas automatizados existentes vinculados al campo de acción.

Uno de los aspectos más importantes a tener en cuenta antes de realizar el desarrollo de cualquier aplicación es el estudio de los antecedentes relacionados con el campo de acción que concierne a la misma, para obtener de aquí, una línea base en cuanto a la estructura y las principales funcionalidades.

En la actualidad existen tanto nacional como internacionalmente múltiples sistemas que gestionan la información oftalmológica, entre los que se destacan:

Internacionales.

Opera.

Es un sistema de gestión de la cirugía, presenta una solución informática para la gestión del quirófano, la administración de salas de operaciones, el programa de cirugía, la gestión de admisiones, la administración de material y otras muchas prestaciones para el bloque quirúrgico.

Opera brinda a los profesionales de la salud y administradores todas las informaciones clínicas y financieras de cada intervención quirúrgica a programar, real o ya realizada. Además soporta la toma de decisiones y la administración general del bloque operatorio, gracias a la producción de una información real y precisa.

La solución informática Opera incluye los módulos para la planeación de las intervenciones, los exámenes de pre admisión, la administración de los recursos, las notas clínicas de la intervención, el resumen de todo el procedimiento, la gestión del material y la administración de la esterilización. Es una fórmula perfecta en la administración del bloque operatorio, ya que permite una información eficaz a un costo mínimo.

Presenta también un sistema de bases de datos implementado sobre SQLServer 2000. Es propiedad de la empresa CHCA computer system. (9)

Aunque presenta beneficios en la documentación del tiempo de los procedimientos por cada intervención (Pre operatorio, Operatorio, sala de recuperación, Post operatorio) y que puede ser muy útil a la hora de definir el sistema, no es capaz de gestionar la información clínica ni especificar los servicios oftalmológicos que se necesitan.

Net Clínicas

Es un potente software de gestión de clínicas, mediante el cual podrá controlar las distintas tareas derivadas de la gestión de la clínica. El software está desarrollado con la última tecnología, lo que hace de Net Clínicas un programa de gestión de clínicas muy potente. El mismo está compuesto por tres módulos Registro de pacientes, Agendas e Historial clínico.

Net Clínicas se basa en plataforma cliente / servidor, lo que permite poder trabajar en red de forma simultánea desde varios ordenadores. (10)

Net clínicas ofrece entre otros beneficios una completa agenda de citas y control evolutivo de los pacientes. Además es un programa totalmente configurable, adaptable a todo tipo de clínicas, esta última característica puede ser estudiada y de gran utilidad a la hora de definir nuestro sistema.

Aunque el programa es totalmente configurable por el usuario, lo que permite que pueda ser utilizado por clínicas de diferentes especialidades, el mismo no está especializado en los servicios oftalmológicos que se necesitan y además solo puede ser ejecutado en máquinas que utilicen sistema operativo Windows, es propietario y es un software de escritorio.

VisionDat.

Es un software de tecnología avanzada que ofrece diversas opciones en el manejo de consultorios oftalmológicos, el mismo no llega a la totalidad de las funcionalidades necesarias para el presente problema pues recoge solamente Datos generales,

Estudios oftalmológicos, la historia clínica y estudios refractivos. Está destinado a los consultorios especializados en oftalmología y optometría.

Está conformado por:

- ❖ Datos de pacientes.
- ❖ Datos de consulta.
- ❖ Citas y actividades.
- ❖ Expedientes.
- ❖ Gráficas, estadísticas.

Su versión 2.4 está disponible desde el 2009 y posee las mismas características y funcionalidades. (11)

Este software se encuentra bajo licencia propietaria, no posee los servicios que necesitamos y solo podrá ser implantado en máquinas que utilicen sistema operativo Windows. Todas las PCs donde se instale este software deben tener obligatoriamente de memoria RAM como mínimo 1GB y en el mismo su rendimiento no es el más óptimo.

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, la misma presenta un sistema de datos cliente / servidor, con un entorno muy amigable Windows, compatible con versiones desde Windows Vista hasta Windows 98. Dicho sistema presenta como Base de Datos estándar y robusta Microsoft SQL Server, la información es totalmente exportable y legible por otras bases de datos. (12) Dentro de las funcionalidades que brinda este software encontramos:

- ❖ 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.

Aunque este sistema brinda todas estas funcionalidades y que algunas son de importante a la hora de definir nuestro sistema es un software propietario que presenta un costo aproximado de 150 euros por puesto de trabajo que es su principal desventaja, debido a que implantarlo en el hospital Ramón Pando Ferrer costaría una fortuna ya que dicha institución cuenta con más 400 puesto de trabajo.

OFTALMOSALUS.

Es el Software especializado para la Gestión de Clínicas y Consultas de Oftalmología. El mismo fue desarrollado por QSOFT desde 1995, Diseñado por oftalmólogos y por expertos en gestión de Clínicas Oftalmológicas, permite gestionar de manera completa e integrada las tres 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: (13)

- ❖ Acceso sencillo y rápido al historial de los pacientes.
- ❖ Historia médica especializada para Oftalmología y configurable a medida.

*Desarrollo de los servicios oftalmológicos Neurooftalmología, Glaucoma,
Uveítis y Córnea.*

- ❖ 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.

Aunque este sistema es bastante completo el mismo es propietario y no es multiplataforma por lo que no se podrá implantar en máquinas con sistema operativo de Linux ni Windows vista.

ActualOftalmo.

Software desarrollado por la empresa argentina ActualSoft; fue desarrollado precisamente para un módulo oftalmológico de un bloque quirúrgico, presenta un protocolo gráfico e interactivo de la consulta oftalmológica y una extensa base de datos de signos y síntomas oftalmológicos, desarrollada sobre Microsoft SQLServer que permite además el registro digital de antecedentes familiares y personales así como el de estudios especializados y tratamientos oftalmológicos. Este software tiene un precio de 240 dólares por cada puesto de trabajo y de 150 dólares por la actualización anual de cada uno de estos puestos.

Cuenta con todas las herramientas necesarias para poder controlar de cada paciente, hasta el mínimo detalle. Historia clínica interactiva, diseñada para el oftalmólogo, aportando características esenciales para la asistencia en el consultorio. Manejo de diagramas y esquemas útiles para el registro y verificación de patologías oftalmológicas. Registro de fichas completas por paciente, Manejo de Turnos y Estadísticas.

Se trata de un software propietario destinado a los consultorios 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 (14)

A pesar de ser una solución bastante cercana a lo que se necesita, debido a que permite el registro y control de todos los estudios necesarios para obtener el

*Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma,
Uveítis y Córnea.*

diagnóstico preciso, el precio que posee este sistema es muy elevado, ya que en hospitales como el Instituto Cubano de Oftalmología “Ramón Pando Ferrer” que existen alrededor de 600 puestos de trabajo se necesitaría pagar aproximadamente 144000 dólares por implantar la solución y otros 90 000 anuales por las actualizaciones del mismo.

Nacionales

Galen Hospital

Sistema desarrollado por la empresa Softel para el MINSAP es orientado hacia la informatización de la gestión de pacientes como elemento básico en la atención médica hospitalaria, a optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos. Está compuesto por los módulos: Inscripción /Admisión. Archivo, Estadísticas, Información, Turnos de Consulta ambulatoria, Dietas y Certificados y Gestión de Salas.

Este solo presenta una pequeña solución al presente problema que posibilita crear informes operatorios de forma general, es decir, la aplicación no permite el desarrollo de la mayoría de los procesos clínico-quirúrgicos oftalmológicos. (15)

MEDISyS.

Es un proyecto de Información Hospitalaria, que permite registrar, controlar y procesar la información necesaria del paciente para la realización de los servicios de salud que requiera, así como la información de la planificación, ejecución y supervisión de los servicios de salud prestados por la institución para facilitar la toma de decisiones con vistas a mejorar la calidad y eficiencia de los mismos.

Este funciona a través de una Red Local en plataforma Cliente-servidor, que enlaza todas las computadoras existentes en la institución en un ambiente que permite comunicarse entre ellas y compartir los ficheros de información desde cualquier punto, teniendo en cuenta las funciones que debe cumplir cada uno de los usuarios, así como los niveles de acceso que le sean permitidos a la misma.

La plataforma tecnológica utilizada para el desarrollo de MEDISyS es:

*Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma,
Uveítis y Córnea.*

- ❖ Lenguaje de Programación Orientado a Objeto (Borland Delphi 3,0 Client-Server).
- ❖ Sistema de Gestión de Base de Datos para plataforma cliente-servidor (SQL Server 6,5).
- ❖ Lenguaje para la elaboración de Página WEB (HTML).
- ❖ Sistema Operativo Windows NT 4,0 en el Server y Windows '95 en los clientes.

Este software se encuentra bajo licencia propietaria, además esta implementado para que se ejecuten con interfaz de escritorio, no es un multiplataforma porque solo corre en Windows y no tiene los servicios que se necesita. (16)

Sistema Automatizado de Microcirugía (SAMC).

El Sistema Automatizado de Microcirugía se encuentra en estos momentos instalado en el Hospital Oftalmológico "Ramón Pando Ferrer" de Ciudad de La Habana, en el área de Microcirugía del Cristalino, dicho software se encarga en parte de este tipo de gestión, desarrollado por el equipo de ingenieros de dicha institución está implementado en "Clipper'85 Winter", lenguaje de programación creado en 1985 por Nantucket Corporation y herramienta líder de desarrollo bajo MS-DOS de aplicaciones relacionadas con bases de datos, lo que dificulta su mantenimiento y actualización.

Sin embargo este solo presenta una solución particular para dicho hospital, es poco configurable y escalable para otros hospitales oftalmológicos, además de presentar problemas con la gestión de la información a nivel de base de datos, específicamente a la hora de realizar reportes y consultas, ya que es necesario elaborar un índice que contenga toda la información ordenada de la base de datos en cada ocasión que se quiere obtener dichos reportes. (15)

Los sistemas descritos anteriormente, ofrecen soluciones deseables por cualquier institución hospitalaria pero en su mayoría poseen dos características fundamentales que los hacen poco aceptados, una de las característica que poseen la mayoría de estos sistemas es que no son portables o multiplataforma por lo que atan al usuario a la plataforma sobre la cual se ejecutan.

Por otro lado, puede verse, que no brindan una solución factible que facilite el proceso de gestión de la información correspondiente a la especialidad de oftalmología, ya que no se ajustan totalmente a las características del mismo. En el caso de los que brindan una solución parcial, presentan precios elevados que dificultan su utilización y el despliegue de los mismos en los hospitales del país tendría un costo muy elevado.

Se puede concluir que los sistemas analizados anteriormente no pueden ser acogidos como posibles soluciones para el presente problema. Pero sí brindan una posible línea a seguir para dar solución al mismo, en cuanto a los principales requisitos y funcionalidades que debe cumplir el sistema, así como sus principales características.

1.3 Análisis de las posibles tendencias, tecnología y herramientas de desarrollo a utilizar.

Para desarrollar un software es necesario haber realizado y analizado las antecedentes nacionales e internacionales así como las tecnologías, tendencias y herramientas existentes, para tener así en cuenta ciertos aspectos como metodologías de desarrollo a usar, lenguajes de programación, gestores de bases de datos, patrones arquitectónicos y de diseño entre otros.

Metodología

RUP

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente Quien, Cómo, Cuándo y Qué debe hacerse en el proyecto.

Este proceso posee tres características, la primera es que RUP está centrado en la arquitectura, ya que relaciona la toma de decisiones de cómo y en qué orden debe ser construido el sistema. La segunda características que tiene RUP es que está dirigido por los Casos de Usos pues se expresa lo que el usuario desea orientando así al proyecto la importancia que tiene. Su última característica, es iterativo e incremental, es decir divide en mini proyectos al proyecto, donde la arquitectura y los casos de usos cumplan sus objetivos.

RUP divide el proceso en 4 fases, la de Inicio, Elaboración, Construcción y Transición donde se realizan varias iteraciones en cada una. En la fase de inicio se define el alcance del proyecto y se identifican los riesgos y los principales casos de uso, es de principal importancia en esta fase que los desarrolladores, clientes y usuarios finales logren ponerse de acuerdo en lo que quieren además de entender el problema, la estructura y la dinámica de la organización. Ya en la fase de elaboración se eliminan los riesgos, se realiza un plan del proyecto y se completan los casos de uso. En la fase de construcción se crea el manual de usuario y se concentra en la elaboración de un producto eficiente y operativo. (17)

CMMI

CMMI es un conjunto de modelos que permiten obtener un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización, y describen las tareas que se tienen que llevar a cabo para mejorar esos procesos. Los módulos CMMI son extractos de los modelos CMMI a los que se han añadido posibles pruebas a realizar, y sirven de base para emprender la mejora de procesos.

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

- ❖ Nivel 1: Inicial.
- ❖ Nivel 2: Administrado.
- ❖ Nivel 3: Definido.
- ❖ Nivel 4: Cuantitativamente administrado.
- ❖ Nivel 5: Optimizado. (18)

Servidor de aplicaciones

Una aplicación web es una aplicación informática o también conocida como una aplicación software que se codifica en un lenguaje soportado por los navegadores web. (19)

Un navegador web 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.

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

Internet Information Server

Internet Information Server (IIS) Es el principal servidor de aplicaciones web de Microsoft. Sus principales funcionalidades son la publicación de sitios y aplicaciones web, sitios FTP, SMTP y Servicios de noticias. Dispone de soporte necesario para crear páginas en ASP. Su principal problema radicaba en la pobre seguridad de sus primeras versiones, aspecto que fue resuelto en la versión 6.0, en la que Microsoft ha cambiado el comportamiento de los controles ISAPI pre - instalados. Es un sistema esencial destinado a la utilización de los servicios de Internet basado en la plataforma Windows. (20)

Servidores WEB Apache.

En lo que respecta a las tecnologías por parte del servidor sobresale Apache. El servidor HTTP Apache es un producto libre, de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras. Implementa el protocolo HTTP/1.1. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.(20)

Patrón Model-View-Controller (MVC)

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes, el modelo que se encarga de administrar el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado y responde a instrucciones de cambiar el estado (habitualmente desde el controlador); la vista que maneja la visualización de la información y el controlador que

interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. (21)

Arquitectura cliente-servidor.

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.

Servicios web

Los servicios web son colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

XML

Es un lenguaje de meta marcaje que proporciona un formato para describir datos estructurados. Facilita declaraciones más precisas de contenido y resultados de búsquedas, con más significado entre muchas plataformas. Además, el XML habilitará una nueva generación de aplicaciones manipulación y visualización de datos basadas en la WEB.

La idea que subyace bajo la creación del XML, es la de crear un lenguaje muy general, que pueda tener muchos usos. Mientras que HTML está diseñado para presentar información directamente a los usuarios, aunque es un lenguaje complejo de procesar para los programas informáticos.

El HTML no indica lo que está representando, se preocupa principalmente de que el contenido tenga determinadas características visuales, pero no especifica lo que está mostrando, si es el título de un libro o el precio de un artículo. Contrario a esto, XML describe el contenido de lo que etiqueta. (20)

Lenguaje de programación

Lenguajes del lado del servidor

Los lenguajes del lado del servidor son aquellos, que se ejecutan en el servidor web, antes de que se envíe la página al navegador. Los códigos puestos en el servidor se encargan de construir las páginas que serán enviadas de respuesta al cliente, estos lenguajes pueden acceder a bases de datos, ficheros en el servidor y recursos en la red. No son visibles por los clientes, ya que cuando se solicitan a través del servidor web lo que se genera es código HTML que es entendible y representable por un navegador web. (20)

C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java). C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. Ya existe la versión 3.0 de C# en fase de beta destacando los tipos implícitos y el LINQ (Language Integrated Query).

Las principales características que hacen del lenguaje C# una elección inteligente es su sencillez e independencia de tipos ya que El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad de código. (20)

Lenguajes del lado del cliente

Se ejecutan en el cliente o navegador, son los encargados de darle dinamismo a la página sin necesidad de realizar viajes al servidor. Estos lenguajes son interpretados,

*Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma,
Uveítis y Córnea.*

pueden acceder a la información HTML que se muestra en un navegador pudiendo modificarla y actualizarla según las necesidades de los programadores.

Javascript

Javascript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, Javascript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código Javascript integrado dentro de las páginas web. Estadísticas brindadas por la w3schools ratifican que en el presente año 2007, el 94% de los sitios muestreados utilizan este lenguaje del lado del cliente. (20)

Framework o Componentes

Mono

Mono es una plataforma de desarrollo de código abierto basada en el .NET Framework. Es una plataforma para ejecutar y desarrollar aplicaciones modernas basadas en los estándares ECMA/ISO. Mono puede ejecutar aplicaciones hechas para los frameworks .NET y Java. Permite a los desarrolladores construir aplicaciones GNU-Linux y multiplataforma de alta productividad. La implementación cumple los estándares ECMA para C# y la infraestructura de lenguaje común (Common Language Infrastructure, CLI). Acoge varias licencias libres (X11, LGPL y GPL) y tiene una comunidad de desarrolladores activa, está patrocinado por Novell y es la base para muchas aplicaciones.

Mono incluye compiladores, un intérprete compatible con el CLR de ECMA y un conjunto de librerías. Las librerías cubren la compatibilidad con Microsoft .NET (incluyendo ADO.NET, System.Windows.Forms y ASP.NET). Mono posee librerías adicionales y otras de terceras partes. Para el desarrollo de aplicaciones gráficas se incluye la librería GTK# que es un "binding" sobre GTK+ y GNOME para permitir el

desarrollo de aplicaciones nativas para Gnome utilizando Mono y cualquiera de los lenguajes soportados. (20)

Microsoft .NET Framework versión 2.0

Microsoft .NET Framework versión 2.0 instala los archivos asociados de .NET Framework necesarios para ejecutar aplicaciones desarrolladas sobre dicho framework. 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.(20)

La biblioteca de clases de .NET Framework

La biblioteca de clases de .NET Framework 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 Framework sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de este. Además, los componentes de terceros se pueden integrar sin dificultades con las clases. Esta librería está escrita en Lenguaje Intermedio de Microsoft (MSIL-Microsoft Intermediate Language) que es un conjunto de instrucciones independiente de la Unidad central de procesamiento que se pueden convertir de forma eficaz en código nativo. Por tanto esta librería puede ser utilizada desde cualquier lenguaje cuyo compilador genere MSIL. (20)

Compilación (Compilador, Máquina Virtual, Interprete)

El Common Language Runtime (CLR)

El CLR administra memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Podríamos decir que esta 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.

El CLR aporta características de vital importancia, algunas de estas son: seguridad avanzada: En el CLR los componentes administrados reciben niveles de confianza

diferentes en función de una serie de factores entre los que se incluye su localización, origen (fabricante) y permisos o roles del usuario que lo esté ejecutando; seguridad de tipos: el CLR impone la robustez del código mediante la implementación de un sistema estricto de comprobación de tipos y código denominada Sistema de Tipos Común (CTS). (20)

Entornos de desarrollo integrados (IDE)

Visual Studio.net

Visual Studio.net 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 Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y servicios web XML.

Visual Studio 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 fácil y rápida de crear páginas de formularios Web Forms.(20)

Sistemas Gestores de Bases de Datos

PostgreSQL

Hoy día, existen muchas empresas y sitios web que necesitan mantener de forma eficiente un gran volumen de datos. Muchos de ellos optan por soluciones comerciales, aunque muchas otras confían en el software libre optando por una solución como PostgreSQL. Dentro de los gestores de bases de datos sobresale PostgreSQL, un motor de base de datos que es servidor de base de datos relacional libre, liberado bajo la licencia BSD. PostgreSQL no está controlado por una sola compañía, sino que cuenta con comunidad global de desarrolladores y compañías para su evolución.

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. Esta técnica elimina la necesidad hacer bloqueos de lectura y asegura los principios ACID (atomicidad, consistencia, aislamiento y durabilidad) de la base de datos de una manera eficiente. (20)

Herramientas

Javascript Asíncrono (AJAX).

Javascript Asíncrono (AJAX). AJAX no es una tecnología, sino la unión de varias, incorporando: presentación basada en estándares usando XHTML y CSS; exhibición e interacción dinámicas usando el Document Object Model; intercambio y manipulación de datos usando XML and XSLT; recuperación de datos asincrónica usando XMLHttpRequest; y Javascript para unir todo el contenido. Es el acrónimo para Asynchronous Javascript + XML y se resume en: cargar y renderizar una página, luego mantenerse en esta, mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma. (20)

Herramientas de modelado

Visual Paradigm

“Visual Paradigm para UML (Lenguaje Unificado de Modelado): es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (20)

Control de versiones

Subversion

Subversion es un sistema de control de versiones libre y de código fuente abierto. Maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros. Para usted, esos ficheros pueden ser código fuente— para otros, cualquier cosa desde la lista de la compra de comestibles hasta combinaciones de vídeo digital y más allá. Es compatible además con sistemas operativos libres, propietarios y presenta además características que lo hacen muy atractivo: Versionado de directorios y metadatos, verdadero historial de versiones, envíos atómicos, manipulación consistente de datos y hackability; esta última hace que Subversion sea extremadamente fácil de mantener y reutilizable por otras aplicaciones y lenguajes.

(20)

En este capítulo se efectuó un estudio sobre los principales sistemas existentes en el mundo que gestionan la información referente a los procesos oftalmológicos en las instituciones hospitalarias y se determinó que ninguno no cumplen con los requerimientos que se necesitan para dar solución a la problemática existente. Además, se explicó la aplicación de los patrones de diseño y la utilización de las tecnologías y herramientas propuestas para la elaboración del sistema.

Capítulo 2: Características del sistema.

En el presente capítulo se presenta los requerimientos no funcionales que presenta el sistema. Se describe la arquitectura además se realiza un análisis de las posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados así como las estrategias de integración. También se aborda aspectos importantes referentes con la Seguridad, la Vista de Despliegues, la Estrategias de codificación y los estándares y estilos utilizados.

2.1 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos 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).

2.2 Descripción de la arquitectura

Uno de los aspectos importantes en el desarrollo de cualquier software es su arquitectura, también conocida como arquitectura lógica, es el diseño de más alto nivel de la estructura de un sistema. La misma consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción de un software.

Una arquitectura flexible y robusta trae consigo el desarrollo de un software con excelente calidad. Para lograr estos aspectos se utiliza el Modelo Vista Controlador (MVC) (descrito en el Capítulo 1), la misma se separa en tres capas con el objetivo que si llegado el caso se tiene que hacer algún cambio no es necesario buscar entre todos los códigos.

En la vista se utiliza como lenguajes ASP y Javascript HTML en la creación de las páginas y la librería AJAX control toolking. En la controladora C# como lenguaje de programación, el Visual Studio 2005 como IDE de desarrollo y .NET como framework desarrollo y en el modelo PostgreSQL 8.2.

En el diseño que se propone se basa en los patrones GRAFS y GOF con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Estos patrones conciben soluciones factibles a los problemas del diseño orientado a objeto lo que trae consigo una implementación más clara.

Dentro de los patrones GRAFS se identifica cinco patrones muy utilizados: Experto, Creador, Alta Cohesión, Bajo Acoplamiento y el Controlador.

Se propone el uso de los patrones Factory method y Abstract Factory de GOF. El primero asigna un conjunto altamente cohesivo de responsabilidades a una clase artificial que no representa nada en el dominio del problema, una clase creada para dar soporte a una alta cohesión, un bajo acoplamiento y reutilización. El segundo facilita una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

La Capa Intermedia (Middleware) forma parte de la solución del problema, a través de la misma se accede al repositorio. Esta subcapa tiene a su cargo la ejecución la lógica de acceso a datos y contiene dos paquetes de clases los repositorios y Fábricas de Objetos.

Los repositorios son los encargados de manejar las colecciones de Objetos Comunes (Entidades de la BD representados como objetos) y realizar operaciones sobre ellas.

Las fábricas de objetos son Paquetes de clases que contiene la funcionalidad necesaria para acceder a la base de datos y realizar las operaciones que se explican a continuación. Por cada entidad mapeada desde la base de datos, existen cuatro clases que heredan de las interfaces:

IInsertFactory: encargada del proceso de inserción.

IDeleteFactory: Encargada de suprimir una entidad determinada.

IUpdateFactory: encargada de actualizar atributos de los objetos en la BD.

Es válido mencionar que existen por cada entidad otra clase que hereda de la clase interfaz IDomainObjectFactory, encargada de realizar el proceso de mapeo de las entidades (convertir tupla a tupla el resultado de un proceso de selección en la entidad a la que corresponde).

Para enlazar la capa de Middleware con el Acceso a Datos que está en el servidor de BD se requiere el uso la Npgsql. Interfaz de Programación de Aplicación (API), encargada de la comunicación de una aplicación .Net con servidores de Bases de Datos PostgreSQL. Este Acceso a Datos está conformado por un conjunto de funciones programadas en lenguaje Npgsql.

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

Como parte de la estrategia de implementación que se desarrolla en el proyecto se utiliza para el mejor desarrollo de trabajo el uso de los controles web que no son más que elementos que se pueden arrastrar a la página, lo cual permite manejar de forma muy sencilla sus propiedades y eventos asociados mediante códigos. Los controles web en múltiples ocasiones son utilizados para agrupar funciones comunes en los distintos servicios, por los mismo se le denominan componente reutilizable.

A continuación se listan algunos de los controles web utilizados en el desarrollo del sistema:

- ❖ Datos generales de la consulta.
- ❖ Datos del paciente.
- ❖ Datos consulta.
- ❖ Valoración.
- ❖ Exámenes oftalmológicos.
- ❖ Hábitos tóxicos.
- ❖ Antecedentes patológicos.
- ❖ Antecedentes patológicos oftalmológicos.

El Sistema Bloque Quirúrgico Oftalmológicos (BQO) necesita intercambiar información con el sistema instalado en el resto de las áreas hospitalarias, la cual se va a realizar a partir de Servicios Webs WSDL, intercambiando información referente a:

Inscripción-Admisión: A través de este módulo se realizan las búsquedas de los pacientes. Una vez seleccionado el paciente al cual se le realizará la consulta, muestra toda la información correspondiente a este, como por ejemplo: número de identificación o carnet de identidad, nombre y apellidos, sexo, edad, municipio, provincia, país, sala y cama en caso de estar ingresado, entre otros.

Laboratorio Clínico: Proveedor de la información referente a los resultados de los análisis que se le han indicado al paciente.

Farmacia: Abastecedor y controlador de los medicamentos que están disponibles, así como los materiales gastables implicados en cada una de las intervenciones quirúrgicas.

Agenda médica y las citas: Con el objetivo de poder ver desde la consultas las citas que el médico tiene para una fecha determinada. Además de dar citas de interconsultas y consultas de seguimiento con otros servicios y consigo mismo desde la consulta.

Sistema PACS/RIS. Con el objetivo de lograr visualizar y gestionar las imágenes médicas desde la consulta y integrar la HC del paciente con los estudios imagenológicos

Estadísticas: Para poder tratar de manera homogénea la estadística generada en la institución.

Estrategia de Integración

Dentro de un mismo componente los códigos se integran o se comunican mediante las llamadas que se realizan de forma directa a eventos o métodos. La comunicación con los distintos componentes se realiza de forma directa a nivel de negocio. (22) El sistema utilizará SAAA (EMP) como servicio web donde la información transmitida estará regida por el formato Web Services Description Language (WSDL), un formato XML que se utiliza para describir servicios Web.

De forma directa la base de datos son accedidas mediante las clases controladoras y los componentes reutilizados son integrados mediante interfaces sencillas. (23)

Un control de origen de datos de ASP.NET que representa un objeto de la interfaz de datos con controles enlazados a datos o un objeto de nivel medio orientado a datos es el objeto

ObjectDataSource, el mismo junto con un control enlazado a datos para presentar, editar y ordenar datos de una página Web que contenga poco código o no lo contenga.

Este componente es utilizado para el módulo de configuración que tiene varias ventajas una de ellas es que permite a los desarrolladores utilizar un control de origen de datos de ASP.NET y a la vez retener su arquitectura de aplicación de tres niveles y otra de las ventajas que ofrece es que utiliza la reflexión para crear instancias de objetos comerciales y para llamar a los métodos que contienen para recuperar, actualizar, insertar y eliminar datos. (24)

2.4 Seguridad

La seguridad de cualquier sistema está dada por la capacidad que tenga el mismo para que la información se encuentre disponible en el momento que se necesite y así un usuario que tengas los permisos autorizados pueda acceder o cambiar cualquier información que necesite, poniendo en práctica así los principio de la se seguridad como la confidencialidad la integridad y disponibilidad de la información.

Para garantizar lo antes mencionado el sistema basa la seguridad en cuanto a la autenticación al componente SAAA el cual constituye un pilar en su funcionamiento. El framework .NET proporcionan clases y servicios que facilitan el uso de la criptografía y la seguridad basada en funciones y junto con el Common Language Runtime existe un número de clases y servicios útiles que permiten a los programadores escribir de manera sencilla código de seguridad, estos servicios y clases le permiten a los administradores personalizar el acceso del código a los recursos protegidos.

El framework .NET proporciona un mecanismo de seguridad denominado seguridad de acceso a código permitiendo proteger los sistemas informáticos del código móvil malicioso, además ayuda a evitar que el código de confianza ponga en peligro la seguridad y permite que el código de origen desconocido se ejecute con protección.

Al utilizar la seguridad de acceso a código se puede reducir la posibilidad de que un código maligno o con errores utilice incorrectamente el código del sistema. Se reduce la responsabilidad del usuario, ya que puede especificar el conjunto de operaciones que debe permitirse al código así como las operaciones que nunca deben permitírsele. La seguridad de acceso a código también puede ayudar a reducir al mínimo el daño causado por los puntos de seguridad vulnerables del código.

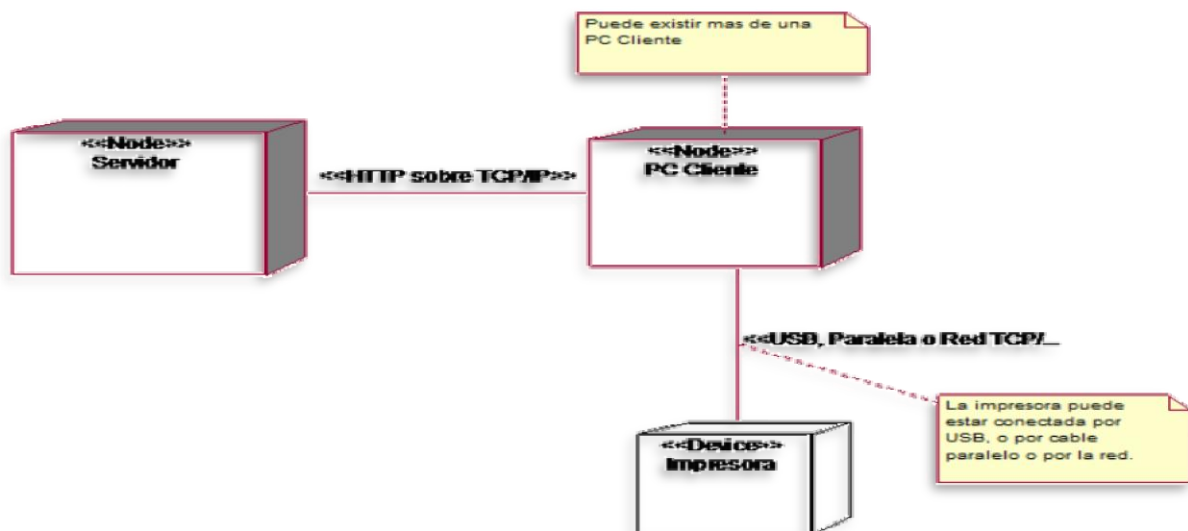
2.5 Vista de Despliegue

Una vista es una representación de un conjunto de elemento del sistema y sus relaciones o de algunas de las estructuras presentes al mismo tiempo en un sistema de software. (25)

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. La misma está encargada de los requerimientos o requisitos no funcionales como desempeño, disponibilidad, escalabilidad y confiabilidad. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos.

Los diagramas de despliegue son grafos de nodos unidos por conexiones de comunicación. Un diagrama de este tipo muestra la relación física entre los diferentes nodos que forman un sistema, donde un nodo sería una unidad de computación de algún tipo.

Diagrama de despliegue



2.6 Estrategias de codificación. Estándares y estilos a utilizar.

Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

La legibilidad del código fuente repercute directamente en lo bien que un programador interpreta un sistema de software. La estabilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

Aunque la legibilidad y la estabilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas.

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

Aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una mejor comprensión del código fuente. En esta fase se tienen en cuenta todos los tipos de código fuente, incluidos los lenguajes de programación, de marcado o de consulta.

En general una técnica de codificación no pretende formar un conjunto inflexible de estándares de codificación. Más bien intenta servir de guía en el desarrollo de un estándar de codificación para un proyecto específico de software.

Cuando se trabaja en equipo es necesario hacer código legible y entendible no sólo para quien lo escribe, sino también para quien lo lee, y para eso es necesario tener en cuenta varios aspectos:

- ❖ Las cláusulas, la notación que se utilizará para nombrar cada uno de los identificadores que se declaran.
- ❖ La estructura del código en sí, referente a las tabulaciones y los espacios entre líneas y dentro de las líneas, los espacios entre los operadores y estructuras que componen el lenguaje en que se desarrolla la aplicación.

El uso de los estándares de codificación en la investigación, están presentes de la siguiente manera.

Desarrollo de los servicios oftalmológicos Neuroftalmología, Glaucoma, Uveítis y Córnea.

Notación Camello

Se usa para denotar variables y parámetros. En esta notación, si el identificador es una palabra simple se escribe todo con minúscula, pero si es compuesta, la primera letra de todas las palabras que viene a continuación de la primera comienza con mayúscula.

Ejemplo:

```
private Int32 idPersona;
```

```
private Int32 numeroConsulta;
```

```
public List<Consulta> BuscarConsulta(Int32 numeroConsulta, DateTime fechaConsulta)
```

```
{  
    //...  
}
```

Notación Pascal:

En la notación Pascal, si el identificador es simple, el primer carácter se escribe con mayúscula y el resto con minúscula; si el identificador es una palabra compuesta, la segunda palabra debe empezar con mayúscula también.

Es necesario seguir algunas convenciones específicas para los distintos tipos de datos que se mencionan a continuación.

Espacios de nombres (namespaces).

No deben contener espacios y deben definir claramente el conjunto que representan, cada espacio estará separado por punto ".". Si el identificador del espacio de nombres es pequeño (tres o menos caracteres) se escribirá enteramente con mayúscula.

Ejemplo:

```
namespace Gehos.Negocio.BQO
```

```
{  
    //...  
}
```

Clases:

Los nombres de las clases deben ajustarse a la entidad que representan, y su primera palabra debe ser un sustantivo. Si con una sola palabra no se puede nombrar dicha entidad, la segunda palabra debe ser un adjetivo, a menos que la palabra sea compuesta.

Ejemplo:

```
public partial class DatosConsulta
{
    //...
}
```

Métodos:

Los nombres de métodos deben describir la acción que realizan, y si el identificador es compuesto, la primera palabra debe ser el infinitivo de la acción. Ejemplo:

```
public List<DatosPersona> BuscarPaciente (DatosPersona persona)
{
    //...
}
```

Propiedades (Properties):

Si modifican o devuelven algún atributo perteneciente a una clase, debe tener el mismo nombre del atributo, pero su primera letra debe ser mayúscula. De otra forma deben seguir las cláusulas de los métodos.

Ejemplo:

```
public String DatosConsulta
{
    get
    {
        return datosConsulta;
    }
    set
    {
        datosConsulta = value;
    }
}
```

Componentes:

Para denotar los componentes se debe mantener la primera palabra que predefine Visual Studio para ellos y agregarle una palabra que empiece con mayúscula, que defina la acción que realiza o los datos que representa

Ejemplo:

- ❖ TextBoxNombre
- ❖ ButtonAceptar
- ❖ DropDownListServicio
- ❖ GridViewListadoConsulta

Ejemplo:

```
public Int32 IdPersona
{
    get
    {
        return this.idPersona;
    }
    set
    {
        this. idPersona = value;
    }
}
```

Dentro de cada condicional o estructura de control el código deberá cumplir con las siguientes condiciones:

- ❖ Una línea para abrir llave.
- ❖ Una línea para cerrar llave.
- ❖ El margen entre las llaves y las sentencias debe ser de una tabla.

Ejemplo:

```
for (int i = 0; i < listadoConsulta.Cont; i++)
{
    if (listadoConsulta[i].Activa )
    {
        //....
    }
}
```

```
}  
}
```

Como se pudo observar los estilos de código hacen que el programa fuente sea más legible, lo cual ayuda en la comunicación entre desarrolladores, y permite una temprana detección de errores. La propuesta es completamente extensible.

Siempre existe un riesgo cuando se definen estilos de codificación en aplicar más prefijos o sufijos en la sintaxis a conceptos que ya tienen una forma de ser expresados, cayendo en una redundancia expresiva.

En este capítulo se determinó la concepción arquitectónica y la seguridad del sistema. Se definieron los requisitos no funcionales que debe presentar el sistema. Se construyó el diagrama de despliegue del sistema visualizando su estructura física. Además, se definió la estrategia de codificación y los estándares y estilos a utilizar.

Capítulo 3. Descripción y análisis de la solución propuesta

En el presente capítulo se realiza una valoración crítica por los analistas donde se presentan los diagramas de clases del diseño y los diagramas de interacción. Además se describen las clases u operaciones nuevas que son necesarias a la hora de realizar el sistema. Se muestra el modelo de datos con una breve valoración de las técnicas de validación. Este capítulo 3 cuenta también con vista de Implementación donde se propone el diagrama de componentes.

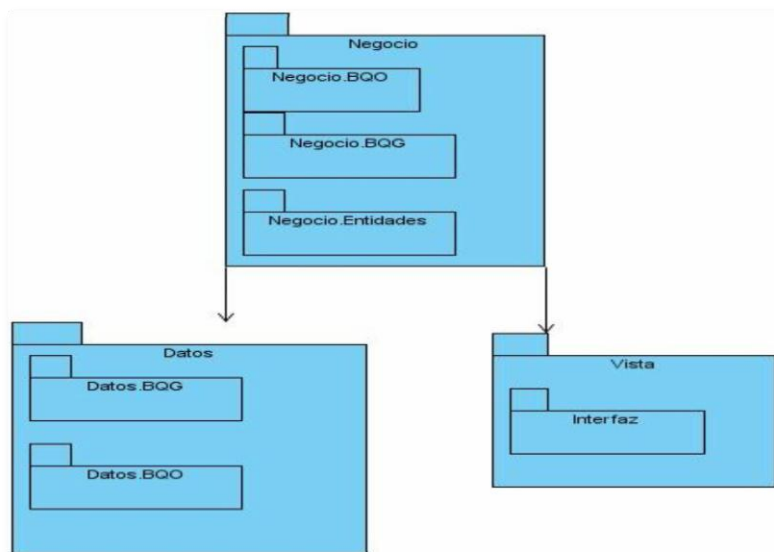
3.1 Valoración crítica del diseño propuesto por el analista.

Una vez terminado el análisis del diseño propuesto por los analistas se pudieron definir las clases principales, atributos y métodos que deben contener las mismas, mostrando con claridad lo que se quiere implementar. Además se muestra los diagramas del diseño y de interacción el cual este último muestran un patrón de interacción entre eventos. Existen dos tipos de diagrama de interacción los diagramas de colaboración y los diagramas de secuencia, este último mencionado muestra una interacción ordenada según la secuencia temporal de eventos.

El diseño propuesto se creó basándose bajo los patrones GOF y GRASP. El patrón GRASP se utiliza con objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño.

Para el desarrollo de los diagramas de clases del diseño anteriormente se deben haber realizados los diagramas de interacción. Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces. (26)

Diagrama de paquetes



Diagramas de secuencia

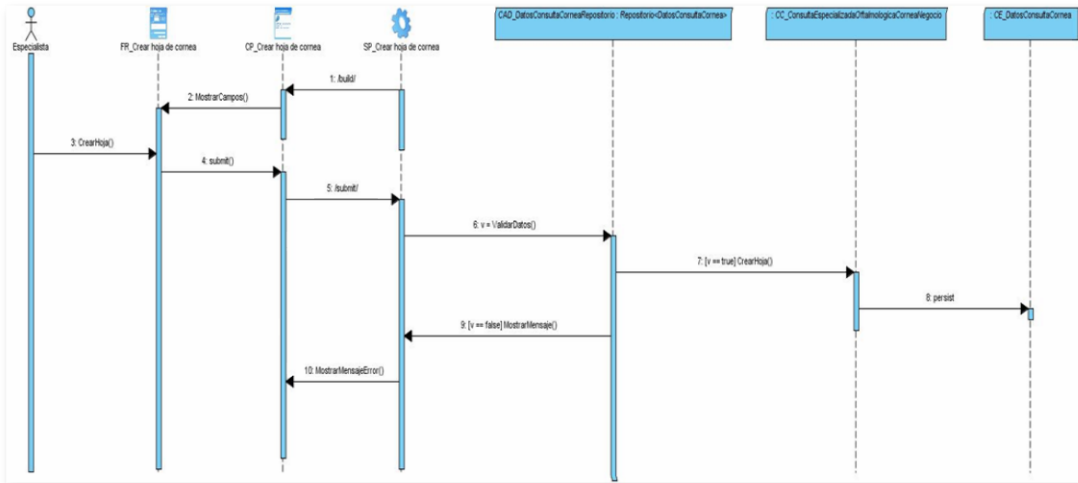


Diagrama de secuencia Crear Hoja de consulta de Córnea

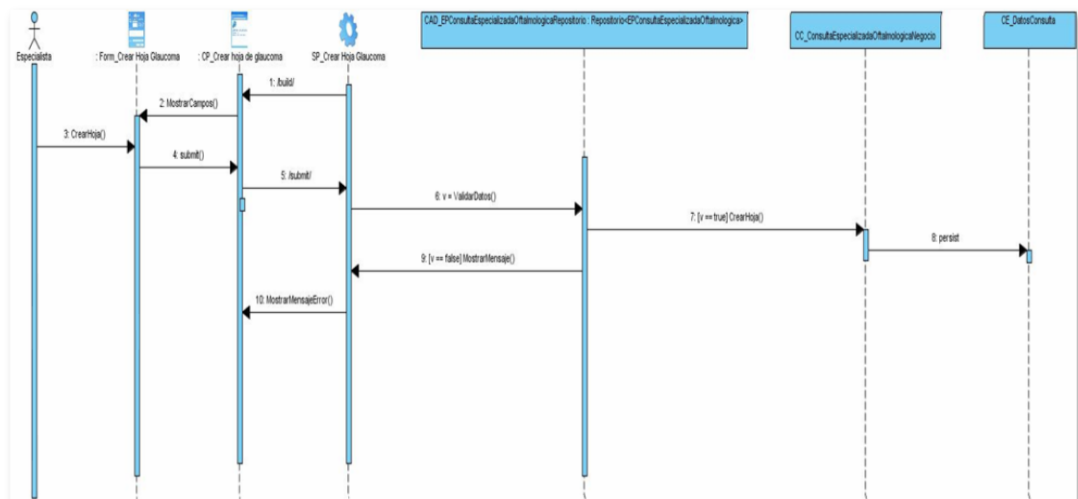


Diagrama de secuencia Crear Hoja de consulta de Glaucoma

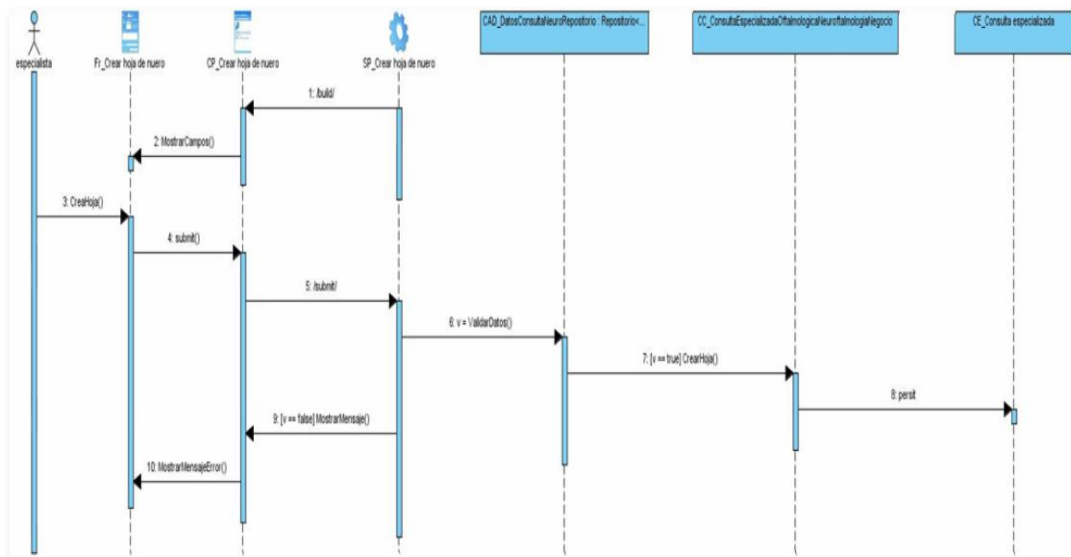


Diagrama de secuencia Crear Hoja de consulta de Neuroftalmología

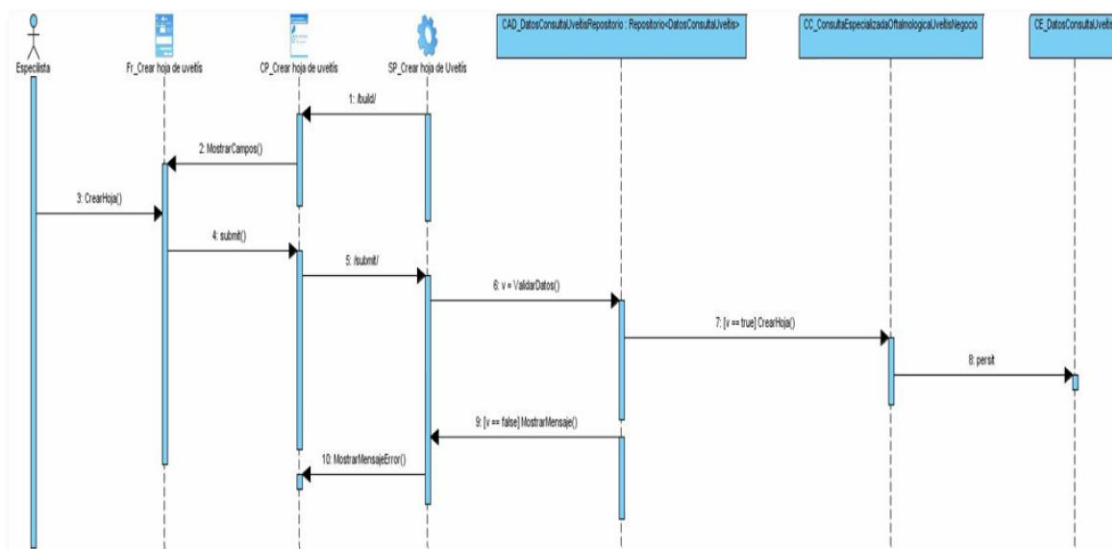
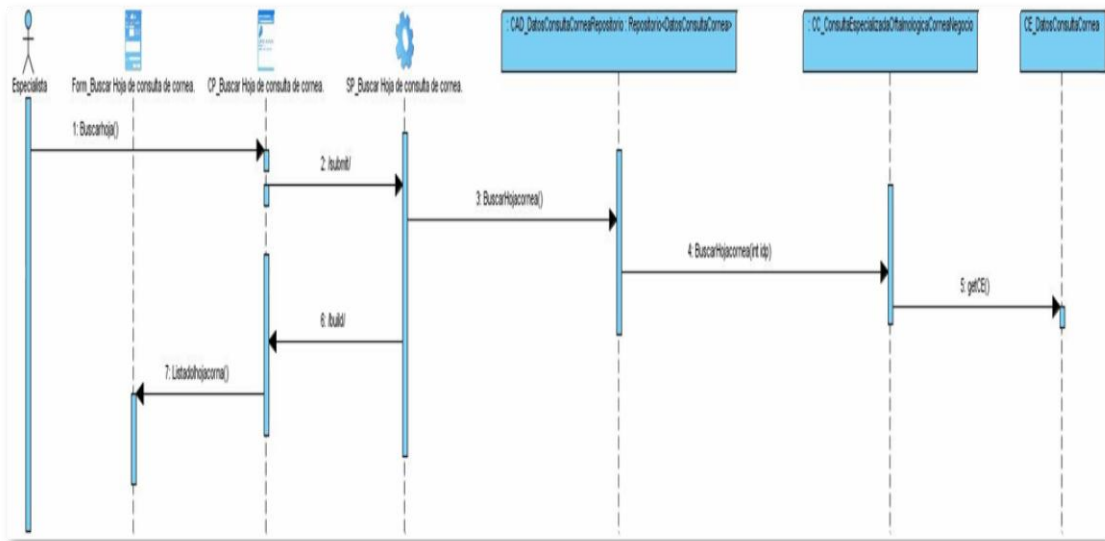
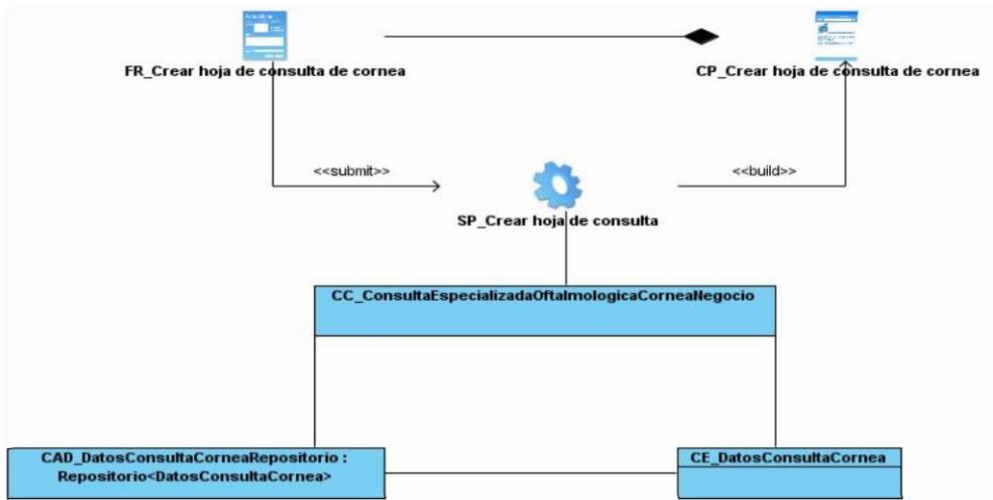


Diagrama de secuencia Crear Hoja de consulta de Uveítis

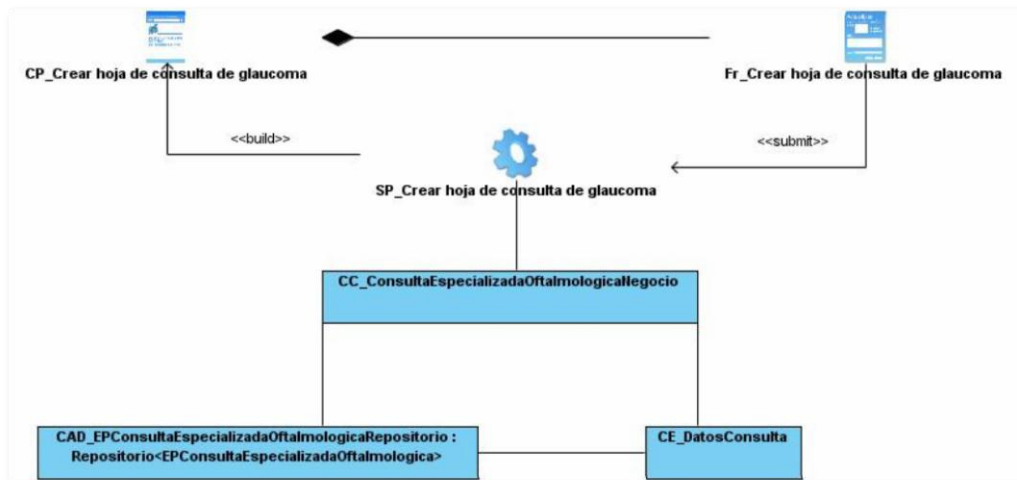


Buscar Hoja de consulta de Córnea

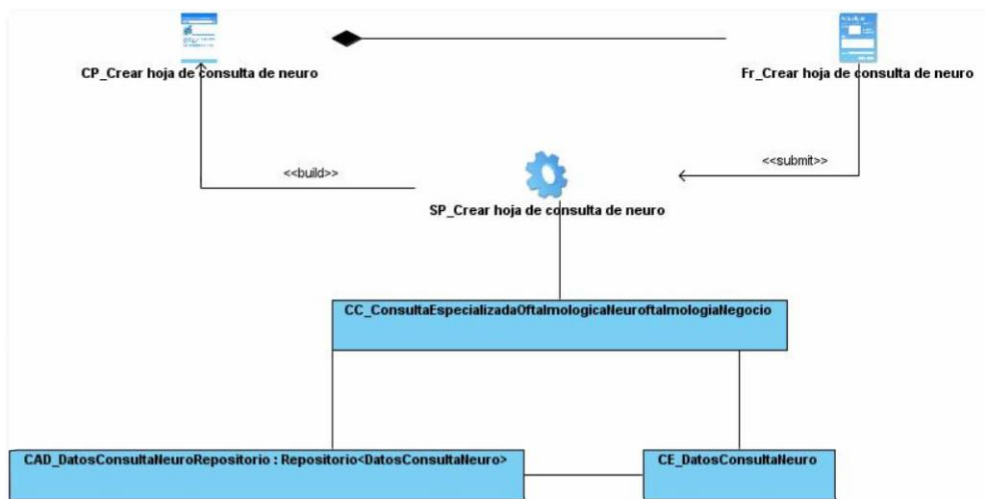
Diagramas de diseño



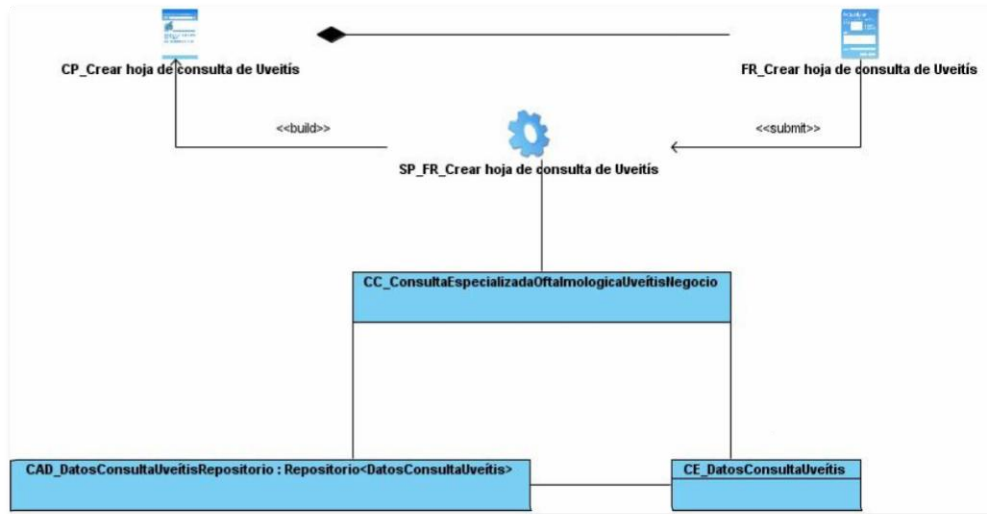
Crear hoja de consulta de Córnea



Crear hoja de consulta de Glaucoma



Crear hoja de consulta de Neurooftalmología



Crear hoja de consulta de Uveítis

Clases Controladoras

Nombre: ConsultaEspecializadaOftalmologicaNegocio	
Tipo de clase: Controladora	
Atributo	Tipo
consultaEspecializadaOftalmologicaRepositorio	EPConsultaEspecializadaOftalmologicaRepositorio
anuncioOperatoriooftalmologicoRepositorio	EPAuncioOperatorioOftalmologicoRepositorio
antecedentePatologicoOftalmologicoRepositorio	AntecedentePatologicoOftalmologicoRepositorio
Para cada responsabilidad:	
Nombre:	ConsultaEspecializadaOftalmologicaNegocio()
Descripción:	Inicializa el constructor
Nombre:	ObtenerUno(EPConsultaEspecializadaOftalmologica consultaEspecializadaOftalmologica)
Descripción:	Obtiene una consulta oftalmológica especializada

Capítulo 3. Descripción y análisis de la solución propuesta.

Nombre:	Adicionar(EPConsultaEspecializadaOftalmologica consultaEspecializadaOftalmologica)
Descripción:	Adiciona una consulta oftalmológica especializada
Nombre:	Actualizar(EPConsultaEspecializadaOftalmologica consultaEspecializadaOftalmologica)
Descripción:	Actualiza una consulta oftalmológica especializada
Nombre:	ObtenerUno(EPAuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Obtiene un anuncio operatorio oftalmológico
Nombre:	Adicionar(EPAuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Adiciona un anuncio operatorio oftalmológico
Nombre:	Actualizar(EPAuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Actualiza un anuncio operatorio oftalmológico
Nombre:	ObtenerTodos(AntecedentePatologicoOftalmologico antecedentePatologicoOftalmologico)
Descripción:	Obtiene todos los antecedentes patológicos oftalmológicos
Nombre:	ObtenerTodosAnteriores(AntecedentePatologicoOftalmologico antecedentePatologicoOftalmologico)
Descripción:	Obtiene todos los anteriores antecedentes patológicos que tenía un paciente

Tabla 1 Clase controladora ConsultaEspecializadaOftalmologicaNegocio

Nombre: ConsultaEspecializadaOftalmologicaCorneaNegocio	
Tipo de clase: Controladora	
Atributo	Tipo
consultaEspecializadaOftalmologicaCorneaRepositorio	EPConsultaEspecializadaOftalmologicaCorneaRepositorio

Capítulo 3. Descripción y análisis de la solución propuesta.

Para cada responsabilidad:	
Nombre:	ConsultaEspecializadaOftalmologicaCorneaNegocio ()
Descripción:	Inicializa el constructor
Nombre:	ObtenerUno(EPConsultaEspecializadaOftalmologicaCornea consultaEspecializadaOftalmologicaCornea)
Descripción:	Obtiene una consulta oftalmológica especializada de Cornea
Nombre:	Adicionar(EPConsultaEspecializadaOftalmologicaCornea consultaEspecializadaOftalmologicaCornea)
Descripción:	Adiciona una consulta oftalmológica especializada de Cornea
Nombre:	Actualizar(EPConsultaEspecializadaOftalmologicaCornea consultaEspecializadaOftalmologicaCornea)
Descripción:	Actualiza una consulta oftalmológica especializada de Cornea

Tabla 2 Clase controladora ConsultaEspecializadaOftalmologicaCorneaNegocio

Nombre: ConsultaEspecializadaOftalmologicaUveítisNegocio	
Tipo de clase: Controladora	
Atributo	Tipo
consultaEspecializadaOftalmologicaUveítisRepositorio	EPConsultaEspecializadaOftalmologicaUveítisRepositorio
Para cada responsabilidad:	
Nombre:	ConsultaEspecializadaOftalmologicaUveítisNegocio ()
Descripción:	Inicializa el constructor
Nombre:	ObtenerUno(EPConsultaEspecializadaOftalmologicaUveítis consultaEspecializadaOftalmologicaUveítis)

Capítulo 3. Descripción y análisis de la solución propuesta.

Descripción:	Obtiene una consulta oftalmológica especializada de Uveítis
Nombre:	Adicionar(EPConsultaEspecializadaOftalmologicaUveítis consultaEspecializadaOftalmologica Uveítis)
Descripción:	Adiciona una consulta oftalmológica especializada de Uveítis
Nombre:	Actualizar(EPConsultaEspecializadaOftalmologicaUveítis consultaEspecializadaOftalmologica Uveítis)
Descripción:	Actualiza una consulta oftalmológica especializada de Uveítis

Tabla 3 Clase controladora ConsultaEspecializadaOftalmologicaUveítisNegocio

Nombre: ConsultaEspecializadaOftalmologicaNeuroftalmologiaNegocio	
Tipo de clase: Controladora	
Atributo	Tipo
consultaEspecializadaOftalmologicaNeuroftalmologiaR epositorio	PConsultaEspecializadaOftalmologicaNeuroftalmologiaR epositorio
Para cada responsabilidad:	
Nombre:	ConsultaEspecializadaOftalmologicaNeuroftalmologiaNegocio ()
Descripción:	Inicializa el constructor
Nombre:	ObtenerUno(EPConsultaEspecializadaOftalmologicaNeuroftalmologia consultaEspecializadaOftalmologicaNeuroftalmologia)
Descripción:	Obtiene una consulta oftalmológica especializada de Neuroftalmología
Nombre:	Adicionar(EPConsultaEspecializadaOftalmologicaNeuroftalmologia consultaEspecializadaOftalmologicaNeuroftalmologia)
Descripción:	Adiciona una consulta oftalmológica especializada de Neuroftalmología
Nombre:	Actualizar(EPConsultaEspecializadaOftalmologicaNeuroftalmologia consultaEspecializadaOftalmologicaNeuroftalmologia)

Descripción:	Actualiza una consulta oftalmológica especializada de Neurooftalmología
--------------	---

Tabla 4 Clase controladora ConsultaEspecializadaOftalmologicaNeurooftalmologiaNegocio

Nombre: AnuncioOperatorioOftalmologicoNegocio	
Tipo de clase : Controladora	
Atributo	Tipo
anuncioOperatoriooftalmologicoRepositorio	EPAnuncioOperatorioOftalmologicoRepositorio
Para cada responsabilidad:	
Nombre	AnuncioOperatorioOftalmologicoNegocio()
Descripción	Inicializa el constructor
Nombre:	ObtenerTodos(EPAnuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Obtiene todos los anuncio operatorio oftalmológicos
Nombre:	ObtenerUno(EPAnuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Obtiene un anuncio operatorio oftalmológico
Nombre:	Adicionar(EPAnuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Adiciona un anuncio operatorio oftalmológico
Nombre:	Actualizar(EPAnuncioOperatorioOftalmologico anuncioOperatorioOftalmologico)
Descripción:	Actualiza un anuncio operatorio oftalmológico

Tabla 5 Clase controladora AnuncioOperatorioOftalmologicoNegocio

Nombre: InformeOperatorioOftalmologicoNegocio	
Tipo de clase : Controladora	
Atributo	Tipo
epInformeOperatorioOftalmologicoRepositorio	EPInformeOperatorioOftalmologicoRepositorio
Para cada responsabilidad:	

Nombre	InformeOperatorioOftalmologicoNegocio()
Descripción	Inicializa el constructor
Nombre:	ObtenerUno(EPInformeOperatorioOftalmologico eplInformeOperatorioOftalmologico)
Descripción:	Obtiene un informe operatorio oftalmológico
Nombre:	ObtenerTodos(EPInformeOperatorioOftalmologico eplInformeOperatorioOftalmologico)
Descripción:	Obtiene todos los informes operatorios oftalmológicos
Nombre:	Adicionar(EPInformeOperatorioOftalmologico eplInformeOperatorioOftalmologico)
Descripción:	Adiciona un informe operatorio oftalmológico
Nombre:	Actualizar(EPInformeOperatorioOftalmologico eplInformeOperatorioOftalmologico)
Descripción:	Actualiza un informe operatorio oftalmológicos

Tabla 6 Clase controladora InformeOperatorioOftalmologicoNegocio

Clases entidades del negocio

Nombre: DatosConsulta	
Tipo de clase : Entidades	
Atributo	Tipo
numeroConsulta	Nullable<Int32>
idPersona	Nullable<Int32>
fechaConsulta	Nullable<DateTime>
motivoConsulta	String
impresionDiagnostico	String
historiaEnfermedadActual	String
sensibilidadMedicamentosa	String
idTipoConsulta	Nullable<Int32>
descripcionHabitoToxico	String
descripcionOtrosApp	String
idFuncionario	Nullable<Int32>

Para cada responsabilidad:	
Nombre:	DatosConsulta (Nullable<Int32> numeroConsulta, Nullable<Int32> idPersona, Nullable<DateTime> fechaConsulta, String motivoConsulta, String impresionDiagnostico, String historiaEnfermedadActual, String sensibilidadMedicamentosa, Nullable<Int32> idTipoConsulta, String descripcionHabitoToxico, String descripcionOtrosApp, Nullable<Int32> idFuncionario)
Descripción:	Inicializa los atributos de la clase
Nombre:	DatosConsulta ()
Descripción:	Inicializa el constructor de la clase vacio

Tabla 7 Clase entidad DatosConsulta

Nombre: DatosConsultaCornea	
Tipo de clase : Entidades	
Atributo	Tipo
idPersona	Nullable<Int32>
numeroConsulta	Nullable<Int32>
idMedicamento	Nullable<Int32>
idVia	Nullable<Int32>
dosis	Nullable<Double>
Para cada responsabilidad:	
Nombre:	DatosConsultaCornea(Nullable<Int32> idPersona, Nullable<Int32> numeroConsulta, Nullable<Int32> idMedicamento, Nullable<Int32> idVia, Nullable<Double> dosis)
Descripción:	Inicializa los atributos de la clase
Nombre:	DatosConsultaCornea ()
Descripción:	Inicializa el constructor de la clase vacio

Tabla 8 Clase entidad DatosConsultaCornea

Nombre: DatosConsultaUveítis

Tipo de clase : Entidades	
Atributo	Tipo
idPersona	Nullable<Int32>
numeroConsulta	Nullable<Int32>
idMedicamento	Nullable<Int32>
idVia	Nullable<Int32>
dosis	Nullable<Double>
Para cada responsabilidad:	
Nombre:	DatosConsultaUveítis (Nullable<Int32> idPersona, Nullable<Int32> numeroConsulta, Nullable<Int32> idMedicamento, Nullable<Int32> idVia, Nullable<Double> dosis)
Descripción:	Inicializa los atributos de la clase
Nombre:	DatosConsultaUveítis ()
Descripción:	Inicializa el constructor de la clase vacio

Tabla 9 Clase entidad DatosConsultaUveítis

Nombre: DatosConsultaNeuro	
Tipo de clase : Entidades	
Atributo	Tipo
idPersona	Nullable<Int32>
numeroConsulta	Nullable<Int32>
tratamientoMedicoInicial	Nullable<Boolean>
tratamientoMedicoMantenimiento	Nullable<Boolean>
tratamientoMedicoCulminado	Nullable<Boolean>
otroTratamiento	String
Para cada responsabilidad:	
Nombre:	DatosConsultaUveítis (Nullable<Int32> idPersona, Nullable<Int32> numeroConsulta,

	Nullable<Int32> idMedicamento, Nullable<Int32> idVia, Nullable<Double> dosis)
Descripción:	Inicializa los atributos de la clase
Nombre:	DatosConsultaUveítis ()
Descripción:	Inicializa el constructor de la clase vacío

Tabla 10 Clase entidad DatosConsultaNeuro

Nombre: EPAnuncioOperatorioOftalmologico : EPAnuncioOperatorioGeneral	
Tipo de clase : Entidades	
Atributo	Tipo
anuncioOftalmologico	AnuncioOftalmologico
anuncioCirugiaCrist	AnuncioCirugiaCrist
List<LenteAnuncio>	listLenteAnuncio
List<DisponibilidadLente>	disponibilidadLente
List<DisponibilidadLente>	disponibilidadLenteAnterior
List<CausasNoImplantesAnuncio>	causasNoImplanteAnuncio
List<PacientePendienteLio>	listaPacientePendienteLente
EPPlanificacion	epPlanificacion
EPPlanificacion	epPlanificacionAnterior
Para cada responsabilidad:	
Nombre:	EPAnuncioOperatorioOftalmologico(EPAnuncioOperatorioOftalmologico anuncioOperatorioOftalmologico) :base(anuncioOperatorioOftalmologico)
Descripción:	Inicializa los atributos de la clase

Tabla 11 Clase entidad EPAnuncioOperatorioOftalmologico

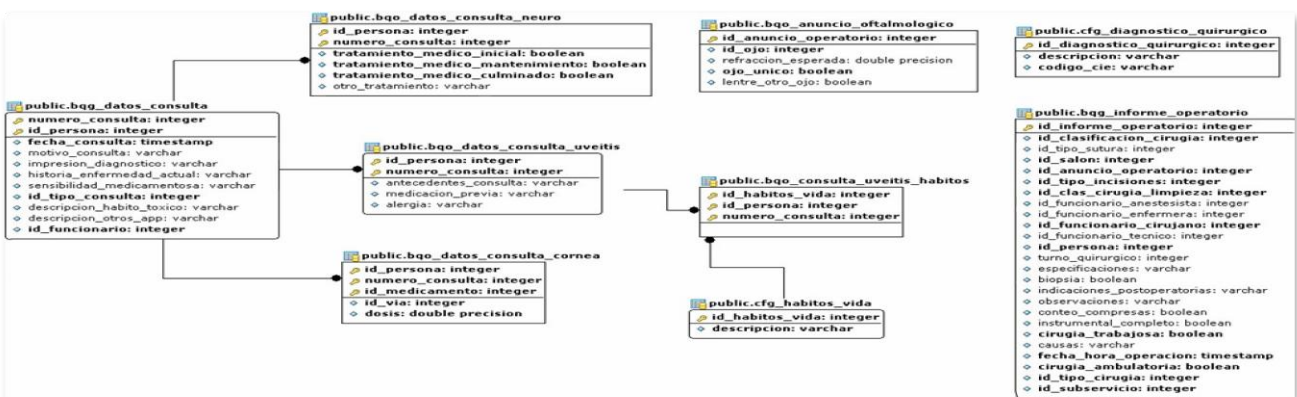
Nombre: EPInformeOperatorioOftalmologico : EPInformeOperatorio	
Tipo de clase : Entidades	
Atributo	Tipo

informeOperatorioOftalmologico	InformeOperatorioOftalmologico
EPInformeCirugiaCristalino	epInformeCirugiaCristalino
InformeCirugiaRefractiva	informeCirugiaRefractiva
EPInformeOperatorio	epInformeOperatorio
Para cada responsabilidad:	
Nombre:	EPInformeOperatorioOftalmologico()
Descripción:	Inicializa el constructor

Tabla 12 Clase entidad EPInformeOperatorioOftalmologico

3.2 Modelo de datos y Breve valoración de las Técnicas de validación (Integridad, Normalización de la Base de datos)

Modelo de dato



Modelo de datos

Breve valoración de las Técnicas de validación

La creación de un sistema informático implica garantizar que el mismo posea la capacidad de responder ante cualquier circunstancia, para lo cual se realizan técnicas de validación para simular todas las posibles situaciones que se puedan presentar. Existe un gran número de técnicas de validación, las cuales deben de estar dirigidas al supuesto comportamiento erróneo que tendrá el sistema. Para evitar sucesos imprevistos es que se ponen en práctica métodos y estrategias que facilitan todo este trabajo.

Uno de los puntos a tener en cuenta en las técnicas de validación son las excepciones y el tratamiento de errores. Cuando se produce un error, el sistema o la aplicación que se está ejecutando en ese momento informa del mismo produciendo una excepción que contiene información sobre el error. Una vez que se produce una excepción, la aplicación o el controlador de excepciones predeterminado controlan dicha excepción.

Common Language Runtime proporciona un modelo de control de excepciones basado en la representación de excepciones en forma de objetos, y en la separación del código de programa y del código de control de excepciones en bloques try y catch, respectivamente.

En el sistema el control de las excepciones se lleva a cabo en todas las porciones de código donde pueda surgir alguna situación imprevista y además, donde se ejecutan sentencias que manipulan los datos que viajan entre la aplicación y la base de datos. También se controla la entrada de datos por el usuario, comprobando así que la información introducida por los usuarios es válida.

Para controlar los posibles errores en la entrada de datos del lado del cliente se utiliza jQuery y JavaScript, proporcionando funcionalidades capaces de detectar entradas inválidas de información sin que esta viaje hasta el servidor. Además ASP.NET ofrece un conjunto de controles de validación que proporcionan una forma eficaz y fácil de usar para comprobar errores y, si es necesario, mostrar mensajes al usuario.

Los controles de validación incorporan lógica para permitir comprobar los controles de entrada de los usuarios. Dichos controles proporcionan un mecanismo fácil de utilizar para todos los tipos comunes de validación estándar (por ejemplo, probar fechas válidas o valores comprendidos en un intervalo), además de otras formas para permitir personalizar completamente cómo se muestra la información de errores al usuario. Los controles de validación se pueden utilizar con cualquier control que se coloque en una página Web ASP.NET, incluidos los controles HTML y de servidor Web.

Del lado del servidor también se realiza la validación aunque ya se haya llevado a cabo en el cliente. Esto permite determinar el estado de la validación en el código del servidor y proporciona seguridad contra la omisión de la validación por el usuario.

Para garantizar dicha validación se emplea Validation, una librería de clases que permite verificar la correcta entrada de los datos por el usuario en los exámenes oftalmológicos, asegurando una adecuada introducción de valores como el cilindro, eje, esfera y poder dióptrico.

El proceso de validación no solo se reduce a garantizar una correcta entrada de datos por el usuario desde la interfaz, también tiene a su cargo garantizar una absoluta integridad de la información almacenada en la base de datos. Con dicho propósito se emplean las transacciones, una forma de controlar si la actualización de datos en una base de datos se deben aceptar, si todo ha ido correctamente, o se deben descartar en caso de producirse un error, esto nos asegura que esa actualización de datos se haga solamente si todo ha ido como teníamos previsto.

Las transacciones se emplean en la realización de varias tareas sobre la base de datos, ya sea realizando tareas de actualización, inserción o eliminación, por ejemplo si intervienen varias tablas, ya que se pueda actualizar los datos de una tabla y al intentar actualizar otra, es cuando se produce un error, en este caso, se puede deshacer los cambios realizados en la primera tabla y cancelar toda la operación, de esta forma se garantiza que no queden datos "colgados" y que sólo tienen sentido si se realiza el proceso completo.

El proceso de normalización de una base de datos (BD) permite obtener estructuras de datos eficientes que eviten las anomalías de actualización. Este proceso de normalización mejora la independencia de los datos, permitiendo realizar extensiones de la base de datos afectando muy poco, o nada, a los programas de aplicación existentes que accedan la base de datos.

La normalización involucra varias fases que se realizan en orden la primera, segunda y tercera forma normal, en esta última se encuentran en las BD del sistema que se encuentra en desarrollo actualmente, la cual debe cumplir que este en segunda forma normal y que no exista ningún atributo no principal (que no pertenezca a la llave primaria) que dependa transitivamente de algún atributo parte de la llave primaria.

Descripción de las tablas

Tabla bqo_datos_consulta_neuro

Capítulo 3. Descripción y análisis de la solución propuesta.

Nombre: bqo_datos_consulta_neuro		
Descripción: Guarda los datos específicos para esta consulta		
Atributo	Tipo	Descripción
id_persona	INTEGER	Identificador del paciente
numero_consulta	INTEGER	Numero de la consulta
tratamiento_medico_inicial	BOOLEAN	Recoge si el tratamiento indicado al paciente es inicial
tratamiento_medico_mantenimiento	BOOLEAN	Recoge si el tratamiento indicado al paciente es de seguimiento
tratamiento_medico_culminado	BOOLEAN	Recoge si el tratamiento indicado al paciente es definitivo
otro_tratamiento	BOOLEAN	Recoge otro tratamiento para indicarle al paciente

Tabla bqo_datos_consulta_uveitis

Nombre: bqo_datos_consulta_uveitis		
Descripción: Guarda los datos específicos para esta consulta		
Atributo	Tipo	Descripción
id_persona	INTEGER	Identificador del paciente
numero_consulta	INTEGER	Numero de la consulta
antecedentes_consulta	VARCHAR	Recoge las crisis anteriores que ha tenido el paciente
medicacion_previa	VARCHAR	Recoge todos los medicamentos que esté administrándose al paciente en el momento de la consulta
alergia	VARCHAR	Recoge todas las alergias que padezca el paciente

Tabla bqo_datos_consulta_cornea

Nombre: bqo_datos_consulta_cornea		
Descripción: Guarda los datos específicos para esta consulta		
Atributo	Tipo	Descripción
id_persona	INTEGER	Identificador del paciente
numero_consulta	INTEGER	Numero de la consulta
id_medicamento	INTEGER	Identificador del medicamento
id_via	INTEGER	Identificador de la vía que se le suministrara el medicamento al paciente
dosis	DOUBLE PRECISION	Cantidad de medicamento que se le suministrara al paciente

Tabla bqo_anuncio_ofthalmologico

Nombre: bqo_anuncio_ofthalmologico		
Descripción: Guarda todos los datos que se recogen en el anuncio operatorio		
Atributo	Tipo	Descripción
id_anuncio_operatorio	INTEGER	Identificador del anuncio operatorio
id_ojo	INTEGER	Identificador del ojo
refraccion_esperada	DOUBLE PRECISION	Medida esperada
ojo_unico	BOOLEAN	Si el paciente tiene un solo ojo o no
lente_otro_ojo	BOOLEAN	Si el paciente tiene lente el otro ojo o no

Tabla bqg_informe_operatorio

Nombre: bqg_informe_operatorio		
Descripción: Guarda todos los datos que se recogen en el informe operatorio		
Atributo	Tipo	Descripción

Capítulo 3. Descripción y análisis de la solución propuesta.

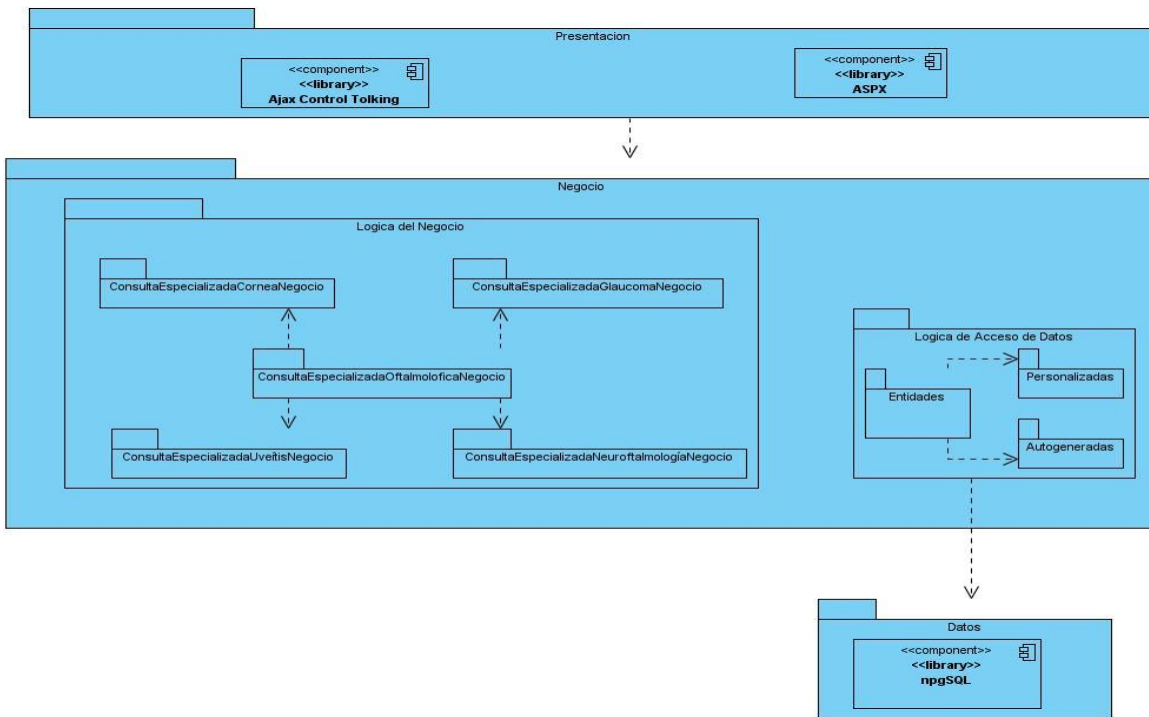
id_informe_operatorio	INTEGER	Identificador del informe operatorio
id_clasificacion_cirugia	INTEGER	Tipo de cirugía que se practicara al paciente
id_salon	INTEGER	Numero del salón donde se realizará la cirugía
id_anuncio_operatorio	INTEGER	Identificador del anuncio operatorio
id_clas_cirugia_limpieza	INTEGER	Identificador de la clasificación de la cirugía
id_funcionario_anestesista	INTEGER	Identificador del anestesista
id_funcionario_enfermera	INTEGER	Identificador de la enfermera
id_funcionario_cirujano	INTEGER	Identificador del cirujano
id_funcionario_tecnico	INTEGER	Identificador del técnico
id_persona	INTEGER	Identificador del paciente
turno_quirurgico	INTEGER	Numero del turno quirúrgico
especificaciones	VARCHAR	Aspectos de importancia
biopsia	BOOLEAN	Recoge si se le realizó la biopsia
indicaciones_postoperatorias	VARCHAR	Recoge la indicaciones que se le realizaron al paciente
observaciones	VARCHAR	Recoge las observaciones realizadas por el especialista
cirugia_trabajosa	BOOLEAN	Guarda si la cirugía es trabajosa o no
causas	VARCHAR	Recoge las causa del porque la cirugía fue trabajosa
fecha_hora_operacion	TIMESTAMP	Recoge la fecha en que se realizó la operación
cirugia_ambulatoria	BOOLEAN	Recoge si la cirugía es ambulatoria o no
id_tipo_cirugia	INTEGER	Identificador de la cirugía
id_subservicio	INTEGER	Identificador del servicio que va a operar esa patología

3.3 Vista de Implantación

La vista de implementación muestra el empaquetado físico de las partes reutilizables de un software en unidades sustituibles, llamadas componentes. Esta vista muestra los elementos físicos del sistema mediante componentes, así como sus interfaces y dependencias entre componentes. Estos últimos son piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas.

Un diagrama de componentes describe la descomposición física del sistema software en componentes, a efectos de construcción y funcionamiento. (27)

Diagrama de componentes



En este capítulo se estableció una valoración del diseño propuesto por el analista y su estructura, describiendo las clases y casos de uso correspondientes. De cada clase se ejecutó una breve descripción de sus atributos y métodos, y se obtuvo el modelo de datos del y el diagrama de componentes.

Capítulo 4. Validación de la solución propuesta

En el presente capítulo se aborda sobre una de las fases más importante en el ciclo de vida de un software que no es más que la fase de prueba. De la misma se describe sus características, objetivo y los métodos que posee, además de la importancia que contiene esta fase.

Una de las últimas fases del ciclo de vida antes de entregar un software para su explotación, es la fase de pruebas. El objetivo específico de la fase es encontrar cuantos más errores, mejor ya que probar un programa es ejercitarlo con la peor intención a fin de encontrarle errores que aun no son descubiertos para así lograr un software con la mejor calidad posible.

Existen dos tipos de pruebas, las de caja blanca y las pruebas de caja negra, la primera realizan un examen minucioso de detalles y procedimientos y la última realiza pruebas sobre la interfaz del software y precisamente es la prueba de caja negra la que realizamos al sistema.

4.1 Pruebas de Caja negra

Las pruebas de caja negra se centran en lo que se espera de un software, es decir, intentan encontrar casos en que el mismo no se atiene a su especificación. Para ello se realizan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el sistema por dentro.

Existen varios métodos de prueba de caja negra entre los que se tienen los Métodos de prueba basados en grafos, Análisis de valores límite, Prueba de la tabla ortogonal, Adivinando el error y Partición equivalente.

Entre todos estos métodos de caja negra el que se utiliza para desarrollarle las pruebas al sistema es el método de Partición equivalente. El mismo divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El objetivo de partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software.

4.2 Descripción de los Casos de Pruebas

Caso de prueba Crear hoja de consulta de Uveítis

Escenario	Descripción	MC	OMC	HEA	AM	ER	EC	EI	EH	EP	OA	HT	Ob	APO	EO	CIE	CQ	Esp	OIP	HS y HV	ACU	MP	A	Respuesta del sistema	Flujo central sistema	
EC 1.1 Crear hoja de consulta de Uveítis	Crear una hoja de consulta de Uveítis satisfactoriamente.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	Crea la hoja de consulta de Uveítis y se visualiza la interfaz ver detalles. • Menú Principal • Selecciona el paciente • Hoja de la consulta	
EC 1.2 Cancelar operación	Cancelar la opción de Crear hoja de consulta de Uveítis	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Regresa a la vista anterior.	
EC 1.3 Existen datos incompletos	Existen campos incompletos.	I	I	V	V	I	I	I	I	I	I	I	V	I	I	I	I	I	V	V	V	V	V	V	Muestra un indicador sobre los campos incompletos.	
EC 1.3 Existen datos incorrectos	Existen campos incorrectos	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	V	V	I	I	Muestra un indicador sobre los campos	

Tabla 4.2.1 CP Crear hoja de consulta de Uveítis

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Motivo de consulta	Son chekbox	No	Se selecciona el motivo o los motivos de consulta.
2	Otros motivos de consulta	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
3	Historia de la enfermedad actual (HEA)	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.

Capítulo 4. Validación de la solución propuesta.

4	Alergia medicamentos a	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
5	Enfermedades Renales	Son chekbox	No	Se selecciona al menos una enfermedad.
6	Enfermedades Cardiovasculares	Son chekbox	No	Se selecciona al menos una enfermedad.
7	Enfermedades Infeciosas	Son chekbox	No	Se selecciona al menos una enfermedad.
8	Enfermedades Hematológicas	Son chekbox	No	Se selecciona al menos una enfermedad.
9	Enfermedades Psiquiátricas	Son chekbox	No	Se selecciona al menos una enfermedad.
10	Otros antecedentes	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
11	Hábitos Tóxicos	Son chekbox	No	Se selecciona al menos un hábito
12	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
13	Antecedente Patológico Oftalmológico	Son chekbox	No	Se selecciona al menos un antecedente
14	Exámenes Oftalmológicos	Son chekbox	No	Se selecciona al menos un examen
15	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
16	Clasificación Internacional de Enfermedades (CIE)	Son chekbox	No	Se selecciona al menos una clasificación.
17	Conducta a Seguir	Lista desplegable	No	Se escoge la conducta a seguir.

Capítulo 4. Validación de la solución propuesta.

18	Especialista	Lista desplegable	No	Se escoge el especialista.
19	Otra impresión diagnóstica	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
20	Historia social y hábitos de vida	Son checkbox	Si	Se selecciona al menos una historia con un hábito
21	Antecedente de consulta de Uveítis	Un cuadro de texto	Si	Admite letras, y números se puede escribir o no.
22	Medicación previa	Un cuadro de texto	Si	Admite letras y números, se puede escribir o no.
23	Alergias	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.

Caso de prueba: Crear hoja de consulta de Neuroftalmología

Escenario	Descripción	MC	OMC	HEA	AM	ER	EC	EI	EH	EP	OA	HT	Ob	APO	EO	TMI	TMM	OT	CIE	CQ	Esp	OIP	Respuesta del sistema	Flujo central	
EC 1.1 Crear hoja de consulta de Neuroftalmología	Crear una hoja de consulta de Neuroftalmología satisfactoriamente.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	Crea la hoja de consulta de Neuroftalmología y se visualiza la interfaz ver detalles.	<ul style="list-style-type: none"> • Menú Principal • Selecciona el paciente • Hoja de la consulta
EC 1.2 Cancelar operación	Cancelar la opción de Crear hoja de consulta de Neuroftalmología	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Regresa a la vista anterior.	
EC 1.3 Existen datos incompletos	Existen campos incompletos.	I	I	V	V	I	I	I	I	I	I	I	V	I	I	V	V	V	I	I	I	I	V	Muestra un indicador sobre los campos incompletos.	
EC 1.3 Existen datos incorrectos	Existen campos incorrectos	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	Muestra un indicador sobre los campos	

Tabla 4.2.2 CP Crear hoja de consulta de Neuroftalmología

Capítulo 4. Validación de la solución propuesta.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Motivo de consulta	Son chekbox	No	Se selecciona el motivo o los motivos de consulta.
2	Otros motivos de consulta	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
3	Historia de la enfermedad actual (HEA)	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
4	Alergia a medicamentos	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
5	Enfermedades Renales	Son chekbox	No	Se selecciona al menos una enfermedad.
6	Enfermedades Cardiovasculares	Son chekbox	No	Se selecciona al menos una enfermedad.
7	Enfermedades Infecciosas	Son chekbox	No	Se selecciona al menos una enfermedad.
8	Enfermedades Hematológicas	Son chekbox	No	Se selecciona al menos una enfermedad.
9	Enfermedades Psiquiátricas	Son chekbox	No	Se selecciona al menos una enfermedad.
10	Otros antecedentes	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
11	Hábitos Tóxicos	Son chekbox	No	Se selecciona al menos un hábito
12	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
13	Antecedente Patológico Oftalmológico	Son chekbox	No	Se selecciona al menos un antecedente
14	Exámenes Oftalmológicos	Son chekbox	No	Se selecciona al menos un examen
15	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
16	Tratamiento médico inicial	Es un chekbox	Si	Se selecciona el tratamiento
17	Tratamiento médico de mantenimiento	Es un chekbox	Si	Se selecciona el tratamiento

Capítulo 4. Validación de la solución propuesta.

19	Otros tratamientos	Un cuadro de texto.	Si	Admite solo letras, se puede escribir o no
20	Clasificación Internacional de Enfermedades (CIE)	Son checkbox	No	Se selecciona al menos una clasificación.
21	Conducta a Seguir	Lista desplegable	No	Se escoge la conducta a seguir.
22	Especialista	Lista desplegable	No	Se escoge el especialista.
23	Otra impresión diagnóstica	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.

Caso de prueba: Crear hoja de consulta de Glaucoma

Escenario	Descripción	MC	OMC	HEA	AM	ER	EC	EI	EH	EP	OA	HT	Ob	APO	EO	Ob	CIE	CQ	Esp	OIP	Respuesta del sistema	Flujo central	
EC 1.1 Crear hoja de consulta de Glaucoma	Crear una hoja de consulta de Glaucoma satisfactoriamente.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	Crea la hoja de consulta de Glaucoma y se visualiza la interfaz ver detalles.	<ul style="list-style-type: none"> • Menú Principal • Selecciona el paciente • Hoja de la consulta
EC 1.2 Cancelar operación	Cancelar la opción de Crear hoja de consulta de glaucoma.	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Regresa a la vista anterior.		
EC 1.3 Existen datos incompletos	Existen campos incompletos.	I	I	V	V	I	I	I	I	I	I	I	V	I	I	V	I	I	I	V	Muestra un indicador sobre los campos incompletos.		
EC 1.3 Existen datos incorrectos	Existen campos incorrectos	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	Muestra un indicador sobre los campos incorrectos.		

Tabla 4.2.3 CP Crear hoja de consulta de Glaucoma

Capítulo 4. Validación de la solución propuesta.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Motivo de consulta	Son chekbox	No	Se selecciona el motivo o los motivos de consulta.
2	Otros motivos de consulta	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
3	Historia de la enfermedad actual (HEA)	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
4	Alergia a medicamentos	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
5	Enfermedades Renales	Son chekbox	No	Se selecciona al menos una enfermedad.
6	Enfermedades Cardiovasculares	Son chekbox	No	Se selecciona al menos una enfermedad.
7	Enfermedades Infecciosas	Son chekbox	No	Se selecciona al menos una enfermedad.
8	Enfermedades Hematológicas	Son chekbox	No	Se selecciona al menos una enfermedad.
9	Enfermedades Psiquiátricas	Son chekbox	No	Se selecciona al menos una enfermedad.
10	Otros antecedentes	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
11	Hábitos Tóxicos	Son chekbox	No	Se selecciona al menos un hábito
12	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
13	Antecedente Patológico Oftalmológico	Son chekbox	No	Se selecciona al menos un antecedente
14	Exámenes Oftalmológicos	Son chekbox	No	Se selecciona al menos un examen
15	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
16	Clasificación Internacional de Enfermedades (CIE)	Son chekbox	No	Se selecciona al menos una clasificación.

Capítulo 4. Validación de la solución propuesta.

17	Conducta a Seguir	Lista desplegable	No	Se escoge la conducta a seguir.
18	Especialista	Lista desplegable	No	Se escoge el especialista.
19	Otra impresión diagnóstica	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.

Caso de Prueba: Crear hoja de consulta de Cornea

Escenario	Descripción	MC	OMC	HEA	AM	ER	EC	EI	EH	EP	OA	HT	Ob	APO	EO	CIE	CQ	Esp	OIP	Respuesta del sistema	Flujo central
EC 1.1 Crear hoja de consulta de Cornea	Crear una hoja de consulta de Cornea satisfactoriamente.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	Crea la hoja de consulta de Cornea y se visualiza la interfaz ver detalles.	<ul style="list-style-type: none"> • Menú Principal • Selecciona el paciente • Hoja de la consulta
EC 1.2 Cancelar operación	Cancelar la opción de Crear hoja de consulta de Cornea	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Regresa a la vista anterior.	
EC 1.3 Existen datos incompletos	Existen campos incompletos.	I	I	V	V	I	I	I	I	I	I	I	V	I	I	I	I	I	V	Muestra un indicador sobre los campos incompletos.	
EC 1.3 Existen datos incorrectos	Existen campos incorrectos	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	Muestra un indicador sobre los campos	

Tabla 4.2.4 CP Crear hoja de consulta de Córnea

Capítulo 4. Validación de la solución propuesta.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Motivo de consulta	Son checkbox	No	Se selecciona el motivo o los motivos de consulta.
2	Otros motivos de consulta	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
3	Historia de la enfermedad actual (HEA)	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
4	Alergia a medicamentos	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.
5	Enfermedades Renales	Son checkbox	No	Se selecciona al menos una enfermedad.
6	Enfermedades Cardiovasculares	Son checkbox	No	Se selecciona al menos una enfermedad.
7	Enfermedades Infecciosas	Son checkbox	No	Se selecciona al menos una enfermedad.
8	Enfermedades Hematológicas	Son checkbox	No	Se selecciona al menos una enfermedad.
9	Enfermedades Psiquiátricas	Son checkbox	No	Se selecciona al menos una enfermedad.
10	Otros antecedentes	Un cuadro de texto.	No	Admite solo letras, se puede escribir el motivo de consulta.
11	Hábitos Tóxicos	Son checkbox	No	Se selecciona al menos un hábito
12	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
13	Antecedente Patológico Oftalmológico	Son checkbox	No	Se selecciona al menos un antecedente
14	Exámenes Oftalmológicos	Son checkbox	No	Se selecciona al menos un examen
15	Observaciones	Un cuadro de texto.	Si	Admite solo letras, se puede escribir el motivo de consulta.
16	Clasificación Internacional de Enfermedades (CIE)	Son checkbox	No	Se selecciona al menos una clasificación.

17	Conducta a Seguir	Lista desplegable	No	Se escoge la conducta a seguir.
18	Especialista	Lista desplegable	No	Se escoge el especialista.
19	Otra impresión diagnóstica	Un cuadro de texto	Si	Admite solo letras, se puede escribir o no.

En este capítulo se abordó sobre el método utilizado para realizar las pruebas de software, y se expusieron algunas de las técnicas existentes para el diseño de casos de prueba, entre ellas la técnica de partición de equivalencia, que fue la seleccionada para llevar a cabo el desarrollo de los casos de prueba que se utilizarán para examinar las funcionalidades y detectar determinados errores presentes en las mismas. Y por último, se muestra la descripción de algunos casos de prueba diseñados.

Conclusiones

1. El análisis realizado a los sistemas existentes en el mundo vinculado a los servicios oftalmológicos de Neuroftalmología, Glaucoma, Uveítis y Cornea del Bloque Quirúrgico Oftalmológico demostró que estos no cumplen con las funcionalidades necesarias, aunque presentaban funcionalidades que fueron tomadas en cuenta.
2. Las tecnologías patrones y herramientas utilizados en el desarrollo del trabajo permitieron la realización de un sistema flexible y robusto.
3. La aplicación de la arquitectura definida por el Departamento de Sistemas de Gestión Hospitalaria, permitió el desarrollo de los servicios oftalmológicos de Neuroftalmología, Glaucoma, Uveítis y Cornea del Bloque Quirúrgico Oftalmológico.
4. La aplicación correcta de las pautas de diseño definidas en el proyecto Bloque Quirúrgico Oftalmológico hizo posible la uniformidad y homogeneidad en las interfaces.
5. El desarrollo de los servicios oftalmológicos de Neuroftalmología, Glaucoma, Uveítis y Cornea del Bloque Quirúrgico Oftalmológico que se obtuvo como respuesta de la optimización de la solución existente, permitirá mayor rapidez en las consultas oftalmológicas de las instituciones hospitalarias, al posibilitar gestionar la información de las historias clínicas del paciente, así como el informe y el anuncio operatorio.

Recomendaciones

1. Adicionar las funcionalidades:
 - ✚ Crear solicitud de análisis de laboratorio
 - ✚ Asignar cita de interconsulta.

A partir de la integración con el sistema GALEN de la empresa Softel.

2. Realizar la gestión del especialista mediante diferentes criterios de búsqueda brindando una interfaz más amigable que agilice el proceso.

Referencias bibliográficas.

1. **Aguilera González, Julio C. Dr.** INFORMÁTICA MÉDICA. EL CUIDADO MULTIDISCIPLINARIO. [En línea]
<http://www.encolombia.com/medicina/enfermeria/enfermeria6203-editorial.htm>.
2. **Drs. Ariel Delgado Ramos, Mirna Cabrera Hernández, Alfredo Rodríguez Díaz.** ESTRATEGIA DE INFORMATIZACIÓN DEL SISTEMA NACIONAL DE SALUD. [En línea] enero de 2011.
<http://informatica2009.sld.cu/Members/mirnacabrera/estrategia-de-informatizacion-del-sistema-nacional-de-salud/>.
3. **Alzamora, Dr. César Larrain.** IV OFTALMOLOGÍA . [En línea] enero de 2011.
http://sisbib.unmsm.edu.pe/bibvirtual/libros/medicina/cirugia/tomo_iv/glaucoma.htm.
4. Doctissimo. [En línea] diciembre de 2010. <http://salud.doctissimo.es/diccionario-medico/uveitis.html>.
5. TheFreeDictionary. [En línea] diciembre de 2010. <http://es.thefreedictionary.com/cornea>.
6. Programas o Planes de las distintas especialidades médicas. [En línea] diciembre de 2010.
http://www.portalesmedicos.com/plan_programa_especialidad/ofthalmologia_1.htm.
7. Definicion de . [En línea] diciembre de 2010. <http://definicion.de/cirugia/>.
8. **Montes, Dr. Abelardo Cuadrado.** IV Oftalmología . [En línea] enero de 2011.
http://sisbib.unmsm.edu.pe/bibvirtual/libros/medicina/cirugia/tomo_iv/neurooftal.htm.
9. Opera, sistema de gestión de la cirugía. [En línea] diciembre de 2010. <http://www.chca.ca/opera.php?lang=es>.
10. Net Clínicas software. [En línea] diciembre de 2010. <http://www.netclinicas.com/>.
11. VisionDat software. [En línea] diciembre de 2010. <http://www.visiondatsoftware.com/pages/home.aspx>.
12. Emetropia. [En línea] diciembre de 2010. <http://www.emetropia.com/>.
13. OFTALMOSALUS. [En línea] diciembre de 2010. <http://www.ofthalmosalus.com/DefaultSalus.aspx>.
14. ActualOftalmo software profesional. [En línea] diciembre de 2010.
<http://www.actualsoft.com.ar/pantoftalmo.htm>.
15. **Chacón, Ings. David Vasallo Lominadze y Disnayle Jorge.** *Desarrollo de la Consulta General de Oftalmología.* junio de 2010.
16. MEDISYS. Versión Cliente-Servidor. [En línea] enero de 2010.
<http://www.sld.cu/instituciones/cedisap/Medclien.htm>.

17. Scribd. [En línea] enero de 2011. <http://www.scribd.com/search?query=RUP>.
18. **López Pérez, Carmelo Vicepresidente primero del CIIRM.** *Modelo de Madurez de la Capacidad del Software1.*
19. aplicaciones-web. [En línea] www.hooping.net/.../aplicaciones-web-146.aspx.
20. **Rabelo Rodríguez, Alexander.** *CESIM_BQO_0120_Arquitectura de software v2.0.* 2011.
21. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] enero de 2011. <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
22. **Miló Pérez, Joselín Ing. Hernández Carralero, Yandy Ing.** *IMPLEMENTACIÓN DE LOS SERVICIOS CIRUGÍA DEL CRISTALINO, CIRUGÍA REFRACTIVA Y PTERIGIUM DEL BLOQUE QUIRÚRGICO OFTALMOLÓGICO Y LOS MÓDULOS DE ADMINISTRACIÓN Y CONFIGURACIÓN.*
23. **Arevalo, Ing.** *mplementación del Módulo Bloque Quirúrgico Oftalmológico del Sistema de Información Hospitalaria.* 2007.
24. Microsoft. [En línea] febrero de 2008. <http://msdn.microsoft.com/es-es/library/system.web.ui.webcontrols.objectdatasource%28VS.80%29.aspx>.
25. sophia. *sophia.* [En línea] http://sophia.javeriana.edu.co/~cbustaca/Arquitectura%20Software/Presentaciones/arquitecturas_software02.pdf.
26. Material_Desarrollo_Software. *Material_Desarrollo_Software.* [En línea] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISCLASES_A12.pdf.
27. elai. *elai.* [En línea] <http://www.elai.upm.es:8009/spain/Asignaturas/InfoInd/apuntesAOOD/cap5UMLDinamicoImpl.pdf>.
28. **Ramos, Prof.Ariel Delgado.** *Informática en la salud pública cubana .* [En línea] diciembre de 2010. http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.

Bibliografía

1. **Aguilera González, Julio C. Dr.** INFORMÁTICA MÉDICA. EL CUIDADO MULTIDISCIPLINARIO. [En línea] <http://www.encolombia.com/medicina/enfermeria/enfermeria6203-editorial.htm>.
2. **Drs. Ariel Delgado Ramos, Mirna Cabrera Hernández, Alfredo Rodríguez Díaz.** ESTRATEGIA DE INFORMATIZACIÓN DEL SISTEMA NACIONAL DE SALUD. [En línea] enero de 2011. <http://informatica2009.sld.cu/Members/mirnacabrera/estrategia-de-informatizacion-del-sistema-nacional-de-salud/>.
3. **Alzamora, Dr. César Larrain.** IV OFTALMOLOGÍA . [En línea] enero de 2011. http://sisbib.unmsm.edu.pe/bibvirtual/libros/medicina/cirugia/tomo_iv/glaucoma.htm.
4. Doctissimo. [En línea] diciembre de 2010. <http://salud.doctissimo.es/diccionario-medico/uveitis.html>.
5. TheFreeDictionary. [En línea] diciembre de 2010. <http://es.thefreedictionary.com/cornea>.
6. Programas o Planes de las distintas especialidades médicas. [En línea] diciembre de 2010. http://www.portalesmedicos.com/plan_programa_especialidad/oftalmologia_1.htm.
7. Definicion de . [En línea] diciembre de 2010. <http://definicion.de/cirugia/>.
8. **Montes, Dr. Abelardo Cuadrado.** IV Oftalmología . [En línea] enero de 2011. http://sisbib.unmsm.edu.pe/bibvirtual/libros/medicina/cirugia/tomo_iv/neurooftal.htm.
9. Opera, sistema de gestión de la cirugía. [En línea] diciembre de 2010. <http://www.chca.ca/opera.php?lang=es>.
10. Net Clínicas software. [En línea] diciembre de 2010. <http://www.netclinicas.com/>.
11. VisionDat software. [En línea] diciembre de 2010. <http://www.visiondatsoftware.com/pages/home.aspx>.
12. Emetropia. [En línea] diciembre de 2010. <http://www.emetropia.com/>.
13. OFTALMOSALUS. [En línea] diciembre de 2010. <http://www.oftalmosalus.com/DefaultSalus.aspx>.
14. ActualOftalmo software profesional. [En línea] diciembre de 2010. <http://www.actualsoft.com.ar/pantoftalmo.htm>.
15. **Chacón, Ings. David Vasallo Lominadze y Disnayle Jorge.** *Desarrollo de la Consulta General de Oftalmología.* junio de 2010.
16. MEDISYS. Versión Cliente-Servidor. [En línea] enero de 2010. <http://www.sld.cu/instituciones/cedisap/Medclien.htm>.
17. Scribd. [En línea] enero de 2011. <http://www.scribd.com/search?query=RUP>.

18. **López Pérez, Carmelo Vicepresidente primero del CIIRM.** *Modelo de Madurez de la Capacidad del Software1.*
19. aplicaciones-web. [En línea] www.hooping.net/.../aplicaciones-web-146.aspx.
20. **Rabelo Rodríguez, Alexander.** *CESIM_BQO_0120_Arquitectura de software v2.0.* 2011.
21. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] enero de 2011. <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
22. **Miló Pérez, Joselín Ing. Hernández Carralero, Yandy Ing.** *IMPLEMENTACIÓN DE LOS SERVICIOS CIRUGÍA DEL CRISTALINO, CIRUGÍA REFRACTIVA Y PTERIGIUM DEL BLOQUE QUIRÚRGICO OFTALMOLÓGICO Y LOS MÓDULOS DE ADMINISTRACIÓN Y CONFIGURACIÓN.*
23. **Arevalo, Ing.** *mplementación del Módulo Bloque Quirúrgico Oftalmológico del Sistema de Información Hospitalaria.* 2007.
24. Microsoft. [En línea] febrero de 2008. <http://msdn.microsoft.com/es-es/library/system.web.ui.webcontrols.objectdatasource%28VS.80%29.aspx>.
25. sophia. *sophia.* [En línea] http://sophia.javeriana.edu.co/~cbustaca/Arquitectura%20Software/Presentaciones/arquitecturas_software02.pdf.
26. Material_Desarrollo_Software. *Material_Desarrollo_Software.* [En línea] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISCLASES_A12.pdf.
27. elai. *elai.* [En línea] <http://www.elai.upm.es:8009/spain/Asignaturas/InfoInd/apuntesAOOD/cap5UMLDinamicoImpl.pdf>.
28. **Ramos, Prof.Ariel Delgado.** *Informática en la salud pública cubana .* [En línea] diciembre de 2010. http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.